

računari

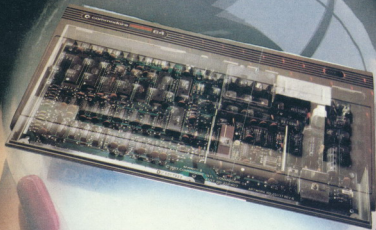
Specijalno izdanje časopisa „Galaksija“
Izdaje BIGZ – OOUR „Duga“

anđelko
zgorelec
u londonu
svašta novo

Izlazi jedanput
mesečno
cena 300 dinara
februar 1986.

12

kad
„spektrum“
zataji
smrtonosna
dijagnostika



dejan ristanović
kako rešiti „elitu“

pokice koje život znače
200 poukova za „komodor“
50 poukova za „spektrum“

12

Izlazi jednom mesečno
Y Izdaje BIGZ OOUR „Duga“
računari
Specijalno izdanje časopisa „Galaksija“

Cena 300 dinara / februar 1986.

Izdaje
Beogradski izdavačko-grafički zavod
OOOUR Novinska delatnost „Duga“
11000 Beograd
Bulevar vojvode Mišica 17

Telefoni
650-161 (redakcija)
650-526 (prodaja)
651-793 (propaganda)

Generalni direktor
Dobrivoj Petrović

Direktor OOOUR „Duga“
Braljubo Babić
Glavni i odgovorni urednik
Gavrilo Vučković
Urednik izdanje
Jovta Regasek

Tehnički urednik
Mirko Popov

Redakcija časopisa „Galaksija“
Tanasje Gavranović, pomoćnik
glavnog i odgovornog urednika
Esad Jakupović, zamjenik glavnog
i odgovornog urednika
Aleksandar Milinković, urednik
Jovta Regasek, urednik
Zorka Simović, sekretar redakcije
Sjđan Stojanović, novinar
Gavrilo Vučković, glavni i odgovorni
urednik

Stručna saradnja
Dejan Ristanović
Dušan Slavić

Nevenka Spalević
Anđelka Zgonicec
Mihajlo Karapandžić

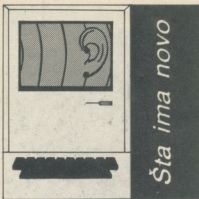
Autori tekstova:
Nada Aleksić
Branko Đaković
Donat Greber
Branko Hebrang
Dorđe Janković
Mihajlo Karapandžić
Vladimir Kostić
Vladimir Kristonošić
Sasa D. Kovačević
Blažimir Miše
Dejan Muhamedagić
Ivan Nador
Radimir Nikočajev
Zoran Obradović
Miodrag Potkonjak
Dejan Ristanović
Jelena Rupnik
Dušan Slavić
Jovan Skuljan
Nevenka Spalević
Srdan Stakić
Zvonimir Vistrička

Fotografije
Vladimir Simović
Ilustracije
Miodrag Marković

Prevodioci
Esad Jakupović
Ksenija Pješčić-Lebedinski
Izdavački savet „Galaksije“
Dr Rudi Debijadi, prof. dr Branislav Dimitrijević (predsednik), Radovan Drasković, Tanasje Gavranović, Zvonar Glišić, Esad Jakupović, Velizar Masić, Nikola Pajić, Željka Perunović, prof. dr Momočić Ristić, Vlada Ristić, dr inž. Milorad Teofilović, Vidojko Veličković, Velimir Vesović, Miloje Vuković
Štampa
Beogradsko izdavačko-grafički zavod
11000 Beograd, Bulevar vojvode Mišića 17
Ziro-račun kod SDK 60802-833-2463
Divizni račun kod Beobanke
60811-620-6-82701-999-01066
Za inostranstvo cena dvostruka (400 D,
2,50 US \$, 6,50 DM, 45 Sch, 5,50 Sfrs,
20 Ffrs)
Na osnovu mišljenja Republičkog
sekretarijata za kulturu broj 413-77-03 i
„Službenog glasnika“ broj 26/72, ovo
izdanje oslobođeno je poreza na promet.

sadržaj

- 3/ šta ima nova
- 4/ razglednica iz londona
- 8/ pogled izbliza
licem u lice sa „atarijem“
- 11/ računari iz mog ugla
mi sve možemo
- 12/ domaći računari
pc iz podravine
- 14/ računari u razgovoru
zašto smo pali na niske grane
- 16/ obrada ideja
misli kao što pišeš
- 17/ istorija računara
ibm i sedam patuljaka
- 20/ dejanove pitalice
nagrada premetaljka
- 21/ umetnost programiranja
holandska zastava
- 22/ operativni sistemi
cp/m
- 26/ programiranje na mašincu
sažimanje teksta
- 28/ kako se koristi
mons
- 33/ umesto umetka
„amstradov“ kalkulator
- 37/ majstorije na računaru/spektrum
operacija fill
- 39/ majstorije na računaru/amstrad
kraljevstvo za registar
- 42/ akcije
sam svoj programer
- 44/ matematički softver
logaritamska funkcija
- 46/ udruženi programeri
jezički procesori
- 49/ programiranje u bejziku
svirka na „amstradu“
- 52/ put u središte rom-a
procedure u akciji
- 56/ pokice koje život znače
- 58/ elitna groznica
- 60/ spektrumovci u nevolji
smrtonosna dijagnostika



Autori: Nada Aleksić, Ivan Nador, Branko Đaković, Vladimir Krstošić, Donat Greber, Dejan Ristanović

Komodor dizel

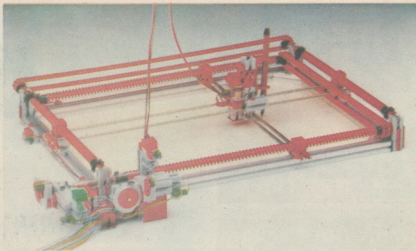
Firma Komodor izbacila je u Evropi još jedan nov model računara Commodore 128 D, duhovito nazvan 128 dizel. Reč je o modelu 128 koji je, zajedno sa novim diskom 1571, smešten u odlično dizajniranu kutiju, dok je samo tastatura odvojena, ali se može uvući u posebno ležište koje se nalazi na dnu same kutije, što olakšava prenos. Tastatura je ista kao kod PC-128, koja je nedavno proglašena za najbolju tastaturu kod kućnih računara, a ceo uređaj prilično podseća na IBM-PC-a. Računar se može dobiti i sa ugrađenim hard diskom od 20 Mb, što ga, zajedno sa odličnim CP/M-om i 80 znakova u redu čini vrlo interesantnim za poslovnu primenu. Cena uređaja je, međutim, možda najinteresantnija od svega — 1600 maraka (sa verovatnim padom u skoraj budućnosti) i predstavlja dosta primamljivamac. Razmislite dobro.

Japanska invazija

Invazija je počela, ali na svu (ne)reću samo na američko tržište personalnih računara. Nekoliko vodećih japanskih firmi je već izbacilo na tržište personalne računare koji su kompatibilni sa IBM Personal Computer, PC, a pri tome su znatno jeftiniji. Seiko Epson Corp. je startovao sa tri računara—Equity I, II, III koji odgovaraju PC-u PC/XT-u i PC AT-u, a prodaju se za \$1300, \$1695 i \$3495, čime je cena prepolovljena u odnosu na cene IBM-a. Sharp Electronics Corp. lansirao je (malog) PC-7000 koji nije puno veći od portabl tranzistora, ali za cenu od \$1795 opremljen je sa dva 5 1/4 inča floppy-diska, 320K RAM-a, displejom sa tečnim kristalom, serijskim i paralelnim portom. Toshiba America Inc. nudi računare težine 4kg, sa 512K RAM-a, displejom sa tečnim kristalom, 3 1/2 inča diskom i paralelnim portom. Mi, bar za sada, ne stremimo od invazije.

N. A.

šta ima novo



Ploter za narod

Čuvena firma za proizvodnju igračka Fischer, poznata po svojim još čuvenijim kockicama, uključila se u opštu računarsku groznicu: od nedavno proizvodi interfejs koji omogućuje kontrolu modela pomoću računara. Interfejs je, naravno, napravljen prvo za „komodor 64“, a kada je interesovanje postalo veće, prilagoden je i za ostale računare (MSX, „spektrum“, BBC). Pored interfejsa, u prodaju su pušteni i posebni računarski setovi sa delovima. Najinteresantniji komplet je ploter-skener. Nabavite komplet, sastavite ga, povežite ga sa interfejsom i računalom, učitajte softver i — dobili ste odličan ploter formata A4! Uređaj radi kao i svaki drugi „ozbiljan“ ploter —

pomera olovku u svim pravcima, diže je i spušta i, zaista, crta veoma precizno. Ako olovku zamenite s glavom za čitanje i promenite program, dobićete odlučan skener koji bi trebao da „čita“ sliku koja se nalazi na papiru. Osim ovog, može se nabaviti još nekoliko različitih kompleta za razne primene, posebno u robotici. Možete napraviti grafičku tablu ili, možda, sebi partnera za šah. Prvi interfejs i pojedini kompleti već su stigli u našu zemlju (i u redakciju „Računara“). Očekujte, dakle, detaljniji prikaz. Cena interfejsa je 250 nemačkih maraka, dok se cena kompleta kreće od 300—600. To, prilično, znači da možete da nabavite solidan ploter za ispod 700 maraka. Sačekajte sledeći broj „Računara“, pa onda donesite odluku.

V. Krstošić

Nova cena

Obaveštavam čitaoce da smo zbog povećanih troškova proizvodnje, pre svega cene papira i štamparskih usluga, bili primorani da cenu „Računara“ povećamo sa 250 na 300 dinara. Čitajte nas i dalje — bićemo još bolji!
Redakcija

I digital u trci

Digital Equipment je rešio da još jednom odgovori na izazov IBM-a. Ovog puta je u pitanju personalni kompjuter kompatibilan sa IBM-ovim PC-jem. Predsednik DEC-a i Kenneth Olsen (Kenneth H. Olsen) najavio je da će DEC-ov MS-DOS biti funkcionalniji i, na žalost, samim tim i nešto skupiji. Ako rešite da ga kupite, ne tražite ga u velikim robnim kućama, tamo se neće prodavati.

N. A.

Isto ali jeftinije

Tandy Corp. upravo počinje prodaju novog modela — 3000. Računar je kompatibilan sa IBM-ovim Personal Computerom AT, ali će koštati oko 1000 dolara manje. Osnovna cena je 2599 dolara. Proizvođač iz Fort Vorta će, istovremeno, utrošiti preko 80 miliona dolara da bi renovirao i učinio atraktivnijim prodavnice svog Radio Shack lanka i tako privukao kupce da se masovnije oduče za kupovinu Tandijevih personalaca iz kategorije koja se najbolje prodaje, a to su prema analitičarima tržišta kompjutera, oni koji koštaju ispod 3000 dolara.

N. A.

Sastavite PC

Onima koji nemaju dovoljno novca, a ne odustaju od IBM PC-a predlažemo da se odmah upišu na ubrzan kurs za sastavljanje hardvera. Za 1427 dolara može se kod ICS Inc. (Puna adresa za kontakt i kupovinu: ICS Inc., 8601 Aero Drive, San Diego, Calif. 92123 — tel. 619-279-0084) nabaviti celokupni hardver za PC.

N. A.



Andelko Zgorelec

Razglednica
iz Londona

u Londonu svašta novo

Neslavnim lansiranjem QL-a pre dve godine, januar mesec je, izgleda, postao tradicionalan datum za evropske premijere novih ili poboljšanih mikro-računara. U razmaku od samo nekoliko dana, ovog januara u Londonu su prikazani „amiga“, novi „mekintosh“ i četiri (ne baš sasvim) nova računara firme Eljorn (Acorn). Iako u znaku visoke tehnologije i izvanrednih specifikacija, svaki od ovih kompjutera na svoj način, a svi zajedno svojim visokim cenama, produžuju krizu koja razjeda industriju mikroročunara već više od godinu dana.

Prošla godina je za svet personalne informatike bila zaista crna. Mi smo već opširno pisali o poteškoćama koje su pogadale čak i velike na području mikroročunarstva. Najveća imena, poput Eljorna, Sinklera, Atarija i Komodora, ipak nisu propala. Neke od ovih kompanija se čak manje-više i uspešno bore sa krizom u industriji mikro-kompjutera. Prvi rezultati o predradnjičnoj prodaji računara bili su čak i nešto bolji nego što se to očekivalo. Organizacija za ispitivanje tržišta u Britaniji „International Data Consultants“ smatra da su pesimističke prognoze, davane još pre samo nekoliko meseci, da će se u 1985. godini prodati samo blizu 900 hiljada mikroročunara bile pogrešne i da ukupna prodaja prošle godine iznosi 1,1 milion malih računara (u rekordnoj 1984. godini prodati je čak 1,5 miliona računara). Prognoze za 1986. godinu su još crnje — doći će od pada za 30%, što praktično znači da će otići samo 800 hiljada kompjutera. No, treba sačekati beričetnu 1987. — eksperti IDC smatraju da će pojavom takozvanih „super“ personalnih računara, poput „amige“ i „atarjevih“ modela, prodaja znatno porasti.

Ružna kao pas

Ova tmurna predviđanja, međutim, kao da se ne odnose na firmu Amstrad —



njihovi računari idu i dalje kao alva. Ova firma zauzela je čak 25% britanskog tržišta i to na račun „staraca“ Sinklera, Eljorna i Komodora. Amstrad i dalje niže poslovne uspehe i dobija veliki publicitet: nedavno je „boss“ ove firme Alan Šuger proglašen kompjuterskom ličnošću godine, a model PWC 8256, koji je i dosta teško i ovde nabaviti, jer se odlično prodaje, uskoro će se naći na policama američkih robnih kuća — najveća američka trgovačka mreža „Sears-Roebuck“ preuzima. U početku 10 hiljada ovih računara mesečno. Ako prodaja dobro krene, „Sears-Roebuck“ bi za dve godine trebalo da kupi od Amstrada čak pola miliona tih malih računara.

Od ostalih računara u toj glavnoj prodajnoj sezoni išao je jako dobro i „spektrum“, koji i dalje zauzima vodeće mesto po broju prodatih mašina. Spektrum, uz put, izgleda jedino još u Jugoslaviji smatraju ozbiljnom mašinom i koriste ga za programiranje. U Engleskoj je, definitivno, prešao u kategoriju igračka — ozbiljniji softver više niko niti otkupljuje niti prodaje.

Komodorovi računari su i dalje popularni, ali je firma ipak u velikoj finansijskoj krizi. Početkom januara je zatvorena Komodorova fabrika u gradiću Korbi (Corby) u srednjoj Engleskoj, sa gubitkom 300 radnih mesta, a proizvodnja je prebačena u Nemačku. (Izgleda da će Komodor sada morati da vrati britanskoj vladi bespovratni zajam od nekoliko miliona funti, koji je dobio kao pomoć za otvorenje ove fabrike). Iz Amerike javljaju da je Komodor zatvorio i svoju fabriku za proizvodnju čipova u Silicijumskoj dolini i prebacio proizvodnju u jednu drugu fabriku na istočnoj obali Amerike. Prema pisanju američkih poslovnih listova, dugovi su toliko veliki da ima malih izgleda da se ovaj pionir personalnog računarstva održi u sadašnjem obliku. I da nema „amige“, Komodor bi, izgleda, već potonuo.

Kada smo već kod tog revolucionarnog računara, da spomenemo da prvi korisnici nemaju baš najbolje mišljenje o njemu. Prema anketi elektronskog biltena „Newsbytes“ na „The Source“ (elektronska datoteka u Americi, poput Prestela-Micro-

neta u Britaniji), „amigi“ je dodeljena titula „dog“ (pas) zajedno sa uzasnim PC Juniorom. To je sve iznenadilo. Treba, međutim, ponekad verovati i korisnicima. „Amiga“ je, inače, i službeno prikazana u Britaniji na sajmu „Which Computer“ održanom sredinom januara u Birminghamu (Birmingham) i uskoro će biti puštena u prodaju, iako osnovni softver još uvek nije obezbeđen.

Više čipova nego čips(ov)a

Lansiranje četiri nova Eljornova računara iz serije „Master“, koji su prikazani javnosti takođe sredinom januara u Londonu, ne predstavlja, na žalost, ništa veselij događaj. Novi računari su, uglavnom, zasnovani na poslovnom Eljornovom ABC računaru prikazanom još pre više od godinu dana, i u njima se, zapravo, krije stari, dobri BBC B sa takođe poznatim proširenjima — dodatnim procesorima 6502, 80186 ili 32016. Tehničke detalje prepuštamo ekspertu u redakciji za BBC računare Dejanu Ristanoviću, koji za sledeći broj priprema prikaz sa obiljem komentara. Za najestrpljivije ljubitelje Eljornovih mašina pripremlili smo uporednu tabelu iz koje će im sve biti jasno.

Na konferenciji za štampu bilo je skoro najmanje novinara kompjuterskih časopisa, jer nisu bili namerno pozvani, a najviše iz listova opšteg profila. Vaš dopisnik je nekako uspeo da provuče sebe i Vladimira Kostića, koji je tih dana bio u trgovinsko-programerskoj misiji u Londonu. Predstavnik časopisa „Practical Electronics“ i saradnik „Računara“ Eljornu nisu izgledali naročito opasno da bi im pravio smetnje.

Konferenciju je otvorio generalni direktor Eljorna Brian Long (Brian Long) sa toliko samohvale i čestitki kao da ova kompanija pravi milione funti profita. Opšti je utisak da su sva četiri nova računara preskupa i da dolaze prekasno, mada će se, ipak, prodati zbog velikog uticaja i popularnosti koju Eljorn ima u školama i među sadašnjim korisnicima BBC računara. Cene su gotovo neverovatno visoke — kao da Eljorn ne shvata da su druge kompanije

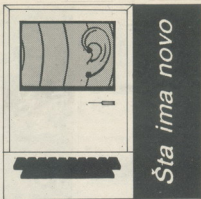
u poslednjih nekoliko godina drastično snizile cene uvođenjem složenijih i modernijih čipova, što je smanjilo njihov ukupan broj. Kako piše dopisnik mančesterskog "The Guardian", u ovim novim računarima „ima više čipova nego što ih izlētnici pojedu na putu za popularno letovalište Blekpu" (u engleskom "chip" je naziv i za pržene krompiriće — pomfrit). Dok su ostale firme u poslednje četiri godine svojim računarima snizile cene i za 300%, Ejrkorn je u to vreme „uspeo" da poveća cenu za model BBC B od 320 na — za nijednostavniji "Master" računar — 490 funti. Takva neposlovnost ili, da budemo otvoreni, drskost, sigurno nije zabeležena u istoriji mikroracunarstva.

Da je neka druga firma imala u svojim rukama model BBC B — koji je, da se razumemo odličan računar i koji poseduje gotovo pola redakcije "Računara", uključujući i vašeg dopisnika — njegova cena bi sada bila oko 150 funti. Sa takvom cenom BBC B bi imao šanse da izraste u najpopularniji računar uopšte, sa dva ili tri miliona

veliku prednost i nešto novaca. U svakom slučaju, firma bi trebalo da pokaže više poslovne sposobnosti a manje arogancije, koja ju je i dovela na ovako niske grane. Interesantno je šta važni londonski gradski finansijski krugovi misle o novim računarima. Oni su pokazali svoje objavljene oduševljenje time što su deonice Ejrkorna odmah pale za 10% kada su cene novih računara. A Ejrkornove deonice su prošle godine zabeležile najveći pad od svih deonica na londonskoj berzi. Neslavni rekord za jednu tako mladu industriju, koja je mnogo obećavala!

Novi „mek"

Za razliku od Ejrkornovih računara, najnoviji model „mekintoša" bio je jedna od najlošije čuvanih tajni. Njegove specifikacije su bile poznate znatno pre konferencije za štampu održane u drugoj polovini januara u Londonu.



Istinski bejzik

Bejzik je nekada bio mijenčue u obrazovnim institucijama i svaki početnik je počinjao upravo ovim jezikom za početnike, kako mu i sam naziv kaže (BASIC je skraćeno od Beginner's All-purpose Symbolic Instruction Code).

A onda je pre više od decenije došao paskal sa svojim famoznim strukturalnim programiranjem i svako dalje pominjanje bejzika je postalo, najblaže rečeno, neprikladno.

Tomas Kurc i Džon Kemenj, nekada profesori Dartmut koleđa, inače izumitelji bejzika, kreu u ponovno osvajanje škola sa ovim potisnutim jezikom. Verzija bejzika koju danas propagiraju ova dva profesora nazvana je Istinski bejzik. Istinski bejzik može da uradi sve što može paskal i više od toga", kažu zagovornici tog novog bejzika.

„U doba kada smo razvijali bejzik na Dartmut koleđu, postojale su komande koje su omogućavale strukturalno programiranje, ali današnje verzije bejzika su sve to izbacile ostavivši svuda prisutnu GO TO komandu, koju danas svi omalovažavaju. Smanjenje broja komandi je samo jedan od primera kako komercijalizacija bejzika na mikroracunarima krši osnovne principe bejzika i kako se od originalnog bejzika sa koleđa Dartmut stiglo do uličnog bejzika," kažu Kurc i Kemenj.

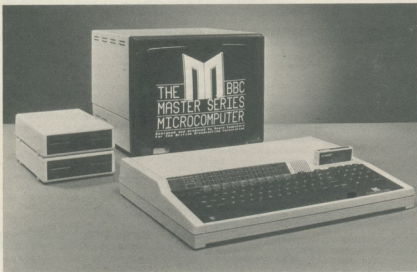
Prema ovoj dvojici uglednih profesora sve je počelo u doba razvoja mikroracunara kada su restrikcije, uslovljene nedovoljnom memorijom računara, izazvale neophodna kresanja. Pri pojavi moćnijih računara gde nije bilo ovih ograničenja, početne greške nisu ispravljene i to čine njih dvojica, danas, istinskim bejzikom.

Da li će Istinski bejzik potisnuti dobro ušankeni i danas najpopularniji Turbo Paskal videćemo u bliskoj budućnosti. Pored toga što je veoma rasprostranjen, Turbo Paskal se prodaje za 70 dolara, dok se za Istinski bejzik traži 150. Većina današnjih računara se isporučuje sa standardno ugrađenim „majkrosoftovim" bejzikom koji je možda slabiji od Istinskog bejzika, ali je, praktično, besplatan.

J. N.

Kolor LCD

Uskoro treba da se pojavi još jedna zanimljiva novost — kolor ekrani sa tečnim kristalima. To najavljuju i Epson i Tošiba, i kažu da bi cene trebale da budu duplo veće nego za običan LCD.

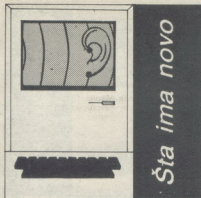


korisnika, a ima ih samo oko pola miliona. Tada bi Ejrkorn mogao da zaradi dosta novaca prodajom softvera i perifernih jedinica i sigurno ne bi bio u finansijskim poteškoćama iz kojih ga je u dva navrata morao da vadi italijanski "Olivetti".

Pitanje je da li će čak i škole kupovati ove nove bez sumnje dobre ali i preskupe računare, bez obzira što za njihove daleko jeftinije konkurente ima manje obrazovnog softvera. Dakle, budućnost za Ejrkorn je još uvek neizvesna. I ukoliko Ejrkorn ne uspe sa nekim drugim proizvodima, poput ROSC (reduced instruction set) čipova, koji su odlično ocenjeni i koji su znatno snažniji i brži od konvencionalnih procesora a mogu da ih u potpunosti oponašaju (dakle, šansa za potpuno novi računar potpuno kompatibilan sa starijim modelima), onda se Ejrkornu loše štiri! Uspех na ovom polju mogao bi, s druge strane, Ejrkornu ponovo da donese

Model „mekintoš plus" ima nekoliko noviteta koji ga čine veoma atraktivnim: specijalni tasteri za brojke (numeric keypad), dva puta veća brzina, udvostručena korisnička memorija na jedan megabajt, dupla (double sided) disk jedinica kapaciteta od 800K. Uz još neka druga poboljšanja, u „mek plus" je ugrađen i specijalni interfejs, nazvan Scusi (Small Computer System Interface), koji omogućuje da se na računar priključi čak sedam perifernih jedinica.

Računar je već pušten u prodaju po dosta visokoj ceni od 2295 funti. Originalni „mek" od 512K će se dalje i proizvoditi, dok se onaj od 128K ukida. Dosadašnji vlasnici modela 512K će moći da nadgrade svoju mašinu da radi poput „meka plus". U Applu smatraju da za ove nadogradnje vlada veliki interes, pa su zato obećali da će ovo proširenje pravadati po „povoljnoj ceni", koja će, kako mi saznavamo, iznositi 300 funti. Svi modeli „mekintoša" za Evropu, Aziju i Indiju proizvodiće se u novo modernizovanoj fabrici u gradiću Korku u Republici Irskoj.



Šta ima novo

Greška, grešnica

Misteriozni Komodorov PC je povučen tek što je predstavljao. Očekuje se novo predstavljanje sada u januaru. Izgleda da su postojale dve suštinske greške — nešto nije bilo u redu u dovodu energije i računar nije primao dodatne pločice koje su spas za svaki PC. Više sreće iduću put!

Koliko košta memorija

Veruje se da će uskoro cene memorijskih čipova porasti zato što se američki proizvođači potpuno povlače sa tog tržišta, prepustajući monopol Japancima. Gde je monopol, tu su i veće cene.

BBC je mrtav, živeo BBC

BBC B zaista ne postoji više, ali je zato tu BBC+. Sjajan računar. Jedino neki pakosnici i dalje tvrde da je cena prevelika.

Bitna Biblija

Zvuči neverovatno, ali pojavile su se prve verzije softverskih Biblija. Sad se zaista može reći da je nekim programerima računarstvo religija.

Uradi sam igru

Activision je proizveo programski paket Game Maker koji, navodno, omogućava „svakome“ da napiše super igru. U paketu su Sprite Maker, Scene Maker, Sound Maker, Music Maker i Editor. Postoje verzije za epf II i „komodor 64“. Ovaj paket će biti jako popularan kod nas dok se ne sazna da Englezi ipak ne kupuju takve igre.

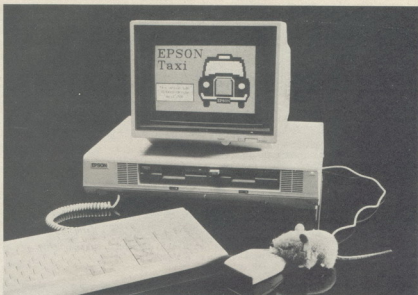
Prstom u računar

Epson je najavio LCD tablicu pomoću koje će se podaci unositi u računar direktno, prstom. Prava balkanska stvar!

Časkanje sa ROM-om

Ubrzo posle puštanja u prodaju Taxi sistema (pisali smo o njemu u prošlom broju) koji je činio čuda za IBMov PC, Epson je izbacio na tržište i svoj novi PC koji se zove, pogodite kako, Taxi PC. Pogodite koji sistem koristi. Lukavo.

6/šta ima novo



Taxi za Epson

Keytronicsova nova dodatna tastatura koja je namenjena običnom PCju i XTju ima nekoliko novina. Najznačajnija je ta što se uz Keytronic 512 V, kako se zove tastatura, dobija i mali mikrofoni pomoću koga se mogu direktno unositi naredbe. Zgodno, a?

Sve su nam bliže

Vuman Computers Systems LTD je proizveo, drži se čvrsto, tekst procesor za ruski jezik. Da li to znači da nam se polako primiču? Pitanje je kako bi to izgledao tekst procesor na „jugoslovenskom“ jeziku.

Međuklase računara

Proizvođači pokušavaju da obrate pažnju na računarska međutržišta i kao rezultati takvih htenja pojavljuju se mnoge nove mašine kao što su Eprikotov XEN i Ejkor-nov Communicator.

Živi dalje

Evo još jednog dokaza da za kreativne ljude nema zastoja. Anđelko Zgorelec, pokretač prvog i još uvek najboljeg evropskog časopisa za kućne računare „Practical Computer World“, i, naravno, naš specijalni dopisnik iz Velike Britanije, započeo je sa radom na novom časopisu koji se zove Practical Electronics. Zasad toliko, uskoro opširnije.

Deset malečkih

Časopis „Practical Computing“ je napravio svoj spisak deset najboljih portabil računara. To su Bondwell 2, Datavee 25, Data General One, Epson PX-8, Husky Hunter, Kaypro 2000, Olivetti M-10, Sharp PC 1500, Tandy 200, Toshiba T-1100. Cene vam nećemo pominjati, jer su one dosta veće od samih računarića.

Sa papira u memoriju

Da li ste nekada pozeleli da unesete tekst u memoriju računara a da se ne grnjavite lupajući u tastaturu? Uplatite 400 funti firmi Oberon International (Hall Road, Maryland Wood Estate, Hemel Hempstead, Hertfordshire HP2 7BH) i dobićete njihov novi Omni Reader. Uredaj pomeću na grafičku tablu u koju se umeće papir sa tekstim preko koga se zatim prelazi specijalnom optičkom olovkom. Uz malo vežbe unosite tekst brže od najbrže daktiografije i to, što je najlepše, bez grešaka! Omni Reader se povezuje sa bilo kojim računarom uz pomoć RS232 interfejsa. U cenu je uračunato i uputstvo za pisanje programa softvera, ali ne i taj softver! D. R.

Dvostruko brže

Udvostručite brzinu vašeg BBC-ja za samo 20 funti, poziva firma Solidisk. Za ovu se, naime, sumu prodaje „FourMeg“ tabla kojom se zamenjuje 6502 mikroprocesor osnovnog računara tako da kombinacija radi na cela 4 MHz. Kompatibilnost sa softverom je stoprocentna ali će vam možda neke igre izgledati prebrzo... U tabeli smo prikazali uporedne karakteristike raznih modela BBC-ja i drugih, uglavnom šesnaestobitnih, mašina pri izvršavanju standardnih „benchmark“ testova. BBC B, dopunjen „FourMeg“ tablom, očito nadmašuje mnoge daleko skuplje kompjutere. Telefon Solidiska je (0702) 354674.

PCW Benchmark testovi

1. BBC B sa 32016 Koprocesorom i BBC bejzikom	6.88
2. IBM PC AT	7.11
3. Olivetti M24	7.66
4. BBC B sa 'FourMeg' tablom	9.79
5. BBC B sa dodatnim 65C02	9.83
6. BBC B sa dodatnim 280	10.42
7. Apple Macintosh	12.15
8. BBC B ili BBC B+ sa bejzikom 2	14.55
9. Apricot PC	16.70
10. IBM PC (standardan)	17.60
11. BBC B+ sa bejzikom 128	17.90
12. Electron sa bejzikom 2	20.55

Valida su podmlađivači BBC-ja toliko dobro obavili posao da on sada ima i himen — znači to je poputno nova i nevina mašina. Hvala na ovom intimnom uvodu u detalje podmlađivanja BBC-ja.

Ljubomir Dragutinović, Zemun

Dole pirati

Da li ima ikakve šanse da se sve ono što je opisano u članku „Piratska je luga pregoljena“ dogodi i kod nas? Zaista mi je muka kad pogledam male oglase u bilo kom računarskom časopisu i vidim one silne oglašne pirate. Koliko znam, strani časopisi ne objavljuju više takve male oglase otkad je donesen zakon protiv piratstva. Ako se kod nas donese takav zakon, da li ćete vi prestaty da objavljujete takve male oglase? To biste mogli da uradite i pre donošenja zakona, kad svi dođrpnos uvođenju reda.

Sead Puškalic, Mostar

Noge 1

Mora se priznati da vaša naslovnica strana u „Računarima 11“ zaista ne liči na stroge i tehnicizirane naslovnice od koje čovek može očekivati da je jedno stručno časopis, ali zar vam se ne čini da je malo preterali. Uostroto će neki ljudi početi da vam kupuju časopis iako ne znaju ni što su to računari, samo zbog naslovnih strana. Meni se bode na naslovnj strani ne svidjaju, ali moja žena ne deli to mišljenje. Imao sam problema dok sam je uverio da vašu naslovnju stranu podržavam samp teorijski, iz profesionalnog razumevanja.

D.S. Beograd

Noge 2

Umalo da ostanem bez „Računara 11“ i to samo zbog onih nogu na naslovnj stranici. Znate li da sam jedva uspeo da ubedim moje roditelje da vaši „Računari 11“ nisu još samo neka vrsta Starta? Tek kada sam im pokazao listinge u sredini i tekst Dušana Slavića, poverovali su mi da ste vi ozbiljan računarski list. Moji tata je, čak, zatražio da mu pozajmim taj broj da bi detaljnije pogledao članak Dušana Slavića. Sada vidite kakve probleme vernim čitaocima prave vaše super naslovnj strane. Gopi, moji drug iz razreda, tvrdi da ti i nisu neke narocite noge, ali ja sam ga poklopilo argumentom da tu do sada najbolje noge u našem računarskom časopisu.

Miško iz 45. bloka

Noge 3

Najzad ste uspeali da napravite najružniju naslovnju stranu.

Zna se čak i vinovnik tog „dela“. To je izvesni Vladimir Simović, koji potpisuje i ostale fotose objavljene u vašoj reviji (srećom, ne baš sve; samo one loše).

Posmatrajući tehničku stranu fotografije za naslovnju stranu, doista sam do zaključio da ona izgleda „Smenom 8“, i to vrlo loše podelenom. Zašto?

Cela slika, osim monitora, je nejasna i mutna, a boje su loše razgraničene. Jedino hajinja opravdava svoju nejasnost. Najvažniji deo slike, natpis „RAČUNARI 11“ je O.K., ostatak je očajan: (pokušajte da pročitate natpise na tasternal!) Inače, i te računaru pridodate ženske noge, su vrlo problematične. Pomalo su krvrgave i krive, a tome je još samo doprinelo loše i nepravilno osvetljenje, koje je nanelo odlučujući udarac celoj slici. Svetlo pada sa desne strane i odzoga, što stvara čudne senke i pokazuje ono što bi baš trebalo da prikrije.

U ekranu monitora se jasno vidi dokle seže bela tkanina ili papir na kojem stoji „Schneider“ i ženska. Tu je i autor, kao i njegova oprema. Što nam jasno govori da je ceo hepening održan u nekom „studiju“.

Devojka je mogla da obuje sandale i neke druge boje, tek da se razdvoji ko je ko. I zašto je „odsečen“ levi kraj tastature?! Zar je „Schneider“ toliko glomazan da ne može da stane u jednu foto-kompoziciju?

Dalje, koncepcija i raspored objekata su jednaki kao na naslovnjoj stranici „RAČUNARA“ br. 8. Jedina razlika je u tom što je ova starija, nastala tehnikom „seckanja“ iz stranih časopisa, a mlađa je „autorski rad“. Sve me to podseća na onaj ručni termin, „šrafciğer — industrija“.

Zar ne umete da napravite dobru naslovnju stranu? Stariji brojevi su bili „pioniri“ u tom poslu, a i niste bili baš načisto sa tim šta staviti na naslovnju stranu, ali sada imate već poveliko iskustvo, pa nam i povećajte estetski nivo i naslovnje stranice, i sadržine.

U vezi s tim, imate još jednog „rutinera“ u vašim redovima koji nam svaki put servira istu „priču“: To je vaš crtač Miša Marković, koji se ubi strajujućeg ženske i tokoznirove tastature... a la ZX 81“; slova naopako i pravci non-stop portrete Andrije Jurkovića. Ni ja ne volim Andriju, jer mu je samo cilj da uzme lovu, ali ni grešni Miša ne sanja da Andriji sam pravi reklamu.

Zato bih vam predložio sledeće: ostavite mogućnost da i čitaoci naprave neku ilustraciju, i da to bude stalna praksa. Naravno, vaša je stvar da li ćete te radove honorisati ili ne, li, recimo, da otvorite konkurs za najbolji crtež ili animaciju napravljenu kompjuterom. Sigurno sam da bi bilo dosta mesta oko ocenjivanja radova.

Pošto sam ja ilustrator, voleo bih da vam uredim poneki crtež ili slično, naravno, bez nadoknade. Ne volim da uzimam lovu od onih koje cenim i poštuju!

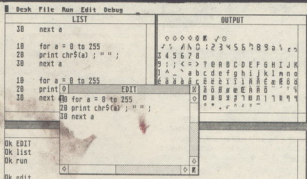
Na ostali, tekstualni sadržaj revije, nemam nikakvih primedbi, već vam savetujem da tako i nastavite. P.S. Nadam se da gorenepomute krive i krvave noge ne pripadaju Jelenu Rупnik, jer nju volim.

P.P.S. Izvinite zbog grešaka i stila, zasad još uvek nemam tekst-prosorak i printer, što ne znači da ih neće izeti!

STUJANOVIĆ DARKO
 Josipa Debeljaka 14a
 11132 Beograd

Priprema: Branko Daković

Pogled izbliza licem u lice sa „atarijem“



Džek Tremiel, zaista, nije preterivao kada je tvrdio da sa super modelom njegove firme može da radi čak i čovek bez bilo kakvog kompjuterskog i matematičkog obrazovanja. S druge strane, čovek sa minimumom kompjuterskog iskustva će se dobro oznojiti dok ne dovede računar u radno stanje!

Mišem po bejziku...

Najpre, kao što i pristoji, treba uključiti računar, disk i monitor (ako ste nekada radili sa nekom Packardovom mašinom, znate da bi se ta operacija mogla izvesti i pritiskom na samo jedan taster), pre čega se u dravj stavljaju sistemska diske. Računar će se tada ozbiljno zamisliti dok će se disk besomučno vrteti skoro čitav minut i GEM će biti učitana. Sada računar odlazi u hibernaciju i čeka da se vi iz njega probudite — treba zameniti sistemske diske drugom na koju je snimljen čitav bejzik interpreter. Novo će učitavanje potrajati sledećih pedesetak sekundi, a onda će se ekran promeniti — u prvom redu videćemo meni — sa prijatnim izborom: 'Desk', 'File', 'Run', 'Edit' i 'Debug'. Za početak treba koristiti miša da se kurzor dovede do reči EDIT, a onda pritisnuti taster: ispod zaglavlja će se pojaviti novi meni i koga treba izabrati opciju 'Start Edit'. Tada ste došli do tačke na kojoj se sve vlasnici „spektruma“ i drugih osmo-bitnih mašina malazili u momentu kada povežu kompjuter sa napajanjem.

Možemo da budemo vrlo zahvalni programerima firme Digital Research što su dopustili da se u EDIT modu nared-

be kucaju sa tastature — moglo se desiti i da se svako PRINT bira iz menija. Pa ipak, miša nećemo smesti sasvim da zabavoravimo — čim otucamo program ili izvršimo neku izmenu u njemu, moraćemo da putujemo do napisa 'Exit Edit', a onda i do glavnog menija sa komandom 'Run'.

Dobro, reći ćete, miš ponekad može da bude koristan: koliko ste samo puta kucali LIST da bi vam deo programa koji ste želeli da ispravljate „pobegao“ sa ekrana? Miš će vam omogućiti da skrolujete program kroz Edit prozor na gore i na dole i da pročitate sve naredbe koje su vam potrebne. Pogrešno! U edit prozoru se ne može skrolovati tekst (ili bar, u su divno napisano uputstvo koje se kupcima isporučuje u fotokopiranom obliku, nismo pogodili kako se to radi); programere ćete tako morati da editujete primenom ekranskog editora koji je vrlo sličan „galskijinoj“ (bolji je utoliko što se po želji nalazi u „Insert“ ili „Overtime“ modu), a zatim ćete svaku ispravku morati da potvrdite pritiskom na RETURN. Ispravljena linija automatski putuje ka u programsku memoriju tako i na programski prozor.

Pošto ste nekako otukali program, sa 'File' ga smeštate na disk a zatim sa 'Debug' tražite greške u njemu (dibeger je stvarno izvanredan i vredno bi mu posvetiti poseban tekst). Kada mislite da ste na tragu neke od njih, ponovo aktivirate opciju 'Edit' i tako u beskrajanom nizu — miš će sve vreme biti prazan, a takođe i poveliki prazan prostor pored kompjutera za njegovo kretanje. Ako

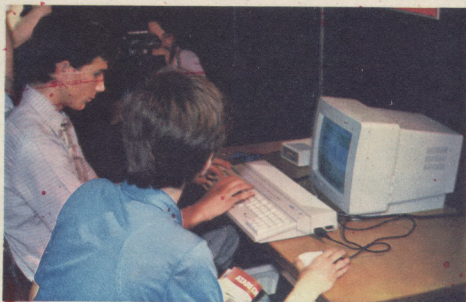
Atari 520 ST nije više samo pusti san domaćih hakera: već i po pionirskim malim oglasima na stranicama kompjuterskih revija možemo zaključiti da su prvi primerci ovog kompjutera na razne načine srećno prešli jugoslovensku granicu. Nedavno je i autor ovoga teksta bio u prilici da se nekoliko dana zabavlja novom Atarijevom mašinom i napiše nekoliko programa za nju. Krivica za činjenicu da su ti programi bili kraći od pedesetak linija je isključivo Atarijeva — kada u računar učitate sve programe koje je potrebno ostave vam slobodno samo pet kilobajta RAM-a, manje nego što imaju vlasnici neproširene „galaksije“.

vam je, kao što programeru i priliči, što prekriven hrpama papira, moraćete da menjate navike ili, ovaj, računar.

... a bejzik iz neolita

Ako ste nislili da mašina sa moćnim procesorom i ogromnim operativnim sistemom mora da bude opremljena moćnim i brzim bejzikom, pogrešili ste: Atarijev bejzik je zapravo prilično stara Microsoft-ova tvorevina dopunjena sa par interesantnih naredbi za rad sa grafikom i tonom (o njima nekom drugom prilikom) i dve nove kontrolne strukture — DO WHILE (petlja sa izlazom na početku) i označeni GOTO (umesto GOTO 1023 možete, na primer, da napišete GOTO ucitaj što, složićete se, čini program mnogo čitljivijim). Sa druge strane, Atarijevom bejziku je oduzeta Microsoftova numerika — radi se samo 5—6 tačnih cifara, poput „galaksije“. U uputstvu je, doduše, pomenuta mogućnost rada sa promenjivima u dvostrukoj tačnosti koje se deklariraju sa DEFDBL ili završavaju tarabom (*), ali se ta mogućnost verovatno izgubila u transportu iz Nemačke. Za uzvrat, računar ima naredbu VAL koja u uputstvu nije spomenuta (drugo je pitanje što ta naredba ne radi dobro) i funkciju VALPRP koje se sećamo još sa starog TRS 80.

Atarijev bejzik, poput svih „neolitskih“ varijanti ovoga jezika, omogućava jedino pozivanje potprograma naredbom GOSUB i definisanje procedure u jednom redu (DEFN). Promenljive kojima nije dodeljena vrednost će, kada ih štampamo ili računamo sa njima, imati vrednost nula, što je veliki potencijal za greške koje nije ni malo lako otkriti. Osim toga, naredba RENUMBER se veoma čudno ponaša ako se koriste reference na nepostojeće linije, a ni komanda AUTO nije lišena „dečjih bolesti“ što se javljaju ako se zahteva ivo veća razlika između linijskih brojeva susjednih naredbi.



Rad sa periferijom je preglodno ali u mnogim stvarima traljavo rešen: štampač, ekran i disk imaju različite i međusobno nezavisne komandne strukture, što znači da će se sa budućim proširenjima komunicirati isključivo naredbama POKE, koje se i na postojećem Atariju 520 ST koriste za mnoge stvari, posebno za pozivanje potprograma iz GEM-a i prenošenje PARAMETARA koje će koristiti funkcije operativnog sistema. Mašinski potprogrami se pozivaju naredbom CALL.

Prijavlivanje grešaka je tekstualno i prilično zgodno, iako se svodi na mali broj poruka. Većina grešaka rezultuje porukom „Something is wrong“ i ponavljanjem linije praćene strelicom koja ukazuje na grešku — treća sličnost između atarija i „galaksije“.

Nikad ne kučaj FRE()

Za kraj ovoga prikaza smao ostavili funkciju FRE() koja, kao i na svakom Microsoftovom bejziku, daje programeru obaveštenje o raspoloživoj memo-

riji. Pošto učitate operativni sistem, GEM i bejzik, pripremite čašu vode i otukajte PRINT FRE(), Atari 520 ST je, kao što svi dobro znamo, opremljen sa pola megabajta RAM-a, pa bi se dalo očekivati da za bejzik bude slobodno najmanje njih stotinaak PRINT FRE() će vas uveriti u gorku činjenicu da Tremielova mašina može da primi najviše bejzik program dug 5 kilobajta, manji od programa koji bi mogao da se smesti u memoriju neproširene „galaksije“! Ako se odrednete grafike i usluga GEM-a, program ćete moći da produžite za 32 K ali — zašto ste onda kupili atari?

Mala memorija za bejzik se, osim toga, prilično neracionalno koristi — „režijski prostor“ je petnaest bajta po liniji, što znači da će PRINT „Računari“ pojesti četrdesetak bajta. Na atariju 520 ST se, dakle, mogu pisati samo najkraći bejzik programi koji će se, uz sve to, izvršavati mnogo sporije nego na nekom amstradu i jedva nešto brže nego na komodoru ili spektrumu. Moguće je da će se

u budućnosti pojaviti nove verzije bejzika i drugih programskih jezika, ali bi one morale da budu komplikovane i samim tim obimnije od postojećeg bejzika; atari 520 ST će, dakle, morati da se oprema memorijom proširenjem da bi se na njemu moglo iole ozbiljnije raditi.

Sve u svemu, nova otkrića o Atarijevom „super kompjuteru“ predstavljaju pravo začoćanje: iako su uslovi nabavke sve povoljniji (cena u Nemačkoj je sa 3000 DM pala na 2200, dok se na božićnim rasprodajama kompjuter mogao naći i za 1900 DM), bejzik i slobodan RAM predstavljaju ogroman korak nazad u odnosu na standarde koje su postavile osmo-bitne mašine. Čak i prikazi u najumerenijim stranim kompjuterskim časopisima iznose slične utiske — Tremiel će morati mnogo toga da promeni ako želi da njegov računar postane interesantan za nezavisne softverske firme i uspe na tržištu.

Dejan Ristanović

Softverska
trpeza

novinar u svojoj kući

Ako već duže vremena sanjate o svom privatnom računarskom listu, pojavio se pravi program za vas. Programski paket Njuzrom (Newsroom – na engleskom „redakcija“) vam omogućava da obradujete novinarske članke, pravite ilustracije, „prelamate“ strane i, naravno, štampate sve to. Treba samo naći čitaoce.

Celim programom se upravlja pomoću džojstika, što rad čini prilično jednostavnim. Rad se obavlja preko opcija koje biramo iz menija, a radni prostor su prozori. Već to govori da je uložen veliki napor da se imitira rad „meka“. Zato je potrebno veoma malo uputstava da bi se program koristio. To ne znači da je on zaista nalik „mekovom“ sistemu.

Prizor posle učitavanja špice je zaista prigodan, pred vama je slika redakcije nekog lista (vašeg lista) i sve moguće opcije rada.

Prva opcija se zove „Banner“. On omogućuje izradu zaglavlja vašeg časopisa. Na ekranu dobijate belu prozor na crnoj pozdizi u kome možete da pravite zaglavlje.

Sledeća opcija je na ekranu označena kao marka (slika) i služi za unošenje ilustracija u novine. Ilustracije u obliku sličica se nalaze na dve diskete koje dobijate uz program. Diskete sadrže nekoliko stotina sličica sasvim različitog karaktera (od ljudi do raznih čudovišta). Kada kursor (koji je u obliku ruke) dovedete do znaka sličice (marke) i pritisnete dugme na džojstiku, program će zatražiti da disketu Clip Art ubacite u dralj i da pritisnete dugme dok je kursor na opciji OK. Tada se na ekranu pojavljuje spisak različitih imena pod kojima su sličice grupisane. Ime najčešće i doslovno govori o sadržaju slike. Pod jednim imenom je grupisano od tri do desetak sličica. Kada izaberete šta vas interesuje, dovedete kursor do imena i pritisnete dugme. Disk počinje da se vrti i na ekranu se pojavljuje veliki prozor sa sličicama. Kada pronađete sličicu koja vam odgovara, treba dovesti kursor do nje i pritisnuti dugme. Program vas vraća u radni prozor gde sličicu možete premasiti po volji. Ako ne uspete da nađete sličicu koja vam odgovara, na raspolaganju vam je opcija „Photo“ koja je na ekranu predstavljena kao mala foto-laboratorija. Pomoću te opcije možete da slobodno crtate, prepravljate gotove sličice i još dosta toga. Kada završimo rad na sličici potrebno je samo izabrati opciju koja je označena malim foto-aparatom. Blis sevine i vi imate vašu sličicu koju možete snimiti na disk i koristiti kao i one koje ste dobili uz program. Ako ste talentovani za crtanje džojstikom, ovo je prava stvar za vas.

Ipak, vi ne želite samo da crtate — u novinama se i piše. Pisanje možete započeti pisanjem naslova i potpisa uz slike, što se čini pomoću opcije koja je nacrtana kao mala olovka. Na raspolaganju su vam tri vrste velikih i dve vrste malih slova. Pisanje članaka vam omogućuje opcija „Panel“.

To je mali specijalizovani tekst procesor

10/novinar u svojoj kući

Newsroom Newsroom

Newsroom

1

news  novosti :

Program za novinare

prizor: Ured Krstonošić

Nedavno se u prodaju pojavio program koji će odvesti sve one koji se ne bilo kog načina bave pisanim. Program se zove NEWSROOM, napravio je za Commodore 64 i odnosi se na sve koji se bave pisanim i pravišne novinskih članaka, njihova ilustracija i naslovo štampaće. Ceo program ima ukupno šest opcija, a one i prinašati tekstove za modera, što je posebno interesantno. Program se rukovodi uz pomoć palica za igru i sa pomoć menija veoma je jednostavan za upotrebu.

Če program se dobijaju i dve diskete na kojima se nalaze vrlo lepo urađene slike za ilustraciju tekstova. Sličica ima nekoliko stotina i saista svih vrsta, pa nije problem izabrati.

koji će vam pomoći da svoj tekst oblikujete u stupce i blokove. Dok ste u „Panelu“ ne možete crtati ali možete učitati neku od sličica sa diskete. Pošto program ne dozvoljava snimanje na „njegovom“ disku, morate uz njega formirati Data disk na kome ćete držati sve naslovne tekstove u slično.

Kada završite sa pisanjem teksta, crtanjem i pravljenjem zaglavlja, stranicu definitivno formiramo pomoću opcije „Layout“. Po velikom prozoru koji predstavlja stranicu možete slobodno da razmeštate „Paneele“ teksta, slike i naslove. Kada stranicu formirate, snimate je na disk. Posao u redakciji je gotov.

Ostaje samo štampaće. Ono se radi opcijom „Press“. Ako ste pomislili da će možda biti problema zbog neuskadnosti sa printerom, prijatno ćete se iznenaditi — na ekranu ćete dobiti izbor od preko trideset raznih printera. Samo pritisnete dugme nad opcijom OK i iz printera će izći prava strana vaših (i zaista vaših) novina. U tom trenutku ćete saznati kako je lep osećaj biti i novinar i urednik i tehnički urednik i slugać i štampar.

Ako ste posle ovakvog opisa rada pomislili da program prosto „guta“ memoriju,

neku ilustraciju za određeni članak. Dan program izgleda izvanredno lepo i elegantno i veoma je jednostavan za korišćenje, ali ipak ostavila neke novosti onako kao je ova radi.

NEWSROOM radi
sa nekoliko različitih tipova slova
koja možete birati po izboru.

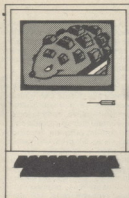
Fotografije:
Možete koristiti nekoliko
stotine fotografija kao
ilustracije za tekstove.

Špiči od koji su zainteresovani za ovaj izvanredno koristan i zanimljiv program mogu pročitati detaljan opis programa u časopisu za vizeznu programera — RACUNARI u svojoj kući. Ako posle toga oduševite da vam je NEWSROOM potpisan, javite se nekome ko program ima i spremni da vam ga „uštiju“.

niste u pravu. Program na zaista impresivan način smešta podatke u memoriju, tako da cela jedna strana teksta sa nekoliko fotografija i zaglavljem zauzima na disku, verovali ili ne, nešto manje od 4K. Impresivno, zar ne? Opcija koju nismo pomenuli je namenjena razmeni podataka. Predviđena je za rad sa modedom, odnosno povezivanje više ovakvih programa sa raznih strana.

Moglo bi se reći da je ovaj program uspešna minijatura sličnih programa namenjenih „meku“. Ono što ga čini uspešnim je jednostavnost korišćenja i relativno niska cena. On ne predstavlja neku ozbiljnu mogućnost za privatno izdavaštvo, ali je izuzetno zanimljiv za eksperimentisanje i „interno“ izdavaštvo. Izvanredan je za male lične formulare i fanzine. Nekom hakerskom klubu bi pametna upotreba ovog programa mogla da potpuno izmeni status. Prava mera ovog programa je izdavaštvo za ukučane i drage prijatelje, što precizno definiše njegovo mesto: na pola puta između igračaka i „ozbiljnih“ uređivačkih programa. Obavezno ga isprobajte, možda ćete otkriti u sebi uredničku crtu. Vredni pokušaji.

Vladimir Krstonošić



Računari
iz mog ugla

Mi sve možemo

Vi i vaš računar ste slični: ne koristite svoje skrivene mogućnosti. Živite u harmoniji sa svojim računarom, i nimalo vas ne zabrinjavaju filmovi na TV u kojima kompjuter-monstrum jede žive ljude. Za sada vaš računar ima više razloga da se plaši vas nego vi njega. Nesпоkој se javlja kada zavirite u računar i ugledate mnoštvo čipova i kada se suočite sa masom nerazumljivih programa.

Kada bi vam neko na vreme pokazao svu gomilu knjiga koju morate da pročitate u toku svog školovanja, ne biste pristali ni da živite, a kamoli da idete u školu. Tako je i sa kompjuterima. Zgranuti ste kvalitetom i zatrpani kvantitetom. Obeshabreni pojedinač pristaje na ono što mu se nudi. Polako se svodi na običan kondenzator ili otpornik - mizeran deo sa ograničenom funkcijom. Ideal svakog društva je da svako zna svoje mesto.

Ovakav defetistički stav pojedinca je u potpunom kontrastu sa zvaničnom politikom: MI SVE ZNAMO I MI SVE MOŽEMO. To je najbolji mogući primer dijalektičkog zakona prelaska kvantiteta u kvalitet.

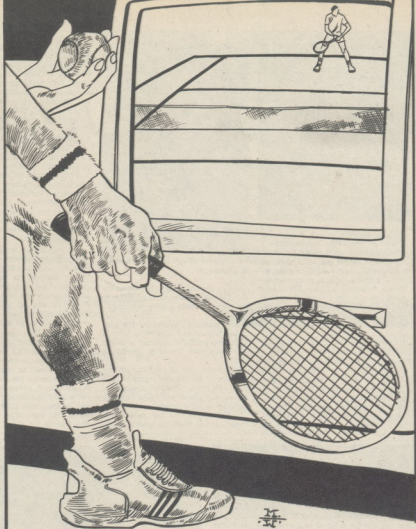
Ne sme se sumnjati u to da mi sve znamo i da mi sve možemo. U vicevima smo uvek bili bolji od Amerikanaca i Rusa. Ove dve tvrdnje (MSZ-MSM) međusobno se prožimaju; istinitost jedne dokazuje se istinitošću druge, i obrnuto.

Situacija u kojoj ste vi zaslepljeni time što MI sve znamo i sve možemo je vrlo zgodna za pojavu raznovrsnih oblika mimikrije računarskih autoriteta. (Obratite pažnju: „MI“ — to smo svi mi, a naročito neki. Piše se velikim slovom iz poštovanja prema nama).

Učili ste iz biologije šta je to mimikrija. To je kada, na primer, bezopasan gušter izigrava zmiju otrovnicu; kao kada biste spektroma opremili tastaturu IBM PC-a.

Na primer, uobraženi naučnik-profesor koji odbija uopšte da komunicira sa svojom visokom postolom, jer je sve to sa malim računarima ispod njegovog nivoa. Vi nikada nećete imati priliku da se timo popnete, kucnete po njemu i ustanovite da je šupalj i'o sponenik.

Jedna druga, vrlo česta varijanta su lažni autoriteti koji nešto znaju polovično. Polutani se služe svim mogućim pomoćnim



средствима ne bi li ostvarili što ubedljiviji utisak. Oni ponavljaju stvari koje je neko već rekao, i, neznačajno ih menjajući, šalju dalje u optičaj. Kruženje priče obavlja se po zakonima kruženja materije u prirodi (učili ste o tome iz Poznavanja prirode i društva). Povremeno polutan ubacuje u optičaj nove sadržaje u koje će biti spreman da poveruje kada ih ponovo čuje od nekog drugog. Snaga tvrdnje ne zavisi od objektivne istinitosti, nego od toga koliko je puta ponovljena.

Ovo su ipak defanzivne taktike u kojima se može naslutiti slabost. Nešto kao pomoćni izlaz u slučaju opasnosti. Pravi, rasni kompjuterski kontrarevolucionar će izabrati potpuno drugu taktiku; kreneće pomezno i agresivno kroz glavni ulaz. Njegova je namera da dobro unovči ono čime raspolaže u ogromnim količinama — svoje neznanje. Šta više, on mora biti apolutno siguran da pojma nema. Uz to mora posedovati čelične živce i odlučan karakter. To ne može običan čovek nego samo Supermen.

Dok je polutan nesпоkојan, pošto je nesiguran u svoje znanje, dotle je Supermen savršeno miran pošto je siguran u svoje neznanje. To je neravnopravni susret pričuđen o blefera i blefera velikog stila. Gleдаće ga pravo u oči i na prvi nagoveštaj nesigurnosti zasuče ga bujicom stručnih

izraza i velikih brojeva, što će potpuno blokirati mozak sirotog polutana, naviknut na preračunavanja na pijaci. Razni debeli direktori mogli bi biti masan zalogaj ako bi Supermen poprimio obličje prodavca računarske robe. Isto tako, on voli da konzumira i lakšu hranu, naročito kada vrši misiju kompjuterskog prosvetitelja.

Sa polutanima je, dakle, lako. Izgleda da ozbiljne teškoće naiđe kada se nameri na nekog ko je takođe siguran u svoje (ne)znanje. Ni tu Supermen ne može da izgubi, jer se verovatno radi o drugom Supermenu. Supermen Supermenu nije vuk; on će mirno razmenjati najnovija tehnološka dostignuća u bleferskoj struci, i raziči će svako na svoju teritoriju.

Supermena ima malo, ali pravih mudriša ima još manje. Verovatnoća da će se sresti i da će mudriša uspeti da ga razbue je dosta mala. Supermen zna da je za nedovoljno upućenog posmatrača istinito ono što se na ubedljiv način tvrdi. Prema tome, on će izgledati sigurno i onda kada bude govorio najveće gluposti.

P.S. Drugovi zgranuti pojedinci, nemojte misliti da je za prepoznavanje kompjuterskih blefera potrebna neka velika škola. Šaljite svoje predloge za Supermena godi-ne na adresu Peek & Poke Show.

Jelena Rupnik

Računari i obrazovanje

Vaše dete govori bejzik, a vi?



Na sasvim drugi način: Mihajlo Marasnov

način, učenici mogu bar donekle da upoznaju način rada sa računom, prve korake u programiranju i sl.

Po mišljenju profesora Marasnova, likovni rad izveden u tehnici računara može se svrstati u svako od klasičnih likovnih područja rada.

„Likovni rad može biti i animiran, zvuk je također sastavni dio likovnog rada. Vizualna mogućnost su neograničene kada računar povežemo sa videom.“

Mogućnosti su za sada skromne; u školi trenutno postoji samo

jedan računar, i to spektrum 48 K sa INES tastaturom. Najbolji je prošle godine u maju, i od tada se nastava likovne kulture izvodi isključivo pomoću komputera. Uskoro će se naći sredstva za nabavku još nekoliko računara, kao i jedne grafičke table. Sledeći korak bio bi povezivanje videa sa računom, čime bi se u potpunosti ova računarsko-likovna radionica.

„Sredstva za opremanje računarnima ne bi trebalo da predstavljaju neki veći problem“, kaže profesor Marasnov. „Problem je u tome što samo u SR Hrvatskoj treba računarski opismeniti preko 25 hiljada nastavnika i oko 220 hiljada učenika, što se po važnosti može uporediti sa posleratnim godinama, kada je trebalo opismeniti preko 70% stanovništva.“

U vezi sa tim, profesor Marasnov i njegovi učenici su već nekoliko puta pozivali nastavnike iz škole „Mate Balota“ da nauče nešto o računarnima i programiranju. Stampali su pozivnice na kojima je pisalo: „Vaši učenici govore bejzik, a vi?“ Mislim da sami možete predpostaviti kakav je bio odziv.

„Najgore što nam se može desiti je da ne živimo u svom vremenu. Položaj nastavnika uključuje i više od toga, jer on mora da predviđi kako će živeti ta deca na koju on prenosi znanja. Ne može se dozvoliti da učenik prvog razreda osnovne škole zna više od svog nastavnika. To postavlja pitanje postojanja škole kao institucije, što je sistemsko pitanje.“

Na osnovu svojih istraživanja, profesor Marasnov je ustanovio da je dete u uzrastu od pet godina već sposobno da samostalno komponuje bojama i grafičkim setom karaktera.

Ako znamo da je učenik na nivou prvog razreda osnovne škole već rešio sve dileme oko oblika ili skupova, da je u drugom usvoju pojam promenljive, onda je logično da u uzrastu trećeg razreda samostalno formira program. Smešno je, stoga, da se tek pred kraj osnovne škole uvodi informatika kao neki poseban predmet, kad se učenik već samostalno služi računom. Neoprostivo je koliko smo potensili i omalovažili dete, pripremišu mu ovakav program rada, koji mu je bio primeren uzrastu u znatno ranijem dobu. Ali, ako ne budemo radili po tom programu onako kako je to zamišljeno, dobijamo u budućnosti ne sistemom školovane generacije, već masu mediodkriteta za koje ćemo mi verovati da su Pismeni, s jedne strane, i sa druge strane elitnu grupu opismenjenu mimo sistema škole, koja će biti u stanju da drži generacijski korak sa svetom.“

Većina nastavnika s podrozrenjem gleda na uvođenje informatike i računarstva u obrazovanje, ipak, bilo bi lepo kada bismo u svakom broju „Računara“ mogli da vam predstavimo po nekog profesora koji ne čeka novu reformu obrazovanja da bi u svojoj nastavi primenio kompjuter.

J. Rupnik

Domaći računari PC iz Podravine

Domaće tržište postalo je bogatije za još jedno osobno računalo, za „Sperry PC 500“, sklopljeno u radnoj jedinici „Elektronika“ Drvne industrije „Gaj“ iz Podravske Slatine. Posljednjeg četvrtka prošle godine kamion za prijevoz najstarija umjesto uobičajene pošiljke prvi je puta iz tvorničkog kruga „Gaja“ odveo za 64 PC-a. Cijeli posao organiziran je u suradnji poznate tvrtke „Sperry Univac“, zagrebačkog „infosistema“ i „Gaja“. Strani partner dao je znanje u hardveru i sklopove. „Infosistem“ je serviser i kreator programske podrške za potrebe kupca, dok „Elektronika“ zasada iz dobivenih dijelova sastavlja računala i opremu.

Svako računalo marke „Sperry PC 500“ sastoji se od sistemske jedinice s disk-jedinicom, profesionalnom tastaturom, kolor-monitorom, a po želji kupaca isporučuje se sa štampačem ili ploterom.

Pedesetak radnika Radne jedinice „Elektronika“, koja je s radom počela prije godinu i pol, za samo četiri je dana sklopila 64 računala iz prve serije PC-a, 25 štampača i pet plotera.

„Sperry PC 500“, koji će kasnije dobiti naljepnicu „Infosistema“ i vjerojatno „gaj“ model A, ima ROM od 128 K i RAM od 256 K s mogućnošću proširenja na 512 K. Računalo je izvedeno profesionalno — tastatura je odvojena od centralne jedinice u cijelom je sklopu i diskjedinica. Monitor u boji upotpunjava taj kompjutorski set u mali poslovni sistem, koji će na domaćem tržištu stajati približno 4,7 milijuna dinara. Vani se prodaje za nekih 3.000 dolara.



Ove godine planira se proširiti krug organizacija koja će raditi na tom programu. Od domaćih proizvođača očekuje se osvajanje nekih pozicija hardvera — u početku mehanički dijelovi, kasnije i elektronički. Tako postoje izgledi da se ova računala opreme domaćim međuprogramima, modemima, štampačima. Na području programa u posao će se uključiti i „Duro Đaković“ i neke druge organizacije koje već imaju formirane softverske grupe.

Početna proizvodnja, prva serija, unaprijed je prodana, što daje realne izgleda za nastavak. Uskoro bi trebala biti sastavljena druga serija do stotinjak računala. Kad se usvoji dio domaće proizvodnje pa tako smanje davanja za uvozne komponente, planira se godišnja proizvodnja od 5.000 komada PC-a iz Podravske Slatine. Da li je to odviše smjelo pokazati će vrijeme!

Mnogi će se zapitati otkud elektronika u žrnju industriji? Zašto „šrafčiji industrija“? Prije dvije godine dva mlada inženjera iz razvoja službe „Gaja“, Danko Potkonjak i Ilija Davidović, predložili su da se ispita mogućnost razvoja na području elektronike. Budući da je sirovinska osnova za drvenu industriju ograničena

Mali poslovni sistem: „Sperry PC 500“ iz Podravske Slatine

— „Gaj“ je jedna od vodećih domaćih industrija namještaja, građevničkih konstrukcija — odučeno je da se podrži takva orijentacija. Ubrzo je sagrađen podzemni površine 140 četvornih metara. Danas tu radi 50 radnika, među kojima je osam inženjera i 15 kvalificiranih i visokokvalificiranih radnika. Oni koji prijavuju još jednoj „šrafčiji“ industriji nisu upoznati da se u većem dijelu svijeta računala i druge elektroničke naprave proizvode na bazi OEM tehnologije, a da tek samo velesila u elektronički mogu proizvoditi čipove i druge suptilne komponente računala i opreme. I dio komponenti što se sklapaju u Podravske Slatine nisu isključivo proizvod „Univaca“, već njihovih kooperanata.

Naše šanse su zasad u proizvodnji mehaničkih dijelova, potom jednostavnijih elektroničkih sklopova i, svakako, programske podrške. Tu bi trebali biti i najjači. Primjera radi, susjedi Madari izvozeći softver i neke druge proizvode lani su uvezli automobile marke BMW.

B. Hebrang

Neravnopravno takmičenje

Sećate li se prošlog takmičenja malih (i maloletnih) šahista protiv kompjutera koje je ne tako davno organizovao Doka Vještica, to jest Studio B? Sećate li se onog klinca koji je matirao kompjuter u sedam poteza? E, pa takvo takmičenje se ponovo organizuje. Predlažemo da klijncima budu zavezane oči, da igraju simultano protiv pet kompjutera i da im se za svu vreme borbi pušta preglasna novokomponovana narodna muzika. Zbog ravnopravnosti.

Čip pobodi agency

„Galaksija — Amerika“

Agencija PPS izveštava da je firma Atari ponudila odgovarajućim institucijama u Jugoslaviji da im isporuči 50 000 komada računara „atari 520 ST+“ sa specijalnim popustom od 80%. Kako se saznaje, jugoslovenska strana se zahvalila na ponudi i odbila je iz principijelnih razloga. Još uvek se ne zna da li je to učinjeno što bi postojale nejasnoće oko podela računara po republikama ili se, možda, radi o želji da se podrže izvanredni domaći tačunari „galaksija“, „lola“, „orao“, MMS i omogući im se prodor na svetsko tržište.

PEEK & POKE SHOW

Postavlja malo nagradno pitanje

Čije noge se nalaze na naslovnoj strani „Računara 11“?

Odgovore slobodno šaljite bez objašnjenja

Nagrade

Kao i uvek, „Peek & poke show“ je predvideo bogate nagrade za one srećnike koji pogode tačan odgovor.

1. nagrada — Fotografija kompletne redakcije „Računara“, snimljena pola časa pre nego što treba predati sve rukopise u štampu, a pola ih fali.

2. nagrada — Fotografija priredivača „Peek & poke show“ u pozi sa naslovne strane „računara 11“.

3. nagrada — Fotografija Jelene Rupnik, ali nikome neće biti dodeljena.

Pišite na adresu „Peek & poke show“, Računari, Galaksija, BIGZ, Bulevar vojvode Mišića 17, Beograd.

Posle praznika

Bilo je logično zapitati se da li će velike novogodišnje i božićne rasprodaje imati ikakvog uticaja na izbor računara kod nas. Uticaj postoji i veoma je vidljiv — sve „Komisionove“ radnje su prepune računara, pogotovo starih VIC-20 i „komodora 64“, koji su za Novu godinu bili izuzetno jeftini. Šta je sa cenama? Ništa, iste su kao i pre. Verovatno se i kod zvaničnih uvoznika može očekivati povećati kontingent računara (najviše „komodora 64“) kupljenih po nižim cenama, samo što njih ne vredi ni pitati da li će cene kod nas biti nešto niže. Oni to ne mogu iz principa da dopuste.

Vizionari iz „Računara“

Kako se saznaje, istaknuti društveno-politički radnici i nosioci odgovornih funkcija Saša Daković i Branko M. Kovačević pružili su punu podršku novoj reformi reformisanog školstva, a posebno futurističkoj akciji „Na svakih milion učenika jedan računar“. Ovaj novi podstrek usmeravanju presumerone usmerenog školstva će sigurno značiti puno za buduće generacije i nema sumnje da će se neka škola u budućnosti ponositi da ponese ime ovih vizionara.

Visoka svest

Evo još jednog primera našeg aktivnog učestvovanja u svetskim tehnološkim trendovima. Radni ljudi Radne organizacije „Jugospecijalitet“ odlučili su da pošalju svog generalnog direktora, sedam direktora i četrnaest šefova odeljenja na ubrzan kurs bezjeka u Los Angeles, u trajanju od šest meseci. Time su naši radni ljudi još jednom pokazali da shvataju neophodnost uvođenja informatike i računarsu u sve grane privrede.

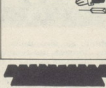
Citat

Anonimni TV citat: „Ovo je „komodora 128“ i on je dva-tri puta bolji od „komodora 64“.“

TV bez Modlija

Televizija Beograd je obnovila svoje Nedeljno popodne. Unutar tog nedeljnog popodneva obnovljena je i prošlogodišnja rubrika Program tvog kompjutera. Na žalost, rubrika je obnovljena tako što je izbačena jedina stvar koja je zaista vredela u njoj — a to je bio Zoran Modli. Bez Zorana Modlija, ta rubrika je zaista nalik ostalim emisijama RTV Beograd — bez mašte, bez šarma.

Pripremaju: Branko Daković i Saša D. Kovačević



Računari, ko Vas kviri!?

Uvod

O računarima pišu ljudi, a ne drugi računari, ili, pak, neke vrste pitomih životinja. I, dok se u ozbiljnijim, računarsko-specijalizovanim listovima i časopisima, ne pojavljuju ljudi koji su sa tom vrstom sprava u najdaljim, ako ne i vrlo zategnutim odnosima, dotle je, u takozvanoj revijalnoj, porodičnoj, magazinskoj i ostaloj štampi, radiju, a pogotovo na malim ekranima naših TV prijemnika situacija sledeća:

Razrada

a) O računarima pišu ljudi koji su imali probleme, i to velikih, i sa računalkama.

b) Ti i takvi poznavaoци materije obično sarađuju i zarađuju u više glasila.

c) Obučavali su se po skraćenim kursovima iz bezjeka, kobola, fortrama, kukičanja i ženjenja, a u poslednje vreme vrlo je popularan tečaj „Naučite komal, brzo klizanje i kuvanje i bićete najbolja partija u ulici, a možda i mnogo, mnogo šire“.

d) Tako naručuzani znanjem oni hrabro kreću u okrsjaje sa računarima i novim simpatizerima računarsu, odvlačeći na krivi put mladu računarsku generaciju.

e) Vrlo su glasni i sveprisutni, baratajući sa oko osamnaest do dvadeset izraza, koje koriste u svakoj pogodnoj i u delu nepogodnih prilika. Istovremeno se kao orlušine obrušavaju na sve svoje protivnike, ne dajući im da dođu do reči, i džojstike.

f) Vrlo su otporni na sve vrste poremećaja i promena klime, a smatra se da će iščeznuti tek kada pokupe sve što se pokupiti moglo.

Zaključak

Ovo je, naravno, sve bilo šala. O računarima pišu i govore pametni, obrazovani, upućeni, jednom rečju stručni ljudi. A i meni se živi još koju godinicu.

Računari u razgovoru: . . . sa dr Branislavom S. Bakićem

zašto smo pali na niske grane

• Druže Bakiću vi ste svedok korišćenja prvih računara u našoj zemlji. Pošto većina naših čitalaca u to vreme nije bila ni rođena, a o pioniskom vremenu primene računara kod nas skoro da nema sistemati-zovanog materijala, recite nam kada i kako je u našoj zemlji započela primena računara.

Početkom pedesetih godina, kada se u Zapadnoj Evropi javio širi pokret za proizvodnju računara, javio se takođe i kod nas. Tada je u SSSR-u počela proizvodnja računara Strela, Ural, Kiev, Minks; u Engleskoj ICT u Francuskoj BULL, a kod nas se javio prvi računar CER koji je konstruisan u Institutu Mihajlo Pupin.

Međutim, dok je u SSSR-u, Francuskoj i Engleskoj nastavljen sa industrijskom proizvodnjom računara, dotle se kod nas stalo na pojedinačnoj proizvodnji. Razlog za to je veoma jednostavan: naša industrija

„Nivo primene računara u statistici bio je visok. U Evropi sigurno na samom vrhu“

nije bila na tom nivou da bi mogla da proizvede računare u većim serijama. Akutne potrebe za računarima (popis stanovništva 1953. godine) dovele su do uvoza računara tada najpopularnijeg proizvođača IBM-a, što je za duži period odložilo domaću proizvodnju računara.

Mada zainteresovanost korisnika za primenu računara u to vreme nije pred industriju, a time ni pred nauku, postavljanje većeg broja računara. Interesenti za primenu računara zadovoljavali su svoj interes uvozom, koji je bio daleko jednostavniji nego razvijanje domaće proizvodnje.

Potencijalni naučni kadrovi odlazili su u inostranstvo i tamo nalazili odlične uslove za naučni rad i život, pa se najčešće više i nisu vraćali u domovinu.

U tom periodu pedesetih i šezdesetih godina bum računarske tehnike bio je na Zapadu tako velik, da su se na samo naučni radnici nego i obični tehničari i programeri bili na visokoj ceni, što je privlačilo naše, ne samo potencijalne naučne kadrove, nego i sve druge stručne kadrove koji su bili u blizoj ili daljoj vezi sa računarima da odlaze na rad u inostranstvo. Tako je u našoj zemlji ostalo veoma malo kadrova koji bi mogli da se bave istraživačkim i inženjerskim radom u oblasti računarstva.

• Ovog meseca, februara 1986. godine, navršava se 35 godina otkako su računari iz laboratorija gde su korišćeni samo u vojne i naučno-tehničke svrhe izašli u svet i krenuli u rešavanje jednostavnijih, ali ne manje važnih, poslovnih problema. Prvi komercijalni računar Ferranti Mark I, kome se ubrzo pridružuju UNIVAC i niz drugih serijski proizvedenih mašina projektovanih specijalno za poslovnu obradu, načinili su preokret u načinu poslovanja velikih preduzeća. Nakon 16 godina, u februaru 1967. godine pojavila se i prva domaća serija od 16 računara „CER-200“. Od tada je proteklo



dovoljno vremena da bi moglo da se sagleda kako se kod nas razvijala poslovna primena računara i gde smo danas u poređenju sa razvijenim zemljama. Možete li nam dati neke kvantitativne pokazatelje koji bi ilustrovali uspešnost ove vrste primene računara?

— Mada naučno-istraživački rad u oblasti računarstva posle prvih uspeha, radovi Dr. Rajka Tomovića i računara CER, nije doživelo procvat, u oblasti primene računara bilo je parcijalnih uspeha.

Primena računara u statistici počela je u Novoj Jugoslaviji vrlo rano. Prvi centar za

„U osamdesete godine smo ušli sa primenom računara 16 puta manjom nego u SR Nemačkoj ili dva puta manjom nego u Čehoslovačkoj“

obradu podataka počeo je sa radom još 1948. godine. Već 1953. godine svi Republički statistički zavodi i Savezni zavod za statistiku raspolagali su sa tada najmodernijim mašinama za obradu podataka. Nivo primene računara u statistici bio je vrlo visok. U Evropi sigurno na najvišem nivou.

Međutim, u drugim oblastima, naročito u privredi, primena je bila dugo vremena na najnižem mogućem stupnju. Tek 1955. godine počinje se sa primenom računara u industriji. Tada su samo tri industrijska preduzeća primenjivala računare u poslovima rukovođenja. Deset godina kasnije, 1965. godine, u našoj zemlji se primenjuje 48 elektronskih računara (približno 2,5 sistema na milion stanovnika). Još deset godina je bilo potrebno da se taj broj poveća na 1494 (1975. godine), uključujući mini računare. Taj broj predstavlja nivo od 25 računara na milion stanovnika, a tada razvijene zemlje imaju nivo: SAD 797, SR Nemačka 337, Francuska 249, Čehoslovačka 46. U osamdesete godine smo, dakle ušli sa primenom računara 16 puta manjom nego u SR Nemačkoj ili dva puta manjom nego u Čehoslovačkoj.

Dok je u periodu između 60. i 80. godine glavni problem u široj primeni računara bio u nedostatku sredstava za uvoz računara (tako se svima činilo, jer nije bilo dovoljno sredstava pa se, logično, dotle se uzimalo kao prvi i jedini razlog), dotle se u sadašnje vreme glavnim problemom smatra nedosta-

tak kadrova potrebnih za širu primenu računara. Danas tzv. domaća proizvodnja nudi daleko veći broj računara nego što se oni mogu plasirati u praksi. Organizacije udruženog rada nemaju sluha za primenu računara, a tamo gde i ima interesa nema spremnih kadrova, na primer organizatora i projektanata, za uspešnu primenu računara.

• Vaša uža specijalnost je upravljanje poslovnim sistemima uz primenu računara, pa nam se čini da merodavno možete da kažete može li i kako primena računara pomoći oporavku naše privrede.

— Savremena privreda zasnovana je na primeni računara. Ako naša privreda želi da ravnopravno učestvuje u svetskoj privredi, mora da na isti način koristi računare kao i ona. Ta istina prihvaćena je kod nas samo u jednom delu, i to u delu industrijske proizvodnje, pa se uvozi oprema koja je zasnovana na primeni računara. Međutim, u mnogo značajnijem delu, u upravljanju poslovanjem, svest o neophodnosti primene računara još nije prodrla u široke mase kadrova, koji upravljaju privredom. Primena računara u čitavom procesu proizvodnje, a naročito u upravljanju, omogućava da se proizvodnja uzdigne na viši, svetski nivo i time čitavo društvo izvede iz ekonomskih teškoća u kojima se sada nalazi.

Primena računara omogućava da se bolje izabiraju proizvodi koji će se proizvoditi, da se pravi više varijanti planova proizvodnje, da se utvrdi stvarno optimalan režim poslovanja, da se neprekidno i bez ikakvog zakašnjenja prati izvršavanje optimalnog režima i plana proizvodnje i u slučaju potrebe reagira pravovremeno i na pravi način. Samo uz primenu računara je moguće proizvoditi na optimalan način, a to znači samo uz primenu računara može se postići željeni stepen korišćenja mašina, sirovina i radne snage.

• Sta je, po vašem mišljenju, uzrokovalo da posle prvih projekata koji su obećavali bistavu budućnost naučno-istraživački rad u ovoj oblasti padne na današnji nivo?

— Zanimanje razvoja naučno-istraživačkog rada u oblasti računara nastalo je kao posledica opšte klime nepridavanja važnosti domaćoj nauci nego oslona isključivo na uvoz. Uvoz kroz uvoznu privredu

„Primena računara u upravljanju omogućava da se proizvodnja uzdigne na viši nivo i time čitavo društvo izvede iz ekonomskih teškoća u kojima se sad nalazi“

donosi sigurne materijalne koristi onima koji uvoze, a domaća nauka te koristi u devizama ne donosi, već na neki pijedestal uzvišenih ličnosti dovodi neke ličnosti koji nisu iz kruga uvoznika.

Zanimarivanje savršenijih oblika domaće proizvodnje, pa čak i onih zasnovanih na uvozu, nastalo je kao posledica opšte klime rasepcanosti potencijalnih proizvođača. Ne samo da je svaka republika izabrala svoj tip računara koji namerala da proizvede, nego su i pojedine regije nekih republika uzela svoje tipove. Za takav sistem proizvodnje nije bilo ni kod jednog od potenci-

Razvoj i primenu računara u našoj zemlji prate mnogi paradoksi. Pre tridesetak godina imali smo izuzetno darovite konstruktore čiji su nas projekti svrstavali među vodeće u Evropi, a danas nemamo ni domaći računar niti racionalno koristimo mogućnosti uvezenih računara. Pokušavajući da utvrdimo zašto primena računara u našoj privredi i našem društvu nije išla uzlaznom linijom, potražili smo jednog od učesnika pionirskih dana korišćenja računara, Branislava S. Bakića, doktora ekonomskih nauka iz oblasti privredne kibernetike, dugogodišnjeg urednika časopisa Automatizacija poslovanja i direktora Centra za automatizaciju poslovanja. Branislav Bakić autor je više knjiga, priručnika, članaka i studija iz oblasti upravljanja poslovnim sistemima uz primenu računara i projektant brojnih projekata primene računara u organizacijama udruženog rada.

jalnih proizvođača niti dovoljno kadrova, a naročito ne dovoljno novčanih sredstava. Još 1982. godine Savezna privredna komora konstatovala je da 30 proizvođača želi vlastitu proizvodnju računara, a to je, svakako, previše i za SAD ili Japan.

• *Od prvih prve serije domaćih računara, cena ulaska u proizvodnju računara svake godine se duplirala, pa je danas, po mišljenju nekih eksperata, potrebno investirati više od 50 miliona dolara da bi se napravilo nešto što spada u informacijsku tehniku. Svima je jasno da ni jedan od pomenutih proizvođača ne može imati ni približnu količinu sredstava. Imamo li mi kao zemlja neku realnu šansu da se konkurentno uključimo u svetsku računarsku industriju?*

— Dosadašnji način snabdevanja korisnika računara bio je kroz uvoz. Zadnjih godina počinje se sa tzv. domaćom proizvodnjom. Poznato je da se proizvodnja računara može zasnovati uglavnom na četiri osnovna načina: preko vlastitog razvoja, što je najbolji ali i najteži put; sklapanjem OEM-a i kupovinom delova na svetskom tržištu delova, pa sklapanjem u finalni proizvod po vlastitoj želji; proizvodnom kooperacijom, pri čemu naši proizvođači proizvode neke delove, a inostrani kooperanti druge delove koji se sklapaju u celini po projektu inostranog partnera; i licencnom proizvodnjom na bazi licence inostranog proizvođača. Naša domaća proizvodnja da-

„Kroz uvozu proviziju uvozi domosi sigurne materijalne koristi onima koji uvoze, a domaća nauka te koristi u devizama ne donosi“

nas još nije prevazišla fazu čiste licencne proizvodnje, ali se javljaju veoma ozbiljni pokušaji da se proizvodi na bazi kooperacije, sklapanja OEM, a u oblasti automatizacije proizvodnje i na bazi sopstvenog razvoja. Može se reći da savremenu koncepciju proizvodnje računara ipak sačinjava shvatanje da će jedini pravi izlaz za domaću proizvodnju računara biti u oslanjanju na sopstvene snage u istraživanju i razvoju. Naravno, realizacija ove koncepcije zasniva se na udruživanju snaga svih potencijalnih proizvođača, jer ako do sada najavljenih više od 30 proizvođača u našoj zemlji ne shvati da samo kroz zajednički rad mogu da ostvare takvu koncepciju, uspeha neće biti. Vrlo je važno da se ovo što ne vidi jer nas neprečeno trendova u svetskoj naučno tehnološkoj revoluciji dovodi u zavisnost od razvijenih zemalja sa kojima trujemo ili od njih pozajmljujemo kapital.

— *Jedini izlaz je da razvijamo domaću nauku u svim onim oblastima gde je to opravdano obzirom na perspektivu našeg razvoja. Informatika je svakako oblast u kojoj mi možemo da postignemo mnogo i to ne samo da se oslobodimo uvoza nego i da postanemo i izvoznici. Mi imamo potencijalne kadrove, masu opšte školovanih*

kadrova, koji se mogu specijalizovati za rad u ovoj oblasti. Razvoj informatike kod nas učinio bi mnogo za podizanje čitave privrede na viši nivo i za smanjenje naše zavisnosti od kolonijalne dominacije inostranih poverilaca i snabdevača.

• *Ali, dok se mi usaglašavamo drugi ne sede skrštenih ruku. Tehnologija se sve brže razvija i jaz između nas i razvijenih svakim danom je sve veći. Kako ga premostiti?*

— *Jedini put za smanjenje jaza u tehnološkom razvoju između razvijenih i nerazvijenih je podizanje svesti širokog kruga samoprovajlača o posledicama tehnološkog zaostajanja. Ako oni koji odlučuju o investiranju kapitala misle da je jedini put njihovog materijalnog zainteresovanosti uvoz tujih proizvoda, oni neće dozvoliti da se razvija domaća nauka i domaća proizvodnja ključne opreme. Da bi samoprovajlači*

„Zanemarivanje savršenijih oblika domaće proizvodnje računara, pa čak i onih zasnovanih na uvozu, nastalo je kao posledica opšte klime rascepanosti domaćih proizvođača“

bili sposobni da oduzmu stvarno pravo odlučivanja o investiranju pojedinih vladajućim strukturama moraju postati svesni svoje lične zainteresovanosti za razvoj domaće nauke i tehnologije. To podizanje svesti samoprovajlača je dug proces i nije zasnovan samo na verbalnom ubeđivanju nego i na praktičnom uveravanju na pozitivnim primerima iz prakse. No, mislim da bi za čitavo celo mnogo interesantnije da umesto mog mišljenja o ovom pitanju čuju mišljenje onih koji se nalaze na mestima sa kojih se može dejstvovati na kreiranje politike.

• *Mišljenja onih koji su u poziciji da kreiraju politiku čitao i tako stalno slušaju, ali slabo se vide rezultati. Kao što nam se čini da bi svakako od nas sastavio bolju fudbalsku reprezentaciju od državnog selektora kad ona zvanično gubi, tako verujem da bismo mogli da uvedemo red i u proizvodnju i primenu računara. Stoga nam, ipak, recite, koje biste mere vi preduzeli za smanjivanje jaza u tehnološkom razvoju između nas i razvijenih, kao i u okviru „jedinstvenog“ YU tržišta kada biste imali odgovarajuća ovlašćenja.*

— *Široka primena računara zasniva se na opštem shvatanju čitave društvene zajednice da je primena računara korisna za zajednicu, nužna u sadašnjosti, a naročito u budućnosti, da je usmerena u korist a ne na štetu čoveka, da ga oslobađa mukotropskog rada i omogućava mu da mu rad postane zadovoljstvo, a ne da ga ostavi bez posla.*

— *Stvaranje opšte društvene svesti o korisnosti primene računara neophodna je angažovanost čitave društvene zajednice. S jedne strane političkih struktura, a s druge strane stručnih kadrova svih struka. Da bi se to postiglo moraju se primeniti određene mere. Pre svega treba organizovati naučnoistraživački rad u pravcu razrade koncepcije primene računara u pojedinim oblastima i osnovati škole koje bi osposobljavale kadrove potrebne za razradu te koncep-*

cije i njenu realizaciju u praksi. U postojećim srednje, više i naročito visoke škole obavezno treba uvesti nastavu o primeni i korišćenju računara. Takođe bi trebalo omogućiti školovanje naših kadrova van zemlje, u školama, institutima ili preduzećima koja primenjuju računare sa velikim uspehom. Naročito u periodu dok naše školske, naučne i privredne institucije nisu u stanju da pruže potrebnu pomoć kadrovima.

• *Može li se prevazići apsurd da se za korišćenje računara u mnogim našim srednjim školama, pa čak i na nekim fakultetima, obučava bez računara?*

— *Ovo se može ublažiti osnivanjem jednog ili nekoliko eksperimentalnih preduzeća koja bi poslužila, s jedne strane, kao demonstracioni centri za upoznavanje društva sa idejom primene računara, a s druge strane kao laboratorije srednjih i visokih škola. Isto tako treba osnivati i specijalizovane organizacije udruženog rada koje bi se bavile pružanjem pomoći organizacijama koje se spremaju ili uvode računare. Važno je razviti širok pokret za podizanje opšte kulture u oblasti primene računara, naročito u pravcu upoznavanja čitavog društva sa koristima koje od njihove primene može da ima. Temelji ovog pokreta moraju biti, pored stručne štampe i popularnih napisa, naročito obrazovne ustanove svih vrsta, kao i televizija. Međutim, sve ovo neće dati rezultate ako se ne donese dugoročni plan razvoja primene računara koji treba uklopiti u plan razvoja čitave privrede i društva kako bi razvojni put ove oblasti bio jasan za sagledavanje svima, a naročito kadrovima koji treba da ulože velike napore, najpre u svoje lično osposobljavanje, a zatim u vrlo težak i odgovoran praktični rad. Oni, to, bez jasne perspektive fiksirane u najvažnijim društvenim aktima, verovatno, har kako to pokazuje sadašnja praksa, neće biti skloni da urade.*

• *Recite nam još, na kraju, mislite li da sredstva informisanja mogu da doprinesu povećanju nivoa opšteg računarskog obrazovanja.*

— *Sredstva informisanja su jedan od ključnih činilaca u stvaranju shvatanja o neophodnosti primene računara, pod uslovom da ovom problemu pridru odgovorno, a ne senzacionalistički. Da objašnjavaju stvarnu korist od primene računara a ne da*

„Informatika je oblast u kojoj možemo da postignemo mnogo — ne samo da se oslobodimo uvoza nego i da postanemo izvoznici“

prizikaju računare kao sredstvo upereno protiv čoveka ili kao nešto što može samo po sebi rešiti sve moguće probleme pridaajući računarnima moć koju oni nemaju. Nije teško da se vidi društvena korist od primene računara i šteta od neprimenjenosti ako se čitava zajednica pravilno informiše o potrebama i budućem razvoju primene računara. jer sadašnjiji otpori primeni računara bili su posledica isključivo neobaveštenosti o koristi koje mogu imati od primene računara kako pojedinci tako i društvo u celini.

Razgovor vodila: Nevenka Spalević

John Naisbitt

★★★ MEGATRENDOVI Deset novih smerova razvoja koji menjaju naš život

John Naisbitt donosi nov način sagledavanja američke budućnosti i nov način razumevanja sadašnjosti. On kaže: „Prelazimo iz industrijskog u informatičko društvo i tjelesnu snagu nadomještati će stvaralačka snaga uma, a suvremena tehnologija povećati će i unaprijediti naše umne sposobnosti. To će omogućiti porast zaposlenosti i ulaganja u industrije u usponu, ali ne smijemo izgubiti iz vida nužnost postizanja ravnoteže između ljudskog elementa i tehnologije“.

MEGATRENDOVI su informativna, zanimljiva i dinamična slika društva u kom je budućnost već počela!
Cena: 2.600 dinara

Fred d'Ignazio

★★★ UVOD U KOMPJUTORE

Ova popularno pisana knjiga je vodič za svet kompjutera. Sta je kompjuter, od čega se sastoji, koji su glavni konstruktori i proizvođači, kako se kompjuter može upotrebiti?

To su samo neka od pitanja na koja će čitatelj naći odgovore u ovoj knjizi. Osim toga, na kraju knjige je rečnik pojmova i termina koji se najčešće upotrebljavaju u vezi s kompjuterskom tehnikom.

Ako ste želeli da na jednom mestu nađete kratku preistoriju i povest kompjutera, biografije i fotografije glavnih protagonista kompjuterskog buma, sažet pregled načina i polja primene kompjuterske tehnike, onda je ovo prava knjiga za vas.

Cena: 2.200 dinara

David Baker

★★★ LASERSKI IZAZOV — RAT ZVIJEZDA

Na popularan, ali naučno i tehnički korektan način, David Baker obrađuje trku u naoružanju dveju supersila. Započeta lansiranjem prvih zemljinih satelita i interkontinentalnih balističkih projektila, ta trka je u naše vreme obeležena razmišljanjem i pregovorima supersila o mogućnosti takozvanog „rata zvezda“, o mogućnosti lansiranja i stavljanja u orbitu snažnih laserskih oružja s energetskim snopom subatomarnih čestica.

Mogući scenarij „rata zvezdada“ u kom svemirska oružja usmerene energije odozgo uništavaju neprijateljske rakete pretvara se u stvarnost. Laserski izazov je odsad pa nadalje prateća konstanta u razvoju oružja budućnosti. A time i budućnosti same.

Cena: 3.000 dinara



★★★ EINSTEINOVA OPĆA TEORIJA RELATIVNOSTI

Priredio: Gerald E. Tauber

Ova knjiga je svojevrsan zbornik međusobno povezanih tekstova Alberta Ajnštajna i dvadesetak drugih vrhunskih fizičara. Tema svih priloga je opšta teorija relativnosti, delo koje se smatra vrhunskim dometom ljudske misli na području znanosti. Ključni pojmovi i stavci ponavljaju se na više načina u raznim tekstovima, što će čitaocu olakšati razumevanje i hvatanje glavne niti izlaganja.

„Najnešvatljivije na svijetu je to da je on shvatljiv“, napisao je Ajnštajn izražavajući svoju veru da se iza svekolike zamršenosti sveta nalaze jednostavni principi kojima se pokorava celi univerzum.

Cena: 2.500 dinara



CGP DELO OOUR GLOBUS
Predstavništvo Beograd
11 000 BEOGRAD
Bulevar vojvode Mišića 14 (Hala 4)

NARUŽBENICA / „Galaksija“ 1, 86.

Neopozivo naručujem knjigu-e pouzdem uz 20% popusta (potcrpati traženi naslov-e) — plaćanje poštaru prilikom preuzimanja knjiga:

John Naisbitt — MEGATRENDOVI, komada _____
Fred d'Ignazio — UVOD U KOMPJUTORE, komada _____
David Baker — LASERSKI IZAZOV — RAT ZVIJEZDA, komada _____
EINSTEINOVA OPĆA TEORIJA RELATIVNOSTI, komada _____

Ime (ime oca) i prezime _____

Broj lične karte i mesto izdavanja _____

Tačna adresa _____

M.P.

Datum

Svojeručni potpis

Istorija
računara
35 godina

IBM i sedam patuljaka

Ovog meseca se navršava trideset i pet godina od pojave prvog komercijalnog računara. Ovo je priča o izlasku računara iz univerzitetskih laboratorija i njihovom pohodu na nove oblasti primene. Ko je IBM a ko su patuljci i zašto se baš tako zovu otkriveno posle analize kako su se proizvođači snašli u poplavi zahteva za simbolom poslovnog uspeha pedesetih godina, a danas, van svake sumnje, neophodnim preduslovom za normalno poslovanje i uključivanje u svetsku podelu rada. Početak priče vezuje se za različita mesta — sve zavisi od toga ko pripoveda. Voleli bismo da možemo da kažemo da je pohod počeo sa brdovima Balkana, ali držimo se činjenica i zato moramo reći — Počelo je na Ostrvu u našem komšiluku, a ne preko „velike bare“.

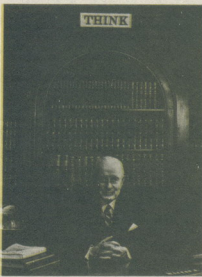
Mančesterski kompjuter MARK i uspešno je počeo sa radom u junu 1948. godine kao prvi računar na svetu šaš unutrašnjim programom. Stvaranjem mogućnosti za korišćenje „Vilijamsove cevi“ za memorijalne uređaje, proizvedena je poboljšana verzija MARK-a i koja je korišćena za rad na optičkom dizajnu i za generisanje prostih brojeva. Britanska vlada, koja je nameravala da koristi računare u sopstvenim projektima za atomsku bombu, finansijski je podržala predlog svog šefa za nauku Sir Bena Lokspira (Ben Lockspier) impresioniranog mogućnostima ovog računara da se komercijalna verzija MARK-a i proizvodi u lokalnoj mančesterskoj kompaniji. Tako je ova komercijalna verzija pod nazivom Ferranti MARK I bila na tržištu već u februaru 1951. godine, pet meseci pre UNIVAC-a, pa je možemo smatrati prvim komercijalnim računarem na svetu.

U Ferranti Mark I bile su ugrađene mnoge inovacije, ali jedna od najznačajnijih bila je mogućnost modifikacije instrukcija za vreme izvršavanja programa. Sa ovom mašinom, koja je po mogućnostima bila ispred svog vremena, Ferranti je prvi ušao u kompjuterski biznis i prodao osam Mark-ova u narednih nekoliko godina.

Još jednom prvi

Džon Moučli (John Mauchly) i Presper Ekert junior (Presper Eckert), konstruktori ENIAC-a, prvog elektronskog računara na svetu, posle priznanja i para koje su dobili za svoj projekat, po starom američkom običaju osnivaju svoju kompaniju za proizvodnju kompjutera — Ekcert-Mauchly Computer Corporation. Međutim, vrlo brzo se pokazalo da bolje umju da konstruišu nego da prodaju računare. Da bi mogli da realizuju svoju novu vizionarsku ideju — univerzalni automatski računar (UNIVERSAL Automatic Computer) koji bi mogao da se koristi i u biznisu, morali su da se nađu pod patronatom neke finansijski moćnije kompanije.

Na žalost, u to vreme većina ljudi, pa čak i onih koji su se projektovanjem računara profesionalno bavili, nisu uvidela svrhu upotrebe kompjutera van univerzitetskih laboratorija, pa ni projekt UNIVAC-a nisu ozbiljno shvatili. Između ostalih, nije pristao da otkupi njihovu firmu čak ni IBM,



Osnivač IBM-a: Tomas Votson

Mislo je kratko

Kada je Vilijams (F. G. Williams), tvorac „vilijamsove cevi“ (memorijskog uređaja na bazi katodne cevi) i profesor mančesterskog univerziteta na kome je načinjen niz značajnih inovacija u računarstvu, bio u upravi IBM-a prilikom svoje posete Njujorku, upitali su ga kako to da mančesterski tim tako produktivno radi na novim tipovima računara, iako se u njega ulaže znatno manje sredstva nego što se to čini u Americi. Budući da je i veliki IBM koristio mnoge od mančesterskih patenata, Vilijams je uz osmeh, pokazujući na jedan od mnogih omiljenih transparenta starog vlasnika kompanije Votsona na kome je pisalo „MSL!“, rekao: „Nismo se suviše zadržavali na razmišljanju.“

najmoćniji proizvođač kancelarijske opreme. Srećom, Remington Rand, stari rival IBM-a, imao je više sluha za inovacije koje su nudili Ekert i Moučli i u februaru 1950. godine uzeo ih je u svoje okrilje. Koliko je to bio dobar potez pokazao je već popis iz

martha 1951. godine za čiju je obradu korišćen prvi UNIVAC.

UNIVAC je imao nekoliko neobičnih svojstava za mašine tog vremena. Koristio je poseban program za prevodenje programa u mašinski jezik (compiler), a sami programi čuvani su, kao i podaci, u unutrašnjoj memoriji. To je bila prva mašina sposobna da obrađuje kako čisto numeričke, tako i alfabetske podatke. Osim toga, ako izuzmemo britanski Mark I koji je imao znatno manje tržište, to je prva mašina sačinjena ne po specijalnoj porudžbini već za slobodnu prodaju.

Ipak, malo je kompanija koje su poput Remington Randa ikada izbacile tako izvanredan proizvod, a prodavale ga tako loše i nesposobno. Ba nije tako bilo, možda danas ne bismo mogli reći da je priča o komercijalnim računarima, u stvari, priča o proizvođaču koji već više od 30 godina podmiruje preko 60% svetskog tržišta — priča o IBM-u.

Saga o Votsonima

Firma koju je osnovao 1924. godine i briljantno vodio Tomas Votson senior (Thomas J. Watson 1874—1956) brzo je zauzela vodeće mesto među proizvođačima kancelarijskih mašina. Već 1939. godine imala je veći profit od svojih glavnih konkurenata Remington Rand-a, NCR-a (National Cash Register) i Boruza (Burroughs), a posle II svetskog rata i njen ukupni obrt kapitala bio je najveći. Tada već ostareli Votson senior nije bio naročito zainteresovan za ulazak u nove tržišne poslove sa elektronikom i kompjuterima, ali svestan značaja reklame ulagao je sredstva i u finansiranje računarskih projekata. Tako je 1939. godine preuzeo finansiranje konstrukcije Automatskog sekvencijalno kontrolisanog kalkulatora (Automatic Sequence-Controlled Calculator), poznatijeg pod imenom Mark I Hauardski Ajkna (Howard Aiken). (Ovaj harvardski projekat nema veze sa desetak godina mladim istoimenim engleskim računarnom.)

Had na harvardskom Mark-u I trajao je oko pet godina i avgusta 1944. računar je zvanično predstavljen. Želeći da zadrži za sebe najveći deo priznanja, Ajken je na toj ceremoniji jedva pomenuo IBM, što je strašno razjutilo Votsona. Njegov stariji sin i naslednik kompanije Tom Votson je kasnije izjavio: „Da su Ajken i moj otac tada imali revolvere, danas bi obojica bili mrtvi.“ No,

I pored ovog nesporazuma, investicija od 500.000 dolara, koliko je koštao Mark I, i činjenica da je ovim računarom, koji je imao sve osnovne blokove „analitičke mašine“ — uređaje za ulaz i izlaz, memoriju i upravljački i aritmetički organ — ostvarena mašta Garisa Bebidza, jako je podigla ugled IBM-a u akademskim krugovima. Mada je Mark I predstavljao veoma skupu vezbu u zastareloj tehnologiji već prevaziđenoj izumom ENIAC-a, ipak je iskustvo stečeno na njegovom projektu i realizaciji značajno koristilo IBM-u u proizvodnji novih tipova računara.

U skladu sa tipično konzervativnim pristupom, umesto da ulaze u projekte elektronskih računara, IBM 1945. godine ulaze sredstva u projekat Valas Ekerta (Wallace Eckert, šef kompanijine naučne laboratorije, nema veze sa Presperom Ekertom) Selektivnog sekvencijalnog elektronskog kalkulatora — SSEC. SSEC je bio tehnološki hibrid koji se sastojao od 12.500 cevi i 21.400 elektromehaničkih releja. Cevi su korišćene za vršenje aritmetičkih operacija i beleženje male količine podataka i instrukcija (osam dvadesetocifrenih dekadnih brojeva). Releji su služili kao sprjela ali veća unutrašnja memorija za 150 brojeva. Još 20.000 brojeva moglo je biti upamćeno

Računar iz kuhinje

Put do relejnog računara vodio je preko kuhinjskog stola — Džordž Štibic (George Stibitz), matematičar zaposlen u „Belovim laboratorijama“ primetio je sličnost između brojanja pluseva i sabiranja. Rađajući kod kuće na kuhinjskom stolu i koristeći nekoliko starih krosbar prekidača i elektromehaničkih releja, načinio je prva relejna kompjuterska kola. Štibic je tada potražio pomoć iskusnog inženjera Semjuela Vilijamsa (Samuel Williams) koji je već 25 godina radio na prekidačkim kolima i uskoro su ova dva naučnika stvorila relejni računar kome su dali naziv Complex Number Calculator. CNC počeo je da se koristi 8. januara 1940. godine.

na 66 koturova bušene papirne trake. Instrukcije i podaci prebacivani su sa traka na releje, a zatim na cevi dove su se instrukcije izvršavale. Mada je ovako izvršavanje instrukcija dosta trajalo, relativno velika memorija davala je ovom računaru veće mogućnosti od ENIAC-a. Ovaj projekat nije imao skoro nikav uticaj na dalji razvoj kompjutera, ali ga pominjemo jer su neki eksperti, razume se IBM-ovi, proglasili ovaj računar završen u januaru 1948. godine prvim računarom na svetu sa unutrašnjim programom. Ovo je samo delimično tačno. Istina je da su njegove instrukcije i podaci bili kodirani brojevima i pohranjeni u unutrašnjoj memoriji, ali računar nije bio potpuno automatski. Koristio je nekoliko tabli sa utičnicama na kojima se mail, ali značajan deo programa zadavao odgovarajućim povezivanjem žica, stoga mančesterski Mark završen nekoliko meseci kasnije ipak smatrano prvim računarom sa unutrašnjim programom.

SSEC je bio instaliran u izlogu IBM-ove zgrade na Menhatnu i izazivao je veliku pažnju prolaznika. Ljudi su se okupljali pred izlogom i fascinirano posmatrali trenjanje sijalica na kontrolnom panelu, okreta-



Tvorci legendarnog UNIVAC-a: Presper Ekert i Džon Močli



Neobičan početak: Prvi komercijalni računari u Engleskoj nastali su u ovoj prodavnici čaja

nje koturova trake i štampače koji su izbacivali podatke fenomenalnom brzinom od 30.000 cifara u minuti. IBM-u nije bila potrebna bolja reklama.

Odbrambeni kalkulator

Zahvaljujući ovim projektima i nizu drugih manjih, Vlade agencije uputile su IBM-u zahtev da za njihove potrebe načini elektronski računar koji je u projektu imao naziv „odbrambeni kalkulator“ (Defense Calculator). Međutim, IBM nije htio da Vladi bez patentnih prava daje projekat, pa su njegovi stručnjaci uradili finansijsku konstrukciju po kojoj bi se projekat i proizvodni program finansirali sopstvenim parama. Po njihovoj proceni, bilo je neophodno uložiti tri miliona dolara, a to je i za

prebogat IBM bio veliki zaloga. U firmi su postojale dve struje: jedna na čelu sa mladim Votsonom, koji se kao vojni pilot upoznao sa mogućnostima radara i drugih elektronskih naprava i pripadao generaciji

Prvi domaći računar

Prvu domaću seriju računara „CER-200“ završio je bez pomoći Industrije početkom februara 1967. godine institut „Mihajlo Pupin“. Još pre no što je bila završena nulta serija, primljeni su zahtevi za izradu novih 50 mašina zahvaljujući ceni od 22 miliona starih dinara + 4.500\$ u vreme kada su uvozne mašine skromnijih mogućnosti koštale oko 28.000 dolara (oko 35 miliona starih dinara).

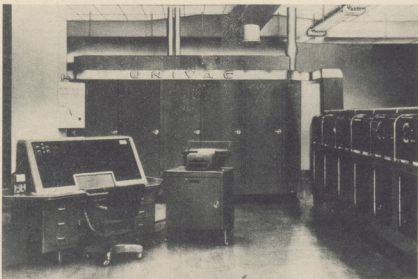
projektanata ENIAC-a i bio „za“ i druga, konzervativna, koja je smatrala da je bolje izvršiti ulaganja u već proverenu tehnologiju.

Odeak za planiranje oštro se suprotstavio projektu sa obrazloženjem da kroz pedesete godine ni jedan kompjuter neće biti iznajmljen po ceni većoj od 1.000\$, a da bi se opravdala ulaganja, cena iznajmljivanja bi morala biti bar 8.000\$. Tom Watson junior se nije predavao. Za svaki slučaj, njegovi savetnici prikupili su 30 garantnih pisama budućih korisnika računara i u martu 1951. godine već je postojao projekat na papiru. Videlo se da cena iznajmljivanja mora biti izuzetno velika, čak 15.000\$, ali kako je većina mušterija ostala pri svojim garantnim pismima, „odbrambeni kalkulator“ počeo je da se proizvodi 1952. godine. Pre no što je njegova proizvodnja počela, u IBM-u je bilo zaposleno samo nekoliko elektroinženjera, a za naredne dve godine zaposleno ih je skoro 500. U tom prvom talasu zapošljavanja visokostručne radne snage hardveraši su bili na visokoj ceni.

Prvi IBM 701, kako je „odbrambeni kalkulator“ iz projekta nazvan, isporučen je Los Alamosu u martu 1953. To je bila binarna mašina u fon Nojmanovom stilu, sa 4000 cevi, read/write elektrostatičkom memorijom od 4 K 36-bitnih reči i mogućnošću da pomnoži dva broja za 456 mikrosek. Računar je bio pun inovacija, uključujući i tako trivijalne detalje kao što je pakovanje — bio je izrađen u delovima koji su pakovani u kutije od kojih je svaka mogla da stane u standardni lift i prođe kroz obična vrata. IBM je konačno na velika vrata ušao u kompjuterski biznis. Ni jednog trenutka njegovo tržište nije padalo ispod 50% u SAD i 60% svetskog tržišta. Pod vodstvom Tomasa Watsona mlađeg, koji je imao neopredeljiv instinkt za nove tehnologije i trgovačku umešnost svoga oca IBM je sve ove godine uvodio prave mašine u pravo vreme. Tako je već 1955. proizveo računar u kome je 1250 elektronskih cevi zamenjeno sa 2200 tranzistora čime je potrošnja energije smanjena za 95%, a danas je u njegovim laboratorijama razvijeno nekoliko kompjuterskih elemenata koji koriste nove tehnologije, na primer tehnologiju baziranu na Josephson efektu. Danas jedino Japan i Velika Britanija imaju nezavisne kompjuterske kompanije koje mogu da se takmiče sa IBM-om na pospovnim tržištima. Nećemo se zadržavati na japanskom tehnološkom čudu, čija bi analiza zahtevala mnogo više prostora, već ćemo se vratiti tridesetak godina unazad u Englesku da vidimo šta su to Englezi preduzeli što je omogućilo da sve vreme ostanu u vrhu kompjuterskog biznisa.

Lajonsov udeo

Komercijalni kompjuteri u Engleskoj vuku deo svojih korena sa prilično neobičnog mesta. Inventivna ideja da se počne sa proizvodnjom računara za automatizaciju kancelarijskog poslovanja potekla je od kompanije Lajons (Lyons) koja je posedovala niz malih prodavnica prehrambenih proizvoda i čaja. Ova kompanija imala je dugu tradiciju inovacija u poslovnim metodama. Uvela je kalkulatorske mašine u svoje prodavnice još davne 1896. godine, a od tridesetih godina ovog veka eksperimentisali su sa mikrofilmovima na kojima su bile zabeležene informacije o svim finan-



Nezasluženi primat: UNIVAC nije, kako se to često misli, prva mašina na svetu sa unutrašnjim programom; primat, zapravo, pripada Englezima

Prvi mikroprocesor

Prvi mikroprocesor 4004 razradio je Intel krajem 1970. godine po projektu Marsijana Hofa (Marčian Hoff), koji je želeći da smanji broj čipova u kalkulatorima došao na genijalnu ideju da načini čip koji bi bio nalik na centralne procesore velikih računara. To je bio četvorbitni procesor koji je obradivao podatke u blokovima od po 4 binarne cifre. Sa 2250 tranzistora mogao je da obavlja 60.000 operacija u sekundi.

sijskim transakcijama. Svakih nekoliko godina Lajons je slao svoje planere u inostranstvo radi proučavanja novih metoda u poslovanju, pa su tako dva njegova službenika 1947. godine otišla u Ameriku da na delu vide „elektronske mozgove“ koji su bili glavni hit tog vremena. Najznačajnija informacija koju je Lajons dobio sa tog studijskog putovanja bila je da se „elektronski mozgovi“ proizvode i mnogo bliže — u Kembridžu.

Na osnovu analize koja je pokazivala da bi u pospovni kompjuter trebalo investirati 100.000 funti, ali da bi njegova primena godišnje uštedela 50.000 funti, kompanija je u oktobru 1947. godine započela rad na projektu pospovnog računara. Ono što izavisa posebno poštovanje je činjenica da se kompanija odlučila za ovaj projekat još u vreme kada je na Kembridžu račun ED-SAC bio u projektu. Za početak, stipendiral su sa 3.000 funti rad na EDSAC-u i time bitno ubrzali njegov završetak tako da je on već 1949. sa uspehom obavio svoj prvi posao — računanje tablice prostih brojeva.

Lajons je analizirao probleme koje bi njegov kompjuter trebalo da rešava i skicirao rutine koje su mu za to neophodne. Ove studije predstavljale su osnovu za prve programe i pomogle da se definišu hardverska rešenja. Već na samom početku bilo je jasno da se poslovni računar mora suštinski razlikovati od univerzitetske mašine koja se upravo razvijala. Dok je EDSAC bio projektovan za izvršavanje komplikovanih i dugotrajnih matematičkih izračunavanja, pri čemu je sa ulaza zahtevao samo

nekoliko podataka, poslovni računar trebalo je da rešava sasvim drugačije probleme. Matematičke operacije u poslovnim obradama su veoma jednostavne, ali se moraju vršiti nad mnoštvom podataka.

LEO (The Lyons Electronic Office) je počeo da se koristi od 9. februara 1954. godine za obračun nedeljnih plata za 1.700 zaposlenih. LEO je za 1,5 sekundi obavljao posao za koji je službeniku bilo neophodno najmanje 8 minuta.

U Lajonsu su ubrzo uvideli da je potrebno imati više ovakvih računara, a postojao je i ogroman interes drugih firmi, pa se Lajons počeo baviti i proizvodnjom i prodajom kompjutera. Tako je iskustvo stečeno projektovanjem LEA iskorišćeno za izgradnju čitave serije poboljšanih verzija ovog računara koji su sa uspehom korišćeni u biznisu, vladinim organizacijama i industriji. Zahvaljujući kvalitetnim računarima koje je nudila domaća industrija, Velika Britanija odošla je u inaviziji IBM-a i sedam patuljaka, koji i danas žare i pale na evropskom tržištu komercijalnih računara.

Savremenu industriju velikih računarskih sistema danas vode IBM i sedam patuljaka, kojima jedino Japan može da urabi deo tržišta. IBM je finansijski tako ojačao da godišnje obrne preko 20 milijardi dolara. Patuljci — Burroughs, NCR, Control Data, Sperry Rand, DEC, Honeywell i Hewlett-Packard, svaki sa godišnjim obrtom kapitala od oko 3 milijarde dolara — imaju takvu finansijsku moć da je teško zamisliti da će u skoroj budućnosti neka nova firma moći da im ugrozi tržište. Međutim, lekcije iz računarske istorije kazuju da su kvalitetno novi proizvodi i nove tehnologije dolazili sa neočekivanih mesta i takvom brzinom da su „veliki“ jedva uspevali da ih prate. Nisu li primeri personalnih računara i Apple-a, firme koja je imala najbrži rast u istoriji američke berze, najbolja potvrda ove teze. U tome se, možda, krija i naša šansa da uđemo u istoriju kompjutera.

**Nevenka Spalevic
Branko Dakovic**



Dejanove
pitalice

Nagraana premetaljka

Naš treći nagradni zadatak je pokazao da su čitaoci „Računara“ poštteni radni ljudi — nije ih mnogo interesovala godina sa najmanjim brojem radnih dana.

Obzirom da smo u predviđenom roku dobili samo tri potpuno tačna rešenja (stiglo je samo pedesetak odgovora, možda zato što uz zadatak, tehničkom greškom, nije objavljen nagradni kupon), nije nam bilo teško da rasporedimo nagrade. Prvu nagradu od 10.000 dinara je dobio **Franjo Gospodnetić** iz Zagreba, drugu (5.000 din) **Zlatko Tratar** iz Ljubljane a treću (3.000 dinara) **Geza Berdalučić** iz Bilja.

Šta je najviše namučilo rešavače? Pre svega, nepoznavanje kalendara: po Gregorijanskom („novom“) kalendaru, godine kojima se završavaju vekovi su prestupne samo ako je njihov broj deljiv sa 400, što znači da je 2000. godina prestupna ali da 2100, 2200. i 2300. nisu. Kalendar se, dakle, ponavlja svakih 400 godina, što je bio osnovni razlog da u postavlci problema pomenomno razdoblje 1986—2386. Sledeći je problem bio što je za rešavanje zadatka **zaista** bio potreban računar; bez njega se teško može obraditi veliki broj godina tako da rezultati budu pouzdani. Dobili smo dvadesetak pisma u kojima su navedeni programi koji bi trebalo da reše problem ali ne i rezultati izvršavanja tih programa — izgleda da u našim krajevima i dalje ima više programera nego računara. Kao tačna rešenja **priznajemo** pisma u kojima nije naveden postupak rešavanja (takva pisma, doduše, retko konkuriraju sa prvu nagradu) ali **ne priznajemo** pisma sa programima u kojima nije naveden konkretan i precizan odgovor na postavljeno pitanje! Poželjno je da taj odgovor ne bude zakopan u sredinu (nečitko pisanog) teksta.

Na slici su data dva rešenja problema: jedno je delo Frana Gospodnetića pisano na kod nas malo poznatom jeziku C i izvršavano na nekoj CP/M mašini a drugo primer umešnosti Zlatka Trataru na Sharpu M7700.

Pokazuje se da najviše neradnih dana (po 114) imaju prestupne godine kod kojih prvi januar pada u nedelju. To su 2012, 2040, 2068, 2096, 2108, 2136, 2164, 2192.

```

118 REM PROGRAM ZA IZRANJE METJE I NAJVICESTEVILJE STEVILA PROSTIH DNI
120 REM PROGRAMMER: ELATKO TARATAR DATUM: 05.01.95 RACUNARNIK+SHARP M7-700
130 I
140 I
150 REM XXX PRIZNANA XXX
160 DIM P(1,3),D(1,4)
165 FOR I=0 TO 31 READ P(I),D(I),I:31 NEXT I
170 FOR I=0 TO 61 READ PD(I),I,D(1),I:31 NEXT I
180 PD=0:TD=0
190 I
200 I
210 REM XXX SLAUNI PROGRAM XXX
220 FOR LL=1085 TO 2385
230 TD=0
240 REM XXX IZRANJEN STEVILA DNI U TEKOCEN LETU XXX
250 XI=INT(11*100):XJ=LL*100:XK=INT(11*100):XL=LL*100
260 IF (XI=XJ)(XJ=XK) THEN DL=365:GOTO 280
270 DL=366-SIN(11-INT(11*100))
275 I
280 REM XXX IZRANJEN STEVILA DELA PROSTIH DNI U LETU XXX
290 FOR I=0 TO 3
300 DD=P(1),D(1):PP=P(1),I
310 REM XXX DOLOCITEV OSENA U TEČNJ: ZELERJEVA KONJURENCIA XXX
320 DL=LL
330 IF PP=2 THEN PP=PP-2:GOTO 340
335 PP=PP+18:L=L+2-I
340 L1=INT(L/2)*100:L2=L-INT(L/2)*100:R1=0
350 DD=DD+INT(11*300*(1+3)*3*INT(2)*L/4)+INT(DL/2+4)
360 DD=DD-INT(DL/2)*2
370 REM XXX IZRANJEN STEVILA PRANJENIH DNI U TEKOCEN LETU XXX
380 TD=TD+(D(1)(D(1),D(1)))
390 IF I=0 THEN 430
400 IF (DL=365)(D(1)=1)+(DL=366)(D(1)=2) THEN TD=TD+1
410 IF D(1)=2 THEN TD=TD-1
420 REM XXX IZRANJEN STEVILA SOBOTI IN NEDELJI U TEKOCEN LETU XXX
430 XI=DL-D(1)=0:TD=INT(XI/7)+SD(XI)-INT(XI/7)*2
430 IF D(1)=0 THEN XI=XI-1:GOTO 440
435 XI=DL
440 TD=TD+INT(XI/7)+SD(XI)-INT(XI/7)*2
445 NEXT I
450 I
470 REM XXX HANJARA NAJVICESTEVILJE STEVILA DELA PROSTIH DNI XXX
480 IF TD=0 THEN TD=TD+LL
500 NEXT LL
510 I
520 I
530 REM XXX REPIS XXX
540 PRINT"P PRVO LETO=";P(1):STEVILA DNI=";PD
550 END
570 I
580 REM XXX DATUMI PRANJENIH DNI XXX
590 DATA 4,7,11,1,1,5,20,11
600 REM XXX STEVILA PRANJENIH DNI (BROJ SOBOTI IN NEDELJI, CE PRADE PRANIK NA)
610 REM NED PON TOR SRE CET PET SOB
620 DATA 1,2,1,3,1,3,1,2,1,1,6,11 REM ENDEJANI+DUODENJANI
PRVO LETO= 2812 STEVILA DNI= 114

```

```

/*****
* RACUNAR.C : Nerađni dani. *
*
* Franjo Gospodnetić. *
* Vinkovci 9-8. *
* Zagreb. *
* 05/01/1995 *
*
* 80 SOFTWARE (Compiler, v1.5. *
*****

#define GLS 64h
#define SINDOF 2
#define (IS_LEAP_YEAR(year) year % 4 == 0 && year % 100 != 0 || year % 400 == 0)

main()
{
    int year, New_Year;

    putchar (GLS);
    New_Year = 1986;

    for (year = 1987; year < 2385; year++)
        if ((IS_LEAP_YEAR(year)) && New_Year == SINDOF)
            printf ("Mda", year);

        if ((IS_LEAP_YEAR(year))
            New_Year = 2;
        else
            New_Year++;
        if (New_Year > 60)
            New_Year = 7;
}

```

2232, 2260, 2288, 2328, 2356 i 2384. Do (na)jrećnije Nove godine će, dakle, proteći još čitavih 26 leta.

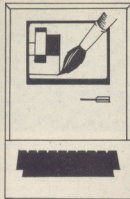
Novi nagradni zadatak se zasniva na igri „Reverse“ koja je u svoje vreme bila čest gost memorije džepnih programabilnih kalkulatora. Dat vam je, naime, tekst 'I_U_VARNICARI' koji treba transformisati u 'RACUNARI_LVI' (i donja crta koju smo koristili u obe reči je karakter ravnoprasvan svakom slovu) primenom što manjeg broja dozvoljenih transformacija. Dozvoljena je jedino transformacija kojom menjate redosled prvih nekoliko slova date reči dok se njen „rep“ ne menja. Ako, na primer, promenite redosled prvih pet slova u početnoj reči 'I-U-VARNICARI', dobićete 'V-U_IARNICARI' a ako zatim u toj reči promenite redosled prvih 8 slova, dobićete 'NRAL-U_VICARI'. Korak po korak, dobićete tekst 'RACUNARI_LVI'. Zatim na list papira ispišite sve korake, popunite kupon koji ovde dajemo i sve to pre 20. marta pošaljite na adresu „Računar!“ — za Dejanove pitalice, Bulevar vojvode Mišića 17, Beograd. Nagrade od 10.000, 5.000 i 3.000 dinara čenaju rešavače koji do rešenja dođu u minimalnom broju poteza; ukoliko ih bude više, odlučiće žreb.

Tekst 'I_U_VARNICARI' se moza transformisati u 'RACUNARI_LVI' primenom najmanje _____ legalnih transformacija.

Ime i prezime _____

Adresa _____

Mesto (_____) _____



Umetnost programiranja

Holandska zastava

Čuveni Dijkstra je 1976. godine postavio zadatak koji je nazvao „problem bojenja holandske zastave“. Iako deluje krajnje jednostavno i lako razumljivo, zadatak još do danas nije dobio najelegantnije rešenje. U čemu je to problem sa holandskom zastavom i kako ga je dva puta rešio sam Dijkstra, a zatim i, nešto bolje, Mejer.

Neka je dat niz od N elemenata od kojih je svaki obojen u jednu od tri boje — crvenu, belu ili plavu. Problem je kako pregrupisati elemente niza tako da svi crveni elementi budu na početku, beli u sredini, a plavi na kraju. Sada je, verovatno, svima jasno zašto se zadatak zove „problem holandske zastave“.

Primitivne operacije

Pri pravljenju zastave Dijkstra dozvoljava korišćenje dve primitivne operacije:

- buck(i) — funkcija koja očitava svu i-tog elementa niza;
- swap(i,j) — procedura koja menja i-ti i j-ti element niza, tj. i-ti premešta na mesto j-tog, a j-ti na mesto i-tog.

Dodatni uslovi za program su da:

- mora da funkcioniše za sve moguće početne raspodele elemenata;
- primena buck operacije dvaput na isti element nije dozvoljena.

Dijkstra je, takođe, naglasio da od programa približne složenosti više ceni onaj sa manjim brojem zamera (swap). Sva rešenja koja ćemo prikazati razlikuju četiri klase elemenata niza — crvene, bele, plave i elemente čiju boju još nismo pročitali. Elementi su grupisani u četiri regije, a informacije o granicama između tri oblasti čuvaju se u tri promenljive r , w i b kao na slici:

1	N
crveni neispitani	beli plavi
r	w b
odnosno za $1 \leq k \leq r$: $r \leq k \leq w$:	k-ti element je crven (red), k-ti element nije ispitan, tj. još ne znamo koje je boje, k-ti element je bec (white), $b \leq k \leq N$: k-ti element je plav (blue).

21/holandska zastava

Tri pristupa

PRVI PRISTUP je da ispitujemo po jedan element iz klase neispitanih i smeštamo ga u odgovarajuću klasu. Pri tome, naravno, ažuriramo granice oblasti. Paskal procedura ovog sistema algoritma data je na slici 1.

DRUGI PRISTUP je poboljšanje prethodnog koje je dao autor problema. Primitimo da prethodni algoritam pravi N zamera i u slučaju kada su inicijalno svi elementi niza crveni, što je vrlo neracionalno. Nova ideja je da

Neispitanu oblast možemo skanirati sleva nadesno (koristeći promenljivu r). Kad naiđemo na element koji nije crven, počinjemo skaniranje neispitane oblasti sdesna ulivo (koristeći promenljivu w). Pri tome preskačemo sve bele elemente sve dok ne naiđemo na neki koji nije bec. Ako je taj element plav, zamenjujemo ga elementom na koji ukazuje b , i nastavljamo skaniranje. Kada promenljivom w naiđemo na crveni element, zamenjujemo ga elementom na koji ukazuje r (dakle nije crven). Osim toga, ako

```

PROCEDURE holandskazastava31
VAR r,w,b:integer;
    colr,colw,color;
BEGIN
    r:=1; w:=N; b:=N;
    WHILE w>r DO
        BEGIN
            colr:=buck(r);
            WHILE (colr = red) AND (w>r) DO
                BEGIN
                    r:=r+1;
                    colr:=buck(r);
                END;
            IF r<w THEN
                BEGIN
                    colw:=buck(w);
                    WHILE (colw < red) AND (w>r) DO
                        BEGIN
                            IF colw = blue
                                THEN
                                    BEGIN
                                        swap(w,b);
                                        b:=b-1;
                                        END;
                                        w:=w-1;
                                        colw:=buck(w);
                                    END;
                            IF colw = blue
                                THEN
                                    BEGIN
                                        swap(w,b);
                                        b:=b-1;
                                        END;
                                        IF (colw = red) OR (colr < white)
                                            THEN
                                                swap(r,w);
                                                END;
                                                IF colr = blue
                                                    THEN
                                                        BEGIN
                                                            swap(w,b);
                                                            b:=b-1;
                                                            END;
                                                            r:=r+1;
                                                            w:=w-1;
                                                        END;
            END;
        END;
    END;
    holandskazastava31;
PROCEDURE holandskazastava21
VAR r,w,b:integer;
    colr,colw,color;
BEGIN
    r:=1; w:=N; b:=N;
    WHILE w>r DO
        BEGIN
            colr:=buck(r);
            WHILE (colr = red) AND (w>r) DO
                BEGIN
                    r:=r+1;
                    colr:=buck(r);
                END;
            IF r<w THEN
                BEGIN
                    colw:=buck(w);
                    WHILE (colw = white) AND (w>r) DO
                        BEGIN
                            w:=w-1;
                            colw:=buck(w);
                            END;
                            CASE colw OF
                                red : BEGIN
                                    swap(r,w);
                                    r:=r+1;
                                    END;
                                    white : w:=w-1;
                                    blue : BEGIN
                                        swap(w,b);
                                        w:=w-1;
                                        swap(r,w);
                                        b:=b-1;
                                        END;
                            END;
                        END;
                    END;
                    IF r<w
                        THEN
                            BEGIN
                                swap(w,b);
                                b:=b-1;
                                END;
                                w:=w-1;
                            END;
                    END;
                END;
            END;
        END;
    END;
    holandskazastava21;

```

(i) granicu r crvene oblasti pomeramo desno što je više moguće bez zamera;

(ii) slično, granicu w bele oblasti pomeramo ulivo što je više moguće bez zamera;

(iii) možda bi bilo korisno da istovremeno ispitujemo po dva elementa niza umesto jednog (granicu crvene i neispitane oblasti i granicu neispitane i bele oblasti).

Ako ovo implementiramo, element na koji ukazuje r pointer (iii, kako bi to rekli naši profesori, polkazivač) biće ili bec ili plav, a element na koji ukazuje w ili crven ili plav. Sve dok je $r \leq w$ postoje četiri klase. Paskal procedura ovog algoritma data je na slici 2.

Majerovo rešenje prevazišlo nedostatak prethodna dva. Naime, u njima postoje situacije u kojima premeštamo element čiju boju ne znamo, što je neracionalno jer postoje šanse da je on već u dobroj klasi.

je taj bivši r plav, zamenjujemo ga elementom na koji pokazuje b . Paskal procedura ovog algoritma data je na slici 3.

Još bolje

Ovaj algoritam se može i dalje poboljšavati. Umesto skaniranja ulivo, što je moguće dalje bez zamera, granicu pomeramo ulivo dok se ne pojavi crveni element i pri tom preskačemo bele, a zamenjujemo plave elemente. Ili, sva tri pojavljivanja „swap(w,b)“ zamenimo sa „if w,b then swap(b,w)“. Zanimljiva je i ideja da se istovremeno posmatraju više od dva elementa istovremeno. Poboljšanja, međutim, treba da budu i korisna — komplikacije u programu moraju da imaju svoje praktično opravdanje! Pokušajte i vi sa nekim poboljšanjima. Još uvek nije rečena poslednja reč.

Zoran Obradović

CP/M je univerzalni operativni sistem implementiran u *raznim verzijama* na mnogim ličnim računarima različitih proizvođača. Izraz u *raznim verzijama* verovatno u vama budi opravdanu sumnju: u kojoj su meri te razne verzije međusobno kompatibilne? Odgovor na ovakvo pitanje nas vodi u rane sedamdesete godine, doba velikog entuzijazma u razvoju kućnih računara. Gary Kildel (Gary Kildall), profesor mornaričkih postdiplomskih studija u Kaliforniji i Intelov softverski konsultant, tih je dana do na otpadu nabavljenih komponentata i Intelovih čipova „van specifikacija“ sastavljao svoj prvi sasvim lični računar koji je, naravno, trebalo opremiti nekakvim softverom. Pošto je bilo skoro nemoguće asemblirati loše složeniji operativni sistem „na ruke“, Gary je posle radnog vremena krišom koristio Intelov razvojni sistem da prevede najnoviju verziju svoga programa sa PL/M-a na mašinar i ugradi ga u eprom koji bi iste večeri isprobavao kod kuće. I pored ovako teških uslova, operativni sistem Garyjevog kompjutera je polako napredovao ka konačnoj verziji koja je nazvana CP/M 1.3 (CP/M je, kao što svakako znate, skraćeni- ca od Control Program/Monitor).

Zakrpa na zakrpu

Gary Kildel je, sasvim prirodno, ponudio svoje remek-delo Intelu koji je, verujući da su kućni računari interesantni samo za malo i sumnjivo tržište amatera, odbio da otupi CP/M i tako napravio jednu od dve najveće gluposti u istoriji svoje inače veoma uspešne egzistencije (druga glupost je što nije našao zajednički jezik sa konstruktorima 8080 koji su se zbog toga odvojili i osnovali Zilog). Garija ovo odbijanje nije previše obeshabralo pa je, zajedno sa ženom Doroti koja se bavila marketingom, osnovao danas čuvenu firmu Digital Research. Početni kapital skupljen od akcionara je bio dovoljan za kupovinu razvojnog sistema i opreme za programiranje eproma (tada veoma skupe), pa je Gary bez mnogo problema ispravio neke bagove CP/M-a 1.3 i tako dobio verziju 1.4, koja je i danas u primeni na nekim kompjuterima. Čak je i početna verzija 1.3, prodata firmi Cromemco, danas u životu pod imenom Cromemco Disc Operating System — CDOS.

Proširivanjem memorije kućnih računara krajem sedamdesetih godina, CP/M 1.4 je bio već zastareli. Ovaj program je, naime, koncipiran tako da zauzima što manje mesta u memoriji, pa mu je bila neophodna česta komunikacija sa diskom koja je odnosila previše vremena. S druge strane, kapaciteti disketa su bili prilično mali, pa se sa dva drajva neki poslovi jednostavno nisu mogli obaviti. CP/M 2.2 je, jasno, lako

prevazišao prvi problem: bitan deo programa je premešten u memoriju. Da bi se prevazišao problem nedovoljnog kapaciteta flopi diskova, Digital je došao na ideju da softverski podrži čak 16 disk jedinica koje bi bile povezane sa centralnim procesorom. U doba današnjeg rapidnog pada cena masivnih (hard) diskova, ovakva solucija može da izgleda veoma smešno: zamislite vaš sto pretrpan drajvovima i sebe kao žonglirate sa disketama! Pa ipak, pre nekoliko godina se samo na ovaj način mogla skladištiti dovoljna količina informacija za ozbiljnije poslovne primene. Fleksibilne izmene CP/M-a su učinile da ovaj operativni sistem krajem 1979. godine koristi preko 1,5 milion vlasnika računara!

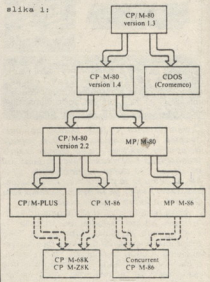
Dalji napredak hardvera je omogućio ozbiljno razmišljanje o personalnim računarima koji bi istovremeno obrađivali više programa ili, u složenijem slučaju, opsluživali više korisnika. Tako je nastao MP/M (Multi Programming Monitor control program), operativni sistem koji nikada neće postići naročitu popularnost. Praktično istovremeno, IBM je odlučio da svoj danas neverovatno popularni računar „PC“, zasnovan na Intelovom mikroprocesoru 8080, opremi operativnim sistemom koji liči na CP/M. Iako je Digital planirao da pripremi operativni sistem koji bi nazvao CP/M 86, firma Seattle Computer Products ga je preduhitrila napisavši operativni sistem koji je najpre nazvan 86-DOS a onda, malo zvučnije, QDOS. IBM je svoj PC opremio MS DOS-om, hibridom Microsoftovog disk operativnog sistema i 86-DOS-a. MS DOS je, na žalost, nekompatibilan sa CP/M-om: lako su naredbe slične, format informacija na disketama je bitno promenjen, što znači da je prelazak sa nekog CP/M računara na IBM PC prilično bolan.

Digital je za to vreme pripremao CP/M 3.0 koji je u poslednjem trenutku (po cenu preštapavanja čitave dokumentacije), preimenovan u CP/M PLUS, jer su stručnjaci za marketing smatrali da se na taj način snažnije podvici unapređenje tada već izuzetno popularnog operativnog sistema. Digital se, osim toga, neprekidno trud da prilagodi CP/M svim novim mikroprocesorima koji se pojavljuju na tržištu, pa je tako nastao CP/M-88K (za Motorolu 68000) i CP/M-28K (za Zilog Z8000). Implementacija je na moćnijim procesorima su, međutim, iznislila na videlo neke loše osobine CP/M-a, pa je ovaj operativni sistem morao da bude dopunjavao dodatnim sistemskim softverom, kao što je GEM na „atariju 520 ST“. Na slici je prikazan stabilniji razvojni operativnog sistema o kojem govorimo.

Metuzalem sa pedigreom

O dobrim stranama CP/M-a se mnogo piše: radi se o standardnom operativnom sistemu sa ogromnom softverskom podrškom, opsežnom dokumentacijom i mnoštvom knjiga za početnike. CP/M je, osim

slika 1:



toga, otvoren operativni sistem koji se, uz malo veštine u pisanju mašinskih programa, može prilagodavati potrebama korisnika. Hardverski zahtevi CP/M-a su prilično skromni, što znači da se ovaj operativni sistem može instalirati uz umerena novčana ulaganja: računar sa Z80, 64 Kb RAM-a, dve disk jedinice i monitorom danas može da se nabavi za manje od 600 funti, što uopšte nije mnogo za poslovne primene, čak ni u našim uslovima.

Sve loše osobine CP/M-a se mogu iskazati jednom rečju: zastareo. CP/M se, naime, razvija već petnaestak godina; za to vreme je bezbroj puta krpljen da bi udovoljio zahtevima novih generacija hardvera. Ukoliko, na primer, aktivirate neki CP/M kompjuter i otkucate DIT, a zatim, primitivši da ste pogrešili, pritisnete DELETE, ne možete se previše iznenaditi ako se na ekranu pojavi DITT. Početne verzije CP/M-a su, naime, predvidele rad sa konzolnim terminalom (teletiprinterom), koji, jednostavno, ne može da vrati glavu za jedno mesto unazad; brisanje simbola je signalizirano njegovim ponavljanjem! Za video terminale koji su danas u isključivoj upotrebi ovaj dak rad predstavlja običnu besmislicu, ali se ipak održao u nekim implementacijama.

Zastarelost CP/M-a se daleko više primećuje u radu sa grafikom i zvukom. Za tvorce CP/M-a je bilo nezamislivo da se kompjuterska grafika na malim računarima može koristiti i za šta drugo osim za igre. Igre su, mislilo se, prilično nepotrebne jednom poslovnom operativnom sistemu. Zaključak: CP/M nema grafiku. Nikakvu!

Pokazalo se, međutim, da se igre mogu pisati i sa rezolucijom 80*24 ili 80*32, a tim

Interesovanje koje je izazvao naš napis „Prvi gem za Tripos“ u prošlim „Računarima“ nas je podstakao da započemo seriju tekstova o popularnim operativnim sistemima ličnih kompjutera. Teško bi bilo pronaći ljubitelja računara koji, kada se pomene operativni sistem, neće pomisliti na CP/M, program o kome je napisano mnogo knjiga i tekstova (čak i na našem jeziku!). I pored toga, verujemo da većini naših čitalaca nedostaje potpuna i relevantna informacija iz koje će naučiti kako se razgovara sa kompjuterom na jeziku koji je firma Digital Research nazvala CP/M.

što će svemirci biti zamenjeni zvezdicama, „pozitivni“ uzvičnikom, a ograda koja ih štiti iksevima. Ne izgleda baš mnogo veselo, pogotovo zato što nema zvuka, ali je sasvim upotrebljivo! Bilo kakvi programi koji koriste grafiku u poslovne svrhe (dijagrami, histogrami, „torta dijagrami“ i slični indikatori), su, međutim, u startu neostvarljivi! Mnogi ozbiljni vlasnici računara bi se danas rado bavili kompjuterskim simulacijama, animacijom i sličnim područjima, ali je za njih potrebno tražiti drugi operativni sistem. Umesto celovitog rešenja, tvorc raznih CP/M kompjutera su pripremali uslužne mašinske programe za rad sa grafikom koji su bili prilično teški za upotrebu, nefleksibilni, slabo dokumentovani i, što je najgore, nestandardizovani. GEM, grafički operativni sistem koji smo opisali u prošlim „Računarima“, predstavlja ozbiljno proširenje CP/M-a (i MS DOS-a) koja tek treba da se pokaže na tržištu. U međuvremenu, CP/M je operativni sistem upotrebljiv jedino za rad sa slovima.

Međusobna kompatibilnost raznih CP/M računara je takođe prilično sumnjiva. Minimalan uslov za kompatibilnost je format diskova — ako ste na poslu snimili neke podatke na disketu, *neophodno* vam je da možete da ih pročitate i na svom kućnom računaru. CP/M je, međutim, često instaliran kao drugi operativni sistem: naprimer, radio pod Apple DOS-om pa se, dokupljivanjem jednog Z80, pretvarao u CP/M mašinu. Jasno je da se uz Z80 obično dobijao minimum hardvera, pa se o komunikaciji sa diskovima i dalje brinuo 6502; format disketa je i dalje bio Appleov!

Kompatibilnost bez pokriva

Kada se problem kompatibilnosti disketa nekako reši, poželećemo i kompatibilnost softvera. Jasno je da programi za dva kompjutera koji koriste razne mikroprocepsore ne mogu ni slučajno da budu kompatibilni: ni jedan od čuvenih tekst procesora koji savršeno rade na „partneru“ neće moći da se izvršava na „atariju 520 ST“! Moguće je da će u budućnosti, zahvaljujući ogromnoj razlici u brzini rada, biti napisan softverski emulator pomoću koga će Motorola 68000 savršeno imitirati Z80, pa će dobar deo softvera biti direktno upotrebljiv. Moguće je, isto tako, da se softverske firme ljuto usprotive proizvodnji ovakvog emulatora: CP/M programi su široko rasprostranjeni (=mnogo se piratuju), pa se proizvođačima softvera daleko više isplati da svoja remek-dela sami prodaju u više sličnih varijanti. Ne sme vas, čak, začuditi ni ako program sa „partnera“ ne radi na drugim CP/M mašinama za Z80: autorima programa se više isplati da posebno prilagodavaju

softver svakom računaru i tako umnožavaju svoje prihode!

CP/M-u se, najzad, zamera da je slabo opremljen naredbama za rad sa periferijom i da su ove naredbe nedovoljno standardizovane. Ovakvim zamerama, po našem mišljenju, nema mnogo mesta: naredbe za rad sa diskom i štampačem su integralni deo CP/M-a, dok ploterima, optičkim perima i drugim hardverom i onako uglavnom upravljaju aplikacioni programi. U ovom trenutku je, ustaoim, specijalne grafičke opreme prilično malo, dok će u danima kada ona postane dovoljno jeftina da bi bila masovna CP/M verovatno „otiči“ u penziju, ili biti dopunjen nekim specijalizovanim operativnim sistemom kao što je GEM.

Govorili smo, dakle, o nastanku CP/M-a, njegovom razvoju, dobrim i lošim stranama. Došlo je vreme da predemo na konkretnije stvari i opišemo standardne naredbe CP/M-a pri čemu ćemo uglavnom imati u vidu CP/M 2.2 i CP/M PLUS dok će MP/M i druge varijante ovoga operativnog sistema biti pominjane samo u posebnim slučajevima.

Osnovne naredbe

Pretpostavimo da su na vaš računari koji radi pod CP/M-om priključene dve dvostrane (ili, što je za funkcionalnog aspekta isto), četiri jednostrane disk jedinice označene slovima A, B, C i D. U svakom drajvu se, jasno, nalazi po jedna disketa spremna za rad. CP/M, dok mu drukčije ne naredite, smatra da će se sve naredbe za rad sa diskom odnositi na drajv A pa ispisuje prompt "A>". znak da očekuje instrukciju. Prva naredba koju ćete otkucati će, vero-

vatno, biti DIR, skraćena od *directory*: zahtevate od računara da prikaže imena svih programa koji su upisani na sistemsku disketu. Disketa će se zavrteti i na ekranu pojaviti nešto poput slike 2.

Na disketi se, dakle, nalazi osam programa. Prvi se zove STAT.COM, drugi MBASIC4.COM, treći BACKUP.COM i tako dalje, sve do poslednjeg fajla sa čudnim imenom PIP.COM. Vidimo da se kompletan naziv programa sastoji od imena (npr. STAT, MBASIC4, PIP i slično) koje može sadržati najviše osam slova i oznake njegovog tipa (u našem slučaju su svi programi tipa COM) koja se čitaju od najviše tri slova. Dok je svakom čitaocu ovih redova jasno da program mora imati unikatno ime, navođenje oznake tipa može da nam izgleda kao nepotrebno usložnjavanje. Pokazalo se, međutim, da korisnici CP/M-a često pišu programe na višim jezicima koji se kompajliraju, na primer paskalu ili fortranu. Kada naredite računaru da prevede program PROBA.FOR na mašinski jezik, objektni kod će morati da bude smešten u neku datoteku koju bi bilo logično nazvati PROBA.OBJ (mogli bismo je, naravno, nazvati i PROBA1 ili PROBAOBJ, ali bi se na taj način znatno gubilo na preglednosti i povećavala verovatnoća nastanka zabune). Zbog toga su standardima CP/M-a predloženi osnovni tipovi programa koji su popisani na slici 3. Nema, naravno, prepreke da i sami kreirate datoteke nekog posebnog tipa (na primer IZVESTAJ.JUN ili REŠENJA.JAN), pri čemu morate da se pridržavate zlatnih pravila: u okviru oznake tipa programa smeju da se pišu **uključivo** slova i cifre, dok se u okviru imena programa mogu

slika 2:

```
A>DIR
A:STAT      COM      :MBASIC4  COM      :BACKUP    COM      :MBASIC5    COM
A:CBAS2     COM      :CRUN2    COM      :XREF      COM      :PIP        COM
A>
```

slika 3:

```
.ASM      Izvorni (source) program na assembleru.
.BAK      Backup neke datoteke.
.BAS      Izvorni program na bezziku.
.COM      Objektni relokativni program (CP/M 86).
.DAT      Izvršni program.
.DOC      Podaci.
.DOC      Datoteka sa tekстом - dokumentacijom.
.HEX      Program u Intelov HEX formatu.
.LNT      Delimično prevedeni bezzik program (CBASIC).
.LIB      Biblioteka potprograma.
.MAC      Izvorni program sa makro naredbama.
.PRL      Program koji se može izvršiti na bilo kojoj strani memorije (MP/M).
.PRN      Izveštaj koji assembler priprema za štampanje.
.SUB      Procedura koja se prosledjuje SUBMIT programu.
.TXT      Datoteka sa tekстом.
.***     Prilivena datoteka koju treba obrisati na kraju izvršene.
```

koristi! i neki specijalni znaci **all** ne upitnik, zvezdica, zagrade (uglaste, obične i znaci veće i manje), tačke, zarez, kosa crta, znak jeonakosti i dvotacka. U tom su smislu korektna imena fajlova DAN-PRVI.COM, CPM-RAC, LUJ.XIV, GALAKS.1, 8271.101 i OHI all ne i CPM-RUN, CPM-A-B, GALAKSLJA.1 ili JE) FAJL.BAS.

Direktorijum na dlanu

Komanda DIR će na ekranu prikazati imena i tipove svih datoteka **osim** programa tipa SYS; da biste dobili spisak ovih programa, moraćete da otkucate DIRSYS (ili DIR* SYS ukoliko ne koristite CP/M PLUS). Čemu ovo skrivanje? Pokazalo se da neiskusne korisnike koji bi želeli da se služe računarnom ne puštajući se u programiranje ili operativni sistem može da zbuni pojava mnoštva programa na disketi pa je ostavljeno mogućnost da se neki od njih „sakriju“ dok će se u direktorijumu nalaziti samo programi koje ima smisla samostalno startovati.

Ako, umesto DIR, napišete DIR PROBA.BAS, računac će prikazati samo programe koji se zovu PROBA.BAS. Na disketi, jasno, može da postoji samo jedan takav program, pa će se ova opcija prilično retko koristiti. Sa druge strane, često ćemo u imenu koje navodimo iz DIR koristiti džoker znake: upitnik i zvezdicu. Upitnik, kao i obično, zamenjuje bilo koji znak, a zvezdica bilo koju grupu znakova pa će tako DIR* BAS ispisati naslove svih bezijk programa, DIR PROBA* * pronaći programe svih tipova koji se zovu PROBA dok će DIR ??AB??*COM dati imena programa koji se, na primer, zovu TABLA.COM, STABLO.COM i DIABLO.COM. Ukoliko poželimo da proverimo postojanje nekog programa na drugim CD-ovima, koristećemo DIR B*.*. DIR C:PROBA* * i tome slično.

Pošto ste isprobali naredbu DIR, pokušajte da otkucate DIR FULL (samo ako koristite CP/M PLUS) i videćete nešto poput slike 4. Uz imena programa se, primećujemo, na disketi čuvaju i neki drugi podaci, kao što je veličina u slovima i bajtovima, podatak o zaštiti programa i, kod nekih verzija, datum kreiranja. Sa DIR ... [EXCLUDE] možemo da dobijemo imena svih datoteka **osim** onih koji odgovaraju nekoj specifikaciji, sa DIR ... [RO] da prikazemo imena svih programa koji su otvoreno isključivo za čitanje i tako dalje.

Pre nego što nastavimo sa opisivanjem naredbi CP/M-a, treba da razmotrimo pitanje zaštite programa koju smo već dva puta pomenuli. CP/M se u globalu ne bavi zaštitom podataka pošto je namenjen prevažno računarnima koje će koristiti jedna jedina osoba. Pa ipak, ponekad će ta osoba poželeti da zaštiti neke programe od samoga sebe, to jest da se zaštiti od grešaka koje mogu da dovedu do gubitka vrednih podataka. Otkucajte, dakle, SET A*.*COM [RO SYS]. Na taj će način sve programe tipa COM proglasiti zatvorenim za upis (bilo kakav pokušaj da se neki od njih menja ili briše će izazvati poruku o grešci) i uzgred zatraži od računara da njihove naslove ne prikazuje kada korisnik otkuca DIR. Ako dodnije poželite da promenite neki od ovih programa, možete da otkucate SET A*.* [RW DIR] (RO dolazi od *read only*, a RW od

slika 4:

A>DIR [FULL]

Scanning Directory...

Sorting Directory...

Directory for Drive A: User 0

Name	Bytes	Recs	Attributes	Name	Bytes	Recs	Attributes	
CPM SYS	18k	138	Svs RW	DATE COM	3k	22	Dir RW	
DEVICE COM	8k	56	Dir RW	DIR COM	15k	114	Dir RW	
ED COM	10k	73	Dir RW	ERASE COM	4k	20	Dir RW	
HELP COM	7k	56	Dir RW	HELP HLP	62k	489	Dir RW	
PIP COM	9k	68	Dir RW	RENAME COM	3k	23	Dir RW	
SET COM	11k	81	Dir RW	SETDEF COM	4k	32	Dir RW	
SHOW COM	9k	66	Dir RW	SUBMIT COM	6k	42	Dir RW	
TYPE COM	3k	23	Dir RW					
Total Bytes = 172k				Total Records = 1315				Files Found = 15
Total 1k Blocks = 172				Used/Max Dir Entries For Drive A: 19/ 64				

read/write što podrazumeva i pravo na brisanje). Ukoliko koristite CP/M PLUS, možete da uradite i nešto mnogo bolje: naredba poput SET *.* [PASSWORD=RAČUNARI PROTECT=DELETE] će onemogućiti bilo kom korisniku da obriše programe sa diskete **osim** u slučaju da zna lozinku 'RAČUNARI!' Premda dobro znamo da naši čitaoci *w/o* rado slušaju o zaštiti programa, ograničen prostor u ovom prikazu nam ne dozvoljava da se njom ozbiljnije bavimo. Na ovu ćemo se temu svakako vratiti u nekom od naših budućih napisa.

Brisanje sa zadrškom

Pošto smo prilično daleko upoznali naredbu DIR, upoznajmo idaleko opasniju naredbu ERA odnosno ERASE (na CP/M PLUS). Naredba ERA, kao što pretpostavljate, omogućava brisanje nepotrebnih datoteka sa diskete. Zašto kažemo da je ova naredba opasna? Pošto je u nehotične ERASE *.* obrisalo sve programe sa više disketa i tako eventualno izazvalo ogromnu štetu. Zato je u CP/M i, posebno, u CP/M PLUS ugrađeno mnogo sigurnosnih brava: ako računac proceni da brišete previše programa (otkucali ste, na primer, pomenutu ERA *.*), na ekranu će se pojaviti poruka poput ALL FILLES (Y/N)? ili ARE YOU SURE (Y/N)? Pritisnite liše "Y" samo ako ste sigurni da znate šta radite! Čini nam se da je daleko bolje koristiti naredbu ERAQ *Erase With Query* koja, pre nego što obriše datoteku, ispisuje njeno puno ime i zahteva od korisnika da pritisne slovo "Y" da bi odobrio njeno brisanje. Opšti je savet da naredba ERA koristite isključivo ako se u imenu iza nje **ne nalazi** ni jedan džoker znak!

Za preimenovanje postojećih datoteka se koristi naredba REN (odnosno RENAME na CP/M PLUS) koju ćemo najbolje razumeti na primeru. REN RAČUNAR.COM = COMPUTER.COM će datoteci čije je ime bilo COMPUTER.COM dodeliti ime RAČUNAR.COM. Važno je zapamtiti da naredba RENAME ne kopira fajl: menja se jedino njegovo ime u direktorijumu. Zbog toga bi naredba RENAME A:PROG.PRO = B:PROG.COM izazvala poruku o grešci jer smo pokušali da premostimo PROG.COM sa drvaja B na drvaj A što se nikako ne bi moglo izvršiti bez kopiranja sadržaja. CP/M će izdati poruku FILE EXISTS čak i ako napišete RENAME A=B pri čemu datoteka sa imenom A već postoji — kada bi neka grupa datoteka zauzela njeno mesto, ope-

rativni sistem ne bi znao šta da radi sa „pokrivenom“ verzijom!

Korisnici CP/M-a, iako mnogo rade sa fajlovima, retko upotrebljavaju naredbu SAVE koja omogućava snimanje podataka na disk. Naredba SAVE je, naime, prilično ekscentrična: umesto da počne sa snimanjem čim pritisnemo RETURN, računac će sačekati da pritisnemo CTRL C ili da se okonča izvršavanje tekućeg programa, a onda će zahtevati ime (i tip) datoteke kao i adrese početka i kraja memorijske zone koja se snima (sve adrese se kucaju heksadekadno). Zbog svega toga se snimanje podataka na disk i njihovo čuvavanje obično poverava sistemskim programima kao što je bezijk interpreter dok se SAVE komanda CP/M-a koristi samo kada treba na brzinu snimiti čitavu memoriju da bi se dodnije posao mogao nastaviti na mestu prekidu.

Multiprogramski rad

Naredba TYPE je na neki način suprotna od SAVE: TYPE PROBA.BAS će na ekranu prikazati sadržaj datoteke PROBA.BAS koja se, pretpostavljamo, sastoji od ASCII karaktera. Pošto je broj linija na ekranu ograničen, sadržaj jeće datoteke bi bilo teško propratiti. Zato se skrolovanje teksta zaustavlja pošto budu ispisane 24 linije, dok za nastavak treba pritisnuti CTRL Q (ili, kod ranijih verzija CP/M-a, bilo koji taster). Ukoliko vaš ekran ima više od 24 linije, možete da otkucate TYPE PROBA.BAS P32 i tako saopštite računaru da se zaustavlja tek pošto ispiše 32 linije. Ako vam ova konvencija ne odgovara, otkucajte TYPE PROBA.BAS NO PAGE pa će ste sami moći da zaustavljate skrolovanje sa CTRL S i da ga nastavljate sa CTRL Q.

Pomenimo i komandu USER koja, premda je uključena u standard CP/M-a, često ostaje neimplementirana. CP/M je, rekli smo, namenjen radu jednog korisnika, ali to ne znači da taj korisnik ne može da radi više stvari. Ukoliko na iste diskete snimate programe koje razvijate, tekstove koje pišete i baza podataka kojima povremeno pristupate, ubrzo ćete se naći u velikoj konfuziji. Zato će, pre nego što počnete sa programiranjem, otkucati USER 1, pre nego što startujete tekst procesor USER 2, a pre nego što započnete rad sa bazom podataka — USER 3. Svaki od takvih fiktivnih korisnika se neće mešati u poslove drugih, pa će vam tako komanda DIR davati samo podatke o datotekama koje ste kreirali dok ste imali odgovarajući korisnički broj. Nema

slika 5: KONTROLNI KARAKTERI:

<CR>	Završava komandnu liniju.
<LF>	Prelazi u novi red.
CTRL J	Kao <LF>.
CTRL M	Kao <CR>.
CTRL U	Ponistava komandnu liniju.
CTRL X	Kao CTRL U.
<BS>	Briše zadnji karakter.
	Briše zadnji karakter i daje eho.
<RUB>	Kao .
CTRL H	Kao <BS>.
CTRL K	Briše tekst do kraja linije.
CTRL E	Nastavak linije u drugom redu.
CTRL R	Ponavlja vazeci deo linije.
CTRL S	Zaustavlja ispisivanje teksta.
CTRL Q	Nastavlja ispisivanje teksta.
CTRL P	Uključuje i isključuje printer.
CTRL A	Kurzor desno.
CTRL B	Kurzor na početak (kraj) linije.
CTRL F	Ubaci SPACE na poziciju kurzora.
CTRL G	Kurzor levo.
CTRL C	Warm start CP/M-a

čak nikakve prepreke da i kao USER 1 i kao USER 3 kreirate datoteku koja će se zvati PISMO.TXT: niti će nastupiti neka zabuna niti će jedan „korisnik“ moći da promeni ili izbriše datoteku drugoga. Broj korisnika, kako kod CP/M-a tako i kod CP/M PLUS, može da se nalazi između 0 i 15 pri čemu svi „korisnici“ mogu da pristupaju datotekama korisnika 0. U korisničko područje 0 se, dakle, smestaju sistemski programi i zajedničke datoteke.

U osnovne naredbe CP/M-a se ubrajaju i velika slova A, B, C itd. iz kojih se stavlja dvotočka. Napisavši B, na primer, stavljamo računaru do znanja da pri svim budućim SAVE, ERASE i sličnim naredbama treba da pristupa drajvu B, a ne početnom drajvu A. Posle ove naredbe osnovni prompt će biti promenjen u „B>“.

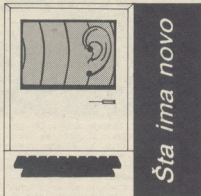
Kontrolni karakteri

Već smo rekli da se listanje neke datoteke može privremeno zaustaviti sa CTRL S i da se nastavak tog listanja postiže sa CTRL Q. S i i Q, naravno, nisu jedini kontrolni karakteri koje CP/M koristi: na slici 5 je prikazan sumarni pregled slova koja možete pritisnuti zajedno sa CTRL da biste pokrenuli ekranski editor, obrisali deo lini-

je, startovali štampač ili izvršili neku drugu specijalnu funkciju. Uz sav rizik da se većina čitalaca nasmee, reći ćemo da **nije potrebno** cijati tastere CTRL i A kako bismo ih pritisnuli istovremeno. Da otkucate CTRL A pritisnite dirku CTRL i, držeći je pritisnutu, pritisnite li otpustite taster A. Tek posle toga možete da otpustite taster CTRL — računar je svakako već izvršio zadatu naredbu!

Skrećemo vam pažnju na komandu CTRL C koju treba koristiti posle bilo koje zamene diskete u nekom od drajvova (osim ako je tu zamenu zahtevao sam računar u kom se slučaju pretpostavlja da je programer znao šta radi). Sa CTRL C, osim toga, prekidamo rad svakog sistemskog programa kao što je bežik interpreter i vraćamo kontrolu operativnom sistemu (prompt „A>“). lako moguće, izlazak iz aplikativnih programa sa CTRL C nije preporučljiv, jer može da ostavi neke datoteke otvorenima što obično čini da poslednje promene koje smo u njemu izvršili ne budu uvažene. Još je opasnije pritisnati dirku RESET koja je obično smeštena sa zadnje strane komputera: ovaj je taster rezervisan samo za situacije u kojima nikako drukčije ne možemo da spasemo podatke! Zaključak: koristite, kada je god to moguće, naredbe poput QUIT da biste prekinuli rad nekog programa i vratili kontrolu CP/M-u.

Dejan Ristanović



Bolji dani za 128

Kada već spominjemo PC-128 i Komodor, treba reći da sa pojavom novog diska 1571 polako ali sigurno skače i prodaja ovog računara. Za 128 se pojavilo preko 50 različitih knjiga, a posebno je interesantan i imponentan broj CP/M programa. Programi se pojavljuju u velikim količinama — već se mogu nabaviti svi bolji CP/M programi, a kompatibilnost sa legendomom Žeđeset-četvorkom je dokazana 100%. Znači, hrpa igara i dosta odličnih poslovnih programa već u startu! Možda je to pravi izbor, naravno u očekivanju evropske „amige“.

Ada za „komodor“

Za one vlasnike „komodora“ koji ne žele da menjaju svoje ljubimce, posle male pauze programi ponovo stižu u ogromnim količinama. Igre već postaju neverovatne i prelaze sve granice (Lukas film games — na primer), HACKER postaje legenda, ali stižu i vrlo korisni i profesionalno urađeni korisnički programi. Mnoge programere će, sigurno, interesovati da mogu da nabave i odličan C kompajler, ali i adu, koji su trenutni vođeci programskih jezika. Ne treba posedeći da je ada zvanični jezik Pentagona i da su na jeziku C pisani operativni sistemi za „amigu“ i, za „atari 520“.

Sam svoj doktor

Za „komodor“ se nedavno pojavio program za lečenje u kući. Program vam postavlja pitanja, deli savete i, na kraju, postavlja dijagnozu. Program zauzima dve diskete i napravljen je potpuno profesionalno. Već posle prvih razgovora dobio sam savet da posetim lekara, što je dokaz da program odlično radi.

Amiga dolazi

Amiga dolazi... Polako, ali sigurno pojedine firme i „dileri“ već reklamiraju Commodore Amigu po obećanim cenama. Očekujemo ozbiljnije vesti sa velikim interesovanjem.

Hakeri igraju hakera

„Komodoristi“ — da li igrate HACKERA. Nemoguće da još niste nabavili ovaj čuveni program. Ako ga igrate, pošaljite nam svoja iskustva i rešenja. Još niko nije rešio HACKERA.

sažimanje teksta

Programiranje na mašincu

Često se za nekoga kaže: „On piše stenografskom brzinom“. Istina je, stenografi pišu vrlo brzo, ali koriste i neke „trikove“ koji i nama mogu dobro doći. Broj slova u tekstu značajno se smanjuje ako se iz njega izbače neki samoglasnici. U većini slučajeva, značenje reči se interpretira bez teškoća: televizor — tivzor. Da značenje reči ipak ne bi došlo u sumnju, slovo iza koga sledi neki samoglasnik piše se na poseban način — ili se spušta malo niže u odnosu na ostali tekst, ili se nekako zobači.

Interni kod

Slična tehnika se može primeniti i na računaru. Pri upisivanju ASCII koda slova u memoriju, setovanjem nekog bita moguće je naznačiti koje je slovo sledeće. Sledeće slovo se ne memorise posebno, već poseban program, čitajući bitove ASCII koda zabeleženih slova, sam umeće znakove koji nedostaju. Osim samoglasnika, iz teksta se mogu odstraniti i grupe slova. Stenografi takvu grupu nazivaju „sigla“. U našem jeziku su česte sledeće grupe: st.zd.zn.sp.pl.spl.pr.tr.kl.kv.dv.vl. ... Ispred ili iza sigle se pojavljuju samoglasnici. Sem sigli, u tekstu postoji i određen broj reči koji se ponavljaju više puta. Takvu reč, ili siglu programeri nazivaju tokenom. Token se u memoriju upisuje slovo po slovo, samo jednom, i to u posebnu tabelu. Svaki token označen je svojim brojem koji ga u proizvoljnom tekstu zamenjuje. Sve bezik instrukcije i imena funkcija na „spektrumu“ su tokenizovane, tj. u memoriji se predstavljaju samo jednim brojem. Na primer, naredba PRINT se beleži kao broj 245.

Prema ASCII kodu, sve znakove (slova, brojevi, specijalni znaci) moguće je predstaviti sa 7 bita. Pošto u bajtu ima 8 bita, poslednji bit možemo upotrebiti kako hoćemo. Postoji, međutim, i mogućnost da se oslobodi još jedan bit. Formiraćemo karakter set koji će se sastojati od velikih i malih slova, brojeva i znakova „?“, „!“, „...“. Ukupan broj znakova sada iznosi 63. Kako ASCII kodovi za navedene znakove predstavljaju neprekidan niz brojeva, potrebno je formatirati interni kod od 0 do 62. To je učinjeno prema datoj tabeli.

Znak	ASCII kod	Interni kod
a-z	97-122	0-25
A-Z	65-90	26-51
0-9	48-57	52-61
	46	62

Slova Q i q zamenjena su blankom znakom i znakom pitanja. Čini se zgodnim da se slova: w.x.y.W.X.Y. ekvivalentiraju sa

26/programiranje na mašincu

```

1 CLEAR 29999
10 GO SUB 9000
11 REM *****
12 REM POCETAK GLAVNE PETLJE
13 REM *****
20 LET n=0:n
30 POKE 23659,9: INPUT LINE a#
35 IF a#="END" THEN SAVE "C.TX
T"CODE n,65535-n+1: STOP
40 LET kr=LEN a#
50 LET i=1
60 IF i)kr THEN GO SUB 400: GO
TO 20
70 LET c=CODE a$(i)
80 GO SUB 200
90 IF (i+1)=(kr THEN GO SUB 30
0
100 POKE n,c
110 LET n=n-1
120 LET i=i+1
130 GO TO 60
131 REM *****
132 REM PODPROGRAM ZA KODIRANJE
133 REM UNESENOG TEKSTA
134 REM *****
201 IF c)143 THEN IF c(147 THEN
LET c=c-122: RETURN
202 IF c)146 THEN IF c(150 THEN
LET c=c-99: RETURN
203 IF c=32 THEN LET c=42: RETU
RN
204 IF c=63 THEN LET c=16: RETU
RN
205 IF c)96 THEN LET c=c-97: RE
TURN
210 IF c)64 THEN LET c=c-39: RE
TURN
220 IF c)47 THEN LET c=c+4: RET
URN
230 LET c=62
240 RETURN
241 REM *****
242 REM PODPROGRAM KOJI VRSI
243 REM SAZIMANJE OKO ZNAKOVA:
244 REM " ", "a", "e"
245 REM LENTIRANJE ZNAKOVA:
247 REM *****
300 LET i=CODE a$(i+1)

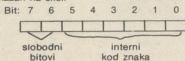
```

```

301 IF i=32 THEN LET c=c+64: LE
T i=i+1: RETURN
310 IF i=97 THEN LET c=c+128: L
ET i=i+1: RETURN
320 IF i=101 THEN LET c=c+192:
LET i=i+1: RETURN
330 RETURN
400 REM *****
401 REM IGPISIVANJE TEKUCIH PO-
402 REM DATAK I STAMPANJE UNE-
403 REM TOG TEKSTA
404 REM *****
410 POKE n,255: LET n=n-1
420 PRINT AT 1,15:n
430 LET u=kr-(n=0:c-n)+u
440 PRINT AT 2,15:u
450 LET n=n+1
460 PRINT AT 3,15:p
470 LET koi=0: LET n#1
480 IF n)kr THEN RANDOMIZE USR
3564: LET koi=0: RETURN
490 IF koi=32 THEN LET koi=0: R
ANDOMIZE USR 3564
500 PRINT AT 3,koi+a$(n)
510 LET n=n+1: LET koi=koi+1
520 GO TO 460
9000 REM *****
9001 REM INICIJALIZACIJA
9002 REM *****
9003 BORDER 0: PAPER 0: INK 7: L
0:0
9004 CODE : CLG
9005 LET p=0: LET u=-170: LET n=
65190
9006 PLOT 0,172: DRAW 255,0: DR
A 0,-64: DRAW -255,0: DRAW 0,64
9007 PRINT AT 1,1:"Tekuca adresa
1:"n
9008 PRINT AT 2,1:"Usteda(bajta)
1:"u
9009 PRINT AT 3,1:"Poruka broj
1:"p
9010 PRINT AT 5,7: PAPER 3:"KOMP
RESOR TEKSTA"
9011 PRINT AT 7,1:"A. Radovanovic
1986."
9990 RETURN
9999 CLEAR : SAVE "T.K." LINE 1:
SAVE "CODE "CODE 55197,170

```

prvih 6 UDG znakova. Ovim se otvara mogućnost da korisnik definiše grafiku u okviru teksta, ili slova: 6,6,5,2. Kod slova se u memoriju upisuje kao bajt čiji je sadržaj prikazan na slici.



Sažimanje teksta

Kompresija teksta u našem programu zasniva se na upotrebi internog koda i slobodnih bitova. U jedan bajt se upisuje dve informacije. Šesti i sedmi bit kodiraju sledeća stanja: 00-bajt sadrži samo znak

predstavljen internim kodom, 01-iza znaka upisanog u prvih 6 bita sledi blanko znak, 10- iza znaka upisanog u prvih 6 bita sledi slovo „a“, 11-iza znaka upisanog u prvih 6 bita sledi slovo „e“.

Bilo da se pišu igre avanture ili obrazovni programi, uvek je u datom trenutku potrebno na ekranu prikazati samo jedan deo teksta koji se naziva poruka. Program dat na bezik listingu prima poruke koje korisnik ukucava, kodira ih, izbacuje slova „a“ i „e“, kao i blanko znake, i upisuje u memoriju. Kraj poruke će biti označen bajtom čija je vrednost 255. U toku rada sa programom na raspolaganju je informacija o uštedi u memoriskom prostoru. Program je izuzetno efikasan — prosečna ušteda je

Kako
se
koristi

devpac mons

Ni jedan iskusni programer ne očekuje da će mu program proraditi od prve. Jednostavno, bez i jedne trunke emocija treba prihvatiti činjenicu da izvršenje najnovijeg programa protiče na način koji nismo mogli ni sanjati. Dešava se to svakome, pri startovanju bilo kakvog programa i pri izboru bilo kog programskog jezika.

Sa bejzikom je stvar nešto prostija, jer bar ne može nikad (gotovo nikad) doći do blokiranja računara. Čim vidimo da program radi nešto što mu nije posao, pritisnemo **BREAK** i potražiti grešku u listingu.

Mašinski programi, međutim, priređuju daleko veće teškoće. Čak i kada se program izvrši do kraja, ne možemo baš biti sigurni da li je on uradio samo ono što smo želeli, ili je, možda, usput izveo i nešto po svom nahođenju.

U takvim situacijama, kada nismo sigurni šta program zapravo radi, ili kada nam nije jasno zašto uopšte neće da radi (a trebalo bi da radi), pozvamo u pomoć **disassembler** — program koji olakšava testiranje mašinskih rutina i obavlja niz drugih korisnih poslova. Pomoću disasemblera možemo jednostavno menjati naredbe mašinskog programa, izvršavati program korak po korak, bilo kom trenutku pročitati sadržaj procesorskih registara, ili sadržaj bilo kog bloka memorije. Jednostavno rečeno, bez disasemblera se ne može ni zamisliti razvoj bilo-kakve komplikovanije mašinske rutine.

Šta je MONS

Disassembler pod nazivom MONS nabavlja se paketu sa odgovarajućim asemblerom. Proizvođač je softverska firma „Hi-soft“, a čitav paket je poznat pod imenom devpac. Namenjen je računarima sa procesorom Z80, a mi ćemo, konkretno, ovde govoriti o verziji za „spektrum“ sa oznakom MONS3M. Razlike među pojedinim verzijama su minimalne, i na njima se nećemo zadržavati.

Program MONS3M zauzima tačno 5800 bajtova i može se startovati sa bilo koje adrese u RAM-u. Najposlednije je smestiti ga u sam vrh memorije, recimo na 58000.

LOAD "" CODE 58000

Prvo startovanje programa vrši se od početne adrese:

RANDOMIZE USR 58000

Međutim, svako sledeće startovanje vrši se od adrese koja je za 29 veća od početne:

RANDOMIZE USR 58029

Ova promena početne adrese uslovljena je izmenama u samom MONS-u, u toku njegovog prilagođavanja adresi na koju je ućtan.

Neposredno po startovanju programa, nakon kraće pauze i poruke proizvođača, ugledaće ekran popunjen slovima i ciframa, gotovo, reklo bi se, sračunat tako da

```
C372 C9          RET
PC C372         C9 00 00 00 00 00 00
SP 7518         2B 2D 65 33 30 40 ED
IY 503A         FF ED 00 2C 75 2C 75
IX 03D4         04 0C 0D 20 FD 0E 3F
HL 5CDD         00 50 02 00 EA 0D 00
>DE 5D12        00 90 E2 00 00 0A 00
BC 0000         F3 AF 11 FF FF C3 CB
AF E720
IR 3F45
```

```
C366 CC         C36E 19 C376 00
C367 B8         C36F C4 C377 00
C368 19         C370 E5 C378 00
C369 E1         C371 19 C379 00
C36A E6         >C372 C9< C37A 00
C36B A7         C373 00 C37B 00
C36C ED         C374 00 C37C 00
C36D 52         C375 00 C37D 00
```

pretraži početnike. Međutim, radi se o vrlo dobro organizovanoj i preglednoj tabeli, iz koje možemo saznati sve što bi nas interesovalo u toku disasembliranja.

Englezi su ovaj ekran označili imenom „Front panel display“, a mi možemo usvojiti naziv „radna tabela“. Prvo ćemo, svakako, objasniti šta znači svaki znak u toj tabeli.

Radna tabela

Primer jeone radne tablete dat je na našoj slici. U donjoj polovini ekrana uvek se nalazi prikazan blok od 24 bajta memorije, u okolini adrese koju unapred izaberemo. U našem primeru, izabrana je adresa &C372, i njen sadržaj je, kao što se vidi, &C9. Znaci „>“ i „<“ okružuju zadatu adresu i predstavljaju tzv. *pokazivač memorije* („memory pointer“).

U samom vrhu ekrana se nalazi disasemblirana mašinska naredba adresirana memorijskim pokazivačem. U našem primeru, radi se o naredbi **RET**.

Zatim slede sadržaji svih registara procesora Z80. Znak „>“ je *pokazivač registra*, a služi nam da izaberemo registar koji želimo da izmenimo. Sadržaj svakog registra je ispisan u istom redu desno, u heksadekadnom obliku. Recmo, HL sadrži &5CDD, a akumulator A sadrži &E7. Registar F je trenutno postavljen na &20, što znači da ni jedan indikator nije setovan (setovan je samo bit 5, ali, kao što znamo, bitovi 1-5 registra F se ne koriste). Kada je neki od indikatora setovan, onda se, desno od sadržaja registra F, ispisuje i oznaka odgovarajućeg indikatora. Recimo, u slučaju F=&40, setovan je indikator nule „Z“, pa bismo na ekranu ugledali:

```
AF &E740 Z
```

Svaki indikator ima svoje slovo, pa ako je to slovo ispisano, znači da je indikator setovan. U suprotnom je resetovan.

Kada su svi indikator setovani, imamo ispis:

```
SZ H VNC
```

Prazna polja odgovaraju nekorisćenim

citovima 3 i 5. Slova se odnose na sledeće indikatore:

S — indikator znaka („sign flag“).
Z — indikator nule („zero flag“).

H — indikator poluprenosa („half carry flag“).

V — indikator parnosti/prekoračenja („parity/overflow flag“).

N — indikator oduzimanja („subtract flag“).

C — indikator prenosa („carry flag“).

Što se tiče 16-bitnih registara PC, SP, IX, HL, DE, BC, desno od njihovog sadržaja nalazi se niz od po sedam bajtova, počev od adrese koju registar trenutno sadrži. Tako, na primer, vidimo da na adresi &5CDD (sadržaj registra HL), počinje blok bajtova: &00, &50, &02, &00, itd.

U samom dnu ekrana nalazi se kursor, spreman za prihvatanje naših naredbi. Prelaz sa malih na velika slova i obrnuto (kursori L i C) vrši se na uobičajen način (**CAPS LOCK**).

Naredbe

MONS3M dopušta veliki broj naredbi koje korisnik unosi sa tastature. Svaka naredba je, u stvari, *jedno ukucano slovo*, bez potrebe za tasterom ENTER. Izvršenje naredbe počinje, dakle, čim je ukucamo, što ponekad može da nam pričini neprijatnosti, ukoliko pritisnemo taster koji nismo želeli. Zato, pri tadu sa MONS-om, moramo biti vrlo opazivi i dobro promisliti pre svakog poteza.

Pogrešne (nepostojeće) naredbe bile jednostavno ignorisane. Radna tabela se menja posle svake izvršene naredbe, tako da uvek možemo videti rezultate onoga što smo upravo izveli.

1. EDIT (CAPS SHIFT 1)

Naredba **EDIT** obavlja povratak u bejzik tačnije skok u glavnu izvršnu petlju bejzik interpretera. Ponovo pozivanje MONS-a vrši se od adrese za 29 veća od početne.

2. # (SYMBOL SHIFT 3, povisliska)

Ovom naredbom zahteva se ispisivanje svih adresa u radnoj tabeli u *decimalnom* obliku, umesto uobičajene heksadekadne forme. Istom naredbom vrši se i vraćanje u heksadekadni ispis.

3. ENTER

Pomeranje pokazivača memorije za jedan bajt unapred.

4. ↑ (CAPS SHIFT 7, strelica gore)

Pomeranje pokazivača memorije za jedan bajt unazad.

5. → (CAPS SHIFT 8, strelica desno)

Pomeranje pokazivača memorije za osam bajtova unapred.

6. ← (CAPS SHIFT 5, strelica levo)

Pomeranje pokazivača memorije za osam bajtova unazad.

7. izmena sadržaja memorije

Kada želimo da promenimo sadržaj bajta, adresiranog pokazivačem memorije, ukucamoće novu heksadekadnu vrednost i zatim pritisnuti **ENTER**, ili bilo koji drugi komandu za pomeranje pokazivača memo-

Nabavka jednog dobrog komercijalnog programa, u našim uslovima, predstavlja pravu malu avanturu. Najčešće nemamo dovoljno novca da bismo kupili originalnu verziju programa u Engleskoj, pa smo, tako, prinuđeni da se okrenemo našem razvijenom piratskom tržištu. Naše softverske „firme“, međutim, ne vole rad sa papirnom i „svoje“ programe isporučuju obično bez uputstava. Ako uputstvo i stigne, brzo shvatamo da od njega nema mnogo koristi — uputstvo je pisano na engleskom, a mi smo u školi učili ruski. Sledi, dakle, потрага за prevodom... Redakcija „Računara“ je, da bi pomogla svojim čitaocima, odlučila da prikaže kvalitetnije komercijalne programe, sa uputstvom za njihovo korišćenje.

Program: DELETE

Test primer za rad sa MONS-om

```

L_ADDR EQU &196E      Rutina LINE__ADDR iz ROM-a
NXTONE EQU &19B8      Rutina NEXT__ONE iz ROM-a
RCLM 1 EQU &19E5      Rutina RECLAIM__1 iz ROM-a
REF_0 EQU &288B      Izdavanje raporta 0
E_PPC EQU &5C49      Sistemska promenljiva
ORG
DELETE LD (E__PPC),DE Postavljanje programskog kursora
AND A                Bisanje indikatora prenosa
SBC HL,DE           Da li je početak veka od kraja?
JP C,REP_Q          Ako jeste, prijavl grešku
ADD HL,DE           Obnovi konačnu liniju.
EX DE,HL            Početak u HL, kraj u DE.
PUSH SP             Sačuvaj krajnju liniju.
CALL L__ADDR        Nadi početnu adresu.
EX (SP),HL          Početna adresa na stek, krajnja linija u HL.
CALL L__ADDR        Nadi adresu krajnje linije.
LD E,L              Kopiraj tu adresu u
LD D,H              registar DE.
CALL Z,NXTONE      Ako je linija nađena, pomeri se iza nje.
POP HL             Početna adresa u HL.
EX DE,HL           Početna adresa u DE, krajnja u HL.
AND A              Briši indikator prenosa.
SBC HL,DE           Da li su početak i kraj isti?
ADD HL,DE          Obnovi kraj.
CALL NZ,RCLM__1   Izbacni blok samo ako početak i kraj
RET               nisu isti.
                  Povratak u bejzik.

```

dobiti heksadekadnu adresu &C350. Zato dalje možemo uraditi M: C350, čime dovodimo pokazivač memorije na prvi bajt programa. Još moramo pripremiti i registar PC. Ukucavamo C350 i pritiskamo tačku. Time je sve spremno za izvršenje programa korak po korak. Jedino nismo pripremili registre DE i HL, kao ulazne parametre.

Recimo da želimo izbacivanje linija 40 — 70. Tada u HL treba staviti &0046 (tri puta tačka, 0046, tačka), a u DE &0028 (tačka, 0028, tačka).

Možemo malo zastati i pogledati izgled radne tabele. U vrhu ekrana je prva naredba programa LD (E__PPC),DE. Registar IY sadrži &5C3A, što je neophodno ako zovemo potprograme iz ROM-a. Registri PC, HL i DE su spremni za izvršenje našeg programa, a SP sadrži adresu &7518. Primjećujemo da je poslednja adresa na steku &2D2B, inače uobičajena za povratak u bejzik (STACK__BC).

Mirno možemo izvršiti prvu naredbu, pritiskom na „;“. Ako nas interesuje da li je naredba zaista izvršena, pročit ćemo sadržaj E__PPC jednostavnim promenom pokazivača memorije M: 5C49. Videćemo da je vrednost DE registra stvarno tamo. Zato se vraćamo nazad, na adresu sadržanu u PC (sledeća naredba našeg programa) M: C354.

Izvršimo, bez mnogo razmišljanja i sledeće dve naredbe: **AND A i SBC HL, DE.**

Sledi veoma opasna naredba JP C, REP_Q. Da je, nekim slučajem, u ovom trenutku setovan indikator prenosa, a mi izvršimo komandu „;“, došlo bi do blokade. Izveštaji za grešku nisu mogući iz MONS-a. Međutim, pošto je kod nas „C“=0, slobodno možemo izvršiti tu naredbu.

Sledeće tri naredbe su, takođe, sasvim bezopasne. Posle PUSH DE možemo samo obratiti pažnju na novo stanje mašinskog steka. Videćemo da je SP umanjen za dva, kao što i treba očekivati.

Nova uzbuza nastaje na naredbi CALL L__ADDR. Pozivanje potprograma iz ROM-a nije uvek preporučljivo, ako ne znamo tačno šta će se desiti. Međutim, ovde se radi o programu koji doslovno za svaki ulazni podatak obavlja uspešan povratak A, osim toga, naš ulazni podatak sigurno ima smisla (linija 40 postoji u programu bez sumnje). Dakle, slobodno možemo izvršiti CALL L__ADDR koristeći naredbu „;“. Nema razloga da tamo idemo korak po korak, ali vama savetujemo da to probate kao vežbu.

Na povratku iz L__ADDR, HL će biti postavljen na početak linije 40 (adresa &5CDD), a DE je na početku prethodne. Indikator nule je setovan, jer je pronađena linija koja je i tražena.

Posle naredbe EX (SP), HL ponovo imamo CALL L__ADDR, za šta opet koristimo komandu „;“. Slede dve jednostavne naredbe, a onda opet oprez: CALL Z,NXTONE. Pošto je indikator nule setovan, do poziva će sigurno doći. Jedino se treba odlučiti da li pozvati potprogram sa „;“, ili ga izvršavati korak po korak. Taj potprogram baš nije blagonaklon prema bilo kakvim ulaznim podacima, ali kod nas je sve u redu, i slobodno možemo izvršiti „;“. Ostatak programa nije nikakav problem, sve do poslednje naredbe, koja poziva rutinu RECLAIM iz ROM-a, a tu se kriju mnoge zamke. Na ulazu nipošto ne smeju HL i DE biti jednaki. Naš program o tome samo vodi računa. Dalje, DE mora biti na prvom, a HL na poslednjem bajtu koji se izbacuje. Kod nas je očigledno sve u redu i mirno možemo izvršiti „;“.

Tako smo, bez većih teškoća stigli do naredbe RET. Radna tabela ima izgled upravo kao na slici koju objavljujemo (jedino registar R može imati drugačiju vrednost). Vrat ćemo se u bejzik naredbom EDIT i uveriti se da u programu nedostaju linije 40, 50, 60 i 70.

Monsom po programu

Ilustrovaćemo rad sa MONS-om na primeru jednog jednostavnog programa, čiji listing objavljujemo. Radi se o operaciji DELETE, koja izbacuje blok bejzik linija iz programa. Na ulazu, DE sadrži linijski broj početne, a HL linijski broj krajnje linije. Već je prethodno provereno da oba linijska broja leže u korektnom opsegu (1 — 9999).

Programski kursor se postavlja u liniju DE.

Ukoliko je konačni linijski broj manji od početnog, program treba da prijavl grešku „O Parameter error“. U protivnom, mora iz bejzika izbaciti sve linije počev od DE (ili od prve sledeće, ako takva ne postoji), pa zaključno sa HL (ili prvom prethodnom, ako takva ne postoji). Ako je HL jednako DE, izbacuje se samo jedna linija, ali ako takve nema ništa se neće preduzeti.

Resetovano računari i izvršiti CLEAR 29999. Zatim učitaćemo GENS na 30000 i MONS na 58000. Startujemo GENS i ukucamo ceo program DELETE, naravno, bez komentara. Asembliramo ga i vratimo se u bejzik.

U bejziku ukucamo kratak program nevažnog sadržaja:

```

10 REM
20 REM
30 REM
40 REM
50 REM
60 REM
70 REM
80 REM
90 REM

```

Konačno, startujemo MONS na adresi 58000.

Potražićemo, pr svega naš program u memoriji. Početku mu je na 50000, pa zato izvršimo H: 50000. Kao odgovor ćemo

rije (naredbe pod brojevima 3, 4, 5 i 6). Vrednost zadatog bajta biće, nakon toga, izmjenjena, a pokazivač memorije postavljen na novu adresu.

8. M

Naredbom M može se postaviti pokazivač memorije na bilo koju zadanu adresu, ukucavanjem odgovarajuće heksadekadne vrednosti iz M. Tasterom ENTER završava uokucavanje adrese.

5. (SYMBOL SHIFT M, tačka)

Naredba „e“ pomena pokazivač registra za jedno mesto unapred. Kretanje je ciklično, u redosledu: PC, IY, IX, HL, DE, BC, AF. Recimo, ako je pokazivač trenutno na PC, a mi želimo da ga dovedemo na HL, moramo tri puta izvršiti „e“. Vraćanje na PC zahteva još četiri iste operacije. Registri SP i IR se ne mogu adresirati.

10. Promena sadržaja registra

Sadržaj registra na kome upravo stoje pokazivač, može se izmeniti direktno ukucavanjem novog heksadekadnog sadržaja, praćenog tačkom. Pri tome se položaj pokazivača neće promeniti.

11. G

Naredba G vrši pretraživanje memorije, počev od trenutnog stanja pokazivača memorije, tražeći zadati niz bajtova. Čim pritisnemo G, od nas se očekuje da unesemo niz bajtova za kojim će se tragati. Ukucavamo redom bajt po bajt, u heksadekadnom obliku, a odvajamo ih međusobno sa ENTER. Kada je niz kompletiran, još jednom pritisakom ENTER, i traganje počinje. Zadati niz će uvek biti pronađen, ako ne na drugom mestu, onda u samom MONS-u, tamo gde smo ga sami ukucali. Čim se niz otkrije, pokazivač memorije se postavlja na njegov početak i vraća se u radnu tabelu.

12. N

Naredba N pronalazi u memoriji sledede pojavljivanje niza za kojim je tragala prethodna naredba G. Na taj način, koristeći naredbu G i zatim uzastopno N, moguće je pronaći u memoriji sva mesta gde postoji zadati niz.

13. I

Naredbom I vrši se kopiranje bloka bajtova sa jednog mesta na drugo. Usledice pitanja „First:“ i „Last:“, na koja moramo odgovoriti početnom i krajnjom heksadekadnom adresom bloka koji kopiramo, dok iz „To:“ ukucavamo adresu na koju blok šaljemo. To može biti čak i neka adresa unutar samog bloka koji se kopira. Slovo I se, u stvari, odnosi na „inteligentno“ kopiranje.

14. H

Pretvaranje dekadnog broja u heksadekadni. Iza H treba ukucati dekadni broj koji prevodimo, praćen sa ENTER. Rezultujući heksadekadni niz će se pojaviti u produžetku istog reda.

15. Q

Prelazak sa glavnog seta registra (HL, DE, BC, AF) na alternativni (HL', DE', BC', AF') obrnuto.

16. P

Popunjavanje izabranog bloka memorije zadatim bajtom. Na pitanja „First:“ i „Last:“ odgovaramo početnom i krajnjom adresom bloka, a iza „With:“ zadajemo bajt kojim treba popuniti blok.

17. L

Listanje bloka bajtova počev od adrese pokazivača memorije. Listanje se vrši sa po četiri bajta u jednom redu, praćeno odgovarajućom ASCII interpretacijom. Pri tome se bajtovima većim od 87F oduzima 80 (resetuje se bit 7). Bajtovi manji od 820 prikazuju se tačkom.

Svaku novu stranicu listanja dobijamo pritiskom na bilo koji taster osim EDIT (povratka u bejziki) i ← (povratka u radnu tabelu).

18. (SYMBOL SHIFT P, znak navoda)

Ova naredba obavlja isti posao kao i L, s tim što se izlaz vrši na štampač.

19. Y

Naredbom Y može se u memoriju, počev od trenutnog položaja pokazivača memorije, ubaciti proizvoljan ASCII niz, bez potrebe za ukucavanjem heksadekadnih vrednosti znakova. U dnu ekrana će se pojaviti nova linija, u koju unosimo niz, slovo po slovo. Kao oznaku za kraj treba ukucati ←. Nova vrednost pokazivača memorije biće na prvaj adresi iza unetog niza.

20. \$ (SYMBOL SHIFT 4, dolar)

Naredba \$ obavlja disasembliranje bajtova počev od trenutnog položaja pokazivača memorije. Listing se prikazuje na ekranu i može se produžavati pritiskom na bilo koji taster, osim EDIT (povratka u bejziki). Vraćanje u radnu tabelu obavlja se, takođe, sa \$.

21. T

Naredba T je daleko moćnija od \$. Moguće je disasemblirati proizvoljan blok bajtova i formirati tekst koji će kasnije biti prihvaćen od strane GENS assemblera.

Na pitanje „First:“ i „Last:“ ovide odgovaramo početnom i krajnjom adresom bloka. Ako krajnja adresa bude manja, izvršenje naredbe će biti obustavljeno.

Na pitanje „Printer?“ odgovorimo sa „Y“ ako želimo listanje na štampaču. Bilo koji drugi odgovor (najjednostavnije ENTER) podrazumeva samo listanje na ekranu.

Pitanje „Text:“ zahteva od nas adresu u memoriji na kojoj će se formirati tekst disasembliranog programa. Ako ne želimo da se tekst formira, pritisnućemo samo ENTER.

Ukoliko smo zahtevali generisanje teksta, usledice pitanje „Workspace:“. Moramo ukucati adresu na kojoj će disasemblir formirati tabelu sa simboličkim adresama. Za svaku simboličku adresu biće utrošena dva bajta radnog prostora. Ako umesto odgovora ukucamo samo ENTER, podrazumeva se radni prostor na 8000.

Konačno, od nas će se tražiti da definišemo sve blokove podataka koji, eventualno, postoje u programu i koje, stoga, treba isključiti iz procesa disasembliranja, uz jedinstavno generisanje DEFDEF naredbi. Usledice niz pitanja „First:“ i „Last:“, na koja odgovaramo početnom i krajnjom adresom bloka podataka. Sve te adrese MONS će pamtili u prostoru za sopstvenog koda. Prema tome, broj blokova podataka koje možemo zadati ograničen je jedino raspoloživim memorijskim prostorom iz MONS-a. Svaki blok zahteva četiri bajta.

Kad budemo zadali sve blokove podataka, na poslednji par pitanja „First:“ i „Last:“ treba odgovoriti sa ENTER.

Ekran se briše i počinje disasembliranje uz listanje odgovarajućeg teksta. Privreme-

ni prekid može se ostvariti pritiskom na ENTER, ili SPACE. Nastavak se tada vrši bilo kojim tasterom, dok ← obavlja povratka u radnu tabelu.

Na kraju disasembliranja, ako smo zahtevali formiranje teksta, imaćemo poruku „End of text:“, iza koje stoje adresa prvog bajta iza samog teksta. Razlika te adrese i početne dade dužinu teksta. Možemo se vratiti u bejzik i snimiti generisani tekst na kasetu, da bismo ga kasnije uzeli u assembler naredbom G.

Međutim, ako je GENS istovremeno nalazi u memoriji, tekst možemo formirati odmah tamo gde ga assembler i očekuje. Procedura u tom slučaju bila bi sledeća:

Prvo u GENS-u izvršimo naredbu X. Od dve adrese koje nam pri tom budu ponuđene, uzemo prvu, u slučaju da želimo brisanje već postojećeg programa, ili drugu, ako želimo da se nadovežemo na postojeći program. Naravno, ako nikakvog programa u GENS-u nema, obe adrese su iste i uzemo bilo koju. Vraćamo se u MONS i tu adresu navodimo u okviru naredbe T, kao odgovor na pitanje „Text:“. A kada na kraju dobijemo adresu završetka teksta, moramo je upisati u GENS-ovu sistemsku varijablu TEXTEND (to su dva bajta koja počinju na adresi za 54 većoj od početne adrese GENS-a). Jednostavno ćemo iz MONS-a izvršiti naredbu M i postaviti pokazivač memorije na TEXTEND, a onda u ta dva bajta upisati zaključnu adresu teksta (prvo bajt manje težine).

Vraćamo se, zatim, u bejzik, i startovati GENS (vruc start). Primitičemo da tekst koji je formirao MONS nije baš najpreledniji, jer nedostaju kodovi za tabulaciju (tekst je maksimalno kondenzovan). To se može ispraviti naredbom c u okviru GENS-a.

Sve simboličke adrese koje MONS generiše počinju slovom L, ispred apsolutne heksadekadne adrese (npr. L75CD).

22. (SYMBOL SHIFT Z, dve tačke)

Naredba „...“ koristi se pri izvršavanju mašinskog programa korak po korak (ako izvršavamo program iz ROM-a, to je i jedini način). Izvršuje se naredba koju trenutno adresira pokazivač memorije, ali i registar PC mora obavezno sadržati istu adresu. U protivnom će doći do kraha.

23. W

Postavljanje prekidne tačke na adresu određenu pokazivačem memorije. Prekidna tačka služi MONS-u kao znak da na tom mestu treba prekinuti izvršenje programa (ukoliko je izvršenje u toku) i obaviti povratka u radnu tabelu. Primitičemo da, po izvršenju naredbe W, tri bajta našeg programa bivaju zamenjena jednom CALL instrukcijom koja označava prekid. Naša tri bajta se čuvaju, i biće obnovljena čim prekidna tačka bude obavila svoju funkciju. Pred ta tri bajta, mora se čuvati i adresa prekidne tačke, što, sve skupa, znači pet bajtova potrebnog memorijskog prostora po svakoj prekidnoj tački. Njih može u programu biti proizvoljan broj, sve dok ima dovoljno memorije iz MONS-a (naredba W koristi isti onaj prostor kao i naredba T, o čemu moramo voditi računa, jer će svaki poziv naredbe T upropasti naše prekidne tačke).

24. + (SYMBOL SHIFT K, plus)

Nastavak izvršenja programa počev od trenutnog sadržaja programskog brojača PC (stanje pokazivača memorije nije pri-

tome bitno). Program se zaustavlja kad naiđe na prekidnu tačku, uz povratak u radnu tabelu.

25. > (SYMBOL SHIFT T, veće)

Postavljanje prekidne tačke iz tekuće naredbe i nastavak izvršenja programa. Jednostavnije rečeno, biće izvršena naredba adresirana pokazivačem memorije i registrom PC, na potpuno isti način kao da smo sami postavili prekidnu tačku iz te naredbe i onda izvršili komandu „+“. Očigledna je razlika ove naredbe od „...“. Naredbom „...“ će, recimo, u slučaju CALL naredbe, samo biti obavljen skok na prvu naredbu potprograma koji se poziva, dok će naredba „+“ izvršiti čitav potprogram i vratiti se na naredbu iz CALL. To je korisno kada smo sigurni da potprogram radi bez problema, tako da nema potrebe da se i po njemu šetamo korak po korak.

26. J

Izvršenje mašinskog programa počev od zadate adrese. Pre nego što ukucamo samu hexadekadnu adresu, možemo obavistiti ovu naredbu komandom „U“ protivnom, skok se više ne može sprečiti. Povratak u radnu tabelu uslediće jedino pri nailasku na prekidnu tačku.

Naredba J je zgodna kada želimo da se vratimo u bežik, a bojimo se da to baš neće biti uspešno posle nekih „zahvata“ koje smo iz MONS-a izveli u protoru sistemskih varijabli i mašinskog steka. Tada prosto treba izvršiti J: 811B7, što je skok na naredbu NEW. Razume se, RAMTOP pri tome treba da je ispod MONS-a, jer će u protivnom i MONS biti uništen. Što se tiče bežik prostora, on se ovim nikadno ne može spasiti.

27. (SYMBOL SHIFT N, zarez)

Postavljanje pokazivača memorije na adresu sadržanu trenutno u vrhu mašinskog steka. Vrh mašinskog steka, s druge strane, određen je registrom SP.

28. O

Naredba O postavlja pokazivač memorije na novu vrednost, određenu relativnim skokom JR ili DJNZ naredbi. Pri tome, prihodni položaj pokazivača memorije ne sme biti na samoj JR naredbi, već na bajtu koji određuje relativnu adresu skoka.

29. U

Naredba U je inverzna naredbi O i obavlja vraćanje pokazivača memorije na vrednost pre skoka. Pri tome se pamti samo poslednja naredba O, dok eventualne prethodne bivaju zaboravljene, tako da u tom slučaju naredba U nema dejstva.

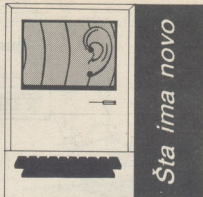
30. X

Naredba X pomera pokazivač memorije na vrednost apsolutnog skoka, datog JP ili CALL naredbom. Pri tome, pokazivač memorije mora stajati ne na samoj JP naredbi, već na nižem bajtu apsolutne adrese.

31. V

Naredba V je inverzna naredbi X i vraća pokazivač memorije na vrednost pre skoka. Pamti se samo skop poslednjom naredbom X, dok svi prethodni skokovi bivaju zaboravljeni, tako da tada naredba V nema dejstva.

Jovan Skuljan



Šta ima novo

Trampa

Ako imate „komodor 64“ i živite u Engleskoj, a želite „komodor 128“ možete da date svoj stari „64“ ovlašćenim prodavnicima i da, za uzvrat, dobijete popust od 50 funti pri kupovini „komodora 128“. Zar „64“ vredi samo 25 000 dinara?

Amstradova „amiga“?

Sve se glasnice čuju vesti o skorom završetku nove Amstradove mašine koja treba da ima oko pola megabajta, 68000, sve one mogućnosti koje i slična klasa (ST, „amiga“, ...) i, naravno, nižu cenu od konkurencije. Možda je to „amiga“ za obične smrtnike, pardon Jugoslovene.

Mek raste i jača

Svi oni koji su plakali nad nemoćnim i ograničenim „mekom“ biće silno obradovani kad čuju da je Epl spremio „mekintoša +“ koji bi imao 1Mb RAMa, 128K ROMa i dvostrane disk drajvove od 800K. Pomnije su čak i neko iznenađenje, pod tajnim imenom Karla, koje bi imalo 4Mb RAMa i hard disk. Eto sna za najvernije „mekovce“.

Slava teško umire

Ejkom je sklopio ugovor sa meksičkom kompanijom Dataum po kome Dataum ima pravo da sklupa i prodaje BBC i obadve Amerike. Nešto slično je uradio i na istoku, Ejkomer veliki tata. Olivetti, će prodavati BBC u istočnoj Evropi. Istočna Evropa — zvuči baš neutralno. Da nismo možda i mi tu uključeni?

Računar u kesi

Ako se pažljivo zagledaju reklame u svim računarskim časopisima (ne kod nas), lako je primetiti da je jako povećan broj reklama za računare u kiti, pogotovo za razne varijante PCa. Eto načina da se malo smanje cene!

Od bežika do bežika

Basic to basic je jako koristan program za vlasnike Enterprisa. On omogućava da se direktno u Enterprisov računar unose „spektrumovi“ bežik programi. BTB tada preinačuje „spektrum“ bežik u „enterprajz“ bežik i naznačava one delove programa koje korisnik mora sam ponovo da napiše. Zgodno?

Najbolje stvari su besplatne

U SAD postoji više hiljada baza podataka koje sadrže informacije koje se kreću u rasponu od cena za sirovu naftu u Dubaju do celokupnog softvera za IBM PC, ali ne postoji baza podataka, bar za sada, o tome šta koja baza podataka sadrži i koliko staje pristup određenoj bazi. Ipak, postoji način da dođete do toga kako i uz koju cenu šta možete da saznate iz odgovarajuće baze podataka. Dovoljno je da pišete na adresu: Information USA, 12400 Beal Mountain Rd., Potomac, Md 20854 (telefon je 301-983-8220) i da vas stave na listu osoba koje svaka tri meseca dobijaju besplatan katalog „Computer Data Report“ koji sadrži spisak svih baza podataka, i onih besplatnih (ako se izuzme telefon) i onih za koje se plaća pristup, kao i šta dotične baze podataka sadrže. Ako ne možete da čekate po 3 meseca, za 48 dolara ćete dobiti ažurne kataloge svakog meseca.

N. A.

Novogodišnji pobednici (i gubitnici)

Još uvek se ne zna precizno ko je najviše dobio od božićnog i novogodišnjeg buma. Veliki okršaj između Atarijevog „520 ST“ i „amige“, koji su najavljivali svi poznavoci mikrobiznisa, izgleda da je odošao za bolje dane. Ovog puta su, izgleda, dobro prošli i Atari i Komodor (Atari malo bolje), dok je žrtva ispao Epl kojem je drastično opala prodaja familije II, što mu je bio glavni izvor prihoda već duže vremena. Eto, tako smo ušli u 1986. koja će sudeći po početku, biti jako zanimljiva za računarstvo.

Jeftina animacija

Broderbund Software proizvodi specijalni Fantavision generator koji omogućava da se sa „epilor ile“ naprave animirane scene. Isti taj generator omogućava i transformisanje oblika između dva lika i postizanje specijalnih efekata. Generator proizvodi i do 64 sličice između dva oblika između kojih vrši animaciju. Sve to za 50 dolara.

Dečija posla

Evo nečeg za one koji su odrasli uz kompletne električne slagalice. Kupite vašoj deci The Robotic Computing Kit koji pruža mogućnost da se sastavi mnogo različitih kompleta (među njima i ploter) koje pose možete da kontrolišete vašim računarom.

Posle PCja AT (pa XT)

Proizvođačima je već dosadilo da prave mašine kompatibilne sa PCjem, pa su sada masovno prešli na savršenije verzije, to jest PC AT i PC XT. Jedna od takvih je Hewlett-Packardov Vectra PC koji je kompatibilan sa IBM PC AT. Sve je isto, samo je sv pomalo drugačije.

Još jedna kartica

Cumana je proizvela svoju verziju softverske kartice pod imenom Astron Integrated Circuit Card. Očekuje se Pocket Wordstar u džepnoj verziji. Za sve značajnije mikroracunare već postoje interfejsi.

Na tajnom kanalu

Javni ključ

Da bi se započela partija pokera telefonom, neophodan nam je odgovarajući način šifrovanja i dešifrovanja karata. Ovak način ima i lepo englesko ime: public-key protocol, i nešto manje romantičan prevod: protokol sa javnim ključem. Ovi protokoli su, po mišljenju mnogih naučnika, danas glavna i najuspešnija oblast istraživanja teorije informacija, teorije brojeva i računarstva i koriste se masovno da bi se obezbedila tajnost informacija.

Ideja protokola sa javnim ključem je jednostavna. Potrebno je konstruisati funkciju koju je lako izračunati u jednom i gotovo nemoguće u drugom pravcu — osim ako vas automatske jednosmerne funkcije (a ovakve se funkcije nazivaju jednosmernim) nije snabdela odgovarajućim informacijama. Pogledajmo blag, ali poučan primer: Data je funkcija:

$$y = x^{**} 4 + 12 * x^{**} 3 + 40 * x^{**} 2 + 24 * x - 37$$

i treba izračunati y za x=10. Lako se dobija da je 26203. Ali, kad je dato da je y=26203, nije preterano trivijalno izračunati x. Naravno, ako se ne zna da je

$$y = (x+3)^{**} 2 - 7)^{**} 2 - 41$$

Kad se to zna, treba jednostavno na y dodati 37, izvaditi kvadratni koren, dodati 7, opet izvaditi koren i oduzeti tri i dobija se traženo x.

Sa okusom pedagogije

Naravno, ovo je pedagoški primer i ni za jednog kriptologa ne bi predstavljao ni najmanju teškoću.

Pogledajmo još jedan protokol, sa javnim ključem prvio u pedagoškoj, a zatim i u znatno jačoj verziji (i ovaj ima lepo englesko ime: knapsack, u prevodu 'opet' prozaično: ranac):

Posmatrajmo vektor $E = (1, 2, 4, 8, 17, 35, 68, 142)$.

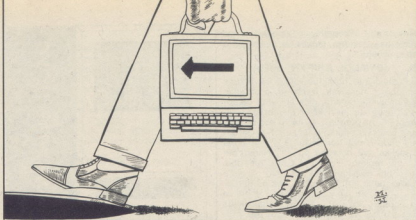
Osam komponenti ovog vektora čine superrastući niz — svaki sledeći član je veći od zbir svih prethodnih. Ako treba da pošaljemo binarnu poruku:

$D = (1, 0, 0, 1, 0, 1, 1, 1)$, možemo da je šifrujemo pomoću E kao skalarni proizvod dva vektora:

$S = E * D = 1+8+35+68+142 = 254$

Decodovanje je jednostavno: superrast omogućava da se od S oduzimaju redom, od većih ka manjim komponentama, članovi niza i kad je oduzimanje moguće (ako se oduzimanje ne smatra mogućim čim je razlika veća od nule) i dobije poruka D:

254 - 142 = 112 0 bit poruke 1
 112 - 68 = 44 0 bit poruke 1
 44 - 35 = 9 0 bit poruke 1
 9 - 17 = -8 0 bit poruke 0
 9 - 8 = 1 0 bit poruke 1



1 - 4 = -3 (0 bit poruke 0
 1 - 2 = -1 (0 bit poruke 0
 1 - 1 = 0=0 bit poruke 1
 i, istovremeno, kraj poruke.

Šifra u rancu

Naravno, E se ne koristi za javni protokol. Za ovaj protokol možete koristiti „hard knapsack“ (težak ranac). Da bi se generisao ovaj „ranac“ biraju se dva tajna prirodna broja M i W tako da je:

(1) M veće od zbira svih komponenti i vektora E.

(2) W i M uzajamno prosti. (Što znači da je W ** (-1) * W = 1 mod M)

Sledeći čuveni članak: R.C. Merkle and M.E. Hellman, „Hiding Information and Signatures in Trapdoor Knapsack“, „IEEE Trans. Information Theory, Septembar 1978, od 525 do 530 stranice, izaberimo M=291 i W=176 (što znači da je W ** (-1) = 167). Sada H generišemo iz E pomoću formule:

$H = W * E$ mod M, gde je malo j oznaka za mesto komponente. Tako imamo: $H = (176, 61, 122, 244, 82, 49, 37, 257)$.

U idealnom slučaju, H izgleda kao niz slučajnih brojeva, a superrast struktura originalnog E je skrivena. Da bi se dešifrovalo D, koristimo H:

$Sh = H * D = 763$. Ako je broj komponenti H veliki, recimo nekoliko stotina, tada se, čak i kad se znaju H i Sh, D ne može saznati u nekom razumnom vremenu. Onaj kome je poruka upućena iako izračunava D pomoću formule:

$Se = Sh * W^{**} (-1)$ mod M. Možda vam sve ovo izgleda pomalo komplikovano. U tom slučaju, probajte da koristite ovaj primer (ili, sledeći ideju primera, konstruišite jednostavniji) na svom računaru ili, čak, ručno i ideje će se razjasniti.

Na proste činioce

Pri igranju pokera telefonom treba biti oprezan i izbegavati protivnike koji imaju „kreja“ ili neki drugi računarski arhitekture i performansi. Naime, arhitektura ovog računara dozvoljava vrlo efikasno rastavljanje na proste činioce brojeva sa velikim brojem cifara, korišćenjem algoritma kvadratnog sita čiji je autor Karl Pomerans (Karl Pomerance) sa Džordžija univerziteta. Ovo rastavljanje, s druge strane, omogućava razbijanje i najtežih ranaca. Gustavs Simons (Gustavus Simmons), Džems Džejms (James Davis) i Džijana Holdridž (Diane Holdridge) iz Los Alamos Laboratorije su rastavljali redom brojeve od

55,58,60,63,67 i 69 cifara, za šta im je bilo potrebno vreme od 1,8 do 32 časa. Posle daljeg ubrzanja algoritma, za 6,45 časova rastavljen je 71 cifreni broj. Ako vas interesuje faktori broja $2^{**} 251 - 1$, to su 503, 54 217, 178 230 287 214 063 289 511, 61 676 882 198 695 257 501 367 i 12 070 396 178 249 893 039 969 681. Ovaj broj je ACM stavio na plaketu koju je poklonio IEEE povodom stogodišnjice ovog izuzetno cenjenog društva. Bilo je to davne 1984. godine. Rad na rastavljanju sve većih i većih brojeva se nastavlja.

Časkanja sa Enom

Ako niste montekarlovski nastrojeni pojedinač, moguće je da će vas više interesovati neki od sledećih problema:

1. Ena iz Zagreba i Mirko iz Beograda razgovaraju telefonom i reše da malo obnove svoja sećanja na lepe dane iz Portofoza. Za početak se dogovore da bacaju novčić. Kako kad jedno od njih baci novčić drugi bude siguran da ga — kao on kaže na koju je stranu pao novčić — onaj koji je bacao ne prevrati i kaže da je pao na suprotnu.

2. Ena u Zagrebu ima sef za koji Mirko zna šifru, a Mirko sef u Beogradu za koji Ena zna šifru. Kako da izmene šifre, a da svako od njih bude siguran da ga ono drugo neće prevariti (bar kad je u pitanju šifra za sef).

3. Ena ima sef čiju šifru, osim nje, znaju Mirko i Mario. Kako napraviti algoritam koji omogućava i Mirku i Mariju da pristupe sefu, ali tako da Ena uvek tačno zna i ko je koliko puta pristupao sefu. Mirko i Mario smeju da znaju samo za sebe a ne i za onog drugog.

4. Ena i Mirko treba da potpišu ugovor. Kako da naprave proceduru potpisivanja tako da Ena ne potpiše dok nije sigurna da je Mirko potpisao i Mirko ne potpiše dok nije siguran da je Ena potpisala.

5. Na kraju, jedan ozbiljni zadatak o naučnicima:

5. Trideset naučnika radi na tajnom projektu, ali dokumentaciji mogu da pristupe jedino ako ih je prisutno šesnaestorica ili više. Kako da naprave šifru na sefu sa dokumentacijom tako da je, ako ih je manje od 16, pristup nemoguć.

Ako su vam se zadaci svideli, sve je jasno — kriptografija je nauka za vas. Mi vam preporučujemo Kanovu knjigu (Kahn, D. The Codebreakers. New York: Macmillan). Ako vam to nije dovoljno, pišite nam i mi ćemo vas uputiti na dalju literaturu.

Miodrag Potkonjak

zadnjeg bajta po važnosti (prvog u memoriji koji se deli sa 256) / 4. Ovo se najbolje može razumeti iz jednog primera:

Broj 438.76 može se predstaviti u vidu proizvoda
 $0.856953125 \cdot 2 \wedge 9$

tako da se u petom bajtu upisuje karakteristika uvećana za 128, odnosno &89. Mantisa se može prikazati u obliku:

$219/256 + 97/256 \wedge 2 + 71/256 \wedge 3 + 174/256 \wedge 4$,

tako da bi sadržaj ostala četiri bajta (od četvrtog do prvog) bio: &DB, &61, &47 i &AE. Međutim, kako je usvojeno da mantisa bude u granicama između 0.5 i 1.0, tako da je sedmi bajt četvrtog bajta uvek jednak jedinici, to se ona može izostaviti, a na njeno mesto staviti predznak broja — 0 za pozitivne, a 1 za negativne brojeve.

Uzimajući u obzir sve ovo, broj 438.76 se u memoriji računara zapisuje u obliku:

&AE &47 &61 &5B &89

Negativna vrednost ovog broja, tj. —438.76, razlikovače se samo vrednošću četvrtog bajta:

&AE &47 &61 &DB &89.

PREMEŠTANJE REALNOG BROJA

&BD3D

Realan broj, definisan sa pet bajtova, koji se nalazi u memoriji računara, prepisuje se u novih pet bajtova na željenoj adresi.

Početni uslovi:

Registar DE sadrži adresu memorije gde se broj nalazi.

Registar HL sadrži adresu na koju se preslikava taj broj.

Krajnji uslovi:

Registar A sadrži eksponent (karakteristiku) broja uvećan za &80. Registri BC, DE i HL ostaju sačuvani.

Efekat:

U pet bajtova memorije, čija je adresa u registru HL, prenose se vrednosti pet bajtova sa adrese koja se nalazi u registru DE. Sadržaj izvornih pet bajtova ostaje neizmenjen.

SMEŠTANJE REALNOG BROJA

&BD40

Broj koji se nalazi u registru HL smešta se kao realan broj u memoriji na datoj adresi.

Početni uslovi:

Registar HL sadrži broj koji se unosi.

Registar DE sadrži adresu na koju se broj želi uneti.

Krajnji uslovi:

Registri HL i DE zamenjuju sadržaj, tako da registar HL umesto broja sadrži adresu gde je taj broj smešten.

Registar AF je zaprjan.

Efekat:

Broj iz registra HL, koji je predstavljen sa dva bajta, pretvara se u realan broj definisan sa pet bajtova, da bi se sa njim mogle vršiti matematičke operacije, i smešta u naznačenu memoriju.

PREUZIMANJE BROJA IZ MEMORIJE

&BD46

Registar HL preuzima realni broj koji se nalazi na datoj adresi u memoriji računara.

Početni uslovi:

Registar HL sadrži adresu gde se nalazi željeni broj.

Krajnji uslovi:

Registar HL sadrži celobrojni deo realnog broja iz memorije.

Registri BC i DE ostaju sačuvani, dok je registar A zaprjan.

Efekat:

Realni broj koji se nalazi u pet bajtova memorije čija se adresa nalazi u registru HL pretvara se u celobrojnu konstantu, koja se zatim smešta u registar HL. Sadržaj memorije gde se taj broj nalazio ostaje neizmenjen.

SABIRANJE DVA REALNA BROJA

&BD58

Vrednost jednog realnog broja iz memorije dodaje se drugom realnom broju.

Početni uslovi:

Registar DE sadrži adresu pet bajtova memorije u kojima se nalazi broj koji se želi dodati.

Registar HL sadrži adresu memorije gde se nalazi realan broj kome se dodaje prethodni broj.

„amstradov“ kalkulator

Pisanje iole složenijih programa u mašinskom jeziku ne može se ni zamisliti bez poznavanja rutina koje se nalaze ugrađene u ROM-u računara, kao i adresa pozivanja istih sa početnim i krajnjim uslovima koji moraju biti zadovoljeni. Na žalost, dodatak „SVE AMSTRADOVE RUTINE“ objavljen u „Računarima 10“ ne opravdava u potpunosti ovaj svoj naslov, jer ne sadrži rutine koje su neophodne pri matematičkim operacijama u raznim više ili manje složenim programima. Kako ni publikacija „CPC-464 FIRMWARE“, po kojoj je i rađen ovaj dodatak, ne sadrži njihov opis, nivo saradnik „Računara“ dr ing. Svetislav Zahar je proanalizirao donji ROM i otkrio rutine koje se ne pominju u „amstradovoj“ literaturi.

Koristeći program za disasembiranje „mašinka“ sopstvene izrade, autor je snimio i odštampeo celokupan donji ROM od adrese &0000 do &3FFF, kao i JUMPBLOCK između adresa &8900 i &BFFF. Kako se skok u donji ROM, gde se nalaze tražene rutine, vrši naredbama RESTARTA, tj. RST1 i RST5, bilo je logično da se adrese poziva nalaze u džamp-bloku sa rastojanjem od tri bajta između dve ulazne adrese. Kako su u pomenutoj publikaciji „CPC-464 FIRMWARE“ analizirane rutine do adrese &BD3A, pretpostavio sam da se tražene rutine nalaze iznad ovih adresa. Ova pretpostavka se pokazala tačnom, tako da sam pronašao ne samo pozivne adrese za većinu matematičkih rutina, već i potrebne polazne uslove.

Pošto smatram da ovo može zanimati veći broj korisnika mikroručunara AMSTRAD-SCHNEIDER CPC464, to u daljem tekstu dajem prikaz najvažnijih matematičkih rutina sa adresom poziva, početnim i krajnjim uslovima, kao i efektom koji se tom rutinom ostvaruje. Kako se od ovog saznanja došlo analiziranjem ROM-a, logično je da se simbolička, odnosno skraćena imena ne mogu dati, već su nazivi opisnog tipa.

Mađa je ovaj tekst namenjen prevasno opima koji su već donekle ušli u tajne mašinskog programiranja, neće biti suvišno da se prikaže način memorisanja realnih brojeva u memoriji računara. Za razliku od celih brojeva koji su predstavljani sa jednim, odnosno dva bajta, bilo u memoriji računara bilo u registrima, za prikazivanje realnih (decimalnih) brojeva koriste se pet bajta u memoriji, raspoređenih po svojoj važnosti u obrnutom redosledu.

Svaki realan broj se može prikazati u obliku proizvoda $M/2^N \cdot K$ gde je:

M — mantisa broja između 0.5 i 1.0, pozitivna ili negativna, a

K — karakteristika odnosno eksponent broja 2

Karakteristika broja uvećana za 128 (odnosno &80) smešta se u peti bajt, čok se ostala četiri koriste za predstavljanje mantisa. Pošto se ovde radi o brojevima manjim od jedinice, to se oni, za razliku od celih brojeva, dobijaju kad se vrednost sadržana u bajtu sa najvećom težinom (četvrti bajt), podeljena sa 256, sabere sa vrednošću susednog (trećeg) bajta podeljenom sa 256 / 4, itd. do

Krajnji uslovi:

Registar HL ostaje sačuvan — sadrži adresu rezultata.
Registri AF, BC i DE su zapriyani.

Efekat:

Realni broj koji je predstavljen sa pet bajta iz memorije čija se adresa nalazi u registru DE sabira se sa realnim brojem čija se adresa nalazi u registru HL. Rezultat se smešta umesto broja čija je adresa određena registrom HL, tako da sadržaj ovog registra na kraju operacije sadržava adresu rezultata.

ODUZIMANJE REALNIH BROJEVA

§BD5B

Sadržaj jednog realnog broja iz memorije se oduzima od drugog realnog broja, koji se takođe nalazi u memoriji računara.

Početni uslovi:

Registar DE sadrži adresu broja koji se oduzima.
Registar HL sadrži adresu broja od koga se oduzima prethodni broj.

Krajnji uslovi:

Registar HL sadrži adresu memorije gde je smešten rezultat.
Registri AF, BC i DE su zapriyani.

Efekat:

Realan broj koji je predstavljen sa pet bajtova smeštenim na adresi koja se nalazi u registru DE oduzima se od realnog broja čija je adresa data registrom HL, a rezultat se smešta umesto umanjitelja (sa adresom u HL). Sadržaj memorije gde se nalazio umanjilac ostaje neizmenjen.

ODUZIMANJE REALNIH BROJEVA

§BD5E

Kao i u prethodnom slučaju, ova rutina vrši oduzimanje dva realna broja, samo što je uloga registra DE i HL izmenjena.

Početni uslovi:

Registar HL sadrži adresu memorije gde se nalazi broj koji se oduzima.
Registar DE sadrži adresu memorije gde se nalazi broj od koga se vrši oduzimanje.

Krajnji uslovi:

Registar HL sadrži adresu rezultata.
Registri AF, BC i DE su zapriyani.

Efekat:

Realan broj koji se nalazi u pet bajtova memorije čija je adresa sadržana u registru HL oduzima se od realnog broja čija se adresa nalazi u registru DE. Rezultat se smešta u memoriji gde se ranije nalazio umanjilac. Sadržaj memorije gde se nalazio umanjitelj ostaje neizmenjen.

MNOŽENJE REALNIH BROJEVA

§BD61

Realan broj iz memorije množi se sa drugim realnim brojem, smeštenim takođe u memoriji računara.

Početni uslovi:

Registar HL sadrži adresu pet bajtova memorije gde je smešten broj kojim se množi.
Registar DE sadrži adresu memorije gde je smešten broj sa kojim se prvi broj množi.

Krajnji uslovi:

Registar HL sadrži adresu memorije gde je smešten rezultat.
Registri AF, BC i DE su zapriyani.

Efekat:

Realan broj koji je predstavljen sa pet bajtova memorije čija je adresa predstavljena sadržajem registra HL množi se sa brojem čija je adresa data u registru DE. Rezultat se smešta umesto prvog broja, čija adresa ostaje u registru HL. Sadržaj memorije drugog broja ostaje neizmenjen.

DELJENJE REALNIH BROJEVA

§BD64

Realan broj smešten u memoriji deli se sa drugim brojem koji se takođe nalazi u memoriji računara.

Početni uslovi:

Registar HL sadrži adresu broja koji se deli.
Registar DE sadrži adresu broja sa kojim se prvi broj deli.

Krajnji uslovi:

Registar HL sadrži adresu memorije gde je smešten rezultat.
Registri AF, BC i DE su zapriyani.

Efekat:

Realan broj koji se nalazi u pet bajtova memorije čija je adresa definisana vrednošću registra HL deli se sa brojem iz memorije čija se adresa nalazi u registru DE. Rezultat se smešta umesto delitelja, dok sadržaj memorije gde se nalazio delilac ostaje neizmenjen.

PROMENA ZNAKA

§BD6D

Realnom broju koji se nalazi u memoriji menja se predznak.

Početni uslovi:

Registar HL sadrži adresu memorije gde je smešten broj.

Krajnji uslovi:

Svi registri osim registra A ostaju sačuvani.

Efekat:

Realnom broju koji se nalazi u pet bajtova memorije čija se adresa nalazi u registru HL menja se predznak, tj. pozitivni broj se pretvara u negativan i obratno. Adresa tog broja i dalje ostaje u registru HL.

ISPITIVANJE ZNAKA

§BD70

Rutina vrši ispitivanje broja u memoriji i vraća predznak broja.

Početni uslovi:

Registar HL sadrži adresu gde se broj nalazi.

Krajnji uslovi:

Registar HL sadrži adresu gde se broj nalazi.

Krajnji uslovi:

Registar A sadrži kod predznaka i to:
A=1 ako je broj u memoriji pozitivan;
A=-1 ako je broj u memoriji negativan;
A=0 ako je broj u memoriji jednak nuli.
Svi ostali registri ostaju sačuvani.

Efekat:

Ispitivanje realnog broja koji je smešten u pet bajtova memorije čija se adresa nalazi u registru HL i vraćanje koda predznaka tog broja. Sadržaj memorije ostaje neizmenjen.

KONSTANTA PI

§BD76

Smeštanje konstante PI na željeno mesto u memoriji.

Početni uslovi:

Registar HL sadrži adresu pet bajtova memorije gde se želi smestiti konstanta PI.

Krajnji uslovi:

Registri BC i HL ostaju sačuvani.
Registar A sadrži eksponent konstante urečan za §80.
Registar DE je zapriyan (sadrži adresu ROM-a gde je konstanta PI smeštena).

Efekat:

Prenosi se vrednost konstante PI iz ROM-a u pet bajtova memorije čija je adresa data u registru HL, da bi se mogla koristiti pri računanju.

KVADRATNI KOREN

§BD79

Nalaženje vrednosti kvadratnog korena realnog broja koji se nalazi u memoriji računara.

Početni uslovi:

Registar HL sadrži adresu pet bajtova memorije gde je smešten broj čiji se kvadratni koren traži.

Krajnji uslovi:

Registar HL sadrži adresu rezultata.
Ostali registri su zapriyani.

Efekat:
Izračunavanje vrednosti kvadratnog korena realnog broja koji se nalazi u pet bajtova memorije čija je adresa smeštena u registru HL i smeštanje rezultata umesto datog broja.

PRIRODNI LOGARITAM

&BD7F

Nalaženje vrednosti prirodnog logaritma realnog broja koji se nalazi u memoriji računara.

Početni uslovi:

Registar HL sadrži adresu gde se nalazi realni broj čiji se logaritam traži.

Krajnji uslovi:

Registar HL sadrži adresu memorije gde je smešten rezultat.

Svi ostali registri su zapriyani.

Efekat:

Izračunavanje vrednosti prirodnog logaritma (logaritma za osnovu „e“) realnog broja koji je smešten u pet bajtova memorije čija se adresa nalazi u registru HL. Rezultat se smešta na isto mesto.

DEKADNI LOGARITAM

&BD82

Nalaženje vrednosti dekadnog logaritma realnog broja koji se nalazi u memoriji računara.

Početni uslovi:

Registar HL sadrži adresu broja čiji se logaritam traži.

Krajnji uslovi:

Registar HL sadrži adresu gde je smešten rezultat.

Svi ostali registri su zapriyani.

Efekat:

Izračunavanje vrednosti logaritma za osnovu 10 realnog broja koji je smešten u pet bajtova memorije čija se adresa nalazi u registru HL. Rezultat se smešta umesto datog broja.

EKSPONENCIJALNA FUNKCIJA

&BD85

Nalaženje vrednosti eksponencijalne funkcije realnog broja koji se nalazi u memoriji računara.

Početni uslovi:

Registar HL sadrži adresu gde se nalazi realan broj.

Krajnji uslovi:

Registar HL sadrži adresu pet bajtova u memoriji gde se nalazi rezultat. Svi ostali registri su zapriyani.

Efekat:

Izračunavanje vrednosti eksponencijalne funkcije e^x , gde je X realan broj smešten u pet bajtova memorije čija je adresa data u registru HL. Rezultat se smešta umesto datog broja.

SINUSNA FUNKCIJA

&BD88

Nalaženje vrednosti funkcije SINUS iz realnog broja koji je smešten u memoriju računara.

Početni uslovi:

Registar HL sadrži adresu realnog broja.

Krajnji uslovi:

Registar HL sadrži adresu gde je smešten rezultat.

Svi ostali registri su zapriyani.

Efekat:

Izračunavanje vrednosti sinusne funkcije realnog broja koji je smešten u pet bajtova memorije čija se adresa nalazi u registru HL. Rezultat se smešta umesto datog realnog broja.

COSINUSNA FUNKCIJA

&BD8B

Nalaženje vrednosti funkcije COSINUS realnog broja koji se nalazi u memoriji računara.

Početni uslovi:

Registar HL sadrži adresu memorije gde je smešten dati broj.

Krajnji uslovi:
Registar HL sadrži adresu memorije gde je smešten rezultat.
Ostali registri su zapriyani.

FUNKCIJA TANGES

&BD8E

Nalaženje vrednosti funkcije TANGES realnog broja koji se nalazi u memoriji računara.

Početni uslovi:

Registar HL sadrži adresu memorije gde je smešten dati broj.

Krajnji uslovi:

Registar HL sadrži adresu memorije gde je smešten rezultat.

Ostali registri su zapriyani.

Efekat:

Izračunavanje vrednosti tangensa realnog broja koji je smešten u pet bajtova memorije čija se adresa nalazi u registru HL. Rezultat se smešta umesto datog broja.

INVERZNA FUNKCIJA TANGESU

&BD91

Nalaženje vrednosti funkcije ARCUS TANGES datog realnog broja koji se nalazi u memoriji računara.

Početni uslovi:

Registar HL sadrži adresu memorije gde je smešten dati broj.

Krajnji uslovi:

Registar HL sadrži adresu memorije gde je smešten rezultat.

Ostali registri su zapriyani.

Efekat:

Izračunavanje vrednosti funkcije arcus tanges realnog broja koji je smešten u pet bajtova memorije čija se adresa nalazi u registru HL, pri čemu se rezultat takođe smešta umesto datog broja.

FUNKCIJA RND

&BD9D

Generisanje vrednosti slučajnog broja.

Početni uslovi:

Registar HL sadrži adresu pet bajtova memorije gde se smešta rezultat.

Krajnji uslovi:

Registar HL je sačuvan, dok su ostali zapriyani.

Efekat:

Iz generatora slučajnih brojeva preuzima se vrednost slučajnog broja i smešta u pet bajtova memorije čija je adresa sadržana u registru HL.

PROMENA PREDZNAKA

&BDC7

Menjanje predznaka celog broja koji se nalazi u registru HL.

Početni uslovi:

Registar HL sadrži vrednost broja kome se menja predznak.

Krajnji uslovi:

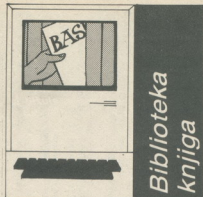
Registar HL sadrži vrednost broja sa izmenjenim predznakom.

Registri BC i DE ostaju neizmenjeni, dok je registar A zapriyani.

Efekat:

Celobrojna veličina koja se nalazi u registru HL zamenjuje se svojom negativnim vrednošću.

Svetislav Zahar



Biblioteka
knjiga

Primjena osobnih računala u poduzećima

Izdavač: „Informator“ Zagreb, grupa autora: H. Birolia, M. Habek, A. Kliment, K. Kliment, A. Pađen i D. Ugarković

Budući da se sve više smanjuju granice između kućnih računala i onih što spadaju u klasu Personal Computers-a, dijelu čitaateljstva koje zanima ova problematika dobro će doći knjiga grupe autora „Primjena osobnih kompjutera u radnim organizacijama“. U redakciji prof. dr. Antuna Klimenta zagrebački „informator“ potkraj prošle godine tiskao je jednu od rijetkih teoretsko-praktičnih knjiga s tom tematikom u nas, a pošto automatizirano upravljanje poslovnim procesima u uredskom poslovanju jeste preduvjet uspješne proizvodnje te plasmana roba i usluga, ovo štivo može pomoći upravljačkim strukturama da se upoznaju s osnovama informatike.

Knjiga sadrži osam poglavlja: Mjesto i uloga osobnih kompjutera u poslovanju poduzeća. Proizvođači osobnih kompjutera. Arhitektura osobnih kompjutera. Software-ska podrška. Programiranje na osobnom kompjuteru. Tehnologija obrade poslovnog teksta. Tehnologija obrade podataka i Poslovni informacijski sustav.

U prvom poglavlju autori razmatraju značaj automatizacije u poslovanju poduzeća, razvoj i primjenu kompjutera. Već tu naglašavaju razliku između već klasične obrade podataka na velikim sistemima i primjenu malih računala u poslovanju poduzeća. Ako su potrebne dodatne informacije na primer, korisnici PC-a su u prednosti jer mogu jednostavno komunicirati s računalom bilo da su na radnom mjestu, u kući ili na putu.

Drugo poglavlje obuhvaća pregled nekih domaćih i inozemnih proizvođača računala. Najviše pažnje autori su poklonili „Iskrino“ modelu „Partner“ i PC-u M-11 što ga proizvodi zagrebački Infosistem-Tehničar.

Ograničavanje domaće proizvodnje na samo dva modela, vjerojatno iz komercijalnih razloga ili žurbe, jedno je i najveća mana u dosta opširnog izlaganju (knjiga ima 236 stranica). Od domaćih proizvođača spomenuti su PEL, „Ivo Lola Ribar“, El Niš, TRS Zagreb, „Velebit“, Novkabei, Infosistem-tehničar i Iskra-Delta. Izostavljeni su „Školski servis“ Zagreb, Ivasim-elektronika, Informatika i elektronika „Javor“ Bitola, „Jozef Stefan“ Ljubljana, Elektronika-inženjering Zemun, „Medimurjepl“ Čakovec kao kooperant RIZ — „Nixdorf“, „Micro“ Zagreb i „ZEMA“ Austrija, UNIS i drugi.

Od inozemnih proizvođača prikazani su IBM i Apple.

Arhitektura osobnih računala detaljno je opisana, a posebni prilog u koji može pripomoći onima koji nisu nikada videli računalo, štampac, diskete i drugo.

Četvrtio poglavlje donosi informacije o softverskoj podršci osobnih računala. Softver je podijeljen u dvije grupe: u aplikativni i sistemski softver. Sistemski je pak raščlanjen na operacijske sustave, prevodioce simboličkih programskih jezika i pomoćne programe. Aplikativni softver podijeljen je na onaj koji razvija i programira sam korisnik i aplikativne programe drugih autora koje korisnik kupuje. Obradeni su i programski jezici ADA, ALGOL, APL, BASIC i sedam drugih programskih jezika, uz dva pomoćna programa EDITOR i SORT.

U petom poglavlju obradeno je programiranje na osobnom kompjuteru. Potanko je razrađen način programiranja u jeziku, kao najčešće korištenom programskom jeziku.

Za poslovne ljude i sve one koji žele računalo koristiti za obradu teksta naročito će biti korisno često poglavlje gdje je iznesena tehnologija obrade poslovnog teksta. Autori se ne bave samo načinima formiranja teksta već donose i obrasce kao se može kreirati poslovna komunikacija, kako se mogu obraditi pomoću kompjutera masovne poslovne komunikacije s naglaskom na komercijalnim poslovnim informacijama. Nastavno, u sedmom poglavlju o tehnologiji obrade podataka objašnjava se primjena „personal computinga“ na modelu IBM-a.

Poslovni informacijski sustav tema je osmog poglavlja, gdje se kibernetičkim pristupom nude rješenja za komuniciranje u poslovnim informacijskim sustavima. Pri tome su se autori pridržavali postojećih zakonskih normi i standarda tako da je model izrade poslovnih informacija primjenljiv u privedi.

Kućni ili hobi kompjutiri danas su sredstvo za rad, za igre i učenje i bit će sredstvo komuniciranja krajem ovog stoljeća. Osobni kompjutiri preuzeli su od kućnih hobi igre, pa se osim za posao PC može koristiti i za zabavu u slobodno vrijeme. U nas je, doduše, odviše velika razlika u cijeni između PC-a i malog računala, ali tako neće vječno i ostati. Stoga „Primjena osobnih kompjutera u radnim organizacijama“ predstavlja još jedan vrijedan pokušaj da se proširi informatičko znanje i da se u poduzeća uključe u svjetski proces primjene malih i poslovnih računala.

Knjiga je tiskana u nakladi od 2.000 primjeraka u već provjerenoj „Informatorovoj“ standardnoj opremi.

B. Hebrang

operacija FILL

Prvi susret sa naredbom FILL vlasnici „spektruma“ su imali kada se pojavio BETA BASIC. Među brojnim naredbama i funkcijama, koje donosi ovo proširenje jezika, našla se tu i operacija za popunjavanje zatvorenih slika, nazvana jednostavno FILL. Na ulazu, naredba zahteva dva parametra: y i y koordinatu tačke od koje počinje popunjavanje. U rezultatu, čitavo polje oko zadane tačke biće popunjeno, sve dok se ne dostigne granica oblasti, ili ivica ekrana.

Kako to radi beta bezik . . .

Reč popunjavanje ovde ima sasvim konkretno značenje: *setovanje odgovarajućih bitova u video memoriji*, na isti način kako to radi naredba PLOT.

S druge strane, imamo operaciju brisanja popunjenih figura, što bi se iz jezika moglo izvesti uzastopnom primenom naredbe PLOT OVER 1; . . . BETA BASIC u tu svrhu koristi takođe naredbu FILL, ali sa atributom PAPER (obična FILL naredba, inače, podrazumeva FILL INK).

Ako želimo, na primer, da na svetoj plavoj pozadini ekrana imamo veliki crveni krug sa manjim zelenim krugom u unutrašnjosti, ukucaćemo, recimo, ovakav program:

```
10 PAPER 5: INK 2: CLS
20 CIRCLE 5:128,88,80
30 FILL 128,88
40 CIRCLE PAPER 4: OVER 1: 160,88,30
50 FILL PAPER 4: 160,88
60 PAPER 7: INK 0
70 STOP
```

U liniji 10 postavljaju se plava boja pozadine i crvena boja za tekst. Zatim se crta i popunjava crvena kružnica (linije 20 i 30). Zelenom bojom podloge crta se manja kružnica (linija 40), a njena unutrašnjost se zatim briše FILL naredbom u liniji 50. Pre završetka programa, boje se vraćaju na uobičajenu crno-belu kombinaciju.

. . . i kako bismo to mogli sami

Program koji simulira naredbu FILL mogao bi se, svakako, izvesti i iz jezika, ali bi to bilo neprihvatljivo sporo rešenje. Jedini izlaz je, očigledno, sastaviti odgovarajuću mašinsku rutinu i nekako je povezati sa bezikom, da bi korišćenje nove operacije bilo, koliko-toliko, jednostavno.

Pred nama su, prema tome, dva problema: sastaviti algoritam i mašinski program za operaciju FILL i povezati taj program sa bezikom na prihvatljiv način, pri čemu u prihvatljive načine ne ubrajamo prenošenje parametara PEEK-POKE metodom . . .

Pre svega, moramo videti koliko ulaznih parametara program treba da ima. Dva su obavezna (x i y koordinata polazne tačke na ekranu), a što se tiče izbora boje, najjednostavnije je zadati samo parametar koji definiše FILL PAPER ili FILL INK, dok se sama boja može određivati i unapred, standardnim naredbama PAPER i INK.

Majstorije na računaru/spektrum

Za Sinklerov bežik ima uvek argumenata i za i protiv. Međutim, izvesno je da taj jezik ne podržava u adekvatnoj meri grafičke mogućnosti računara, koje, uprkos ograničenju u radu sa bojama (najviše dve boje u istom znakovnom polju), nipošto nisu beznačajne: rezolucija 256x192 predstavlja i te kako velik potencijal — samo ga treba na pravi način iskoristiti. U tom cilju, svakako, moramo raspolagati širim izborom naredbi. Instrukcije: PLOT, DRAW i CIRCLE, koje „spektrum“ jedino raspoznaje, mogu služiti tu samo kao osnova za izgradnju potpunijeg sistema. U ovom tekstu ćemo posvetiti pažnju naredbi FILL.

Odlučili smo se da ulazne parametre prosljedujemo metodom DEF FN funkcija, o čemu smo u „Računarima“ već više puta pisali. Recimo, ako mašinski program FILL počinje na adresi 50000, ukucamo u bežiku liniju:

```
10 DEF FN f(x, y, z)=USR 50000
```

Prva dva parametra funkcije FN f predstavljaju x i y koordinatu početne tačke, a treći parametar može uzeti vrednost 0, u slučaju FILL PAPER, ili 1, u slučaju FILL INK. Tako, na primer, ako želimo da popunimo (FILL INK) neku figuru sa početnom tačkom (100,50,1), izvršićemo:

```
RANDOMIZE FN f(100,50,1)
```

```
Istu figuru izbrisati ćemo sa:
```

```
RANDOMIZE FN f(100,50,0)
```

Što se tiče onog primera u BETA BASIC-u, sa crvenim i zelenim krugom, sada bismo ga mogli rešiti i ovako:

```
10 DEF FN f(x,y,z)=USR 50000
```

```
20 PAPER 5: INK 2: CLS
```

```
30 CIRCLE 128,88,80
```

```
40 RANDOMIZE FN f(128,88,1)
```

```
50 PAPER 4
```

```
60 CIRCLE PAPER 4: OVER 1:160,88,30
```

```
70 RANDOMIZE FN f(160,88,0)
```

```
80 PAPER 7: INK 0
```

```
90 STOP
```

Očigledno, nikakvih suštinskih izmena nema, osim što umesto FILL PAPER 4... moramo neposredno izvršiti PAPER 4 ispred naše FILL instrukcije.

Ono što sada preostaje da se uradi, jeste sastaviti program FILL koji će raditi ovako kako smo opisali.

Jedan loš... Nismo se upuštali u analizu algoritma koji koristi BETA BASIC, ali se već na prvi pogled čini da on radi prilično čudno: kad jednom popuni celu figuru, program provodi iznenađujuće dugo vreme proveravajući da li je obavio sve kako treba, ili je možda negde preskočio neku „rupu“ (!?). Ako među našim čitaocima ima takvih koji su ovaj program analizirali, pozivamo ih da nam se jave i objasne u čemu je stvar.

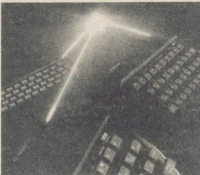
Mi ćemo, međutim, predložiti jedan drugačiji algoritam, koji je dovoljno jasan i prihvatljiv, da ga ovde možemo detaljno opisati. Poći ćemo, prve svega, od formulacije samog problema.

Data je neka oblast ograničena proizvoljnom krivom linijom ili livicom ekrana. Date se koordinate (x,y) proizvoljne tačke u toj oblasti. Zadatak je popuniti sve tačke obuhvaćene datom figurom, polazeći, pri tome, od zadate tačke (x,y).

Ograničeno se sada samo na FILL INK, mada je sve potpuno isto i u slučaju FILL PAPER, s tom razlikom što je oblast, zapravo, već popunjena, a granicu čine prazna polja. Tu je zadatak *obrisati* sve

tačke u okolini zadate, sve dok se ne dosegne granica.

Osnovna teškoća u svemu ovome je kako pronaći sve tačke unutar zatvorene oblasti.



Zbrka u matrici

U tekstu „Matrice i nizovi“, objavljenom u „Računarima 11“, došlo je do niza oštri tehničke prirode, zbog kojih dugujemo izvinyenje našim čitaocima.

Kada je broj već bio izašao iz štampa, ispostavilo se da smo jedan od priloženih programa stoji na svom prirodnom mestu u tekstu. Svi ostali su razbacani bez ikakvog reda, pa je čitalac u situaciji da traži naokolo šta mu treba... Srećna okolnost je što svaki primer ima ime u prvom REM liniji, tako da je traženje olakšano. Manjih problema će jedino biti sa potprogramom za sistem jednačina N x N, koji je, jednim sjajnim potezom makazama, rasečen na dva dela. Prvo čete naći na deo od linije 3050, a tek pri kraju teksta na ono što tome prethodi... I konačno, potpuno je ispušten program 2000 za računanje determinante N x N. Redakcija se izvinyenja čitaocima i autoru teksta.

Jedan vrlo jednostavan postupak bio bi sledeći:

Kada god se otkrije nova tačka, ispita se i osam tačaka koje je okružuju. Svaka prazna od tih osam predstavlja novu tačku oblasti, i njene koordinate se mogu upamtiti. Postupak se ponavlja tako što se stalno uzima poslednja upamćena tačka i vrši testiranje osam okolnih. Sve će tačke biti

pronađene, ali po cenu ogromnog utroška memorije: praktično će se u isto vreme čuvati koordinate svih tačaka oblasti!

Ođ ovakvog algoritma, jasno, treba odustati. Mi smo ga pomenuli jedino zato što se iz njega vidi koliko postavka problema utiče na postupak rešavanja. Jer, sve vreme, govoreći o naredbi FILL, mi smo imali na umu popunjavanje tačku po tačku. A ako malo razmislimo, viđemo da je sasvim izvodljivo i popunjavanje liniju po liniju, horizontalno ili vertikalno.

... i jedan dobar algoritam

Tako dolazimo na pravi algoritam:

Ođ zadate tačke povući ćemo levo i desno horizontalnu liniju do granice oblasti. Upamtiti ćemo koordinate početne tačke linije i ukupan broj tačaka u liniji (dužinu).

U ciklusu, zatim, uzimamo poslednju upamćenu liniju, krećemo se duž nje tačku po tačku i gledamo stanje ispod i iznad nje. Kad god otkrijemo prazninu, povlačimo horizontalnu liniju i pamtimo njen početak i dužinu.

Postupak se ponavlja sve dok stek na kome čuvamo linije ne ostane prazan. Utrošak memorije je minimalan, jer se na steku istovremeno nalaze jedna do dve linije, ukoliko se radi o jednostavnoj oblasti, bez mnogo krivina i „zaliva“. Kod složenijih oblasti, broj linija koje se čuvaju opet ne može biti veći od broja „zaliva“.

Ostaje nam, sada, da ovaj opis algoritma prevedemo u mašinske naredbe.

Objavljujemo kompletan program sa komentarima, namenjen onima koji iz svega ovoga žele nešto da nauče. Čitaoci koje više interesuje sama primena programa, treba jednostavno da učitaju assembler (npr. DEVPAC GENS) i onda pažljivo ukucuju svaku liniju iz priloženog listinga, preskačući, naravno, komentare. Instrukcija ORG se može, po volji, menjati, ali se tada mora korigovati i adresa iz funkcije USR u odgovarajućoj DEF FN naredbi u okviru bežika.

Posle asemliranja program zauzima 179 bajtova. Treba ga snimiti na kasetu naredbom SAVE „FILL“ CODE 50000,179, i kasnije ga učitati sa LOAD ""CODE.

Program FILL, u poređenju sa odgovarajućom naredbom BETA BASIC-a, radi nešto sličnije kada su u pitanju manje figure. Recimo, u slučaju kruga radijusa 70, izvršenje traje oko 6s, prema 5s, koliko to radi BETA BASIC. Međutim, pri popunjavanju većih polja, BETA BASIC daleko zaostaje. Umesto 34 s, koliko je potrebno BETA BASIC-u za popunjavanje čitavog ekrana, programu FILL je dovoljno samo 18 s.

Ostale mogućnosti oba programa su međusobno jednake. Ograničenje jedino stvara toša „spektruma“ rezolucija u boji, tako da bojenje bliskih figura može da nam pričinjava probleme.

Jovan Skuljan

Program: FILL

Autor: Jovan Skuljan

J. Definicija simboličkih adresa. Prvih šest se odnosi na rutine iz ROM-a, a preostale tri su standardne sistemske promenljive. Prvi bajt sistemske promenljive MEMBOT (&5C92) program FILL koristi kao privremenu memoriju lokaciju, što ne ometa rad računara, jer kalkulator ne koristi memoriju van bezjick naredbe u kojoj je pozvan.

```
FINDI1 EQU 81E94 Rutina FIND_INT1
TEST_R EQU 81F05 Rutina TEST_ROOM.
P_ADDR EQU 822B1 Izlaz u rutinu PIXEL_ADD.
GET_BC EQU 82307 Rutina STK_TO_BC.
REP_B EQU 824F9 REPORT_B: Inlager out of range.
STKNUM EQU 833B4 Rutina STACK_NUM.
DEFADD EQU 85C0B Sistemska promenljiva.
ATTR_P EQU 85C8D Sistemska promenljiva.
MEMBOT EQU 85C92 Sistemska promenljiva.
```

2. Početak. Prikupljanje ulaznih parametara. Koordinate x i y će direktno biti unete u registre C i B, a indikator PAPER/INK u varijablu MEMBOT. Pripremljen za kasniju upotrebu od strane potprograma PLOT. Na mašinski stek se postavlja oznaka za dno steka, povlači se početna linija i ulazi se u glavnu petlju.

```
ORG 50000
LD HL, (DEFADD) U HL se uzima adresa prvog parametra
INC HL funkcije FN f i pomera se na prvi bajt
INC HL binarnog zapisa.
CALL STKNUM X-kordinata ide na računski stek.
INC HL Preskoči zarez.
INC HL i pomeri se na početak drugog
INC HL parametra funkcije FN f.
CALL STKNUM Y-kordinata ide na računski stek.
INC HL Preskoči zarez.
INC HL i pomeri se na početak trećeg
INC HL parametra funkcije FN f.
CALL STKNUM Indikator PAPER/INK ide na stek,
CALL FINDI1 a ostale odmah u akumulator.
CP 502 Da li je veći od jedan?
JP NC, REP_B Ako jeste, to je greška.
DEC A Posle dve jednostavne operacije, ind-
XOR 601 kator postaje 601 za INK, a 8FE za PAPER.
LD (MEMBOT),A Sačuvaj indikator za kasnije.
CALL GET BC X-kordinata u C, a Y u B.
LD A,175 Maksimalna vrednost za Y.
SUB B Da li je 'Y' van opsega?
JP C,REP_B Ako jeste, priavi grešku.
EXX Pradi u alternativni set registara.
PUSH HL Sačuvaj HL' za povratak u bezjick.
LD BC,60000 Pripremi BC' za kasnije.
EXX Vрати se u osnovni set registara.
LD HL,RETADD Adresa RETADD biće iskoristena za
PUSH HL povratak u bezjick.
LD LD invertovano Y, tj. 175-Y, ide nazad u B.
LD LD HAFF Signal za dno mašinskog steka: nijedna
DEC SP tačka ekrana nema Y=255. SP se smanjuje
PUSH HL jer stek treba da sadrži tri bajta.
CALL DRAW Iscrtaј početnu liniju, ako je moguće,
i postavi na stek njene parametre.
```

3. Glavna petlja za ispitivanje horizontalnih linija upamćenih na mašinskom steku. Tri poslednja bajta na steku određuju koordinate početne tačke i dužinu linije. Ali, ako je Y=255, to je signal da je operacija FILL privedena kraju.

```
EACHL1 POP BC Koordinate početne tačke ide u BC.
DEC SP Stek sadrži tri bajta.
POP DE Dužina linije ide u D.
```

4. Petlja za ispitivanje svake tačke linije. Gleda se ispod i iznad, i povlači nova linija, ako je potrebno.

```
EACHLP INC B Pomeri se na liniju ispod tekuće.
RET Z ali, ako je to bilo dno steka, vрати se.
LD A,B Testiraj novu Y koordinatu.
SUB 176 Da li je ispod dna ekrana?
CALL NZ,DRAW Pozovi DRAW samo za Y u okviru ekrana.
DEC B Vрати se na tekuću liniju.
JR Z,NEXT_C Ako je Y=0, onda nema linije iznad.
```

```
DEC B Pređi na liniju iznad tekuće.
CALL DRAW Povuci liniju, ako treba.
INC B Vрати se na tekuću liniju.
NEXT_C INC C Pređi na sledeću tačku tekuće linije.
DEC D Umanji brojčanu dužinu.
JR NZ,EACHP Ako ima još tačaka, vрати se u petlju.
JR EACHL Vрати se i uzmi sledeću liniju.
```

5. Povratak u bezjick. Ispred RET naredbe obnavlja se sadržaj HL'.

```
RETADD POP HL Obnovi HL'.
EXX Zameni registre.
RET Vрати se u bezjick.
```

6. Potprogram DRAW proverava tačku sa koordinatama BC i crta liniju levo i desno, samo ako je tačka prazna (slučaj FILL INK), ili samo ako je popunjena (slučaj FILL PAPER). Parametri nove linije (početna koordinata i dužina) stavljaju se na mašinski stek u vidu tri bajta: u vrhu je X, odmah ispod Y, a u dnu dužina.

```
DRAW LD H,00 Brojč dužine linije stavi na nulu.
CALL PLOT Nacrtaj tačku, ako je moguće.
RET NC Vрати se ako je tačka već postojala.
LD A,C Sačuvaj X privremeno.
LD E,C Traži levu granicu.
FIND_S AND A Da li je to leva ivica ekrana?
JR Z,EXIT_S Ako jeste, idi napred, na izlaz.
DEC C Umanji X za jedinicu.
CALL PLOT Crtaј tačku, ako je moguće.
JR C,FIND_S Ako granica nije nađena, traži ponovo.
INC C X-kordinata početne tačke.
EXIT_S LD L,C Prenesi X u registar L.
LD C,E Obnovi polaznu vrednost X.
FIND_E INC C Traži desnu granicu.
JR Z,EXIT_E Idi napred ako je iza ivice ekrana.
CALL PLOT Crtaј tačku, ako je moguće.
JR C,FIND_E Ako granica nije nađena, traži ponovo.
EXIT_E LD C,L Vрати X-kordinatu u C.
EXX Sačuvaj sve registre.
CALL TEST_R Testiraj memoriju (BC'=60000).
EXX Obnovi registre.
EX (SP),HL Razmeni dužinu linije i povratnu adresu.
INC SP Dužina linije je samo jedan bajt.
PUSH BC Početna tačka ide na stek.
LD C,E Obnovi polaznu X-kordinatu.
JP (HL) Vрати se iz potprograma.
```

7. Potprogram PLOT popunjava ili prazni tačku sa koordinatama BC, i uvećava pri tome brojč tačaka H. Takođe modifikuje odgovarajući atribut za boju i vraća se sa setovanim indikatorom prenosa („C'=1). Međutim, ako je tačka već bila na svom mestu, ništa se ne preduzima, i povratak se vrši sa „C'=0.

```
PLOT PUSH HL Sačuvaj HL.
PUSH DE Sačuvaj DE.
LD A,B Uzmi Y-kordinatu.
CALL P_ADDR Nađ adresu bajta u video memoriji.
LD E,B Sačuvaj Y privremeno.
LD B,A Položaj bajta u bajtu.
INC B Priprema brojača.
LD A,(MEMBOT) Indikator PAPER/INK (8FE/801)
ROT R,CA Priprema maske uzastopnom rotacijom
DINZ ROT udesno potreban broj puta.
LD B,E Obnovi Y-kordinatu.
LD E,A Sačuvaj masku privremeno.
OR (HL) Setuj zadati bit u video bajtu.
BIT 7,(Y+88) 'estiraj indikator u MEMBOT.
JR Z,TEST Idi napred ako je u pitanju FILL INK.
LD A,E Obnovi masku.
AND (HL) Resetuj zadati bit u video bajtu.
TEST CP (HL) Da li je tačka već bila prisutna?
JR Z,FINISH Ako jeste, idi napred.
PUSH AF Sačuvaj indikatore.
LD (HL),A Postavi video bajt na svoje mesto.
LD A,H Nizom operacija
RRCRA izračunaj adresu
RRCRA odgovarajućeg
RRCRA atributa za boju.
AND 603
OR 85B
LD H,A
LD A,(ATTR_P)
LD (HL),A
POP AF
POP DE
POP HL
RET Z
INC H
SCF
RET Vрати se iz potprograma.
```

Majstorije
na računaru
amstrad

kraljevstvo za registar

Svakom mašinskom programeru se (ne jedanput) otelo nešto slično naslovu ovog članka. Nedostatak registara je većita patnja svih koji su osuđeni na osmoblitne mikroprocesore, a žele da se što više približe mašini na kojoj rade. I pored prave navale 16-bitnih procesora, koji su sve jeftiniji i kod kojih ima mnogo registara, njihovi osmoblitni preci još nisu rekli poslednju reč. Još uvek se sva sila stvari može na njima uraditi na sasvim zadovoljavajući način i još će mnogo vode, nafte i svega ostalog preteći našom Savom, a oni neće sići sa scene. Tako nešto potvrđuje i pojava računara „amstrad“. Na njemu je sve staro bar po pet godina, a nema savremenije osmoblitnog personalca od njega. Ni procesor nije izuzetak — koristio ga je još TRS 80. Nevolja je jedino u tome što je upotreba alternativnog seta registara na „amstradu“ potpuno zabranjena. Nema, međutim, toga čemu pravi programer ne može da doskoči, pa čak i ovakvim ograničenjima.

Može se slobodno reći da je kod nas najpopularniji mikroprocesor Z80. Tu popularnost su mu, pre svih, doneli Sinkleriovi računari ZX81 i ZX82 (pardon, „ZX spektum“). Većina domaćih hakera je odnegovana upravo na starom dobrom „spektumu“. A hakeri najviše vole dobar asembler i, po mogućstvu, da je gluvo doba. Na nesreću, programeri koji rade na mašinskom jeziku osmoblitnih mašina su, upravo zbog nedostatka registara, pravi siromasi, i njihovo vreme je najjeftinije, ako znate šta se pod tim podrazumeva. Prava drama, međutim, nastaje kada tvorci operativnog sistema zabrane upotrebu određenih registara.

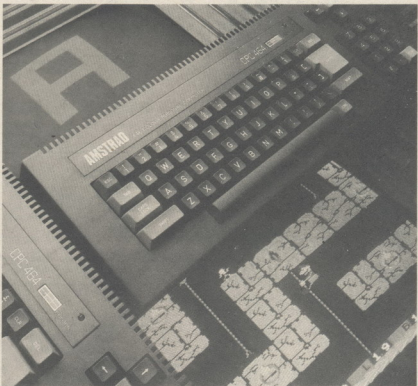
Amstradovi računari koriste Z80. On u odnosu na 6502, koji radi sa samo tri registra, ima svu silu registara: u svakom trenutku dostupno je 7 osmoblitnih registara opšte namene (A, BC, DE i HL), dva šesnaesoblitna indeksa registra (IX i IY) i redovni statusni registar (F), brojač naredbi (PC) i pokazivač steka (SP). Pored ovoga, postoje i alternativni registarski set i dva flip-flopa koji određuju koji set registara se u datom trenutku koristi. Logično, ti flip-floповi se kontrolišu pomoću dve naredbe:

- 1) EXX — za registre BC, DE i HL
- 2) EX AF, AF' — za akumulator i statusni registar.

Oba registarska seta mikroprocesor tretira na potpuno isti način. Među njima jedino čovek (programer) pravi razliku i trebalo bi da u svakom trenutku zna koji od setova koristi, pa tako jedan od njih dobija epitet glavnog, a drugi alternativnog seta. Ako ste dobar programer, pa još i dobar strateg, ovo je prava stvar za vas. Na žalost, moramo računati i na...

Amstradove specifičnosti

Kod ovog računara su dve pomenute naredbe asemblera „zabranjene“. Naravno, radi se o softverskoj „mučki“ programera „lokomotiv softvera“. Izgleda da su imali prilično dobro mišljenje o sebi (ko ga nema), pa su iskoristili alternativni set, ljubomorno ga čuvajući samo za svoj operativni sistem. Radi se o dosta čudnoj



odluci ljudi koji su pravili računar, ali je ona ipak motivisana. Činjenica je da je mnogo više ljudi čulo za benčmark testove, nego za alternativni registarski set. Računar se bolje prodaje kada ima bolje benčmark testove i oni se redovno prvi podmeću potencijalnim kupcima (budite sasvim sigurni da se svi prikazi računara tipa „dva-sata-nasamo-sa“ započinju gubljenjem pola sata na merenje raznih testova, a takvi prikazi često odlučuju umesto vas, iako su neobjektivni i najčešće netačni). Pošto je, na žalost, novac vrhovni arbitar na zapadu, doneta je odluka u korist brzine dobijene na pomalo veštački način, a na štetu programera.

Alternativni registarski set se na „amstradu“ ne sme koristiti iz dva razloga: prvo — krahirać naš program koji ga koristi, jer OS ne „čuva“ registre alternativnog seta, i drugo — krahirać OS, jer u nekim od tih registara „drži“ svoje promenljive. Nemojte pomisliti da to može da se reši izbegavanjem rutina operativnog sistema — obrada interupta, do koga dolazi 300 puta u sekundi, pripada kernalu, a upravo on „brlja“ po alternativnom setu. Zbog toga se u interapt rutini „amstrada“ instrukcije PUSH i POP ne mogu naći ni pomoću lupe. Da nije primerjen ovaj način da se prilikom prekida sačuvaju sistemi „amstrad“ bi bio sporiji bar za 10 procenata, a time bi bila izgublje-

na svaka šansa da se sustigne BBC B. Ono što je Ejkorn pružio vlasnicima svojih računara kvalitetnim hardverom, to je Amstrad izveo softverski na prilično nepopularan način. Tako nam sada jedino preostaje da sami, softverskim zakrparama, pokušamo da rešimo ono što je trebalo rešiti mnogo, mnogo ranije.

Pustio bih ja njega, ali...

Verovatno je već mnogima pala na pamet ideja da se prekidni omogućuje dok se koristi alternativni set. To je, ipak, više nego polovično rešenje. Nije preporučljivo provoditi veći deo vremena bez interapta — tastatura će biti „gluva“, neće biti merenja vremena, itd. Pojedini programeri (uglavnom oni iskusniji) ne preporučuju, iz istih razloga, ni upotrebu instrukcija tipa LDIR, ako se radi o većim količinama podataka.

Na početku je najbolje da upoznamo način, na koji OS koristi alternativni set. Za smeštanje promenljivih koriste se samo BC' (alternativni BC registar) i keril' fleg. U B' registru čuva se adresa I/O porta video vrata (Video Gate Array) — #7F. To je vrlo zgodno mesto u kome se odvijaju mnoge značajne stvari, ali ostavimo to za neki prikladniji trenutak. U C' se nalazi vrednost koja se šalje na pomenuti port, pa pogledajmo kakva značenja može imati:

- bit 7,6 — određuju tip komande: u ovom slučaju to će biti, respektivno, 10
- bit 5 — 0
- bit 4 — ovaj bit je u vezi sa interaptom i biće objašnjen u drugom napisu slati 0)
- bit 3 — kontrola gornjih ROM-ova i znaci „isključiti“
- bit 2 — kontrola donjih ROM-a (i znaci „isključiti“)
- bit 1,0 — određuju mod u kom se računar nalazi, na sledeći način:
 - 00 — mod 0 16 boja, 160x200 tačaka)
 - 01 — mod 2 (4 boje, 320x200 tačaka)
 - 10 — mod 2 (2 boje, 640x200 tačaka)
 - 11 — ilegalna vrednost (ne koristiti)

Alternativni keril fleg (carry') je obično 0, a kada je setovan znači da je nastupio interapt.

Preostale registre alternativnog seta keril koristi da je izbegao operacije sa ste-kom (PUSH i POP), koje su prilično spore. Na taj način, značajno su ubrzani pozivi rutine operativnog sistema preko glavnog džamp-bloka.

Pokušajmo sada, uz jedan primer, da objasnimo upotrebu alternativnog seta registra. Primer je gotovo trivijalan, ali poslužiće svrsi.

Ukoliko ste čitali prethodne „Računare“, sigurno se sećate zbrke sa ROM-ovim i RAM-om. Kada je neki od ROM-ova „uključen“, „čitanje“ sa lokacije na kojoj se preklapaju RAM i ROM vrši se iz ROM-a. Ponekad programer nije siguran kakvo je stanje nekog ROM-a, a samim tim i odakle će dobiti podatke. Za takve slučajeve može se napraviti rutina koja uvek „čita“ podatke iz RAM-a. Recimo da ta rutina simulira instrukciju LD A, (BC).

```
iz_BC_UA:
DI                ;interapt bi napravio zbrku
EXX              ;iskoristimo alternativni
                set
LD A, C          ;C se nalazi vrednost koju
                možemo poslati na port
                #7FXX
SET 2, A        ;„isključiti“ donji...
SET 3, A        ;... i gornji ROM
OUT(C), A      ;sada ih stvarno „isključiti“
EXX            ;adresa je u BC
LD A, (BC)     ;„pročitaj“
EXX
```

```
Program II
usp_all: CALL sys_all
          RST #30
          JP usp_all

;
sys_all: DI
          EX AF,AF'
          LD (usr_MJ),HL
          LD (usr_DE),DE
          LD (usr_BC),BC
          PUSH AF
          POP HL
          LD (usr_AF),BC ;sačuvaj alternativni
          LD (usr_MJ),HL ;set korisnika
          LD HL,(sys_int)
          LD BC,(sys_BC) ;izad sturu adresu,
          DE A          ;sistemski BC'
          EXX           ;i keril' na nulu
          EX AF,AF'
          SET

;
usp_all: DI
          EX AF,AF'
          EXX
          LD (usr_BC),BC ;sačuvaj sistemski BC
          LD HL,(R30)    ;i adresu rutine za
          LD (sys_int),HL ;obradu interapta
          LD HL,(usr_int) ;izadsti adresu naše
          LD (R30),HL    ;rutine
          LD HL,(usr_AF)
          PUSH AF
          POP AF
          LD HL,(usr_MJ)
          LD DE,(usr_DE) ;restauriraj vrednosti
          LD BC,(usr_BC) ;alternativnog seta
          EXX
          EX AF,AF'
          EI
          RET

;
sys_int: DEFB 2 ;adresa sistemske rutine
sys_BC: DEFB 2 ;sistemske promenljive iz BC'
usr_AF: DEFB 2 ;vrednosti
usr_BC: DEFB 2 ;korisničkog
usr_DE: DEFB 2 ;alternativnog
usr_MJ: DEFB 2 ;seta
```

```
OUT(C), C      ;vrați staro stanje ROM-ova
EXX            ;preklopi registre
EI             ;sada može da nastupi
              prekid
RET
```

Dakle, kad god izvršimo instrukciju CALL iz_BC_UA, u A registru dobijamo

vrednost RAM ćelije na adresi BC, pri čemu će svi ostali registri i flegovi biti sačuvani (zbog poslednjih su korišćene dve instrukcije SET čiji bi ekvivalent bio instrukcija OR %00001100 koja je kraća, ali bi poremetila rutine). Ovakvo, otprilike, deluju i izgledaju flegove OS-a koje koriste alternativni registarski set (usput, identična rutina koja simulira LD A, (HL) već se nalazi u računaru, poziva se sa RST #20 i odziva na ime RAM LAM). Odmah se primećuje zavidna efikasnost, pre svega zbog promenljivih koje se nalaze u BC registru.

Ova tehnika upotrebe alternativnog seta dobra je samo za ovako jednostavne i kratkotrajne operacije. Ukoliko ono što radimo namerava da uzme veći deo procesorskog vremena, moramo se poslužiti nešto komplikovanim tehnikom, koja bi nam pružila veću slobodu — drugim rečima omogućen interapt.

Sa interaptom

Ukoliko naš program duži period vremena (ili sve vreme) koristi alternativni registarski set, moraćemo da se pobrinemo da interapt funkcioniše na valjan način. To je moguće učiniti jedino umetanjem naše interapt rutine između zahteva za prekidom i same obrade prekida. Pošto svaki zahtev indukuje poziv potprograma koji počinje od adrese #38, na tom mestu moramo da ubacimo rutinu koja će sprečiti nastanak gore-opisane zbrke. Po uključenoj računara, na adresi #38 nalazi se jedna džamp instrukcija koja upućuje na neku adresu u RAM-u — tu je potprogram za obradu interapta. Ta adresa nije u svim „amstradi-ma“ ista, jer je OS pretrpeo dosta izmena od modela 464 do 6128. Umesto tog džampa, ubacićemo adresu sopstvene rutine. Njen zadatak bi bio da posle izvesnih operacija pozove rutinu za obradu prekida.

„Izvesne operacije“ bi bile sledeće:

- 1) sačuvati korisnički alternativni registarski set;
- 2) restaurisati sistemske promenljive u BC' registru i keril' flegu
- 3) vratiti adresu, koju pri EMS early-morning startu OS postavlja iza koda za džamp (#C3), na adresu #39;
- 4) pozvati rutinu za obradu interapta



RST #38);

5) opet postaviti naš džamp na adresu #38;

6) sačuvati sistemski BC';

7) restaurirati korisnički alternativni set.

Verovatno će se velikom broju čitalaca korak 3), a samim tim i korak 5), učiniti na prvi i sve ostale pogledе potpuno nepotrebnim. Naprotiv, oni su od „vitalnog značaja“, što bi rekli naši rukovodioci — neka im se grlo nikad ne osuši od promaje. Ako se sećate „događaja“ koji su opisivani u prošlom broju „Računara“, setite se i činjenice da se neki od njih normalni asinhroni)

Ovim rutinama nije potrebno naročito objašnjenje. Pre povratka iz potprograma usalt se ne omogućuje prekid (nema EI naredbe) jer se on poziva pošto je već nastao interapt. Pre početka upotrebe alternativnog seta, potrebno je izvršiti CALL usr_alt, a kada nam ne bude više bit potreban — CALL sys_alt, pa EI.

Postoji niz slabosti ovakve upotrebe alternativnog seta. Na primer, nemoguće je menjati ROM, a to znači da će se u ćelijama #39, #3A ROM-a uvek nalaziti adresa kernalove rutine za obradu prekida. Znači, dok koristimo alternativni set, donji ROM

Sa operativnim sistemom

Prilikom svakog poziva operativnog sistema potrebno je izvršiti istu proceduru kao kod interapta. To bi moglo ovako da izgleda:

CALL sys_alt :vratiti alternativni set tako da može da ga koristi OS
CALL sys_rout :pozovi potrebnu rutinu pre ovoga nije potrebno EI — ta naredba će ionako biti izvršena posebnim „uključivanjima“ donjeg ROM-a
CALL usr_alt :vratiti nazad korisnički alternativni set

Sve ovo izgleda lepo, ali će za samo 10 poziva biti potrađeno 60 bajtova. Potrađeno, jer se svaki poziv može obaviti sa 3 bajta. Jedno rešenje bi moglo da bude upotreba jednog slobodnog restarta, s tim da adresa (sys_rout) sledi posle instrukcije RST. Na takav način je regulisan glavni džamp-blok, pa svaki njegov ulaz zauzima 3 bajta. Pošto je jedan restart već odujvo u ovu svrhu, zašto bismo trošili sledeći? S druge strane, ova rutina nije ni izdaleka tako komplikovana kao ona za obradu interapta.

Opet ćemo, kao u prethodnom rešenju, malo preurediti nultu stranicu RAM-a. U glavnom džamp-bloku koristi se RST #08. Tamo je opet neki džamp, koji ćemo zameniti skokom na našu rutinu. Prethodno moramo „zapamtiti“ staru adresu.

Pored glavnog, u memoriji „amstrada“ se krije još jedan džamp-blok. On koristi drugi restart za pozive rutina OS-a — RST #28 (FIRM JUMP). Preko tog džamp-bloka poziva se amstradov kalkulator.

U programu 2, pored ova dva restarta, uzeti su u obzir i RAM LAM i FAR CALL. Tako se dati program može koristiti i sa bežikom. Kada počnete upotrebu alternativnog seta, potrebno je pozvati rutinu start, a kada završavate — potprogram end. Pora važi ako rutinu pozivamo iz bežikja, a ako to činimo iz mašinskog programa, potrebno je izbaciti naznačene linije. Rutina end se može koristiti u oba slučaja. Za one koji se još sećaju priče o brzini i alternativnom setu, da kažemo jedan podatak: sa programom 2 bežik interpreter će biti sporiji za oko 25%, dok će OS raditi istom brzinom. To je cena koja mora da se plati ako želimo da alternativni set nađe mesta u našim programima. Još nešto — ako ste sigurni da se neka od kernalovih rutina, koje su uzete u obzir u ovom programu, neće koristiti, možete slobodno da izbacite sve što se na nju odnosi. S druge strane, ako nameravate da koristite neku rutinu koja ne ulazi u ovaj izbornik, bilo bi korisno da je ubacite. To možete da učinite na način analogan datom, sagledavajući razliku između programa 1 i 2. Nadamo se da ste zapazili da neke rutine dobijaju podatke preko steka (tzv. inline parameters), na šta takođe treba obratiti pažnju.

Na kraju, naglasimo još jednom da donji ROM ne sme biti „uključen“ ni pod kakvim uslovima ako alternativni set nije prethodno prebačen na sistemsku varijantu.

Autor ovog teksta se nada da će ovo biti podstrek za pisanje programa koji koriste alternativni registarski set, a to znači programa koji neće biti napisani da bi zaboravljani čamili na polici za kasete.

Dejan Muhamedagić

10 org 0a00			
11 [na registar #10 (FIR CALL) bica aznaceno u 2	576 sysalt: deis 2	10	
20 ussalt: 41	480 sysalt: deis 2		
30 os al,al'	416 calcors: deis 2		
40 osz	432 i		
50 1d [oydci,lc	432 ussalt: call sysalt		
60 1d [i,ussalt	448 rcl 830		
70 1d [8291,kl	450 jz ussalt		
80 1d [i,ussalt	464 i		
90 1d [897,kl	470 ussalt: call sysalt		
100 1d [i,ussalt	480 osz		
110 1d [8191,kl	490 1d [i,ussalt		
120 1d [i,ussalt	500 os [top,kl		
130 1d [8291,kl	510 push kl		
140 1d [i,ussalt	520 osz		
150 1d [8211,kl	530 jz 810		
160 1d [i,ussalt	540 i		
170 push kl	550 ussalt: call sysalt		
180 pop al	560 osz		
190 1d [i,ussalt	570 1d [i,ussalt		
200 1d [8v,osrde	580 os [top,kl		
210 1d [lc,ussalt	590 push kl		
220 osz	600 osz		
230 os al,al'	610 jz 810		
240 os	620 i		
250 rot	630 ussalt: call sysalt		
260 i	640 osz		
270 sysalt: 41	650 1d [i,ussalt		
280 os al,al'	660 os [top,kl		
290 osz	670 push kl		
300 1d [ussalt,kl	680 osz		
310 1d [ussalt,de	690 jz 820		
320 1d [ussalt,lc	700 i		
330 push al	710 ussalt: call sysalt		
340 pop al	720 rot 830		
350 1d [ussalt,kl	730 jz ussalt		
360 1d [i,tyssalt	740 i		
370 1d [8291,kl	750 end: call sysalt		
380 1d [i,tyssalt	760 os		
390 1d [897,kl	770 rot		
400 1d [i,tyssalt	780 i		
410 1d [i,calcors	790 start: 1d [i,8291		
420 1d [8291,kl	800 1d [tyssalt,kl		
430 1d [i,tyssalt	810 1d [i,897		
440 1d [8211,kl	820 1d [tyssalt,kl		
450 1d [lc,tyssalt	830 1d [i,8191		
460 os a	840 1d [tyssalt,kl		
470 osz	850 1d [i,8291		
480 os al,al'	860 1d [calcors,kl		
490 rot	870 1d [i,8211		
500	880 1d [tyssalt,kl		
510	890 os		
520 ussalt: deis 2	900 deo jz ussalt (posobaj)		
530 ussalt: deis 2	1110 pop lc		
540 ussalt: deis 2	1120 pop bc		
550 ussalt: deis 2	1130 end [i,c		
560 sysalt: deis 2	1140 osz		
570 sysalt: deis 2	1150 os		
580 sysalt: deis 2	1160 jz ussalt		

obradju neposredno pre povratka iz rutine za obradu prekida i niko vam ne garantuje da će trajati manje od tri milisekunde, kada se bude javio novi interapt. Sada je jasno šta bi se desilo da u tom trenutku na adresu #39 bude adresa rutine koja sledi pravila datog algoritma.

Po ugledu na strukturirano programiranje, izdvojimo korake 1), 2) i 3) u jedan potprogram i korake 5), 6) i 7) u drugi (program 1).

bezuslovno mora biti „isključen“ (osim naravno, kada nastupi prekid). Stoga se moramo potpuno odrediti svih usliga OS-a.

Postoji i drugi način kojim bismo zaobišli ove teškoće, ali on zahteva mnogo više truda. Trebalo bi napraviti novu rutinu za obradu prekida koja bi čuvala alternativni set i ispunjavala iste uslove kao i postojeća. Možda bi tako nešto ogadilo ručak i ostale obroke čak i najzagriženijim hakerima, pa ćemo to ostaviti za bolja vremena. Uz to, trebalo bi uraditi još mnogo stvari. Da ih ne nabrajamo — sve ono što je trebalo da urade oni koji su pisali operativni sistem.

Objavlivanjem sheme kertridža načinili smo tek u „Računarima 10“ prvi korak — sada prelazimo na daleko interesantniju stvar: kreiranje programa koji ćemo u njega smestiti. Iako bi nekoga zadovoljilo da u kertridž smesti neki od programa koji često koristi i tako skрати muke oko njegovog stalnog učitavanja sa spoljne memorije, mi smo izabrali težu, ali zato daleko interesantniji put — udruženo pisanje potpuno novog roma! Krajnji cilj je iskorišćenje svih hardverskih potencijala kojima raspolaže „komodor“ Uz to ćemo se truditi da maksimalno povećamo komfor u radu, naročito sa perifernim jedinicama. Nedostatak koji oseća najveći broj korisnika je nemogućnost rada sa grafikom visoke rezolucije. Odgovor je SIMON'S BASIC, ali je taj program potpuno zaštićen neverovatnom količinom novih (i nepotrebnih) naredbi i deluje kao lečenje kompleksa jednog računara koji „može mnogo, ali se to ne vidi“. Ako vam se dopada rešenje koje vam nudimo, prionimo zajedno na posao.

```

1 INICIJALIZACIJA
LDA #*COMM
LDA #*COMM
LDY #76
STY #20
STA #21
STY #22
RTB

DISPECER NAREDBI
COMM
PHR
TAA
LDA #2
CMP #4
SCC OK
PHR
LDA #255
SEC
RTS

EMFOR EXIT
PLA
TAX
PLA
PLP
LDA #255
SEC
RTS

OPREDIVJANJE POZITIVNE ADRESE
OK
CMP #29
BCS NEPLIN

STANDARDNE FUNKCIJE KERNALA
ASL
CLC
ADC #2
ADC #*ASO409
STA #23
LDA #255
STY #24

EXECUTE
EXEC
PLA
TAX
PLA
PLP
JMP (#23)

NOVE FUNKCIJE OPERATIVNOG SISTEMA
NEMLIN
SBC #29
ASL
TAX
LDA #TAB.X
STA #23
LDA #TAB+1,X
STA #24
CLC
RCC EXEC

TABLA ADRESA NOVIH FUNKCIJA
FTAB
.BYT <EXPEN, >EXPEN
.BYT <IDERR, >IDERR
  
```

prekine rad i samo proslodi ovo stanje na izlaz. Ako, pak, taj deo prođe u redu, ali se, recimo, u disk jedinici ne nalazi disketa, ovo se mora posebno ispitati očitavanjem stanja preko sekundarne adrese i 15 kada će disk poslati poruku (ne zaboravite: format je ASCII a ne binarni) broj 74 (Drive Not Ready).

Korisniku vaše rutine treba da bude svejedno gde je nastao problem. U akumulatoru se u oba slučajeva mora naći kod greške, u prvom slučaju 5, u drugom 174, što vi treba da obezbedite. Jednu takvu rutinu smo razvili, IOERROR na slici 2, kao primer kako treba da izgleda prilog, pa se ona može koristiti za ovu namenu. Pošto operativni sistem koristi kodove grešaka od 0 do 9 i 24, a za disk su namenjene one preko 100, ostaje nam prostor od 10 do 99. Vi treba samo da navedete tekst greške, a mi ćemo joj dodeliti broj i uvrstiti je u mapu da bismo izbegli preklapanja.

ROJEVI FUNKCIJA OPERATIVNOG SISTEMA

0 - CINT	21 - OPEN
1 - IOINIT	22 - CLOSE
2 - RRTAB	23 - OKIN
3 - RESTORE	24 - CHKOUT
4 - VECTOR	25 - CLRCHN
5 - SETMSG	26 - WRIN
6 - SECOND	27 - CHROUT
7 - TWIN	28 - LDAB
8 - MENTOP	29 - SAVE
9 - MEMBOT	30 - SETTIM
10 - SINKEY	31 - ROTIM
11 - SETTH	32 - STOP
12 - ACPT	33 - GETIN
13 - CIOUT	34 - CLALL
14 - UNTLK	35 - LDTIM
15 - UNLSN	36 - SCREEN
16 - LISTEN	37 - PLOT
17 - TALK	38 - IDWBS
18 - READST	39 - EXPEN
19 - SETLFS	40 - IOERROR
20 - SETNAM	

SPISAK KODOVA GREŠAKA

0-9, 240 (vidi dodatak "Računar 8", str. 22)
100 - 174 (prema uputstvu za CHK1541)

7. Rutine koje pripadaju operativnom sistemu ne smeju se oslanjati na bejzick ROM.

I, na kraju, pravilo 8:

8. Dokumentacija uz svaku novu rutinu treba da sadrži:

- simbolički naziv;
- kratak opis namene i načina rada;
- potrebne ulazne parametre i način njihovog prenošenja u rutinu;
- izlazne parametre i način prenošenja sem ako nastane greška, što smo standardizovali;

- spisak lokacija u prve četiri strane memorije (0—1023) koje se koriste — koji su sadržaji registra očuvani; — dužinu programa u bajtovima i potrebnu količinu radne memorije.

Otvaranje datoteke

Postujući sve ove propozicije, opisaćemo postupak koji treba primenjivati u radu na primeru rutine EXPEN (Extended Open) koji smo razvili. Koristićemo Profi.Ass 64 koji smo prilagodili za kertridž, redakcija vrši programiranje (eproma). Prvo treba prekucati program sa slike 3, koji je pri razvoju programa neophodan jer se njime obezbeđuje standardizovan način pozivanja funkcija. Primećujemo da nema DRG adrese. Upotrebite bilo koju — za početak će biti pogodna i 49152 (C000). Zatim prekucajte listing sa slike 2 koji predstavlja novu funkciju (broj #40 IOERROR status disk jedinice) koja će nam biti potrebna za ostvarivanje EXPEN. Isto ćete uraditi sa svim ostalim rutinama drugih autora koje budemo objavili (ako su potrebne za realizaciju vaše) i u tabeli FTAB dodajte naziv vaše rutine. Time će ona samo privremeno dobiti broj i u prema mestu u tabeli (prva rutina u tabeli ima broj 39), dok ćemo joj konačan broj dodeliti mi, kada je definitivno uvrstimo u spisak. Sve ovo može izazvati malu zabunu — kako da razvijate program kada ne znate tačne brojeve svih postojećih funkcija? Zato je neophodno da se pridržavate sledećeg: brojevi funkcija koje su već objavljene moraju se poštovati pa od ovog trenutka važi da EXPEN nosi broj 39, a IOERROR 40 i njihovim nazivi se moraju nalaziti u FTAB u redosledu koji smo dali. Ovo važi čak i ako ih ne prekucate pri razvoju vaše; u tom slučaju negde na kraju, da assembler ne bi prijavljivao grešku, samo dodajte dve labela sa ovim nazivima i recimo NOP instrukcijom. Kada završite razvoj, listing vaših rutina izdvojite i samo njega pošaljite. Ako ih ima više i jedna poziva drugu, onda brojevi funkcija takođe treba da budu dati simbolički. Da bi pozivanje funkcija preko broja bilo moguće, potrebno je da inicijalizujete program. Ovo obavlja prvih 6 instrukcija sa slike 3, pa ih treba izvesti pre startovanja vašeg programa.

EXPEN je zamišljena kao jednostavnija rutina za otvaranje datoteke, prvenstveno na disku. Pošto za ovu operaciju treba pozvati 3 KERNAL rutine SETNAM, SETLFS i OPEN) sa ukupno 6 bajtova ulaznih parametara, a postupak se često pojavljuje u programima, smatrali smo da je potrebno znatno smanjiti broj potrebnih instrukcija za ovaj zadatak. Tako se u X I Y registru nalazi adresa početka naziva datoteke, dok smo navođenje dužine izbegli tako što naziv mora završavati bajtom 0. Sekundarna adresa i broj uređaja se mogu sasvim fino spakovati u jedan bajt (bitovi 0—3 = broj uređaja, 4—7 = sekundarna adresa), a EXPEN će sam odrediti logički broj pod kojim je veza otvorena i ako je otvaranje proterko u redu vratiti ga kao izlazni rezultat u X registru.

NASTAVAK NA 55. STRANI

lovana. Nije dozvoljeno nedefinisano stanje niti drugačija signalizacija.

Ovakav način je već postao standard za prijavljivanje grešaka, pa smo ga jednostavno preuzeli iz postojećeg sistema. Da bi na neki način čvršće povezali sa računarem i operativni sistem disk jedinice, preuzeli smo i njegove brojeve grešaka, ali ih uvećavamo za 100. Tako pri izlazu iz novih rutina operativnog sistema neće biti pravih razlika u načinu prijavljivanja grešaka računara i diska. Ako se otvara datoteka na disku koji nije uključen, KERNAL OPEN rutina će javiti grešku br. 5 i postaviti carry zastavicu, pa je dovoljno da vaša rutina

Matematički softver To može i bolje (4)

Logaritamska funkcija

Izrada algoritma za izračunavanje vrednosti logaritamske funkcije na računaru sastoji se od više faza. Prva faza je utvrđivanje opsega argumenta za koji se može računati logaritam. Druga faza je utvrđivanje osnovnog intervala u koji će se preslikavati svi argumenti. Treća je izbor aproksimacije u osnovnom intervalu. Četvrta faza je identifikovanje kritičnog intervala u kome je izračunavanje vrednosti algoritama posebno delikatno zbog moguće velike relativne greške, koju, naravno, treba sprečiti. Peta faza je pisanje programa. Dakle, za programera ima više problema koje treba uspešno rešavati i rešiti.

Korisnik računara veruje da je dovoljno za zadani argument X pozvati firmnu funkciju LOG i ona će izračunati vrednost LOG(X) u koju se (korisnik računara) može pouzdati. Ta greška može da ima (i obično ima) proizvoljan broj netačnih bitova mantise za programerski odabranu (nesprijetnu) vrednost argumenta. Koje su to vrednosti argumenta korisnik računara ne služi. Ako uspe samostalno da ih otkrije, to znači da ima sklonosti ka programiranju. Razvijanju tih sklonosti posvećena je ova serija tekstova.

Na osnovni interval

Vrednost logaritamske funkcije može se izračunati za sve pozitivne vrednosti argumenta. Ovdje imamo u vidu, pre svega, računare „komodor 64“, „spektrum“ i „šarp MZ-700/800“ koji imaju odvojen 1 bajt za smeštaj karakteristike k i 4 bajta za smeštaj mantise m. Broj X u pomenutim računarnim predstavljen je sa

$$X = \pm 2^m,$$

sem nule koja je predstavljena nulama.

Karakteristika k je ceo broj za pomenute računare od -127 do 127-

$$k = -127 (1) 127$$

pa je opseg argumenta $2^{-128} \leq |X| < 2^{127}$ ili, grubo, od 10^{-38} do 10^{38} .

Mantisa m uzima vrednosti od 1/2 do $1 - 2^{-k}$ u koraku po 2^{-k}

$$m = 1/2 (2^{-k}) - 1 - 2^{-k}.$$

gde J označava broj bitova mantise (za pomenute računare J=32, pa je tačnost 2^{-32} , skoro 10 značajnih cifara). U prvom bajtu broj nalazi se karakteristika uvećana za 128. Umesto vodećeg bita normalizovane mantise (koji je uvek 1) zabeležen je znak broja (1 za negativne, inače 0).

Izrada algoritma i programa za logaritamsku funkciju je nezamisliva bez svodenja argumenta na osnovni interval. Razlog za to je nemogućnost primene nekog potencijalnog, verižnog ili kakvog drugog razvoja za proizvoljno pozitivno X. Dakle, ne radi se o tome da li svesti argument na osnovni interval, već kako ga svesti.

Način predstavljanja bitova u tome pruža nesumnjivu pomoć. Naime, kako je za pozitivno X

$$X = 2^m,$$

onda je

$$\log(X) = k \log(2) + \log(m).$$

log(2) je poznata konstanta, pa je ne treba iznova računati. Računanje $\log(X)$ proizvoljnog argumenta X svodi se na izračunavanje logaritma mantise, jedno sabiranje i jedno množenje. Izračunavanje logaritma mantise je znatno jednostavnije jer se vrednost mantise kreće od 1/2 do 1. Umesto intervala [1/2, 1], uzimaju interval [1, 2) a neki uzimaju (sqrt(1/2), sqrt(2)). Uopšte, može se predstavljati

$$X = 2^m$$

zameniti sa

$$X = 2^f$$

gde je

$$I = k-1, F = 2m \text{ za } 1/2 < m < m_g,$$

$$I = k, F = m \text{ za } g < m < 1.$$

Tako se dobija za očuvanje tačnosti izračunavanja pogodniji oblik

$$\log(X) = i \log(2) + \log(f),$$

gde je $g < f < 2g$ u $1/2 < g < 1$.

Krajnje vrednosti za g su vrlo nepovoljne za računanje. Pojedini metodi su najuspešniji za $g = \text{sqrt}(1/2)$, ali to nije opšte pravilo.

Prema tome, izračunavanje vrednosti logaritma proizvoljnog argumenta jednostavno se svodi na izračunavanje vrednosti logaritma broja između i i 2g. To „jednostavno se svodi“ je ponekad čist optimizam. Naime, ne želimo da pravimo reklamu jednom (domaćem) proizvođaču računara, ali taj računar sadrži program koji je sušta negacija fraze „jednostavno se svodi“. Ako je argument veći od gornje granice (2g) program polovi argument i dodaje međurezultatu log(2), zatim ponovo itd. Ako je argument manji od donje granice (g), program udvostručava argument i oduzima log(2) od međurezultata, zatim ponovo itd. Broj tih ponavljanja i dodavanja (ili udvostručavanja i oduzimanja) je poveći čak i za tako skroman opseg brojeva $2^{-128} \leq x < 2^{127}$. Proizvođač računara pitanje o trajanju izračunavanja po ovom „algoritmu“ smatra nepristojnim. Polovina je što su umesto preuzimanja tuđe pameti pokušali sami da nešto urade. Ipak, izrada matematičkog softvera proizvoda koji ide u prodaju ne sme se prepustiti početnicima.

Svodenje na osnovni interval je zaista jednostavno ako se karakteristika uzme iz prvog bajta argumenta X. Računar SHARP MZ700 u HUBASIC-u ima funkciju

VARPTR(X)

koja daje adresu prvog bajta argumenta X. Kod „spektruma“ i „komodora“ to se postiže na komplikovaniji način, jer se mora promenjivaju čiju adresu tražimo nalaziti na odgovarajućem mestu u programu.

Karakteristika k je sadržaj prvog bajta promenjive, manje 128. Ako se u prvi bajt upiše broj 128, onda se na adres promenjive X nalazi m. Poređenje nove promenjive m sa g omogućava da se sadržaj 128 eventualno zameni sa 129, uz smanjenje upamćene karakteristike k za 1. Dakle, testiranje na g treba da se obavlja samo jednom (a ne stotinu puta).

Kako izračunati $\log(X)$ u intervalu od g do 2g?

Obično se preporučuje . . .

Jedan od načina je po definiciji

$$\log(x) = \int_1^x dx/x \quad (x > 0).$$

To nije tako beznaždan metod ako se integral računao na valjan način. U ovoj seriji napisa o matematičkom softveru još smo daleko od integracije, jer treba najpre umeti izračunati vrednosti matematičkih funkcija, pa tek onda nešto njima računati. Šta je valjan metod integracije je pitanje na koje će se odgovoriti posle pregleda izračunavanja matematičkih funkcija.

Neki autori za računanje vrednosti logaritma preporučuju formulu (1)

$$\log(x) = \lim_{h \rightarrow 0} (x^h - 1) / h$$

„Preporučuju“, jer je sami ne koriste i to zato što se ni ne može koristiti na računaru. Na slici 1 za nekoliko vrednosti argumenta x prikazana je

```
1 REM LOG(X)  COMODORE 64
2 X=0: Y=0
3 INPUT X: GOSUB 4: PRINT Y: GO TO 3
4 Y=X: IF Y<=0 STOP
5 I=PEEK(46)*256+PEEK(45)+9
6 K=PEEK(1)-128: POKe I,128
7 IF Y,7871867812 K=K-1: POKe I,129
8 S=(Y-.75)-.25/(Y+1): Y=54S
9 Y=((((1,1136858Y+.14285714286 J
+Y,2)Y+.333333333333 J)Y+5)2
+K,88564718856 +K,6875: RETURN
```

```
1 REM LOG(X)  ZX SPECTRUM
2 LET X=0: LET Y=0
3 INPUT X: GOSUB 4: PRINT Y: GO TO 3
4 LET Y=X: IF Y<=0 THEN STOP
5 LET I=PEEK(23628)*256+PEEK(23627)+7
6 LET K=PEEK(1)-128: POKe I,128
7 IF Y,7871867812 THEN LET K=K-1:
POKE I,129
8 LET S=(Y-.75)-.25/(Y+1):LET Y=54S
9 LET Y=((((1,1136858Y+.14285714286 J
+Y,2)Y+.333333333333 J)Y+5)2
+K,88564718856 +K,6875: RETURN
```

```
1 LOG(X) SHARP MZ-700 (HUBASIC)
3 INPUT X: GOSUB 4: PRINT Y: GOTO 3
4 Y=X: IF Y<=0 STOP
5 I=VARPTR(X)
6 K=PEEK(I)-128: POKe I,128
7 IF Y,7871867812 K=K-1: POKe I,129
8 S=(Y-.75)-.25/(Y+1): Y=54S
9 Y=((((1,1136858Y+.14285714286 J
+Y,2)Y+.333333333333 J)Y+5)2
+K,88564718856 +K,6875: RETURN
```

relativna greška r u funkciji od h, inače, stepenovanje x^h može se izvršiti sa nekoliko uzastopnih korenovanja. Slika pokazuje da se za najpovoljniju vrednost parametra h (približno 2^{-50} , koren iz relativne greške predstavljanja brojeva) dobija samo polovina broja tačnih cifara rezultata. Analitički gledano, sa smanjenjem h trebalo bi da se tačnost formule povećava. Međutim, pored analitičke greške oblika 0(h) ovde postoji još i numerička greška oblika 0(1/h). Zbir te dve greške je veliki za svako h, pa je ovaj metod neupotrebljiv. Treba uočiti da greška ne poliče od tačnosti izračunavanja čitavih X, pretpostavlja se da je taj član izračunat tačno do poslednjeg bita mantise. Greška poliče od razlike x^h i 1.

Na slici 2 data je relativna greška formule

$$(2) \log(x) \approx \sum_{k=1}^n (1-1/x)^k/k$$

u funkciji argumenta x. Kako smo usvojili tačnost 2^{-32} , za $g = \text{sqrt}(1/2)$ izlazi da je broj sabiraka n=22. Za $g=0.53$ broj sabiraka je n=8.

Na slici 3 predstavljen je relativna greška potencijalnog razvoja

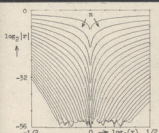
$$(3) \log(x) \approx \sum_{k=1}^n (1-x)^k/k$$

u funkciji argumenta x. Za $g = \text{sqrt}(1/2)$ je n=22, inače za $g=0.67$ je n=8.

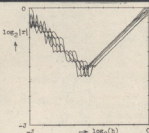
Na slici 4 data je relativna greška formule

$$\log(x) \approx \sum_{k=2}^n (x-1+k) / (k+1) \cdot x$$

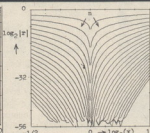
Korisnik računara zna da je računar snabdeven programom LOG koji za zadanu vrednost argumenta X izračunava približnu vrednost logaritamske funkcije LOG(X). Programer zna za koje argumente X je taj firmin rezultat neupotrebljiv zbog ogromne relativne greške. Programer zna i algoritme tačne do poslednjeg bita mantise rezultata. Prof. dr Dušan Slavić nastavlja seriju „To može i bolje“ detaljima iz svog dugogodišnjeg rada u oblasti računarskih algoritama.



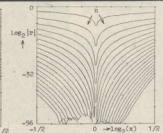
Slika 2. Relativna greška formule $\log(x) \approx \sum_{k=1}^n (1-1/x^k)/k$ u funkciji argumenta x.



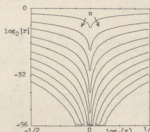
Slika 3. Relativna greška formule $\log(x) \approx \sum_{k=1}^n (x^k-1)/k$ u funkciji argumenta h.



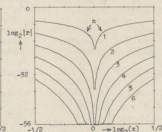
Slika 3. Relativna greška formule $\log(x) \approx \sum_{k=1}^n (1-x)^k/k$ u funkciji argumenta x.



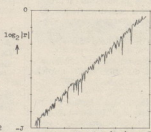
Slika 4. Relativna greška formule $\log(x) \approx (x-1) \sum_{k=1}^n (1-x)^k/(k(k+1))/x$ u funkciji argumenta x.



Slika 5. Relativna greška formule $\log(x) \approx \sum_{k=1}^n \frac{1}{x^k} - \frac{1}{k}$ u funkciji argumenta $x = 1+t$.



Slika 6. Relativna greška formule $\log(x) \approx \sum_{k=1}^n \frac{2 - (3-1)^{2k-1}}{2k-1}$ u funkciji argumenta x.



Slika 7. Relativna greška koje nastaje zbog nebrzišljivog oduzimanja 1 od x u kritičnom intervalu.

u funkciji argumenta x. Za $g = \sqrt{1/2}$ je $n=18$, inače za $g=0.67$ je $n=16$. Navedene formule očigledno se odnose za sledećim.

Na slici 5 predstavljena je relativna greška veriznog razvoja

$$(5) \log(x) \approx \frac{1 \cdot 1^2 + 2 \cdot 2^2 + 3 \cdot 3^2 + 4 \cdot 4^2 + \dots}{1 + 2 + 3 + 4 + 5 + 6 + 7 + 8 + \dots} \quad (t=x-1)$$

u funkciji argumenta x. Za $g = \sqrt{1/2}$ dobija se da je red veriznog razvoja 10.

... a trebalo bi

Po efikasnosti posebno se izdvaja sledeća formula

$$\log(x) = 2 \operatorname{arctanh}\left(\frac{x-1}{x+1}\right)$$

$$(6) \log(x) \approx \sum_{k=1}^n \frac{2}{2n-1} \left(\frac{x-1}{x+1}\right)^{2k-1}$$

Na slici 6 data je relativna greška te formule u funkciji argumenta x. Za tačnost 2^{-32} u intervalu x od $\sqrt{1/2}$ do $\sqrt{2}$ dovoljno je uzeti $n=6$.

Ekonomizacija potencijalnih (i drugih) razvoja je posebna tema. Kratko se može reći da se ovaj

mali broj sabiraka ($n=6$) može smanjiti, uz istu tačnost u zadanom intervalu, ako se izmene vrednosti koeficijenta.

Zaključak treće faze je jasan: treba koristiti formulu (6) ili odgovarajuću ekonomiziranu formulu.

Identifikovanje kritičnog intervala je lako ako za trenutak zamislimo da je argument x zadan sa nekom relativnom greškom p. Kao je

$$(7) \frac{\log(x(1+p)) - \log(x)}{\log(x)} = \frac{\log(1+p)}{\log(x)} \approx \frac{p}{\log(x)}$$

zaključujemo da se za ma kako malu vrednost p u okolini tačke $x=1$ teorijski relativna greška može proizvoljno velika. To je zato što log(x) ima nulu u tački $x=1$.

Okolina tačke $x=1$ je kritičan interval. U kritičnom intervalu vrednosti funkcije mora posebno biti brižljivo računata da bi bila vrednost logaritma izračunata sa pristojnom tačnošću. Kada su u pitanju tako često korišćeni programi, onda je za programera „pristojna tačnost“ tačnost do poslednjeg bita mantise rezultata.

Ovde se naravno podrazumeva da je argument x, ne neki zašljepi (obično decimalni) broj, već binarni sadržaj ćelije x. Uz tu pretpostavku može se dobiti rezultat kod koga je samo poslednji bit mantise neizvestan. To je moguće ako se razlika $x-1$ računa na suprotniji način, napr. $x-0.75-0.25$.

Objašnjenje tog na izgled nepotrebno komplikovanog oduzimanja jedinice leži u predstavljanju brojeva u računaru. Ako je argument x između $g = \sqrt{1/2}$ i 1 on ima karakteristiku 0, a broj 1 ima karakteristiku 1 ($1=0.5 \cdot 2^1$).

Oduzimanje $x-1$ obično nije tako valjano izvedeno (što se može) da se ne izgubi poslednji bit mantise broja x. Videli smo da je funkcija $\log(x)$ vrlo osetljiva na promenu argumenta u okolini tačke $x=1$; formula (7). A ovo je promena argumenta, i to potpuno nepotrebno izazvana nebrzišljivim oduzimanjem jedinice. Ako se (umesto 1) oduzme 0.75 ne čini se nikakva greška jer brojevi x i 0.75 imaju istu karakteristiku. Zatim se oduzima i 0.25, što takođe ne unosi grešku. Ako je x između 1 i $\sqrt{2}$, nije neophodno, ali ne smeta, da se $x-1$ realizuje sa $x-0.75-0.25$.

Šta se događa kada se $x-1$ nebrzišljivo izračuna prikazano je na slici 7. Kada x teži i odzdo relativna greška rezultata raste tako da broj tačnih bitova od J opada na 0.

Ako je x između $\sqrt{1/2}$ i $\sqrt{2}$, onda je $\log(x) = \log(f)$, jer je $i=0$, pa je član $i \log(2)$ takođe 0. Moduo člana $\log(f)$ je u svim drugim slučajevima mali u odnosu na član $i \log(2)$.

Ako i nije 0, onda je član $i \log(2)$ pretežan. Veći deo tog člana je

$$0.6875 \text{ mogućno je egzaktno izračunati, a ostatak je } i \log(2) = 0.6875$$

srazmerno mali. Optimalnu tačnost u izračunavanju vrednosti logaritamske funkcije mogućno je postići ako vrednosti $\log(f)$ dodamo $i \log(2) = 0.6875$, pa tek ovom zbiru glavni deo rezultata (koji je tačan) 0.6875 i.

U priloženim programima izvršena je ekonomizacija razvoja (6), tako da je $n=5$.

**UDRUŽENI
PROGRAMERI**

jezički procesori u 16 lekcija

Lekcija 1: Lekcije 2 do 14 daju početna znanja koja su neophodna, ali često nedovoljna za trgovinu i korišćenje jezičkih procesora (kada ga već kupite, možda ne bi bilo loše da možete da ga koristite i u druge svrhe sem da izazivate zavist kod suseda). Za svežije informacije, pogotovo pri konkretnoj kupovini, sledite Brenkin primer. Ako ne znate ko je Brenka, i šta radi, strpite se do lekcije 9.

Lekcija 2: U lekcijama 4 do 14 objašnjeno je kako možete da odaberete i kupite odgovarajući jezički procesor za vaš „personalac“.

Lekcija 3: Za preprodaju i ostala piratstva nema odgovarajućih lekcija. Izuzev lekcije 16, bar ne u „Računarima“, već upotrebite ličnu maštu i kreativnost, ako vi to tako nazivate.

Lekcija 4: Kada kupujete jezički procesor, imajte na umu da način prevodenja bitno utiče na brzinu izvršavanja programa, veličinu potrebne memorije, udobnost i brzinu prevodenja. Ako ste jedan od onih koji žele da uvek drže konce u svojim rukama, kupite i program koji vam daje mogućnost da kontrolišete efikasnost računara. Ovakvi programi obezbeđuju i opcije kao što su kontrola nad tipovima promenljivih koje se koriste (bit, logička ili celobrojna promenljiva) ili kontrola preciznosti. Neki kompajleri daju mogućnost da im se specificira i koji će se memorijski modul koristiti, tako da generišu efikasnije programe kada se zna da će kôd biti ograničene veličine.

Lekcija 5: Pri izboru jezičkog procesora ne treba zaboraviti da, pored potrebne memorije za kompajler i linker, treba obezbediti i mesto na disketi ili disku za datoteke sa izvornim programom, kao i datoteke koje se generišu pri prevodenju i linkovanju.

Lekcija 6: Standardi za jezike se primenjuju strožije kad su u pitanju veliki računari nego personalci. Ipak, i za personalce, posebno one namenjene prvenstveno profesionalcima, postoje određeni standardi: to su najčešće podskupovi ili varijante jezičkih procesora za velike računare.

Lekcija 7: Mnogi jezički procesori imaju proširenja za pristup operativnom sistemu.

Aristotelian Logicians
2631 East Pinchot Avenue
Phoenix, AZ 85016
USA
(602) 956-7678

BD Software
Box 9
Brighton, MA 02135
USA
(617) 782-8836

Per Brinch-Hansen
Computer Science Department
University of Southern California
Los Angeles, CA 90007
USA

Borland International
4987 Scotts Valley Drive
Scotts Valley, CA 95056
USA
(408) 438-8400

c-systems
Box 3253
Fallerton, CA 92634
USA
(714) 637-5362

C Ware
Box 710097
San Jose, CA 95171-0097
USA
(408) 736-6905

The Code Works
5266 Hollister, Suite 224
Santa Barbara, CA 93111
USA
(805) 683-1583

Computer Innovations
10 Mechanic Street
Red Bank, NJ 07701
USA
(201) 530-8995

Digital Research
Box 579
Pacific Grove, CA 93950
USA
(408) 649-3896

Ellis Computing
3917 Noriega Street
San Francisco, CA 94122
USA
(415) 753-8186

Harvard Associates
260 Beacon Street
Somerville, MA 02143
USA
(617) 492-6660

IBM Corporation
Box 1328
Boca Raton, FL 33432
USA
(800) 447-4700

Integral Quality
6265 26th Avenue
Seattle, WA 98115
USA
(206) 527-2918
(206) 527-2918

Krell Software
1328 Stony Brook Road
Stony Brook, NY 11790
USA
(516) 751-5139

Laboratory Microsystems
4147 Beethoven Street
Mar Vista, CA 90666
USA
(213) 386-7412

Lifeboat Associates
1651 Third Avenue
New York, NY 10028
USA
(212) 866-8300

The Lisp Company
Box 437
Redwood Estate, CA 95014
USA
(408) 354-3668

Logitech
1655 University Avenue
Palo Alto, CA 94301
USA
(415) 326-3085

Microsoft
10790 Northrup Way
Bellevue, WA 98004
USA
(206) 828-0800

Miller Microcomputer Service
61 Lakeshore Road
Matick, WA 91760
USA
(617) 653-6136

Modula-2 Research Institute
950 North University Avenue
Provo, UT 84604
USA
(801) 375-7402

Morrell Data Systems
3400 Wilshire Blvd., Box 78127
Los Angeles, CA 90010
USA
(213) 257-2926

No Visible Support Software
2080 Center Street, Box 1344
Berkeley, CA 94701
USA

R. R. Software (James Ada)
Box 1512
Madison, WI 53701
USA
(608) 244-6436

Ryan-McFarland Corp.
9057 Jowell Drive
Aptos, CA 95003
USA
(408) 662-2522

Silicon Valley Software
10340 Phar Lap Drive
Cupertino, CA 95104
USA
(408) 725-8890
(408) 725-8890

David Smith
44 Die Masket Lane
Danbury, CT 06810
USA

Snobol4
Box 6995
Rochester, NH 55903
USA

Softronics
36 Homestead Lane
Roosevelt, NJ 08555
USA
(609) 448-2685

The Software Toolworks
15233 Ventura Blvd., Suite 118
Sherman Oaks, CA 91403
USA
(213) 986-8885

SoftTech Microsystems
16685 Bernardo Drive
San Diego, CA 92127
USA
(619) 451-1230

Springer-Verlag New York Inc.
Computer Science Editor
175 Fifth Avenue
New York, NY 10019
USA

STDC Inc.
2115 East Jefferson Street
Richmond, MD 20784
USA
(301) 984-5000

TeleSoft
10639 Roselle Street
San Diego, CA 92121
USA
(619) 457-2700

Unified Systems Software
Box 2644
New Carrollton, MD 20784
USA
(301) 552-9590

Volition Station
Box 1236
Bel Mar, CA 92014
USA
(619) 481-2286

Richard Walters
University of California
Davis, CA 95616
USA
(916) 752-2793

Watsoft Products, Inc.
158 University Avenue W
Waterloo, Ontario
N2L 3E9 CANADA
(519) 886-3780

Western Digital
2455 McCabe Way
Irvine, CA 92714
USA
(714) 557-3550
(714) 557-3550

Na primer, možete da proverite veličinu vaše datoteke ili veličinu slobodne memorije na disku i da na njemu obistite, primenujete ili kreirate datoteku.

Lekcija 8: Odradjeni jezički procesori ne mogu da na svim (a Marfijev zakon lako može da učini da to bude upravo vaša) računarnima da adresiraju svu memoriju. Slični problemi se pojavljuju i sa stekom i područjem podataka.

Lekcija 9: Važan faktor pri kupovini jezičkog procesora su i prateća biblioteka i sredstva za razvoj programa i ispravljanje grešaka.

Lekcija 10: Koji jeziči su vam na raspolaganju? Ada, APL, bezjzik, C, kobol, fort, fortran, lisp, logo, modula-2, paskal, PL/1 i prolog.

• Brenka Doktorović •
• Dobrojska 231 •
• 23000 Kikinda •
• Jugoslavija •

• Znojbo 4 •
• Box 6995 •
• Rochester, NY 14610 •
• USA •

• February 28, 1986 •

Dear Sir:

I am very much interested in obtaining the Znojbo 4 language processor for my Commodore 64. Please send me at the above address the price list for such processor and all the information concerning the payment. If you have a catalog I would appreciate if you could send me one.

Sincerely,

• Brenka Doktorović •

Deo teksta iznad je zvezdica zamestite tako da odgovara vašim željama i potrebama.

Nekoliko kompanija nudi podskupove ade, ali jedino Western Digital nudi potpunu ade — na žalost, jedino za svoj hardver. Ako vas podkupovi zadovoljavaju, pišite firmama R. R. Software ili TeleSoft.

APL nije našao mnogo poklonika među korisnicima personalnih računara, ali ako vi mislite da je to ono pravo, IBM, STSC Inc. i Softronics vam nude interpretatore za 8-bitne i 16-bitne računare.

Bezjzik je još uvek jezik № 1 personalaca i možete ga nabaviti kod Microsofta i Digital Researcha. Iako je bezjzik predviđen prvenstveno za male aplikativne programe, tekuća ANSI verzija je modernizovala jezik i tako mu proširila polje primenljivosti.

Posle bezjzika jezik C je na neprikosnovenom drugom mestu. Kompleksne aplikacije koje su za 8-bitne personalne računare pisane isključivo u assembleru za 16-bitne mašine uglavnom se pišu u C-u. Uz C kompajler obično idu i operativni sistemi inspirisani UNIX-om. Za kupovinu pišite firmama BD Software, Microsoft, Digital Research, C Ware ili Computer Innovations.

Microsoft, Digital Research i Ryan McFarland prodaju kompajlere za kobol.

Fort, stek orijentisani jezik, uživa fanatičko oduševljenje među korisnicima personalnih računara od prvih dana svoje pojave.

Koristi se prvenstveno za pisanje programa za kontrolu, ali je našao i svoju ulogu u obradi teksta kao i programima za poslovne potrebe.

Fortran se na personalcima koristi prvenstveno da bi se omogućila upotreba velikog softvera koji je napisan za velike računare. Kompajleri su napisani i za 8-bitne i za 16-bitne personalne računare, a realizovane verzije su skoro potpuni podskupovi fortrana 77. Između ostalih, fortran kompajlere prodaju i Microsoft, Digital Research i Silicon Valley Software.

Pojava ekspertnih sistema, približila je veštačku inteligenciju i korisnicima personalnih računara. Jezički procesor lisp vam može pomoći da i vi zakoračate u ovu novu oblast, a možete ga kupiti kod Integral Quality Systems, Microsoft i The Lisp Company.

Među kompjuterskim edukatorima logo uživa nepodeljene simpatije. Potpuni kompajleri ga čine sve više i aplikativnim jezikom. Za potajnije kupce najbolje je da se obrate IBM, Digital Research i Krell Software-u.

Mnogi modulu-2 smatraju idealnim prelazom sa paskala na adu i za njih su Springer-Verlag, Logitech i Volition Systems obezbedili kompajlere.

Virtova definicija paskala proširena je ANSI i ISO standardima koji se mogu nabaviti kod Microsofta, Digital Researcha i SoftEcha.

Ako imate IBM PC XT/370 možete puštati 370 PL/1 programe. Ako nemate IBM PCa, a život bez PL/1 vam je nezamisliv, spašće vas Digital Research svojim podskupom G PL/1.

Za one koji ne vole lisp, a vole veštačku inteligenciju, Prologov britanski interpreter prodaje Springer-Verlag.

Lekcija 11: Sem pomenutih kompanija, jezičke procesore prodaju i druge firme i zato je najbolje da pišete nekoj od firmi iz tabele za šira obaveštenja.

Lekcija 12: Ako ste sujevni ili znate engleski, preskočite lekciju 13.

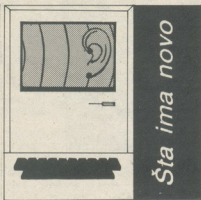
Lekcija 13: Dok ne utvrdite svoje znanje engleskog, koristite odgovarajuće adaptirano pismo Brenke Doktorović. Dolaze teška vremena — ne dozvolite da vas preteknu žene.

Lekcija 14: Jezički procesori iz lekcije 14, mogu da koštaju i do 2000 dolara pa i više, iako se retko sreću čine veće od 500 dolara. Za 50 ili manje dolara Microsoft prodaje bezjzik, Software Toolwork i The Code Works C. Ellis Computing kobol i fortran, Aristotelian Logicians, No Visible Support Software, Unified Systems Software prodaju Forth, Norell i Software nude lisp, a Modula-2 Research Institute, pogodili ste, modulu-2! Paskal se može kupiti kod Borgland Internationala, a kupcima snobila preporučujemo da se pridruže Brenki.

Lekcija 15: Pored firmi i veliki broj pojedinaca prodaje razne specijalne jezičke procesore. Pored Hansena, Dejvida Smita i Ričarda Valtersa, na tom spisku su i Horovic, Susman i drugi velikani računarnstva, ali i široj javnosti nepoznati pojedinci. Spisak njihovih adresa se priprema i, ([kao odobri urednik]), „Računari“ će ga objaviti kad bude gotov. Za sad se morate zadovoljiti oficijelnim adresama iz tabele.

Lekcija 16: Pišite nam ako ste autor nekog od jezičkih procesora. Sa zadovoljstvom ćemo vas uvrstiti u tabelu autora.

Nada Aleksić



Million po milion — posao

Čuje se da Ser Klav Sinkler ima nekih manjih problema sa prikupljanjem deset miliona funti koliko mu je potrebno za izradu sledeće serije računara koje je zamislio. Zar je moguće da mu finansijeri ne veruju?

Poljubac za skor

Dok ovo budete čitali, ovogodišnji FEST će već biti završen. Neki od vas će na njemu, verovatno, gledati i američki film „Povratak u budućnost“. To je još jedan iz niza uspešnih filmova koje je finansirao Stiven Spielberg. Ono što je za nas interesantno je da postoji i video igra napravljena po tom filmu. U toj igri se po prvi put pojavljuje (ako ne računamo honkoške porno igre) poljubac! Da, dobro ste pročitali, poljubac u video igri! Verovatno ćemo još čuti o toj igri.

Hit lista

Da vas obavestimo da se na top listi „Your computer-a“ na prvom mestu nalazi „Elite“ o kojoj ćete malo detaljnije pročitati u članku Dejana Ristanovića.

Novosti od Hjusova

Ako vam se puno sviđeo Paradirol, znajte da se upravo sprema još jedna igra sličnog dizajna i slične zanimljivosti. Ime joj je Uridjilm i prvi reklamni skrinovi koji su je u štampi pojavili izlaze čine da pomislite da se vaš „komodor 64“ pretvorio u pravu arkanud mašinu. Čim se pojave prve kopije dobićete prikaz.

Brljivi „mek“

Ako imate „Macintosh“, obavezno nabavite Smooth talker — program koji omogućava „meku“ da svaki tekst ispisan na ekranu izgovori jasno i glasno preko svog zvučnika. Za samo 95 dolara, pod uslovom da ste nabavili verziju 2.0, imate mogućnost da birate muški ili ženski glas. Ali ovaj... niko nije idealan — muški glas govori sa jakim kanadskim akcentom, dok je ženski pol po poreklu iz Škotske. Ali, ako možete da predete preko takvih nesavršenosti programa, naći ćete ga u svakoj boljoj robnoj kući, trafici ili apoteci... u Kaliforniji.

I.N.

Najveći konkursi sveta

konkurs „Abakus“

Konkurs, da li ne? Pre svega kakav? Najbolji su, svakako, oni sa nagradama od desetak hiljada dolara ili više, ili oni koji donose izuzetnu slavu, zar ne? I šta čekaju ti „Računari“? Što ga već jednom ne objave? Dok „Računari“ ne skupie 100.000 dolara za nagradni fond, ili ne ubede Virla ili Hariana Milsa da učestvuju, a vi ih pobeđite, pogledajmo šta nudi ostatak sveta. U skladu s pravilima čemo, s vremena na vreme, objavljivati najbolje svetske konkurse u skladu sa sledećim pravilima:

1. Konkurs mora biti otvoren još najmanje 6 meseci posle objavljivanja u „Računari“.
2. Nagrada na konkursu moraju da budu kao u tački 3. ili 4.
3. Nagrada mora biti bar 10.000 dolara, što više to bolje, ili kao u tački 4. Nagrada mora biti takva da pobeđniku i svim ostalima nedvosmisleno ukazuje na to da je pobeđnik bolji programer od nekog dobitnika Turingove nagrade, bar kad je u pitanju nagradni izdatak.
5. O svim pobeđnicima, čitaocima „Računara“, „Računari“ objavljuju reportaže i upoznaju javnost sa našim superprogramerima.
6. Posebna nagrada pripada onome ko nas obavesti o nekom konkursu koji potom objave „Računari“.

```
2 REM "KONKURS CASOPISA ABACUS"
3 REM "B.A. TEAGUE ZA TRS 80 III"
5 DIM ARRAY$(9)
7 GOSUB 500
9 DEFB L B-H,J-Z
10 N=11111
20 S=N*N
30 SS=STR$(S)
40 X=LEN(SS)-1
45 IF X > 9 THEN 300
47 GOSUB 500
50 FOR I=1 TO X
60 CS=MID$(SS,I+1,1)
65 C=VAL(CS)
67 ARRAY$(C)="OK"
70 NEXT I
80 FOR A = 1 TO 9
90 IF ARRAY$(A)="BAD" THEN N=N+1: GO TO 20
100 NEXT A
110 LPRINT "N=";N;" S=";S$;"
115 N=N+1:GOTO 20
300 END
499 REM "POTPROGRAM"
500 FOR A=1 TO 9
510 ARRAY$(A)="BAD"
520 NEXT A
530 RETURN
```

7. Sva pravila u vezi sa konkursima menjaju se i proširuju isključivo u skladu i dogovoru sa čitaocima.

Ako znate za neki konkurs koji zadovoljava uslove za objavljivanje, pišite nam. Dok ne dobijemo vaše predloge, mi ćemo objaviti nekoliko konkursa koji su nama poznati: Prvi na listi je konkurs „Abacus“-a, izvanrednog lista za popularizaciju računarstva, koji izdaje Springer-Verlag. (Za one koje interesuje pretpiata, adresa izdavača je: Springer-Verlag New York Inc., 175 Fifth Avenue, New York, NY 10010 USA.) Nagrade na konkursima su skromne, pretpiata na „Abacus“-u, par knjige sa rešenim zadacima iz računarstva, ali su zato redovni učesnici na ovom konkur-

su, recimo, Gris i Knut, a njih pobeđiti nije šala! Zadaći su takvi da omogućavaju da konkurs bude stalno otvoren i svako novo rešenje bilo kog zadatka se priznaje pod uslovom da ono ranije nije publikovano na stranicama „Abacus“-a. Zadaći se numerišu i mi ćemo ih numerisati kao u „Abacus“-u tako da kad ih rešite možete ih poslati Richard-u V. Andree-u, University of Oklahoma, USA ili uredništvu „Abacus“-a.

Problem 1: Odrediti sve pozitivne cele brojeve tako da kvadrat $S=N*N$ sadrži 9 cifara (1, 2, 3, 4, 5, 6, 7, 8 i 9) tačno jednom.

Problem 2: Naći sve pozitivne brojeve takve da je suma kubova njihovih cifara jednaka tom broju? Primer: $371 = 3^3 + 7^3 + 1^3$

Ovaj zadatak je rešio Dejvid Gris Red profesor na Kornelju, autor divne knjige „Nauka programiranja“, navlaš rešenja 1, 1634, 8208 i 9474. Možete li bolje?

Problem 3: Znate li šta je palindrom? Na primer: Ana voli Milovanu ili Udovica baci vodu. Unapred i unazad se čita isto. Znate li neki engleski? Able was ere i saw Elba ili Norma is as setless as I am, Ron. A da li znate neki kvadrat koji se čita unapred i unazad isto? Sa neparnim brojem cifara: $11 \times 11 = 121$
 $26 \times 26 = 676$
 $264 \times 264 = 69696$
 $22865 \times 22865 = 522808225$
a sa parnim: $836 \times 836 = 698896$
Da li znate neki treći stepen koji je palindrom? Na primer: $7^3 = 343$
 $2201^3 = 1066256601$
A za četvrti stepen? $101^4 = 104060401$
Nadite isto za 5, 6 i 7 stepen. Nudimo nekoliko rešenja čitalaca za kvadrat: $798644^2 = 63783238736$
 $64030648^2 = 40999238829904$
I, zasao, najveći: $63083295898^2 = 4046011955111$
 8611559106404

Dovoljno?
Problem 7: Znate li da rešite Hanojsku kulu: imate tri štapa — a, b i c — i na štapa u a kolovutava tako da je najmanji na vrhu, a veći redom ispod, do najvećeg koji je na dnu. Treba ih premostiti na štap c, koristeći štab b kao pomoćnik. Rešite isti zadatak kad su a) diskovi alternativno crveni i beli, a veličinski odnos kao malopre i b) jedan disk se sme prebaciti ili na prazan štab ili na veći disk, ali samo ako je štab druge boje, i rešili ste problem 7.

N. Aleksić i M. Potkonjak

Kolumbovo jaje

Softverski perpetuum mobile

Prenos programa sa jedne mašine na drugu predstavlja problem sa kojim se suočava veliki broj ljudi koji se profesionalno ili iz hobija bave računarstvom. I taiku je jasno da ne u vide glatko čak ni kada se radi o vrlo sličnim računarima. U računarstvu je, međutim, iz-

gleda sve moguće — čak i to da se program za jedan računar izvršava na nekom sasvim drugom.

Metod koji ovde opisujemo Amerikanci zovu „bootstraps“. Za one kojima engleski nije jača strana recimo da bootstrap znači šare na čizama, pa bi bootstrapping bio postupak kad čovek uzima samog sebe za šare od čizama i podigne uvis!

Pretpostavimo da posedujemo mnogo softvera u izvornom kodu (neprevedenom obliku) nekog jezika L i mašinu M1 sa kompajlerom C1 za taj jezik. Između ostalog, imamo i izvorni neprevedeni oblik kompajlera C1 pisan takođe u jeziku L.

Sors verziju (izvorni kod) nekog programa označavamo sa Ps, a prevedenu verziju (objektni kod) sa Po.

Pretpostavimo da smo nabavili sasvim novu mašinu M2 i tanki smo sa softverom za nju. Ako bismo nekako uspeali kompajler za jezik L da prebacimo

na računar M2, automatski bi i na M2 imali puno softvera (svaki Po bi preveli na M2 uz pomoć tog kompajlera).

Setimo se da naš kompajler C1 imamo i u sorsu C1s. Zato nam neće biti baš previše teško da ga prepravimo tako da umesto koda mašine M1 generiše kod mašine M2 (narodiće ako smo baš mi napisali i kompajler C1s). Čak i ako nam taj posao oduzme prilično vremena, to je još uvek jednostavnije nego da za M2 pišemo sve programe u mašincu.

Program pisan u jeziku L koji pravi kod za M2 nazovimo C2s. Lepo je to što sada imamo C2s, ali kako da ga prevedemo na mašinski jezik mašine M2?

Za to ćemo se poslužiti starijim računarom M1. Uz pomoć njegovog kompajlera C1 prevećemo program C2s.

Dobijeni objektni kod C2o može da se izvršava na M1. Uočimo da je C2o program čiji je ulaz neki program pisan u jeziku L, a izlaz mašinski kod računara M2.

Sada ćemo na M1 pustiti program C2o, a kao ulaz podmetnućemo mu baš C2s!!! Dobijeni izlaz C2i je prevod kompajlera C2s na mašinski kod računara M2, tj. moguće ga je izvršavati na M2.

To je upravo ono što nam je i bilo potrebno — kompajler jezika L na mašini M2.

Neverni Toma će reći — ovo je fenomenalna stvar, ali samo pod uslovom da u prepravci kompajlera C1s u C2s nismo nigde pogrešili, što je malo verovatno. To, međutim, i nije neki problem, jer imamo mogućnost jednostavnog testiranja kvaliteta novog kompajlera!

Dokazivanje korektnosti i mnogo trivijalnih programa od jednog kompajlera je mukotpan posao, ali opet ćemo koristiti strašan trik. Uz pomoć C2t prevećemo kompajler C2s. Ako u pisanju C2s nismo omanuli, kao izlaz ćemo dobiti C2t. Jednostavno, zar ne!

Zoran Obradović

Programiranje
u bejziku

svirka na „amstradu“

Računar „amstrad“/šnajder CPC 464“ je opremljen jednim od najboljih zvučnih čipova na tržištu AY-3-8912, koji istovremeno proizvodi tri tona plus šum. Ako se zna da je zvuk sa CPC 464 (kao i na drugim amstrad/šnajderovim računarima) namenjen i za stereo reprodukciju, onda je jasno da je ova mašina prava stvar za kompozitore na kućnom kompjuteru. To nije slučajno, jer je fabrika Amstrad inače poznati proizvođač solidnih audio uređaja.

Za svaku primjenu zvuka na računala je vrlo važno na koji način ga možemo ostvariti, tj. da li se zvuk ostvaruje pomoću jednostavnih naredbi, ovdje u solidnom Locomotive BASIC-u, ili tek pomoću strojnih programa. Kod računala CPC 464 ili naprednijih varijanti (664 i 6128) manipulacija zvukom je u velikoj mjeri moguća upravo iz bejzika, što predstavlja olakšanje početnicima, ali i naprednijim programerima. Koliko god ta upotreba bila jednostavna, zbog nesnaalaženja u osnovnom priručniku zvukove mogućnosti ovog računala mogu ostati neiskorištene, pa čak i za korisnike s minimalnim zahtjevima. Tema ovog napisa je kako što bolje iskoristiti zvukovne mogućnosti. Do prave svirke može doći tek nakon sistemskog upoznavanja sa svim osnovnim aspektima upotrebe zvuka za konkretne namjene.

Kod-računala CPC 664 se zvuk reproducira preko ugrađenog zvučnika, a regulacija glasnoće se vrši pomoću potenciometra s desne strane računala. Ukoliko nas taj zvuk ne zadovoljava, (relativno ih, izobličena), moguće je koristiti ulazno-izlazni (Input-Output) port koji omogućuje priključenje na stereo Hi-Fi liniju ili klasično audio pojačalo i tako dobiti jači i kvalitetniji zvuk. U praksi to izgleda da se jedan kanal reproducira preko lijevog izlaza, drugi preko desnog, a treći, u stvari, daje „srednji kanal“. Zvučni izlaz je dosta niske razine, pa pri radu s nekim pojačalima može biti problema.

SOUND

Glavna instrukcija za rukovanje generatorom zvuka je instrukcija SOUND, koja se može koristiti uz ostale instrukcije, tako da daje veće mogućnosti primjene nego kada je koristimo samu za sebe. Instrukcija SOUND je određena sa sedam parametara, od kojih je prvi tzv. status kanala. Status kanala se određuje prema sljedećoj tablici:

određivanje statusa kanala

bit decimalno značenje

0	1	odabire kanal A
1	2	odabire kanal B
2	4	odabire kanal C
3	8	spajanje s kanalom A

4	16	spajanje s kanalom B
5	32	spajanje s kanalom C
6	64	održavanje tona-HOLD
7	128	treperenje tona-flushing

Dakle, prva tri parametra odabiru kanal — za generiranje tona s kanala A odabiramo 1 za prvi parametar, ili, za zvuk sa sva tri kanala — 7 ($1+2+4=7$). Ostali parametri će biti kasnije objašnjeni. Sljedeći parametar u instrukciji SOUND određuje visinu tona. U osnovi, generator zvuka koristi kloak 4 MHz čiju frekvenciju dijeli sa 16, te daje konačnu frekvenciju od 125 kHz. Ona se, zatim, dijeli sa brojem — veličinom parametra koji se koristi za visinu tona, tako da veći broj daje veći omjer, tj. nižu izlaznu frekvenciju. Visina tona koja je zadana kao 4 ili manje daje izlaznu frekvenciju koja je na granici čujnosti ljudskog luha (okvirno 20 Hz—16 kHz), tako da samo oni s dobrim sluhom mogu čuti frekvencije proizvedene s računala koje su definirane vrijednošću parametara 5—8. S druge strane, najveća vrijednost parametra visine tona je 4095, što daje rezultirajuću frekvenciju od cca. 30 Hz — dosta duboki bas ton. Kod takvih frekvencija se mora suočiti s činjenicom da mali zvučnik ugrađen u računalo ne može reproducirati vrlo niske zvučne frekvencije točno onako kako bismo željeli, odnosno dolazi do manjih ili većih izobličenja.

Trajanje tona određuje treći parametar u odnosu na jednu stotinku sekunde. Na taj način su omogućene varijacije zvučnih efekata kao i muzika koja se može lakše proizvesti. Vrijednost 32767 određuje najduže trajanje koje, u krajnjem slučaju, može biti 5 minuta.

Četvrtim parametrom se kontrolira jačina zvuka — ako se koristi samo instrukcija SOUND bez instrukcije ENV, vrijednost parametra je od 0, koja isključuje generator,

do 7, što daje najveću jačinu. Kada se instrukcija ENV koristi uz SOUND, raspon jačine se proširuje od 0 do 15. To bi bilo najosnovnije uz instrukciju SOUND, jer se momentalno ostala tri parametra mogu postaviti, tj. pridružiti im vrijednost 0.

Kao primjer se može uzeti instrukcija: SOUND 1,250, 200, 4, 0, 0, 0 koja znači da će kanal A proizvesti visinu tona (određenu s 250) u trajanju od 2 sekunde (200:100) nivo jačine 4. Za potpuniju ilustraciju može se prikazati rutina: 15 SOUND 2,150, 200, 4, 0, 0, 0 25 SOUND 7,150, 200, 4, 0, 0, 0 koja prvo proizvodi zvuk s kanala B visine tona zadane s 150 u trajanju 2 sekunde, a potom to isto proizvodi sa sva tri kanala. Može se vidjeti da je jedina razlika između jednog kanala i sva tri samo u jačini zvuka.

Kao primjer se može izvesti jednostavan generator tona koji reproducira vrijednosti visine tona od 1 do 4095, a to se postiže petljom:

```
15 FOR A=1 TO 4095
25 SOUND 1, A, 1, 7, 0, 0, 0
35 NEXT A
```

Za bolje razumijevanje generiranja tonova preporučljivo je da početnici koriste neku literaturu iz audio-tehnike.

Na sličan način se zvučni efekti mogu proizvesti rutinom, koja uz modifikacije, može imati primjenu pri razvijanju vlastitih programa:

```
15 FOR X=20 TO 80
25 SOUND 1, X, 1, 7, 0, 0, 0
35 NEXT X
```

Ova rutina, jednostavno, daje visine tonova definirane od 20—80 minimalnog trajanja od jedne stotinke sekunde.

Varijacije jačine proizvedenog zvuka se mogu nazvati i oblikovanjem zvučnih ovojnica-anvelopa (engl. envelope). Na primjer, smanjivanje jačine zvuka se može postići rutinom sljedećeg oblika:

15 A=7
25 FOR B=20 TO 80
35 SOUND 1, B, 1, A, 0, 0, 0
45 A=A-0.1
55 NEXT B

Linija 15 postavlja jačinu zvuka na 7, a linija 45 smanjuje tu vrijednost za 0,1 pri svakom koraku petlje. Računalo može ponekad zablokirati (crashing) ako parametar jačine zvuka (u ovom slučaju varijabla A) nije cjelobrojna veličina.

MUZIKA

Na amstrad/schneideru je moguće programirati muzike u trodijelnoj harmoniji. Međutim, oni koji nisu vični komponiranju treba da koriste što jednostavnije muzičke izraze. Također im na raspolaganju stoje solidni programi koji olakšavaju komponiranje. CPC 464 ima impresivan raspon moguće upotrebljivih nota — točnije 8 oktava. Kao što je ranije rečeno, broj čujnih visokih frekvencija je relativno ograničen.

To praktično znači da nam na raspolaganju stoji čujan raspon od 5 oktava koji se proširuje ako računalo priključimo na kvalitetniji audio uređaj. Računalo ne možemo programirati direktno u muzičkim vrijednostima, već moramo notama pridružiti određene vrijednosti za visinu tona, što je opisano u VII poglavlju priručnika koji se dobija uz računalo. Ostvarivanje jednostavne melodije može se vidjeti na sljedećem primjeru:

```
15 A=1
25 WHILE A()
35 READ A, B, C
45 SOUND 1, A, B, C, 0, 0, 0
55 WEND
65 DATA 239, 50, 5, 213, 50, 5, 190, 50,
5, 179, 50, 5, 159, 50, 5, 142,
50, 5, 127, 50, 5, 119, 100, 7,
0, 0
```

Ovdje se koristi petlja WHILE... WEND pomoću koje je melodija kompletnira. Željena visina tona, trajanje i jačina se dobijaju iz podataka u DATA liniji. Program je zaključen korištenjem .nule za vrijednost visine tona, ali treba paziti da trajanje i jačina budu uključeni, ili će u protivnom program zablokirati u liniji 35.

Spajanje kanala

Prvi parametar u instrukciji SOUND određuje odabir kanala, ali je moguće upravljeti i njihovim spajanjem. Drugim riječima, moguće je odjednom upravljeti sa više kanala. Za spajanje kanala upotrebljavaju se sljedeće vrijednosti:

8 spajanje s kanalom A
16 spajanje s kanalom B
32 spajanje s kanalom C

Tako npr. za slanje zvuka na kanal C i spajanje s kanalom A i eventualno održavanje tona postizemo sa: $4+8+64=76$, što unesimo kao prvi parametar instrukcije SOUND.

Kako funkcionira spajanje kanala može se vidjeti iz primjera:

```
15 SOUND 12, 100, 50, 5, 0, 0, 0
25 FOR A=1 TO 1000: NEXT A
35 SOUND 33, 200, 50, 5, 0, 0, 0
```

Linija 15 je SOUND instrukcija za kanal C koji se spaja s kanalom A ($4+8=12$). Linija 25 proizvede malo kašnjenje, a linija 35 je SOUND instrukcija za kanal A spojen s

kanalom C ($1+32=33$). Sljedeća rutina prikazuje praktičnu upotrebu spajanja:

```
15 SOUND 1, 120, 50, 5, 0, 0, 0
25 SOUND 2, 50, 100, 5, 0, 0, 0
35 SOUND 17, 100, 50, 5, 0, 0, 0
45 SOUND 10, 150, 50, 5, 0, 0, 0
```

Prve dvije linije proizvode tonove sa kanala A i B, ali je trajanje različito za oba kanala. Sljedeće dvije linije predstavljaju također instrukcije SOUND za kanala A i B, ali nose u sebi brojeve statusa kanala koji prouzrokuju spajanje. Prema ovom primjeru, moguće je pisati dvodjelne ili trodijelne harmonije. Spajanje kanala predstavlja, u stvari, točnu sinhronizaciju muzičkih dionica.

Održavanje tona

Održavanje tona moguće je postići davanjem vrijednosti 64 u vrijednost statusa kanala. Primjer:

```
15 SOUND 65, 100, 100, 5, 0, 0, 0
25 SOUND 66, 150, 100, 5, 0, 0, 0
35 SOUND 68, 250, 100, 5, 0, 0, 0
45 FOR A=1 TO 1000: NEXT A
55 RELEASE 7
```

Instrukcija SOUND u linijama 15, 25 i 35 odnose se na sva tri kanala A, B i C kojima je dodana vrijednost 64 za postizanje održavanja tona. Linija 45 proizvodi kašnjenje od otprilike 1 sekunde prije nego što dođe do otpuštanja (RELEASE). Instrukcija RELEASE pomici održavanje na označeni kanal. Kod instrukcije RELEASE se odabir kanala postize kao i prvi parametar u instrukciji SOUND — pridruživanjem određenih vrijednosti statusa kanala. Ukoliko je potrebno „otpuštiti“ više kanala, njihovi brojevi statusa kanala se jednostavno zbroje — u prethodnom primjeru: RELEASE 7 — ($1+2+4$).

„Flushing“

Posljednja karakteristika u instrukciji SOUND koji se može definirati u statusu kanal je „flushing“. Vrijednost koja se upiše u prvi parametar SOUND instrukcije iznosi 128. Sljedeći primjer ilustrira efekt flushing:

```
15 FOR A=1 TO 60
25 SOUND 130, 1, 1, 0, 0, 0, 0
35 NEXT A
```

Petlja se ponavlja 60 puta, a parametar za dužinu trajanja u instrukciji SOUND daje ton u trajanju od 1 sekunde za svaki korak petlje. Vrijednost statusa kanala 130 je dobivena iz upotrebe kanala B i flushing $2+128=130$). Jedna od namjena flushing je i ta da isključuje jedan ili više kanal generatora zvuka, što se može vidjeti na primjeru:

```
15 SOUND 65, 200, 100, 7, 0, 0, 0
25 SOUND 129, 1, 1, 0, 0, 0, 0
35 RELEASE 1
```

Programska linija 15 generira ton trajanja 1 sekunde sa držanjem tona u kanalu A, a linija 25 vrši flushing kanala A i ne proizvede iz zvuk jer je parametar jačine zvuka jednak nuli. Linija 35 otpušta kanal A.

ŠUM

Šum (noise) može biti proizveden i mješavinom u jednom ili u više tonskih kanala, ali istovremeno može biti proizveden samo jedan zvuk šuma. U principu nije ni potrebno da se neki ton mješa sa šumom, a koristiće malu vrijednost (1 ili 2) visine tona proizvedeni ton može biti nečujan tako da samo šum bude prisutan na izlazu. Šum je obično pištiće, a visina tona može biti

određivana po želji. Šum je moguće aktivirati koristeći vrijednosti od 1 do 31 i to kao posljednji parametar instrukcije SOUND. Ako se koristi vrijednost 0, šum je potpuno potisnut. Vrijednost 31 daje minimalnu visinu tona, a 1 maksimalnu. Relativno širok raspon visine tonova šumova je koristan za različite zvučne efekte. S obzirom da se šum koristi uglavnom kod igara, najčešće se upotrebljavaju oblici koji liče na eksplozije i pucnjeve, što prikazuje naredni primjer:

```
15 A=12
25 FOR B=7 TO 0 STEP -1
35 SOUND 2, 1, 20, B, 0, 0, A
45 A=A+2
55 NEXT B
```

Instrukcija SOUND u liniji 35 proizvodi zvuk na kanalu B, ali je ton previsok da bi se čuo. Trajanje je jedna petina sekunde pri svakom koraku petlje. Jačina i visina tona i šuma se upravlja varijablama A i B. Vrijednost varijable B (7) se postupno umanjuje sve do 0, što daje realističnu simulaciju zvuka eksplozije. Linija 45 smanjuje visinu šuma, dajući na kraju minimalnu visinu. Prethodna rutina se može modificirati za npr. simulaciju pucnja, samo je potrebno koristiti veći opseg visine šuma te odrediti kraće trajanje zvuka; npr:

```
15 A=3
25 FOR B=7 TO STEP -3
35 SOUND 1, 1, 3, B, 0, 0, A
45 A=A+1
55 NEXT B
```

Primjer rutine za šum može imati i oblik:

```
15 FOR X=1 TO 31
25 SOUND 1, 0, 100, 4, 0, 0, X
35 NEXT X
```

ili:

```
15 FOR X=1 TO 2
25 FOR Y=16 TO 31
35 SOUND 1, 0, 5+Y, 5, 0, 0, Y
45 NEXT Y
55 FOR Y=31 TO 12 STEP -1
65 SOUND 1, 0, 7, 6, 0, 0, Y
75 NEXT Y
85 NEXT X
```

Iz ovih primjera se vidi da u instrukciji SOUND može biti unesen i neki matematički izraz — linija 35.

Najraznovrsniji zvučni efekti se mogu postizati varijacijama upotrebe naredbi za generiranje zvuka i njihovih parametara, te se na taj način ostvaruje široko polje primjene tonskih mogućnosti računala amstrad/schneider.

ENV

Kod mnogih primjena instrukcija SOUND uz korištenje petlje daje zadovoljavajuće rezultate! Međutim, tamo gdje je potrebno složenije oblikovanje jačine zvuka ili visine tona, potrebno je uz instrukciju SOUND koristiti i instrukcije ENV (envelope — volume/ovojnica-jačina) ili ENT (envelope-tone/ovojnica-ton). Prvi parametar instrukcije ENV je identifikacioni broj (od 1 do 15) koji omogućuje željenu ovojnicu koja će biti pozvana instrukcijom SOUND (njezin prvi parametar). Ostala tri broja kod instrukcije ENV određuju broj koraka, povećanje ili smanjivanje jačine zvuka za svaki korak, te trajanje (u stotinama sekunde) svakog koraka. Broj koraka mora biti u rasponu od 0 do 127, dok koraci s obzirom na jačinu zvuka mogu biti između -128 i $+127$. Uobičajeno se koristi raspon od -15 do $+15$. Maksimalno trajanje po koraku je 255. Kao jednostavni primjer za smanjivanje jačine od 15 do 0 u petnaest koraka,

trajanja deset stotinki sekunde, može poslužiti instrukcija: ENV 1,15,—1,00. Za tvorbu ovojnice se u praksi koristi do 5 sekcija-odsječaka. Pri tome u svakoj sekciji mora biti naznačen broj koraka, veličina koraka i trajanje koraka. Time se mogu postići složene ovojnice.

Na slici 1. se može vidjeti zvučna ovojnicca istog oblika kao što je npr. ima pianino ili neki slični instrumenti — brzi početni uspon od nule do pune jačine, potom jačina brzo opada na neku srednju vrijednost koja se zadržati neko vrijeme, a potom jačina polako slabi do nule. U stručnoj literaturi se takva ovojnicca naziva ADSR ovojnicom (attack, decay, sustain, release—negli skok, propadanje, održavanje, opuštanje).

U praksi to prema slici 1. izgleda: Zvuk skače u pet stotinki sekunde na svoju punu jačinu. To zahtjeva sa 15 nivoa jačine pet koraka minimalnog trajanja, te veličinu koraka od +3. Prvi dio instrukcije ENV ovdje ima oblik:

ENV 1, 5, 2, 3, 1

Sljedećih 20 stotinki sekunde jačina tona pada za jedinice što zahtjeva četiri koraka od 5 stotinki sekunde, pa jačina opada za jedinicu po koraku.

Naredna sekcija trajanja 50 stotinki sekunde ima samo jedan korak u kojem ne postoji nikakav pad jačine tona. Posljedni, četvrti korak obuhvaća jedanaest pojedinačnih koraka koji predstavljaju padove jačine zvuka. Svaki traje 10 stotinki sekunde. Cijela bi se ovojnicca dala definirati instrukcijom:

ENV 1, 5, 3, 1, 4, —1, 5, 1, 0, 50, 11, —1,10

25 READ A, B

35 IF A=0 THEN 75

45 SOUND 1, A, B, 0, 1, 0, 0,

55 GOTO 25

65 DATA 239, 50, 213, 50, 190, 50,

179, 50, 159, 50, 142, 50, 127,

50, 119, 185, 0, 0

Vrijednosti u instrukciji DATA predstavljaju parove nota i dužine trajanja tonova. Parametar jačine tona ne treba biti uključen u instrukciju DATA.

Važna činjenica je ta da svako trajanje pojedine note mora biti određeno u instrukciji SOUND. To je moguće samo onda ako je trajanje naznačeno u instrukciji SOUND manje od postavljenog u instrukciji ENV ili ako ENV daje konačnu vrijednost jačine tona različitu od nule.

ENT

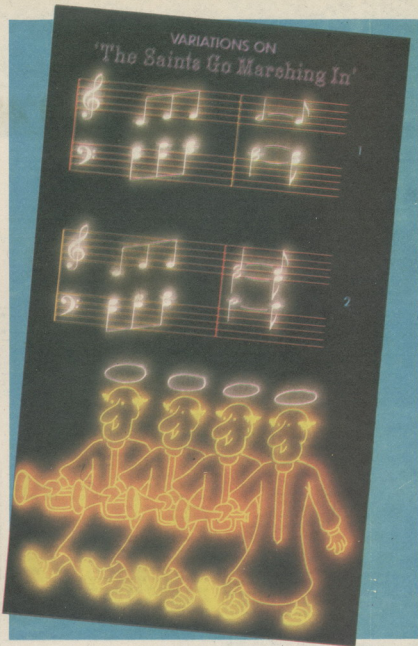
Instrukcija ENT obavlja svoju funkciju na sličan način kao i ENV. Prvi parametar je broj od 1 do 15 koji identifikira instrukciju. ENT se poziva instrukcijom SOUND, točnije njezinim šestim parametrom. Kod instrukcije ENT ponovo se koriste tri broja za definiranje karakteristike svake sekcije razvoja ovojnice, a može se koristiti do 5 sekcija. Prvi broj postavlja broj koraka (0 do 239), drugi postavlja broj visina tonova u svakom koraku (—128 do +127), a treći upravlja trajanjem svakog koraka (0 do 255) prema jednoj stotinki sekunde.

Može se prikazati primjer koji oponaša zvuk policijske sirene:

15 ENT —1, 20, 2, 1, 20, —2, 1

25 SOUND 1, 50, 1000, 6, 0, 1, 0

Prvi broj je negativan, što znači da daje automatsko ponavljanje zvučne ovojnice.



Pri korištenju pozitivnog broja ovojnicca se izvršava samo jednom, a visina tona ostaje na svojoj konačnoj vrijednosti sve dok je određuje instrukcija SOUND. Vrijednost u instrukciji ENT daje 20 povišenja visine tona koji bude povećan za 2 pri svakom koraku. Svaki korak ima minimalno trajanje od jedne stotinke sekunde. Tri naredna parametra, jednostavno, okreću taj proces, odnosno visinu tona vraćaju na originalnu veličinu. Inicijalna veličina visine tona je postavljena veličinom visine tona u instrukciji SOUND. U tom slučaju veličina visine tona se kreće prema gore do 40, a potom dolje sa 40 naniže, dajući potreban kontinuitet snižavanja i povišavanja visine tona. Ako je parametar vremena trajanja korišten u SOUND instrukciji prekratak za tvorbu ovojnice tonova, signal je unatoč tome

određen nakon definiranja trajanja u instrukciji SOUND ENV i ENT je moguće zajedno koristiti sa instrukcijom SOUND, tako da npr. prijašnji program može biti nadopunjen linijom 10 i nešto izmjenjenom linijom 45:

10 ENT —1, 1, 1, 4, 2, —1, 4, 1, 1, 5

45 SOUND 1, A, B, 0, 1, 1, 0

Instrukcija ENT na jednostavan način varira visinu tona sa plus i minus 1, dajući vrlo dobar efekt vibrata.

Ovo je bilo osnovno upoznavanje s instrukcijama za korištenje zvuka sa računala amstrad/schneider, a za potpuno savladavanje tih zvučnih mogućnosti poželjno je koristiti što više literature i dobrih profesionalnih programa.

Zvonimir Vistrčka, dipl. Ing

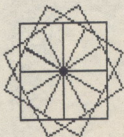
procedure u akciji

Logo je jezik kojim treba da se služe najmlađi, pa je i definisanje procedura jednostavno. Navodimo, najpre, službeni reč TO zatim ime procedure i, u sledećim redovima, njenu definiciju koju završavamo sa END. Procedura za iscrtaavanje kvadrata bi, na primer, mogla da se napiše kao:

```
TO drawsquare
PENDOWN
REPEAT 4
[FORWARD 100 LEFT 90]
END
```

Procedura se poziva jednostavnim navođenjem imena: sa 'drawsquare' ćemo navesti računaru da iscrta kvadrat počevši od tekuće pozicije „puža“, dok ćemo uz tek malo složeniji program dobiti prilično efektanu sliku 1.

slika 1:

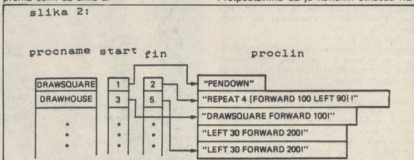


```
REPEAT 12
[ DRAW SQUARE LEFT 30 ]
```

Definisanje procedura

Da bismo na što efikasniji način implementirali procedure, podsetimo se našeg ranijeg napisa „Lične stvari bejzika“, u kome smo opisali različite mogućnosti generisanja tabela sa imenima promenljivih. Potrebno je, najpre, oformiti alfanumerički niz *procname* u koji ćemo smestiti imena potprograma; niz ćemo dimensionisati tako da „pamti“ najviše 30 četvoroslovnih imena. Zašto samo toliko? Verujemo da se program koji razvijamo može veoma zgodno koristiti za mala crtanja, ali da bi za ozbiljnije primene trebalo kreirati editor, potprogram za snimanje na disk i slične module koje ćemo ostaviti za neku drugu priliku. Ako neko želi duža imena procedura ili promenljivih (imena i u ovoj varijanti programa mogu da budu proizvoljno dugačka, ali se procedura RAČUNARI nećemo razlikovati od procedure RAČUNICA), jednostavno će promeniti vrednosti konstanti *no_par* i *syml*.

Osim imena procedura, treba nekako da pamtimo i tekstove naredbi koje ih realizuju. Za to je rezervisan niz *proclin* u kome su, jedna za drugom, smeštene sve deklarirane procedure. Kako će računaru znati koja procedura počinje od koje linije *proclin*-a i na kojoj se njegovoj liniji završava? Nizovi *start* i *fin* nose odgovarajuće informacije prema šemi sa slike 2.



Potrebna nam je, najzad, i informacija o broju trenutno definisanih procedura kako bismo znali kada da prijavimo grešku i gde da upišemo definiciju novog potprograma; uvešćemo, zato, celobrojnu promenljivu *lastpr* kojoj ćemo na početku izvršavanja glavnog programa dodeliti vrednost 0 a zatim je uvećati kada se god definishe nova procedura. Ne bi bilo teško uvesti logo naredbu za brisanje definicije neke procedure, ali nam se čini da bi ona i onako podugačak program načinila još dužim bez nekog stvarnog efekta.

Najjednostavniji način da obezbedimo deklarisanje procedura je dodavanje naredbe *if symbol = TO then define* u proceduru PLINE; tako će simbol 'TO' na početku svakog reda izazvati poziv procedure *define* koja će primati naredbe sa tastature i smestati ih u niz *proclin* sve dok ne učita naredbu END kada će adrese početka i kraja definicije upisati u *start* i *fin*. Program *define* je dat na slici 3.

slika 3:

```
procedure define;
var i: integer;
dumst: stati;
begin
  getkey[line,lp,symbol];
  lastpr:=lastpr+1;
  for i:=1 to dumst do procname[i,lastpr,lp]=symbol[i];
  beg[lastpr]:=lastpr+1;
  getline[no:to:lines,lp,lines];
  repeat
    lastpr:=lastpr+1;
    for i:=1 to dumst do proclin[i,lastpr,lp]=line[i];
    getline[no:to:lines,lp,lines];
  until (not dumst[no:lines]);
  fin[lastpr]:=lastpr;
end;
```

Pošto smo definisali proceduru, potrebna nam je neki način da je pozovemo. Zato ćemo modifikovati proceduru *command* prema slici 4. Pošto nije prepoznao nijednu od standardnih logo naredbi, interpretator će, umesto da prijavi grešku, pre-

tražiti tabelu imena procedura i pokušati u njoj da pronađe unesenu naredbu. Ako pretraga bude uspešna, procedura *lookup* će dodeliti fleg-u *proc* vrednost *true* i tako izazvati poziv procedure *call* i izvršavanje naredbe.

Realizovati potprograme, dakle, nije teško, ali se mora paziti na mnoge finise. Pretpostavimo da je korisnik otkucio RE-

PEAT 6 CRTAJ LEFT 90, gde je CRTAJ ime ranije definisane procedure. Ukoliko procedura CRTAJ ispiše svoju prvu naredbu u bifer liniji, naredba LEFT 90 će, jednostavno, biti prebrisana i neće moći da se izvrši

slika 4:

```
procedure command(n:word;diver:integer;stivar:integer);
var count:stati;
i:proc;integer;
procst:boolean;
begin
  count:=0;
  i:=diver;
  while i<=stivar do
  begin
    proc:=proclin[i];
    dumst[i]:=procname[i];
    count:=count+1;
    proc:=proc+dumst[i];
  until proc=or(proc=lastpr);
  end;
  procedure call(procst:integer);
  var i:stati;
  i:=count+1;
  integer;
  begin
    count:=count+1;
    repeat
      for i:=1 to dumst do
        line[i]:=proclin[count,i];
        i:=i+1;
        getkey[line,lp,symbol];
        group[lp,lp,lines,symbol[i]];
        count:=count+1;
      until (count=stivar) or failed;
    end;
  end;
  procedure call(procst:integer);
  var i:stati;
  i:=count+1;
  integer;
  begin
    count:=count+1;
    repeat
      for i:=1 to dumst do
        line[i]:=proclin[count,i];
        i:=i+1;
        getkey[line,lp,symbol];
        group[lp,lp,lines,symbol[i]];
        count:=count+1;
      until (count=stivar) or failed;
    end;
  end;
  for i:=1 to count do
    count[i]:=i;
  if word='PEQD' then goto:failed;
  if word='PEQD' then goto:failed;
  if word='CLEAR' then clear;
  if word='FQW' then for:for:word;
  if word='BACK' then for:for:word;
  if word='LEFT' then left:right:word;
  if word='RIGHT' then left:right:word;
  if word='REPE' then repeat;
  else begin
    lookup(procst);
    if procst
      then call:proc;
    else failed;
  end;
end;
```

Verujemo da ste tokom protekla dva meseca otkucali i proučili jednostavni logo interpreter kojim smo ilustrovali naše teorijske napise iz serije „Put u središte ROM-a“. Ako ste logo interpreter koristili i za „pravo“ crtanje, svakako ste poželeli da ga opremite mnogim „de luks“ dodacima. Jedan od takvih dodataka je mogućnost pozivanja procedura i definisanja njihovih argumenata.

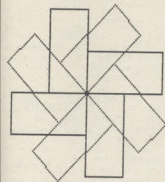
po izlasku iz potprograma. Još će veći problem nastati ako jedna procedura poziva drugu, druga treću i tako dalje. Problem smo rešili definisanjem lokalnih promenljivih *line*, *lp*, *count*, *proct* i *proc* u potprogramima *call* i *lookup*. Korišćenje lokalnih promenljivih i rekurzija koje su skrivene u pozivima mnogih procedura su osnovni razlog koji nas je naveo da logo interpreter realizujemo na paskalu: iako bi bežik bio u prednosti zbog jednostavnijeg rada sa alfanumericima, program pisan na njemu bi bio mnogostruko duži zbog simulacije rekurzija neophodnog steka.

Procedure sa parametrima

Procedure koje smo do sada pisali se uvek izvršavaju na isti način: možemo da kažemo *drawsquare* ali ne i da zadamo dimenzije kvadrata koji treba nacrtati! Zar ne bi bilo lepo definisati proceduru *rectangle* kao na slici 5, a zatim, primenom jednostavnih programa, dobiti crteže sa slike 1?

```
TO RECTANGLE :LENGTH :WIDTH
  PENDOWN
  REPEAT 2 [FORWARD :LENGTH LEFT 90
            FORWARD :WIDTH · LEFT 90]
END
```

slika 5:



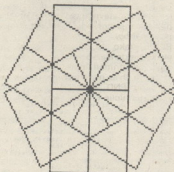
```
REPEAT 8 [RECTANGLE 200 100 LEFT 45] REPEAT 6[RECTANGLE 200 100 LEFT 30
RECTANGLE 100 200 LEFT 30]
```

Vidimo da se, pri definisanju procedure, iza njenog imena navode imena formalnih argumenata (ako ne možete da se setite razlike između stvarnih i formalnih argumenata, pogledajte napis „Funkcije, potprogrami i procedure“ iz „Računara 7“ — neka vas ne uvredi činjenica da je tekst

namenjen početnicima) kojima prethodi oznaka „:“. U okviru definicije procedure ti se formalni argumenti mogu proizvoljno koristiti sve dok svakom od njih prethodi već pomenuta dvotačka. Pri pozivu procedure se iza imena navode stvarni argumenti razdvojeni blankovima bez vodećih dvotački. Nema, naravno, nikakvih problema da se iza poziva procedure i liste parametara navede neka nova naredba koja po potrebi može da bude i poziv novog potprograma.

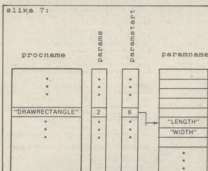
Za uspešnu realizaciju prenošenja parametara u potprogramu treba rešiti nekoliko važnih problema. Pre svega, interpreter mora da zna koliko neka procedura ima parametara kako bi po njenom pozivu mogla da pokupi potreban broj simbola iz komandne linije. Osim toga, vrednost svakog stvarnog argumenta mora da bude povezana sa imenom odgovarajućeg formalnog argumenta kako bi se u okviru procedure pozivala na ispravnom mestu. Rešenje prvog problema su novi nizovi

slika 6:



params, *paramstart* i *paramname*. Da počnemo sa kraja: u stringove *paramname* se upisuju imena parametara, *paramstart()* ukazuje na ime prvog parametra procedure čiji je redni broj *i* dok *params(i)* čuva ukupan broj njenih parametara. Ako vam je ova rečenica izgledala previše komplikovano, pogledajte sliku 7.

Najpre ćemo napisati proceduru *getparam* sa slike 8. Ovaj potprogram će, naravno, pozivati proceduru *define* koja, kao što



znamo, unosi podatke o definisanoj proceduri u odgovarajuće table.

slika 8:

```
procedure getparam(var line:stri;
                   var lp:integer);
begin
  params[lastpr]:=0;
  getsy(line,lp,symbol);
  if symbol[1]=':' then
  then
  begin
    paramstart[lastpr]:=lastpn+1;
    repeat
      params[lastpr]:=
        params[lastpr]+1;
      getsy(line,lp,symbol);
      lastpn:=lastpn+1;
      paramname[lastpn]:=symbol;
      getsy(line,lp,symbol);
    until symbol[1]<>' ';
  end;
end;
```

Daleko je složenije realizovati poziv procedure. Kako smo želeli da jedan potprogram može da poziva drugi i da mu nezavisno prenosi argumente koji mogu da imaju potpuno jednaka imena, morali smo da primenimo mehanizam steka. Pri svakom pozivu neke procedure u niz *vars* se upisuju imena njenih formalnih argumenata dok se u *vals* upisuju vrednosti stvarnih. Svako dodicije referenciranje nekog formalnog argumenta će preći preko funkcije *varval* koja će pronaći i vratiti njegovu vrednost. Konačni logo interpreter (ili, bolje rečeno, logo interpreter koji mi proglašavamo konačnim) dat je na slici 9.

Posle prvih nastavaka „Radionice logičkih igara“ i „Putovanja u središte ROM-a“, autor ovoga teksta je bio u prilici da čuje mnogo korisnih komentara o formi prezentacije softvera koji objavujemo. Pre svega, naši programi zaista imaju premla komentara i nisu dovoljno nazubljeni, što se može opravdati samo na jedan način: prostor u „Računarima“ je toliko ograničen da ne možemo da objavimo duže listine opterećene ponekom veoma dugom ili veoma uvučenom linijom. Druge primedbe smo u prilici da uvažimo: naš konačni logo inter-

```

1 0 = (SC4500)
2 0 = program logoinput,output;
3 0 = (
4 0 C
5 0 C
6 0 C
7 0 C
8 0 C
9 0 C
10 0 C
11 0 = const buffr256; conl4; symb130; pri=9;
12 0 = pr_line:=no_var:=100; no_par:=30;
13 0 = a0:=0; a9:=57;
14 0 = type st=packed array [1..buffr] of char;
15 0 = st=packed array [1..symb1] of char;
16 0 = stat=packed array [1..conl] of char;
17 0 = line:line1:st;
18 0 = var
19 0 = symbol:lineof words;
20 0 = i,j:ip:integer;
21 0 = x,y:angle:vd:vr:vd:vr:real;
22 0 = par:failed:Boolean;
23 0 = procname array [1..no_par] of word;
24 0 = startarray [1..no_par] of integer;
25 0 = fin larray [1..no_par] of integer;
26 0 = paramlarray [1..pr_line] of st;
27 0 = lastof:integer;
28 0 = paramstarray [1..no_par] of word;
29 0 = varstarray [1..no_var] of integer;
30 0 = valstarray [1..no_var] of integer;
31 0 = stack:lastof:integer;
32 0 = paramstarray [1..no_par]
33 0 = of integer;
34 0 = paramstarray [1..no_par] of word;
35 0 =
36 0 = function upcc(char:char):char;
37 1 = begin
38 1 = if (ord(a)>96) and (ord(a)<123)
39 1 = then upcc:=chr(ord(a)-32)
40 1 = else upcc:=a;
41 1 = end;
42 0 =
43 0 = procedure getsv( line1:st;
44 1 = var ip:integer;
45 1 = var symbol:word);
46 1 = var j: integer;
47 1 = function delim(char:Boolean)
48 1 = begin
49 1 = delim:=(' ' or '(' or ')') or
50 1 = ('<' or '>') or ('=' or '<=' or '>=');
51 1 = end;
52 1 =
53 1 = begin
54 1 = while line[ip]= ' ' do
55 1 = ip:=ip+1;
56 1 = j:=1;
57 1 = if delim(line[ip])
58 1 = then begin
59 1 = symbol[j]:=upcc(line[ip]);
60 1 = ip:=ip+1;
61 1 =
62 1 = else end
63 1 = begin
64 1 = repeat
65 1 = symbol[j]:=upcc(line[ip]);
66 1 = ip:=ip+1;
67 1 = until delim(line[ip]=1);
68 1 = ip:=ip+1;
69 1 = symbol[j]:=j+1;
70 1 =
71 1 = until ip=1;
72 1 =
73 1 = end
74 1 = end;
75 1 =
76 0 = procedure getline( promt:stat;
77 1 = var i:integer;
78 1 = var i:=integer;
79 1 = for i:=1 to pri do write(promt[i]);
80 1 = i:=1;
81 1 = repeat
82 1 =
83 1 = repeat
84 1 = at:=line[i*1000];
85 1 = until (at[25] or (at[3]
86 1 = if (at[127] and (i<buffr)
87 1 = then
88 1 = begin
89 1 = line[i]:=chr(at);
90 1 = write(line[i]);
91 1 = i:=i+1;
92 1 = end
93 1 = else if am127 then (DELETE)
94 1 = begin
95 1 = if i>1 then
96 1 = begin
97 1 = write(chr(a));
98 1 =
99 1 = end
100 1 =
101 1 = else vdo (7) (NEE:=uvisive slow)
102 1 = until am13;
103 1 = line[i]:=';';
104 1 = writein
105 1 = end;
106 1 =

```

```

107 0 = procedure fail(erro:integer);
108 1 = begin
109 1 = writein('Error',erro:3);
110 1 = failed:=true;
111 1 = end;
112 1 =
113 0 = function varval(var line:st;
114 1 = var varfnd:Boolean)
115 1 = var i:=1;
116 1 = while (i<=symb1) and varfnd and
117 1 = var del:=2;
118 1 = begin
119 1 = getsv(line,ip,symbol);
120 1 = if i=stack then
121 1 =
122 1 = else begin fail(5);varval:=end;
123 1 =
124 1 = begin
125 1 = sp:=stack+1;
126 1 = repeat
127 1 = varfnd:=true; i:=i;
128 1 = while (i<=symb1) and varfnd and
129 1 = (symbol[i]<'>') do
130 1 =
131 1 = begin
132 1 = if vars[sp]:=symbol[i]
133 1 = then varfnd:=false;
134 1 = i:=i+1;
135 1 = until varfnd or (sp=1);
136 1 = if varfnd then varval:=val[sp];
137 1 = else begin fail(5); varval:=end;
138 1 =
139 1 = end;
140 1 =
141 0 = function value(var line:st;
142 1 = var ip:integer):integer;
143 1 = var i:=1;
144 1 = corr:Boolean;
145 1 = begin
146 1 = getsv(line,ip,symbol);
147 1 = if symbol[i]='+'
148 1 = then value:=varval(line,ip);
149 1 = else begin
150 1 = i:=i corr:=true;
151 1 = while symbol[i]<'>' and corr do
152 1 =
153 1 = begin
154 1 = t:=ord(symbol[i]);
155 1 = if (t<60) or (t>69)
156 1 = then corr:=false;
157 1 = i:=i+1;
158 1 = end;
159 1 = if corr then i:=val(symbol);
160 1 = else i:=i;
161 1 = if i=0 then fail(2);
162 1 = value:=i;
163 1 = end;
164 1 = end;
165 0 = procedure getparname(var line:st;
166 1 = var ip:integer);
167 1 = begin
168 1 = parame[lastr]:=0;
169 1 = getsv(line,ip,symbol);
170 1 = if symbol[ip]='+'
171 1 = then
172 1 = begin
173 1 = paramestart[lastr]:=lastr+1;
174 1 = repeat
175 1 = parame[lastr]:=
176 1 = parame[lastr]+1;
177 1 = lastp:=lastr+1;
178 1 = getsv(line,ip,symbol);
179 1 = parameam[lastr]:=symbol;
180 1 = getsv(line,ip,symbol);
181 1 = until symbol[ip]='+';
182 1 = end
183 1 = end;
184 0 = procedure defines;
185 1 = var i:= integer;
186 1 = dum1:stat;
187 1 = begin
188 1 = dum1:=stati;
189 1 = lastp:=1;
190 1 = lastpr:=1;
191 1 = procname[lastr]:=myymbol;
192 1 = getsv(line,ip,symbol);
193 1 = start[lastr]:=lasti+1;
194 1 = getline('TO line1',line);
195 1 = repeat
196 1 =
197 1 = lasti:=lasti+1;
198 1 = procin[lasti]:=lines;
199 1 = getline('TO line1',line);
200 1 = for i:=1 to conl do
201 1 = dum1[myymbol]:=upcc(line[i]);
202 1 = until dum1[END];
203 1 = fin[lastr]:=lasti;
204 1 = end;
205 0 = procedure clear;
206 1 = begin
207 1 = vdu(16); ( brize graficki prozor)
208 1 = x:=402; y:=570;
209 1 = plot(4, trunc(x), trunc(y));
210 1 = ( kurzor na centar stranu)
211 1 = xdir:=0; ydir:=0; angle:=90;
212 1 = pen:=true;
213 1 = end;
214 1 =
215 0 = procedure group( term:char;
216 1 = var ip:integer);
217 1 = var line:st;
218 1 = var symbol:word;
219 1 =
220 1 = procedure command(stword:var line:st;

```

```

221 1 = var ip:integer);
222 2 =
223 2 =
224 2 =
225 2 =
226 2 =
227 2 =
228 2 =
229 2 =
230 2 =
231 2 =
232 2 =
233 2 =
234 2 =
235 2 =
236 2 =
237 2 =
238 2 =
239 2 =
240 2 =
241 2 =
242 2 =
243 2 =
244 2 =
245 2 =
246 2 =
247 2 =
248 2 =
249 2 =
250 2 =
251 2 =
252 2 =
253 2 =
254 2 =
255 2 =
256 2 =
257 2 =
258 2 =
259 2 =
260 2 =
261 2 =
262 2 =
263 2 =
264 2 =
265 2 =
266 2 =
267 2 =
268 2 =
269 2 =
270 2 =
271 2 =
272 2 =
273 2 =
274 2 =
275 2 =
276 2 =
277 2 =
278 2 =
279 2 =
280 2 =
281 2 =
282 2 =
283 2 =
284 2 =
285 2 =
286 2 =
287 2 =
288 2 =
289 2 =
290 2 =
291 2 =
292 2 =
293 2 =
294 2 =
295 2 =
296 2 =
297 2 =
298 2 =
299 2 =
300 2 =
301 2 =
302 2 =
303 2 =
304 2 =
305 2 =
306 2 =
307 2 =
308 2 =
309 2 =
310 2 =
311 2 =
312 2 =
313 2 =
314 2 =
315 2 =
316 2 =
317 2 =
318 2 =
319 2 =
320 2 =
321 2 =
322 2 =
323 2 =
324 2 =
325 2 =
326 2 =
327 2 =
328 2 =
329 2 =
330 2 =
331 2 =
332 2 =
333 2 =
334 2 =

```

```

325 3 - begin
326 3 -   line:=bl:line
327 3 -   count:=start[proc]:
328 3 -   repeat
329 3 -     line:=proc[in:count]:
330 3 -     ip:=1: getvc[line,ip,symbol]:
331 3 -     group["'",ip,line,symbol]:
332 3 -     count:=count+1:
333 3 -     until count=>fin[proc]: or
334 3 -     fail:
335 3 -     stack:=stack+params[proc]:
336 3 -   end:
337 3 -
338 2 - begin
339 2 -   for i:=1 to count do smd[i]:=i:
340 2 -   if cmd="END" then pen:=false else
341 2 -   if cmd="PENU" then pen:=true else
342 2 -   if cmd="CLEAR" then clear else
343 2 -   if cmd="FORM" then for_jcc:=1 else
344 2 -   if cmd="BACK" then for_jcc:=1 else
345 2 -   if cmd="LEFT" then le_rig:=1 else
346 2 -   if cmd="RIGHT" then rig:=1 else
347 2 -   if cmd="REPE" then rept
348 2 -   else begin
349 2 -     ip:=proc[in:line,ip]:
350 2 -     if not failed
351 2 -     then call[proc]
352 2 -     else fail():
353 2 -   end
354 2 -   end
355 2 -   else fail():
356 2 - end:
357 2 -
358 1 - begin
359 1 -   fail:=false:
360 1 -   repeat
361 1 -     command[symbol,line,ip]:
362 1 -     getvc[line,ip,symbol]:
363 1 -     until (symbol[""]<term) or failed:
364 1 -   end:
365 1 -
366 0 - procedure pline(var line:str):
367 0 - begin
368 0 -   getline["line",line]:
369 0 -   ip:=1:
370 0 -   getvc[line,ip,symbol]:
371 0 -   if (symbol[""]<term) and
372 0 -   (symbol[""]<O)
373 0 -   then define
374 0 -   else group["'",ip,line,symbol]:
375 0 - end:
376 0 -
377 0 - begin
378 0 -   laststr:=0: lastlin:=0:
379 0 -   lastpnt:=0: stack:=0:
380 0 -   vdu(22,4):
381 0 -   if define mod 520*256 =
382 0 -   vdu(20,0,31,39,20):
383 0 -   if ( zadni crtaji linije
384 0 -   = " " )
385 0 -   then
386 0 -   vdu(24,0,0,500,0,577,4,577,33):
387 0 -   ( ostatak ekrana
388 0 -   = grafički predzor )
389 0 -   else
390 0 -   clear:
391 0 -   for i:=1 to buff do bl:line [i]:= ' ':
392 0 -   line:=bl:line:
393 0 -   for i:=1 to sval do bl:ord[i]:= ' ':
394 0 -   for i:=1 to sval do var[i]:=bl:word:
395 0 -   symbol:=bl:word:
396 0 -   repeat
397 0 -     getline[line]:
398 0 -   until false
399 0 -   end:
400 0 -   end.
401 0 -
402 0 - # Compilation errors(4)
403 0 - Code size = 4205 bytes

```

preter intenzivno koristi pakovane matrice za smeštanje stringova (raspali smo se oko i saznali da paskal kompajler svih popularnih personalnih modela podržavaju ovu strukturu), koristeći čemo FORWARD referencu funkcija, smanjili smo upotrebu globalnih simbola i prenosili argumente gde god je to isplativo, sve smo veličine koje ima smisla menjati deklarirali kao konstante da bi ih čitaoci lakše menjali (nije prijatno „loviti“ dužinu nekog stringa ili maksimalnu dimenziju matrice po čitavom programu) ... Promenili smo najzd, funkciju value tako da, pre poziva funkcije ival, ispita string koji treba pretvarati u broj i, ako u njemu nađe slova, prijaviti grešku. Na taj se način izbegava da program prekine izvršavanje i prijavi grešku "Bad value" baš kad se otkuvasi LEFT FORWARD 100, nameravali da iscrtae poslednju liniju na nekoj komplikovanoj slici.

Završavajući prezentaciju logo interpretera, reći ćemo nekoliko reči o načinima na koje bi on mogao da se učini još boljim. Trebalo bi, pre svega, omogućiti unošenje

logo programa u memoriju i njihovo snimanje, učitavanje i editovanje. Interpreter bi trebalo dopuniti modulima za izračunavanje vrednosti aritmetičkih izraza (time ćemo se baviti već u sledećem nastavku ove Škole) i rad sa stringovima, a moglo bi se napraviti i nove kontrolne strukture. Knjiga „Advanced Programming Techniques“ Džima Mekgregora i Alana Vata (Jima Mc Gregora i Alana Watta) koju upravo napravimo govori o dodavanju naredbe STOP koja izaziva izlazak iz procedure i omogućava izuzetno efektene programe poput onoga sa slike 10.

slika 10:

```

TO spiral :len :wid
  IF <twid<:d THEN [STOP]
  rectangle :len :wid
  LEFT 10
  spiral (:len-d) (:wid-d)
END

```

Upotreba jezičkih procesora

Logo interpreter nije sama ilustracija raznih oblasti sistemskog programiranja koje smo pominjali u ovoj Školi; on je i dobar primer jezičkog procesora. Zamislite da je neko naručio složeni programski sistem za crtanje ili CAD (Computer Aided Design). Ovakvi se paketi obično prave tako da specijalni tasteri vašeg računara dobiju razne funkcije i da se, po izboru neke od njih, preko tastature unose potrebni parametri, lako potencijalno veoma moćni, ovakvi su programski paketi teški za upotrebu. Korisnik, pre svega, mora da zapamti funkciju svakog tastera, zatim da po pritisku na njega odgovara na niz pitanja i da, najzad, posle svake greške preduzima komplikovane manevre (koje takode treba znati napamet) da bi izbrisao neki element ne oštećivši ostatak. Drugi pristup problemu je naoko teži za realizaciju, ali omogućava daleko komforniju upotrebu: treba kreirati programski jezik za crtanje koji se brzo i lako uči i još lakše upotrebljava. Budućnost će korisniku biti daleko lakše da zapamti desetak reči koje imaju smisla nego da memorishe funkcije mnoštva specijalnih tastera. Osim toga, korisnik će biti opušten u radu ako zna da svaku grešku neće platiti ponovnim crtanjem slike; posle izmene neke naredbe logo programa, računac će samostalno i veoma brzo ponoviti čitav crt.

Pisanje jezičkog procesora je još prosperitetnija aktivnost kada se piše neki program paket koji obavlja složene matematičke ili inženjerske proračune: napisao-ćemo standardni interpreter koji podržava naredbe za izračunavanje izraza, unošenje i dodeljivanje vrednosti ka i osnovne kontrolne strukture (IF ... THEN ... ELSE ... WHILE ... END WHILE, GOTO ...), a zatim uvesti naredbe, jasno, izvršavati relativno dugo ali to ne treba da nas zabrinjava; korisnik će i dalje dobiti odgovor nekoliko miliona puta brže nego da je radio „na ruke“ ili koristio neki kalkulator. Sa druge strane, korisnik našeg programskog paketa neće morati da opterećuje svoju memoriju stvarima koje mu nisu potrebne: pamtiće samo naredbe koje koristi u svakodnevnom radu. Sve ostale informacije o korišćenju programskog paketa se mogu naći u okviru

uputstva za upotrebu i to u vrlo kratkom vremenu: obavezno priložimo abecedni spisak na kome je svakoj naredbi posvećena po jedna (ili po nekoliko) strana sa detaljnim opisom.

Recimo, na kraju, da je pisanje specijalizovanih programskih jezika povoljna varijanta čak i za autore softvera: jeste da je teže napisati interpreter nego program koji radi sa menjalima, ali se čitav posao radi jedanput u životu: kada smo razvili opšti deo jezika posao smo, naravno, obavili tako da jezik bude okrenut proširenljima i dobro dokumentovan; za sve ostale programerske pakete koje proizvedimo ćemo samo koncipirati specijalizovane naredbe i dodavati module koji ih izvršavaju.

Dejan Ristanović

NASTAVAK SA 43. STRANE

Sam svoj programer

Programi koji koriste EXOPEN treba ovaj broj da sačuvaju i pri kasnijem obračunu periferiji da ga koriste kao logički broj veze, što je već standardan način. Određivanje logičkog broja od strane programa, umesto da to čini korisnik, ima i jednu prednost; ne može se slučajno upotrebiti već upotrebljen broj, čime otpada greška broj 2. EXOPEN u tablici logičkih adresa već otvorenih veze pronalazi najveću i zatim kao novi logički broj uzima jedan veći. Tako će prva otvorena veza uvek imati broj 1, sledeća broj 2 itd. Ako se, zatim, zatvorimo samo broj 1, a otvori nova veza, njen broj će biti 3 iako 1 više ne postoji. Ostatak rutine teže ubičajeno, traži se baft na kraju na kraju naziva da bi se odredila njegova dužina, izdvajaju iz akumulatora prva i sekundarna adresa i pozivaju standardne KERNAL rutine, ali i funkcija broj 40 (OER-ROR) ako je veza otvorena prema disku da bi se proverilo njegovo stanje. Obratite pažnju kako je obezbeđeno stanje CARRY zastavice na izlazu.

Evo i nekoliko predloga za rutine koje treba razviti. Pošto nema oslanjanja na bežik ROM, nedostajace rutina za ispis bilo koje poruke, na primer SPRINT (String Print), čiji bi ulazni parametri bili početna adresa poruke, a kraj može biti obeležen bajtom 0. Isto važi i za SINPURT (String Input), koji treba imati kao ulazne parametre takođe početnu adresu od koje treba upisati liniju, dok se krajem linije smatra terminator 13 koji rutina treba da zameni sa 0 ako bi se održala kompatibilnost sa SPRINT. Vreme je da pokrenemo i pitanje grafike visoke rezolucije sa, recimo, rutinama MODE (izbor jednog od tri video moda), PALETTE (postavljanje boja koje će biti korišćene) i PLOT (setovanje/resetovanje jedne tačke na ekranu visoke rezolucije). Ovo su relativno jednostavni zadaci, ali baš zato zahtevaju da budu i maksimalno efikasno rešeni. Kadu to bude urađeno, možemo da vam uputimo i prvi izvost: napravite rutine SPRINT i SINPURT koje će razlikovati u kom modu je video procesor (tekst ili visoka rezolucija) ali će i dalje obavljati istu zadatak bez promene moda. Ako sve to bude realizovano, konačna proširenja jezika će biti vrlo lako realizovati, a naš operativni sistem će moći da ponese i, baš od prvog slova, atribut GRAFIČKI.

Zoran Životić

KOMODOR

Iste programat

POKE:

Annihilator	6295,11	Falcon patrol	1.	16764,36
Attack of the mutant camels	11639,255	Falcon patrol	2.	16785,21
Aphay, Gateway of	2264,99	Falcon patrol		16764,234
Blogger	3569,8	Falcon patrol (sys 16640)		16765,234
Blogger	53266,26	Fantastic Freddy		31887,255
Blogger	5574,34	Fantastic Freddy		34535,24
Burnin rubber	18432,173	Fantastic Freddy		2465,165
Bruce Lee	1.	Frogger Sega		22341,173
Bruce Lee	5677,128	Flak		4798,36
Bruce Lee	2.	Galaga		17388,173
Bruce Lee	5686,128	Galaxions	1.	7865,238
Bruce Lee	7828,165	Galaxions	2.	17288,165
Bruce Lee	5472,99	Galaxy	1.	3369,238
Bagitman	5472,128	Galaxy	2.	3378,238
Bagitman	19813,189	Garstner		5989,58
Bagitman	53264,126	Gangster		3574,44
Bagitman	22236,255	Gangster		8367,255
Bagitman	22236,39	Ghost		8567,99
Back Rogers	8852,36	Ghost		3999,255
Back Rogers	2498,9	Gyross		3999,200
Black hawk	8298,255	Gyross		4491,59
Black hawk	8289,99	Hard hat sack		16877,100
Black hawk	2771,238	Hard hat sack		8472,108
Battle trough time	22845,255	Hard hat sack		16877,234
Battle trough time	16877,234	Hunchback		9591,234
Bungling bay	47465,176	Hunchback	1.	9521,234
Bat attack	11861,234	Hunchback	2.	9522,234
Battle zone	8789,108	Hunchback	3.	9523,234
Boulder dash	16494,165	Hero		7155,255
Crazy kong	58264,173	Hero		19131,8
Cross fire	27625,173	Hero		13865,0
Cross fire	5353,44	Hero		19131,0
Choplifter	80811,173	High noon		18833,255
China miner	33381,245	Hoop of usher		6721,238
China miner	33457,255	Hoop of usher		7878,60
China miner	34623,44	Jumping jack		27984,173
China miner	1.	Jumpman junior		9458,44
China miner	34623,234	Jumpman junior		9458,173
China miner	2.	Jet Set Willy		11345,33
China miner	34624,234	Jungle hunt	1.	2242,234
China miner	3.	Jungle hunt	2.	2243,234
Crisis mountine	1.	Jungle hunt		2242,165
Crisis mountine	2.	Kickcan		7424,238
Crisis mountine	3.	Kickcan		4565,255
Cross fire	1.	Kid grid		10828,234
Cross fire	2.	Laser strike		16475,173
Cross fire	3.	Lazy Jones		2971,9
Crown	3566,255	Loderunner		7892,255
Cavelon	23789,255	Lady Tut		2392,50
Cavelon	33789,99	Lancer lords		16424,60
Cavelon	15458,255	Miner 2049'er		9458,173
Chiller	22581,189	Miner 2049'er		2652,165
Cuthbert in jungle	2459,5	Miner 2049'er		2652,168
Dinky doo	11989,99	Miner 2049'er		2471,255
Dinky doo	11989,18	Miner 2049'er		11510,255
Dinky doo	12296,165	Motor mania		8646,255
Donkey kong	12118,234	Mrs. Pacman		8898,173
Diamonst X	8645,129	Manic miner (sys1384)		16571,173
Dig dug	18473,255	Manic miner (sys1384)		15456,75
Dare devil Dennis	29173,255	Manic miner (sys1384)		15424,75
Dare devil Dennis (x-brzina)	17958,30	Moon buggy		25151,173
Dragon hawk	3477,255	Moon buggy		38624,173
Defender	3885,15	Moon buggy		24151,173
Defender	2814,255	Matrix	1.	7629,238
Evolutin	4947,25	Matrix	2.	7983,238
Eagle espire	22144,50	Mr. Robot & H.F.		11518,255
Encounter	38438,0	Maggot mania		2532,4
Frogger	22381,173	Monty sole		7864,3
Fire ant	17568,188	Moon patrol (Atari)		2872,255
Fire ant	17567,7	Moon patrol		2872,200
Fort apocalypse	1.	Neptuns daughters		7878,60
Fort apocalypse	2.	Omega race		6380,238
Fort apocalypse	3.	Pooyan		28634,173
Fort apocalypse	4.	Pogo Joe		23558,169
Fort apocalypse	1.	Pect (Fengo)		328295,44
Fort apocalypse	2.	Punchy		15458,50
Fort apocalypse	3.	Pedestrian		2288,255
Fort apocalypse		Pacaluda		7815,234
Fort apocalypse		Pitfall		5393,255



Hajde da se
igramo

Pokice koje život znače

Da bi vam malo pomogli u igranju bezbrojnih igara koje upravo zagušuju tržište ev vam prava gomila pokica koji će vam pomoći da lakše stignete do kraja. Najveći deo pokova daje bezbroj života, dok neki rešavaju neke konkretne prepreke u igri. Koliko igara svakog dana stize u Jugoslaviju, tek da bi imali vremena da ih sve igrate do kraja da nema raznih pokova, zar ne? Izvolite.

- Arcadia POKÉ 23776,0 ili POKÉ 25776,0
- Cookie POKÉ 35730,52
- Ground Attack POKÉ 29063,0
- Hunchback POKÉ 26888,0
- Jet-Pack POKÉ 25020,0
- Kokotino Will Prvo treba uneti sledeći programčić i izvršiti ga, pa zatim uneti igru
- 10 CLEAR 24100: LOAD "CODE"
- 20 RANDOMISE USR 65100: LOAD "CODE"
- 30 POKÉ 42214.X: PRINT USR 41200
- gde je X=broj života.
- POKÉ 36212,0
- POKÉ 36965,0 (bezbroj života)
- POKÉ 37999,201 (brisanje svih neprijatelja)
- POKÉ 33693,0
- POKÉ 46441,0
- POKÉ 24984,0
- POKÉ 25446,0
- POKÉ 34610,0
- POKÉ 44685,0
- Uneti i startovati sledeći program pre igre.
- 10 BORDER O: PAPER O: INK O
- 20 CLEAR 26870
- 30 LOAD "SCREENS"
- 40 LOAD "CODE 26880
- 50 RANDOMISE USR 26880
- 60 POKÉ 58294,0
- 70 RANDOMISE USR 52500
- POKÉ 25062,254
- Uneti i startovati neposredno pre igre.
- 10 CLEAR 24599: FOR f=23296 TO 23309: READ a:POKE a, NEXT f
- 20 LET L=USR 23296: POKÉ 40060,0: LET

- Mr Wimpy
- Mutant Monty
- Piballed
- PSST
- TranzAm
- Free Beez
- Pyramid
- Booty
- Spectral invaders
- Cauldron
- River Rescue
- Zaxxon
- Cavern fighter
- Black Hawk
- A View to a Kill

- L=USR 24600.
- 30 DATA 221, 33, 24, 96,
- 17, 232, 159, 62, 255, 55,
- 205, 86, 5, 201.
- POKÉ 33420,0 (prvi igrac)
- POKÉ 33452,0 (drugi igrac)
- POKÉ 48825,255
- POKÉ 31683,0: POKÉ 31684,0
- POKÉ 34695,183
- Dodati liniji 30 ispred nardebe Rand USR POKÉ 28032,255: POKÉ 28087,8: POKÉ 29243,8: POKÉ 30543,8
- POKÉ 27662,2

- Road Racer POKÉ 27150,0
- Ali Diddums POKÉ 24786,0
- Orion POKÉ 37319,201
- Zip Zap POKÉ 54065,0
- Jet Set Willy 2 Uneti i startovati pre igre
- 10 Clear 64998
- 20 Let obj=150
- 30 Let room=320
- 40 For n=65000 TO 65047:READ a:POKE n, a: NEXT n
- 50 PAPER O: INK O: BORDER O: CLS
- 60 RANDOMISE USR 65000
- 70 DATA 221, 33, 0, 64, 17,
- 56, 185, 62, 255, 55, 205.

Piffall	2660,250
Popeye	1. 2405,255
Popeye	2. 2486,295
Protector 2.	16425,46
Q - bert	4446,173
Q - bert	4446,238
Quest for tires	14864,0
Quest for tires	7341,99
Quest for tires	11486,125
Radar rat race	7194,234
Revenge camels	1. 2899,230
Revenge camels	2. 2746,230
Revenge camels	3. 39931,238
Revenge camels	35518,250
R - nast	4446,173
R - nast	4446,238
R - nast	2759,180
Round about	12843,234
Robin rescue	6144,234
Ring of power	43,287
Ring of power	44,24
Snookie	33242,255
Snookie	33242,280
Snookie	38116,173
Sea fox	7337,173
Sawey lightfoot	3678,189
Shanus	1. 27185,169
Shanus	2. 27185,165
Shanus	3. 27185,141
Shanus	4. 27185,54
Shanus	5. 27185,179
Shanus	1. 18486,169
Shanus	2. 23558,169
Spelunker	1. 19125,234
Spelunker	2. 19126,234
Spelunker	10487,44
Shanus case 2	15473,238
Shanus case 2	3889,128
Shanus case 2	15476,176
Space taxi	16911,280
Sheep in space	35839,44
Shish'em	2562,188
Skrabble (x=brzina)	4691,8
Survivor	11291,175
Survivor	19543,295
Survivor	19523,280
Tales of Arabian nights	1. 2531,173
Tales of Arabian nights	2. 2632,141
Tapper	15899,165
Time runner	8543,9
Trollis wallie	7711,173
Wheellin wallie	27427,173
Wheellin wallie	27916,173
Whistler brothers	26188,8
Xerona	2273,234
Zeppelin	18546,44
Zeppelin	10801,180
Zeppelin	14337,180

289 komada pokica na broju		
GHOSTBUSTERS:		
ime	novacat	kantor
petar.	29.980	31664380
tuert,the	48.000	06660801
butterfly	68.000	04664701
nnn	77.000	20406281
s.s	100.000	1984
12345	415.500	25393120
herbie	500.500	05250624
butterfly	512.000	58542224

Preparao: Vlada Kratonovic

86, 5, 243, 48, 240
80 DATA 33, 6, 254, 17,
197, 100, 1, 59, 0, 237, 176,
195, 0, 95
90 DATA 62, 205, 50, 67,
117
100 DATA 62, obj, 50, 126,
135
110 DATA 62, room, 50, 75,
117
120 DATA 195, 0, 112

57/pokice koje život znače

Komodor 64



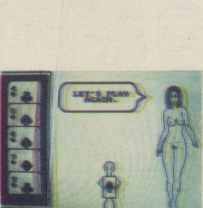
Winter games

Zimske Olimpijske igre

Evo igre koja će u predelima izvornog novog primitivizma biti izuzetno dobro prihvaćena. Samo, imaju li novi primitivci računare? Sve što vam treba je tu: sneg, planine, tereni, pa čak i publika. Hladnoća ne nedostaje. Ali zato nedostaje čevabodžinice.

Možete da birate između oznaka čak sedamnaest zemalja, a disciplina je samo sedam. Očigledno je da su ljudi iz Deklatona planirali da im se igra prodaje (pardon, piratizuje) u dosta zemalja, pa su obezbedili boje za domaće. Za nas je ovo verovatno jedini način da, osim one Frankove, osvojimo još koju medalju. Džojstik se dosta upotrebljava u svakoj igri, tako da je potreban fin osjećaj za odskoke akretanja i sične finise jer neretke misu ratke. To se najbolje vidi na skokovima gde se lako strada ako odskočite prerano. Meni lično je najzanimljivije klizanje, jer neodoljivo podseda na neku borilačku veštinu. Ono što njemu/fali je nazalni glas Milke Babovič, ali ništa nije savršeno, zar ne?

amstrad



Strip poker

Ovu igru ne treba posebno predstavljati, jer se igrala još kada se o računarima moglo samo maštati. Nemojte misliti da vam je protivnik neki računarski akrep u obliku nekog maštovitog sprajta. Protivnik vam je tako dobro nacrtao da ne treba da se iznenadite ako najveći (muški) deo vašeg komšiluka iznenada dobije želju da isproba vaš "amstrad" i, naravno, ovu igru. Čuvajte se, vaš protivnik igra suviše dobro i skoro ga je nemoguće pobediti. Zato vi lepo varate i ne skidate se, dok vaša protivnica naivno veruje u pravila i posla svake izgubljene partije skida pod jezan ide odeva. Ovo je igra za koju je najčešće traženo kako stići direktno do kraja, bez borbe. Pogodite zašto. Ako vas baš interesuje, pišite nam! Do tada, puno sreće u pokeru, i pazite da ne — nazabetel!

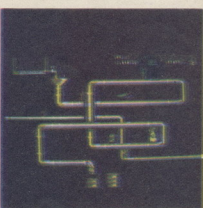


Racing Destruction set

Staza uništenja

Evo još jedne trkačke simulacije. U ovom slučaju ime vara — ne radi se o trkama na našim putevima, već o pravoj trkačkoj stazi. Igra ne počinje vožnjom. Prvo treba konstruisati stazu, postavljajući ukrštanja, mostove, nadvožnjake i ostale delove staze. Sledeći korak je podešavanje automobila koji želite da vozite. Možete praktično da konstruisate bilo kakvo vozilo koje zamislite. Samo treba odabrati dodatke ili izmene i vozilo je tu. Oni se izrazito mažohističkim svojstvima će sigurno biti obradovani kad saznaju da što su veću i jača kola to se one lepše skrckaju kad izletite sa staze.

Pogled na igru je, što je neuobičajeno, iz ptičje perspektive. Ekran je podeljen u dva dela, tako da mogu učestvovati i dva igrača. Onaj deo trke koji je najzanimljiviji vozačkom atmosferu (to sam ja) su skokovi na onim mestima gde ste postavili odskočne rampe. Sve je to lepo i realistično, ali ako ne pazite može postati i suviše realistično — automobilički izgledaju tako realistično kad su zgužvani u lopte kraj puta.



Super Pipeline II

Super Cevovod II

Da igra nemaju granica već znamo. Ovdje je to pokazano line što ste nadzornik vodovoda (??) čiji je zadatak da nadgledate instalacije cevovoda i da napuni par buradi dnevno. Pošto ste nadzornik, vi, u stvari, ne radite (uobičajeno, zar ne?), već imate dva radnika koji popravljaju oštećenja umesto vas. Naravno, tu je gomila gamadi i nevođa koji pokušavaju da vas srede. Za odbranu imate pištolj (pomalo neuobičajeno za vodovodnog nadzornika) sa kim morate biti vešti kao Kimi Istvud' ako želite da preživite. Ako baš prigrusti, možete da žrtvujete jednog radnika da biste se zaštitili (nije humano, ali je jako korisno). Poseban plus u igri su mali crtači i odlična muzika posle svakog savladanog nivoa. Igrajte i uživajte, čak je i u cevovodu rad ponekad zanimljiv.

lako su verzije „Elite“ za BBC, „komodore 64“ i „spektrum“ dela raznih softverskih firmi, sami programi su veoma slični, što znači da će vlasnici različitih računara moći manje—više ravnopravno da se koriste savetima koje dajemo, pa čak i da upotrebe program za „varanje“ koji pripremao za sledeće „Računare“. Verujući da su uputstva za upotrebu sve tri verzije programa identične, dajemo ih na slici 1.

slika 1

ELITE Komande

UPOREDBA

- POI: ustanište.
- F1: kupovina robe.
- F2: prodaja robe.
- F3: kvaliteta opreme.
- F4: vrsta galaksije.
- F5: vrsta oružja.
- F6: pomoć u izlasku na novu svetu
- F7: pomoć u proizvodnji kolonizirane robe.
- F8: trenutno stanje broda
- (vrsta, vrstina, novina...)
- F9: inventar robe koju brod nosi.
- STRAJANJE: postavljanje na svet od kojeg se putuje ka novom svetu hipersvemir.
- U: postavljanje komanda u centar mesta.
- D: postavljanje de mesta na koji ste postavljalni koridor.
- P: izbacivanje sveta po imenu (dise verzija).
- E: smanjenje vrstine na trasu/dio.
- CTRL: odobravanje igre
- DELETE: odobravanje igre

DOG JE IZVAJ SMOGOSTI

- ESCAPE: prekid igre.
- Z: preokretanje keyboard/ispustnik.
- N: okretanje kamala na zveztoce.
- U: isključivanje zvuka.
- U: isključivanje zvuka.

Bez alata nema ni zanata

Na svemirskim stanicama se mogu nabaviti razni uređaji koji povećavaju vatrenu i odbrambenu snagu vaše Kobre 3. Mnogi vlasnici piratskih kopija „Elite“, na žalost, ne kupuju ove dodatke jer, bez uputstva, ne umeju da ih iskoriste. Slika grešaka: vaš rejting nikada neće bitni napredovati ako ne opravimate vaš arsenal svim postojećim naručenjima.

U startu ste opremljeni samo slabim „pulse“ (pulsirajući) laserom koji je smešten se prednje strane vašeg broda. Ukoliko, pritiskajući neki od funkcijskih tastera, pogledate na proglavnu stranu, videćete da se u sredini ekrana ne pojavljuje ništa, što znači da ne posedujete lasere kojima biste mogli da gadate brodove koji vam prilaze sa strane ili leđa. Takvi se laseri mogu relativno jeftino dobiti. S druge strane, daleko je lakše investirati u snažnije „beam“ (zračni) lasere ili, bar, u jedan takav uređaj koji ćete staviti sa prednje strane broda—neprijatelji koji će tebe pobiti i teret koji ćete zapleniti će veoma brzo opravdati takvu investiciju.

„Beam“ laseri nisu, međutim, ni u kom slučaju konačni: za 5000 kredita se može kupiti „military“ laser koji je višestruko snažniji od običnog: jedan pogodak vojnim laserom će uništiti gotovo svakog neprijatelja! Bez ovakvog se lasera ne možete boriti protiv targoida niti uspešno okončati drugu misiju (kaseta verzija „Elite“ za BBC ne podržava misije niti vojne lasere). Cim skupite potrebne broj kredita, kupite prednje i

DOG JE IZVAJ SMOGOSTI

- A: laseri.
- T: naručivanje proizvodnje.
- N: isključivanje proizvodnje.
- I: imeno ako je prethodno uspešno, tj. ako njegov kvadrant je isprazan/otprazan.
- U: deaktiviranje naručivanja/projektija.
- E: uklanjanje svih (svetli i protivnikovi) proizvodnja u odeljku: smanjuje ECM.
- ESCAPE: isključivanje kompjutera sa zveztoce.
- TAB: aktiviranje energetske bombe.
- V: aktiviranje kompjutera sa zveztoce.
- P: deaktiviranje kompjutera sa zveztoce.
- E: gore.
- N: dole.
- K: rotacija levo.
- J: rotacija desno.
- SPACE: gde.
- /: nošenje.
- Z: svaka kroz prostor
- I: imeno ako je „Condition Green“.
- N: svaka kroz hipersvemir.
- CTRL: kontrola kvaliteta.
- F1: pogled napred.
- F2: pogled levo.
- F3: pogled desno.
- F4: pogled levo.
- F5: pogled desno.
- CTRL: smanjenje igru.
- DELETE: odobravanje igru.

VRSTINA U KONTOJNOJ PRAKTIKI

- Levo gore: stanje prednjih zveztoce.
- Ispod: stanje zadnjih zveztoce.
- Ispod: kolonizirane galaksije.
- Ispod: temperatura u kamali.
- Ispod: temperatura lasera.
- Ispod: naručivanje na planeti.
- Desno gore: broj trgovača koje posedujete.
- Desno sred: brojevi kretanja.
- Ispod: penjanje/apustanje i rotacija.
- Ispod: energija (ostali red).
- Određena: trodimenzionalni prikaz pripremi van brod je u centru a oko njega vidite sve druge brodove. Na desnoj strani se pojavljuje oblog S ako ste u alarmnom stanju stanicama sa leve strane ključ E ako je aktivan ECM sistem (vsa jili tužili). U desnom gornjem uglu videte je preokretanje pravca kade je u njegovom centru beli kvadrat kvadrata sa crnom piketom. Izvičivačve line (sliku u hipersvemir) soni stanicama koje deli kvadratu u centru. oznaka vrstine proizvodnje.

zadnji vojni laser, laseri sa leve i desne strane nisu naročito potrebni, jer je njihova skoro nemoguće rukovati.

Sledeća važna investicija je nimalo jeftini „Energy Unit“ (Energjska jedinica). Kada, naime, vaš prednji štitnici budu istrošeni u toku borbe, počete intenzivno pražnjenje energetskih rezervi broda. Kada se energija iz četiri rezervoara potroši, brod eksplodira, što predstavlja kraj igre. Srećna je okolnost što se energetski rezervoari puni iz spoljnog prostora, što višestruko ubrzava energetska jedinica — bez nje nikako: ne biste smeli da putujete u svetove u kojima vlada anarhija, jer će piratski talasi naljavljati na vas pre nego što vam se energija dopuni. U sledećim ćemo „Računarnima“ objavit program pomoću koga ćete sebi

dodeljivati energetsku jedinicu strahovite snage i tako postajati praktično neuništivi.

Skupljači goriva i energetske bombe

„Fuel scoops“ (Skupljač goriva) su vredni dodatci koji ćete vrlo retko koristiti za ono za šta je po imenu namenjen. Vlasnik „skupljača goriva“ može, naime, da pride bilo kom suncu (treba prići toliko da temperatura u kabini dostigne maksimum ali ne više od toga), postaviti se tako da vidi samo vrh zvezde i krenuti napred maksimalnom brzinom. Rezervoari će se ubrzo napuniti i bićete u prilici da kroz hipersvemir otputujete do susedne planete; odredite skoka treba odmah unapred izabrati, jer u uslovima visoke temperature nećete moći da mislite na to.

Verujemo da ćete se veoma retko upuštati u sakupljanje goriva: ovaj je sirovina relativno jeftina i nema je smisla skupljati uz ogroman rizik od pregrevanja. Ovak čete metod, dakle, koristiti samo ako ostanete bez goriva u blizini stanice koja šalje policijske brodove za vama i onemogućava vam sletanje. „Fuel scoops“ vam, međutim, omogućavaju jednu mnogo korisniju stvar: skupljanje tereta koji je ispuatio vasionski brod koji ste upravo uništali: svaki kontejner koji „upecate“ može da bude bezbedan (ako sadrži hranu ili tekstil) ili vredan čitavo bogatstvo. Uz malo hrabrosti možete da pokupite čak i kapsulu za spasavanje u kojoj se krije pilot neprijateljskog broda; on postaje vaš rob koga možete da prodate na prvom susednoj planeti.

„ECM system“ nije mnogo koristan, ali nije ni mnogo skup. On vam omogućava da uništite sve protivničke (na žalost, i sopstvene) projektile u vidokrugu i tako sa lakomom parirate piratima koji nose dirigovane rakete. S druge strane, dirigovana raketa se, uz malo truda, može zaobiti ili uništiti tako da čete taster „E“ pritiskati samo u trenucima panike kada vas nekoliko neprijatelja gada projektijama. Direktno pogodan projektila vas neće trenutno uništiti, ali će teško oštetiti vaše štitnike ili isprazniti bar tri od vaših četiri energetska rezervoara.

„Energy Bomb“ (Energjska bomba) je skup i malo isplativ dodatak koji čete kupovati tek kada vaše finansije ozbiljno uznapreduju. Svaka energetska bomba može da eksplodira samo jednom i tom prilikom uništi sve neprijateljske brodove, projektile, teret i ostale objekte u vidokrugu.

„Escape Pod“ (Kapsula za spasavanje) se koristi samo u specijalnim situacijama. Za trenutak će vas vratiti na stanicu gde čete od osiguravajućeg društva dobiti nov brod opremljen poput onoga koji ste izgubili; nedostajace vam jedino „potrošena“ kapsula za spasavanje i teret koji ste prenosili. Daleko je pametnije snimiti vaš status na disketu (ili traku) u svakoj stanici i u slučaju pogibe jednostavno ponovo učitati ovaj snimak.

„Elite“, igra koju smo predstavili još u „Računarima 4, nije više privilegija vlasnika BBC-ja i „elektrona“; verzije „Elite“ za „komodor 64“ i „spektrum“ su se pojavile sredinom 1985, da bi krajem iste godine, uz oglasne fanfare, pristigle i na jugoslovensko piratsko tržište. Zato smo odlučili da ponovo posvetimo pažnju ovoj fascinantnoj igri i da, kroz napise u ovim i sledećim „Računarima“, opišemo kako se stize do njenog kraja. Pri tom ćemo, naravno, obratiti posebnu pažnju na raznorazna „varanja“ koja će vam omogućiti da za veoma kratko vreme i praktično bez nпора dostignete magični status „Elite“.

Kompjuter za sletanje

„Docking Computer“ (Kompjuter za sletanje) je san svih početnika u „Elite“: da biste trovali, odnosno nabavljali gorivo i opremu, morate da sleteče na svemirske stanice, a pri tim sletanjima redovno ginetel Kupovina kompjutera za sletanje će automatizovati ovaj postupak i smanjiti rizik, ali ga neće potpuno ukinuti: čak i kompjuter za sletanje vas ponekad može, razlupati o stanicu, pogotovu ako ga startujete previše kasno. Nevolja sa kompjuterom za sletanje je što je veoma skup, tako da ćete, kada steknete novac za njegovu kupovinu, verovatno već savršeno dobro sletati i bez njega. Ipak ga kupite: videćete, pre svega, kako kompjuter upravlja brodom, a i čuti lepu interpretaciju kompozicije „Na lepom plavom Dunavu“ samo verzija za „komodor 64“. Dužni smo vam, naravno, nekoliko saveta za bezbedno sletanje bez kompjutera.

Otvor kroz koji se ulazi u stanicu je uviek okrenut prema planeti. Najlakše ćete ga pronaći ako se približavate stanici tako da vam ona ostaje sa desne a planeta sa leve strane. Gledajte kroz desni prozor sve dok ne ugledate stanicu, a zatim zakočite i okrenite brod prema njoj. Dajte umereni gas i pokušajte da usaglasite rotaciju broda sa rotacijom stanice; idealna je situacija u kojoj se vrata stanice prividno ne kreću. Sam ulazak ne morate da izvodite minimalnom brzinom kao što piše u uputstvu za upotrebu programa; uz dosta vežbe ćete čak naučiti da ulećete u stanicu punom brzinom. Najvažnije je da vežbate sletanje; nema nikakvog smisla da počinete trgovinu ili gusenare ako ne umete nepogrešivo da sletite i unovčite rezultate svog truda!

„Galactic Hyperspace“ (Hiper prostor) je veoma skup i teško ga je nabaviti. Omogućava vam da pređete u susednu galaksiju (galaksije su obeležene brojevima 1—9; iz devete ponovo dolazite u prvu) tako poselite čitav niz novih svetova. Skok u susednu galaksiju koji će potrošiti vaš „Galactic Hyperspace“ se naročito ne isplati — tamo nećete naći ništa novo, a i cene raznih roba se neće bitno razlikovati. Jedini korist od intergalaktičkog skoka je što drugu misliju doljaksite isključivo ako se nalazite u trećoj galaksiji.

Za opremanje broda je, dakle, potrebno dosta novca koji se može zaraditi isključivo trgovinom (i gusenare nije loš izvor prihoda, ali za njega treba dobro opremiti brod). Mnogo ćete pogrešiti ako počnete da trgujete pošteno — bankovni račun će vam rasti sasvim postepeno dok će vam neprijatelji neprekidno napadati da bi vam proteleli teret. Daleko je bolje obavljati manje ali unosnije poslove: švergovati naroklike. Naroklici se, naime, mogu kupiti za svega četiri kredita po kilogramu dok se na nekim drugim svetovima mogu prodati i za 104.

Obrzirom da, kada kupite „Large Cargo Bay“ (Veliko spremište), možete poneti 35 tona naroklika u jednom putovanju, zarada će iznositi celih 3500 kredita! Nekoliko švercerskih poduhvata i biće dovoljno opremljeni da se upuste u jednu još nečasnu rabotu: uništavanje policijskih brodova.

Igra za nepošteno

Za ovaj je zločin potrebno posedovati prednje i zadnje 'military' lesere. Izletite iz stanice i sačekajte da se malo udaljite od nje, a zatim potpuno zaustavite brod i aktivirajte zadnji prozor. Otvorite snažnu vatru iz lasera na ulaz u svemirsku stanicu (gadajte je i projektilom) i za trenutak će se na izlazu pojaviti dva policijska broda. Neka vas to ne zabrinjava — nekoliko hitaca će ih uništiti, a ta će sudbina zadesiti i sve koji ih budu sledili. Kada se vaš brod previše udalji od stanice, brzo ga okrenite za 180 stepeni, uništite policijske brodove koji su u međuvremenu izleteli i legoro se približavate stanici uništavajući nove prgonitelje. Za to će vreme vaš policijski status postati 'Fugitive', ali vas to neće previše brinuti — nekoliko skokova kroz hipersvemir i vaš će se potencija izgubiti. Važno je, međutim, da zapamtite dve stvari: nikada ne pokušavajte da sletite na stanicu u čijoj ste blizini postali 'fugitive' i nikada ne napadajte policijske brodove dok nosite naroklike. Sva je prilika da će vas;prattiti kordon policije, ali je vrlo neverovatno da će vas napasti. Ako se napad ipak desi, borba vam ne koristi — čak i ako uništite sve prgonitelje, nećete moći da sletite na stanicu. Jednostavno skočite do susednog sistema kroz hipersvemir. Uništavanje policijskih svemirskih lovaca vam nikada ne može doneti neku neposrednu korist; a neće vam biti ni zanimljivo; ono je, međutim, najbrži način da steknete viši rejting.

Uništavanje jednog broda (i postizanjem nekih drugih ciljeva) vaš se skor, naime, povećava za 1. Čim uništite 256 brodova, računac će ispisati poruku 'Right On Commander' (Razumem, komandante), i vaš će se rejting popeti za jednu stepenicu (rejtnji su, kao što verovatno znate, 'Hermless' (Bezopasan), 'Mostly Hermless' (Uglavnom bezopasan), 'Poor' (Slab), 'Average' (Prosečan), 'Above average' (Iznadprosečan), 'Competent' (Sposoban), 'Dangerous' (Opasan), 'Deadly' (Smrtonosan), i, konačno, 'Elite' (Elitan). U početku će svaka 'Right On Commander' poruka povećati vaš rejting za jednu stepenicu, ali će biti veoma teško preći poslednji korak — da biste se priključili 'Eliti', treba da dobijete 25 poruka, to jest da uništite oko 6500 brodova! Možete nam verovati na reč da nije ni malo lako postići ovaj skor. Jedini je način da unistavate sve brodove na koje nađete, ne razmišljajući da li je reč o pratima ili o mirnim trgovcima. Od starog kaubojskog principa „prvo pucaj;pa onda postavljaj pitanja“ treba odustati jedino u

sigurnosnoj zoni svemirske stanice; čak i ako vas gusari napadnu, bežite od njih ili skočite u susedni sistem. U protivnom ćete sebi naučiti na vrat poliju i to u situaciji u kojoj niste spremni da joj se suprotstavite (čak i ako ste spremni, okrajaj sa policijom znači automatsku zabranu ulaska u stanicu).

slika 2

10	REM
20	REM
30	REM
40	REM
50	REM
60	REM
70	REM
80	REM
90	REM
100	REM
110	REM
120	C#0
130	FOR I=6000 TO 6000
140	READ A
150	C=C-A
160	TI=A
170	NEXT I
180	IF C<=390 THEN
	PRINT "Proveri DATA liste":END
190	FOR I=6000 TO 6000
200	TI=0
210	NEXT I
220	*SAVE E.Krez 0000 +100 0 0
230	END
240	
250	REM Zamena linije 220 za
260	REM kasetnu verziju Elite:
270	
280	*SAVE Krez 0000 +100 0 0
290	
300	DATA 10,18,160,82,210,66,16,150
310	DATA 189,32,18,26,42,70,132,3
320	DATA 151,151,151,151,0,0,37,0
330	DATA 0,0,0,0,0,0,0,0
340	DATA 0,0,0,0,0,0,0,0
350	DATA 255,255,0,20,255,255,255,0
360	DATA 0,0,0,4,0,6,18,2
370	DATA 0,3,54,32,56,18,41,29
380	DATA 0,53,2,15,2,0,88,0
390	DATA 0,0,207,134

Ovaj napis završavamo programom sa slike 2: njegovim čete startovanjem na kaseti ili disku dobiti status 'Hermless' komandanta čiji je brod opremljen kompletnim inventarom uključujući i strahovito snažnu energetsku jedinicu koja se ne može nabaviti ni na jednom svetu. Dobar način da započnete igru, zar ne? Ne pokušavajte, ipak, da sami menjate podatke o komandantu (promenom 72. bajta iz DATA liste u &20 postali biste, na primer, 'Elita') jer su programeri Acornsofta u program ugradili veoma složen čeksum koji otkriva sve marifetluje. U idućim ćemo „Računarima“ otkriti tajne ovaog čeksuma i objaviti program pomoću koga ćete lako i brzo kreirati komandante proizvoljnih karakteristika.

Dejan Ristanović

**Spektrumovci
u nevolji**
**Kad spektrum
stane...**

smrtonosna dijagnostika

Kada govorimo o otklanjanju kvarova, u hronično slabim uslovima postojanja i delovanja servise mreže, pojavio se na tržištu priličan broj samostalnih poduzetnika koji nude usluge ovog tipa. Ti ljudi se bave popravljivanjem na manje-više amaterskoj osnovi, ali su uvek pravi spas za vlasnika neispravnog spektruma.

Nakon kvara

Servisere muči niz problema. Prvi i najveći je problem nabavke i visoka cena rezervnih delova. Centralni procesor i standardna logička kola uglavnom nisu problematični. Malo veća prepreka su 32K dinamički RAM-ovi. Bez podatka da su to, u stvari, pokvareni 64K RAM-ovi, ne može se mnogo uraditi. ROM se može programirati, ali je za to nepremostiv problem ULA. Za nju je još uvek jedini dostavljač Sinkler i njegovi distributeri.

Sledeći korak kod organizovanja servisa „u garaži“ je nabavka opreme i mernih instrumenata. Dobra lemilica i manji osciloskop su na gornjoj granici dostojnog. Na tom nivou možete otklanjati jednostavnije kvarove, dok je za komplikovanije potrebne logički analizator ili originalna Sinklerova naprava za testiranje. I jedno i drugo je izvan domašaja našeg servisa.

Olkšana dijagnoza

Osnovna ideja u ovom projektu je dijagnosticiranje pomoću jednostavnih programa koji se izvršavaju u centralnoj procesorskoj jedinici računara. Kada procesor izvršava samo jednu do dve instrukcije u petlji, mogu se na osciloskopu lepo videti tokovi signala. Pri normalnom radu to nije moguće, pošto CPU izvršava mnoštvo raznih instrukcija, pa se signali ne ponavljaju.

Kada se jednom tok pojedinog signala uhvati na ekranu osciloskopa, nalaženje greške više nije problem.

Uredaj za testiranje se uključuje u pokvareni spektrum pomoću standardne utičnice za proširenje sistemskog basa. Kada je uređaj priključen spektrumom ROM sa sistemskim programom i bejzikom je „ugašen“ a „upaljen“ ROM u tekstu. Ovak ROM sadrži zbirku programa za testiranje, koji se biraju pomoću preklopnika na komandnoj ploči uređaja.

Konstrukcija tastera

Prototip uređaja za testiranje sadrži dva ROM-a, odnosno eprama — jedan sa programima za testiranje — IC7, i kopiju spektrumovog ROM-a — IC8. Trenutno aktivni ROM se bira preklopnikom na komandnoj ploči — S4. Ovim je izbegnuto višekratno uključivanje i uključivanje uređaja za testiranje kad god želimo da aktiviramo spektrumov sistemski program.

Adresne linije spektrumovog basa se vode neposredno na oba ROM-a. Eprom sa programima za testiranje dobija od CPU samo adrese od A0 do A7, a A8 do A11 se postavljaju pomoću preklopnika S0, S1, S2, S3. Na taj način se bira program za

testiranje, koji se izvršava nakon što procesor primi signal RESET.

Vidimo da eprom sa programima za testiranje sadrži najviše 16 programa, najveće dužine od 256 bajtova svaki.

Dvoja „NAND“ vrata IC3/1,2,3 i IC3/4,5,6 su vezana kao multivibrator. Stanje multivibratora se bira pomoću dvostrukog trenutnog preklopnika S4. Izlazi TEST odnosno SPECTRUM odlučuju u kojem režimu uređaj radi. Svaki od signala omogućuje rad jednog od dva adresa dekodera. Integrirana kola IC4/9, 10, 11, 13, 14 — IC3/8,9,10 — IC3/11,12,13 — IC6/1,2,4,5,6 signalom TEST=1 uključuju eprom sa programima za testiranje, dok IC4/2,3,5,6,7 — IC6/8,9,10,12,13; signalom SPECTRUM=1 uključuju eprom sa sistemskim programom.

Treći adresni dekodier IC5/1,2,3 — IC5/4,5,6 omogućava programsko postavljanje stanja bistabilnih multivibratora IC2 kada je signal IZLAZ=0. Izlazna stanja se prenose na otpornike R1 do R8 ako je S6 vezano na masu. Logičko „1“ na bilo kom od izlaza IC2 dovodi odgovarajući tranzistor T1 do T8 u stanju zasićenja, te kolektorskom strujom pali odgovarajuću svetleću diodu D1 do D8.

Izlazni sklop reaguje na izvršavanje OUT instrukcije u CPU, sa adresom 1101 1111 binarno, odnosno &DF. Stanje akumulatora se prenosi na IC2, odnosno, priključenom S6, na LE diode.

Signal ROMCS na spektrumovom basu se postavlja na logički nivo „1“, čime se „gasi“ unutrašnji ROM, a trenutno povezivanje linije RESET na masu preko S5 omogućava uspostavljanje početnog stanja u spektrumu.

IC1 sa filterima C1, C2, C3 snabdeva logička kola uređaja električnom strujom za napajanje. Priljučnici za napajanje ne nalaze se na svakom integrisanom kolu na istom mestu. Njihov raspored (brojevi nožica) dat je u sledećoj tabeli:

+5V	IC1/3	IC2/20	IC3/14	IC4/16	IC5/14	IC6/14	IC7/24	IC8/28
GNP	IC1/2	IC2/10	IC3/7	IC4/8	IC5/7	IC6/7	IC7/12	IC8/14

Programska oprema

Program za testiranje dat je u assembleru na listingu. 1. Startovanjem programa u računaru, će biti proizveden odgovarajući kod koji treba preslikati u eprom. Osim tri kratke rutine, sve ostalo će biti bajtovi & 76.

Programiranje uređaja za testiranje u okvirima datih mogućnosti hardvera ograničeno je jedino maštom konstruktora. Vođeći princip: je prilagodjenje programa tipu greške koja se dijagnostičira, te koncipiranje testova od jednostavnijih ka složenijim.

Pogledajmo kako su ta načela uzeta u obzir u našem prototipu.

Na početnoj adresi prvog testa, 0000, nalazi se skok na samog sebe. To je najjednostavniji pokušaj da se ustanovi da li

HISOPT GEN53M2 ASSEMBLER
ZX SPECTRUM

Copyright (C) HISOPT 1983,4
All rights reserved

Pass 1 errors: 00

	1	%C-		
00DF	10	PORT	EQU	#DF
	20			
C000	30		ORG	#C000
C000	40			
C000	50	LOOP	JR	LOOP
	60			
C100	70	ORG	#C100	
	80			
C100	90		LD	B,#00
C102	100		LD	C,#00
C104	110	LOOP_1	LD	HL,#4000
	120	LOAD_1	LD	A,L
	130		XOR	B
	140		XOR	B
	150		LD	(HL),A
	160		INC	HL
	170		LD	A,H
	180		CP	#80
	190		JR	NZ,LOAD_1
	200		LD	HL,#4000
	210	READ_1	LD	A,L
	220		XOR	B
	230		XOR	B
	240		XOR	(HL)
	250		OR	C
	260		LD	C,A
	270		OUT	(PORT),A
	280		INC	HL
	290		LD	A,H
	300		CP	#80
	310		JR	NZ,READ_1
	320		INC	B
	330		JR	LOOP_1
	340			
C200	350		ORG	#C200
	360			
C200	370		LD	B,#00
C202	380		LD	C,#00
	390	LOOP_2	LD	HL,#8000
	400	LOAD_2	LD	A,L
	410		XOR	B
	420		XOR	B
	430		LD	(HL),A
	440		INC	HL
	450		LD	A,H
	460		CP	#00

uopšte rade CPU i bas. Ukoliko pokušaj ne uspe, CPU „izleće“ u HALT instrukciju (76), procesor se zaustavlja i — greška će brzo biti locirana!

Sledeći testovi se bave memorijskim bankama. Na & 0100 je test za memorijsku banku na adresi &4000. Taj RAM imaju svi spektrumi, a koriste se za sistemske varijable, ekran i korisnikov program. Memorija se prvo puni orednim sadržajem, a onda čita i proverava. U slučaju greške, pali se jedna od svetlećih dioda na izlaznim vratima & DF. Redni broj LED diode označava koji od RAM čipova je u kvaru.

Na adresi & 0200 nalazi se sličan program za proveru delovanja memorijske banke na adresi &8000. Ovu banku imaju

Spektrum, u tehnološkom smislu, spada među najslabije mašine na tržištu. Zbog komercijalnih razloga, on je izraden prilično amaterski — kako konstrukcijom tako i pouzdanošću rada, on spada u donji razred klase „zabavne elektronike“. U doba uvođenja u prodaju, to mu je, između ostalog, dalo odlučujuću prednost pred konkurentskim proizvodima, dok danas još samo izaziva glavobolju velikom broju vlasnika. Ali, šta je tu je: oni imaju spektrum moraju da žive s njim i — stalnim strahom od iznenadnog kvara! Stoga uređaj za dijagnostiku kvarova nije namenjen samo potencijalnim serviserima, nego i nepreglednoj armiji uradi-sam amatera kojima će se kad-tad neko od rodbine ili prijatelja obratiti za pomoć.

```

C20F 470 JR NZ_LOAD_2
C211 480 LD HL,#8000
C214 490 READ_2 LD A,L
C215 500 XOR H
C216 510 XOR B
C217 520 XOR (HL)
C218 530 OR C
C219 540 LD C,A
C21A 550 OUT (PORT),A
C21C 360 INC HL
C21D 570 LD A,H
C21E 580 CP #00
C220 590 JR NZ,SPAD_2
C222 600 INC B
C223 610 JR LOOP_2
C225 620
C300 630 ORG #C300
640
C300 650 LD C,#00
C302 660 SET ON LD HL,#0000
C305 670 WAIT INC HL
C306 680 LD A,H
C307 690 OR L
C308 700 JR NZ,WAIT
C30A 710 INC C
C30B 720 LD A,C
C30C 730 OUT (PORT),A
C30E 740 JR SET_ON
750
E000 760 ORG #E000
770
E000 780 HALTS LD HL,#C002
E003 790 LD DE,#C003
E006 800 LD BC,#0FF-#002
E009 810 LD (HL),#76
E00B 820 LDIR
E00D 830 LD HL,#C125
E010 840 LD DE,#C126
E013 850 LD BC,#1FF-#125
E016 860 LD (HL),#76
E018 870 LDIR
E01A 880 LD HL,#C225
E01D 890 LD DE,#C226
E022 900 LD BC,#2FF-#225
E023 910 LD (HL),#76
E025 920 LDIR
E027 930 LD HL,#C310
E02A 940 LD DE,#C311
E02D 950 LD BC,#FF-#310
E030 960 LD (HL),#76
E032 970 LDIR
E034 980 RET
990
E000 1000 ENT HALTS

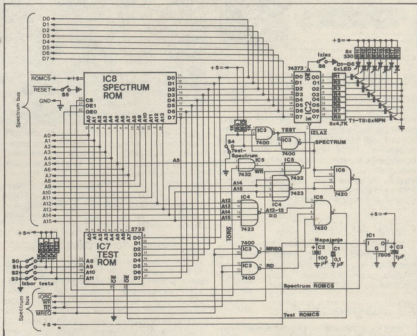
```

Pass 2 errors: 00
Table used: 149 from 800
Executes: 57344

Program TEST

Ovaj program će generisati kompletan kod (4096 bajtova) spreman za unošenje u EPROM. Treba resetovati računar i učitati DEVPAC GENS assembler počev od adrese 30000. Posle startovanja assemblya, ukucati ceo program i assemblerati ga. Zatim izvršiti naredbu R, koja će startovati program od adrese HALTS (E8000). Time se generišu sve potrebne HALT instrukcije. Po povratku u bezik, formirani kod se može snimiti na kasetu, ili se direktno preneti u EPROM. Početak bloka je na adresi &C000 (49152), a dužina 4096.

61/smrtonosna
dijagnostika



Spisak sastavnih delova

Kod	Broj	Elementi	Dodatni podaci
IC1	1	7805	Stabilizator napona 5V/1A, kućište T0220
IC2	1	74373	TTL osmostruki latch, kućište DIL20
IC3	1	7400	TTL četvorostruki NAND, sa po 2 ulaza, kućište DIL14
IC4	1	7423	TTL dvostruki NOR, sa po 4 ulaza, kućište DIL16
IC5	1	7432	TTL četvorostruki NOR, sa po 2 ulaza, kućište DIL14
IC6	1	7420	TTL dvostruki NAND, sa po 4 ulaza, kućište DIL14
IC7	1	2732	EPROM 4K x 8/450nsek, sa programom za testiranje Spectrum ROM (Nije nužno potreban)
IC8	1	T1-78 B	NPN tranzistor
D1-D8	8		Svetleća LED dioda, boja crvena
C1	1		Keramički kondenzator 0,1µF/50V
C2	1		Elektrolitski kondenzator 100µF/25V
C3	1		Tantalski kondenzator 1µF/15V
R1-R8	8		Karbonski otpornik 4,7K/0,25W
R9, R10	2		Karbonski otpornik 1K/0,25W
R11-R18	8		Karbonski otpornik 330E/0,25W
R19-R22	4		Karbonski otpornik 1K/0,25W
S0-S6	7		Mikropreklopnik DPDT
	1		Prototipna štampana pločica 100 x 160
	1		Konektor EDGEBOARD (28 x 28) x 0,1"

samo spektrumi 48K, a koristi se isključivo za korisnikove programe.

Na adresi &0300 se nalazi program koji na svetlećim diodama izvodi mali „light show“.

Uz utrošak srazmerno skromnih sredstva može se izraditi vrlo snažno pomagalo za otkrivanje grešaka i kvarova na spektru. Za rad sa uređajem nije potrebno posebno poznavanje centralne procesorske jedinice i ostalih logičkih kola. Kod upotrebe logičkog analizatora, koji košta nekoliko desetina pa i stotina puta više, serviseri i konstruktori moraju biti dobro potkovani i ovakvom vrstom znanja.

Stefan Kirn, Karel Svetek, Davor Viličić



Šta ima novo

Inteligentni displej

Američka firma Litronix (sada u sastavu Siemens-a) nudi veliki izbor inteligentnih višecifarskih LED pokazivača koji se lako povezuju sa postojećim mikroprocesorima CPU ih vidi kao RAM) pošto sadrže i interni generator ASCII karaktera, mogućnost upravljanja kursorom, radni napon je standardni 5V. Oznake ovih komponenti koje su idealne da se na njima napravi inteligentni DISPLAY su DL 1416, DL 2416, DL 3416, DL 1814 (osam ASCII znakova).

Prvi u Evropi

Siemens je prva evropska firma koja je započela proizvodnju dinamičkih memorija tipa RAM kapaciteta 256 K (postoje dva modela sa po tri brzinske verzije: vremena pristupa su 120/150/200 ns).

Ka superkompjuteru

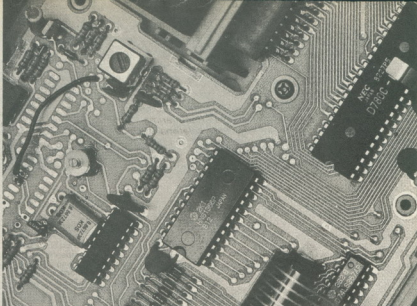
Ako se spremate da pravite dobar računar, morate znati gde su vam (komercijalne) granice: mikroprocesori se prave do 32 bita, UV EPROMi sa kapacitetom do 512 Kbita (planira se 1 Mbit), CMOS EPROMi do 256 Kbita, ROM-ovi (Factory Masked) do 1 Mbit, dinamičke memorije do 256 Kbita (uskoro 1 Mbit), statičke RAM memorije (isključivo CMOS) do 64 Kbita, pseudostatičke (periferija CMOS, a interna dinamička matrica HMOS) do 256 K, Floppy diskovi od 5.25 inča do 1,6 Mb, Winchester diskovi od 5.25 inča do 280 MB, SMD diskovi reda gigabajta, itd, itd. Naravno pod „komercijalno nabavljivim“ komponentama podrazumevamo ono što se danas može kupiti na tržištu, a ne ono što se najavljuje u reklamama, ili je još uvek „u razvoju“.

Mini vinčester

Već danas se prave „mali“ Winchester diskovi (3.5 inča) sa kapacitetom od oko 13 megabajta (rodine).

RAM...

Ako ste se već odlučili da napravite radnu memoriju sa RAM dinamičkog tipa kapaciteta 256 K, onda bar iskoristite 8419



DRAM kontroler (National Semiconductor), koji obavlja celokupan „posao“ bez usporavanja do CLOCK-a od 16 MHz (bez WAIT stanja) i lako se povezuje sa praktično svim CPU (mada najbolje radi sa 32032). Primjenjive DRAM sa pristupom od 150 ns ili sporije, a ako se ovom kontroleru pridoda DP 8400-2 ECC Unit (Error Checking and Correction), onda ovaj set garantuje uvek tačne podatke iz memorije.

... i disk kontroleri

Do pre nekoliko godina, kontroleri iz floppy disk jedinice su bili građeni kao štampane ploče sa preko dve stotine TTL (SSI i MSI) integriranih kola; danas se takav sklop pravi sa jednim MOS LSI kolom (Floppy Disk Controller — FDC) i nekoliko TTL bafera. Najveći izbor FDC LSI kola ima američka firma Western Digital (po licenci ih pravi i Siemens), a jedan od najkompleksnijih među njima je FDC sa oznakom WD 2797 koji podržava sve vrste diskova, PLL data separator, prekompenzacija upisa, standardne veze (Interface) kako prema disk jedinici tako i CPU basu, transfer podataka iz/u memoriju ide direktno (DMA) po sektorima, a mikroprocesor sa njim komunicira lako razumljivim „makroinstrukcijama“.

Statičke memorije

Statičke RAM su dostigle zavidnu popularnost u mikroročunarima zahvaljujući, pre svega, dobrim osobinama prvih modela na tržištu Hitachi serija HM6116 od 16 Kbita (2 Kx8), kakva se koristi i u „galaksiji“, a zatim i HM6264 od 64 Kbita, kao što su solidna brzina (reda 100 ns) i niska potrošnja (aktivna struja 40 mA, „stand-by“ potrošnja reda 100 uA). Iako ih Hitachi u katalogu predstavlja kao CMOS statičke RAM, one su napravljene tzv. „MIXED CMOS“ tehnikom (što znači NMOS matrica kao „srce“ memorije, a CMOS periferija), dok Toshiba pravi slične tipove statičkih RAM (to im je specijalizovano) čistom CMOS tehnologijom, što je skuplje, ali daje bolje rezultate: uz istu brzinu, aktivna struja je reda 5 mA, a „stand-by“ potrošnja samo 0,2 uA (maxi). Obe firme su najvile i 256 K tipove (32 Kx8) ali im je pristup drugačiji (Hitachi pravi pseudostatički RAM).

Prekidački regulatori

Blok napajanja u malim personalnim računarima je obično izveden kao klasičan ispravljač plus linearni regulator za 5V (7805 u „galaksiji“ ili „spektrumu“ je dobar primer fiksnog naponskog regulatora). Međutim, kako su lični računari dobijali u kompleksnosti (potrošnja energije raste), tako su i ispravljači/regulatori postajali sve veći (veća disipacija, veći hladnjaci, veća težina, manja pouzdanost), što je neminovno uslovlilo primenu prekidačkih izvora napajanja tzv. „Switching Mode Power Supply — SMPS“) koji su „do juče“ imali svoje „deće bolesti“. Razvojem modernih prekidačkih komponenti POWER MOS-FETs, Feriti, kondenzatori sa malim ESR/ESL, IC za pobudu i regulaciju SMPS) ovi izvori napajanja naglo zamenjuju linearne regulatore zbog mnogo veće efikasnosti (znači, mala disipacija, mali hladnjaci, mala težina i zbog feritnih transformatora), malih dimenzija (posledica rada na nekoliko stotina kHz) i veće pouzdanosti.

Ako vam je potreban izvor napajanja za 5V i nekoliko ampera (2—3), možete izabrati između monolitnog regulatora firme SGS (ista je inače poznata po svojim POWER komponentama) tipa GS-R400 koji daje 5V pri 4 A uz skoro 90% efikasnost, ima svoj mali hladnjak, dodatne izlaze (npr. reset), tačnost 1% pri ulaznim naponima od 12—30 Volti i Hibridni SMPS regulatora firme RIFA (PKA serija) koji pored 5V daje i druge napone (npr. +12V za serijski I/F), ima snagu od 40 W uz efikasnost od 85%, radi na 300 kHz, dimenzije reda 9x9x2 cm, regulacija izlaznog napona je 10 mV i ima srednje vreme izmedu otkaza (MTBF) od 1.7 miliona časova (preko 200 godi), zahvaljujući činjenici da nema ugrađenih elektrolita. Ove POWER komponente su napravljene za princip „ugradi-pa-zaboravi“, a njihove mane (koje postoje) nisu bitne za digitalne sisteme. Linearni regulatori će i dalje imati primenu u analognim kolima zbog bržeg odziva i manje talasnosti. U računar BBC, inače, odavno se ugrađuje prekidački regulator.

Priprema: Miše Blažimir, dipl. Ing.

Mali oglasi

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

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

Ako ne volite da se dopisujete sa „Računarima“, svoj mali oglas možete nam izdiktirati preko telefona 011/650-161 svakog radnog dana od 10—14 sati. Mi ćemo vam onda naknadno poslati ispunjenu uplatnicu.

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

CENA OBIČNOG MALOG OGLASA do dvadeset reči je 600 dinara. Svaka naredna reč košta još 40 dinara, s tim što oglas ne sme da ima više od 50 reči. Adresa oglašivača ne računa se u cenu.

CENA UOKVIRENOG MALOG OGLASA je 600 dinara po visinskom centimetru, s tim što se mogu zakupiti najmanje po 32 slova znaka. Ako se ne iskoristi čitav prostor u jednom redu, računa se broj redova a ne broj znakova. Za uokvirene oglase preko 5 cm cena je 900 dinara po centimetru.

Poželjno je da vaš mali oglas počinje sa Prodajem, Kupujem, Držim časove, Menjam... ili nečim sličnim što ukazuje na sadržaj oglasa.

Svi mali oglasi koji stignu u našu redakciju do 28. februara do 12 sati biće objavljeni u „Računarima“ broj 13 koji izlaze iz štampe polovinom marta.

Da ne bi bilo zabune, obavezno naznačite da li želite običan ili uokviren mali oglas, i zajedno sa tekstom vašeg malog oglasa pošaljite i priznanicu o uplati na adresu redakcije: „GALAKSIJA“, BULEVAR VOJVODE MIŠIĆA 17, BEOGRAD, sa naznakom „za male oglase u Računarima“.

ŠPEKTRUM

ŠPEKTRUMOVCI! — HOBBY ŠPEKTROVCI optički programi po niskim cenama. Usluga brza i kvalitetna. Snimanje na PANASONIC kasetofonu za kompjuter. Besplatan mali katalog. Dakačević Igor, Karlovačka 16, 11080 Zemun, tel. 011/219-702. Nećemo vas iznevrediti!

• ANALOG 20 — za sve koji žele upoznati svijet analognih kompjutera na Špektrumu. Brzo i stručno rešavanje svih dinamičkih sistema iz oblasti strojarstva, elektrotehnike, robotike, NC upravljanja, regulacije, nelinearnih diferencijalnih jednačina, operacione istraživanja. Programski paket za stručnjake i one koji to žele postati. ALPLACE CONTROL SOFT, Cesta na Svetinju 19, 63270 Laško

□ Špektrumovci — OFB COMPUTER STUDIO vam nudu komplet od 10 programa po cijenji od 700 din., (+400 din. kasete i poštarina). Komplet 1 sadrži programe kao napr. SNIPI, GLASS, SOHO SEX, QUEST, W.S. BASKETBALL itd. Prva dva kupca od 20.3.1988. plaćaju samo kasetu i poštarinu. Omer Eškić, Fra Grpe Martića 31, 78000 Banjaluka, tel. 078/31-422

□ Prodajem Špektrum Plus, oceanarini, interfejs, džojstik, literaturu i preko 20 programa. Milanović Vladimir, Kristijana Karpošića 43, 92000 Šip, tel. 092/21-232

• BLAST — ubrzajte svoje brzicke programe do 40 puta upotrebom do sada najboljeg kompjajera za Špektrum. Traka sa BLAST-COMPILER-om i BLAST TOOLKIT-om i opširno uputstvo — 1500 din. MACHINING CODE FOR BEGINNERS — uživajte u efikasnosti mašinskog programiranja. Traka i opširno uputstvo — 1500 din. Oba programa sa uputstvima — 2500 din. VRCA Milan, Zarija Vujkovića 79, 11070 Novi Beograd

□ Špektrumovci! Prodajem programe po najnižim cenama. Tražite besplatan ilustrovani katalog. IMPOSSIBLE MISSION, WINTER GAMES, F. WARRIOR, POPEYE, MARSPORT, ELITE. Tadić Slobodan, Rumenakić ul. 5, 21000 Novi Sad, tel. 021/317-910 ili 311-576

• Špektrumovci! MERLIN SOFT — najbolja YU piratska kompanija! Objavljuje i stavlja u prodaju specijalnog aviona na beogradski aerodrom na liniji London—Beograd. U avionu su najpoznatiji svetski soft hitovi: Pentagon, Myra Mare, Mugsy's Revenge, Superman, Critical Mass, Yie Ar Kung Fu, Transformers, Winter Games, Shadow of Unicorn, Fairlight... Petrović Aleksandar, Olge Alkaias 7/113, 11000 Beograd, tel. 011/4880416

• Verovali ili ne GHOSTBUSTERS STUDIO prodaje komplete najnovijih programa za ZX Špektrum po ceni od 600 din. Babić Srđan, Generala Ždanova 48, 11000 Beograd, tel. 659-031

• Špektrumovci — NEEDLE — program za presnimavanje (anima 99% programa). Posjeduje opciju za ubacivanje POKE-ova direktno u program prilikom presnimavanja — pouzdano. Program+uputstvo+100 besplatnosti=790 din. (Na vašu kasetu za 400 din. Pouzedećem Branko, pp 57, 47300 Ogulin, tel. 047/72-289

□ Prodajem Špektrum 48K, mikroiraj, Za interfejs, printer — piscu mašinu Brother EP-22, sve potrebne kablove, papir i trake za printer, programe. Čirić Zoran, Splitska 4, 11000 Beograd, tel. 011/414-274

• 12 SUPER HITOVA za Špektrum samo 700 din. + kasete. D. T. Superest 1 i 2, Dambusters, Exploding Fish, Nodes of Yesod, Boulder Dash 2, Macadam Bumper, Impossible Mission, Fighting Warrior, Sorcery, International Karate, Skool Daze 2, ASTERIX Club, Ljube Nenodovića 1/6, 11500 Obrenovac, Nikolić Dušan, tel. 011/873-777

□ ŠPEKTRUMOVCI, trenutno najbolji programi pojedinačno i u kompletima. Komplet 8: BACK TO SKOOL, SUPER PIPELINE 2, BEACH HEAD 2, FIGHTING WARRIOR, W.S. BASKETBALL, POPEYE, COMMANDO, RASPUTIN, SABOTEUR, Mc.MADAMS BUMPER, BOUNTY BOB, ROLLER COASTER. Komplet 7: SKOOL DAZE 2, BEACH HEAD 2, BOULDER DASH 2, INTERNATIONAL KARATE 1, INTERNATIONAL KARATE 2, IMPOSSIBLE MISSION, SORDERONS SHADOW, DYNAMITE DAN, MONTY ON THE RUN, EXPLODING FISH, DUNDRACH, FRENKIE. Jedan komplet + kasete + poštarina, 1400 — din. Nastasović Srđan, Oslobođenja II deo 6, 11194 Beograd — Rusian, tel. 011/888-222

• MOBIL SOFTWARE — najbolji hitovi za Špektrum (EliteII, Winter Sports, Commando) i mnogo drugih. Cena pojedinačno 50 d., u kompletu 25 programa za 1000 d. Katalog besplatno. Goković Dragomir, Ljermontova 2/151, 11104 Beograd, tel. 011/4881-758

□ Uporabe programe za špektrum programom, izrađujemo programe po narudbi. Možete nabaviti mnogo novih programa (ŠTA JE MISSION 2 + kasete, 1000 din.) (Tražite Špektrum, Brelih Andrej, Titova 18, 64270 Jesenice ili Brišar Ludo, Tavčarjeva 3b, Jenice)

□ ŠPEKTRUMOVCI, već tri godine stojim vam na usluzi svojim visokom profesionalnošću i najvećim mogućim izborom starijih i najnovijih programa. Garantiram kvalitetu i ekspresnu isporuku. Naručite katalog (50 d.) sa oznakom „za Špektrum“. Prodajem i kompletne i pojedinačne programe (1000 dinara plus kasete), uz velike popuste za veće narudžbe. Za odrasle poseban program „Sex Mission“ (sa uputima, kazetom i troškovima otpreme 950 d.). FUTURE ORION, 11000 ZAGREB, Rubčićević Z, tel. 041/417-052. □ Prodajem mikroiraj i interfejs 1 za špektrum. Tel. 011/214-880

□ ŠPEKTRUM — NAJNOVIJI I NAJBOLJI PROGRAMI U KOMPLETIMA. KOMPLET 48: YIE AR KUNG FU, ELITE, RAMBO, RING OF WOOD, FAIRLIGHT, MIKIE, SIR FRED, TALOS, WRIGGLER, CLUEDO, TRANSFORMERS, SUPER BRAT. KOMPLET 50: B.C.'S QUEST FOR TILES, CRITICAL MASS, CYCLO ATTACK, ROLLER COASTER, JETSET WILLY 3, CHAOS, NIFTY LIFTY 3, ITD. JEDAN KOMPLET+KASETA+POŠTARINA 1250 d. KATALOG BESPLATAN. TRITICA GORAN, STEVANA LUKOVIĆA 9, 11090 BEOGRAD, TEL. 011/563-348

□ Prodajem nov nekorisćen lajpen za špektrum. Lajpen+uputstvo+kasete poklon program+poštarina 8000 din. Tel. 022/413-342

• ŠPEKTRUM — PROFESIONALNI PREVOĐI: NAPREDNI MAŠINSKI JEZIK 1500 d. ŠPEKTRUM ROM DISASSEMBLY 1500 d. MAŠINSKI JEZIK ZA APSOLUTNE POČETNIKE 1300 d. ŠPEKTRUM PRIRUČNIK 800 d. MEGA BASIC UPUTSTVO 500 d. MEGA BASIC NA KASETI 500 d. DEVPAC 3 UPUTSTVO 600 d. DEVPAC 3 NA KASETI 500 d. TINI BASIC PROGRAMIRANJA 500 d. BETA BASIC 1.8 UPUTSTVO 400 d. BETA BASIC 1.8 NA KASETI 500 d. ARTIST UPUTSTVO 400 d. ARTIST NA KASETI 500 d. TRAZITE BESPLATAN KATALOG. TRITICA GORAN, STEVANA LUKOVIĆA 9, 11090 BEOGRAD, TEL. 011/563-348

□ Menjam i prodajem programe za špektrum. Kompleti od 14—48 programa samo 500 din. Katalog 50 din. Star War Soft, Mariborska 123, 35000 Svetozarevo, tel. 035/26-585

• ŠPEKTRUM — NAJNOVIJI I NAJBOLJI PROGRAMI U KOMPLETIMA: KOMPLET A: SKY RANGER, SABOTEUR, BACK TO SKOOL, RASPUTIN, I OF THE MASK, NEVER ENDING STORY, SORD ERON'S SHADOW, HACKER RATS. KOMPLET B: FIGHTING WARRIOR, BEACH HEAD 2, BOUNTY BOB STRIKES BACK, IMPOSSIBLE MISSION, DYNAMITE DAN, BOULDER DASH 2, GLASS, SUPER GRAN, MACADAM BUMPER, SORCERY, INTERNATIONAL KARATE 1 i 2. KOMPLET C: POPEYE, W. S. BASKETBALL, SUPER TEST 1 i 2, MONTY ON THE RUN, MARSPORT, SUPER PIPELINE 2, ACTION BIKER, DAM BUSTERS, TWO GUN TURTLE, THAT'S THE SPIRIT, ABU SIMBEL JEDAN KOMPLET+KASETA+POŠTARINA 1250 d. TRAZITE BESPLATAN KATALOG SA 1000 PRITACIMA. TRITICA GORAN, STEVANA LUKOVIĆA 9, 11090 BEOGRAD, TEL. 011/563-348

□ Rečnik, englesko-srpskohrvatski (oko 1400 najčešće upotrebljivanih reči)+kasete +poštarina (1300 din.). Tel. 011/497-682 od 17—19 h.



Sve za ZX SPEKTRUM
— video igre
— namenski programi
— programska uputstva
— literatura
— SPECIJALNO —
BETA BASIC 3.0 sa
kompletnim originalnim
uputstvom, kasetom i
poštarninom za samo 1000.-d.

Za sve usluge garancija!
MILOVANOVIĆ LJUBIŠA
Petra Lekovića 57, 11030
BEOGRAD
tel: 011/558-007 posle 17
časova

● SPEKTRUM — 12 PROGRAMA ZA UČENJE ENGLESKOG + KASETA 1000 d. RADIOSKOOL MATEŠKI PROGRAM + KASETA 1000 d. 38 COPY PROGRAM + KASETA 1000 d. TRAZIŠTE BESPLATAN KATALOG SA NAJNOVIJIM PROGRAMIMA. TRTICA GORAN, STEVANA LUKOVIĆA 9, 11090 BEOGRAD, TEL. 011/563-348

SPEKTRUM — VELIKI IZBOR NOVIH PROGRAMA IZ LONDONA POJEDINAČNO I U KOMPLETIMA PO POVOLJNIM CENAMA. SABOTEUR, BACK TO SKOOL, I OF THE MASK, TRAZIŠTE BESPLATAN KATALOG SA NOVIH PROGRAMIMA. RANKOVIĆ MIRAN, BRAĆE MIHAJLOVIĆA 46, 11273 BEOGRAD

□ VRHUNSKI GRADEVINSKI PROGRAMI za ZX Spektrum: OKVIRI, REŠETKE, ROŠTILJI, PILOTI, DIMENZIONIRANJE, VODOVOD i mnogi drugi. Za radne organizacije pojedince. Besplatan katalog: Gino Gracin, Kozala 17, 51000 RIJEKA, tel. 051/517-291

□ 5 — SOFT nudi vam kompletne programe za spektrum. Komplet sa vašom kasetom 700 din., sa mojom 1000 din. Katalog 100 din. Milan Vunjak, Mrakovičića 50/8, 11000 Beograd, tel. 011/535-532

□ SPEKTRUMOVCI — COMMANDO, RASPUTIJE, SABOTEUR, ROBIN HOOD, MARSFOT, ELITE WINTER GAMES, ARTIST, TURBO... po ceni od 80 din. Svaka narudžbina 200 POKE-ova i 3 programa na pakob. Besplatan katalog, Saša Lubudović, Jovana Tomaševića 16, 85000 Bar, tel. 085/27-616

□ Prodajem programe za Spektrum, pojedinačno: ELITE, QUEST FOR TIRES, RAMBO 2, ROBIN HOOD, WRIGGLER, ENIGMA FORCE, CRITICAL MASS, COMMANDO, S.H. POP QUIZ, Tel. 021/716-024, Kisić Zoran, Dr. Madlena Stojanovića 8, 21201 Rumenka

OBRADUJTE SVOJ RAČUNAR! OBAVIJATE MU MALI OGLAS U „RAČUNARIMA“!

□ SUPER TURBO II ubrzava učitavanje — snimanje (Program, kasetna, uputstva 1000 din.). Englesko-srpskohrvatski rečnik sa 2000 reči (program, kasetna, uputstvo 1000 din.). Hit u Engleskoj — program ELITE sada i za Spektrum. Cena programa 400 din.+kasetna, Krištić Aleksandar, Balzakova 30, 21000 Novi Sad

□ Prodajem programe za spektrum. Komplet od 14 programa 450 din.+kasetna. Sa mojom kasetom 850 din. Stojanović Volkan, Marsala Tito 42, 2/4, 35230 Čuprija, tel. 035/64-777 ili 27-979

□ SPEKTRUMOVCI!!! KOMPLET 14: Winter Sports (8 disciplina), Commando, School Battle, Kung Fu, 2. Roller Coaster Island, Mopp, Metagalactic Llamas, Special Delivery, Crazy Caverns, Lancer Lord, KOMPLET 15: Rasputin, Robin Hood, Boulder Dash 2, Rambo, B.C. Quest for Tires, Critical Mass, 10 Little Indians, Mardons Quest, Merry Xmas, I of the Mask, Sky Ranger, S. Striker. Cena jednog kompleta 800 din. Radović Branislav, Sonje Marinkovića 14/4, 21000 Novi Sad, tel. 021/28-682 ili 022/424-624 (vikendom)

□ SPEKTRUMOVCI, SUPERKOMPLET 16: Elite, Enigma Force (Shadowfire 2), Wriggler, Stripper 2, Uncle Tom, Sapphire, Transformers, Mickie, Hyperblaster, Samurai Warrior, Gyroscope, Pacific War. Cena: 800 din. Radović Branislav, Sonje Marinkovića 14/4, 21000 Novi Sad, tel. 021/28-682 ili 022/424-624 (vikendom)

□ SPECTRUM RAINBOW SOFTWARE vam nudi izbor od 70 raznih kompleta. Kompleti od 25 programa 800 din. Imamo i sve najnovije programe koji su trenutno već u Jugoslaviji. Uverite se Katalog je besplatan. Mihajlovići Kičko, Moša Pijade 128, 91300 Kumanovo, tel. 0901/23-800

□ Prodajem kvalitetne snimljene programe za Spektrum. Cena bilo kojeg programa je samo 50 dinara. Stigli su svi novi programi koje ste čitali u drugim oglasima. Janković Silard, Petefi Šandora 84, 25222 Telečka

KOMODOR

□ KOMODORCI najpopularnije igre januar 86: GHOSTBUSTERS 2, JUNGLE HUNT 2, SWS 3, BOUNDER DASH 3, ZORRO +kasetna 1200 din. Može i pojedinačno. Petković Dejan, Milana Rakića 28, 11000 Beograd, tel. 011/424-744

□ KOMMODORE 20, 16, +4, 128, 64 programi. Tražite besplatan katalog, Derman Sandor, Rade Končara 23, 23000 Zrenjanin

□ Za C-64 veliki izbor najnovijih programa. ZORRO, WAR GAMES, FISTFULL OF BOOKS, DYNAMITE BOLD, BOUNDER DASH 3 i drugo. Besplatan katalog. Todоровić, tel. 011/458-134

Novo za Komodor 64! Prerađivanje programe — Program koji čekate dve minute možete loadirati za svega 30 sekundi. Prerađeni program ima auto-start. — Na disketu možete staviti tri puta više programa. — Prerađeni program nema zaštitu. — Besplatno dobijate FILE COPY kojim kopirate prerađene programe.

— Ušteda 1: u vremenu loadiranja — Ušteda 2: potrebno vam je tri puta manje disketa. Ovaj oglas je za vas: — ako radite programe... — ako skidate zaštitu... — ako prodajete programe... — ako nemate mnogo disketa... — ako se igrate... — ako menjate programe... (Sve ovo samo na disketi) Damir Brkić, V. Gaćeša 6, 44260 Glin, tel. 044/88-221 od 8—12 i 16—18 h.

□ Komodor 64 — izbor najboljih programa u kompletima od 20 komada (1000 din. sa kasetom) ili pojedinačno po 100 din. Besplatan Katalog, Panić Saša, Komudraška nova 13, 11000 Beograd, tel. 011/461-778

□ KOMMODORE 64 — PROFESIONALNI PREVODI: REFERENCE GUIDE 1700 d. MAŠINSKI JEZIK ZA POČETNIKE 1400 d. PRIRUČNIK OD C-64 1300 d. KAKO DA PROGRAMIRATE C-64 900 d. DISK SISTEMI I ŠTAMPACI 900 d. GRAFIKA I ZVUK 900 d. MATEMATIKA NA C-64 1000 d. SIMONČI BASIC 700 d. C-64 BASIC 700 d. PRACTCALC 800 d. EASY SCRIPT 400 d. VIZAWRITE 400 d. PASCAL 400 d. MULTIDATA 400 d. HELP 400 d. GRAF 400 d. TRTICA GORAN, STEVANA LUKOVIĆA 9, 11090 BEOGRAD, TEL. 011/563-348

□ Prodajem 50 najboljih igara za KOMMODORE 64 za 1000 dinara, plus kasetna 400 dinara. Kvalitet i usluga zagarantovani. Isporuka odmah. Stojić Slobodan, 7 Jul 31 b/2, 18000 Niš, tel. 018/51-805

□ KOMMODORE 64 — najpopularnije i najbolje programe za Vaš C-64 prodajemo. Preko 300 naslova. Besplatan spisak. DENI — OZREN ĐUKIĆ, 41020 Zagreb, Čalovićevića 5/III, tel. 041/688-004

□ KOMMODORE 64 — Paket igara: Beach Head 2, Friday 13, F-15 Strike Eagle, Exploding Fish, Rambo 2, Winter Games, 911 Tiger Shark, Elite, Staff of Karnath 1, Staff of Karnath 2, Commando prodajem. Sa kasetom i poštarninom 1500 din. Husetović Elmira, A. Herljevića 37, 75000 Tuzla, tel. 075/216-044

□ Za KOMMODORE 64 po ceni od 1500 din. prodajem PIZZA DISK III koja kopira 225 blokova. Ovim programom možete kopirati sa kasete na disketu: DIGITAL DRUMS, BLAGGER GOES T.H. P. ROLAND'S RAT RACE I st. KRESOJA DRAGISA, VOJVUDE STEPE 421/B, 11000 BEOGRAD, tel. 011/475-419

□ Za KOMMODORE 64 prodajem: Komplet 1: SUMMER GAMES, RAMBO II, MC GULLIAN BOXING, Komplet 2: AMAZON WARRIOR, BLAGGER GOES T.H. DIGITAL DRUMS, HACKER, KUNG FU MASTER, MACBETH, Komplet 3: CHIMERA, COMMANDO, CYLU, SKOOL DAZE, SUPER HUEYE (4 helikopterske igre), Komplet 4: EMERALD ISLE, ENTOMBED (STAF 2), FRANK BRUNO'S BOXING, SENTINEL, SPITFIRE 40, VALHALLA itd. Cena jednog kompleta je 600 din. KRESOJA DRAGISA, VOJVUDE STEPE 421/B, 11000 BEOGRAD, tel. 011/475-419

□ KOMMODORE 64 — PYUJAMARAMA 1,2 i 3, ELITE, ACTION BIKER, STOP EXPRESS, JUMP JET, BOULDER RASH II, KNOCKOUT II i mnogi drugi po najnižim cenama. Besplatan katalog. Stjepčević Stredrag, Bulevar Revolucije 290/V 11000 Beograd, tel. 011/417-064

KOMMODORE 64 — neverovatno izbor kasetnih programa, 20 najboljih u prošlom broju Nove igre: Nodes of Yesod, Radar Moon, Fighter Pilot, Staff of Karnath II i III, Speed King, Specijalna ponuda — najbolji COPY programi: COPY 190, T. COPY — 190 blokova, PIZZA TURBO — 200 bl., PIZZA II — 202 bl., ABC TURBO — 202 bl., TURBO 2002 — 195 bl. (autor M. Zaccari) + kasetna i poštarnina = 1500 din. Uputstva za početnike, veliki popisati, besplatan katalog, Željčević Kemal, Braće Kosić 13, 72220 Zavidovići, tel. 072/874-441

□ Za C-64 hitovi u kompletima po 15 + kasetna — 1500 din. Imamo Elite, Hacker, Flight Simulator 2, School Daze, Night Shade... G-SOFT, B. Valjina 26/A, 57000 Zadar, tel. 057/32-113

Prodajem programe za Commodore 64. Najefitniji paketi na zemlji. 380 programa za samo 4900 din. Rok isporuke dva dana na besplatnim compact kasetama. Tel. 072/39-846

□ Commodore 64, Indiana Jones, Hyper sports, Pitstop III, Rambo, Petak 13, Forest of Doom, Boulderdash III, Artzet Tomb, F 15, Hacker, Basketball III, Meris of Fun, Kasetna, POKORNAJEROVSKI DODATAK, Komplet 2, Summer Games II, Winter Games, Beach Head II, Kasetna POKLON SEME MIKROPROCESSORA. Komplet košta 1000 din., oba 1800 din. Sakać Rajko, Novi Svr 86, 21466 Kucura, tel. broj druga 021/725-060 Željko.

□ PRODAJEM OKO 1400 NAJPROGRAMA ZA „KOMODOR 64“ — NA KASETAMA, CENA 22 DINARA, KOMPLET 30.000 DINARA, EKSPRES USLUGA, BESPLATAN KATALOG, N. KALABA, Slavke Đurđević-BŠ, 35000 Svetozarevo, tel. 035/27-119

o **KOMODORE 64** — Veliki izbor kasethnih programa (20 najboljih broj 11). Novi su: Basketball 3, Boulderdash 3, Winter Games, Walkiria 17, Witch's Coudron, Supermen, Commando, Lords of Midnight, Cyllu, Kliff Hanger, Paranoid (ABC), Bounty Bob 201, Besplatan katalog, saveti za početnike, P. S. Tomopje (preko 2 kaseta) i nove dr. Zujuevic Kemal, Braće Kosorić 13, 72220 Zavidovići, tel. 072/874-441

o **KOMODOROVCI TRAJE LI:** Friday 13, Indiana Jones, FIGHT, Hyper Sports, Commando, Supermen, Staff 1—3, Impossible 2, Winter Games...? Javite se Locomotion Software! Tražite katalog i uverite se! Perić Emil, Rastocić SŠ, 51000 Rijeka, tel. 051/511-032

o **C-64 komplet: D.T.'s SUPER-TEST, ZORO, BOLDERDASH 3, DYNAMITE DAN, GHOSTBUSTERS 2, PITSTOP 3, RAMBO, FRIDAY 13, NODES OF YESOD.** Programi + kasete = 100 dinara, pouzede. Isporuka odmah. Arto Ibrahimagić, Cetinjska 12, 71000 Sarajevo, tel. 071/456-329

o **KOMODORE SUPER SOFT** — hitovi na kazeti (RAMBO 2, COMMANDO, KUNG FU MASTER, SKOOL DAZE, KOKO, BOULDER DASH 3...), Domjanović, Albinjeva 2, 41020 Zagreb, tel. 041/687-081

o **KOMODORE 64** Komplet sa kasetom 1700 dinara. Komplet 1: Boulder Dash III, America Road Race Koko Baseball III My Chess II 2D Wargames II Jungle Quest Nick Faido Golf Transformers Ghostchaser Trolls Pitstop III (ili zamena) Komplet 2: Kung Fu Master, Commando, Willow Pattern, Skool Daze, Superman, Zorro, Alien 8, Koko, Rambo, II, Who Dares Wins II, Wizards Lair, Cyllu, Čurčić Dušan, Osmana Dikića 15a, 11000 Beograd, tel. 011/782-022

o **uzetna prilika** — komplet od pet najboljih COPY programa (PIZ 2a 202, TT — COPY, TURBO 200, ABC TURBO, TURBO 199) + uputstva = 1000 din. Domjanović, Albinjeva 2, 41020 Zagreb, tel. 041/687-081

o **JOKER SOFTWARE** vam predstavlja samo najnovije i najbolje programe za vaš C-64. Ponudite besplatan katalog na telefon 021/398-245, Jočić Dragan, Put Bačkapalanskog odreda 20/2, 21000 Novi Sad.

o **KOMODOROVCI** Najnoviji programi na kaseti u najnižim cenama. Tražite besplatan katalog. Stanić Branko, II kule 128, ulaz 11070 Novi Beograd, tel. 130-684 ili 152-083

o **KOMODOROVCI**, već tri godine na tržištu dokazuju svoju profesionalnost, stručnost i solidnost. Veliki izbor programa vrstanih u komplete isto sadrže 20 do 60 programa, a stoji 1000 do 1500 dinara plus kasete. Upute za početnike, Isporuka odmah. Tražite katalog (sa napomenom — za CBM 64 i 50 d.), skratice čekanje naruđbom preko telefona 041/417-052. FUTURE ORION, 41000 ZAGREB, Rubeticeva 7. Za veće naruđbe veliki popusti!

o Komodor 64: RAMBO 2, SKOOL DAZE, SUMMER GAMES 2, COMMANDO, KOKOTINI WILF, BOULDER DASH 3, i još mnogo novih programa po niskim cenama. Adrović Miroslav, Milutina Uskokićeva 33, 24000 Subotica, tel. 024/29-760

o Najnoviji svetski hit-programiza **KOMODORE 64** u kompletu: — FIGHTING WARRIOR (Ex. Ili)

— KARATEKA (3x Karate)
— BOULDER DASH III
— CHUCKIE EGG II
— PITSTOP III
— FRIDAY 13th (Petak 13.)
— MASTER KUNG FU
— ZORRO
— BLADE RUNNER (popularni film)

o **SUPERMAN** Komplet navedenih programa sa kasetom stoji 1200 din. Rok isporuke 48 časova. Plaćanje pouzede. Čajković Karlo, Anke Matić 3/3 11210 Beograd, tel. 011/711-358

o Za Komodor 64 kasetofon program. Cena po dogovoru. Bata, tel. 021/21-758

o **KOMODORE 64** — Paket A: RAMBO II, KUNG FU MASTER, COMMANDO, SKOOL DAZE, PYJAMARAMA III, FRENKIE... INDIANA JONES, FRIDAY 13, Paket B: WINTER GAMES, FLIGHT SIMULATOR II, GIRLS FOR FUN, BEACH HEAD II, SUPERMAN, STAFF III. Cijena paketa + kasete + poštarina 1000 din. Alan Đurić, Edvarda Kardaševića 20, 50000 Dubrovnik, tel. 050/22-807

o Komodor 64 — prodajem programe (WHO DARES WINS 2, NODES OF YESOD, SKOOL DAZE, RAMBO 2, COMMANDO, WINTER GAMES, HYPER SPORTS) pojedinačno ili u kompletima. Bulatović Vukasin, Berdapska 10, 11000 Beograd, tel. 011/406-836

o **Commodore VOLVO SOFT** — prodajem i menjam programe snimane direktno iz računara na kasetofon na koje nije menjan azimut. Svakom naruđbici tri programa besplatno. Mitojević Ivan, 38218 Leposavić

o **C-64** — TESLA SOFT — 30 programa + kasete + poštarina 1000 din. Nikolić Nebojša, Đure Dakovića 2/1, 34000 Kragujevac, tel. 034/66-491

o **Prodajem programe za C-64**, Zovite za besplatan katalog. ELITE, SWS 2 i druge igre čekaju na vas kod nas. Bakić Vladan, Polimska P+7, 84300 Ivandjuro, tel. 084/61-277

o Komodor — razdelnik „DVA-TASET“: Prikućite dva datase-ta na C-64. Presnimava sve programe — sa ili bez zaštite. Preklopnik za dva režima rada. Uputstvo, garantovan kvalitet. Kesler, Viktor, Rumenačka 106/1, 21000 Novi Sad, tel. 021/334-717

o **C-64 ALL STAR SOFTWARE** — Komplet najnovijih igara: WINTER GAMES (6 igara), COMMANDO, RAMBO II, SUPERMAN, KUNG FU MASTER, WIEV TO A KILL, BOULDER DASH II, BRIAN BLOODAXE, EXPLODING FIST, SKOOL DAZE, ALIEN 8+ kasete + poštarina = 1200 din. Stojnić Boris, Bratsva jedinstva 10, 75000 Tuzla, tel. 075/213-964

o **KOMODOR 64** — besplatan katalog sa 1200 kasethnih programa: HYPER SPORTS, RAMBO 2, KUNG FU FRIDAY 3, WORLD CUP 2, SQUASH, GHOSTBUSTERS 2, HACKER i ZORRO. Niske cene, popust. Milan Stamenović, Vladimira Nazora 77, 18000 Piroć

o **Prevedene igre za C-64**, Četvrti paket — hitovi: NODES OF YESOD, SKOOL DAZE, SORCERY, GHETO BLASTER, CIPHER 0, CHOPPER, ELIDON. Svi tekstovi na ekranu su na srpskohrvatskom jeziku. Uz paket dostavljamo uputstvo za apsolutne početnike i katalog, Cijena 1500 din. Predrag Cvetković, 11000 Beograd, Radmile Rajković 12/28, tel. 011/768-741

GALAKSIJA

o **Povoljno prodajem integralna** kola 6264 (8K RAM), interfejs 1 za GALAKSIJU kao i čipove 74 LS... sa, sve podnožja (profesionalna). Dimitrije Stojić, M. Piljate 22, 25250 Odžaci, tel. 025/74-448

o **HARDWARE** za prepravku galaksije u galaksiju plus, veliki izbor mikroprocesora, logičkih i linearnih kila. Posedujem preko 1000 šema iz najnovijih stranih časopisa. Naznačite koja vas oblast interesuje. Jelčić Dragan, Jansenička 37, 11420 Smed. Palanka

o **GALAKTIČARI** Ukoliko želite da se probate u najboljoj igri za svoju mašinu, kupite igru inspektora SPIDRINI 1500 d. Komplet: PROTON I i II, SUPERPROTON I i II, SIREN COPY TAPE, SPEEDMASTER, V 2.0 i TAPE MECHANIC COPY, Silivlje Lukić, Dimitrovićeva 6, 41000 Zagreb, tel. 041/510-296

o **PRADJEM GALAKSIJU 8/6K**, ispravljajući original kutiju, ostalo... 30.000 din. CASIO FX750P nov, plus dodatak FA-20 sa printerom i priključcima. Ima konstantnu memoriju. 70.000 din. Štefić Vaso, Celovska 87 61000 Ljubljana

o **Prodajem računar „galaksija“ 8/6K**, Subotić Radovan, Omladinskih brigada 182/32, 11070 N. Beograd, tel. 151-277

QL

o **QL** — Prodajem programe, uputstva, literaturu. Sve na jednom mestu (najprejevite). Veliković Kiro, Nikola Trnjarec br. 4/8, 91000 Skoplje

AMSTRAD

o **AMSTRAD** — prodajem programe: Assembler/Diasesembler, HiSoft Pascal, Transformer 800 din, Rolland, Ghostbusters 300 din, COPY programi: Proton I i II, Speedmaster 600 din, Katalog, Tomaš Kastelec, Česta na Vrhovce 7, 61111 Ljubljana, tel. 061/267-632

o **AMSTRAD** — PROFESIONALNI PREVODI. PRIRUČNIK ZA CPC-464 1300 d. MAŠINSKI JEZIK ZA POETNIKE 1300 d. LOCOMOTIVA BASIC 1200 d. DEVPAC 700 d. MASTERFILE 700 d. PASCAL 700 d. TAS-WORD 400 d. TRICA ALEKSA STEVANA LUKOVICA 9, 11000 BEOGRAD, TEL. 011/563-348

o **Prodajem Šnajder 464**, zeleni monitor. Tel. 023/522-518

o **Amstrad** — dve kasete + poštarina + programi (HYPER SPORTS, DEVIL'S CROWN, ARABIAN NIGHTS, FIGHTING WARRIOR, RAID OVER MOSCOW, B.J. SUPERSTAR, CHALLENGE, 3D SLAPSHOT, HARD HAT MACK, 3D GRAND PRIX) = 5000 din. Jovanović Dragan, Dragoslava Jovanovića 12, 37000 Kruševac, tel. 037/30-568

o **COMPUTER BOX SOFTWARE** vam nud komlet COPY programa sa uputstvima i sa kasetom za samo 3000 n.d. odnosno bez kazete 1500 d. Komplet: PROTON I i II, SUPERPROTON I i II, SIREN COPY TAPE, SPEEDMASTER, V 2.0 i TAPE MECHANIC COPY, Silivlje Lukić, Dimitrovićeva 6, 41000 Zagreb, tel. 041/510-296

ATARI

o **ATARI-520ST** programe i literaturu razmjenjujem. Dajem besplatne stručne savjete za programiranje i korišćenje svih tipova ATARI računala. Zainteresiranim mikro-klubovima poklanjam stare strane stručne mikro-časopise. Zvonimir MAKOVEC, tel. 062-714-115 (do 14,30h)

o **ATARI SOFT-CLUB** Zrenjanin. Prvi u YU. Veliki izbor programa svih vrsta i literature. Pozivamo vas na razmenu i na saradnju. Besplatan katalog. Lacmanović Dejan, Sindiceva 31/A, 23000 Zrenjanin, tel. 023/31-665

BBC/ELEKTRO

o **Kupujem programe i literaturu za ACORN ELECTRON**. Ponudite slihu na telefon 011/328-957

BBC/ELECTRON

□ Poklonici kompjutera!

Kompjuteri Acorn BBC

prvi put u Jugoslaviji uvoznik „Partizan“ — Čačak tel. 032/51-710, 41-366 teleks 13745 13814

□ BBC/Elektron — gotovi programi — inspiracija za samostalne programe. Izbor od 350 programa. Mogućnost izmene. Usluge štampanja. Prevodi uputstva. Besplatan katalog. Mitutin Marković, Sindeličeva AC/20 37000 Kruševac

HARDVER

□ GALAKSIJA — SPECTRUM HARDWARE
Prodajni komplet čipova za proširenje Spektruma sa 16K na 48K (ili 80K) (9900 din.), te čipove: 2764 (2200 din.), 27128 (3000 din.), 4164 (1100 din.), 6116 (2200 din.), 4416 (4200 din.), Z80A — CPU (2200 din.), Z80A — BIO (2500 din.), 6502 (4500 din.), 6522 (4500 din.), 8255 (2200 din.), AY3-8910 (5500 din.), AY3-8912 (5500 din.), TEXT02L 28 (5500 din.), te ostala TTL i CMOS kola za gradnju „galaksije+“ i druge gradnje.
Servisiram brzo kvalitetno i računarske Spektre i C-64.
N. Četković, J. Leskovaara 1, 42000 Vraždin, tel. 042/38-56

□ BLAST! — ubrzajte vaše bazične programe i do 40 puta. Traka sa BLAST COMPILER-om i BLAST TOOLKIT-om i opširno uputstvo na 35 strana — svega 1500 din.
UZ MACHINE CODE FOR BEGINNERS uživajte i sami u efikasnosti i brzini mašinskog programiranja. Traka i opširno uputstvo na 65 strana — svega 1600 din.
Oba programa — 2500 din.
VRCA MILAN, Zarija Vujševića 79, 11070 Novi Beograd

□ SPECTRUM SOFTWARE STUDIO — izbor od preko 1000 programa — svaki program sa uputstvom — veliki izbor literature — knjiga i originalnih programskih uputstava na engleskom i srpskohrvatskom jeziku. Spisak programa je besplatan, za katalog sa opisom pošliti 200 din. Brza i kvalitetna usluga — proverite.
PAJNIC MIRKO, STRAHINIČA BANA 56, 11000 BEOGRAD, tel. 011/188-190 poste 15h.

□ HURRICANE SOFTWARE — veliki izbor najboljih i najjeftinijih programa, kvalitetni animacijski, isporuka u roku od 24h. Besplatan katalog naručite na adresu Milosavljević Saša, Titogradska 43, 36000 Kraljevo, tel. 036/332-580

LITERATURA

□ AMSTRAD: Profesionalni prevodi: PRIRUČNIK (1100), LOCOMOTIV BASIC (1200), MAŠINSKO PROGRAMIRANJE (1300), ZAJEDNO (3400). Prevedena uputstva za uslužne programe: DEVPAC, TASMWORD, PASCAL, QUILL, MASTERFILE, HOME BUDGET Pojedinačno (700), KOMPLET (3500). Snimljeni program (200). AMSTRAD FUTURE, Bate Jankovića 79, 32000 Čačak, telefon 032/30-34

□ QL: PROFESIONALNI PREVODI: QL-ARCHIVE (1500), QL-TOLKIT (1200), QL-PASCAL (1500), QL-FORTH (1500), U KOMPLETU (5000), ZARIĆ SLOBODAN, Bate Jankovića 79, 32000 Čačak, tel. 032/30-34

□ KOMODOR: Profesionalni prevodi: PRIRUČNIK (650), KAKO DA PROGRAMIRATE C-64 (1000), PROGRAMERS REFERENCE GUIDE (1200), MAŠINSKO PROGRAMIRANJE ZA POČETNIKE NA C-64 — Prevod 1985. g. (1300), GRAFIKA I ZVUK (900), MATEMATIKA (1000), DISK SISTEMI I ŠTAMPARIJE (900), UPUTSTVO ZA 1541 (700), SIMONS BASIC (650), PRAKTIČNAK (750), EASY SCRIPT (400), MAE (400), VIZAWRITE (800), PASCAL (400). Svih 19 knjiga po povlašćenju ceni (10.000). NOVO NA TRŽIŠTU: Prevod najbolje knjige „MAPPING THE C-64“ (2500). Na oko 200 stranica su objašnjene sve memorijske lokacije. Isporuka u martu: „KOMPIJUTER BIBLIOTEKA“, Filipa Filipovića 41, 32000 Čačak, tel. 032/31-20

□ SPECTRUM: Profesionalni prevodi: MAŠINAC ZA POČETNIKE (1000), DISAPSEMBLIRANI ROM (1200), NAPREDNI MAŠINAC (1300) i nova knjiga SET INSTRUKCIJA ZA Z80 (2500). SVE ČETIRI (5400). Prava uputstva: DEVPAC, MEGA BASIC, BETA BASIC, ARTIST, MELBOURNE DRAW, EDITOR ASEMBLER, MONITOR DISAPSEMBLER, QUILL, TASMWORD, FITF. Pojedinačno (500). Komplet (3500). Svih 10 programa snimljenih na kasetu (2000). „KOMPIJUTER BIBLIOTEKA“, FILIPA FILIPOVIĆA 41, 32000 Čačak, tel. 032/31-20

RAZNO

Kupujem prvih pet brojeva časopisa „Moj Mikro“, Tel. 603-554

„PISTE U NOĆI“

KNJIGA KOJU PILOT NAMENJUJE PILOTIMA



Najzad je izašla iz štampe knjiga koju očekujete još od prošlog leta! To je novi „pilotski bukvar“ Zorana Modlija, „Piste u noći“. Knjiga je u međuvremenu promenila izdavača, što je bio jedan od značajnijih razloga ovog zakašnjenja. Autor vam se zahvaljuje na strpljenju, a za uzvrat je dopunio knjigu nizom novih i uzbudljivih detalja.

„Piste u noći“, anegdotski i lako razumljivo, opisuju tehniku instrumentalnog letenja i vođenja aviona u savremenoj vazdušnoj saobraćaji, elektronsku opremu na zemlji i u pilotskoj kabini koja to omogućava, simulacije letenja na kućnim računarima — ali i udese nastale kao posledica nepridržavanja propisanog „bon-tona“ i pilotiranja u hazardnim meteorološkim uslovima.

Zahvaljujući tome, moto knjige je: KAKO OSTATI PILOTI! Stručni konsultanti i recenzenti knjige su saobraćajni piloti-kapetani JAT-a, kontrolori letenja i profesori Više vazduhoplovne škole.

Izuzetno zanimljivo i (ne samo za pilota) upotrebljivo štivo, rasuto je na 288 strana standardnog formata, sa isto toliko ilustracija, u broširanom povezu i koricama u punoj boji.

... Bogato medijsko iskustvo autora, sklonost lakom i razumljivom kazivanju, a uza sve to i njegovo profesionalno bavljenje letenjem, učinili su da dobijemo još jednu knjigu koja, poput nekadašnje „Krilate katedre“, sugestivno i nadahnućno mami za sobom novu armiju vazduhoplovnih zaljubljenika, pronoseći suštinski ideju vazduhoplovstva: da spaja obale okeana i pretvara ovaj naš globus u provinciju.“ (Recenzent Đorđe Jovanović, profesionalni pilot JAT)

NIRO „TEHNIČKA KNJIGA“ 7. jula 26 11000 Beograd

Ovim neopozivo poručujem _____ primeraka knjige „Piste u noći“ u izdanju „Tehničke knjige“ iz Beograda, po ceni od 1600. dinara. Platiću prilikom prijema pošiljke — POUZETEM.

Ime i prezime _____

Adresa _____

Potpis _____

Datum _____

Testovi za „kreja“

Pod pojmom performanse računara većina ljudi podrazumeva broj operacija koje računar može da uradi za određeno vreme. Na žalost, za razliku od stalih performansi koje su nesumnjive (na primer, količina ROM-a i RAM-a), podaci o brzini se znatno razlikuju od jednog prikaza do drugog. Razlog je u tome što računari pri različitim zadacima pokazuje različite brzine, u zavisnosti od toga koji je kritičan resurs računara prouzrokovao izvestan zastoj u odnosu na maksimalne mogućnosti. Ne retko, proizvođači računara na uporednim demonstracijama sa konkurentskim mašinama testiraju brzinu zadacima koji se izuzetno brzo izvršavaju na njihovom računaru, a na konkurentskom sporo.

Tabela 1. Broj operacija po petlji

PETLJA	broj operacija
1	7217
2	8448
3	14815
4	9783
5	8347
6	9359
7	4573
8	4831
9	4918
10	4412
11	12882
12	11999
13	8846
14	9915

Stvarnu brzinu računara nije lako predstaviti kao jedan broj ili kao skup brojeva. Međutim, želja da se uporede različiti računari i da se ustanovi koji od njih je brži je toliko jaka da se veliki broj izuzetno kompetentnih institucija u svetu daju testove za merenje performansi sistema. Brzinski testovi su, uglavnom, na fortranu jer je brzina najbitnija kod numeričkih zadataka. Najcenjeniji brzinski test je test Livermorske laboratorije kojim možete da izmerite i brzinu vašeg računara. Ako nemate fortran kompajler, jednostavno transpirajte programe i pustite programe, izmerite vreme i izračunajte brzinu po trivijalnoj formuli:

Brzina (u topovimaj)=broj operacija/vreme (u sekundama)

Broj operacija za pojedini zadatak (ukupno ima 14 zadataka i svaki, u stvari, predstavlja jednu petlju, što je učinjeno da bi računar pokazao svoju stacionarnu brzinu) dat je u tabeli 1, a petlje u tabeli 2.

Da biste mogli da se uporedite sa jačim mašinama, pogledajte tabelu 3. Ako vam se učini da vam računar nije dovoljno brz,

Tabela 2. Petlje Livermor laboratorije za merenje performansi računara

No. 1:	$10 \text{ } i = 1, 1000$ $1 \text{ } x(i) = x(i) + y(i) \text{ } i = 1, 20000 \text{ } i = 1, 20000$
No. 2:	$10 \text{ } i = 1, 10000, 5$ $2 \text{ } k = x(i) + y(i) \text{ } k = x(i) \text{ } i = 20000 \text{ } k = x(i) + y(i)$ $4 \text{ } k = x(i) + y(i) \text{ } k = x(i) \text{ } i = 20000 \text{ } k = x(i) + y(i)$ $6 \text{ } k = x(i) + y(i) \text{ } k = x(i)$
No. 3:	$10 \text{ } i = 1, 10000$ $3 \text{ } i = x(i) + y(i)$
No. 4:	$10 \text{ } i = 1, 10000, 5$ $10 \text{ } i = x(i) + y(i) \text{ } i = x(i) + y(i)$ $4 \text{ } i = x(i) + y(i)$
No. 5:	$10 \text{ } i = 1, 10000, 5$ $i(i) = x(i) + y(i) \text{ } i(i) = x(i) + y(i)$ $3 \text{ } i(i) = x(i) + y(i) \text{ } i(i) = x(i) + y(i)$
No. 6:	$10 \text{ } i = 1, 10000, 5$ $i(i) = x(i) + y(i) \text{ } i(i) = x(i) + y(i)$ $4 \text{ } i(i) = x(i) + y(i) \text{ } i(i) = x(i) + y(i)$
No. 7:	$10 \text{ } i = 1, 1000$ $x(i) = x(i) + y(i) \text{ } i = x(i) + y(i)$ $4 \text{ } i = x(i) + y(i) \text{ } i = x(i) + y(i) \text{ } i = x(i) + y(i)$ $6 \text{ } i = x(i) + y(i) \text{ } i = x(i) + y(i) \text{ } i = x(i) + y(i)$
No. 8:	$10 \text{ } i = 1, 1000$ $x(i) = x(i) + y(i) \text{ } i = x(i) + y(i)$ $4 \text{ } i = x(i) + y(i) \text{ } i = x(i) + y(i) \text{ } i = x(i) + y(i)$ $6 \text{ } i = x(i) + y(i) \text{ } i = x(i) + y(i) \text{ } i = x(i) + y(i)$
No. 9:	$10 \text{ } i = 1, 1000$ $x(i) = x(i) + y(i) \text{ } i = x(i) + y(i)$ $4 \text{ } i = x(i) + y(i) \text{ } i = x(i) + y(i) \text{ } i = x(i) + y(i)$ $6 \text{ } i = x(i) + y(i) \text{ } i = x(i) + y(i) \text{ } i = x(i) + y(i)$
No. 10:	$10 \text{ } i = 1, 1000$ $x(i) = x(i) + y(i) \text{ } i = x(i) + y(i)$ $4 \text{ } i = x(i) + y(i) \text{ } i = x(i) + y(i) \text{ } i = x(i) + y(i)$ $6 \text{ } i = x(i) + y(i) \text{ } i = x(i) + y(i) \text{ } i = x(i) + y(i)$

	$cr = br - ps(i) \text{ } i$ $ps(i) = br$ $ar = cr - ps(7) \text{ } i$ $ps(7) = cr$ $br = ar - ps(8) \text{ } i$ $ps(8) = ar$ $cr = br - ps(9) \text{ } i$ $ps(9) = br$ $ar = cr - ps(10) \text{ } i$ $ps(10) = ar$ $br = ar - ps(11) \text{ } i$ $ps(11) = ar$ $cr = br - ps(12) \text{ } i$ $ps(12) = br$ $ps(14) = cr - ps(13) \text{ } i$ $ps(13) = cr$
10	CONTINUE
No. 11:	$x(i) = y(i)$ $10 \text{ } i = 1, 2, 10000$
11	$x(i) = x(i+1) - y(i)$
No. 12:	$10 \text{ } i = 1, 1, 999$
12	$x(i) = y(i+1) - y(i)$
No. 13:	$10 \text{ } i = 1, 1, 120$ $i(i) = p(i) \text{ } i$ $p(i) = p(i) \text{ } i$ $p(3) = p(3) + b(i) \text{ } i$ $p(4) = p(4) + c(i) \text{ } i$ $p(5) = p(5) + p(2) \text{ } i$ $p(2) = p(2) + p(4) \text{ } i$ $i(2) = p(i) \text{ } i$ $i(2) = p(2) \text{ } i$ $p(i) = p(i) + p(i) + y(i) \text{ } i$ $p(2) = p(2) + z(i) \text{ } i$ $i(2) = i(2) + x(i) \text{ } i$ $i(2) = i(2) + y(i) \text{ } i$ $b(i) = b(i) + b(i) \text{ } i$
13	CONTINUE
No. 14:	$10 \text{ } i = 1, 1, 150$ $i(i) = g(i) \text{ } i$ $g(i) = g(i) \text{ } i$ $v(i) = v(i) + w(i) \text{ } i$ $w(i) = w(i) + v(i) \text{ } i$ $v(i) = v(i) + w(i) \text{ } i$ $w(i) = w(i) + v(i) \text{ } i$ $v(i) = v(i) + w(i) \text{ } i$ $w(i) = w(i) + v(i) \text{ } i$ $v(i) = v(i) + w(i) \text{ } i$ $w(i) = w(i) + v(i) \text{ } i$
14	CONTINUE

Znak # označava nastavak linije i nalazi se u koloni 5.

Tabela 3. Performanse superračunara testiranih petljama Livermor laboratorije (u megaflopovima)

petlja br.	FUJITSU		CFT	CRAY 1		CRAY X-MP (UNIPROCESSOR)	
	HITACHI S810/20	VP100 VP200		CIVIC COMPILER	CIVIC COMPILER	CFT 110	CFT 112 +
1	259	187	331	68	89	146	155
2	255-306	105	189	16	15	28	138
3	213-322	175	338	29	56	67	136
4	60-92	74	80	21	36	4	48
5	5-11	10	10	4	5	5	11
6	4-11	10	10	3	3	4	9
7	255	189	331	87	87	148	165
8	85	86	99	52	65	75	91
9	227	161	261	56	90	93	123
10	63	59	86	28	29	54	64
11	10	5	5	3	3	3	8
12	104	59	115	23	23	68	76
13	4	6	6	4	3	5	5
14	7	13	14	7	8	10	7

pogledajte članak o superračunarima u prošlom broju „Računari“ i izračunajte odnos brzina/cena za vas i neki od superračunara i raspodeljenje će vam se popraviti.

Ako ste i posle toga nezadovoljni, pišite nam i mi ćemo vas uputiti kako da nabavite

neke druge testove, na kojima će se vaš računar pokazati možda boljim. Ako već nemate brzi računar, imajte bar dobre rezultate na testovima.

Nada Aleksić

NOLIT T. F. Fraj RAČUNARI ZA POČETNIKE

T. F. Fraj

RAČUNARI ZA POČETNIKE



Cena 1.200 dinara

NOLIT

IRO „NOLIT“ OUR I

Beograd, Terazije 13/IV

NARUDŽBENICA Računari br. 12

Ovim neopozivo naručujem knjigu RAČUNARI ZA POČETNIKE po ceni od 1.080 dinara (sa popustom 10%)

Porudžbine telefonom: 011/338-150