

Specijalno izdanje časopisa „Galaksija“ Jul 1984. / Cena 200 D

# računari

u vašoj kući

škola  
akcionih igara

memorija  
za „galaksiju“  
48 K

QL • Electron • BBC B **2** Commodore 64





55

БЕОБАНКА  
*свуда са вама*



Бнкит

# računari

u vašoj kući

# 2

Specijalno izdanje  
časopisa „Galaksija“  
Jul 1984.  
Cena 200 D

## Sadržaj

4/ni golub na grani, ni vrabac na ruci

6/na zapadu ništa novo

8/u susret superinteligenciji

11/era tehnoloških genija

16/nemojte plakati za eplom

18/računari u izlogu: QL, Elektron, BBC B, Commodore 64

32/štampani u akciji

42/moj prvi program

47/biblioteka programa

55/„galaksija“ bez tajni

66/svi „spectrumovi“ restarti

68/softverska veza

72/škola akcionih igara

82/programator eproma

88/memorija za „galaksiju“

94/„ubiću se, neće da radi!“

96/konstruktorski hod po mukama

Na naslovnoj strani:  
CRTEŽ SA KORICA KNJIGE „THE JOY OF COMPUTERS“: KOJA ĆE POD NASLOVOM „KOMPIJUTOR U KUĆI“ IZÄCI OVE JESENI NA HRVATSKOSRPSKOM ODNOSNO SLOVENACKOM JEZIKU. U IZDANJU „CANKARJEVE ZALOŽBE“ (višeti oglas na str. 38)

### Izdaje

Beogradski izdavačko-grafički zavod  
OOUR Novinska delatnost „Duga“  
11000 Beograd  
Bulevar vojvode Mišića 17

### Telefoni

650-161 (redakcija)  
650-528 (prodaja)  
651-793 (propaganda)

### Generelni direktor

Gojko Zečar

### Direktor OOUR „Duga“

Zoran Milošević

### Glavni i odgovorni urednik

Gavrilo Vučković

### Urednik izdanja

Jova Regasek

### Likovna i grafička oprema

Dušan Mijatović

### Redakcije časopisa „Galaksije“

Tanasije Gavranović, pomoćnik glavnog i odgovornog urednika

Esad Jakupović, zamenik glavnog i odgovornog urednika

Aleksandar Milinković, urednik

Jova Regasek, urednik

Zorka Simović, sekretar redakcije

Gavrilo Vučković glavni i odgovorni urednik

### Autori tekstova

Dejan Ristanović

Miša Hadži-Dorđević

Ivan Gerendić

Nedeljko Parežanović

Milan Pavičević

Dorđina Perzić

Jova Regasek

Slobodan Vujović

Anđelko Zgorelec

### Prevodilci

Domagoj Božić

Zdenko Dizdar

Gavrilo Vučković

### Tehnička saradnja

Ljubiša Milovanović

Mira Todorović

### Specijalna saradnja

Anđelko Zgorelec, London

### Izdavački savet „Galaksije“

Dr Rudi Debijadi, prof. dr Branislav Dimitrijević

(predsednik), Radovan Draškić, Tanasije Gavranović,

Živorad Glišić, Esad Jakupović, Velizar Maslać,

Nikola Pajić, Željko Perunović, prof. dr Momčilo Ristić,

Vlada Ristić, dr inž. Milorad Teofilović,

Viđojko Veličković, Velimir Vesović,

Milivoje Vuković

### Štampa

Beogradski izdavačko-grafički zavod  
11000 Beograd, Bulevar vojvode Mišića 17

### Žiro-račun kod SDK

60802-603-17132

Devizni račun kod Beobanke

60811-620-6-82701-999-01066

Za inostranstvo cena dvostruka

(400 D, 3 US \$, 8 DM, 55 Sch, 7 Str, 24 Flr, 2 Lstg)

Na osnovu mišljenja Republičkog sekretarijata za kulturu broj 413-7772-03 i „Službenog glasnika“ broj 26/72.

ovo izdanje oslobođeno je poreza na promet

# ni golub, na grani ni vrabac u ruci

## Zašto kasnimo?

Kada smo pre nepunih godinu dana u redakciji počeli da razmišljamo o jednom posebnom izdanju koje bi u celini bilo posvećeno kućnim računarima, nismo ni slutili šta se sve nalazi na drugom kraju filijala koji smo time nameravali da potpalimo. „Računari u vašoj kući“, koji su se pojavili na tržištu prvih dana januara, planirani su u tiražu od 50.000 primeraka. A mala „galaksija“ izrasla je preko noći u naš nacionalni računar. I ne samo to. Računari već mesecima ne silaze sa stranica štampa i programa radija i televizije, a pored nekoliko računarskih naslova koji se već nalaze u izlozima knjižara, izdavači najavljuju još slijaset novih. Stencioni su pokrenuli već dva specijalna časopisa, a sasvim određene planove ima i redakcija „Galaksije“.

Računarski izazov produkcije je, barem na računima, i domaću elektronsku industriju, koja do pre nekoliko meseci, smatrajući za samo previše siromašni za to, o kućnim računarima nije želela čak ni da razmišlja. Uz „galaksiju“, „Iolu 8“ i varaždinskog „galeba“, originalne računare najavljuju i Elektronska industrija, Istra-Defta i Slovenske. Na domaća kompjuterska komešanja nema odziva jedino iz SIV-a, koji, praktično, kućne računare još uvek drži pod zabranom. To je razlog više da domaćoj industriji čvrsto stisnemo palčeve. Računarsku revoluciju, ustoično, možemo ostvariti samo ako budemo imali domaći računar.

I sama redakcija je, ruku na srce, pomalo izgorela na vatri koju je zapalila sopstvenim rukama. Akcija sa računarom „galaksija“ uvela je u redakciju vredno štampanje, koje traje već mesecima. Epromi, tastature, štamparska kola, čipovi... neprekidno odvlače našu pažnju sa svakodnevnih poslova. To je malo usporilo pripreme „Računara 2“. Na ovom specijalnom izdanju, osim toga, radi veoma skromna ekipa ljudi

— mala redakcija i mala grupa spoljnih saradnika. Zbog toga nismo bili u stanju da sa njim izademo pred čitaoca onih kada smo to želeli, ali smo redakcijske poslove okončali znatno ranije nego što je naša štamparija, zbog nedostatka papira, bila u stanju to da prihvati. To, naravno, ne znači da pred vas izlazimo sa bajajim tekstovima. Većina među njima ima trajnu vrednost, a oni kod kojih je aktualnost posebno važna, dopunjavani su do poslednjeg dana. Članak o računaru OL napisan je ponovo nakon što su u redakciju prispeli prvi realni prikazi neposredno pred ulazak izdanja u štampu.

Čitav taj neplanirani vremenski pomak ustovio je i izvesne promene u već najavljenoj koncepciji specijalnog izdanja. U obliku ideja i materijala i na ipak ograničenom prostoru pojedine nagoveštene rubrike i priloge morali smo da odložimo za sledeći broj „Računara“. Izostao je, između ostalog, i najavljeni umetak: taj prostor, sa standardnim „prelomom“, ustupili smo izvesnom broju tekstova koji će, verujemo, više odgovarati ukusu većine. Dodatni razlog za sve ove promene bio je i povećan priliv oglasa, za koje se nadamo da nikako nisu „nužno zlo“ nego specifične informacije čiji značaj nije za podcenjivanje.

Ako ovim izdanjem nismo u svetu zadovoljili svakuju ukus, možemo vas za malo strpljenje: pripreme „Računara 3“ već su uveliko u toku (nastojaćemo da iz štampa izadu krajem septembra) i u njima ćete, verujemo, pored ostalog naći i one priloge koji su po vašem mišljenju trebalo da našu svoje mesto u publikaciji koju upravo čitate.

Redakcija

Ovogodnji Sajam tehnike u Beogradu, koji je trajao od 22. do 26. maja, protekao je, više nego ijedan pre njega, u znaku kućnih i, naročito, malih poslovnih računara. U hali 1, na ogromnom izložbenom prostoru, moglo je da se pogleda i proba obilje domaćih i stranih mini i mikroračunara, kojih, na žalost, i dalje nema u prodavnicama. Sa posebnim uzbuđenjem smo očekivali da vidimo da li je (i kako) domaća elektronska industrija reagovala na „provokaciju“ zvanu računar „galaksija“. U nezvaničnim razgovorima sa zvanličnim licima na štandovima pokušali smo da saznamo nešto više o planovima domaćih proizvođača računara i modelima u koje se ulažu najveće nade.

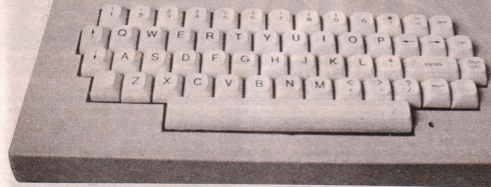
Jugoslovenski računari, nije bilo teško zaključiti, nisu u pravom smislu „kućni“. Dimenzije su im, doduše, najčešće takve da sasvim lepo mogu da se smeste na vaš radni sto, mogućnosti dovoljne za sve (pogotovo kućne), primene, ali sve to po ceni od koje vas odmah zaboli glava. Naši proizvođači su, dakle, najviše okrenuti firmama koje bi želele da se kompjuterizuju plaćajući ovaj tehnološki napredak u dinarima: maloprodaja interesuje slabo koga (za maloprodaju, naravno, trebaju i posebne dozvole) i to najpre zato što je u našim uslovima teško realizovati veliku i jeftinu seriju.

## Šta sprema Ei

Obilazeći halu 1 u smeru kretanja kazaljke na časovniku, naišli smo najpre na štand Tovarne Meril. Ona prodaje poznate Kopa terminale, koji su u poslednje vreme postali standard u mnogim računarskim centrima. Ništa za nas, reklo bi se. Ali, na štandu je izložen i domaći računar PMP, koga se ne bismo posideli ni na svetskom tržištu. Imali smo zadovoljstvo da porazgovaramo sa Marjanom Miletićem koji je, zajedno sa kolegom Brankom Jevtićem, konstruisao PMP, računar koji, kako ćemo videti, ima najmoćniji programsku podršku od svih kućnih modela. Ovaj kompjuter je, naime, zasnovan na mikroprocesoru TCT 11, koji predstavlja PDP 11 računar u malom. Operativni sistem je takav da svi programi pisani za PDP računare bez ikakvih problema rade i na PMP-ju. A poznato je da PDP računara ima proman broj, da se koriste bezmalo u svim računarskim centrima u svetu kao centralne ili periferijske jedinice i da, kako reče Marijan Miletić, on lično poklanja jedan primerak svoga računara onome ko navede jedan programski jezik koji je nješt prilagođen ROM računara PMP ima svega 2 KB i u njega je smešten samo kratki program koji učitava u RAM operativni sistem sa jedne od disketa. RAM u osnovnoj konfiguraciji ima 64 kilobajta, ali se, obzirom da je TCT 11 sesnaestobitni procesor, može praktično neograničeno proširiti.

Kako stoji stvar sa cenom? Centralna jedinica računara košta 150 hiljada dinara, a svaka disk-jedinica po što. Stičeće se sa nama da ova cena, kada se uzme u obzir, nije ni što se dobija, nije velika, i da ni strani računari, uz carine, ne bi koštali manje u odgovarajućoj konfiguraciji. Na žalost, postoji jedno ali: PMP ne može da se koristi bez terminala a terminal, zavisno od mogućnosti, može da košta 30—70 starih miliona! Zavisno od jedino tvrdnja Tovarne Meril da rade na integralnom video delu za televizor i posebnog tastaturu, koji će, ukoliko ispitivanje tržišta koje je u toku da povoljne rezultate, učiniti PMP pristupačnim širim krugovima korisnika.

Računar „galaksija“: Iako je prva serija od 300 komada privedena kraju, komercijalna varijanta našeg računara nije bila izložena na sajmu



Ne obratitvi, možda i neopravdano, dovoljno pažnje Digitronomovom štandu, stigli smo do sledećeg izlagača — El-Honeywella. Naša turba nije bila neobična — „Politika“ je nedavno donela vest da će El uskoro izbaciti na tržište prvi (svi koristimo taj epitet!) kućni računar pristupaćim svakome. Čim smo videli da na štandu nema gužve, znali smo da na njemu nema ni tog računara. Tačno tako — El je izložio već dobro poznati Honeywell 6/10. Ovaj računar je zasnovan na Honeywell-ovom mikroprocesoru LSI 6 koji se, radi kompatibilnosti, može dopuniti čuvenim Intelovim 8086. Korisnička memorija od 128 kilobajta može da se proširuje do 512 Kb, dok su na asinhroni komunikacioni port priključene dve disk-jedinice po 650 Kb. Oni koji žele da obraduju veće količine informacija mogu da dodaju Winchester masivne diskove po 20 Mb. Štampač se naknadno dokupljuje i može biti 132-kolonski matricni (100 karaktera u sekundi) ili sa lepezom — 30 karaktera u sekundi. Za cenu ovakve konfiguracije se nismo raspitivali — smatrali smo da vas neće interesovati.

## Od „galeba“ do „orla“

Na sledećem štandu smo ugledali dobro poznati računar (Apple II) i dobro poznatog kolegu (Julija Mekaneca mlađeg). Došli smo, jasno, do štanda Velebita. Citaoćima „Galaksije“ je, svakako, poznato da su Vražedinci pre tačno godinu dana („Galaksija“ broj 135) izbacili na tržište seriju od 40 računara „galeb“. Ovaj računar, zbog preudog razvoja i unekoliko zastarele koncepcije koju je dopunila veoma visoka cena 90.000 dinara, nije zaživio na tržištu i, po svemu sudeći, pripada prošlosti. No, „galeb“ov naslednik koji se zove „orao“ treba uskoro da uzet i zauzme mesto na našim stolovima. „Orao“ je logičan nastavak „galeba“ i delo je istog konstruktora — Mire Kocijana. Zasnovan je na mikroprocesoru 6502 i poseduje ROM od 16 i RAM od 6 Kb koji može dalje da proširuje. (Sve ove informacije su dobijene u nezvaničnom razgovoru, što znači da neke od njih mogu i da se promene). Kao periferijske jedinice koriste se standardni crno-beli televizor na kome računar kontrolishe finu grafiku sa 256x128 tačaka i kasetofon, ali postoji i RS 232 port za dalja proširenja, a u računar može da se ugradi i disk-interfejs. Bežik bi, obzirom na pristojan kapacitet ROM-a, trebao da bude vrlo dobar i dopunjen grafičkim rutinama (crtanje krugova itd.) ali, kako izgleda, ne i asemblerom koji bi omogućio jednostavno pisanje mašinskih programa.

„Orao“ će, izgleda, koštati koliko i „galeb“ — oko 9 starih miliona. Može li, kao takav, da konkuriše Spectrumu koji se prodaje preko malih oglasa i, za duplo manje novca, nudi enormno programsku podršku? Na to pitanje mi je teško odgovoriti; kakav god bio „orao“ je ipak naš računar. „Velebit“ jasno pokazuje da smatra da je njegov novi računar maksimalno

konkurentan — izrađeno je 1000 štampanih ploča, što je za naše prilike ogromna serija. Međutim, na sajamskom štandu još nema prototipa ni prospekata; izgleda da razvoj „orla“ (pogotovu njegovog softvera) još nije završen!

Očekujući „orla“, ostalo nam je samo da pogledamo Apple II računare. Velebit će već u trećem kvartalu ove godine prodavati poznati Apple IIe, koji će, sa jednom floppy-disk jedinicom, monitorom i štampačem Epson MX 100 ili FX 80 koštati nešto manje od 100 starih miliona. S obzirom na moćnu programsku podršku koju Apple ima, čini nam se da za mnoge škole predstavlja prilično zanimljivu alternativu.

## „Partner“ i „lola 8“

Iskra Delta je i ove godine izložila poznati Partner. Radi se o poslovnom sistemu koji je zasnovan na mikroprocesoru Z80A, opremljen sa 128 Kb korisničke (RAM) memorije, jednom disk jedinicom od 1 Mb i Winchester masivnim diskom od 10 Mb. Tastatura je odvojena od centralne jedinice (veza se ostvaruje preko kablja), na kojoj se nalazi i crno-beli monitor koji omogućava ispisivanje 80 znakova u svakom od 24 reda. Pošto je osivarena CP/M kompatibilnost, Partnerovim korisnicima su na raspolaganju mnogobrojni programski jezici, ali izgleda da su mnoge firme zainteresovane samo za računar koji će izvršavati određene knjigovodstvene poslove. Za njih se Partner isporučuje bez programskog jezika i opremljen samo potrebnim programom. Cena računara u standardnoj konfiguraciji koja uračunava i TRS štampač i jedan programski jezik iznosi 330 starih miliona.

Završavajući krug, stigli smo do kompjuterskog štanda „Ive Lole Ribara“ sa desetak računara lola 8 na kojima su zainteresovani mogli slobodno da rade. Lola 8 je prisutan već dve godine i polako (nadajmo se i sigurno) krči put ka tržištu. Operativni sistem računara je definitivno završen. Određen broj primeraka lola 8 je već našao svoje mesto u školama, a njima se pridružio i računar koji su na ovogodišnjem republičkom takmičenju u programiranju osvojili učenici beogradske Matematičke gimnazije „Veljko Vlahović“. Na sajmu je održano i Savezno takmičenje, koje se nije baš slavno završilo: učestvovali su tri ekipe (Beograd, Novi Sad i Zagreb), kojima su nadežni na sajmu u 18 časova jednostavno isključili struju i tako obrisali programe koji su trebali da budu pisani još čitav čas!

Tri nova domaća kućna računara „galeb“, „galaksija“ i PMP — predstavljaju, bez sumnje, bogatu konstruktorsku žetvu. Nove „prve domaće“ računare najavljuju i Elektronska industrija, Iskra Delta i Slovenjales. Bilo bi nam drago da se porodica YU računara još više proširi — možda čemo po broju modela uspeti da prestignemo Engleze! — ali bismo, iznad svega, voleli da se barem jedan od njih pojavi u knjižarama, robnim kućama i, razume se, našim domovima. Plašimo se da i ove godine, sa padom zavese na sajamsku priredbu, ne zaboravimo i na sve sjajne planove do sledećeg — sajma.

Dejan Ristanović

# Pismo iz Slovenije

## SLOVENAČKI RAČUNARSKI BUM

**Ono što se sada dešava u Sloveniji mogli bismo da okvalifikujemo kao „komputerski bum“. Prošlo je više od godinu dana od kako je slovenački mesečnik SUPER-PIP počeo da objavljuje prve skromne priloge sa područja savremene elektronike. Ali svet se menja, potrebe rastu, a i želje mladih za komputerskim znanjem sve su veće. Zato nije slučajnost što su tom problematikom počeli da se bave i drugi slovenački listovi i revije.**

Neuobičajeno u svemu tome je što su se ovih dana na slovenačkom novinskom tržištu, koje i nije osobito veliko, pojavile jedna za drugom dve računarske revije: BIT, koji će od sada izlaziti jedanput mesečno, a izdaje ga „Ljubljanski dnevnik“, i posebno „Teleksovo“ izdanje MOJ MIKRO. „Tačku na i“ dodali su „Radio študent“ i ZOTK (Savez organizacija za tehničku kulturu) Slovenije sa PROGRAMSKOM KASETOM, na kojoj se nalazi 10 programa za kućni računar Sinclair ZX SPECTRUM. Programi su isključivo rad domaćih autora, koji su se predstavili javnosti 6. juna ove godine na konferenciji za štampu u Cankarevom domu u Ljubljani.

Mladi su to momci, mladi kao što je mlad i razvoj računarske kod nas, i zato mislim da će biti pošteno da navedem i njihova imena: Žiga Turk, Matevž Kmet, Darko Vovk, Tomaž Iskra, Dragan Vidic, Primož Jakopin, Janez Kanič, Andrej Vibelčić i Dean Mozetič. Dva programa su takozvane igre, a jedna od njih, KONTRABANT (avantura-krijumčarenje) ima sva obeležja domaćeg „folklorā“. Cilj ove igre je nabaviti kućni računar, koji, na žalost, još uvek mora da se krijučkari. Komputerska igra KONTRABANT je, u suštini, putovanje po Jugoslaviji i zato je, kažu, dosta teška. Da bi se nagradio trud najboljih i najistrajnijih „igrača“, obezbeđeno je nekoliko primamljivih nagrada, među njima i INTERFACE 1 ZX Spectrum. Tačne propozicije će biti objavljene na komputerskim stranicama slovenačkih listova.

Možda ćemo ipak imati sreće, kažu mladi iz Slovenije. Sreća da do kućnih računara ne moramo više da dolazimo — kršenjem zakona. Naime, Savez socijalističke omladine Slovenije (ZSMŠ) upućio je saveznom i republikom izvršnom veću predlog da se dozvoli uvoz kućnih i džepnih računara. Ako želimo da se oslobodimo komputerske nepismenosti, za šta se zauzima i ZSM Slovenije, potrebno je da svi mladi, od

osnovne škole pa dalje, dobiju osnovna računarska znanja u vaspitno-obrazovnom procesu. U tom procesu trebalo bi da učestvuju zavodi za školstvo, komiteti za obrazovanje i vaspitanje i druge obrazovne institucije.

Jugoslavija je jedna od retkih zemalja u kojima je uvoz kućnih i džepnih računara faktički onemogućen. Kada bi se lični računari svrstali u predmete široke potrošnje, bio bi omogućen uvoz i, što je još važnije, njihova konsignaciona prodaja. Ako tome dodamo i razvoj domaće proizvodnje ličnih računara (koji, doduše, ne mogu da konkurisu stranim računarima ni po ceni ni po snazi), to bi već bio ogroman korak napred. Savezni sekretarijat za spoljnu trgovinu obznanio je da je u toku rad na izmenama i dopunama Odluke o uslovima pod kojima fizička lica mogu uvoziti i primati određene predmete iz inostranstva. Da li će se nešto promeniti u tretmanu kućnih računara?

U Iskri obećavaju da će se već ove jeseni na tržištu pojaviti prvi kućni računari HR-84 domaće proizvodnje — prvih 500 komputera, iz serije od 2.500 komada. Do tada bi Iskra trebalo da okonča i pregovore oko konsignacije Sinclairovih računara.

Sasvim mali broj Sinclairovih računara ZX81 iznela je u maju na tržištu i Mladinski knjiga iz Ljubljane. Međutim, to su računari koji imaju svega 1K RAM-a, što znači pola kucane stranice naredbi, pa zato stoji pod znakom pitanja njegova upotrebivost za osnovne i srednje škole. Bilo kako bilo, možda su to ipak prve svetle tačke na horizontu računarske pismenosti kod nas. Ipak, jedno je sasvim sigurno: mladi imaju i dovoljno energije i dovoljno znanja da iznesu komputersku revoluciju, ali, na žalost, za sada manje-više samo kao samonikli entuzijasti.

I baš zbog toga možemo da smatramo prvu slovenačku kasetu sa prvim domaćim programima za značajan događaj, za hrabar i pohvalan čin, utoliko pre što je softverska redakcija Radija-študent mlada — oformljena tek ove godine.

Traž kasete je 2.000 primeraka, cena 700 dinara (što je malo ako imamo u vidu da se u inostranstvu za jedan program plaća od 5—7 funti). Najvažnije u svemu je da oni za jesen pripremaju, kao što su obavestili na konferenciji za štampu, novu kasetu sa područja obrazovanja i saobraćaja. Dakle, prva slovenačka lista — nije i poslednja.

Kristina Iskra

Sajam u los Andelesu je osnovan još davne 1976. Njegov osnivač je Džim Voren (Jim Warren) — bivši hipik, urednik nekolicom komputerskih časopisa, nastavnik matematike, i, na kraju, doktor komputerskih nauka sa čuvenog Stenford univerziteta blizu San Franciska. On je, pošto je dobro poznao situaciju u Silicijumskoj dolini (južno od San Franciska, nekad čuvena po šljivcima i suvim šljivama, a danas centar svetske moderne tehnologije), gde se 1977. puno toga događalo u vezi sa kućnim računarima, odlučio da organizuje skromnu izložbu. Ali, već prve godine interes je premašio sva očekivanja i mala izložba je prerasla u sajam.

Džim je uvek bio široke ruke i nije odbijao nijednog izlagača — uvek bi se i za najmanju firmu ili pojedinca našao neki kutak. Priča se, tako, da osnivač APPLE-a Stiv Džob (Steve Job) nije imao nekoliko stotina dolara da 1977. plati za nekoliko prostora, ali Džim ga nije odbio, već je pristao da čeka na novac. Tako je Apple-a predstavljav javnosti. Znamo šta se kasnije dogodilo sa ovom kompanijom. No, Džim je prošle godine prodao sajam, jer kako kaže, nema više onog zadovoljstva kao prvih dana.

## Posetilaca sve više, ali nema varnica

Kako je ove godine bilo u San Francisku? Na žalost, nema više malih izlagača koji iznenađuju velikim idejama — sve same velike kompanije, a među njima gotovo polovina izdavači časopisa i knjiga. Kako piše novinar dnevnog lista San Francisco Kronikl (San Francisco Chronicle), „posetilaca sve više, ali nema varnica“. Naravno, vodeći IBM i APPLE zauzimali su najveće prostore. Apple, koji ove godine troši 15 miliona dolara na promociju Mekintoša (kakva razlika u odnosu na onih nekoliko stotina dolara iz 1976!), izložio je četiri metra visoki „Mek“ na čijem su se velikom ekranu slikovito pokazivale mogućnosti ovog računara. Bilo je i nekoliko starih modela Apple II/e i Lise, ali na njih niko nije obratio pažnju.

IBM-ov izložbeni prostor bio je najviše posvećen novom PC junioru, uz stariji PC kompjuter, ali je posetiocima ostavio prilično ravnodušnim! Dobro poznavajući tržišta tvrde da se on slabo prodaje (vaš dopisnik je obišao više radnji u Kaliforniji i dobio je isti utisak). Od ostalih novosti sa sajma je to bio OSBORNE, posle bankrotstva, ponovo oživeo (njegov osnivač Adam Osborne je smenjen — sada vodi jednu softversku kompaniju). Interesantan je bio i štano japanskog nacionalnog Mikao udruženja — oni su prikazali najnovije proizvode japanskih kompanija.

Sa knjigama i časopisima nastupilo je mnoštvo izdavača. Dosta interesovanja izazvala je i humoristička knjiga „Mrzim kompjutere“ (I hate computers). Svaki računar, kao i u Evropi, prati po nekoliko časopisa, „Mekintoš“ ih sada ima već dva (Mackword i St. Mac), a izdavači najavljuju još pet novih.

I pored visoke cene ulaznica (12 dolara) sajam je posetilo preko 40.000 entuzijasta. Na američkim sajmovima se, uzgred, uporedo održavaju i besplatni seminari, kojima može prisustvovati svaki posetilac. Na njima govore najpoznatija imena iz ovog područja i može se naučiti dosta toga.

# na zapadu ništa novo

Računari u sve

U proleće svake godine na zapadnoj obali Amerike održava se nekoliko sajmova za kućne računare na kojima se najbolje može videti koji vetrovi duvaju u ličnoj informatici. Vaš dopisnik već sedam godina napravi bar jedno putovanje preko Atlantika da vidi nove proizvode i trendove. Tako je bilo i ove godine. Posetili smo najstariji sajam te vrste, IX West Coast Computer Faire u San Francisku, i sajam Comedex (zatvorenog tipa) u Los Anđelesu. Dok je prethodnih godina bilo iznenađenja i uzbuđenja na pretek, prošle, a pogotovo ove godine, ovi sajmovi su počeli da gube svoj tradicionalni šarm. Od kako su poslednjih godina velike multinacionalne kompanije zavladaile ovom industrijom, nema više genijalaca koji će u svojoj garaži napraviti novi računar ili u studentskoj sobici sastaviti softverski program koji će se prodati u 100.000 primeraka. Da li starimo ili smo previše romantični?



Na visokoj nozi. Tridesetdvobitni Macintosh, nova nada kompanije Apple i čitave američke lične informatike, košta svega oko 2.500 dolara

## Računari, telegrafi i...

Sajam u Los Anđelesu je drugačijeg karaktera — trebalo bi da bude samo za ljude iz kompjuterske industrije. Međutim, ko god je bio raspoložen da plati 25 dolara, dobijao je bez da natisom „gost izlagača“ koji mu je otvarao vrata u dvoranu. Uz oko 350 manjih izlagača, bili su prisutni i svi računarski džinovci: IBM, APPLE, RADIO SHACK, DIGITAL. Uz dve britanske (Acorn i

Memotech), bilo je, takođe, i nekoliko japanskih kompanija. Najviše prostora zauzela je sa svojim računarom džinovska kompanija ATT (American Telephone and Telegraph) koja ima skoro monopol u Americi na telefonsku i telegrafsku mrežu (za razliku od mnogih zemalja u svetu, ovde je to u privatnim rukama — samo je poštanska služba u državim).

Pojava ove kompanije na tržištu računara predstavlja prvorazredno iznenađenje. Možda će od toga zaboleti glava mnoge proizvođače — ne toliko zbog samog računara koliko zbog velikih finansijskih sredstava koje nameravaju da uložu u publicitet i propagandu. I pored svih pokušaja, japanskoj industriji još uvek ne polazi za rukom

da zauzme iole značajniju ulogu na tržištu. Sada su uveli novu taktiku — neki njihovi računari se prodaju pod imenima RADIO SHACK COMPAQ, jer ih oni proizvode jeftinije od Amerikanaca. Na tržištu periferijske opreme, posebno štampača i disk jedinica, Japanci vode glavnu reč. Posle pobune koju je ovde doživeo Acorn BBC (ali to je prošlo), najpoznatija britanska kompanija ne postiže neki veći uspeh. Acorn se prodaje školskim institucijama, ali dosta slabo, jer je sa cenom od oko 1.000 dolara previše skup.

Ove godine, dakle, dosta razočarenja. Sledeće će, možda, biti bolje.

Andelko Zgorelec

# Evolucija mozga od dinosaura do kompjutera u susret superinteligenciji

Zamislite tiranosaura reksa, najmoćnijeg među dinosaurima, kako se osvrće oko sebe u bujnoj prašumi jednog davnog sveta. Visok je dva sprata; bedra mu se nadičaju od snage; u njegovim razapjelim četvrtima bleska nekoliko redova zuba oštih kao brijac; on je jedna moćna mašina za razaranja. Ali u njegovoj ogromnoj lobanji smestana je majusana mozak. On je glupa životinja.

Dok se mrak spušta, jedno malo, dlakavo stvorenje izlazi iz svojeg skrovišta, gonjeno ljutom gladu, i kreće oprezno u okolno šipražje po svob okrov crva i insekata. Tog trena ogromna telesina tiranosaura zamračuje nebo; njena senka pada na malu životinju i ova žurno beži tražeći spas.

To stvorenje je jedan od prvih sisara na Zemlji. Ono nije impozantna borbeno mašina, ali je intelektualni džin svoga vremena; ono je najinteligentniji oblik života koji se razvio na planeti do tog perioda. Srazmerno težini svog tela, ono je 5 puta pametnije od tiranosaura, a 20 puta pametnije od bilo kojeg dinosaura kojima se tiranosaur hrani.

Žašto je taj rari sisar bio toliko pametniji od dinosaura? Niko nema pouzdan odgovor, ali verovatni razlog je taj što je on tada imao male šanse na uspeh, a pritisci pod kojima je živio nametali su potrebu velike inteligencije u borbi za opstanak. Malo stvorenje mora da je bitisalo u stanju neprekidne strepnje, krijuci se preko dana i tragajući noću za hranom pod mučnim i neizvesnim okolnostima. Bespomoćno, moralo je da živi od svoje pameti.

Mada su se sisari i dinosauri međusobno veoma razlikovali, oni su bili rođaci, jer su poticali od iste loze drevnih reptila. Drevni reptili, sa svoje strane, razvili su se iz amfibijsa, a amfibijsa su potekle od riba. Svi ti oblici života — ribe, amfibijsa i reptili — bili su relativno neinteligentni, a i njihovi potomci ostali su takvi sve do danas. Ali sisari su bili drukčiji; priča o evoluciji inteligencije u stvari je njihova priča. Pojava sisara označila je prvi veliki korak u evoluciji mozga.

## Gospodari planete

U početku, sisari — ili, tačnije rečeno, reptili nalik na sisare, dakle jedan prelazni oblik — bili su svirepi, mnogobrojni i strahovito uspešni; u stvari, pre 250 miliona godina oni su bili dominantan oblik života na kopnu. Ali kasnije, iz nekog razloga, opali su u broju i veličini i potonuli u tamu, dok su ih dinosauri nasledili kao gospodari planete.

## 8/evolucija mozga od dinosaura do kompjutera

Sisari su ostali potčinjeni vladajućim dinosaurima više od 100 miliona godina, promenivši se vrlo malo tokom tog dugog intervala. A onda, otprilike 70 miliona godina unazad, klima na Zemlji počela je naglo da se menja, i dinosauri, ne više u harmoniji sa svojom životnom sredinom, postepeno su iščezli. U roku od svega nekoliko miliona godina potpuno su nestali. Sisari, sada odjednom nespunati, uselili su se u životne prostore koje su upraznili mrtvi dinosauri i rascvetali se u čudesno mnoštvo raznovrsnih oblika. Neki su se opredelili za vazduh i postali tetaci; drugi su se vratili moru i postali kitovi i delfini; treći su ostali na šumskom tlu; a jedna grupa se uspenjala na drveće da bi postala preteča majmuna i čovekolikog majmuna.

Poput njihovih srodnika koji su istraživali i osvajali druge životne prostore, ovim životinjama nastanjenim na drveću bile su potrebne specijalne telesne odlike koje će ih opremiti za taj novi život: ruke za dohvatanje grana, na primer, i binokularan vid za procenjivanje razdaljine od grane do grane. Tokom vremena, te odlike su evoluirale. Ali stanovnik drveta morao je da ima veći i bolji mozak. Kad je ukapao s jedne grane na drugu bio mu je potreban brz, precizan kompjuter u glavi za kombinovanje faktora razdaljine, brzine vetra, pokreta grane i razloženja tela. A izračunavanje je moralo da bude objavljeno u magnoevanju, jer bi svaka pogreška bila plaćena smrću ili teškom povredom. Tom prirodnom kompjuteru — mozgu — bila su potrebna prefinjena mentalna kola za obavljanje neophodnih proračuna, a bila mu je potrebna i velika banka memorije za skladištenje rezultata bivših iskustava u vazdušnim manevrima.

Drevni sisari, koji su živeli na šumskom tlu, posedovali su jednostavniji mozak bez tih izvanrednih kvaliteta. Ali još jednom zakon opstanka najsposobnijih stupio je u dejstvo. Zahvaljujući malim varijacijama od jedne do jedne koje se dešavaju u svakoj populaciji, neke životinje su posedovale nešto bolji mozak od drugih; te životinje su imale veće šanse na opstanak, prenoseći svoju superiorniju moć mozga na potomstvo; dok su one koje su bile u manjoj meri obdarene tim mentalnim odlikama pokazivale sklonost da nestanu u ranom dobu, i njihovi geni su iščezli iz populacije.

## Dejstvo selekcije

Pod dejstvom prirodnog odabiranja mozak stanovnika drveta poboljšao se i u pogledu kvaliteta i u pogledu veličine; istovremeno, njegova manuelna spretnost i oštrina vida nastavili su da se razvijaju. Nakon 40 miliona godina neprekidnog poboljšavanja tela i mozga, potomci prvog stanovnika drveta postali su nova vrsta stvorenja. Ta nova stvorenja bila su maj-

mun — i njegov rođak, pračovekoliki majmun.

Majmun se nije mnogo promenio od tog vremena do danas; njegova je priča bila završena. Ali evolucija čovekolikog majmuna se nastavila. Postao je veći i teži, i siao je sa drveta da bi otpočeo novo bitisanje na šumskom tlu, vrativši se prebivalištu koje su njegovi majmušni preci napustili 40 miliona godina pre toga.

Sada je bila postavljena pozornica za još bržu evoluciju inteligencije.

Pre otprilike 15 miliona godina, jedna napredna grupa čovekolikih majmuna — pređaka šimpanze i gorile — nastanjivala je šume istočne Afrike. Još jednom, promena klime na planeti skrenula je tok evolucije. Nastupila je era hladnijeg i suvljeg vremena, uslovivoj pojavu otvorenih komada zemljišta za napananje. Dok je suvo vreme nastajvalo da traje, pašnjaci su se sve više širili, i teren se transformisao u poznatu afričku savanu.

## Životinja slična čoveku

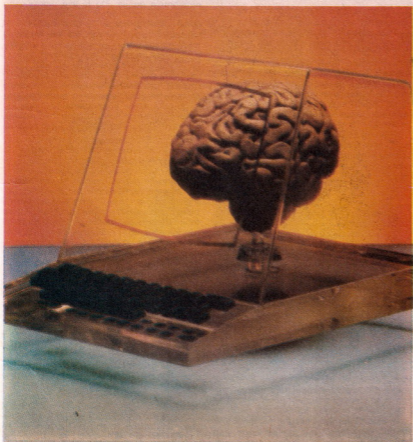
Savana je imala privlačnu draž nepoznatog za čovekolike majmune. Postepeno, najsmeliji i najradozniji među njima napustili su bezbednost drveta da bi istražili tu čudnu novu sredinu. Ne znamo šta se iza toga desilo, jer u fosilnim nalazima tada nastupa jedna praznina od nekoliko miliona godina. Znamo samo da se šumskim čovekolikim majmunima otvoreno zemljište dopalo i da su ostali tamo, jer kada se fosilni zapis ponovo pojavljuje, on pokazuje da su se njihovi potomci uveliko odomacili u savani. Ali potomci nisu više čovekoliki majmuni; oni su jedina nova životinja koja tada uspravno, korakom nalik čovekovom. A ta nova životinja ima superiornu mozak; u odnosu na telesnu veličinu, mozak je dva-već veći od mozga čovekolikog majmuna.

Prema fosilnim zapisima, ta dvonoga, inteligentnija životinja pojavila se u Africi pre tri do četiri miliona godina. Njeno ime je australopithec afarensis. Bio je to sposoban lovac koji je jurio sa ostalim mesožderima svog vremena, nadmećući se sa lavom i dzinovskom hijenom. Međutim, australopithec je bio slabšaš, nejake grade, težak možda samo oko trideset pet kilograma, i bez oštih očnjaka ili drugog prirodnog oružja. Ali njegove fizičke slabosti bile su kompenzovane snagom njegovog intelekta.

Koje su okolnosti stvorile tu neobičnu moć mozga australopitheka? Odgovor nije jasan, ali proučavaoci ljudske evolucije uglavnom se slažu da je uspravno držanje verovatno bilo ključni faktor. Sa rukama odnedavno oslobođenim za nošenje hrane i oružja i za izradu i korišćenje oruđa, australopithec je imao potencijal za stvaranje kulture. Ali kako je taj potencijal bio realizovan? Kako su te telesne odlike dovele do porasta mozga?



*Ako je prošlost neki putokaz za budućnost, onda možemo očekivati nastanak jednog višeg oblika inteligencije. Kako će izgledati taj prelinjeni mozak? U čemu će se razlikovati od našeg? Da li elektronski računari, pa i ovi kućni, predstavljaju samo primitivni zametak buduće superpametli ili se radi o stranputici na putu za budućnost? O tome piše Robert Džestrou (Jastrow), američki naučnik, publicista i esejista.*



Teorija o prirodnom odabiranju nudi odgovor. Prvi izradivači oružja u toj grupi dvonogih čovekolikih majmuna mora da su se razlikovali međusobno po stepenu dovrtljivosti i ingenioznosti koje su upražnjavali u svom svakodnevnom životu. Oni koji su posedovali te osobine u najvećem stepenu bili su istinski junaci dana; oni su imali najviše izgleda da načine efikasna oružja za ubijanje divljači i najbolju strategiju za lov. Oni su imali uspeha; oni i njihovi potomci su napredovali; njihovi geni su se sve više ukorenjivali u populaciji; odlike dovrtljivosti i ingenioznosti intenzivirale su se među dvonogim čovekolikim majmunima.

Dovrtljivost i ingenioznost — to su dva lica inteligencije, a njihovo sedište je u

mozgu. Ako se ovakve odlike intenziviraju, mozak mora da poraste. Na taj način je uspravno držanje, koje je dovelo do izrade i korišćenja oruđa, doprinelo ubrzanjem porasta mozga u populaciji dvonogih čovekolikih majmuna.

### *Automatske poruke*

Mozak koji je rezultirao nije bio velik u apsolutnom smislu; imao je tek veličinu pesnice i težio jedva 450 grama; s druge strane, međutim, ni telo koje je kontrolisao nije bilo veliko. U procenjenjivanju inteligencije neke životinje po veličini njenog mozga, važno je izračunati proporciju veličine ili težine mozga u odnosu na telesnu težinu, zato što svaka životinja koristi jedan deo svog mozga za nesvesnu kontrolu tela. Taj segment mozga služi kao telefonska razvodna tabla, koja prima signale od čula i, sa svoje strane, odašilje automatske poruke: Beži od opasnosti! Približi se hrani! I

tako dalje. Samo višak sive materije, koji preostaje posle tih automatskih funkcija, stoji na raspolaganju za postupke koji mogu da dodu pod kontrolu inteligencije, kao što su pamćenje, planiranje i fleksibilno, neprogramirano reagovanje na nove izazove. Australopiteku, sa malim telom, bio je potreban relativno mali broj moždanih ćelija za telefonsku razmenu u njegovoj glavi, pa je zato imao na raspolaganju veću količinu sive materije za pamćenje, planiranje i apstraktno mišljenje.

Australopitek je stekao tu moć mozga pre dva ili tri miliona godina, nekoliko miliona godina nakon što su se njegovi drevni preci prvi put preselili u savanu. Izmenio se donekle tokom nekoliko sledećih miliona godina, evoluirajući uspešno u životinju poznatu pod imenom australopitek afrikanus a zatim u telom krupnijeg australopiteka robustusa. Posle toga se promenio veoma malo, kako u pogledu mozga tako i u pogledu tela. Poput mnogih drugih oblika života u prošlosti, australopitek je dospao u čoršak. Pre sedamsto pedeset hiljada godina potpuno je izumro.

Pre nego što će iščeznuti, australopitek je iznedrio drugu inteligentnu životinju afričke savane. Novo stvorenje je nazvano homo erektus — čovek koji stoji uspravno. Njegovo držanje bilo je uspravno, kao u australopiteka, ali plemenitije; njegovo telo i crte lica bili su više čovekoliki a manje majmunoliki; a njegov mozak bio je veći.

U početku, razlika u veličini mozga nije bila osetna, ali inteligencija australopiteka ostala je gotovo nepromenjena, dok je inteligencija homo erektusa i dalje rasla. Najnovija etapa u evoluciji inteligencije na Zemlji tek što nije bila otpočela.

### *Neumitni trend evolucije*

Sada nas je priča dovela do jednog vremena, pre otprilike milion godina, kada je australopitek upravo trebalo da iščezne iz savane. Homo erektus već je lovio sa zapanjujućom veštinom; nijedna divljač nije bila suviše velika za njegovo junaštvo; a njegova tehnologija kamenog oruđa bila je impresivna. Ojednom, rast mozga se ubrzao. Niko ne zna tačno zašto; smatra se da oruđa, jezik i prilisci ledenog doba spadaju među glavne razloge; ali, ma šta da je bilo uzrok, fosilni nalazi demonstriraju rezultate.

Blizu 450 grama sive materije bilo je pridodato ljudskoj lobanji za manje od milion godina. Veći deo bio je u prednjem delu mozga, koji je centar najviših domena apstrakcije i kreativnosti. Dok su se dodatni neuroni gomilali u lobanji ranog čoveka, njegovo celo se ispućavalo, lice je izgubilo svoj zverski izraz, i on je počeo da poprima izgled intelektualnog bića. Homo sapiens — čovek obdaren razumom — stigao je na planetu.

Danas čovek predstavlja vrvinsku tvorinu prirode na Zemlji. Kakva budućnost predstoji toj izuzetnoj životinji? Možda će i on iščeznuti, kao što je australopithec izumro pre njega; devedeset odsto svih oblika života koji su postojali na Zemlji vremenom su izumrli. Ili će, možda, živeti neizmenjen sve do u daleku budućnost — jedan živi fosil poput školjke kamenice. Ta sudbina možda se već nadnela nad nas, jer ljudsko telo se izmenilo veoma malo tokom poslednjih milion godina, a ljudski mozak se nije izmenio, bar ne u ukupnoj veličini, tokom proteklih 100.000 godina. Organizacija mozga možda se poboljšala za to vreme, ali ograničene su količine informacija i uzajamnih poruka koje mogu da budu strpane u lobanji fiksirane veličine. Činjenica da se mozak više ne širi, posle milion godina eksplozivnog rasta, sugeriše da je priča o ljudskoj evoluciji možda već završena.

To ne znači da je završena i evolucija inteligencije. Naime, razumno je pretpostaviti da ljudska bića ne predstavljaju poslednju reč u evoluciji na Zemlji, već samo izvornu lozu iz koje će se razviti jedan novi i viši oblik života, da bi prevazišao naša dostignuća kao što smo i mi prevazišli dostignuća australopithec. Istorkija života potvrđuje taj zaključak, jer pokazuje neuobičajen trend ka većoj inteligenciji kod viših oblika životinja. Očigledno, među svim odlikama nekog živog organizma, nijedna nema veću vrednost za opstanak od fleksibilnog, inovacijskog reagovanja na izmenjene okolnosti, reagovanja koje nazivamo inteligencijom. Izgleda neverovatno da će taj trend u evoluciji biti iznenađeno okončan na jednom određenom nivou mentalnog dostignuća koji nazivamo ljudskim. Ako je prošlost neki putokaz za budućnost, čovečanstvu je, izgleda, sudeno da dobije jednog još inteligentnijeg naslednika.

## Intelektualac

Kakav će izgled imati naš naslednik? Sudeći po skorajšnjoj čovekovoj istoriji, novi oblik života ličiće na stari, ali će imati još veći mozak. Prema tom predviđanju, sledeća vrsta inteligentnog života na Zemlji biće intelektualac, stvorenje slično nama ali sa većom lobanjiom i slabijim mišićima.

Međutim, izvesni trendovi u modernoj tehnologiji sugerišu sasvim drukčije viđenje budućnosti. U dejstvu su moćne sile evolucije — prevashodno kulturne a ne biološke — koje mogu da dovedu do jednog egzotičnijeg oblika inteligentnog života — oblika koji će se razviti iz čoveka, ali pre kao dele njegovog mozga nego kao dele njegovih slabina.

Prema tom viđenju, novi oblik života se danas već stvara u laboratoriji kompjuterskog naučnika. To je jedan veštački život načinjen od silicijumskih čipova a ne iz neurona, a ipak, on misli, pamti, uči se na iskustvu i reaguje na nadražaje. Njegovu razmišljanje je još uvek nepristojno; nije mnogo kreativan; ali evoluiru zapanjujućom brzinom.

Ideja izgleda apsurdna: kako može bogatstvo ljudske misli da bude uporedivano sa mehaničkim razmišljanjem jednog kompjutera? Pa ipak, iznenađujuća stvar kod kompjutera nije u tome što oni misle

manje dobro od čoveka, nego što uopšte misle. Tačno je da su elektronski mozgov današnje primitivniji u poređenju sa ljudskim mozgom; u stvari, oni nam daju malo razloga da im se divimo, sem zbog svoje ogromne memorije i izvesnih matematičkih veština. Međutim, najnoviji modeli mogu biti uprogramirani tako da prate neku raspravu, postavljaju drčna pitanja, ili da pišu dopadljivu poeziju i muziku. Isto tako, oni mogu da obavljaju ponešto neobavezne konverzacije toliko ubedljivo da njegovi ljudski partneri nisu ni svesni da razgovaraju sa mašinom.

## Silicijumski mozak

To su dopadljivi kvaliteti kompjutera; on imitira život kao neki elektronski majmun. Dok kompjuter postaje kompleksniji, i imitacija postaje bolja. Na kraju, crta koje razdvaja original od kopije postaje zamagljena. Kroz sledećih pedesetak godina — oko 1995. prema tekucim trendovima — mi ćemo videti kako se silicijumski mozak javlja kao jedan novi oblik života, koji može da se takmiči sa čovekom.

Pre nego što odbacimo tu viziju, razmotrimo neka skorajšnja dostignuća u kompjuterskoj industriji. Prvu istinski modernu generaciju kompjutera reprezentovala je mašina IBM 360 koja je puštena na tržište 1964. godine. Jedna mašina tog tipa obavljala je posao za Godardov institut za proučavanje svemira tokom poslednjih 15 godina. Koliko bi velika bila takva mašina ako bi je trebalo proširiti toliko da odgovara kapacitetu ljudskog mozga?

Mozak je težak približno 1.400 grama, troši elektricitet snage otprilike 20 vata, i zauzima zapreminu tridesetog dela kubnog metra. Unutar tog malog prostora on sadrži memoriju od 10 milijardi informacionih jedinica. Jedan kompjuter IBM 360 tog istog kapaciteta trošio bi električnu energiju od milijardu vata, ispunjavao bi čitav Svetski trgovački centar, i koštao bi mnogo milijardi dolara. Bila bi to nezagrapna imitacija mozga.

Ali najnovije mašine na tržištu 1980. samo 16 godina kasnije, pričaju drukčiju priču. Jedna od tih novih mašina, proširena toliko da odgovara kapacitetu mozga, mogla bi se smestiti u prostorije neke kancelarije, i trošila bi samo 100.000 vata elektriciteta. Bila bi još uvek nezgrapna, ali gotovo vredna izgradnje.

## Portabl kompjuter

Dalja poboljšanja, koja se sada odvijaju u laboratorijama za kompjutere, biće spremna za tržište oko 1990. godine. U to vreme moći će se izgraditi kompjuter čovečjeg kapaciteta koji bi mogao da se smesti u kofer i da operiše pri utrošku električne energije od 1.000 vata.

Oko 1995. krivulja rasta silicijumskog mozga bi da pređe crtu ljudskih sposobnosti — 10 milijardi činjenica u jednoj aktaentasi — sa utroškom od svega 20 vata električne energije.

Naravno, silicijumskom mozgu koji odgovara ljudskom mozgu samo po smeštaju, kapacitetu i moći grubog rasuđivanja nedostajae kreativna inspiracija i intuicija koje karakterišu više domena ljudske misli. Poput nekog novopečenog doktora filozofije, on će imati ogromne mentalne moći, ali ne i mudrost. Pa ipak, i tu kvalitativnu superiornost mozga nad kompjuterom sve

više podrivaju izvesna novija razvoja kretanja. Da bi se shvatilo značaj tih kretanja, razmotrimo za časak arhitekturu ljudskog mozga. Svaka moždana ćelija je direktno povezana sa mnogim drugim moždanim ćelijama, u nekim slučajevima čak sa 100.000. Kao rezultat toga, kad god pošaljemo neki svesni impuls dole u dobine naše memorije da prizovemo neku određenu informaciju, ćelije u kojoj je ta informacija pohranjena komuniciraju na podsvesnom nivou sa hiljadama drugih ćelija, i jedan talas mentalne aktivnosti prosturu kroz čitav mozak. Plodovi te podsvesne aktivnosti su intuitivna spoznaja, bleesak percepcije i nadahnuća genija, što je sve omogućeno zahvaljujući bezbrojnim vezama među ćelijama ljudskog mozga. Te veze među ćelijama objašnjavaju mnogo šta od neobične moći mozga.

Arhitektura kompjutera je sasvim drukčija. Kompjuterske memorije su kao jedna garnitura otvora na golubarniku porađanih uža zid, bez povezanosti jednog otvora sa drugim. Informacija može da bude stavljena u golubiji otvor ili izvađena iz njega, ali golubiji otvori ne komuniciraju međusobno, niti razmišljaju. Razmišljanje se obavlja na drugom mestu.

## Integrirana kola

Najnoviji razvoj koji menja arhitekturu kompjutera baziran je na izumu koji nazivamo čipovima. U svom najnovijem obliku čipovi nose naziv integrirana kola. To su tanke četrnaest pločice silicijuma, druge otprilike šezdeset milimetra po jednoj strani, koje su povezane međusobno provodnicima u velikom broju, one formiraju mislećim i memorijske sekcije elektronskog mozga. U starim danima, oko 1960. ili 1970. svaki silicijumski čip mogao je da drži samo mali broj tranzistora i elektronskih elemenata. Rečnik tog bio je da je jedan čip mogao ili da misli ili da pamti, ali ne oboje. Danas, međutim, vodeći proizvođači integriranih kola u Sjedinjenim Državama i Japanu naučili su kako da natrpuju 100.000 tranzistora i elektronskih elemenata na jedan jedini čip. Sada jedan čip može i da misli i da pamti. Ali se svaki čip povezan sa mnogim drugim čipovima, kompjuter postaje pravi silicijumski mozak; poput ljudskog mozga on funkcioniše kao jedna jedinica, obavlja misaone procese u talasima unutrašnje mentalne aktivnosti na „podsvesnom“ nivou. Ove se već približavaju onoj vrsti asocijativnog mišljenja koje predstavlja tako moćan element u ljudskom rasuđivanju.

To su izgledi koje možemo predvideti za poslednju deceniju našeg veka. U to vreme silicijumski mozgovci će u saradnji sa našim najboljim umovima raditi na svim ozbiljnim problemima dana, u jednoj nepobedljivoj kombinaciji moći sirovog rasuđivanja i ljudske intuicije. A šta će u samom zadatku? Šta u sledećem stoleću? I šta u stoleću iz toga? Ne mogu se predvideti granice rastuće krivulje silicijumske inteligencije. Jedan od vodećih ljudi u kompjuterskom istraživanju, profesor Marvin Minski sa MIT-a, veruje da će se moći ostvariti mašina sa opštom inteligencijom jednog prosečnog ljudskog bića. Mašina će početi da obrazuje same sebe, kroz nekoliko meseci će biti na nivou genija, a nekoliko meseci poste toga njena će moć biti neizračunljiva. A onda, kaže Minski: „Ako nam sreća bude naklonjena, one će možda odlučiti da nas drže kao kućne ljubimce.“

# Računari i mladi *era* tehnoloških genija

*Mikroelektronska revolucija, koja je, sa prilično zakašnjenja, najzad zapljusnula i naše obale i za sobom povukla na hiljade mališana, zatekla je roditelje i pedagoge potpuno nespremne. Šta se to najednom dešava sa mladima? Da li su se oni naprečak okrenuli „korisnim“ stvarima ili i njihov novi hobi ima svojih skrivenih zamki i opasnosti? Koje je stvarno mesto računara u obrazovanju, a koje u životu? Šta očekuje one koji se profesionalno opredele za računarstvo, a šta one koji ostanu „kompjuterski nepismeni“? Ove stranice, koje prenosimo iz knjige „Računari i vaše dete“ engleskog sociologa Reja Hemonda (Ray Hammond), treba da pomognu (zbuđenim) roditeljima da bolje shvate odnos dece prema računarima i pravilnije usmere njihov razvoj.*



*Na muškoj teritoriji: Interesovanje za računare je strogo podvojeno na polove — ne zato što su devojčice za njih „glupe“ a dečaci „pametni“ nego isključivo iz socijalnih razloga*

U mnogim delovima sveta mladi se obrazuju u sistemima koji su u haotičnom stanju. Masovni priliv računara, uz povećanu verovatnoću da se posle završenog školovanja neće moći doći do zaposlenja, lišio je većinu obrazovnih sistema svake koherentne politike. Svedoci smo, od 1980. godine, lude trke da se mladi donekle kompjuterski opismene pre nego završe školu; sa širenjem recesije u svetu, pretnja nezaposlenosti je tu tendenciju samo pospela. Većina roditelja smatra da njihovo

dete treba da „razume kompjuter“, ali ni dve osobe, izgleda, ne mogu da se slože kako da se to ostvari ili u čemu bi se to sastojalo.

## *Greška u koracima*

Nezaposlenost je posebno oštra među onima koji završavaju školu. U nastojanju da mladima otvori veće mogućnosti zaposlenja, većina srednjih škola pruža sada neki oblik kompjuterskog obrazovanja. To se najčešće radi tako da se deci predaje programiranje u okviru nekog kursa „kompjuterskih studija“. Na ovim kursevima najčešće se predaje bejzik, ali mnogi nastavnici već razmišljaju i o alternativnim kompjuterskim jezicima, kao što su komal i algol.

Deca mlada od 12 godina, u stvari, nisu u stanju da rade ni sa jednim od ovih apstraktnih algoritamskih jezika, i u mnogim srednjim školama obrazovanje u računarima ne počinje pre nego što dete uđe u treću godinu (osmi ili deveti razred).

Postoji mišljenje da je pogrešno uvoditi mlade u svet računara tako što će im se predavati programiranje. To je, smatra se, kao da se onome ko treba da nauči da vozi avion govori o tome kako je on konstruisan. Odvajajući računarstvo u poseban studij i tražeći od učenika da dolazi u računarsku sobu (često lociranu u matematičkom kabinetu), škole odvajaju računarstvo od drugih „običnih“ predmeta i ističu njegovu matematičku ili naučnu osnovu. Ta se situacija često uporno održava, jer je nastavnik matematike ili egzaktnih nauka prvi uvek računarstvo u školu, pa nastoji da zadrži ličnu vlast nad svojom „teritorijom“.

U nekim školama, računarski predmet se slabo probijaju u osnovni nastavni program, pa se računarstvo još uvek predaje kao grana nauke. Po jednom interesantnom tumačenju, računarska nauka je sa drugim predmetima, kao što su ekologija i fizičko obrazovanje, izbačena iz nastavnih programa jer su ambiciozni nastavnici pažnju usmerili na predmete koji im donose više prihoda i daju više vlasti u njihovim institucijama. Postojeći nastavni programi u srednjim školama „ispitnu su usmereni“; sposobnost kako učenika tako i nastavnika određuje se samo uspehom na ispitima. Ugled, unapređenja i novac idu nastavnicima koji predaju teorijske predmete koji se najlakše mogu uključiti u ispitni format. Izgleda da ima dosta dokaza u prilog ovog mišljenja; kao neposredna posledica ovog i drugih pritiska, u mnogim srednjim školama računar ostaje zaključan u matematičkom ili naučnom kabinetu.

## *Rezervat za dečake*

Ovo odvajanje računara ima i drugu tešku posledicu. Među decom između 11 i 18 godina, računari i računarstvo smatraju se prvenstveno „muškim“ predmetom. To je, delimično, stoga što se računar često čuva u „muškim“ odeljenjima — naučnim ili matematičkim — pa je njegovo korišćenje još uvek u vlasti muškaraca, a delom zbog načina na koji se računar koristi u nastavi.

Razlike u intelektualnom razvoju polova dugo su predmet istraživanja i suprotnih gledišta; postoje neki površni dokazi da su u toku mladalačkih godina dečaci u matematičkim uspešniji od devojčica. Ovakvi nalazi se moraju primati s rezervom, jer mogu biti

posledica načina na koji se devojčicama predaje matematika. Uticajna studija britanske vlade „Socijalni trendovi“, koja se objavljuje svake godine, pokazuje da su devojčice u celini uzev, uspešnije od dečaka, da 32% devojčica koje završe srednju školu odlazi na dalje školovanje, prema 22% kod dečaka. U najboljim grupama, međutim, dečaka je nešto više od devojčica.

Ono što je, međutim, jasno, jeste da računari pripadaju dečacima — bar u ovom trenutku. Zahtevi tinejdžera (od kojih su mnogi ljudi za vasijskim igrama) prisiljavaju mnoge porodice da nabave računar. Proizvođači su svesni ogromne snage tog tržišta i koriste je, podstičući postojeće trendove. U ratom mladalačkom dobu, seksualne uloge su polarizovane. Kada dečaci i devojčice počnu da otkrivaju svoju seksualnost, oni se povlače svako u svoj ugao kako bi razmotrili situaciju. Na tom tržištu, oglašivači izložbe kućnih računara uglavnom počivaju na igrama koje se nepogrešivo odnose na divlje ratove u kosmosu i herojska dela — sva muška po pristupu. Proizvođači se ne usude da tinejdžerima njihovo interesovanje za računare prikažu kao „detinjasto“ — radi se o grupi nedovoljno odrasloj da spozna i kontroliše svoj divlji separatistički svijet. Tako se računar svrstava u kategoriju kartinga i futbola, a devojčice koje nemaju sreće da u njemu otkriju ono što ih interesuje, celu stvar tako i klasifikuju.

Kućni računari se masovno kupuju na tinejdžerskom tržištu. Pre dve decenije tako je bilo sa električnom gitarom. I kao što su se šezdesetih godina roditelji nadali da će njihovo dete napredovati i stići do „pravog“ instrumenta, tako se današnji roditelji nadaju da će „osvajajući kosmos“ dovesti do konstruktivnijeg pristupa primeni računara. Verovatno je da će oni u tom svom nadanju imati više sreće nego što su je imali njihovih roditelji.

Akcione igre su izvanredna atrakcija za dečake (mada ih vole i mnoge devojčice). Pravilno usmereni, dečaci i devojčice će napredovati ka kreativnijem korišćenju računara. Izgleda da neka deca postaju veoma vezana za „pakmena“ ili njegov ekvivalent, ali iako se često koristi reč „zavisan“ (kao u slučaju droge), nema dokaza da dolazi do lako ozbiljnog stanja.

## Veliki igrač

Veoma se mnogo istražuju motivi koji navode mlade da troše novac i slobodno vreme na akcione igre. Kod pakmena, na primer, svaka uništena tačka predstavlja dostignuće, trenutni uspeh, a ako naša igra nije vrhunska, protivnik — mašina — nije u mogućnosti da nam se podsmejuje. Igre daju učesnicima mogućnost da se istaknu i veoma vešti igrači mogu osvojiti zavidan „igrački ugled“. Da bi se dalje shvatila privlačnost akcionih igara, treba razmotriti šta one stvarno daju. Jasno je da igrači bivaju potpuno apsorbirani — osećanje koje je teško moglo ostvariti u mnogim drugim aktivnostima. O toj apsorbiranosti se govori kao o posebnom duševnom stanju u kome se osoba koncentriše na trenutni zadatak, isključujući sve druge unutrašnje ili spoljne stimulanse. Akcija i svest se



Status quo: Devojčice u Engleskoj se znatno ređe odlučuju za računarske studije od dečaka i taj odnos se već godinama ne menja

stapaju tako da osoba prosto radi ono što se mora uraditi, bez kritičke, dvojne perspektive sopstvenih akcija. Nastoji se da ciljevi budu jasni, sredstva se koordiniraju sa ciljevima, i povratna sprega sa akcijom je neposredna i jednoznačna. U takvoj situaciji osoba ima jak osećaj kontrole — ili lične motivisanosti — pa ipak, što je paradoksalno, uključivanje ega je slabo ili nikakvo, tako da se doživljava osećaj sopstvene transcendencije, ponekad osećaj jedinstva sa okolinom. Izgleda kao da je proticanje vremena poremećeno; neki događaji kao da traju nesrazmerno dugo, ali časovi, uopšte uzv, prolaze u minutima.

Najbliže poredenje stvarnog života sa doživljavanjem akcione igre verovatno je borba ili izvršenje nekog izuzetno hrabrog dela. Duš i telo su stopljeni u jedinstven, „veličanstveni“ cilj, lišeni svega drugog. Radost koja se doživljava, zajednička je simuliranom i stvarnim akcijama. Beskonačno vezanje i dresiranje vojnika upravljenju je ka postizanju tog „stapanja svesti“ u trenutku kad se suoči sa neprijateljem. Još nije utvrđeno da li to zamenjuje potrebu mladih za nasilničkim delovanjem.

## Programiranje na strampucima

Kada su se pojavili 1975. godine, kućni računari (i veći deo školskih mikračunara) mogli su da se uporede sa kristalnim detektorima koje su u ranijim generacijama konstruisali mnogi daci (ali vrlo malo devojčaka). To je bio hobi u kome su teškoće konstruisanja bile sastavni deo privlačnosti. Ali računarski jezici su se usavršavali sa usavršavanjem računara i verovatno je da u vrlo bliskoj budućnosti nećemo morati razumeti računarski jezik da bismo mogli programirati računare — kao što danas ne moramo biti u stanju da podesimo oscilator

da bismo mogli da izaberemo radio-stanicu. Za nekoliko godina, računarski jezici, kao što su bejzik, algol, fortran i kobil biće zastareli za normalno programiranje i biće samo od akademskog značaja. Sa porastom kapaciteta računara i inteligencije softvera, detaljno programiranje će postajati sve manje nužno. Već postojeći programi, kao što je „Poslednji“, koji sami pišu programe za korisnika, i nagoveštavajući vreme kada nam neće biti potreban nijedan kompjuterski jezik, lako je to obećanje u ovom trenutku pomalo optimističko, jasno je da su dani „peeking“ i „poking“ prošli. (Mislim da te dve reči iz programerskog žargona sumiraju sve što je najgore u mikračunarenju. One opisuju postupke programiranja koji su već pedesetih godina smatrani primitivnim, ali za koje i današnji poklonici „kristalnih detektora“ pokazuju najveće interesovanje.) Ono što će biti potrebno, jeste potpuno razumevanje načina na koji radi kompjuterska logika. Bez toga se neće moći koristiti programi kao što je „Poslednji“, koji samo automatizuju pisanje programa.

Uprkos predstojećoj pojavi računara koji će sami sebe programirati, škole nastavljaju da predaju programiranje, odvajajući računara od glavnog nastavnog programa. Svudaci treba da uče kako radi računarska logika, ali mučenje nad bilo kojim posebnim računarskim jezikom prevaziđeno je i treba ga što pre napustiti. Samo neznanat broj dece nastaviće školovanje sa ciljem da se obrazuje kao profesionalni računarski programer, kad jednom dođu na univerzitet ili na neki profesionalni kurs, oni će odbaciti svu veštinu u bejziku ili nekom drugom višem jeziku i počće da uče kako se sa mikroprocesorom direktno saobraća korišćenjem mašinskog koda ili asemblerkog jezika. Temeljno poznavanje računarske logike biće ovim daccima korisnije od učenja nekog posebnog višeg programerskog jezika.

## Fanatici koda

Nekoj deci, posebno inteligentnijoj, programiranje računara može postati neka

vrista droge, čak mnogo više nego akcije igre koje su ih, možda, prvo privukle računaru. Često socijalno nedorasli, ovi fanatici kodu razvijaju sklonost ka računarskoj inteligenciji, hrabreni roditeljima koji smatraju da je njihovo inteligentno dete na putu da postane tehnološka zvezda; oni se povlače iz realnog sveta u svet spavače sobe ispunjen nezgrapnim zastarelim računarskim jezicima. Ta situacija postaje nezdrava kad dete počne da piše programe u potpunosti za sopstveno zadovoljstvo. Privatni svet spavače sobe postaje svetlište u kome je dete bog, a računar pruža detetu i roditelju opravdanje za potpuno iracionalno i štetno ponašanje.

Nepoznavanje računarske materije često dovodi do situacije u kojoj se ta deca iskreno, ali pogrešno, smatraju zvezdama. Nastavnici ne mogu da idu u korak sa detetom kada se ono dublje upusti u mogućnosti veštačke inteligencije i potpuno su nesposobni da procene stvarnu vrednost dečijih dostignuća. Često dolazi do velikih zloupotreba programiranja i dete nije u stanju ili nije voljno da prihvati nikakav praktičan savet u vezi sa svojim radom na programiranju. Za trenutak dete impresionira svoje vršnjake, roditelje i nastavnike složenim programima sa bleštavim ekranima i zvučnim efektima. U stvari, dete je pobeglo iz realnog sveta i navikava se na programiranje u potpunom akademskom vakuumu.

Pod pretpostavkom da ovi daci ostanu dovoljno konformisti da bi zadovoljili uslove prijema, univerziteti obično moraju da ih reprogramiraju pre nego što ih usmere na studij komputera.

Postoji mišljenje da nabavka kućnog računara može ponekad da ima loš efekat na učenje mladog čoveka i čak na izgled za njegovu zapošljavanje. Prema tom mišljenju, davanje računara mladima u oseljitivom periodu njihovog života, u nadi da se time postavlja osnova za uzbuđivju karijeru, može, u stvari, imati suprotan efekat. Da bi na računarima radili kako treba, programeri moraju da misle na jeziku koji koriste, kao što je i za tečan francuski ili nemački jezik potrebno misliti na tim jezicima. Većina jezika koji su danas dostupni za kućne računare zastarela je i pretrpana svim i svacim. Ako se mlada osoba duboko uživa u njih, biće joj otežano učenje novog jezika.

Jedna mogućnost da se ta teškoća izbegne, jeste proširiti pristup računaru. Deca pre treba da uče šta oni mogu da urade, nego kako da ih programiraju. To znači, kad je reč o kući, da roditelji treba da obezbede što svesranije korišćenje računara. Što se škole tiče, računar treba izvući iz muškog naučnog odeljenja i uključiti ga u ceo nastavni program, tako da bude od koristi svim predmetima. On mora postati učilo koje će se normalno koristiti, ravnoopravno sa globusima i videorikorderima.

Nastavnici računarskih predmeta retko stvarno poznaju predmet koji predaju. To je javno izjavila jedna britanska komisija koja je našla da su nedavni ispiti na kompjuterskim studijama A-nivoa otkrili zaprepašujuću nedostatak razumevanja kod učesnika kako se pripremiti za ispitne programe i kako se obezbediti od kvara hardvera.

## Računarske sveznalice

Jedna britanska studija je nastojala da ustanovi od kolike je važnosti kompjutersko predznanje kada učenik ode na univerzitet ili stupi u industriju. Istraživači su prikupili mišljenja iz obe oblasti i otkrili da akademске obrazovne institucije više vole da imaju dake sa zdravom osnovom u matematici i fizici nego u kompjuterskim naukama, jer su daci sa računarskom osnovom verovatno usvojili rade navike. Industrija, sa svoje strane, malo značajno pridaje kvalifikaciji u kompjuterskim naukama. Računarstvo se, uz to, tako brzo menja da prikupljeno znanje zastari pre nego što dak dode do diplome.

Decu opterećuju složeni problemi: ako prihvatimo da je učenje programiranja pogrešan pristup, kako prekalifikovati naše nastavnike da daju šire, opširnije obrazovanje zasnovano na računaru, u kojem bi mašina bila sredstvo a ne cilj. Svaka nada da se nešto brzo može promeniti, nije opravdana. Međutim, ako slegnemo ramenima i odlučimo da iz lošeg posla izvučemo najviše što možemo prihvatajući „kompjuterskih studija“, suočeni smo sa situacijom da nastavnici ne razumeju predmet dovoljno da bi ga mogli uspešno predavati. Kako onda možemo da pomognemo decu?

Potpuno se odgovor mora naći u kući. Roditelji koji su svesni opasnosti izolacije mladih mogu da urade mnogo da sprede pretvaranje svog deteta u fanatika koda. Oni mogu nastojati i na svesranijem odnosu svog deteta prema računaru, iznalazeći primere računara van školskog nastavnog programa. Pri tome se mogu suočiti sa teškim zadatkom. Programiranje je u mnogim školama postalo snobovski predmet, i dečaci koji se pripremaju za upis na univerzitet, sa kompjuterskim naukama visoko na svojoj listi akademskih dostignuća, ubrajaju korišćenje računara (nasuprot njihovom programiranju u „glupavo“ računarenje. Ovaj tip omalovažavanja podmakao je i teško je izaći s njim na kraj, Ipak, dak koji je zadovoljan što koristi „glupavo“ računarenje može biti u boljem položaju da shvati ulogu koju će veštačka inteligencija imati kasnije u životu.

## Mladi mozgovi

Pošto su istaknute opasnosti od podučavanja dece programiranju računara, treba reći i to da stvaralačka mašta i sposobnost učenja, ako su usmereni kako treba, mogu kod tinejdžera da izbace neke odlične programere koji su u stanju da sjajno zaraduju i pre nego što završe školu. Mnoge zemlje imaju sada neku vrstu takmičenja „mladih računarskih mozgova godine“, kada se stvarno sposobnoj deci daje prilika da pokažu šta mogu. Pobjednici na ovim takmičenjima usmerili su se bez izuzetka na određenu primenu računara i dobijaju nagrade jer su razvili programe koji pomažu hendikepiranim osobama, olakšavaju otkrivanje kriminalaca ili su od neke druge direktne koristi za društvo. Neka deca su tako vešta u pisanju programa (najčešće igara) koji privlače drugu decu, da proizvođači mikroročunara kupuju njihove programe po visokoj ceni, odobravajući im i odredeni procent od prodaje. Poznati su mnogi slučajevi 15-godišnjaka koji su zaradili mnogo novaca u svom slobodnom vremenu.

Ta obdarena deca uspeša su, međutim, da postanu „uravnoteženi“ programeri, sposobna da sagledaju potrebu društva za

specifičnim tipom programa. Ta je sposobnost isto tako značajna, ako ne i značajnija, od čiste sposobnosti da se pišu računarski programi. Ta deca mogu biti potpuno apsorbovana u programiranju, ali je malo verovatno da bi tu aktivnost koristila kao psihološki oslonac ili kao odbranu od spoljnog sveta.

Mnogi dečaci i devojke sa zdravim pristupom programiranju, završavaju školu peškom i, namesto da se subotom angažuju u lokalnoj prodavnici poput njihovih roditelja, opredeljuju se za lokalnu softver kompaniju (ili svoju spavaču sobu) da bi nastavili sa razvojem određenog komercijalnog projekta. Treba naglasiti da ta deca predstavljaju neznatnu manjinu i obično spadaju u kategoriju „obdarene dece“. Njihov uspeh počiva na sposobnosti da otkriju takvu primenu računara koja će ili naći tržište ili na neki način pomoći društvu. Njihova sposobnost da stvaraju programe manje je važna i malo je verovatno da će sa napredovanjem u karijeri sami mnogo raditi na fizičkom programiranju. Verovatno je da će oni raditi u tri faze:

1. identifikaciji određenog problema koji traži rešenje
2. analizi problema
3. skiciranju programa koji će dati rešenje.

Svaki detaljan rad na razvoju programa biće prepušten drugim, manje kreativnim programerima.

Zbog ovoga, za većinu dece računarstvo treba da bude sastavni deo njihovog obrazovanja; na kraju tog obrazovanja, obraćajući računaru za pomoć u istorijskom istraživanju ne bi predstavljalo ništa više od odlaska u biblioteku i uzimanja knjige sa police. Deca bi bila upućena u to kako računari rade, šta mašine mogu a šta ne mogu, kao i u najbolji način korišćenja raznih tipova hardvera. To bi bilo idealno stanje i do njega će se vremenom doći. Problem je jedino ako vaše dete treba da završi školu već za sledeće dve ili tri godine.

Mnoge škole koriste računare samo u nastavi matematike i nauka. Mnogo ređe se oni sreću u nastavi engleskog i francuskog jezika, geografije ili istorije, da i ne govorimo o umetnosti i muzici. Postoje za to dva glavna razloga: nedostatak sredstava i nedostatak nastavničke stručnosti. Sa padom cene mikroročunara, nedostatak stručnosti pokazuje se kao najteži problem koji treba savladati.

## Računari i nauka

Dobre strane korišćenja računara u nauci su očigledne. Ispitivanje životnog ciklusa stanovnika nekog jezera predmet je koji se lako uključuje u računarski program. Program Odell Lake koji je razvila MECC (Minnesota Educational Computing Consortium), jedna od najboljih obrazovnih računarskih organizacija u svetu, omogućuje decima da stvore model jezera i ispituju ishrambeni lanac u njemu. Program je deo jednog šireg programa pod nazivom „Razumevanje lanca ishrane“; svaki korisnik može da planira svoje sopstveno jezero i da uvodi razne oblike života u njega. Program trenutno razabire koji su oblici života inkompatibilni — na primer, pastirka i bela riba — i to tome obaveštava učenika. U nastojanju da stvori uravnoteženo jezero, učenik će morati da prihvati neke gubitke u količini bele ribe i da razmotri način kako da reguliše broj snažnijih grabljivica.

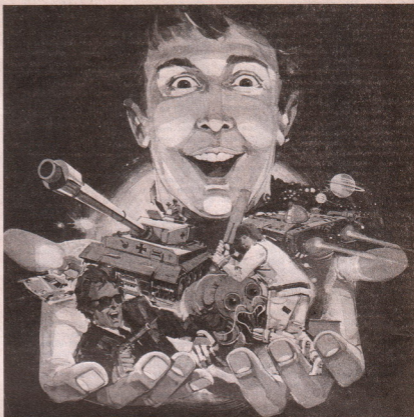
Ova vrsta interakcionog programa stimuliše učenika da traži sve više informacija o ulozi raznih oblika života u vodenju sredini i, u zavisnosti od obimnosti programa, treba da je sposoban da primi mnogo informacija direktno iz računara. Ovaj tip programa pokazao se odličnim u osvajanju i držanju pažnje učenika i praktički oslobađa učenika potrebe za nastavnikom dok stvara svoj akvatični svet. Na kraju projekta, učenik stvara više ili manje uravnoteženo jezero, sa svim važnim informacijama o veličini i zapremini, biljnom, ribljem i životinjskom svetu dobijenim u štampanom obliku.

Programi, kao što je *Quake*, koji takođe potiču sa MECC, omogućuju učeniku da igra ulogu seizmologa i pomažu mu da odredi epicentar zemljotresa posle sekundarnog udarnog talasa. Programi se oslanjaju na raznu dokumentaciju i sastavljaju su tako da utiču i na bogaćenje dačkog rečnika i bolje razumevanje geometrije i srodnih nauka.

Nastava geografije se danas dobrim delom bavi ljudima koji žive u drugim krajevima sveta. Postoje mnogi programi koji pomažu darcima da ubelažavaju u mape kretanje stanovništva i određene socijalne i ekonomske promene, kao što je industrijski razvoj. Sa visokokvalitetnim grafičkim displejem, daci mogu da prikažu razne delove Afrike, na primer, i da ih dopune senčenjem u boji o padavinama, žetvenim prinosima, gustini stanovništva itd. Pri ruci je računar koji omogućuje da se učenik u svakom trenutku prebaci sa mape na informacioni displej i tako doda do što tačnije predstave o jednoj oblasti. Tako moćni programi omogućuje da se izbršu granice koje je obrazovanje veštački postavilo između predmeta. Dok ispituje topografiju jedne oblasti u Zimbabveu, učenik će moći, ako želi, da doda i senčenje koje će pokazivati industrijsku proizvodnju, prosečan dohodak, zdravstvenu statistiku ili geološke informacije.

## Računari i matematika

Daci su u stanju da najpre u matematičkim stvaraju modele problema. Matematika je možda jedna od najapstraktnijih disciplina i pruža velike mogućnosti za dobijanje grafičnih i matematičkih modela. U svojoj aritmetičkoj ulozi, računar je izuzetno koristan kad odgovara na pitanje „šta ako?“, jer otkriva interakcije različitih matematičkih događaja. Koristeći popularni komercijalni program kao što je „*Visicalc*“, dakle može da osnuje imaginarni biznis upotpunjen sa svim fiksnim i proizvodnim troškovima i tome doda podatke o prodaji sa odgovarajućim izdascima. Pošto je taj moćni program potpuno interaktivan, dakle samo treba da menja jedan faktor — broj artikala prodatih za mesec dana, na primer — da bi saznao efekat koji bi takva jedna promena u poslu imala na zaradu i na finije elemente, kao što je udeo fiksnih troškova koji otpada na amortizaciju vozila. Moguće je ovakve modele sačiniti i ručno, ali svaki bi zahtevao mnogo časova rada. Računarski program omogućuje daku da se beskonačno „igra“ sa modelom svog biznisa i tako stiče temeljitije razumevanje vitalnih elemenata koji zajedno proizvode dobit — ili gubitak.



**U svetu mašte: Igre na računaru, koje su, po pravilu, često veoma agresivne, pokatkad zamenjuju potrebu mladih za nasilničkim delovanjem**

Instinktivni osećaj za biznis koji se ovim dobija posle nekoliko nedelja igranja, često je mnogo sigurniji od instinkta koji imaju mnogi profesionalni biznismeni koji nisu imali mogućnosti da izučavaju modele na bazi računara.

Računarske nauke kao odvojeni predmet sada su sasvim jasno definisane u većini škola i pokrivaju oblasti kao što su razumevanje elektronike, programiranje i primena. Vrlo malo škola (za razliku od koledža u daljem obrazovanju) bavi se uticajem računara na društvo.

U drugim naukama — hemiji i fizici, na primer — računari su dobro uvedeni i dostupni su mnogi provereni programi. Van oblasti nauka stvar stoji drukčije.

Britanska organizacija MUSE (Micro-computer Users in Secondary Education — Korisnici mikoročunara u srednjem obrazovanju) trenutno ima spisak od 120 programa namenjenih školama, od kojih je više od 100 orientisano ka matematičkim i naučnim. Postoje mnogi programi za umetničke predmete za osnovce, ali mali broj za decu između 11 i 18 godina.

Muzika i umetnost predmeti su u kojima mikroračunari imaju izuzetno značajnu i posebnu ulogu. Ipak, uprkos mnogim danasnim dostupnim pogodnostima, mali broj škola koristi računare u tim oblastima.

## Istraživački projekti

Svake godine firma Commodore prikazuje kako njeni Pet mikroračunari kontrolišu jedno teško putovanje dece kroz podzemlje Jorkširske močvare u severnoj En-

gleskoj. Uz nešto nadzora, deca unose u program potrebne podatke: rutu, stanje okoline, broj i imena učesnika itd. a na komandnom mestu se, zahvaljujući računaru, veoma pažljivo prati tok kretanja. To je idealan tip primene koji pokazuje da uloga računara nije ograničena na jedan ugaon naučne laboratorije.

Jedna škola u San Dijegu, Kalifornija, beleži na mikroračunaru godišnji tok migracije kitova. Iako se ovo može smatrati čisto naučnom primenom, ona je obuhvatila ništa manje nego 150 daka, koji su se jednog dana, posle mnogo uloženoj truda, ukrcali na brod da bi isplivali u Pacifik i na licu mesta videli da li njihova računarska simulacija uve najduže migracije sisara u svetu, olakšava pronalazanje kitova. Šteta je da veći broj škola ne može da priušti ovaj tip stvaralaštva u rešavanju problema pomoću računara.

Ali računarsstvo se brzo razvija i, mada je situacija trenutno daleko od zadovoljavajuće, za nekoliko godina mogla bi se ipak izmeniti tako da će deca koja uče stolarski zanat smatrati prirodnom da se obrate računaru stolarske radnje da programira ruku robota za delove koji se ponavljaju.

Šta je, međutim, sa decom koja završavaju školu? Neka su možda naučila nešto bezjika, druga su možda predmet pratila detaljnije, dok je najviše njih verovatno gurnuto u grupu koja je imala da sluša usmena objašnjenja i radi praktički nekoliko minuta u nedelji.

Oni koji završavaju školu ozbiljno se interesuju za računare, mogli bi nastaviti da predmet izučavaju na kursu u nekom

koleđžu ili da se direktno zaposle na računarenju. Mnogi koleđži i univerziteti nude kurseve sa računarskom kvalifikacijom na kraju. Studenti koji sa uspehom završe takav kurs, verovatno će biti u stanju da u mnoštvu ponuda izaberu ono što im najviše odgovara. Bilo je poslednjih godina nekih diskusija u novinama o nezaposlenosti koja pogađa čak računarsku industriju (koja se samo često smatra prvenstvenim uzrokom nezaposlenosti). Radi se, međutim, o računarskom osoblju nižih kvalifikacija — kao što su programeri i operatori — dok se nastavlja (i verovatno će se nastaviti) jaka potražnja za inženjerima, sistemskim analitičarima i projektantima.

Interesantna mogućnost koja upravo postaje realna, a pruža se studentu računarsstva, jeste da se vrati u školu i preduje računarsvo. Do sada je ono kao specifičan predmet bilo rezervisano za univerzitetne i koleđžne. Problem je u tome da nastavnici treba da imaju daleko širi pogled na računarsvo od ograničenog pogleda razvijenog pri radu u računarskoj industriji. Jasno je da je obrazovanje očajno osudno u obrazovanom kompjuterskom osoblju, i takva karijera bi se nesumnjivo veoma isplattla.

## Poslovi u računarsvo

U računarskoj industriji postoji nekoliko nivoa hijerarhije ili, mada je moguće napredovati u sistemu, oni koji dolaze sa univerziteta ili sličnih institucija, mogu očekivati da će napredovati brže.

Postoji šest glavnih hijerarhijskih nivoa u kompjuterskoj industriji. Najniži stepen mogao bi se nazvati „kvalifikovanim korisnikom“. Ta kategorija obuhvata radnike koji su naučili da na računarima rade specijalne poslove, kao što su obrada reči ili knjigovodstvo. Često se ovaj kadar regrutuje iz drugih zanimanja — na primer, sekretarijata ili kancelarijske administracije — ali on uključuje i funkcije specifične za računarsku industriju, kao što su operator na terminalu, operator na bušaću kartica i slično. Angažovanje na tim poslovima zahteva osnovne školske kvalifikacije i sposobnost jasnog mišljenja. Znanja, kao što je rad na tastaturi, normalno se uče na samom poslu.

Sedeći viši stepen jeste operator. Operatori kontrolišu rad na računaru, bilo na trakama ili na magnetnim diskovima. Operatori moraju razumeti osnovu na kojoj počiva rad računara i taj tip posla često podrazumeva rad u smeni, pošto mnogi veliki računski centri rade 24 časa dnevno. Poslodavci zahtevaju dobro opšte obrazovanje, i izvesno poznavanje rada računara smatraju određenom prednošću. Postati operator na računaru dobra je odskaćna daska za posao programera.

Programer unosi u računaru uputstva na jeziku koji računaru razume, kao što su kobal ili fortran. Kvalifikacije potrebne da se postane programer obuhvataju jasno metodičko mišljenje i sposobnost koncentrisanja i na najsitnijim detaljima. Vešt programer je sasvim poseban tip osobe, pa iako su za prijem potrebne odgovarajuće školske kvalifikacije, prirodna sposobnost da se radi sa logikom mašine često ima veću težinu. Mnogi programeri čak ulaze u industriju bez formalnih kvalifikacija.

Programiranje je razvijeno u dve grupe: primenjeno programiranje i sistemsko programiranje. Primenjeni programeri pišu programe da bi rešili neki specifični zadatak. Sistemski programeri pišu programe da bi držali u redu računarski sistem — takav je, na primer, program koji ustanovljava greške računara. Primenjeni programer je u situaciji da sreće ljude van računarske sobe — na primer, stranke koji koriste računaru — i da radi sa timom programera na razvijanju programa za određeni zadatak. Sistemski programeri su specijalizovaniji i skloni su ka samostalnom radu. Oni su, verovatno, najbliži direktnom komuniciranju sa inteligencijom mašine.

Posle programiranja, industrija računara veštački povlači granicu iz koje je pristup dozvoljen samo najsjajnijim programerima i najboljim diplomiranim studentima. To je carstvo sistem-analitičara i projektanata.

## Sistemski analitičari

Sistem-analitičari razmatraju probleme i odlučuju kako se oni mogu rešiti pomoću računara. Neka naftna kompanija, na primer, otkrila je nalazište nafte ispod morskog dna. Njeni stručnjaci su izmerili veličnu nalazišta i našli da kvalitet nafte varira; neki delovi naftonosnog polja daju sirovu naftu pogodnu za proizvodnju benzina, dok drugi delovi daju naftu iskoristivu jedino u proizvodnji plastičnih masa. Treba odlučiti da li vredi investirati milione dolara u eksploataciju polja. Odluka će zavistiti od predviđanja prilika na međunarodnom tržištu nafte u toku radnog veka polja (recimo za sledećih 20 godina). Kompanija, isto tako, mora da odluči u kom će delu polja najpre bušiti. Pošto su investicije ogromne, kompanija smatra razumnim da potraži savet od računara i preda mu problem da ga on analizira.

Problem razmatra analitičar. On najpre treba da prikupi informacije o ranijem stanju naftnog tržišta, zatim da se obavesti o potražnji raznih tipova sirove nafte i da konsultuje vodeće ekonomske stručnjake o njihovom sagledavanju budućnosti različitih tržišta u raznim krajevima sveta. Prikupljena su i sva dostupna merenja sa naftonosnog polja da bi se uradio računarski model koji će omogućiti simuliranje raznih rešenja za bušenje i eksploataciju polja. Analitičar, sa ekonomistima i projektantima programa, oblikuje potrebne računske programe. Konačni rezultat će se verovatno sastojati od velikog broja programa (koji, možda uključuju grafičku predstavu strukture polja), koji pažljivim oblikovanjem i dedukcijom mogu pružiti logične zaključke o mogućnostima eksploatacije polja. Izvršni organi naftne kompanije biće u stanju da se na ovom modelu igraju pitanja „šta ako?“, otkrivajući kako bi razne odluke o ceni, tehnikama rafinisanja i tržišnim pristupima uticale na čitavu stvar. Dolazeći tako do svih informacija koje su im potrebne, oni će moći da donesu konačnu odluku o najboljem načinu eksploataisanja svog naftonosnog polja.

Ovo je uprošćen primer. Stvarni projekt možda zahtevati nekoliko godina rada, angažovanje velikog računarskog tima i zasnovano na detaljnim računarskim predviđanjima produkcije polja i perspektiva svetske privrede. Iz primera se, ipak, može videti da je uloga analitičara i projektanta programa vitalna, jer računari mogu da rade samo u okviru unapred propisanog metoda i na osnovu primljenih informacija. Analitičar je odlučujući faktor u dobrom korišćenju ra-

čunaru, kada računaru stvarno postaje dopuna u produženje čovekovog mozga.

Ima i nekih drugih važnih poslova u industriji računara, mada ih je malo koji se tako visoko cene kao posao analitičara. Izuzetak je, možda, projektovanje hardvera. Posetite dobri izgledi za elektronske inženjere svih nivoa, od centara za popravku obuputeva do istraživačkih odeljenja, ali oblasti razvoja proizvođača čistog istraživanja otvorene su samo onima sa najvišim kvalifikacijama u elektronskom inženjerstvu.

Mnogi analitičari i projektanti mašina softvera napreduju do rukovodećih i konsultantskih položaja, ali te titule često znače samo to da pojednac ima više vlasti, vrlo često radeći u sopstvenoj režiji. Radni sadržaj posla često nije promenjen.

Ako industrija računara cveta, mnoge druge industrije podleću recesiji; jedan OECD izveštaj objavljen krajem 1982. ukazuje na to da će i nekoliko sledećih godina ta nevolja prisliskati većinu zapadnih zemalja — sa značajnim izuzetkom SAD i Japana.

## Mikroračunarski centri

Neke vlade, posebno britanska, smatraju da mikroelektronika može pomoći da se reše neki kratkoročni problemi nezaposlenosti. Po jednoj shemi koju je ustanovila britanska vlada da bi omogućila sticanje izvesnog radnog iskustva velikom broju nezaposlenih mladih (pre nego postanu nezaposlili), nudi se mesto u 40 mikroračunarskih centara u Britaniji. U tim centrima, mladi koji su završili školu, mogu pohađati kurseve o raznim aspektima mikroračunarenja, primajući uz to malu novčanu naknadu. Ta shema pruža dvostruku korist: onima koji su prošli kroz školu bez dodira sa računarom (jer su napustili školu pre nego što je računaru stigao, ili nisu bili „zabrani“) da ga koriste) pruža mogućnost da se s njim upoznaju, a poboljšava i njihove izgleda da nadu neki posao.

Može se postaviti pitanje, kako osposobljavanje jednog završenog daka da dobije zaposlenje koje počiva na računaru (čija je uloga verovatno da zameni nekoliko manuelnih radnika), pomaže problemu nezaposlenosti. Dvosmisleno o ulozi računara u zapošljavanju podstaknuta je izveštajima koji predviđaju masovnu nezaposlenost kao direktnu posledicu računarske automatizacije. Jedan nedavni izveštaj kazuje da će nezaposlenost izazvana automatizacijom dostići 16% u Zapadnoj Evropi. Za one koji završavaju školu bez ikakve računarske pismenosti, izgledi da se zaposle zaista su ništavni. Mnogi stručnjaci smatraju da mikrotehnologija per povećava jaz između onih koji imaju i onih koji nemaju, nego što dovodi do veće jednakosti, kao što neki futurolozi predviđaju.

Bez obzira na svoje sposobnosti i interese, deca koji se približavaju dobu kada završavaju školu, zaslužuju našu simpatije. U obrazovanju, u suštini jednom informaciono-prelaznom procesu, ona su se našla zahvaćena informacionom revolucijom na koju se mnogo lakše adaptiraju od svojih učitelja. Po završetku obrazovanja, oni su suočavaju sa jednim svetom koji se i sam u prelaznom periodu između pred- i posle-računarskog doba našao u meteu. Ona moraju biti spremna da prežive u tom svetu, a jedna od bitnih kvalifikacija te spremnosti jeste sposobnost da vladaju kompjuterskom inteligencijom i da je koriste.

# Računari iz drugog ugla nemojte plakati za eplom!

Najzamršenija i najzбудljivija kompjuterska igra koja se trenutno igra zove se kupovina računara. Ona je obavijena misterijom, opasnostima, intrigama i prisnim pozivima. Kao i u svakoj pustolovnoj igri, glavni igrači — bez obzira da li traže Izgubljeno Blago, Malteškog Sokola ili Pravi Kompjuter — pitaju se: Kome mogu da verujem? To je pitanje o kome zaista vredi razmisliti.

## Kupiti ili umreti

Pre svega, ne verujte nikome ko tvrdi da vam je računar potreban *odmah*. Fanatici kompjutera su usvojili nešto žara — i neke od izreka — religioznih zanesenjaka. „Računajte! Kraj je blizu!”

U stvari, ne verujte na žurbom. Živite ste dugo bez računara i moći ćete sasvim fino još malo. Jedno od prvih pravila u kupovini kućnog računara jeste da se ne žuri. Nema potrebe da kupite računar još ove nedelje. Kupovina kućnog računara zahteva dosta strpljenja, izdržljivosti — i vremena.

Ne verujte oglasima za računare. Oglasašvanje se pretvorilo od nenamerne samoprodaje iz sedamdesetih godina u nadrealizam u osamdesetim. Jedan tekući oglas koristi slogan: „Daleko od očiju — daleko od pameti!”

Ne verujte, takođe, ni časopisima. Neki specijalizovani časopisi za računare svakim danom postaju sve sličniji haj-faj magazini. Jeste li ikad pročitali loš izveštaj o velikom proizvođaču (čitaj — oglašivaču) stereofonskih uređaja u nekom od haj-faj časopisa? Prikazi su toliko laskavi i toliko puni obzira da su potpuno beskorisni.

Kao i haj-faj časopisi, i magazini za računare zavise od prihoda od oglasa za iste proizvode o kojima objavljuju prikaze. Ako su suviše često iskreni — dovidenja oglašivači i, na kraju, zbogom časopis! „Popular Computing” je odbio da objavi jedan moj prilog o tome zašto je Apple ili slaba vrednost. Razlog? Prema izjavi urednika: „Desilo se da ja volim Apple Kompjuter korporaciju”. Mislim da je dozvoljeno zamisliti da on, takođe, voli i mnogo strana Apple-ovih oglasa u bobi svakog meseca.

## Vapaj u divljini

Dakle, kome verovati? Prijateljima koji imaju računare? Ne obavezno! Ljubitelji računara su obično skloni da se podaju onoj vrsti kompjutera koji poseduju bez obzira da li stvarno vredi ili ne. Sem toga, nakon što potroše ne baš zanemarljivu



sumu novca da bi kupili računar, malo ljudi će reći: „Pogrešio sam. Trebalo je da uzmem nešto drugo”. Više ljudi se kune u svoj „ševrolet” nego što ga proklinje.

I dok smo kod teme kola: prodavcima računara možete verovati otprilike isto toliko koliko i prodavcima polovnih automobila. Ponekad razmišljam da su se svi nezaposleni prodavci automobila prijavili i odmah dobili zaposlenje u prodavnicama računara od jedne do druge obale (misli se na Atlantski i Tihi okean).

Ima, doduše, i nekoliko dobrih prodavaca računara. Oni znaju isto tako dobro kao i

ja da većina daje lošu reputaciju manjini. Budimo fer — isuviše stvari se dešava previše brzo da bi se bilo ko održao, a da se ne zapusti. Ja se, možda, ne mogu zapustiti, ali ja ne moram da čekam na mušterije osam časova na dan.

Zato, za sve na svetu, ne uzimajte moju reč ni za šta. Ja sam samo glas koji vapi u divljini (Los Angeles). Misli iznesene u ovom članku nisu ništa drugo nego moja subjektivna, pristrasna, izuzetno lična mišljenja.

Kome, dakle, verovati? Ako želite da imate pravi računar za pravu cijenu —



*Jugoslovenski ljubitelji računara okrenuti su, uglavnom, engleskom i nemačkom tržištu. U domovini kućnih računara, Americi, na čitav fenomen i, rekli bismo biznis, gleda se sa znatno manje nervoze. Koje je njihovo viđenje kućnih računara i pojedinih zvezda na računarskom nebu? Članak Pitera Vilijamsa (Peter Williams) o najpopularnijim modelima kućnih računara „sa druge strane velike bare“ radikalan je čak i za američke prilike, ali može da pomogne i našim ljubiteljima računara da raščiste sa nekim predrasudama. U svom članku Vilijams propušta kroz sito i rešetko četrdesetak modela računara — mi smo izbor suzili samo na one koji su poznati i u našoj zemlji.*

imate poverenja u sebi! Uronite, učite koliko možete, razgovarajte sa što je moguće više sveta i na kraju ćete razaznati ono što će vam biti korisno.

#### TIMEX SINCLAIR 1000

Timex Sinclair je poznat kao prvi svetski računar za jednokratnu upotrebu. Kupite jedan za oko \$49,95, odnesite ga kući, koristite nekoliko dana i odlučite: (A) računari vam se dopadaju i želite da (B) rađujete neki bolji ili (B) računari vam se ne dopadaju i ne želite da imate više ništa sa njima. U oba slučaja oslobodite se Timex Sinclaira sa otprilike isto toliko ceremonije kao kada praznite mišolovku.

#### COMMODORE

VIC 20 i Commodore 64 predstavljaju izuzetnu vrednost među jeftinim kućnim računarima. Za manje od 100 dolara VIC 20 je mnogo bolji računar za jednokratnu upotrebu nego Timex Sinclair. Za oko 200 dolara Commodore 64 ima skoro sve hardverske uređaje kao i Apple II, koji košta 1395 dolara. U grupi kućnih računara sa cenom ispod 500 dolara Commodore 64 je jasni pobednik.

#### TEXAS INSTRUMENTS 99/4A

Gde li su samo našli ime kao što je 99/4A? Izgleda da je čak i Bil Kozbi (Bill Cosby) bio nesposoban da spase ovu mašinu. Texas Instruments ju je početkom ove godine (1983) prodavao uz gubitak, nadajući se da će razliku nadoknaditi prodajom programa. Plan nije proradio. Iskazan je veliki gubitak. Akcije su pale. Ova situacija nije mnogo iznenadila one koji prate računare. Texas Instruments je obeshabrio treće druge da rade softver za 99/4A. To je jednako pametno kao da obeshabrite Standard Oil da proizvodi benzin za vaš automobil. Ako T.I. ne promeni tu politiku, i to brzo, neće još dugo biti na tržištu kućnih računara.

#### ATARI

Uzduž i popreko ATARI zemlje, ljudi hodaju okolo sa staklinsim pogledom, mrmļajući „Šta se to dogodilo?“ Za trenutak je izgledalo da je Atari zaoposario tržištem kućnih računara. Koštao je manje od Apple-a. Imao je „pakmens“. Imao je primamljiv slogan („Jeste li igrali Atari de-

nas?“) Šta se desilo? Desila se konkurencija. Coleco je ponudio bolje igre — Commodore je ponudio jeftiniji računar. Ako Atari ne izvuče nešto maglično iz svog šešira, i to uskoro, može završiti na sve većoj listi kalifornijskih ugroženih vrsta.

#### APPLE II

Apple računar ne predstavlja dobru vrednost već čitave dve godine. I Apple II i Apple III su preskupi, u poređenju sa slično opremljenim računarima, još od 1980. Apple II (Apple kaže da „e“ znači poboljšani-enhanced) ja kažem da to znači skup — expensive) dobio je još nekoliko znakova, mala slova, malo više memorije i priključak za štampač; smanjen mu je broj čipova (smanjenje ne predstavlja vrednost za kupca, ali čini računar jeftinijim za proizvodnju) i povećana cena. Čak i ja, koji sam imao malo poverenja u Apple, mislio sam da će ponuditi više i naplaćivati manje.

Glavni razlog zašto ljudi kupuju Apple II je softver i periferijska oprema koju proizvode drugi proizvođači. Većina toga potiče iz kasnih sedamdesetih godina kada je Apple bio jedini računar u koji na tržištu. Ali Apple gubi te prednosti. Postoji više lepih računara koji rade sa softverom Apple-a.

U stvari, ako pogledate na spisak od 16 000 programa koji stoje na raspolaganju za Apple, i koji je formata telefonskog imenika, videćete da je njihova velika većina bezvredna. Izgledaju kao prilici na konkursu za upis u višu školu. („imate tri sata na raspolaganju da napišete program za računar. Počnite.“)

#### RADIO SHACK

Radio Shack je bila jedna od prvih kompanija (druge su Commodore i Apple) koja je ponudila lične kućne računare. Nakon nekoliko dosadnih godina u kojima je izgledao da se Radio Shack, kao i Apple, odмара na lovorikama (Apple se više ne odmara — on se ucvrjlo) izišao je sa nekoliko računara koji su dobri i jednim koji je vrlo dobar.

Dobra vest kod Radio Shacka je model 100. Možda ne biste želeli da vodite poslovanje svoje kompanije na njemu, ali biste mogli poželeti da ga stavite u svoju ručnu torbu. Trastatura je pune veličine i ima veliku preglednost. Ekran je tečni kristal, kao kod džepnih računara i ispisuje, osam redova od po 40 karaktera. Nije ono što bi nazvali mašina za obradu teksta — više je mašina za zapis teksta. Sve što pohranite u model 100 može se prenети na drugi računar za naknadno izdavanje ili reviziju. Podaci se smeštaju u vrstu RAM koja nikad ne zaboravlja. Nakon što se podaci prenesu ili iskoriste, memorija se može obrisati za naredne zadatke.

#### IBM PERSONAL COMPUTER

IBM je sagradio izvanredan kućni računar. Umesto da „sklapa“ mali računar korišćenjem malo od ove IBM mašine a malo od one, IBM je sastavio grupu inženjera, dizajnera, programera i, da, nekoliko zaljubljenika u kućne računare, i izdvojio ih sa zadatkom da projektuju i izrade IBM PC.

Među proizvođačima malih računara postojalo je strah da će IBM progutati tržište kućnih računara. Umesto toga, izgleda da je IBM stvorio čitavu novu industriju: Svako je dobio deo učešća: proizvođači periferijske opreme, prodavci programa, firme za prodaju putem pošte i preprodavci. IBM ispoljava neobičan stepen susretljivosti prema ljudima koji izrađuju stvari koje se ugrađuju u, na ili priključuju za IBM PC. Pre ovoga, politika IBM-a je bila: ako to nije napravio IBM, ne treba ni da se koristi za IBM. Sada je IBM čak objavio shemu PC-a i nudi svu moguću pomoć.

Čak, kao čudo nad čudima, IBM prodaje proizvod druge kompanije pod imenom IBM. To je jedan Epson, proizveden u Japanu, na koji se dodaje pločica IBM-a. To je toliko radikalno kao da Steinvay stavi svoje ime na klavire proizvedene u Japanu. S druge strane, IBM/Epson izgleda malo star uz nove Epson tastature.

Tastatura je jedinstvena. Neki ljudi je vole, a neki mrze. IBM tastatura je dizajnirana sa onim što je poznato pod nazivom „audio taktilni feedback“. Kada pritisnete taster, vi čujete i osećate klik.

Drugi čudan element tastature je smeštaj SHIFT tipke. Godinama, proizvođači malih kompjutera, želeći da s ponosom istaknu kvalitet svojih tastatura, navodili su da su „selectric-tipa“. Tipke su bile na istim pozicijama kao kod IBM Selectric-a. IBM, iz razloga poznatih jedino stvaraocu ili stvaraocima, nije siedo selectric format... Ali, Keytronics je već predstavio tastaturu koja se može priključiti na IBM i ispravi sve te, um, ekscentričnosti. To je lep stvar sa IBM-om: on će najverovatnije postati lični računar sa najviše podrške u istoriji. U stvari, on to već sada jeste.

Kao i kod Apple II, za sve što želite da radite sa sopstvenim računarom, bez sumnje ćete naći periferijski uređaj ili softver koji će vam to omogućiti. Apple II, ipak, to radi na nivou više škole, a IBM za — odrasle.

Da li biste verovali, čak i pre godinu dana, da će IBM ikada prodavati proizvod koji se zove XT? Zvuči kao ime sportskog automobila ili poboljšanog uređaja za pranje usta. Kako brzo veliki moćni dozvoljavaju da im izmakne ugled kada treba „napraviti“ novac. I tako to ide. Dobre vesti i loše vesti o svetu kućnih računara su da se on brzo menja. Niko ne može biti stalno u toku. Sad uzmete malo slobodnog vremena da izadete napolje i dodirnete robu.

# čekajući QL-a

Računari u izlogu



Mašina na koju se mora računati: QL

Nikada, verovatno, nećemo saznati šta je Klajva Sinklera navelo da istripi pred rudu sa svojim novim računarom — preterano poverenje u svoje razvojne inženjere, pritisak javnosti koja je čitavih četrnaest meseci sa nestrpljenjem iščekivala njegov novi potez, želja da na samom početku godine preduhitri konkurenciju koja je počela da najavljuje nove mašine, ili se već pojavila sa njima. Ili, zapravo, poslovna nesmotrenost — niti je to za čitavu priču previše bitno. Tek, pred nama se, lako je od njegove januarske promocije prošlo više od pet meseci, još uvek nalazi samo desetak istih fotografija i more nagađanja iz engleske štampe o računaru čiji razvoj, po svemu sudeći, neće biti okončan još nekoliko meseci. Zato se ovaj tekst neće previše baviti detaljima računara QL — može se dogoditi da se oni promene pre nego što se na našem izdanju osuši štamparska boja.

## Više „hop“ nego skok

Pošto je na januarskoj promociji zašao među publiku i napunio svoj šešir čekovima na po 400 funti, Klajv Sinkler je prestao da se pojavljuje u javnosti. Za to je imao dobrih razloga — isporuka prvih mašina se toliko oteglja da su čak i Englezi, koji su spremni da se zakunu na ZX znak, počeli da sumnjaju da će se ovaj računari ikada pojaviti na tržištu. Početak isporuke, sa zakašnjenjem od gotovo četiri meseca, malo je umirio računarske duhove, ali nije ni izdaleka otklonio sve dileme. Računar se isporučuje sa jednom ručnom izraslinom na zadnjoj strani kutije, koja treba da otkloni neke konstruktorske zablude iz prvih dana, i nedovršenim operativnim sistemom. Šta, ustalom, misliti o računaru koji nakon

nekoliko meseci upotrebe treba vratiti fabrici na doradu?

Za nagli skok na polju kućnih računara, koji, po mišljenju njegovih tvorca, donosi QL, mogao je da bude primeren samo neki egzotičan termin poput „quantum leap“ iz kvantne fizike (kvantni ili nagli skok — otprilike kao svetionik na sred beskrajnog, pustog okeana). Za ovaj skok trebalo bi najviše da bude zaslužan mikroprocesor 68008 firme Motorola, koji Sinklerovi stručnjaci za propagandu deklariraju kao trideset-dvostručnu mašinu. To je, praktično, bila prva stvar na kojoj je Sinkler bio uhvaćen i koja je nagovestila potonje komplikacije sa QL-om, lako inženjeri ne uspevaju da se slože šta je to mikroprocesor od 32 bita, sasvim je sigurno da 68008 to nije — njegova unutrašnja arhitektura je, istina, u formatu od 32 bita, ali magistrala za podatke ima samo osam linija. To, u suštini, znači da on operiše sa podacima kao i bilo koji osmo-bitni procesor. Na račun Sinklerove pretenzionosti Englezi zbijaju šalu kako QL, zapravo, ima samo dva bita — jedan spređa i jedan pozadi! lako se radi o veoma brzom i veoma moćnom procesoru, od naglog skoka je, izgleda, ostalo samo jedno „hop“.

## Ko mnogo ima . . .

Oko drugog procesora iz QL ima znatno njanje dilema. Radi se o mikrokompjuteru na jednom čipu, sa oznakom Intel 8049, u kojoj je, pored procesora, ugrađena i sistemski i radna memorija! Njemu su povereni trivijalni ali česti zadaci — kontrola tastature, generisanje zvuka i prijem serijskog signala — čime je glavni procesor rasterećen jedne velike i „dosadne“ brige. U svome radu glavni procesor se diste oslanja i na dva specijalno projektovana čipa — jedan kontrolise ekran i memoriju, a drugi mikrodrajvove, časovnik, lokalnu mrežu i emitovanje serijskih signala — koji preuzimaju dosta veliki teret, štedeći snagu „glavnog gazde“ za udarne zadatke. I to je, kad je reč o operativnoj i kontrolnoj logici, sve!

Za sistemsku memoriju u prvobitnoj verziji računara bilo je odvojeno 32 K, ali se pokazalo da je operativnom sistemu na tom prostoru previše tesno. Prvi primerci isporučeni su sa dodatkom od još 16 K ROM-a. U konačnoj verziji, ovaj dodatak će biti ugrađen u računar, čime će se QL po sistemskim bajtovima približiti američkim mašinama, čiji su konstruktori u ovom detalju oduvek bili široke ruke. Pošto će, uz čip više koji nije ukalkulisao u cenu računara, dobiti i neke neplanirane pogodnosti, ovaj previd vlasnicima QL ne bi trebalo previše da kvari raspoloženje.

Prilikom dimenzionisanja radne memorije konstruktori QL su, reklo bi se, bili znatno manje škrti — 128 kilobajta, u koje odjednom staje četrdesetak stranica teksta, mogu se vlasnicima računara „galaksija“ učiniti „dovoljnim za čitav Beograd“. Pokazuje se, međutim, da „ko mnogo ima mnogo i troši“, da bi razvio svoju punu snagu, operativnom sistemu QL-a nije dovoljan postojeći RAM i moraće, po svemu sudeći, da sačeka na memorijsko proširenje od 500 kilobajta! lako se konstruktoria QL može uputiti zamerka za neusklađenost mašinskog i programskog dela računara, potreba za obilnom memorijom pre predstavlja vrlinu nego nedostatak. To je, u suštini, jedini način da se ostvari novi kvalitet.

## Ipak nešto drugo

O tastaturi računara QL nije se, praktično, znalo ništa sve dok prve mašine nisu isporučene kupcima. Stručnjaci za propagandu su, nedorečenim podacima, vešto podgrevali utisak da se Sinkler ovoga puta opredelio za mehaničke tastere i profesionalne performanse. Pokazalo se, međutim, da je i ova tastatura zasnovana na principu membrana (!), ali i da je to urađeno dovoljno solidno za profesionalnu primenu računara — brz i tačan unos podataka u mašinu. Prilikom nedavne posete „Galaksiji“ — Delko Zagorelec se žalio da je tastatura previše meka, osetljiva i „neposlušna“.

**Januarska prašina nad računarem QL je već odavno pala, ali se još uvek ne vidi ništa. Mikroprocesor sa tridesetdvojitnom arhitekturom, 128 kilobajta radne memorije, novi bejzik koji sa arhaičnim mikrosoftovskim varijantama ima zajedničko samo ime, novi operativni sistem koji pretenduje da postane standard za tridesetdvojitne procesore, dva ugrađena mikrodrajva, četiri besplatna programa i ostale pogodnosti za svega 400 funti dovele su u iskušenje čak i najbolje poznavaoce računara, koji počeli javno da se odriču svojih skupocenih računarskih sistema. Jutro je, međutim, pametnije od večeri — vreme je počelo da pokazuje da je u QL ugrađeno više lepih želja njegovih tvoraca nego čipova i sistemskih bajtova.**



Tačka na „i“ ili kamen oko vrata: Ugrađivanje mikrodrajva u sam računar predstavlja hrabar i rizičan potez koji se još uvek nije srećno završio

dok su kritičari časopisa „Your Computer“ došli do zaključka da se na njog radi sa lakomom. U svakom slučaju, na njoj se, uz sve standardne znake i široku razmaknicu, nalazi pet tastera opšte namene kojima se mogu dodeliti proizvoljne funkcije. Naredbe se, kao i na svim ozbiljnim računarima, ne unose preko tastera sa dodeljenim funkcijama (tzv. keyword), nego se kucaju slovo po slovo, kao i bilo koja druga reč.

## Fina grafika . . .

Najočiglednija iznenađenja, međutim, vezana su za ekran QL-a. Zamisljen veoma ambiciozno, on je organizovan tako da vlasticima nove Sinklerove mašine obezbedi pogodnosti znatno skupljih sistema. Pored fine grafike 256×256 tačaka sa osam boja, QL može da radi sa ultrafinom grafikom 512×256 tačaka u četiri boje. Grafički modaliteti se postavljaju automatski — u zavisnosti od toga da li se nakon uključivanja računara odbere rad sa televizorom ili radi sa monitorom. Korišćenjem pogodnosti za crtanje tačkama — vertikalnih i horizontalnih pruga i velikih i malih tačaka — na ekranu se može proizvesti 255 različitih boja! Boje i oblici slova su, kažu, veoma stabilni u oba grafička modaliteta — ni nalik lelujanju kod računara Spectrum.

Ekran je — prvo u nizu iznenađenja — podeljen na tri dela. Dno ekrana — zelena slova na crnoj pozadini — rezervisano je za četiri programske linije. Leva polovina ekrana — bela pozadina sa crvenim slovima — predviđena je za listing programa, a

### Procesor

Motorola 68008 na 7,5 MHz (unutrašnja arhitektura 32 bita, magistrala za podatke 8 bita, direktno adresira 1 megabajt radne memorije)

### Sistemska memorija

32+16 K sa operativnim sistemom QDOS i superbejzikom — može se proširiti do 64 K

### Radna memorija

128 K od čega je 32 K zauzeto za ekran — može se proširiti do 640 K

### Operativni sistem

QDOS sa sledećim prednostima: multiprogramski rad, sistematizovanje zadataka po prioritetu, prozori na ekranu, rad sa perifernom opremom nezavistan od vrste uređaja

### Jezik

Superbejzik sa sledećim prednostima: strukturirano programiranje, jednostavno uvođenje novih naredbi (uključujući i sintaksu), brzina izvršenja nezavisna od dužine programa, jednostavno pozivanje mašinskih programa, jednake pogodnosti za reči i nizove, ekranski editor

### Ekran

Ultrafina grafika na crno-belom ili monitoru u boji u dva modaliteta — 512×256 (četiri boje) i 256×256 (osam boja) — i 25 redova

teksta sa do 85 (monitor), odnosno 40 do 60 znakova u svakom redu (TV)

### Tastatura

QWERTY punog formata sa 65 tastera, širokom razmaknicom, levim i desnim shift tasterom, četiri tastera za kontrolu kursora i pet tastera opšte namene

### Mikrodrajv

Dva ugrađena mikrodrajva minimalnog kapaciteta 2×100 K, prosečnim vremenom pristupa podatku od 3,5 sekundi i brzinom učitavanja od 15 K u sekundi

### Proširenje

Pored priključaka za napajanje, TV i monitor, obezbeđeno je i devet portova za perifernu opremu: radna memorija, mikrodrajv, sistemska memorija, serijski interfejs (2) lokalna mreža (2) i palice za igru (2)

### Serijski interfejs

Dva standardna RS232C interfejsa za printer, modem itd. sa podešivom brzinom prenosa između 75 i 19200 bita u sekundi za dvostruki dvosmerni prenos

### Dimenzije

138×472 mm

### Masa

1388 G

### Cena

399 funti

desna — crvena pozadina sa belim slovima — za njegovo izvršavanje. Unošenje programa i naknadnih izmena je olakšano ekranskim editorom, a testiranje dovedeno do savršenstva — dok se preko leve strane ekrana kreće listing, na desnoj se može pratiti njegovo dejstvo. Na žalost, ove pogodnosti nisu izbušene do kraja. Kada korisnik pokuša da dvaput izlista neki program — a jedan program se u toku razvoja lista bezbroj puta — u belom prozoru dolazi do „zagušenja“. Osim toga, podela ekrana na tri dela predstavlja ubijačiji način rada QL-a, u „monitoriskom“ modalitetu i ne može se ni na koji način preskočiti osim direktnim komandama. Da bi ekran dobio ubijačiji izgled, treba otkucati najmanje dve naredbe, što, očigledno, nije previše elegantno.

## . . . i softverske smicalice

Mogućnost da se na ekran postavi više prozora, pri čemu se u svakom mogu dešavati različite stvari, predstavlja jednu od najudarnijih pogodnosti novog Sinklerove operativnog sistema. Časopis „Your Spectrum“ je, međutim, naziva običnom šalom i softverskom — smicalicom. Umesto „čvrstog“ sistema, kakav, na primer, koriste Apple, Digital Research ili Microsoft, tvorci QL-a su, naprosto, po ekranu iscrtili okvire. Za svaki okvir, doduše, obezbeđena je programska kontrola sa svim parametrima

kao i za čitav ekran, ali ona ne deluje pouzdano: kada ne može da stane u jedan, tekst „ladno“ prelazi i u drugi prozor — brisanjem jednog prozora lako se može obrisati i onaj do njega. Slični efekti, zaključuje „Your Spectrum“, mogu se postići i na bilo kom drugom računaru, uključujući i četiri puta jeftiniji Spectrum.

## „Dva časa nasamo“

Ovim se sličnosti između QL i Spectruma ni iz daleka ne iscrpljuju — njihovi operativni sistemi i njihovi bejzici imaju, očigledno, zajedničke temelje, koji bi upornog računarskog „arheologa“ verovatno odveli sve do računara ZX 80. Nije, razume se, nimalo jednostavno suditi o programskom jeziku jednog računara na osnovu spiska naredbi i funkcija i šturih tudih utisaka tipa „dva časa nasamo sa QL-om“. Programski jezik QL-a, nazvan superbejzik, pisan je očigledno sa velikim ambicijama i snažnom željom da nadmaši sve svoje prethodnike. Mada pomalo pati od „kambrijskog akademizma“ — „dugačke, nezgrane naredbe i šture, jezgrovite poruke“ — superbejzik, izgleda, ima sve odlike zrelog programskog jezika: višelinejske funkcije, procedure sa lokalnim promenljivim, promenjive se daju: imenima, formatirano listanje i ekranski editor Arhaične naredbe tipa GOTO i GOSUB izvršene su, izgleda, samo iz nostalgije, a FOR-NEXT tekst liče na klasiku samo po imenu.

Da bi se otkrile dobre i slabe strane superbezjeka potrebno je barem nekoliko meseci rada sa QL-om, ali je već sada jasno da mu nedostaje malo više logike i finoc u detaljima. Uočena su, takođe, i neka veoma uspešna i moćna rešenja. Procedure se mogu izvršavati i potpuno nezavisno — izvan svog matičnog programa. Pošto svaka procedura može imati i svoje parametre, uvođenje novih naredbi u superbeziček predstavlja pravu dečiju pesmicu. Fleksibilnost, očigledno, čini jednu od jačih strana ovog jezika. Na osnovu hardverskih specifikacija moglo bi se pomisliti da se programi izvršavaju „kosmičkom brzinom“. Prvi utisak, međutim, govore da je QL samo dvaput brži od Spectruma, a u FOR-NEXT petljama ni toliko.

## Biser u školjci...

Konstruktorima QL-a, po svemu sudeći, najviše problema zadaje operativni sistem nazvan QDOS. Iako je prošlo već više od pet meseci od najave računara, on ni do danas nije završen. niti ima izgleda da se to dogodi u nekoliko narednih meseci. Zamisljen možda previše pretenciozno — da postane standard za Motorolne tridesetdvo-bitne procesore — QDOS trenutno najviše hramlje upravo u onim osobinama koje treba da predstavljaju njegovu najjaču udarnu snagu: promena parametra u toku izvršavanja programa, na primer vrednosti neke sistemske promenljive, paralelan rad više programa u memoriji i na ekranu i udoban i brz rad sa mikrodajivovima.

Nedorečenost, verovatno, predstavlja osnovni razlog koji je naveo tvorca QL-a da operativni sistem zatvora za korisnika. QDOS se, na žalost, ne može koristiti direktno u sopstvenim programima — za sada se iz superbezjeka ne može prići nijednoj od njegovih udarnih pogodnosti. Deo za rad sa diskovima ima dovoljno snage za dva mikrodajiva sa po desetak fajlova na svakom, ali se može pokazati potpuno nemoćnim kada se jednog dana na QL priključi tvrdi disk.

Rad sa tastaturom, ekranom i perifernim uređajima zasnovan je na sistemu „struja“ i „kanala“ koji je razrađen na računaru ZX Spectrum (vidi str. 69—71). Ovaj sistem omogućuje veoma jednostavnu i elegantnu softversku vezu sa perifernim jedinicama — takvu da na priključni uređaj QL gleda kao na organski deo svoje sopstvene mašine. Povezivanje određene struje u mašini sa određenim kanalom (periferni uređaj) može se izvršiti sa čitavim spisakom parametara koji se, na primer, odnose na vrstu štampača i modalitet rada. Otvaranjem struje u memoriji se automatski rezerviše prostor za 16 K sistemskog programa za pogon priključenog uređaja. Sistem je, čak, u stanju da prepozna da li je periferni uređaj za sukcesivni (štampači, tastatura, mreža) ili slučajni pristupom (mikrodajiv, tvrdi disk) podacima. Vreme će, možda, pokazati da su struje i kanali uobitljivi QL-ovo oruđe od multiprogramskog rada.

## ... a školjka na dnu mora

Mogućnost paralelnog rada više programa u memoriji — dok čekate da na printeru

### Rečnik QL-a

Procedure		MRUN		PAPER		INT		IF THEN ELSE
ARC	OPEN	NEW		RECOL		LN		DEFINE
ELLIPSE	OPEN-IN	RETRY		SCROLL		LOG 10		PROCEDURE
LINE	OPEN-NEW	RUN		STRIP		SIN		RETURN
POINT	POKE	SAVE		UNDER		SORT		DATA
SCALE	POKE-W	SBYTES		WINDOW		TAN		END FOR
BEEP	POKE-L	SEXEC		Funkcije		PI		END IF
INPUT	RANDOMISE	STOP		BEEPING		CHR\$		END REPEAT
PAUSE	CLEAR	AT		INKEY\$		CODE		END DEFINE
PRINT	CONTINUE	BLOCK		KEYWORD		PEEK		END SELECT
BAUD	EXEC	BORDER		ABS		PEEK-W		EXIT
CLOSE	EXEC-W	CLS		ACOS		PEEK-L		NEXT
COPY	LBYTES	CSIZE		ACOT		RND		
COPY-N	LIST	CURSOR		ASIN				
DELETE	LOAD	FLASH		ATAN				
DIR	LRUN	INK		COS				
FORMAT	MERGE	OVER		COT				
		PAN		EXP				

izliza vaš najnoviji program, računar vas može zabavljati lepom partijom šaha — sa svim tokovima na ekranu trebalo bi da predstavlja biser QDOS operativnog sistema. Nevolja je, jedino, u tome što je ta školjka još uvek duboko na dnu mora. Paralelni rad, uostalom, nije neobičan čak ni za najefitniji računar. Svaki operativni sistem radi paralelno na različitim zadacima — očitavanje tastature, prikazivanje znaka na ekranu, generisanje zvuka — ali je obično toliko spor da izgleda kao da se ovi poslovi odvijaju jedan za drugim. Oni se, da stvar bude lepša, tako i odvijaju. U čemu je onda štos kod QL-a? On ima dva procesora. Prelazak sa jednog programa na drugi je, osim toga, dovoljno brz da se to ne primeti. Multiprogramski rad značajno povećava kvalitet obrade podataka. Voljeli bismo, zato, da tvorcii QL-a istraju u svojim namerama. Time bi se u kućne računare preselila još jedina pogodnost velikih računarskih sistema.

## Kamen oko vrata

Nijedan hardverski detalj na QL-u nije izazvao toliko protivurečnih reakcija kao mikrodajiv. Dok jedni njihovu ugradnju u sam računar smatraju mudrim, ključnim potezom, drugi su ubeđeni da oni predstavljaju QL-u kamen oko vrata i da će Sinkler kad-tad morati da potraži drugo rešenje. Mnogi su još i sada ubeđeni da će u konačnu verziju QL-a biti ugrađeno nešto sasvim drugo — minidiskovi firme Hitachi. Zagovornici mikrodajivova smatraju ovaj Sinklerov izum dosta elegantnim sredstvom za trajno skladištenje podataka, a protivnici neozbiljnim igračkom koja ni po čemu nije primerena mogućnostima QL-a.

Verovali u mikrodajiv ili ne, pokazuje se da on zasad predstavlja najveću kočnicu razvoju ovog Sinklerovog računara. Da bi se na traci pronašao određeni podatak, računar radi „u prazno“ 3,5 sekunde, a učitavanje jednog od besplatnih programa traje sedamdesetak sekundi — trenutak u poređenju sa kasetom, večnost u poređenju sa disk jedinicom! Preterane sporosti su, u zadnji čas, postali i svesni i tvorcii QL-a, koji grozničavo tražaju za efikasnijim operativnim sistemom. Tražajući za podatkom na traci, operativni sistem ide od jednog do drugog bloka — ako ono što se traži nije u prvom bloku, prelazi se na drugi, treći i tako sve do kraja. Direktni pristup blokovima će, verovatno, ubrzati učitavanje podataka, ali ni to, po svemu sudeći, neće biti dovoljno da se spase postojeća koncepcija.

## Neko to voli drugčije

Na zadnjoj strani QL-a nalazi se nekoliko priključaka — računska mreža, monitor, TV, serijski interfejs, palice za igru, spoljašnji ROM i spoljašnji RAM. Ma kako to zvučalo neverovatno, u mašinu koja je, između ostalog, namenjena i za obradu teksta, nije ugrađeno paralelni interfejs za štampači niti se našlo mesta za port opšte namene Sinkler, koji se, očigledno, upinje iz petinih žila da nametne svoje standarde računarskom svetu, pokazuje, izgleda, veliku odojbnost kada se radi o standardima koje su izmislili drugi.

Uz računar QL isporučuje se i četiri ozbiljna, „profi“ programa koja dolaze iz radionica firme Psion — program za obradu teksta, za obradu podataka, knjigovodstvo i poslovnu grafiku. U zapadnom svetu, gde je odnos prema programima znatno drugačiji nego kod nas, jedan ovakav softverski paket predstavlja sam za sebe dovoljan razlog da se neko odlučuje za kupovinu QL-a. Mada imaju veoma efekatan izgled ekrana, oni su pisani u trenutku kada je operativni sistem QL-a bio u embrionalnom stadijumu i, sasvim sigurno, ne mogu da demonstriraju najvažnije mogućnosti računara. Ne treba, međutim, sumnjati da će vrlo brzo na tržište biti izbačene unapredne verzije.

Performanse računara QL objavljene su šest meseci ranije nego što je trebalo i konstruktori sada imaju velikih problema da održe datu reč. U tome, možda, leži i najveći nedostatak QL-a — od njega se očekuje znatno više nego što može da pruži. A brzopletost, kao i svuda, ima svoju cenu — jedan inače i savremeno i zrelo koncipiran računar izlazi pred svoje kupce sa neizbrušenim bezizlom, nedovršenim operativnim sistemom i nekoliko otvorenih pitanja oko konačne verzije mašine. Isporuča računara je, izgleda, bila isto toliko brzopletka kao i njegova promocija — porod svih nevolja Sinkler sada ima za petama i nezadovoljive vlasnike čije mašine ne rade kako treba.

„Ko igra na oštrici tehnološkog noža, može i da se poseče“ — mudruju Englezi. A Kljavu Sinkleru preostaje da veruje u izreku da se ipak najslabije smeje onaj koji to čini poslednji. Sudeći prema prvoj ocenama da „QL sasvim zavređuje svoju cenu“ i „da će izvršiti snažan uticaj na razvoj kućnih računara, sličan onome koji su ostvarili Apple, IBM PC i Spectrum“, možda će ipak on biti taj.

Jova Regasek

# Elektron

*Ejknornu nije bilo nimalo lako da koncipira kompjuter koji bi se uklopio u liniju koju su činili Atom, BBC A i BBC B: novi računar je morao da bude toliko jeftin da ugrozi modele iz ekonomske klase, toliko moćan da konkurise Commodoru 64 i drugim računarima iz srednje klase, a ipak ne toliko moćan da uzdrma pozicije BBC modela A i B. Svi ovi zahtevi, jasno, nisu mogli da budu ispunjeni, pa je Ejknorn odlučio da žrtvuje BBC A i izbaci na tržište računar koji će u mnogome kopirati BBC, ali koji će biti lišen nekih skupih delova i tako biti na raspolaganju onima koji za kompjuter nisu spremni da izdvoje više od 200 funti. Electron je ispunio očekivanja iz našeg prvog specijalnog izdanja: došao je na sam vrh tabele najtraženijih modela na evropskom tržištu!*

## Dlaka u jajetu

Electron ima 32 Kb ROM-a i isto toliko RAM-a. ROM je, kao i kod BBC B, podeljen na operativni sistem (ekvivalentan sa verzijom 1.20) i bežik (BASIC II, naravno). Umesto bežika, jasno, može da se priključi EPROM sa nekim drugim programskim jezikom (za sada je na raspolaganju Fort), procesor teksta ili nešto treće. I pored ovih očiglednih sličnosti, hardver Electrona je sasvim originalan i nema većih dodirnih tačaka sa BBC-jem. Upotrebjeno je manje komponenti i ostavljeno manje praznih podnožja za proširenja; računar je, tako, postao manji, jeftiniji i, na žalost, sporiji.

Glavna zamerka koju korisnici BBC-ja obično upućuju Electronu je nedostatak teletekst kompatibilnosti. Nije nikakva tragedija što računar nije kompatibilan sa teletekst mrežom, već što mu nedostaje MODE 7 koji većina korisnika BBC B (računajući i autora ovoga teksta) upotrebljava otprilike 90% vremena rada. Zašto je mod 7 važan? Jednostavno zato što računar, dok radimo u njemu, rezerviše samo 1 Kb RAM-a za video-memoriju, pa je 31 Kb slobodan za naš program i podatke. Korisnici Electrona obično rade u modu 6, koji za ekran rezerviše čitavih 8 Kb.

Electron ima generator tonova koji u svemu imitira BBC-jev osim u jednoj „sitnici“: u jednom momentu možemo da generišemo samo jedan ton što, na prvi pogled, značajno smanjuje njegovu vrednost. Čini nam se da zaključke ne treba donositi preuranjeno: treba biti pravi majstor da bi se sinhronizovao rad BBC-jevih kanala i dobila „orkestarska“ muzika. U najvećem broju programa (čak i izvanrednih akcionih igara poput Snappera) i BBC B koristi samo jedan kanal za sintezu tona!

## Razlike bez značaja

Ostale razlike između BBC-ja i Electrona su primetne praktično na prvi pogled: tastatura je, da počnemo od nje, manja i

znatno manje tastera (55 prema 74). Korisnici Electrona su na prvi pogled ostali bez tastera kojima mogu da dodeluju proizvodnje literale. Kažemo na prvi pogled, jer se istovremenim pritiskom na CAPS LOCK i neki od brojeva dobija funkcija ekvivalentna pritisku na neki od BBC-jevih naranđastih tastera. Kao posebna pogodnost, taster BREAK takođe može da dobije dodatne funkcije.

Taster BREAK je jedna od velikih BBC-jevih i Electronovih pogodnosti. Ma šta računar radio u nekom trenutku (izvršavanje bežik ili mašinskog programa, potpuna blokada usled neke greške, traženje programa sa kasete ili diska ili nešto četvrto), pritisak na BREAK ga vraća u normalno stanje i reinicijalizuje bežik. Program koji se nalazio u računaru biva privremeno izgubljen i vraća se tek kada otkucamo OLD. Zašto bismo kucali OLD kada god pritisnemo BREAK? Lakše je, pre početka rada, otkucati KEY 10 OLD CTRL M i taster BREAK će, pored uobičajene funkcije, izvršavati i OLD po reinicijalizaciji računara.

Loša strana Electrona je što praktično nema priključenih interfejsa. Na njegovoj zadnjoj strani su samo prikljucci za kasetofon, televizor i kolor odnosno crno-beli monitor, dok je na donjoj strani samo običan port za ekspanziju. Svi oni koji žele da kupuju periferijske uređaje moraju, dakle, da nabavljaju razne dodatke kojima ćemo posvetiti sledeće redove.

U prodaji je takozvani teletekst adapter koji Electronu dodaje mod 7 sa svim opcijama — izgleda da je teletekst mod podržan programom u ROM-u i da je potreban samo hadver kojim bi se realizovao. Programi za podršku štampača su takođe u ROM-u; dokaz je što su naredbe VDU1 — VDU 3, kod BBC-ja namenjene kontroli štampača, označene kao "Raserved" u uputstvu za upotrebu. Da bismo, dakle, povezali Electron sa štampačem, treba da kupimo Centronics interfejs koji će nas koštati tridesetak funti. Povezivanje sa diskovima je takođe moguće, ali je potrebno kupiti disk interfejs koji se, prema neproverenim informacijama, smešta u posebnu kutiju i povezuje sa portom za ekspanziju. Čak je i za dodavanje palica za igru potrebno dokupiti A D konvertor. Jasno je da svi ovi dodatci

značajno povećavaju cenu Electrona, koja postaje sve približavaja BBC-jevoju.

## Razmislite još jednom

Programska podrška je jedna od jačih strana Ejknornovog novog kompjutera. Praktično svi programi pisani za BBC B (a takvih je, kao što smo rekli, dosta na tržištu) mogu da se p... Electronu i obratno. Jedini problem je mod 7 koji je, zbog mogućnosti pisanja velikih slova, korišćen za „naslovne strane“ mnogih BBC-jevih igara. Izvršavanje ovakvih programa na Electronu je, naravno, moguće ako se lišimo prizora koji obično traje dok se program učitava sa trake. I kod nas su u mod 7 zajednički aranžmani kupovine stranih programa koje ravnopravno i u međusobnoj saradnji nabavljaju vlasnici BBC-ja i Electrona. Vlasnicima Electrona je, osim Acorn Usera i A&B Computing, na raspolaganju i izvanredni časopis Eibug koji izdaje klub Beebug.

Sve u svemu, Ejknorn Electron predstavlja izvanredan izbor. Nije lako naći računar koji košta 200 funti sa moćnim i brzim strukturiranim bežikom, grafikom (640x256, mogućnošću sinteza tonova i eventualno) glasa i visoko profesionalnom tastaturom. Preporučiti bismo, ipak, onima koji razmišljaju o kupovini Electrona da realno procene svoje potrebe za ekspanzijom sistema. Ukoliko shvatite sva ograničenja računara i zaključite da vam ona ne smetaju, kupite Electron što pre. Ako, sa druge strane, želite proširenja, biće svakako bolje da razmotrite mogućnosti nabavke duplo skupljeg BBC B — ne samo što ćete proći jeftinije nego ćete imati manjih problema sa uzvozom. Filozofija kupovine osnovne verzije računara i postepenih proširenja prema potrebama je za strance preporučljiva, ali kod nas vladaju drugi zakoni.

# Acorn Electron

CPU: 6502

Klok: 1,79 MHz

Memorija: 32 K ROM-a

i 32 RAM-a

Ekran: do 32 reda sa

po 80 znakova, 8 boja,

127 definisanih i 255

karaktera koje može

definisati korisnik

Interfejsi: TV,

kompozitni video, RGB

video, kasetofon,

sistematski bas

Jezič: bejzick sa

ugradenim assemblerom

Tastatura: 56

mehaničkih tastera,

opciono unos naredbi

jednim tasterom,

10 tastera za

dodejivanje funkcija

Dokumentacija: Dobrim

izvanredna sa dobrim

opisom operativnog

sistema i obiljem

detalja za

programiranje na

bejziku i mašinskom

jeziku

Dimenzije:

340 x 160 x 65 mm

Cena: 199 funti



tastatura računara krasi jedna od najboljih tastatura koje se uopšte ugrađuju u kućne računare

glavni klok kristal

kristal za kontrolu TV signala zahvaljujući kvarcnoj kontroli ekrana, slika je izvanredno čista i stabilna

TV i izlazni priključak

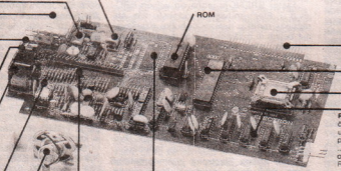
priključak za kompozitni video

priključak za RGB signal

priključak za kasetofon

rotacioni kasetofonski motor

pružnik



ROM

konektor za proširenja priručnik ne daje ni najodgovornije podatke o signalima na konektoru

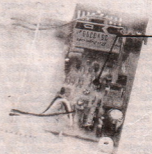
CPU 6502A

RAM četiri čipa (isti tip kao u memorijskom proširenju za „galaksiju“) obezbeđuju 32 K RAM-a

ULA u računaru je ugrađen najveći ULA čip koji je ikada proizveden, zahvaljujući njemu, upotrebjeno je samo devet standardnih TTL kola

konektor za tastaturu sudeli po broju pinova, izlaz sa tastature nije u ASCII kodu, o konverziji najverovatnije brine ULA

prostor za još jedan



U ovom specijalnom izdanju dajemo ujedno kompletan prikaz računara BBC B. Ovak prikaz bi svakako trebalo da pročita svako ko namerava da nabavi Electron: dva računara su izuzetno slična i gotovo sasvim kompatibilna, pa ćemo se uglavnom usredsrediti na razlike.

*U specijalnom izdanju „Računari u vašoj kući“ BBC mikrokompjuter je ocenjen kao najkompletniji stoni računar koji se može naći na tržištu. Ta konstatacija je izrečena posle opsežne konsultacije stranih časopisa, ali se autor ovoga teksta u nju do kraja uverio tek kada je dobio priliku da intenzivno radi sa BBC-jem i upozna njegove tajne. Šta je, dakle, to što BBC B (model A se praktično više i ne proizvodi) čini najboljim?*

# BBC



*Računar bez premca - BBC B*

Prvi utisak o BBC-ju mora da bude povoljan: na (za jedan stoni kompjuter) velikoj ali veoma lepo oblikovanoj kutiji dominira visoko profesionalna tastatura, na kojoj su, osim standardnih alfanumerika, zastupljeni mnogi specijalni znaci (računajući čak i velike i srednje zagrade, ali, začudo, ne i znak copyright koji ostaje privilegija vlasnika Spectruma). Utisak o kompletnosti tastature dopunjava deset narandžastih tastera kojima možemo da dodeljujemo proizvodnje literale. Imamo utisak da jedino tastature Hewlett-Packardovih kompjutera mogu da se mere i (verovatno) nadmaše BBC-jevu.

## *Ekranski editor*

Po uključenju računara nalazimo se u takozvanom „modu 7“. Još smo u „Računarima 1“, naime, rekli da je kod BBC kompjutera napravljen kompromis između potrebe za visokom rezolucijom i velikim slobodnim RAM-om tako što su uvedeni

razni grafički modovi — ukoliko želimo visoku rezoluciju, moramo da se posirimo sa timo da ćemo imati malo RAM-a i obratno. Mod 7 na ekran troši samo 1 kilobajt, pa je, verovatno, zato i izabran za početak rada. To je dobro, ali se pojavljuje jedan drugi, po našoj oceni za početnike vrlo ozbiljan, problem: mod 7 je kompatibilan sa takozvanim „teletekstom“, dok ostali sadrže karaktere oblikovane po ASCII standardima. To znači da, na primer, taster na kome je ispisana srednja zagrada u modu 7 daje levu strelicu što, u prvi mah, može ilekako da zbuni. Zato bismo početniku svakako savetovali da, pre nego što započne „razgovor“ sa računarom, izabere MODE 6.

Čim počnu da se pišu prvi bezijk programi, ukazuje se potreba za upoznavanjem editora koji omogućava ispravljanje grešaka u programu. BBC može da se podiže neobičnim ali izuzetno moćnim i fleksibilnim ekranskim editorom. Pre nego što ga opišemo, nećemo smetati da pomenemo razlike između ekranskih i linijskih (linijski editori se koriste kod velikog broja računara iz ekonomske i srednje klase) editora: da bismo ispravili neku naredbu treba da otkucamo EDIT (ili neku odgovarajuću naredbu) i broj linije. Zatim pomoću leve i desne

strelice pomeramo kursor i, odgovarajućim komandama, vršimo ispravke, posle kojih pritisnemo ENTER. Kod ekranskih editora su u upotrebi i druge dve strelice: pomoću njih pozicioniramo kursor na neku poziciju ekrana i vršimo ispravke „na licu mesta“. Samo se po sebi razume da su ekranski editori daleko fleksibilniji i moćniji, ali i oni imaju veliku manu: ako smo, na primer, kucali neku dužu komandu (naredbu bez linijskog broja) i pri tom pogrešili, ostaje nam jedino da je ponovo kucamo osim ako želimo da se pomirimo sa naredom i kofuzijom na ekranu.

BBC-jev editor je, u osnovi, ekranski, ali otklanja ovaj nedostatak: čim pritisnemo neki od četiri tastera na kojima se nalaze strelice, standardni kursor se pretvara u beli kvadrat i pojavljuje se drugi kursor koji se, daljim pritiscima na strelice, pozicionira na proizvoljnu poziciju ekrana. Pritiskom na taster COPY sa te pozicije kopiramo tekst na poziciju kursora-kvadrata (taster COPY, kao i svi drugi tasteri, ima funkciju autorepeat ako ga duže pritisnemo). Na taj način se može kopirati proizvoljna grupa slova — nijedan tekst sa ekrana nije „mrtav“: ako je kompjuter, na primer, negde ispisao „Mistake“, možemo slobodno da tu reč uklopimo u neku našu naredbu ili

komentar. Samo se po sebi razume da je moguće kopirati i proizvoljne komande, što postaje posebno korisno kada želimo da snimimo isti program nekoliko puta.

## Strukturirano programiranje

Po upoznavanju editora možemo da počnemo sa pisanjem prvih bejzik programa i upoznavanjem BBC-jevog programskog jezika. Ubrzo uvidimo koliko je on kompleksniji i moćniji od ostalih verzija bejzika. BBC bejzik je, naime, sakupio u sebi mnoga dobra svojstva drugih programskih jezika, ponajviše paskala. Tu, pre svega, mislimo na velike potencijale za pisanje strukturiranih programa. Strukturirano programiranje omogućava ne samo jednostavnije pisanje programa sa manje grešaka nego i jednostavne docnije modifikacije programa i njegovo prilagođavanje drugim kompjuterima.

Strukturirani program se piše po takozvanim stepenima složenosti: početni problem se deli na nekoliko etapa, a zatim se svaka ta etapa posebno programira daljim raščlanjivanjem na jednostavnije probleme. Postupak se nastavlja sve dok se ne dođe do krajnje jednostavnih problema koji se rešavaju u nekoliko programskih linija. Svaka etapa rešavanja problema se izdvaja u poseban potprogram koji je u principu nezavisan od ostalih i ima svoje ulazne i izlazne veličine. Pošto je taj program (nazivamo ga i „modul“) razvijen i potpuno testiran, snimamo ga na traku i prelazimo na sledeći. Na kraju postaje jednostavno učitavamo module i povezujemo ih u celinu. Kada je, na primer, docnije potrebno da prilagodimo već postojeći program drugom računaru, biće dovoljno da, bez ikakvih intervencija u glavnom programu, preradiamo module koji operišu sa tastaturom i ekranom.

Da bi se na racionalan način pisali strukturirani programi, treba imati mogućnost pozivanja potprograma sa prenošenjem argumenata i definisanjem lokalnih promenljivih. Ovakva mogućnost je BBC-ju data preko naredbi PROC, FN i LOCAL. U glavnom programu se, na primer, može naci naredba PROCsuma (A, B, C). Kada računar naide na nju, potražiće naredbu DEFFPROCsuma (X, Y, Z) (umesto X, Y, Z mogu, jasno, da se koriste i bilo koja druga slova ili reči) i promenljivoj X dodeliti vrednost promenljive A, a promenljivima Y i Z, respektivno, vrednosti promenljivih B i C. Zatim se izvršavaju naredbe iz DEFFPROCsuma. Ukoliko je prva od njih naredba LOCAL A, u potprogramu će biti nezavisna od promenljive A u glavnom programu, što znači da će potprogram biti sasvim autonoman. Umesto procedure, može da se koristi i funkcijски potprogram koji počinje sa DEFFNsuma (A, B) a poziva se, na primer, sa:  $X=2+FNsuma$  (total, sa-birak)+32.

## Ništa nije savršeno

Iako su konstruktori BBC-jevog bejzika očigledno uložili veliki napor da omoguće pisanje savremenih, strukturiranih progra-

## INSTRUKCIJE RACUNARA BBC I ELECTRON

ABS	ACS	ADVAL	AND	ASC	ASN	ATN	AUTO
BGET#	BPUT#	CBLL	CHRN	CHR#	CLEAR	CLOSE#	CLG
CLS	COLOUR	COS	COUNT	DATA	DEF	DEG	DELETE
DIM	DIV	DRAW	ELSE	END	ENDPROC	ENVELOPE	EEOF#
EOR	ERL	ERR	EVAL	EXP	EXT#	FALSE	FN
FOR	GCOL	GET	GET#	GOSUB	GOTO	HIMEM	IF
INKEY	INKEY#	INPUT	INPUT#	INSTR	INT	LEFT#	LEN
LET	LIST	L1LIST	LH	LOAD	LOCAL	LOG	LOMEM
MID#	MOD	MOVE	MOVE	NEH	NEXT	NUT	OLD
ON	OPENIN	OPENOUT	OPT	OR	PAGE	PI	PLOT
POINT	POS	PRINT	PRINT#	PROC	PTR#	RAD	READ
REM	RENUMBER	REPEAT	REPORT	RESTORE	RETURN	RIGHT#	RND
RUN	SAVE	SGN	SIN	SOUND	SPC	SOR	STEP
STOP	STR#	STRING#	TAB	TAN	THEN	TIME	TO
TOP	TRACE	TRUE	UNTIL	USR	VAL	VDU	VPOS
WIDTH							

ma, očigledno su propustili da posao dovedu do kraja. Pre svega, prenošenje matrica i nizova kao argumenata nije jednostavno i, što je još gore, nije dokumentovano. Tek se uvidom u stručne časopise ulazi u ovu tajnu i tako pisanje potprograma za rad sa matricama i sistemima jednačina (školski primer korisnosti strukturirano programiranja) postaje moguće. No, izgleda da nikakvim trikovima nije moguće sintetisati lokalnu ON ERROR naredbu: naredbu pomoću koje bismo bili u stanju da kontrolišemo eventualne greške u proceduri a da ne remetimo tretiranje grešaka u drugim procedurama i glavnom programu.

Nedostaje, osim toga, lokalna RESTORE naredba, kao i mogućnost da se jednostavno „sačuva“ status računara (sadržaji svih sistemskih promenljivih koje su, inače, prilično razasute po memoriji) koji bi se „povratio“ po nailasku na standardnu naredbu ENDPROC. Ove nedostatke 90% korisnika neće ni zapaziti, ali nikome, sigurno, neće biti pravo što, u obliku naredbi, MERGE nije našlo mesto. Ova naredba se, naravno, može jednostavno sintetisati ali, pogotovo za početnika, to nije isto kao da je ona u ROM-u. Pri povezivanju modula neobično je korisno što računar ima vrlo fleksibilnu i izuzetno brzu naredbu RE-NUMBER.

## Ultrasvisoka rezolucija . . .

Rezolucija 640x256, sasvim jasno, pokazuje da su Ejrkornovi inženjeri želeli da BBC ima moćnu grafiku, pa su se potrudili da obezbede kompletan set komandi za kontrolu displeja. Lepa je pogodnost što je mapa ekrana ista u svim modovima: jasno je, na primer, da u modu 1 ima duplo manje tačaka koje mogu da se kontrolišu nego u modu 0, ali je tačka sa koordinatama 1279, 1024 uvek u levom gornjem uglu pa će program planiran za mod 0 raditi i u modu 1 iako će, jasno, slika biti grublja.

Za crtanje se koriste naredbe MOVE i DRAW, čija funkcija postaje jasna kada pročitamo njihove nazive. No, mnogo je korisnija naredba PLOT koja ima tri argumenta: prva dva govore o funkciji koju treba izvršiti (PLOT 4 je, na primer, isto što i MOVE), a preostala dva daju koordinate tačke. Naredba PLOT obavlja neke poslove koji bi, inače, mučili programera: crtanje isprekidane linije, bojenje kvadrata, trouglova i mnogouglova i slično. Autoru ovoga teksta se posebno dopala paralelna mogućnost apsolutnog i relativnog (prema adresi zadnje crtane tačke) adresiranja grafičkih jedinica: svaki od ovih metoda ima svoje prednosti pa je lepo ako su oba prisutna.

U perspektiva se obično navodi da BBC poznaje 16 boja, što je samo delimično

tačno: različitih boja ima, u stvari, 8, dok se preostalih osam dobija kombinacijom osnovnih. Osim toga, nema smisla raditi u modu 2 (preveliko zauzeme RAM-a) — bolje se opredeliti za mod 4 ili mod 5 i imati na raspolaganju 4 boje, lepu rezoluciju i više slobodnog memorijskog prostora. Ni u ovim modovima, međutim, nismo ograničeni na boje koje je Ejrkorn izabrao, što je jedna od najlepših finesa na BBC kompjuteru: primenom naredbe VDU može se bilo koja od četiri standardne boje predefinisati u neku drugu. U modu 0, na primer, raspoložemo samo crnom i belom bojom, ali pose VDU 19, 1, 4, 0, bela boja postaje zelena!

## . . . i kontrola ekrana

Naredba VDU je neobično moćna, pa se oni koji saznaju da je VDU X isto što i PRINT CHR\$(X), uobično nemalo iznenađeni. Zaista, VDU je odevno jedino zbog jednostavnijeg i bržeg pisanja i uvek može da se zameni sa PRINT CHR\$(Kodovi 0—31 su rezervisani za specijalne namene od kojih je jedna (promena početnih boja) već pomenuta. Pomoću VDU se, da pomenamo samo neke primene, jednostavno definišu „prozori“ za tekst i grafiku (na taj način korisnik može da radi na jednom delu ekrana i time ne kviri izgled ostatka na kome je, recimo, neka slika), menja grafički koordinatni početak, uključuje i isključuje kursor, menja njegov oblik, aktivira i isključuje štampač . . . VDU 25, na primer, potpuno imitira već dovoljno moćnu naredbu PLOT, C, X, Y.

Mikroprocesor: 6502A

Klok: 2 MHz

Memorija: 32K, ROM-a sa bejzikom i operativnim sistemom i 32 K RAM-a

Ekran: osam različitih grafičkih modova sa maksimalnom rezolucijom 640x256 tačaka, do 32 reda sa po osamdeset znakova i do 16 boja

Interfejs: televizor, crno-beli i monitor u boji, disk-jedinica, printer, analogni ulaz port opšte namene, port za vezu sa drugim mikroprocesorima

Jezik: bejzik i assembler

Jezici na raspolaganju: lisp, fort, logo, Tastatura: 74 mehanička tastera (64 standardne namene i 10 za dodeljivanje funkcija)

Dokumentacija: odlična, sa detaljnim prikazom rada i programiranja mikroprocesora 6502, ali previše napredna za početnike



**analogni ulaz**  
pored digitalnih računara prihvaća i analognu signalu, pa uz korišćenje menoraže i priključni kabl raz menah indusmenata

**video kontroler**  
specijalni čip preuzima potrebne informacije uz korišćenje menoraže i priključni ka na ekranu

**kazetni interfejs**  
programi se mogu smisliti na dva načina — jedan omogućuje radikalniju verziju a drugi omogućuje pouzdanost

**RGB ulaz**  
obdobitnije posebne signale za crvenu, zelenu i plavu komponentu slike u kombinaciji sa posebnim kablom namenjen za pouznu visokokvalitetnih monitora

**video ulaz**  
pogon crno-beloj monitora

**modulator**  
kvarc-kristali

**serijski interfejs**

**mikroprocesor**  
ili mikroprocesor 6502-ati čip koji upravlja računaru. Ovaj čip je od različitih komponenti na svakiom tržištu

**korišćenje memorije**  
za ostale programe i 32 K RAM-a

**UUA**  
specijalno projektovan čip koji omogućuje kontrolnu logiku. Čip je od različitih komponenti na svakiom tržištu

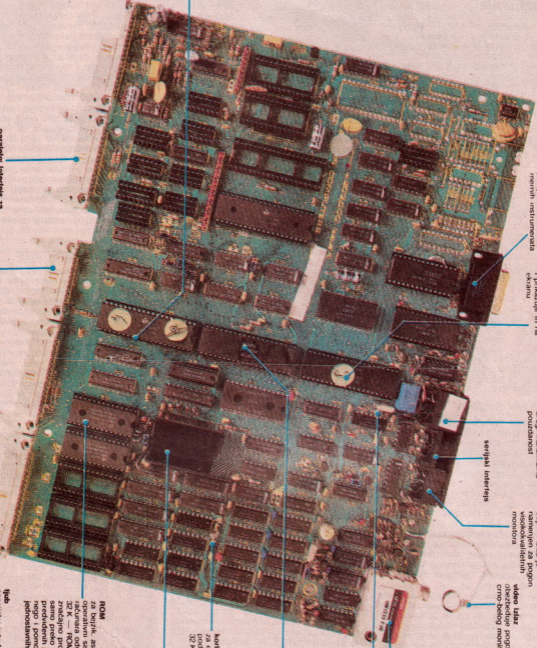
**ROM**  
za bojk, assembler i operativni sistem. Ovaj čip je od različitih komponenti na svakiom tržištu. Ovaj čip je od različitih komponenti na svakiom tržištu.

**hibe**  
specijalni interfejs koji omogućuje računaru da radi sa alternativnim mikroprocesorima

**čip za paralelni interfejs**

**paralelni interfejs za štampač**

**port opšte namene**  
na ovaj ulaz koriste se različitih čipova različitih silava



Set naredbi za crtanje dopunjavaju GCOL i COLOUR za biranje boja. Naredba GCOL, osim standardne funkcije, omogućava da se crtež logički sabira (funkcije AND, OR i XOR) sa onima što već postoji na ekranu i tako formiraju vrlo složene slike. Za igre i obradu teksta je korisno i definisanje karaktera koje se jednostavno postize jednom od opcija naredbe VDU. Što je najljepše, predefinisani karakteri mogu da budu i na tastaturi pa se, na primer, srednja zagrada može zameniti našim slovom č.

## Sintezajzer bez tastature

Generalisanje tonova je jedna od najjačih karakteristika BBC-ja. Četiri kanala (jedan samo za generisanje bubnje) omogućavaju kako sintezu komplikovane muzike tako i proizvodjenje glasa (mada se u BBC ugrađuje posebna opcija u vidu sintetizatora glasa koja uglavnom dodaje potreban softver). Jednostavne note se dobijaju naredbom SOUND koja ima četiri parametra: broj kanala, visinu note, trajanje i jačinu zvuka: SOUND 1—10, 101, 10 daje, na primer, četvrtinu note C u „Moderato“ ritmu. Pravi efekti se, međutim, dobijaju primenom naredbe ENVELOPE, pomoću koje definišemo odvojniciu tona koju najpre može da se pojačava (modulacija može da bude i po frekvenciji, a ne po amplitudi), smanjuje, opet pojačava i tako ponavlja dok se ne sinhronizuje sa tonom na nekom drugom kanalu ili, jednostavno, dok korisnik ne pritisne taster Escape.

Dok se naredbama, za generisanje slike zaista ne mogu uputiti nikakve zamke, naredbe za generisanje tona su izuzetno moćne, ali je autor ovoga teksta stekao utisak da su neverovatno složene i da najveći deo korisnika (računajući i njega) nikada neće uspeti da iskoristi sve njihove potencijale. Trebalo je da budu formirane naredbe koje će biti prilagođene potrebama korisnika, a ne čipu 76489.

## Operativni sistem

Konstruktori računara BBC su težili i uspeći da on bude neobično brz i efikasan. BBC B, premda ima star osmoiniti procesor 6502, radi daleko brže čak i od većine šesnaestobitnih kompjutera, što samo potvrđuje lezu o ravnopravnosti hardvera i softvera pri konstrukciji računara. Jasno je da su morali da budu učinjeni neki kompromisi (ne mnogo njih) da bi se dobilo u efikasnosti. Jedan od tih kompromisa je rad sa stringovima: ukoliko na početku rada programa definišemo maksimalnu dužinu alfanumerika, utrošak memorije će biti minimalni. Računar neće imati ništa protiv prodavanja tih alfanumerika u toku rada, ali će tada utrošak memorije biti skoro katastrofalan.

Iskusniji korisnici će, svakako, poželeti da rade sa mašinskim jezikom i da iskoriste ugrađeni asembler. Firma se potrudila da taj posao što više olakša dobrom dokumentacijom. Ugrađeni asembler je podržan bežikovim editorom, što znači da se asemblerke rutine pišu poput bežikov programa: čim računar naiđe na otvorenu srednju zagradu, počinje sa asembliranjem i smešta dobijeni sadržaj u memoriju počevši od lokacije na koju ukazuju promenljiva P%. U sors programu se, jasno, koriste labela, komentari i silne asemblerke naredbe, pa

je rad sasvim komforan. Pozivanje mašinskog programa se postize naredbama USR i CALL koje imaju mnogobrojne opcije za prenošenje parametara u potprogram.

U okviru uputstva za upotrebu date su početne adrese svih važnijih rutina operativnog sistema. Tako je onaj ko piše mašinske programe pošteđen brige o crtanju i pisanju po ekranu, radu sa kasetofonom, generisanju tonova... Namerno smo rekli „adresa rutina operativnog sistema“ a ne „adresa rutina bežikov interpretatora“ — ova dva programa su, naime, sasvim odvojena!

Operativni sistem zauzima 16 Kb i obuhvata sve rutine neophodne za kontrolu računara. Bilo je mnogo verzija operativnog sistema, što stvara nebrojene probleme sa kompatibilnošću. Ako kupujete BBC B, obavezno proverite da li je u njega ugrađena verzija operativnog sistema 1.20! Ukoliko računar, kada otkucate „HELP“, odgovori nekim drugim brojem, slobodno potražite drugog prodavca!

BBC-jev bežikov takođe zauzima 16 Kb i smešten je u jedan EPROM 27128 (sada se u računare stavljaju i ROM-ovi što, verovatno, znači da je razvoj operativnog sistema okončan). Neposredno pored ovoga EPROM-a se nalaze još tri podnožja — u njih korisnik može da smesti druge EPROM ili ROM čipove. Programe u njima, jasno, možemo i sami da razvijamo, ali će se većina korisnika opredeliti za gotove. Ukoliko, na primer, kupimo ROM sa paskal kompjajlerom, otkučaćemo „PASCAL“ po uključivanju računara i bežikov će biti isključen, ne zauzimajući nikakav adresni prostor. U podnožju ne moraju da se stavljaju samo interpretatori novih programskih jezika — mnogi koriste procesor teksta ili neki drugi uslužni program kao alternativnu bežikovu. Ovi programi, pogotovu ako imate BASIC II (proverite to tako što ćete pritisnuti BREAK i otkucati REPORT, ako računar ispiše (C) Acorn 1983, imate drugu verziju bežikova i vaš računar poznaje mnoge naredbe koje nisu spomenute čak ni u vašem uputstvu za upotrebu!), mogu da pozivaju programe iz bežikov interpretatora, premda proizvođač nije ponudio potrebnu dokumentaciju.

## Drugi procesor

BBC B ima ogromne mogućnosti ekspanzije. Na njegovoj zadnjoj strani su, pored standardnih priključaka za televizore, RGB i crno-beli monitor i kasetofon ugrađeni dvanaestobitni AD konverter, RS423 interfejs (verzija čuvenog RS232 serijskog interfejsa) i priključak opcionog Econet interfejsa (dopunjuje se kao posebni dodatak i služi za povezivanje više BBC računara u mrežu). Sa donje strane kulje se portovi za printer, disk-jedinice (disk interfejs se posebno dokupuje ali nije mnogo skup), „tube“ interfejs (za izuzetno brzu komunikaciju 6502 sa drugim procesorom), — 1 MHz bus za povezivanje sa laboratorijskom i drugom specijalnom opremom i standardni „korsnički port“ univerzalne namene.

Posebno je interesantno dodati drugi procesor obzirom da se tako rešava najveći problem BBC-ja — nedostatak RAM-a. Za sada može da se kupi još jedan 6502 za 64 Kb RAM-a (njegovim dodavanjem se gotovo neverovatno ubrzava izvršavanje bežikov i mašinskih programa jer softver podržava multiprocesorski rad) i Z80 procesor sa istim RAM-om koji daje CP/M kompatibilnost. Čini nam se, ipak, da u ovom

momentu ne treba trošiti nekih 200 funti za kupovinu drugog procesora. Bolje je sačekati do leta u nadi da će se, prema običajima, pojaviti nova opcija: National semiconductors 16032 mikroprocesor sa šesnaestobitnim basovima i tridesetdvo-bitnom arhitekturom koji će moći da opslužuje 16 megabajta RAM-a. Opremljen njime, BBC B će postati toliko moćan kompjuter da će teško biti prevaziđen bilo čime osim cenom u sledećih nekoliko godina i, svakako, dovoljan za sve programe koje poželeti!

Mogućnost dodavanja drugog procesora ima, na žalost, i negativnih posledica. Da bi programi bili kompatibilni sa proširenim kompjuterima, ne treba nikako „poukovati“ direktno u video memoriju, već se treba opredeliti za stalno pozivanje potprograma iz ROM-a. To ne bi bilo ništa strašno da firma Ejkm nije zaključila da će najbolje sprečiti korisnike da to rade ako uskratiti potrebne informacije. Tako u uputstvu za upotrebu nalazimo adrese sistemskih potprograma ali i objašnjenje „pakovanja“ video memorije, tačnu mapu RAM-a i sistemskih promenljivih, podršku direktnog pristupa portovima... Korisnicima ostaje jedino da sami pribave potrebne informacije što se, naravno, obavlja preko klubova programera.

## Klubovi programera

Pominjanjem klubova programera dolazimo do poslednjeg (nikako ne najmanje važnog) poglavlja ove priče o BBC B — programske podrške. Za BBC je napisano zaista mnogo izvanrednih uslužnih, sistemskih i, naravno, zabavnih programa. Igre su neobično efektne i teško da se izgled ekrana igara za druge računare može meriti sa BBC-jevima. Za ovaj računare se najviše pišu akcione igre (brz je i ima odličnu grafiku i ton), dok je avantura znatno manje, najviše zbog nedostataka RAM-a. Logičkih igara takođe ima i to izvanrednih — šahovski program White Knight (Beli konj) je jedan od najboljih silnih programa za stone računare upšte! Od akcionih igara posebno vredni nabaviti Sneppera (virtuozno urađena verzija pakmena), a od avantura — Snowball i, naravno, Hobbita.

Možemo da vam preporučimo dva komercijalna časopisa: A&B Computing (izlazi šest puta godišnje) i Acorn User, naročito ovaj drugi. Ipak, prava stvar je tek učešćanje u kompjuterski klub Beebuv. Ovaj klub može da se podiči sa preko 30000 članova i predstavlja najveću sličnu organizaciju u Evropi. Izdaje časopis koji, u poređenju sa komercijalnim, izgleda sasvim neugledno, ali u njemu nema reklama a tekst iz koga se ne može naučiti ništa novo treba tražiti svečom. Što je najljepše, časopis objavljuje programe koji se skoro mogu oceniti kao odlični i ravni komercijalnim! Sa drugim sistemskim klubom koji se zove Laserbug od sada nismo imali kontakt ali može da se pretpostavi da deluje na sličnim osnovama.

Sve u svemu, BBC B predstavlja izvanredan izbor za onoga ko za računare može da potroši 400 funti. Treba zaista biti kritički raspoložen da mu se nadu mane. I pored toga, preporučujemo vlasnicima BBC B da uloze još 100 funti i nabave proširenje od 20 Kb, koje oslobađa čitavih 32 Kb RAM-a bez obzira na grafički mod u kome se radi. Ovo proširenje, pogotovu ako bude više programa koji bi koristili njegove „usluge“, može da bude dobra zamena drugom procesoru!

Dejan Ristanović

# Commodore 64<sup>u</sup> računari izlogu

*Commodore 64 se pojavio na tržištu gotovo bez imalo galame, toliko uobičajene za promocije novih računara, ali je na njemu odmah stekao veoma čvrste pozicije. Iako je praksa iznela na videlo nekoliko njegovih dosta neprijatnih slabosti, dobri znalci računarske tehnike smatraju ovaj računar jednim od najboljih modela u tzv. srednjoj klasi. Prava snaga ovog računara, ipak, dolazi do izražaja tek kada radi u sistemu sa svojim štampačem i svojom disk jedinicom i kada mu se dokupe neka proširenja. Među njih, na žalost, spada i malo bolji — bejzik.*

Poznavalac računara odmah zapaža nedostatak kablova za povezivanje sa kasetofonom. COMMODORE 64 je konstruisan tako da traži svoj originalni kasetofon, kome može sam da uključi i isključi motor. Moguće je i povezivanje sa normalnim kasetofonima, što traži malo poznavnije elektronike, ali se tada moralo odrediti programske kontrole rada motora.

Kasetofon koji prodaje COMMODORE spaja se posebnim višezilnim kablom na računar. Standardnog je kvaliteta, sa brojačem. Brzina komunikacije računara sa kasetofonom je vrlo mala, otprilike kao kod ZX—81, ali je pouzdanost izuzetna. Svaki bajt se na traku snima dva puta i pri učitavanju se njihovim poređenjem konstatuje greška.

## „Rastegljiva“ memorija

Po uključenu se inicijalizuju boje slova, ekrana i okoline ekrana redom na belu, svetlo plavu i tamno plavu, što se pokazalo kao vrlo iritirajuća kombinacija za oči, pa je redovan postupak da se ta kombinacija promeni recimo na zelenu, crvenu i crnu. Kvalitet slike je vrlo dobar, bez uobičajenog prelivanja koje se javlja kod spektroma.

Na ekranu se pojavljuje poruka: 64K RAM SYSTEM 38911 BAJTA SLOBODNO. Nije štamparska greška! Računar zaista ima 64K RAM-a, ali je za bejzik dostupno svega 38911. Razlog ovome leži u koncepciji računara. U njega je ugrađeno 8K bejzik ROM-a, 8K ROM-a u kome su rutine za rad sa diskom i printerom i 4K ROM u kome se nalaze 4 karakter seta: po jedan za velika i mala slova i dva grafička seta.

Na nekim mestima se ROM i RAM preklapaju. Tada je ROM kao važniji jedino dostupan. Za prosečnog korisnika, dakle, COMMODORE 64 ima pristupačno relativno malo memorije prema onoj količini koja je u njega ugrađena. Ostatak memorije se može iskoristiti samo u mašinskim programima. Tada se ROM može programski isključiti i tako prići memoriji koja se nalazi „pod“ njim.

Uputstvo za programiranje je najslabiji deo ovog kompleta. COMMODORE je poznat po lošim uputstvima koja ne sadrže ni deo potrebnih i neophodnih informacija.



*Nikoga ne ostavlja ravnodušnim: Commodore 64*

Nameće se poređenje sa Spectrumovim uputstvom, koje je i po količini informacija i po kvalitetu daleko ispred COMMODORE-ovog. Korisniku koji želi da programira neophodna je knjiga „COMMODORE 64 PROGRAMMER'S REFERENCE GUIDE“, u kojoj se nalazi neuporedivo više informacija o računaru. Cena je 10 funti u Engleskoj.

Sam računar izgleda impozantno sa pravom tastaturom od 66 tipki, uključujući i izdvojena 4 funkcijska tastera sa desne strane, koji nemaju praktično nikakvu namenu osim što se programski može registrovati pritisak na njih. Sa desne strane računara nalaze se dva konektora za priključivanje komandnih palica ili svetlosne olovke. Na zadnjoj strani nalaze se konektori za dodatni ROM, TV priključak, video izlaz Hi-Fi izlaz i ulaz (za ulaz nigde nema ni reči kako ga iskoristiti), za vezu sa diskom i printerom, za kasetofon i takozvani „USER“ port na koji se verovatno mogu priključiti neki dodatni uređaji i povezati paralelni printer.

Naime, veza računara sa diskom i printerom je serijska, što znači da se prenosi bit po bit informacije, čime je izmena podataka između računara i spoljnih uređaja relativno spora. Program od 20K, na primer, učitava se oko 50 sekundi sa diska. U poređenju sa paralelnim diskovima, to je prava večnost. No, kako disk ne služi samo za snimanje i učitavanje programa, već i za rad sa podacima (RANDOM FILES, na primer) kada se može vrlo brzo prići bilo kom podatku na disku bez učitavanja prethodnih, ne treba ga se unapred odrediti.

COMMODORE 64 je baziran na procesoru 6510 — nadgradnja poznatog procesora 6502. — iako 6510 ima nekoliko novih mogućnosti u odnosu na 6502, programiranje na njima je identično. Ovi procesori su jednostavniji od poznatog Z80 i mašinsko programiranje na njima je relativno lakše. Za displej se brine poseban čip koji sa određenih lokacija u memoriji uzima podatke i generiše sliku. Postoje dva načina rada ovog čipa: u niskoj i visokoj rezoluciji.

porti za proširenje omogućuju priključivanje čitavog niza uređaja — prvi se mogu programirati pojedinačno da rade kao ulaz ili kao izlaz

konektor za tastaturu

korisnička memorija osam čipova obezbeđuje 64 K RAM-a, ali nije sve dostupno korisniku — prilikom inicijalizacije računar rezervirale dobar deo za sebe

mikroprocesor 6510

generator zvuka svi zvučni potencijali smešteni su u jedan jedini čip sa oznakom 6581

priključak za kasetofon port je prilagođen specijalnom kasetofonu i na njega se ne mogu direktno, bez izmena, priključiti klasični kasetofoni

sistemski program i pored čitavih 11 čipa koja su odvojena za njega, bezik ovom računaru nije najjača strana

serijski priključak obezbeđuje serijsku komunikaciju sa periferijskim uređajima

priključak za televizor istovremeno sa video signalom prenosi se i zvuk

ključice za kertridž prihvata sistemske programe, jezike i igre koje se nalaze u spoljnjem ROM-u

prekidač za napajanje

video čip video čip 6566 generiše grafiku visoke rezolucije i podižava sprajlove; čip je prekriven velikim oklopom-hladnjakom

komandni portovi primaju čitav niz uređaja — od svetlosne cionke do palica za igre

**Procesor:** 6510

**Klok:** 1 MHz

**Memorija:** 64 K — 20 K

ROM-a sa operativnim

sistemom i bezikom i

44 K RAM-a; kada se ne

koristi bezik interpreter,

dostupno je maksimalno

54 K RAM-a

**Ekran:** 25 redova sa po

četdeset znakova; visoka

rezolucija sa 320 x 200

tačkica; 16 boja

**Interfejsi:** kasetofon, TV,

kertridž, monitor 232,

korisnički port

**Jezici:** bezik

**Jezici na raspolaganju:**

fort komal, pilot, logo,

USCSD, paskal i druge

verzije bezika

**Tastatura:** 62 mehanička

tastera standardne

namene i 4 višenamenska

**Dokumentacija:** slaba,

nerazumljiva i nedovoljna

**Dimenzije:**

404 x 216 x 75 mm

**Cena:** 229 funti

## Grafika bez podrške

Niska rezolucija ili takozvani TEXT mod je „normalan“ način rada. Tada su na ekranu 25 linija sa 40 karaktera na svakoj. U visokoj rezoluciji na ekranu su 320 linija sa 200 kolona. Treba odmah naglasiti da BASIC 2.0 koji je ugrađen u računar ne podržava visoku rezoluciju, odnosno ne postoje komande za crtanje tačaka, linija, krugova i elipsa. Za to će se potruditi

SIMON'S BASIC koji se prodaje kao proširenje ROM-a ili na disku, a u zadnje vreme može se naći i na kaseti, po ceni od 190 DM. Osim tih, ovaj prošireni BASIC 2.0 ima još mnoštvo korisnih novih komandi. Između ostalog, podržava muzičke mogućnosti koje je osnovni bezik takođe zapostavio.

Korišćenje visoke rezolucije i muzičkih pogodnosti moguće je, doduše, i na BASIC-u 2.0, ali isključivo POKE naredbom, što je i dosta nepraktično i dosta sporo.

Jedna od dobrih mogućnosti BASIC-a 2.0 je kontrola (opet POKE naredbom) takozvanih sprajtova (SPRITE). Sprajt je „karakter“ veličine 24 sa 21 tačkice. Kada se negde u memoriji nalaze 63 bajta koja određuju jedan sprajt, dovoljno je postaviti jedan bit na 1 da bi se taj sprajt pojavio na ekranu. Njegova X i Y koordinata se takođe nalaze u memoriji i mogu se promeniti POKE naredbom, čime se sprajt seli sa jednog na drugo mesto brzinom kojom se izvršava POKE. Svaki sprajt može biti u jednoj od 16 mogućih boja.

U svakom trenutku se može kontrolisati do 8 sprajtova na ekranu. Sprajtovi mogu biti povećani po X i/ili Y osi 2 puta, pri čemu im se samo veličina udvostručuje, dok rezolucija ostaje ista.

Za svaki sprajt se može odrediti prioritet po kome se prikazuje kako za pozadinu tako i za ostale sprajtove. Tako sprajt može stajati iza slova u TEXT modu, a može biti i preko njih. U slučaju kada se dva sprajta preklape delom ili potpuno, sprajt sa višim

prioritetom će biti prikazan, a onaj sa nižim će biti zaklonjen. Video čip će se sam javiti na pogodan način setovanjem odgovarajućih bita ako je došlo do dodira dva sprajta, što se može lepo iskoristiti u igrama.

Sprajlovi su normalno u dve boje, ali se, ako se žrtvuje rezolucija po horizontali, mogu praviti i u četiri. Grafika u visokoj rezoluciji je takođe sama u dve boje, ali i za nju vredi mogućnost 4 boje uz smanjenje rezolucija sa 200 na 160.

Za muziku se brine poseban čip koji, kao i video čip, sa određenih lokacija u memoriji uzima podatke za tonove. Zato je i moguće programiranje tih čipova običnom POKE komandom. Mogućnosti zvučnog čipa su upravo izvanredne i mogu se meriti sa mogućnostima skromnih sintesizera. Moguće je istovremeno korišćenje tri zvučnih kanala. Na svakom od njih za svaki ton se može odrediti njegov oblik pomoću 4 parametra (ADSR), a može se regulisati i jačina. Povrh svega, može se menjati i kvalitet tona određivanjem talasnog oblika od koga se sastoji zvuk. I na kraju, zvuk se šalje kroz antenski priključak u televizor, tako da mu kvalitet zavisi od kvaliteta zvučnika u televizoru. Hi-fi izlaz je mono, ali se može jednostavnim elektronskim kolom simulirati i stereo.

## Bledunjava bejzik

BASIC 2.0 predstavlja klasičnu varijantu bejzika, uz vrlo malo odstupanja, i zato se programer, koji je navikao na komotnosti koje pruža Sinklerov bejzik u Spectrumu, u njemu teško snalazi. To se naročito odnosi na rad sa slovnim varijablama, sa kojima se operacije vrše pomoću LEEFT \$, MID \$ i RIGHT \$.

Programe pisane na COMMODORE 64 je lakše prevesti na Sinklerov bejzik nego obratno. Dosta je nezgodna strana BASIC-a 2.0 što su nazivi varijabli ograničeni na svega dva karaktera i što se programska linija ne može sastojati od više od 80 karaktera uključujući i broj linije. Dobra strana mu je takozvani „ekranski editor“ čime je ispravljanje programskih linija znatno olakšano. Ako se želi neka izmena u programskom listingu, dovoljno je dati LIST komandu, doterati kursor na odgovarajuće mesto, izvršiti izmenu (obrisati, zameniti ili umetnuti) i pritisnuti RETURN, čime će u programski listing umesto stare biti unesena nova varijanta linije.

COMMODORE 64 ima nekoliko „nezgodnih“ osobina koje mogu često da zasmetaju. Jedna je da brojevi programskih linija na koje „skaču“ GOTO i GOSUB komande moraju obavezno da postoje u programu — i računar ne podrazumeva sledeću liniju ako nedostaje ona na koju se „skace“. Druga osobina je da se sa svakim unosom bilo koje programske linije koja počinje brojem gube sadržaji svih varijabli. Ta osobina je naročito nezgodna pri testiranju programa: primetite da program poredne radi, izvršite izmenu (izmenite liniju i pritisnite RETURN) i ne možete nastaviti rad programa sa CONTINUE jer ste upravo izgubili sadržaje svih varijabli. Jedino rešenje je ponovo RUN. U slučaju greške na kraju dužeg programa (kome treba, recimo, 10 minuta da stigne do nje) bice vam jasna važnost ove osobine.

BASIC 2.0 dopušta rad sa numeričkim varijablama čije ime mora završiti znakom %. Opseg brojeva se kreće između — 32767 do 32767.

Osim bejzika, postoje još neki programski jezici koji mogu raditi na COMMODORE 64. To su, pre svega, pilot, logo, fort, pascal i komal, pri čemu su, bez sumnje, najznačajnija poslednja dva: pascal je kompajler sa vrlo dobrim mogućnostima, a komal je donekle sličan bejziku ali ima mnogo moćnije komande.

## Disk jedinica

Najpristupačniji (najjeftiniji) disk za COMMODORE 64 je VIC 1541 koji u Nemačkoj košta oko 700 DM. Njegov kapacitet na jednoj disketi je 170K i do 144 fajla odnosno programa. Za ozbiljno korišćenje računara neophodno je posedovanje bar jednog diska od četiri koliko se može istovremeno priključiti. Kasetofon je, jednostavno, suviše spor. Možda je za prosečnog korisnika dovoljno da nabavi program pomoću koga kasetofon snima i reprodukuje brzinom diska, ali ostaje činjenica da to još nije disk u pogledu ostalih mogućnosti.

VIC 1541 se sa računarem povezuje šestozilini „DIN“ kablom koji se dobija uz disk. Na mrežu se povezuje svojim kablom. Diskete koje se ulažu u disk su standardne, veličina 5.25 inča (13 cm). Uz disk se dobija uputstvo koje je i, ovog puta, vrlo šturo i sa malo neophodnih primera. Uz to, ima i dosta štamparskih grafskih i nedorečenosti, što sve zajedno stvara loš utisak.

Na demonstracionoj disketi, koja se takođe dobija uz disk, nalazi se desetak programa koji su samo donekle demonstracioni a više su korisni u kasnijem radu sa diskom, i za to COMMODORE treba pohvaliti.

Njihovom analizom se može saznati mnogo onoga što nije rečeno u uputstvu. Na disketi se nalazi i WEDGE koji služi da bi se izbegle dosta glomazne komande koje su neophodne pri radu sa diskom. DOS (DISK OPERATING SYSTEM) je ugrađen u ROM računara i to predstavlja značajnu prednost u odnosu na druge sisteme, kod kojih se DOS nalazi na disketi i zauzima njen veći deo, a takođe zauzima i RAM u računaru kada se u njega učita.

U disku se nalaze procesor 6502, 16K ROM i 2K RAM. Dakle disk je dosta „inteligentan“. Tehnički je izvodljivo spoljno programiranje procesora u disku, čime on prelazi na mašinski program koji se nalazi u ta 2K RAM-a.

## Matrični štampači

Od matričnih printera koji se mogu povezati na COMMODORE 64 neophodno je pomenuti dva najbolja modela koji istovremeno imaju i dosta pristupačnu cenu.

Prvi je VIC 1525 koji je, u stvari, prepravljeno SEIKOSHA GP100, u koji je ugrađen COMMODORE-ov interfejs. On sa svojim cenom od oko 700 DM predstavlja vrlo dobar izbor. Posедуje nešto lošiji karakter set kada su u pitanju mala slova, ali je to sve vrlo čitljivo. Brzina štampanja mu je oko 50 karaktera u sekundi. Širina papira koju može da prihvati je 10 inča, što odgovara formatu A4. Papir mora biti perforiran po ivicama. Može se nabaviti i kod nas.

Drugi je VIC 1526 koji, verovatno, dolazi iz COMMODORE-ovog razvojnog biroa. Cena mu je oko 900 DM. Karakter set

## Naredbe i funkcije

ABS	INPUT	RESTORE
AND	INPUT#	RETURN
ASC	INT	RIGHTS
ATN	LEFT\$	RND
CHR\$	LEN	RUN
CLOSE	LIST	SAVE
CLR	LOG	SGN
CMD	MIL\$	SIN
CONT	NEW	SPC
COS	NEXT	SQR
DATA	NOT	STEP
DEF	ON	STOP
DIM	OPEN	STR\$
END	OR	SYS
EXP	PEEK	TAB
FN	POKE	TAN
FOR	POS	THEN
FRE	PRINT	TO
GET	PRINT#	USR
GET#	READ	VAL
GOSUB	REM	VERIFY
GOTO		WAIT
IF		

mu je izvanredan, a brzina štampanja, zahvaljujući radu u oba pravca, nešto veća nego kod 1525. Širina papira je ista kao kod 1525, ali ne mora biti perforiran. Može se štampati i na običnom A4 papiru, što predstavlja značajnu prednost.

Oba ova printera rade sa 80 karaktera u redu i imaju mogućnost grafike. SIMON'S BASIC ima COPY komandu koja služi da sliku u visokoj rezoluciji iskopira na printer. U kombinaciji sa 1525 sve lepo radi, a sa 1526 dolazi do „raspada sistema“ — printer i računar se zaglave pa se moraju isključiti oba. COMMODORE je, svojevremeno, povukao iz prodaje celu seriju 1526 jer je greška u ROM-u koji se nalazi u printeru. Najvaljena je nova serija sa istom oznakom 1526 kod koje će greška biti ispravljena. Zato treba obavezno isprobati printer 1526 sa SIMON'S BASIC-om prilikom kupovine.

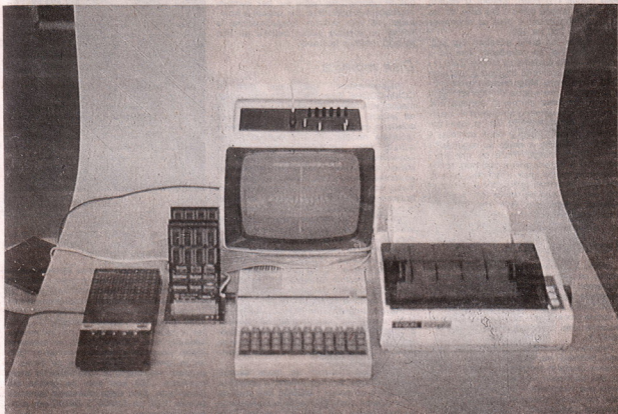
## Programska podrška

Programska podrška je dosta solidna, ali uglavnom prevladaju igre. Pošto je COMMODORE 64 sa diskom i printerom dosta pogodan za obradu teksta ili neki manji biznis, ima i takvih programa. Naravno, onim cenom dosta odskaku od programa za igru, čije su cene od 5 do 15 funti. Za sada je najskuplji program SIMPLEX 64 koji je namenjen za unos i obradu dnevnih bilansa neke manje firme sa cenom od 172 funte.

Postoji nekoliko programa za obradu teksta: WORDCRAFT 40, EASY SCRIPT, UZAWRITE. Programi su profesionalno urađeni i imaju odlične karakteristike. Tekst se može držati na disku, izmene se mogu vršiti po želji itd.

Sa cenom u Nemačkoj od oko 700 DM, COMMODORE 64 je vrlo pristupačan i našem džepu. Ako se kupuje kod nas preko oglasa, može se očekivati dupla ili veća cena. COMMODORE 64 ne preporučujem početnicima: za njega je bolji Spectrum. Za ozbiljniji korisnika, koji namerava da investira u disk i printer, COMMODORE 64 je vrlo dobar izbor. Ako ne želi da programira, imaće podršku dobrih (i skupih) programa. Ako želi da piše svoje programe, u tome neće daleko otići bez SIMON'S BASIC-a.

Ivan Gerenčir



U prvom izdanju „Gataksije“ „Računar u vašoj kući“ upoznali smo vas sa računarom LOLA 8 i njegovim osnovnim mogućnostima. U međuvremenu on se dalje razvijao i širio oblasti primene i krug ljubitelja.

LOLA 8 je nastao u računarskoj laboratoriji LOLE RIBARA jer se pojavila potreba za malim računarom koji bi se jednostavno programirao, a pored standardnih mogućnosti personalnih računara, uz dodatne module, koristio za obradu poslovnih i tehničko-tehnoških informacija, po potrebi i u realnom vremenu.

BASIC je izabran kao najpristupačniji jezik za ovakvu vrstu računara. Ipak je realizovan veoma moćan set instrukcija, naročito u oblasti matematičkih operacija, sa tačnošću do 10 decimala i velikom brzinom izvršenja.

LOLA 8 je, i bez posebne reklame, stigao u mnoge škole, fabrike, institute, projektne biroe, laboratorije, domove zdravlja, kancelarije. Veliko interesovanje i mogućnosti primene u raznim oblastima kao i iskustva u korišćenju računara LOLA 8 su usloveli dalji rad na njegovom razvoju.

Pokazalo se da su 6K reči RAM memorije nedovoljni za ozbiljnije programe na BASIC-u. RAM memorija se proširuje modulima od po 16K. Poređenja radi, u 6K reči RAM memorije može da stane program od oko 150 linija (dužine 1 reda ekrana), 650 različitih promenljivih, 100 vektora dimenzija 10 elemenata ili pak

230 nizova prosečne dužine 40 karaktera. Dodavanje samo jednog modula od 16K dužina programa može da bude i do 500-600 linija; čak 2500 promenljivih, 330 vektora dimenzija 10 ili 850 nizova.

Većina korisnika personalnih računara zna da je njihova slabla lačka nepouzdanost zapisa na kasetofonu, koja zavisi od kvaliteta i tipa korišćenog kasetofona. Čistoća magnetne glave, kasete i mnogih drugih faktora. Rešenje sprege za kaseofon LOLE 8 omogućuje pouzdano snimanje i čitanje programa za gotovo sve kasetofone u komercijalnoj upotrebi. Novost su i 3 nove BASIC komande:

DSAVE, DLOAD i DVERIFY koje omogućavaju snimanje i čitanje vrednosti promenljivih na kasetofonu, a vezane su za primene u poslovnoj obradi informacija.

Logičan nastavak razvoja računara LOLA 8 je modul za priključenje štampača. Programi pisani rukom su često nepregledni i nečitki, pa se zbog mnogobrojnih izmena dešava da mislimo da smo uneli jedno, a u memoriji računara je nešto drugo. Štampanjem sadržaja programa ovi problemi nestaju, ali je još značajnija mogućnost štampanja rezultata obrade informacija.

Većina štampača koje srećemo na inostranom tržištu ima paralelni ulaz/izlaz (tzv. Centronix standard) i svi oni mogu da se priključe na LOLA 8. Situacija u domaćem tržištu štampača je, dosta loša. Postoji nekoliko domaćih

proizvođača (TERA, TRS), ali su njihove cene za sada previsoke.

Zato je razvijen modul koji omogućava da se električne mašine za pisanje mogu koristiti umesto štampača. Postoji nekoliko tipova ovih mašina na domaćem tržištu OLIMPIA, ROBOTRON-EI Niš... Njihova cena je znatno pristupačnija, a veliki broj firmi ih već poseduje. Naravno ove pisane mašine ne mogu da pruže sve pogodnosti kao savremeni štampači, ali su za većinu primena više nego dovoljne.

LOLA 8, sa dodatnim modulima, omogućava izgradnju malih sistema obrade informacija za različite primene: evidencije, knjigovodstvo ili praćenje i upravljanje jednostavnim procesima.

Nadamo se da vas ovim poslednjim nismo preplašili. Hteli smo da vam ukažemo na još neke mogućnosti primene LOLE 8. U LOLA RIBARU svakog dana preko 30 učenika III razreda OC „Beogradski bataljon“ radi na računaru LOLA 8, radaju se ideje za nove igre, svira Bahova svita u H molu... drugim rečima stasava nova generacija koja zna šta je računar.

O saradnji LOLE sa srednjim i osnovnim školama i drugim mogućnostima računara LOLA 8, recimo ton generatora, u nekom od sledećih nastavaka Galaksijinoj časopisa „Računar u vašoj kući“. Dotle, za sva obaveštenja obratite se na tel. 570-227 ili pismom na adresu Industrija mašina IVO LOLA RIBAR, za računar LOLA 8, 11250 BEOGRAD-ZELEZNIK.

## LOLA RAČUNARSKI SISTEMI

Industrija IVO LOLA RIBAR jedan je od najnaprednijih jugoslovenskih proizvođača računarskih sistema namenjenih upravljanju u industriji. Famija LOLA računara, izgrađena na bazi standardnih modula, odlikuje se fleksibilnošću, pouzdanošću u industrijskom druženju i jednostavnošću eksploatacije koja ne zahteva posebno obrazovanje u oblasti računara.

Svi predstavnici LOLA familije industrijskih računara se mogu preko standardnog modula spregnuti sa centralnim računarom, omo-

gućavajući tako izgradnju distribuiranih sistema upravljanja za veće proizvodne celine i čitave fabrike.

Istovremeno više generacija stručnjaka u projektovanju i proizvodnji upravljačkih sistema i korišćenje najsavremenije mikroprocesorske tehnologije garancija su kvaliteta ILR proizvoda u oblasti industrijske elektronike. IVO LOLA RIBAR danas raspolaže timovima stručnjaka koji rade na razvoju novih računarskih sistema, obuhvatajući tako svu širinu polja automatizacije u industriji.

## LOLA SISTEMI NUMERIČKOG UPRAVLJANJA

U oblasti programskog upravljanja LOLA RIBAR predstavlja svoje sisteme numeričkog upravljanja LOLA 30 (za bušice, glodalice i centre obrade), LOLA 30S (za strugove) i LOLA 42 za upravljanje industrijskim robotima. Sva tri sistema su predviđena za upravljanje složenim kretanjima max. 6 rotacionih ili linearnih osa simultano.

LOLA 30 CNC (computer numerical control) obezbeđuje sve standardne funkcije najmodernijih alatah mašina.

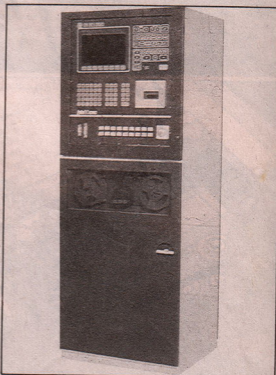
- Prilagodjenje na različite tipove mernih sistema (sa rezolucijom 0,001 ili 0,01 mm) • 3D linearna interpolacija • kružna interpolacija (pun krug) u tri ravni • unošenje podataka preko čitača bušene trake, kasetofonske jedinice ili alfanumeričke tastature • Tabeli korektura za max 96 alata • Fiksni ciklusi bušenja i struganja • Glodalice i rezanje navoja • Apsolutno, inkrementalno i reološko programiranje • Parametarsko programiranje •

Ekraneka jedinica (31 cm) omogućava prikazivanje aktivnih blokova, trenutnih pozicija svih osa, aktivnog i četiri sledeća bloka programa, kao i svih potrebnih informacija, upozorenja i uputstava za operatera.

Integrirani deo LOLA 30 sistema je programabilni automati PA 512, na bazi kojih je realizovana fleksibilna spreza sa mašinom.

Specifičnost LOLA 42 CRC (computer robot control) sistema za upravljanje industrijskim robotima je način unošenja programa u režimu obučavanja. Operator donosi osu po osu robota u željenu poziciju koja se automatski memorise. Ovakvo uneti program se zatim može izvršavati u automatskom, pojedinačnom i ciklisanom režimu.

Osim više tipova instrukcija kretanja, upravljački sistem prima i posebne instrukcije sinhronizacije rada robota sa tehnološkim okruženjem. U tom smislu na raspolaganju korisniku stoje 32 ulazno-izlazna signala kojima se robot povezuje sa drugim robotima ili mašinama.



## LOLA PROGRAMABILNI AUTOMATI

Najširu primenu u svim granama privrede (eksploatacija i proizvodnja ruda i nafte, preradivačka industrija, transport, poljoprivreda i proizvodnja hrane) upotrebe našla je gama LOLA programabilnih automata. Pojava PROGRAMABILNIH AUTOMATA izazvala revoluciju u upravljanju mašinama i procesima. I pored sve raznolikosti primene, programabilni automati se dimenzionišu prema konkretnom upravljačkom zadatku i "rastu" zajedno sa njegovim eventualnim proširenjem.

Obezbeđuju realizaciju „relejni“ logike sa novom savremenom tehnologijom, brojački i vremenske funkcije, kao i obradu numeričkih podataka. Za rešavanje problema regulacije u industriji razvijeni su specijalni moduli PID regulacije, koji prema standardnim algoritmima obdružuju do 32 regulacione petlje.

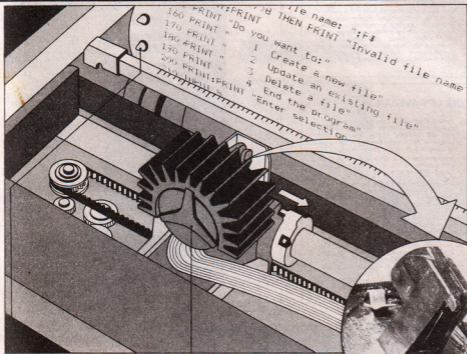
Poseban uređaj za razvoj i testiranje upravljačkih sistema na bazi programabilnih automata — PRDG PA omogućava unošenje

upravljačke logike u vidu standardnih relejnih šema uz istovremeno prikazivanje na ekranu. Uneti program se može sačuvati u EPROM memorijama, na magnetnoj kaseti, ili se, priključivanjem štampača može dobiti kompletna dokumentacija. Specijalni rebrni prikazivanja omogućava operateru da u svakom trenutku prati na ekranu odvijanje procesa u realnom vremenu. Svi aktivirani elementi „relejnih šema“ prikazani su svrznuto. Time se vreme potrebno da se otvori reprogramabilnost u radu uređaja svodi na minimum.

LOLA programabilni automati (mini PA, PA 206, PA 512, i MPA) koncipirani su tako da se optimalnim ekonomskim efektom zadovolje širok opseg kategorija primene, od minimalnih konfiguracija sa max. 32 ulaza/izlaza, do složenih upravljačkih zadataka koji podrazumevaju ne samo veliki broj ulazno/izlaznih signala (do nekoliko hiljada), već i visok nivo numeričkog procesiranja.

# Štampači u akciji

Periferijska oprema



Matrični štampač: Za oblikovanje karaktera sistem koristi matricu sa tačkama; štampanje slova se vrši kolonu po kolonu. Pomeranjem glave sa vertikalnim nizom iglica: tačkasta struktura otisnutih slova ne može se sakriti čak ni kod najkvalitetnijih štampača



U specijalnom izdanju „Računari u vašoj kući“ posvetili smo dosta prostora karakteristikama štampača i kriterijumima kojima se treba rukovati pri njihovom izboru. Zbog nedostatka prostora (da, i 100 strana nije mnogo) nismo mogli da prikazemo detaljnije nijedan model i tako, na praktičnim primerima, ilustrujemo savete koje smo davali. Ovoga puta ispravljamo taj propust i opširnije prikazujemo modele za koje smatramo da su vredni pažnje domaćih ljubitelja računara.

## Seikosha GP100

Seikosha GP100 je veoma popularan štampač sa najboljim odnosom mogućnosti/cena. Radi se o matričnom tipu koji proizvodi slova na matrici 6x7 (zapravo 5x7, jer poslednja kolona mora da ostane slobodna zbog razmaka između slova). Ova matrica je, u osnovi, mala i nije dovoljna za definisanje naših latiničnih slova, posebno velikog „S“. Ipak, uz malo „umetničkog“

dara i malo više mašte, postiže se zadovoljavajući kvalitet teksta.

Štampač piše na kompjuterskom tabulir papiru (ima ga u prodaji i kod nas) čija je širina promenljiva (maksimalna 10 inča, ili oko 25 centimetara). Traka je, na žalost, uvozna, ali ne košta više od 5 funti (900 dinara — što znači da se bez problema naručuje preko pošte) i veoma je kvalitetna i trajna — dobar deo rukopisa ovog specijalnog izdanja je urađen na Seikosha štampaču, a traka je i dalje kao nova.

Štampač ima četiri različita seta karaktera koji se biraju pomoću mikroprekidača na zadnjoj strani njegove štampane ploče. Ako, na primer, želimo da koristimo nemačka slova umesto engleskih, jednostavno ćemo odvrnuti četiri zavrtnja, skinuti kutljiu i pomeriti prvi mikroprekidač u položaj ON. Osim nemačkog, možemo da izaberemo američki, engleski ili švedski set karaktera; pošto se štampači ne prodaju u Jugoslaviji, Seiko (firma koja proizvodi štampače Seikosha, pominjemo je zato što mnogi, računajući i autora ovog teksta, teško uočavaju vezu između ova dva imena dok im se ne skrene pažnja) očigledno nije smatrao za

potrebno da uvrsti i naš set karaktera u standardnu opremu.

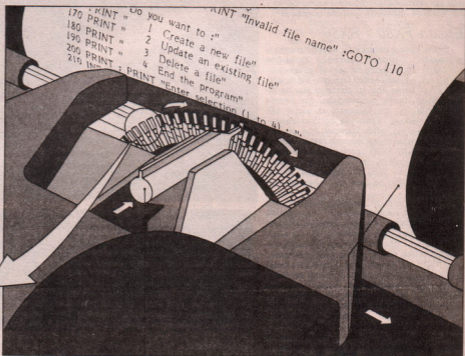
Slova su normalno visoka 2,82 mm i široka 2,11 mm, ali se mogu odabrati i dvostruko veća — tada u jedan red staje samo 40 slova (red ima 80 normalnih slova), što je prilično pogodno za naslove i podnaslove. U jednom redu se može naći kombinovani tekst pisan normalnim i uvećanim slovim.

Oznaka štampača, GP100, predstavlja skraćenicu od Graphic Printer št, naravno, znači da postoji mogućnost crtanja slika. Korisnik ima pristup svakoj od 480 tačaka u redu (80 karaktera po 6 tačaka) definisanjem sopsvenih kolona. Pretpostavimo da, na primer, želimo da štampano karakter poput onoga na slici. Za svaku njegovu kolonu sabiramo brojeve koje predstavljaju svaku tačku. Za prvu tačku dodajemo  $2^0 = 1$ , za drugu  $2^1 = 2$ , za treću  $2^2 = 4$ , za četvrtu  $2^3 = 8$  i tako dalje. Za poslednju tačku u koloni dodajemo  $2^7 = 64$ . U našem primeru brojevi koji predstavljaju karakter bi bili 7, 42, 112 i 0. Njih ćemo, najpre, povećati za 128 (jedna od specifičnosti GP100A), a za tim saopštiti računaru, jedan po jedan. U



**Štampač, bez sumnje, predstavlja najvažniju periferijsku opremu svakog kućnog računara koji svome vlasniku predstavlja nešto više od igre. Bez njega se ne može zamisliti obrada teksta, rad sa datotekama, pa čak ni ione ozbiljnije programiranje. Na tržištu se trenutno mogu naći dva tipa štampača — matični (dot-matrix) i sa lepezom (daisywheel). Prvi omogućuju definisanje posebnih karaktera (kao što su naša latinična slova) i crtanje i rade samostalno i veoma brzo. Drugi su previše spori, zahtevaju mehaničke intervencije na lepezi da bi se uvela i naša slova i stalno opsluživanje papirirom, ali imaju izvanredan otisak. Kvalitet otiska matičnih štampača je dovoljan za ličnu upotrebu — listinzi, datoteke, dokumentacija, evidencije — ali na njemu ne možete napisati tekst za novine, seminarski ili diplomski rad, pa čak ni poslovno pismo. Tu postoje samo dve mogućnosti: štampač sa lepezom ili — ručni rad na klasičnoj pisacoj mašini.**

**Štampač sa lepezom:**  
Lepezu na čijim se špicama nalazi po jedno slovo okreće veoma precizan motor; kada dođe u poziciju za štampanje, slovo otiskuje na hartiju elektromagnetski udarac; otiskuje slovo na hartiju mašina i štampača sa lepezom nema značajnijih razlika u kvalitetu otiska, ali je rad sa štampačem neuporedivo udobniji, jer je pre štampanja moguće sve ispravke na tekstu izvršiti elektronskim putem



okviru jednog reda možemo slobodno da mešamo obične, dvostruke i karaktere sačinjene od kolona koje smo definisali bez ikakvih ograničenja, što nam omogućava pisanje normalnog latiničnog teksta, pa i grčkih slova potrebnih za neko naučno štivo.

Seikosha GP100 ne spada u brze štampače — ispisuje samo 50 karaktera u sekundi, što bi bilo sasvim dovoljno da je štampač „pametniji“: da piše i dok se glava vraća u početni položaj. To, na žalost, nije slučaj, pa ispisivanje jedne strane teksta uma da potraje prilično dugo. Za većinu primena ovo nije velika smetnja. Posebno ograničenje u brzini predstavlja okretanje doboša koje je sporije nego kod mnogih drugih printera — prelazak na sledeću stranu ume, dakle, i da potraje. GP100A je, uz to, i prilično bučan štampač, što neće biti previše omiljeno u vašem domu.

Autoru ovoga teksta ne smeta ni jedno od ovih ograničenja jer mu je jasno da je ono rezultat težnje da štampač bude jeftin. Pri izradi softvera koji je smešten u ROM se, međutim, škrtilo daleko više nego što je dopustivo. GP100A je, tako, ostao veoma „neinteligentan“ uređaj: ne zna kada nailazi kraj strane, pa će pisati po perforaciji ukoliko se programer ne pobrine da prebroji ispisane redove, ne omogućava softversko smanjenje broja slova u redu niti promenu razmaka između redova, pravi neobično mnogo problema, kada programer poželi da piše dva puta preko istog teksta (radi podvlačenja važnih poruka, na primer) i, što je posebno zamorno, ima neprijatno dugo pamćenje: ako u jednom trenutku nesmotrenosti pošaljemo nekoliko karaktera u njegov bafer, ne postoji regularan način da ih odalje obriskujemo pre nego što red bude štampan (isključivanje štampača iz napajanja, jasno, ne smatramo regularnim rešenjem, jer uvek može da izazove probleme kod drugih istovremeno priključenih uređaja). Ne bi bio nikakav problem da je firma upotrebila neki od ionako neiskorišćenih kontrolnih karaktera za reinicijalizovanje štampača; ovakav propust, jednostavno, navodi na pomisao da firma nije želela da ugrozi prodaju svojih

daleko skupljih modela i da je zato GP100A ostao bez boljeg ROM-a.

Uz štampač se dobija knjižica od 30 strana kao jedina dokumentacija. Za iskusnog korisnika računara dokumentacija je sasvim dovoljna, a obuhvata čak i sve što je potrebno za gradnju interfejsa za lične potrebe. Za početnika je knjižica, međutim, previše lakonski pisana: čini nam se da će neko kome je GP100A prvi štampač utrošiti časove, a možda i dane da shvati ono što je moglo da bude jednostavno objašnjeno. Žalosa je činjenica da oni koji pišu uputstvo za korišćenje računara izbegavaju da se pozabave radom sa štampačima (svakako zato što ima mnogo nekompatibilnih tipova), računajući da će ovaj najpoučasniji periferijski uređaj biti posebno dokumentovan, a oni koji pišu uputstva za upotrebu štampača računaju sa tim da je onome ko poseduje računaru dovoljna dokumentacija koju već ima. Malo više standardizacije i saradnje poštedelo bi početnike mnogih gubitaka.

I uve u svemu, Seikosha GP100A je umereno dobar izbor i verovatno može da se pohvali najboljim odnosom mogućnosti/cena. Začudo, nije postigao preveliku popularnost (možda će uskoro biti bolje?), što se odrazilo na slabiju softversku podršku. Sof-

*Povlačan za hobiste: Iako nema baš najbolji kvalitet otiska, štampač Seikosha 100, ipak, u odnosu na cenu, predstavlja veoma dobar izbor*

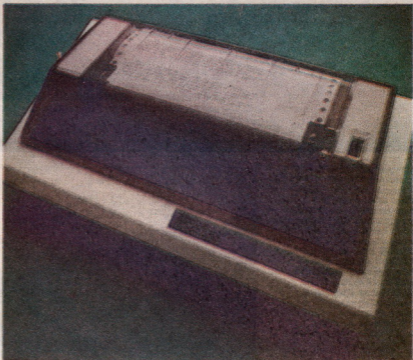
Iverska podrška za štampač? I ona ponekada može da bude važna: ako imate BBC B ili Electron, na tržištu i po časopisima ćete naići na gomilu programa koji omogućavaju obradu teksta na Epsonu, kopiranje sadržaja ekrana na Epsonu, stvaranje utiska o bojama na Epsonu i mnogo što-šta slično. Ukoliko ne nameravate da sami pišete ovakve programe (nije teško ali oduzima vreme), zaboravite na Seikoshu i počnite da razmišljate o Epsonu kome ćemo i mi, naravno, posvetiti sledeće poglavlje.

## Epson FX80 i RX80

Firma Epson danas drži vodeće mesto u proizvodnji i prodaji štampača za kućne računare. Za to može da zahvali prvenstveno svome modelu MX80 i drugim štampačima iz MX serije koji su postavili standarde za mnoge druge proizvođače. No, vreme čini svoje, pa su štampači iz serije MX postali preskupi. Epson je odlučio da pripremi dovoljno jeftin štampač koji će dostojno reprezentovati renome firme i nastali su FX80 i RX80. FX80, zapravo, ima sve dobre osobine štampača iz MX serije, a košta samo 380 funti (320 bez interfejsa), što je sasvim dovoljno da vam ga najtoplije preporučimo.

FX80 je, naravno, matični štampač koji formira slova na matrici 6x8, što znači da slova g, j i slična ne bivaju deformisana kao kod Seikoshe. Set karaktera je više nego univerzalan i obuhvata preko 250 simbola. Otkle toliko slova? FX80, osim standardnih ASCII karaktera, ima i nekoliko vrsta slova, kao i karaktere od interesa za razne jezike (naših slova, na žalost, ni ovdje nema). Što se vrsta slova tiče, na raspolaganju su normalna (80 u redu), povećana (40 u redu), kondenzovana (137), kondenzovana-povećana (68), elitna (96) i elitna-povećana (48 slova u redu). Sve ove kategorije slova su, uglavnom, objašnjene samim imenom osim jednih — elitnih. Elitna slova bi trebala da se što više približe kvalitetu koji se postiže na pisačkoj mašini, odnosno da ulepšaju izgled dokumenta. Zašto bi neko onda koristio bilo koja druga slova? Jednostavno zato što mnogi vole da dokument ispisani na štampaču izgleda u pravom smislu kompjuterski.

Epson FX80 je jedan od vrlo retkih štampača koji omogućavaju automatsko podvlačenje teksta i takozvani „double strike“ mod. O čemu se radi? Ponekad nam je potrebno da posebno istaknemo deo teksta ili, jednostavno, dobijemo kontrastniji dokument koji će ići u štampu. Tada ćemo izabrati opciju koja štampaču nalaže da svaki red štampa po dva puta, pri čemu će drugi trag biti za pola tačke pomeren u odnosu na prvi (sve tabele u Računarima 1 su, na primer, ovako kucane). U tom slučaju štampač, zbog povećane preciznosti, ne štampa u dva smera, što značajno umanjuje brzinu koja u normalnim uslovima dostiže više nego dovoljnih 160 karaktera u sekundi. Postoji još jedan način da sa razlogom prepolovimo ovu brzinu: ukoliko se odlučimo



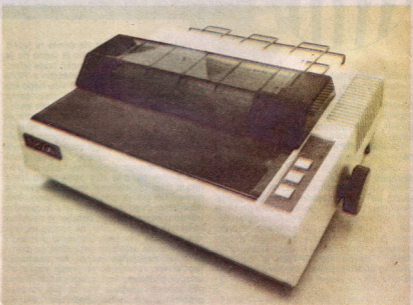
## Seikosha GP 100A

Tip: matični  
Matrica: 5x7  
Interfejs: centronics  
Memorija: 256 znakova  
Linija po inču: 6  
Karakter po inču: 6-12

Brzina štampanja: 50 znakova u sekundi  
Karakter u redu: 80  
Broj kopija: 1  
Podvlačenje: ne  
Isticanje: da  
Proporcionalni razmak: da

Blok grafika: ne  
Grafika visoke rezolucije: da

Dvosmerno štampanje: ne  
Logično traženje: ne  
Maksimalna širina papira: 10  
Pomeranje papira: traktor  
Dimenzije: 42.7x23.8 cm  
Masa: 4.5 kg  
Cena: 247 funti



„Kraljevska loza“ među matičnim štampačima: Epson MX 80 može da zadovolji potrebe i najrazmaženijeg ljubitelja kućnog računara ali je za njega potreban više nego dubok džep

## Epson FX 80

Tip: matricni  
Matrica: 9x9  
Interfejs: centronics  
Memorija: 256 karaktera  
Linija po inču: 6.8  
Karakter po inču: 5, 8, 2,  
5, 10, 16, 5

Brizna štampanja: 80  
znakova u sekundi  
Karakter u redu:  
maksimalno 132  
Broj kopija: 3  
Podvlačenje: da  
Isticanje: da  
Proporcionalni razmak:  
opciono  
Blok grafika: ne

Grafika visoke rezolucije:  
da  
Dvosmerno štampanje: da  
Logično traženje: da  
Maksimalna širina papira:  
25 cm  
Pomeranje papira: traktor  
Dimenzije: 40,6 x 38,1 cm  
Masa: 5,5 kg  
Cena: 400 funti

mo za 80 karaktera u sekundi, štampač će pisati daleko tiše, što je korisno ako radimo u nekoj kancelariji ili, jednostavno, ako nam ukućani ne trpe buku.

Za one koji su navikli na skupe pišaćne mašine, kod kojih su slova proporcionalna razmaknuta („m“, na primer, ne zauzima onoliko mesta koliko i „i“). FX80 nudi opciju koja omogućava čak i takvu pogodnost. Da stvar bude još lepša, bilo koja druga opcija (npr. veličina slova) može da bude izabrana zajedno sa proporcionalnim razdvajanjem slova.

Onima za koje ni ovoliko slova nije dovoljno, na raspolaganju su naredbe za definisanje karaktera. Možemo, na primer, da predefiniramo neke nepotrebne ASCII karaktere i dobijemo naše slovo „Č“ koje će, samim tim, biti dodeljeno nekom tasteru (ako imamo BBC ili Electron, efekat ćemo kompletirati predefinisanjem tog karaktera u setu slova računara, pa ćemo na ekranu videti baš ono što treba da se pojavi na štampaču). Interesantno je, međutim, zapaziti još nešto: posle malo eksperimentisanja sa definisanjem karaktera, primetićemo da naša slova izgledaju ružnije od originalnih. Da li to znači da imamo manje dizajnerskog talenta? Možda, ali se i Epson pobrinuo da tako bude: kod slova iz ROM-a se ne adresiraju samo tačke već i prostori između njih (po „pola tačke“) što, jasno, poboljšava izgled dokumenta. Ta mogućnost, na žalost, nije ostavljena i korisnicima.

FX80 ne omogućava definisanje karaktera. Epson FX80 je izvanredan za one koji žele da crtaju. Moguće je, naravno, kontrolisati svaku tačku u redu, ali koliko ima takvih tačaka? Skoro da bismo mogli da kažemo „koliko poželite“: u normalnoj rezoluciji na svakih 8 inča (20 cm) ima 480 tačaka, u dvostrukoj rezoluciji na taj prostor stane dvostruko više (960), ali se na raspolaganju i takozvana „četvorstruka tačnost“ koja nudi 1920 tačaka! Ovakva rezolucija se približava ploterima i omogućava, na primer, generisanje bar koda koji bi „razumeo“ HP41C ili neki drugi računar opremljen optičkim čitačem. Osim ovih modova, na raspolaganju su i CRT grafike i li (640 odnosno 720 tačaka na 20 cm) koje su idealne za vlasnike BBC računara, jer se podudaraju sa maksimalnom rezolucijom na ekranu računara. Za šta oni mogu da koriste veće rezolucije? Vešti programeri su već sastavili rutine koje senče slike, čineći tako da se stekne utisak o bojama koje su bile zastupljene na ekranu koji je dämpovan.

Za crtanje je vrlo korisno što može da se izabere razmak između redova i to sa velikom preciznošću (od šestine do dvesta šezdesetine inča). Međe, osim toga, lako da se izabere položaj leve i desne margine (samim tim i broj slova u redu), pomeranje

početka teksta od vrha i dna strane i mnogo što-šta drugo. Vidimo da, za razliku od Seikoshe, Epson FX80 prepoznaje kraj strane, ali on radi i više od toga: prepoznaje i zvučno signalizira trenutak kada je vaša zalih papira pri kraju. U istom trenutku štampač „koči“ kompjuter i čeka da stavite još papira. Jednostavan pritisak na taster ON LINE će, zatim, naložiti računaru da nastavi sa slanjem karaktera. Osim tastera ON LINE, na kucištu printera su i tasteri LF (pomeranje papira za jedan red) i FF (pomeranje na početak sledeće strane), koji imaju i sekundarnu ulogu: ako pritisnete LF dok uključujete štampač, biće obavljena automatska dijagnoza (korisno kada kupujete štampač), a ako u tom trenutku držite pritisnute FF i LF, svi podaci koje kompjuter bude slao biće štampani kao heksadekadne vrednosti (čak i kontrolni karakteri).

Osim tastera na kutiji, na samoj štampajnoj ploči printera postoji čitava serija mikroprekladača. Uz njihovu pomoć možemo da biramo stanje u kome će naš štampač biti kada ga uključimo. Ako, na primer, stalno koristimo „elitna“ slova, nema nikakvog smisla da posle svakog uključivanja šaljemo čitavu seriju kontrolnih karaktera — jednostavno ćemo pomeriti jedan preključnik. Slično se odnosi i na druge opcije koje se u toku rada softverski menjaju, kao i na setove slova koja biramo (RAM u kome se nalaze karakteri koje smo mi definisali biva, jasno, izbrisan kad god isključimo štampač).

Krajnje je vreme da kažemo da Epson FX80 piše na tabulir hartiji, ali da njegova nešto skuplja verzija, FX80, piše na standardnom papiru A4 formata. Dobra investicija? S vremena na vreme svakako vredni imati ovako nešto, ali nam veruje na reč da nije minimalo lako stalno nadzirati štampač i svaki minut ga opsluživati novim listom papira. Zato je za duže tekstove bolje nabaviti bank post tabulir papir (ima ga ponekad i kod nas).

Epson FX80 u standardnoj verziji ima Centronics paralelni interfejs, ali se za njega može dokupiti i RS232. Obzirom da bafer ovog štampača ima čitava dva kilobajta, Centronics interfejs će omogućiti brži prenos i manji губитак računarevog vremena na beskorisno čekanje. Ovaj bafer, na žalost, ne može da se koristi ukoliko definišemo neke karaktere (RAM je ograničen), što znači da je za većinu primena u

kojima su potrebna slova Č, Ć, Š, S i računar čekati da bude ispisano svako slovo, pa će i RS232 biti više nego dovoljan.

Treba stvarno biti veliki probirač da biste štampaču FX80 uputili neka zamerka. Dokušamo, ipak, i to. Pre svega, upustvo prilično opširno, ali ne i naročito napisano, nedostaju upravo one informacije koje ljubitelji računara koji je kupio svoj pr štampač bile od neke koristi. Tekućama snalaženju potpomaže i nekoliko „bagova“ u operativnom sistemu štampača: neki kr dovi u nekim slučajevima (kada smo, na primer, odlučili da pišemo nekim posebnim slovima) jednostavno ne rade ono što trebalo da rade, dok drugi izazivaju ne neprijatne efekte. Moguće je, međutim, da su ti „bagovi“ već otklonjeni ili da će uskoro biti.

Sve u svemu, Epson FX80 predstavlja izvanredan izbor kako za one kojima je potreban brz printer sa velikom grafičkom rezolucijom tako i za one koji žele kvalitetan izlazni dokument. Kod njega, međutim, treba obratiti pažnju na jednu stvar: štampač je namenjen „novim“ računarima, koji je set naredbi koje podržavaju (FX80, na primer) i novim računarima sa skromnom grafikom (mislimo, naravno, na računar „galaksija“) može da zameta činjenu da FX80 ne podržava blok grafiku, što otežava direktno prenošenje sadržaja teksta na štampač. Blok-grafika se, jasno, može zamisliti definisanjem karaktera, ali takvo rešenje zahteva učitavanje posebnog programa u računar posle svakog uključivanja štampača. Osim toga, njegov ROM od 16 Kb je kompaktan (EPROM 27128), pa nije lako pronaći način da se u njega upišu neka naša slova. Oba ova nedostatka osloboden je treći štampač o kome ćemo govoriti: Okidata Microline 80. Microline 80 je stariji, sporiji i daleko slabiji štampač od FX80 ali, kao što ćemo videti, predstavlja prihvatljivu alternativu za mnoge primene.

## Oki Microline 80

Microline 80 je, da to kažemo na samom početku, prilično star štampač, što, u kompjuterskom svetu, svakako nije naročita preporuka. Iako se, jasno, radi o štampaču opšte namene koji može da bude opremljen nekim od standardnih interfejsa, Microline 80 (u daljem tekstu, jedino radi kraćeg pisanja, MC80) specijalno je pripreman za računar TRS 80 (odatle verovatno oznaka). Za potencijalne vlasnike štampača u Jugoslaviji MC80 ima nekoliko velikih kompjuterskom papiru (oba, kao što smo rekli, mogu da se nabave kod nas, ali je uvek jednostavnije tražiti standardni papir), koristi običnu traku za pišaću mašinu i, najzad, njegova „blok grafika“ je kao stvorena za naš računara „galaksija“.

Svi karakteri MC80 je više nego kompletan: pored velikih i malih slova, brojeva i standardnih ASCII specijalnih simbola, tu

## Oki Microline 80

Karakter u redu:  
maksimalno 132  
Broj kopija: 1  
Podvlačenje: ne  
Isticanje: ne  
Proporcionalni razmak:  
ne  
Blok grafika: da  
Grafika visoke rezolucije:  
ne

Dvosmerno štampanje: ne  
Logično traženje: ne  
Maksimalna širina papira:  
9,5 inča  
Pomeranje papira: valjak,  
traktor, frikciono  
Dimenzije: 34,2 x 24,5 cm  
Masa: 6,5 kg  
Cena: 242 funte

Brizna štampanja: 80  
znakova u sekundi

Izvanredan kvalitet otiska: Štampač sa lepezom Juki 6100 predstavlja jedan od najboljih modela u svojoj klasi

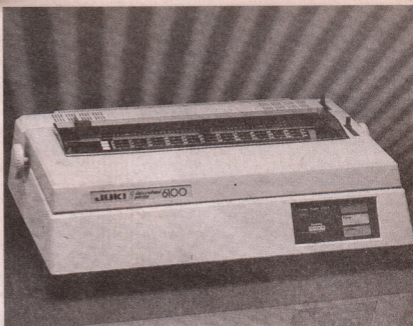
su 64 simbola od kojih je neke vrlo teško protumačiti (gledajući ih, autor ovoga teksta dolazi na smešnu pomisao da su dizajneri želeli da omoguće igru „vešanje“ (hangman), pa su se potrudili da pripreme karaktere u vidu čovečuljka kome nedostaje pola tela, samo ruka... kao i karakter koji bi lepo izgravao scenu sa kraja igre). Niko se, naravno, neće buniti zato što mu je set karaktera povećan, ali vlasnici MC80, za razliku od vlasnika GP100A i FX80, nemaju mogućnost da se dopune sopstvenim karakterima, što je ozbiljan hendikep: šta ako su nam potrebna naša latinična slova? Odgovor na ovo pitanje možemo da damo samo onima koji poseduju programator EPROM-a: dovoljno je da izmenite deo štampačevog generatora karaktera; kodiranje je vrlo jednostavno, praktično kao kod računara „galaksija“ (pogledajte „Računare 1“ i tekst „Prvih nekoliko mikrosekundi“).

Pored kompletnog seta karaktera, MC80 ima i nekoliko veličina slova: u redu možemo da štampano 132, 80 ili 40 karaktera (takozvane „duge linije“) odnosno 105, 64 ili 32 karaktera („kratke linije“). Da li je ovakva raznovrsnost potrebna? Možda i nije, ali su na ovaj način pomireni mnogi standardi: štampanje 132 ili 80 karaktera u redu, kopiranje TRS-ovog ekrana (64 slova u redu) i, naravno, listanje „galaksijinih“, „Spectrumovih“ itd. programa (u doba kada je MC80 projektovan, bilo je relativno malo računara koji imaju 32 karaktera u redu, ali se dalekovidost firme Oki danas pokazuje u punom sjaju). Brzina pisanja retko prelazi 50 karaktera u sekundi (zavisí od broja slova u redu i razmaka redova i može da bude najviše 80 CPS) što je umereno sporo ali prihvatljivo.

MC80 nema grafiku visoke rezolucije, što znači da korisnik jedino može da kontroliše grafičke blokove koji su veliki koliko šestina svakog karaktera. On je, dakle, više nego nepogodan za vlasnike BBC-ja, Electrona, Commodora 64 i za one vlasnike Spectruma kojima štampač treba da služi za kopiranje silka sa ekrana.

I pored toga što mu nedostaju kodovi za kontrolu grafike visoke rezolucije, MC80 ima više kontrolnih kodova od Seikosha GP100A, koji je u ovom prikazu već okarakterisan kao „neinteligentan štampač“. Znači li to da je MC80 „pametniji“? Izgleda da jeste: iako ni on ne može da prepozna kraj strane (zar je toliko teško smestiti u RAM štampača jedan brojač redova?), MC80 nepogrešivo detektuje trenutak kada je list običnog papira pri kraju (tada za njim ne sledi još jedan list) kao i trenutak kada rolna kompjuterskog papira spadne na malu debljinu. Osim toga, na njemu se nalazi taster pomoću koga štampač prelazi u mod OFF LINE, tj. kada kompjuter mora da čeka na našu intervenciju da bi nastavio da štampa. Takva mogućnost nedostaje Seikoshi — kada je baš neophodno da štampač „zakloči“ računar, pribegava se njegovom isključivanju čime se, jasno, gubi sve iz internog batera.

Uz MC80 se dobija dobro uputstvo na 56 strana; u njemu se relevantne činjenice



### Juki 6100

Tip: lepeza  
Matrica: —  
Interfejs: centronics  
Memorija: 2000 znakova  
Linija po inču: 6,8  
Karakter po inču: 10, 12, 15  
Brzina štampanja: 18 karaktera

Karakter u redu: maksimalno 165  
Broj kopija: 4  
Podvlačenje: da  
Isticanje: da  
Proporcionalni razmak: da  
Blok grafika: ne  
Grafika visoke rezolucije: ne

Dvosmerno štampanje: da  
Logično traženje: da  
Maksimalna širina papira: Pomeranje papira: frikciono  
Dimenzije: 454×520×151  
Masa: 14 kg  
Cena: 400 funti

ponavljaju i po nekoliko puta, ali je to po našem mišljenju, velika pogodnost za početnike kojima je često teško da prate terminologiju, pogotovu kada je tekst pisan na stranom (engleskom) jeziku.

Sve u svemu, stekli smo utisak da je Microline 80 dobar štampač, ali to ne znači da vam ga pretoplo preporučujemo: njegova cena od oko 270 funti nije toliko manja od Eponsovih 320 funti (obe cene bez uračunatog interfejsa) da bi se čovek opredelio za ipak daleko slabiji uređaj. Došlo je vreme računara sa visokom grafičkom rezolucijom i šteta je da tu rezoluciju ne prati i printer!

## Štampači sa lepezom

Štampači sa lepezom (daisywheel) do skoro su bili preskupa alternativa brzim i pogodnim matricnim modelima. U poslednje vreme su razvijeni jeftiniji „margarita štampači“ (popularni naziv za štampače koji proizvode tekst okretanjem specijalne lepeze sa slovima) koji daju izvanredan izlazni dokument bolji nego na električnoj pisačkoj mašini. Ovi štampači danas koštaju svega 300—500 funti, što nije mnogo više cene od one koju plaćamo za neki matricni štampač. Pitanje je, dakle, za kakav se printer odlučiti?

Pre nego što pokušamo da odgovorimo na njega, prikazaćemo, bar kratko, nekoliko štampača sa lepezom: Brother HR15, Silver Reed EX44, Juki 6100 i DataLine 1200 DL.

Pomalo je paradoksalno reći da najbolji kvalitet pisanog teksta daje najjeftiniji od pobrojanih štampača — Silver Reed EX44 koji košta 300 funti. Za ovaj novac se, zapravo, dobija pisača mašina koja je opremljena interfejsom za povezivanje sa računom, što je idealno za one koji se tek privikavaju na korišćenje tekst-processora.

EX44 je, na žalost, prilično spor štampač — njegovih 12 karaktera u sekundi postize i prosečan daktilograf. Ovakvo mala brzina je rezultat kompromisa mogućnosti/cena: EX44 ima bafer od samo 15 bajta, štampa samo u jednom smeru i nema mogućnosti logičkog traženja. Sve u svemu, odličan štampač za one kojima se mnogo ne žuri!

Juki 6100, kao i mnogi dobri i jeftini uređaji, dolazi iz Japana. Njegova cena, kada je opremljen interfejsom i potrebnim kablovima, nešto je manja od 400 funti. Štampač je brz i prethodnog (18 znakova u sekundi) što je dopunjeno mogućnošću dvosmernog štampanja i internim baferom od 2 kilobajta. Osim ispisivanja normalnih slova, Juki 6100 omogućava proporcionalno razmicanje slova i slične pogodnosti koje imaju daleko skuplji štampači. Na žalost, mehanizam koji okreće traku nije najsrećnije rešen, pa se mastiljava traka ponekad zamotava oko njega. Druga loša strana Jukija je što je neprijatno bučan, što se tek delimično rešava poklopcem koji se uz njega dobija.

Čini nam se da je najbolji izbor DataLine 1200 DL. Očekuje se da će njegova cena biti znatno povoljnija (sada 450 funti) što bi,

# U pripremi RAČUNARI 3 u vašoj kući

## • HAKERSKI MANIFEST

šta piše u jednom papiru koji tajno kruži među programerima sveta — šta jednom programu dolikuje a šta ne i kako prepoznati pravog programera

## • RAČUNARI U IZLOGU

noviteti na evropskom tržištu — šta mogu, koliko koštaju a koliko vrede, kako se nabavljaju

## • RAČUNARI U AKCIJI — D-ISK-JEDINICE

disk jedinice su postale dovoljno jeftine i za kućnu primenu — kako izbor primeriti svojim potrebama i mogućnostima računara — karakteristike, kvalitet, cene, izbor, nabava

## • INTERFEJSI

izbor i povezivanje interfejsa za štampač sa računarem ponekad nije nimalo jednostavna stvar — kako se pišu softverski interfejsi — prikaz svih interfejsa za računar Spectrum

## • NEKE NOVE IGRE

prikaz najnovijih igara za računare Spectrum, Commodore 64 i Electron

## • SOFTVERSKA TRPEZA

uporedni tabelarni prikaz sistemskog i uslužnog softvera za računare Spectrum, Commodore 64, Electron i BBC B

## • SPEKTROTEKE

kako koristiti kućni računar za obradu podataka — prikaz svih datoteka za Spectrum — koja datoteka u kojoj prilici

## • ŠKOLA AVANTURISTIČKIH IGARA

kako se igrati a kako pišu igre-avanture

## • OPERATIVNI SISTEM/ADRESNI DEKODERI

prvi nastavak softversko-hardverske škole za buduće graditelje računara koja vas korak po korak praktično uvodi u hardversku i programsku koncepciju kućnog kompjutera

## • MAŠINAC ZA POČETNIKE

paralelna škola mašinskog programiranja u strip; za procesore Z80 i 6502 sa ilustriranim rečnikom menmonika

## • NAŠI PROJEKTI

— trotonski generator zvuka za računar „galaksija“

— uređaj za automatsko učitavanje EPROM-a prilikom uključivanja računara ili nakon hardverskog reseta

## • COMMODORE 64

specijalan prilog za vlasnike računara Commodore 64 koji se i kod nas polako probija ka vrhu rang-liste najpopularnijih računara

## • BIBLIOTEKA PROGRAMA

novi programi za računare „galaksija“, Spectrum, Commodore 64 i BBC/Electron

**... NIZ NOVIH TEKSTOVA O RAČUNARIMA „GALAKSIJA“ I SPECTRUM**  
Ukoliko želite da već sada obezbedite svoj primerak „Računara 3“, možete naručiti na adresu: „Galaksija“, 11000 Beograd, Bulevar vojvođanskih Mišića 17  
— isporuka pouzdom

kombinovano sa dobrim karakteristikama ovoga štampača, moglo da mu pomogne da postane bestseller. Maksimalna brzina štampanja je pristojnih 25 karaktera u sekundi, pri čemu štampač piše u oba smeru sa mogućnošću logičkog traženja. Bafer od 512 bajtova i Centronics interfejs omogućavaju povezivanje sa većinom postojećih računara: na štampanju ploči se nalazi određeni broj prekidača koji omogućavaju izbor brzine prenosa.

Čak i onima koji planiraju nešto da crtaju, 1200 DL može da posluži umesto matičnih štampača: njegova rezolucija od 165 karaktera na 11 inča (oko 28 cm) i minimalni razmak linija od samo 1/48 inča (0.021 cm) nisu karakteristike koje se tako lako mogu zanemariti.

Pre nego što se odlučite na kupovinu štampača sa lepezom, morate ozbiljno da razmisлите o jednom problemu: retko ko koristi štampač samo za pisanje teksta na stranom jeziku, što znači da printer bez slova Č, Ć, Š i Ž nema mnogo smisla. Kod ovih štampača slova ne mogu da se definišu, što znači da je potrebna „jugoslovenska lepeza“ koju, koliko znamo, niko ne može da vam isporuči. Jedan servis za preciznu mehaniku u Beogradu je kraće vreme pokušavao da prepravlja lepeze za naše uslove, ali su rezultati bili nezadovoljavajući. Neprijatno, zar ne?

Zamislite, osim toga, da ste napisali neki tekst od stotina strana, uređili ga pomoću

računara i poželeli da se ispišete na štampaču. Štampači sa lepezom, jasno, pišu na običnoj hartiji, što znači da će vam biti potrebno nekoliko časova sedenja pored opreme i menjanje listova svaka dva minuta! Ništa, međutim, nije savršeno: štampači sa lepezom jedini imaju poseban kvalitet otiska za profesionalnu primenu.

## Hewlett-Packard 2225

HP 2225 predstavlja prvi komercijalno najavljeni „inkjet“ štampač. Odakle mu ovo ime? Radi se o potpuno novom metodu štampanja teksta koji se zasniva na izbacivanju kapljice mastila na papir. Svako slovo je, naime, predstavljeno matricom od 11x12 tačaka. U glavi štampača nema pokretnih delova: to je jednostavno rešetko od 132 otvora iz kojih se nalaze mikro rezervoari sa mastilom i grejači. Kada štampač treba da ispiše neko slovo, neki od grejača zagrevaju mastilo koje se širi i kapljica izleće. Zatim nastupa praktično trenutno hlađenje, iz većeg rezervoara u glavu biva povučeno novo mastilo, a glava se, za to vreme, pomerila do sledeće pozicije na kojoj treba pisati.

U čemu je prednost ovakvog metoda? Pre svega, u glavi nema pokretnih delova, pa je štampač jeftin (500 dolara, što je neverovatno niska cena za firmu kao što je Hewlett-Packard) i, što je možda još važnije, praktično nečujan. Otvarena je, osim toga, i prenosivost — dimenzije HP 2225 su 8,9x29,2x20,6 cm, što znači da lako staje čak i u manju tašnu. Napajanje je na struju ili baterije, a firma tvrdi da jedan set baterija omogućava štampanje oko 200 (!) strana, što znači da je potrošnja više nego mala.

HP 2225 može da se pohvali i mnogim drugim dobrim karakteristikama: karakteri mogu da se definišu na matrici 11x12, u jedan red staju 40, 71, 80 ili 142 slova, što znači da je grafička rezolucija 192x96 tačaka po inču (u=2,5 cm) ili 1296 tačaka u red (može, dakle, da se štampa slika sa standardnog televizora ako se raspolaze pogodnom opremom). Set karaktera obuhvata preko 150 znakova zajedno sa slovima iz danske, škotske, i engleske, finske, francuske, nemačke, italijanske, norveške, španske i švedske abecede.

Kada su svojevremeno najavljeni u časopisima, mnogi su smatrali da „inkjet“ štampači nemaju šanse jer im je potreban skup „repromaterijal“. Hewlett-Packard je pokazao da nije baš tako: HP 2225 piše na specijalnom ili običnom papiru (u oba slučaja kvalitet je sasvim zadovoljavajući, s tim što se na specijalizovanom eliminiše razlijevanje mastila, pa je kvalitet uporediv samo sa matičnim nego i sa štampačima sa lepezom), dok se umesto trake menja čitava glava kada se mastilo iz nije istrošilo. To se događa posle ispisanih 500 strana (rukopisne knjige), a jedna glava košta svega 8 dolara — ne kupuje nego traka za matični štampač!

Cini nam se da HP2225 predstavlja izvanredan izbor, naročito za one koji su odlučili da kupe prenosiv i moćan računar kao što su HP75 ili novi HP71B (HP2225 može da se nabavi sa Centronics, HP IL ili HP IB interfejsom koji je uračunat u cenu od 500 dolara). Kada u bliskoj budućnosti neka druga firma proizvede sličan štampač, njihove cene će opasti, pa „inkjet“ modelima predviđamo veliku budućnost!

Dejan Ristanović

**Knjiga koja vas popularno uvodi  
u svijet kompjutora!**

**Peter Laurie**  
**KOMPJUTOR**  
**U KUĆI**  
**priručna enciklopedija**

**Najtraženija engleska knjiga o kompjutorima  
— sada i na našem jeziku!**

Iz sadržaja: — OPĆENITO O KOMPJUTORU — PROGRAMIRANJE — PRIMJENA U PRAKSI: Poslovni programi, kompjutor i slikarstvo, simulacije, manipulacija slike, kompjutorska animacija, kompjutor koji govori, kompjutorska glazba, robotika, roboti u industriji, programiranje prilagođeno za kućne kompjutore SINCLAIR itd. — POGLED U BUDUĆNOST — RJEČNIK POJMOVA — SHEMATSKÉ SKICE I UPUTE itd.

Temeljni priručnik za svakog koji namjerava nabaviti kompjutor za kuću, školu ili za poslovne potrebe

Bitne osobine:  
jasnoća i preglednost teksta,  
stručna pouzdanost objašnjenja  
i bogatstvo korisnog  
ilustrativnog materijala

Pretplatna cijena: 2.200.- dinara  
Prodajna cijena: oko 3.000.- dinara  
Izlazi iz tiska: 24. oktobra 1984.

**Narudžbenica Računari II**

Izdavačkoj jedinici „Cankarjeve založbe“  
Ilica 26, 41000 Zagreb

Ovime naručujem \_\_\_\_\_ primjeraka knjige KOMPJUTOR U KUĆI (izdanje na srpskohrvatskom) po pretplatnoj cijeni od 2.200.- dinara. Što ću platiti u pet mjesečnih rata po 440.- dinara. Knjigu ću primiti po izlasku iz tiska i uplati cjelokupne pretplatne cijene.

Ime i prezime naručitelja .....

Adresa .....

.....

..... Vlastoručni potpis

Napomena: pretplatiti se možete i na telefonski broj (041) — 432-325

 PARTNER

## PARTNER - SARADNIK U SAVREMENOM POSLOVANJU

ISKRA DELTA uvodi na jugoslovensko i strano tržište mikroračunar PARTNER, rezultat sopstvenog razvoja. Ne nudimo vam samo mašinsku opremu, nego i savremeno integrisano rešenje vašeg finansijskog, skladišnog i materijalnog poslovanja. Međutim, i pored svega toga, primena je tako jednostavna da možete da je savladate u jednom danu.



# galaksija

Računari  
i obrazovanje

# u školi

... u rukama Mandušice Vuka  
svaka puška biće ubojita!  
(Njegoš: Gorski vijenac)

## Brojni podaci

Obrazovanje u oblasti programiranja može biti zasnovano na izradi programa koji rade sa različitim objektima: brojevima, tekstom ili grafičkim objektima. Međutim, ipak se najčešće, za ovu svrhu, koriste brojevi. U jeziku za „Galaksiju“ brojevi se u internom kodu registruju u pokretnom zarezu. Međutim, za cele brojeve iz intervala -999999, 999999 korisnik ima utisak da radi sa celim brojevima, jer ih unosi kao cele, a računar ih izdaje u istom obliku. Za ostale brojeve koristi se pozicioni ili eksponencijalni zapis brojeva.

Na ovaj način „Galaksija“ omogućuje izučavanje svih elementarnih struktura, nad brojnim podacima, koje se javljaju u programu: linijske, razgranate i cikličke, kao i korišćenje potprograma. Posebno treba istaći da razgranata struktura, koja često u obliku dovodi do loše čitljivosti programa, vedne u naredbi IF ima opciju ELSE, što daje bolje strukturirane programe. Tako, program

```
10 INPUT X
20 IF X < 0 PRINT X * X : ELSE PRINT X
30 STOP
```

izdaje kvadrat unetog broja ako je uneti broj manji od nule, odnosno izdaje uneti broj ako ovaj nije manji od nule.

Od strukturiranih brojnih podataka postoji samo jednodimenzionalni niz, čiji broj elemenata zavisi od raspoloživog memorijskog prostora. Dakle, posebno se ne dimenzioniše. Za obuku u programiranju ovo se ne može uzeti kao neki bitan nedostatak, jednostavno iz razloga što je jednodimenzionalna struktura podataka, zapravo, prirodno najbliža fizičkoj strukturi memorije. Višedimenzionalne strukture, kao što su tabelle i matrice, mogu se obrazovati nad jednodimenzionalnim nizom i predstavljaju lepe zadatke za vežbe iz programiranja.

Od elementarnih funkcija postoji samo celobrojna funkcija (INT), dok se za trigonometrijske, eksponencijalnu i logaritamsku funkciju mogu napisati odgovarajući potprogrami na jeziku. Pisanje ovih potprograma na raznim obrazovnim nivoima može imati različiti smisao, od programiranja potprograma i izgradnje biblioteke potprograma, do numeričkih problema aproksimacije funkcija algebarskim izrazima u kojima se javljaju samo aritmetičke operacije sabiranja, oduzimanja, množenja i deljenja.

Primena računara na rešavanje numeričkih problema može se u školama tako

demonstrirati od najnižeg do najvišeg obrazovnog nivoa. Ovde se kao poseban izazov javlja traženje efikasnijih numeričkih algoritama, što „galaksija“ omogućuje merenjem vremena rada programa, korišćenjem programski kontrolisanog časovnika. Evo jednostavnog primera: sastavite program koji određuje broj šestocifrenih prirodnih brojeva koji zadovoljavaju uslov da su im zbrovi tri cifre veće i tri cifre manje težine jednaki. Ako program zasnujete na ideji da se ispituju svi šestocifreni prirodni brojevi trajanje na računaru oko 15 časova. Međutim, ostavljamo čitaocu da nađe efikasniji algoritam koji će se na računaru izvršiti za samo 75 sekundi!

Ukratko, jezik na računaru „galaksija“ pruža mogućnosti obuke u programiranju, kao i u pripremi računara u oblasti jednostavnijih naučno-tehničkih izračunavanja.

## Rad sa tekstom

Praktično, pri svakoj upotrebi računara korisnik ima potrebu da u manjoj ili većoj meri radi sa tekstom. Naravno, postoje takve primene računara u kojima se isključivo radi sa tekstom, kao što je izučavanje prirodnog jezika, prevodjenje sa jednog jezika na drugi, u velikom broju informacionih sistema i sl. Rad sa tekstom treba da omogući: • prirodni dijalog korisnika sa računarem, • uređenje teksta, • pretraživanje teksta i • unošenje i izdavanje teksta.

U interaktivnom radu sa računarem korisnik najčešće odgovara na poruke koje izdaje računar. U ovom slučaju, tekst služi da podrži komunikaciju računara sa korisnikom. Na primer, u jednom programu za obračun kamate računar izdaje pitanje:

KAMATA U PROCENTIMA? 30.

a korisnik unosi vrednost kamate, u ovom primeru 30%. Izdavanjem pitanja korisniku se olakšava korišćenje programa, jer bi bez pitanja mogao da pogreši u redosledu unošenja podataka, što bi dovelo do pogrešnih rezultata u programu. Naravno, isto tako je važno, a možda još značajnije, da računar uz brojeve, koji su rezultati rada programa, izda i tekst koji objašnjava rezultate. Na primer, već pomenuti program za obračun kamate može izdati rezultate u obliku:

```
PRETHODNO STANJE: 20000
KAMATA: 6000
NOVO STANJE: 26000
```

Ovo je očigledno znatno čitljivije nego da su izdati samo tri broja 20000, 6000 i 26000. „Galaksija“ u potpunosti omogućuje unošenje i izdavanje teksta u smislu podrške dijaloga između korisnika i računara.

Neki mikroročunarski sistemi su orijentisani na rad sa tekstom, pa omogućuju

uređenje i pretraživanje veće količine teksta. „Galaksija“ nije takav sistem iz dva razloga: • u jednom redu na ekranu se izdaju 32 znaka, i • ne postoje mala slova.

Izborom rešenja sa 32 znaka u redu postignuta je dobra čitljivost brojeva i teksta na ekranu bilo kojeg televizora. Ovo je onemogućilo ozbiljniji rad sa tekstom, pa nije imalo smisla uvoditi malo slova, jer bi to samo nepotrebno poskupelo sistem. Međutim, „galaksija“ ipak pruža izvesne mogućnosti koje su značajne u obuci programiranja, a odnose se na rad sa tekstom. Tako se može vršiti dopisivanje teksta (konkatenacija) i ispitivanje na jednakost. Neke dalje mogućnosti mogu biti ostvarene izgradnjom biblioteke potprograma. Tako, sledeći potprogram vrši poređenje dva azbučna (znakovna) podatka koji su susedni elementi azbučnog niza:

```
3000 FOR K=0 TO 15
3010 A=BYTE (PTR X$(I)+K)
3020 B=BYTE (PTR X$(I+1)+K)
3030 IF A < B ELSE K=0:RETURN
3040 IF A < B K= -1:RETURN
3050 IF A < B K=1:RETURN
3060 NEXT K:K=0:RETURN
```

Rezultat poređenja se postavlja kao vrednost promenljive K na sledeći način: K dobija vrednost -1, 0 ili 1, prema tome da li je X\$(I) redom manje, jednako ili veće od X\$(I+1). Gornji potprogram se može upotrebno koristiti za uređenje azbučnih nizova podataka u azbučni redosled, što svakako treba da je prisutno u svakom kursu iz programiranja.

## Grafika na računaru

Većina mikroročunarskih sistema ima određene grafičke mogućnosti, koje se sa stoje u crtanju na standardnom ekranu crno-belog ili kolor televizora. Crtanje se postize tako što se izvestan broj tačaka na ekranu nalazi pod programskom kontrolom. Prema tome koliki je ovaj broj tačaka, grafika se deli na grubu i finu grafiku. U gruboj grafici, programski se kontroliše nekoliko hiljada tačaka, a u finoj nekoliko desetina hiljada tačaka, pa i više. „Galaksija“ raspolaze grubom grafikom sa  $64 \times 48 = 3.072$  tačke na ekranu. Iako su ovo vrlo skromne grafičke mogućnosti, njihov postojanje čini „galaksiju“ znatno interesantnijom i u obrazovanju iz sledećih razloga: • daje mogućnost upoznavanja učenika sa grafičkim mogućnostima računara, • znatno proširuje fond zadataka interesantnih za programiranje, • daje mogućnost programiranja nekih video igara, što posebno doprinosi interesovanju mladih za programiranje.

Da bismo ilustrovali kako i vrlo jednostavna rešenja mogu da proizvedu intere-



**Mikroračunari nalaze sve veću primenu u raznim oblastima. Jedna od vrlo značajnih primena mikroračunara jeste i obrazovanje. Na svetskom tržištu postoji bogat izbor „obrazovnih“ paketa, ali je cena ovih uređaja veoma značajna, posebno u našoj sadašnjoj ekonomskoj situaciji. Može li, međutim, biti kakve koristi u školama i od mikroračunara minimalnih mogućnosti? Ovoj klasi pripada i mikroračunar „galaksija“: lako skromnih hardverskih potencijala, „galaksija“ omogućuje rad sa brojevima, tekstom i grafikom. Koje su njene stvarne mogućnosti kao učila za programiranje, a koje kao sredstva za unapređenje nastave?**



Prva lasta: Sa računarnom „galaksija“ počinje da se ostvaruje ideja o uvođenju računara i u naše škole

santhe crteže na ekranu, posmatrajmo sledeći program:

```

100 IDEMONSTRACIJA GRAFIKE
110 HOME
120 X=INT(32*RND)
130 Y=INT(24*RND)
140 FOR I=X TO 63-X
150 FOR J=Y TO 47-Y
160 IF DOT I, J UNDOT I,J:ELSE DOT I,J
170 NEXT J:NEXT I
180 FOR I=1 TO 2000
190 IF KEY(4) GOTO 220
200 NEXT I
210 GOTO 120
220 FOR X=0 TO 63
230 FOR Y=0 TO 47
240 IF DOT X, Y UNDOT X,Y:ELSE DOT
X,Y
250 NEXT Y:NEXT X
260 GOTO 260
    
```

Program bira, slučajnim izborom, tačku u levoj gornjoj četvrtini ekrana, a zatim, koristeći ovu tačku kao jedno teme pravougaonika, crta centralni pravougaonik na ekranu. Pravougaonik ispunjava tako da „pali“ ugašene tačke, a „gasi“ upaljene tačke. Posle iscrtanog svakog pravougaonika, program čeka 18 sekundi, a onda bira novu slučajnu tačku, itd. Ako korisnik želi da zadrži sliku, u pauzi između crtanja pravougaonika treba da pritisne taster D, posle čega nastaje završna obrada slike i program se završava. Pri svakom izvršavanju programa dobija se drukčija slika na ekranu.

## Mašinski jezik

Profesionalno obrazovanje u oblasti računarstva ne može se u potpunosti ostvariti bez prisustva mašinskog nivoa programiranja. Zapravo, tek upoznavanjem mašinskog jezika i arhitekture konkretnih računara na

pravi način se sagledavaju i mogućnosti računara. U slučaju „Galaksije“ može se upoznati mašinski jezik mikroprocesora Z80A, koji je, inače, često korišćen osmo bitni mikroprocesor. Manji programi na mašinskom jeziku mogu biti pisani u heksadekadnom obliku. Unošenje ovakvih programa najbolje je organizovati u obliku programske datoteke. Na ovaj način će se lako vršiti ispravke u programu. Ilustrujemo ovo sledećim primerom:

```

10 FOR I=&2C3A TO &2C48
20 TAKE A
30 BYTE I,A
40 NEXT I
50 INPUT X$
60 BYTE &2C41, BYTE(PTR X$)
70 X=USR (&2C3A)
80 GOTO 80
90=&01, &0C, &02, &21, &00, &28, &36,
&41
95=&23, &0B, &CB, &76, &28, &F8, &C9
    
```

Pre unošenja programa u memoriju treba применiti komandu NEW 15 da bi se oslobodio prostor za smeštaj potprograma na mašinskom jeziku. Startovanjem programa vrši se prenošenje mašinskog potprograma iz programske datoteke u memorijski prostor od adrese &2C3A do &2C48, a zatim uneti znak sa tastature (naredba 50) postavi se u adresu &2C41 i pređe na izvršavanje mašinskog potprograma koji upiše uneti znak tako da se izda u svim pozicijama na ekranu.

Pisanje ozbiljnijih programa na nivou mašinskog jezika zahteva simbolički jezik, a za ovo je neophodno raspolagati odgovarajućim assemblerom za „galaksiju“. Razvoj ovakvog assemblera nije veći programski problem, pa će, nesumnjivo, uskoro biti na raspolaganju i za „galaksiju“.

## Proširenja „Galaksije“

Većina mikroračunarskih sistema je otvorena prema raznim vrstama softverskog i hardverskog proširenja pa i računar

„galaksija“. Osnovna varijanta „galaksije“ može se programskom nadgradnjom učiniti moćnijom za razne primene. Ovo se lako postiže izgradnjom biblioteka potprograma koje podržavaju neke specifičnosti pojedinih primena, kao što je biblioteka lemenih matematičkih funkcija, rad sa tekstom, grafički rad, kao i izrada najzavrnijih aplikacionih programa, posebno za potrebe obrazovanja.

Hardverska proširenja treba orijentisati, pre svega, na one delove koji baziraju na elektronskim komponentama, kao što je proširenje RAM-memorije, fina grafika, ton i sl. Veza računara sa spoljnim svetom, koja omogućuje primenu u manjim eksperimentalnim situacijama, bila bi takođe vrlo značajna u nekim obrazovnim sadržajima. Posebno bi, svakako, bilo korisno povezivanje električne pisače mašine sa „galaksijom“, pre svega za potrebe štampanja programa.

## Prvi korak

Osnovna varijanta „galaksije“ sa 4 KB ROM i 4 KB RAM-memorije predstavlja mikroračunar koji u obrazovanju obezbeđuje:

- zblizavanje učenika sa računarnom,
- obuku u programiranju na jeziku,
- upoznavanje grafičkih mogućnosti računara na nivou grube grafike, i upoznavanje mašinskog nivoa programiranja, i obuku u programskim tehnikama koje su od značaja za navedene nivoe programiranja.

Prema tome, svi kursevi informativnog karaktera iz oblasti računarstva, bez obzira na obrazovni nivo, mogu se zasnovati na „galaksiji“. Kada učenik savlada programiranje na „galaksiji“, može pisati programe za prenošenje stečenih znanja iz raznih predmeta na računar. Ako bi profesori i bili u stanju da postavе ovakve programske zadatke učenicima i da nadgledaju njihov rad, to bi bio izvanredan način za kreativnu i kritičku primenu stečenih znanja.

Autor ovog teksta veruje da bi ovakva primena računara u školama doprinela motivisanosti učenika za sticanje novih znanja, a time i unapređenju nastave iz opštih i stručno-obrazovnih predmeta. Pisanje programa je kreativan posao u kojem projekatn program unosi svoja znanja o određenoj temi i rezultate svog rada daje na korišćenje drugima. To je intelektualna aktivnost čija je obrazovna vrednost najveća za one koji pišu programe, a znatno manja, često i nikakva, za one koji samo koriste rezultate rada tuđih programa. „Galaksija“ pruža mogućnost da se prvi korak u primeni računara u nastavi učini, najzad, i u našim školama.

Prof. dr Nedeljko Parežanović

# moj Programiranje za početnike

## prvi program

### Škola programiranja u 100 lekcija

Šta, najpre, znači „znati bežik“? Pod ovim terminom se podrazumeva poznavanje određenog broja naredbi i njihovog dejstva, razumevanje strukture programa i potprograma i, uopšte, sposobnost da se dati algoritam pretvori u program za neki računar. No, programiranje je nauka koja se nikako ne može svesti na pretvaranje gotovog algoritma u program — većina je napravila algoritam ili, kada dalje napredujemo, pronaći problem koji može algoritamski da se reši, pa ga onda reši. Ova „Škola programiranja“ ne može, obzirom na prostor koji zauzima, da zameni čitave biblioteke knjiga posvećene onima koji uče programiranje, profesionalno ili amaterski. Njen zadatak je, kroz sastavljanje jednog programa koji će prolatiti kroz nekoliko etapa, da vas podstakne na dalji samostalni rad i istraživanje. Ne treba, kao i obično, da očekujete da ćete program koji sastavimo moći koristiti da upotrebite — koristan program može da bude toliko stavan da nije pogodan za početnika. No, ukoliko budete pažljivo čitali sledeće redove, verovatno ćete se priseliti nekog problema iz vašeg svakodnevnog života koji može da se reši primenom računara i skupiti hrabrosti da se posvetite njegovom rešavanju. Ukoliko se to zaisla dogodi, ova „Škola“ je potpuno ispunila svoj cilj.

Programi koje budemo pisali biće namenjeni računaru „galaksija“, pa će njeni vlasnici imati lepu privilegiju da ih ukucavaju bez ikakvih izmena. No, „Škola“ je u suštini nezavisna od računara, pa ćete moći da je pratite i ako ste vlasnik Spectruma, ZX81, pa čak i (premda uz malo problema) nekog džepnog računara sa mogućnošću programiranja. Ukoliko imate problema sa razumevanjem neke naredbe koja vam izgleda kao „specijalitet“ računara „galaksija“, uzmite uputstvo za njegovu upotrebu (jedan od razloga što smo se opredelili za ovaj računar je u tome što svi posedujuće njegovu dokumentaciju) i ušetite se time da se prevodenjem programa mnogo što-šta nauči.

Ukoliko smatrate da se dobro upućeni u programiranje, možda ćete ipak poželeti da pročitate ovi „Školu“. Algoritmi i tehnike koje smo koristili su za početnike dobri jer su jednostavni. U mnogim slučajevima postoje bolja rešenja — na nama je da ih pronađete.

## Cas 001 — Algoritmi i programi

Pre nego što počemo sa pisanjem prvog programa, moraćemo da pogledamo

kako sve to izgleda na daleko jednostavnijim primerima. Tako će, doduše, naš prvi program biti naš treći program, ali — nigde ne žurimo.

Reč „algoritam“ potiče od imena uzbekog (autor ovog teksta mora da prizna da mu reč „uzbeskog“ ne znači baš mnogo) matematičara Al Horezmija koji je živeo u IX veku naše ere. Ovaj matematičar neobičnog umerja je definisao pravila za izvršavanje četiri osnovne računске radnje (pravilo za deljenje je bilo znatno primitivnije od onoga koje danas poznaju učenici nižih razreda osnovne škole) i tako postao začetnik jedne danas neobično popularne grane matematike koja se naziva „teorija algoritama“.

Verovatno je za većinu naših čitalaca potpuno besmislena egzaktna definicija algoritma koju matematičari danas usvajaju (samoo potpuno isti radi, najmanje „strašna“ definicija algoritma je „skup instrukcija koje može da izvrši Postova mašina“). Ipak, pojam algoritma je veoma lako shvatljiv: svakoga dana izvršavamo hiljade algoritama a da toga uopšte nismo ni svesni. Jeste li, na primer, nekada čekali autobus 84? Ako jeste, ovo algoritma koji ste tada primenili:

1. Dodaj na stanicu.
2. Da li je na stanici neki autobus? Ako jeste, pređi na korak 3 a ako nije — ponovi korak 2.
3. Pogledaj broj autobusa.
4. Ako je taj broj 84, pređi na korak 5. U protivnom vrati se na korak 2.
5. Udi u autobus.
6. Kupi kartu.
7. Poništi kartu.
8. Da li se autobus kreće? Ukoliko se kreće, vrati se na korak 8 a ukoliko miruje — pređi na korak 9.
9. Da li autobus stoji na stanici? Ukoliko je odgovor da, pređi na korak 10; u protivnom, vrati se na korak 8.
10. Da li je to stanica na kojoj treba da izadeš? Ako nije, vrati se na korak 8, a ako jeste — izadi iz autobusa.
11. Stigao si na cilj.

Kao što vidimo, algoritmi se formulišu „po tačkama“ (koracima). Svako od njih je obavezno dat broj na koji se drugi koraci, ako je to potrebno, pozivaju. U okviru svake tačke dato je neko precizno uputstvo koje treba izvršiti. Za formulisanje algoritma neobično je važno da poznavanje naredbe koje onaj kome je uputstvo namenjeno (računar, robot, drug kome objašnjava gde stanuju...) u ime da izvrši. U gornjem primeru korak 7 glasi: poništi kartu. No, moguće je da onaj kome je algoritam namenjen, jednostavno, ne zna što to znači. Njemu bi bilo nužno objasniti kako li da pronađe automat za poništavanje karata, kako da okrene kartu pre nego što će je staviti u prorez automata, koliko treba da čeka dok se automat ne oglasi zvonom, šta

da radi ako automat ne funkcioniše... Obzirom da nas najviše zanima pisanje programa za računare, algoritamski koraci će najčešće biti ekvivalentni naredbama nekog programskog jezika, u našem slučaju bežika.

Vreme je da napišemo i mali program. Obzirom da ne posedujemo robota koji bi putovao autobusom 84, moraćemo da se okrenemo nečemu mnogo prozaičnijem — brojevima. Možda ste čuli za problem nazvan „Ulamovo nagađanje“ (Ulam's Conjecture). Ako niste, pomoći ćemo vam da nagađate i to uz pomoć računara.

Evo najpre algoritma koji je, za početak, napisan samo rečima. Ulazna veličina je bilo koji prirodan broj  $n$ .

1. Da li je broj  $n$  paran? Ako jeste, pređi na korak 3, a ukoliko nije na korak 2.

2. Pomožni  $n$  sa tri i dodaj jedan rezultat. Zbir nazovi  $n$  (time je stara vrednost  $n$  „zaboravljena“). Pređi na korak 1.

3. Podeli  $n$  sa dva i količnik ponovo nazovi  $n$ .

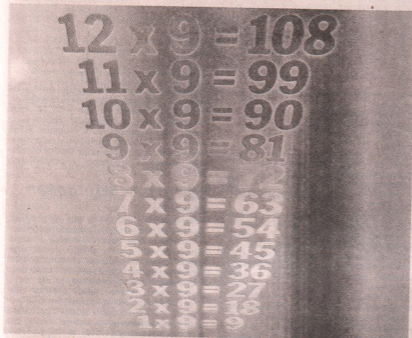
4. Da li je  $n$  jednako jedan? Ukoliko nije, pređi na korak 1, a ukoliko jeste završi sa radom.

Algoritam je prilično jasan, pa reč „nagađanje“, izgleda, nije pogodna za njegovo ime. No, nije tako. Vidi se da se polazi od broja  $n$ , i da se on neprekidno deli sa 2 (ili množi sa tri, pa se rezultatu dodaje jedan) sve dok se ne dobije broj 1. Pitajte je da li će, bez obzira na broj  $n$  od koga smo počeli, uvek na kraju da se dobije 1, ili postoji neki broj  $n$  za koji jedinica nikada neće biti dostignuta. Poljski profesor matematike Ulam je pretpostavio da se, za svako polozno  $n$ , na kraju dobija 1, ali mu nije pošlo za rukom da to i dokaže što, bar za sada, nije uspeo ni jednom drugom matematičaru (za egzaktno rešenje problema je i dalje raspisana privlačna novčana nagrada) pa je ostalo samo da se — nagađa. Sastavimo, dakle, program koji transformiše neko  $n$  u broj 1 i pokušajmo da pronađemo neko  $n$  za koje se to neće dogoditi!

Na početku rada treba da zatražimo od korisnika da unese neko  $n$ . Za to nam, jasno, služi naredba INPUT. Zatim treba da ispitamo da li je  $n$  deljivo sa dva (broj je paran ako je deljiv sa dva). To ćemo učiniti tako što ćemo  $n$  najpre podeliti sa dva i naći oco delo rezultata koji ćemo nazvati  $m$ . Zatim ćemo ispitati da li je  $m$  jednako  $n/2$ . Ukoliko je odgovor na ovo pitanje potvrđan,  $n$  je paran broj (na primer  $INT(4/2)=4/2=2$ ), a ako je određen —  $n$  je neparno ( $INT(5/2)=5/2=2.5$ ). Osta-tak algoritma je prilično jasan — poseduje-mo bežik naredbe koje odgovaraju pojedini-m algoritamskim koracima.

```
10 INPUT N
20 M=INT(N/2)
30 IF M=N/2 GOTO 60
40 N=3*N+1
50 GOTO 20
```

„Osnovna škola bejzika u stripu“, koju smo objavili u našem prvom specijalnom izdanju o računarima, bila je namenjena onima koji u programiranju čine prve korake. Iako takvih sigurno ima dosta, još je više onih koji su upućeni u osnove bejzika ali ne znaju šta bi dalje radili sa njim. Ovo poglavlje je namenjeno njima.



60 N=N/2

70 IF N=1 PRINT „DOSTIGNUT BROJ 1“;  
ELSE GOTO 20

Čim startujemo program uobičajenim RUN, računar će ispisati znak pitanja i tražeći od nas da unesemo neki broj. Učinimo to i pritisnimo ENTER (ili RET — zavisi od tastature koju imamo) i, nekoliko trenutaka posle toga, primetimo da je računar završio posao i na to nas upozorio odgovarajućim komentarom. Ovim je, ujedno, završen i čas 001. Na kraju svakoga časa u pravoj školi su dopuštena pitanja. Mi, jasno, ne možemo da predvidimo sva pitanja koja će čitaocima ove „Škole“ „pasti na pamet“, ali možemo da odgovorimo na ona koja se nameću sama od sebe.

— Ja ne verujem računarima. Sve što ovaj program radi je da, posle dužeg ili kraćeg vremena, prikaže neki komentar na ekranu. Da li bi bilo moguće modifikovati program tako da računar prikazuje brojeve u nizu, sve dok ne prikaže broj 1?

• Nema većih problema: dovoljno je dodati naredbu 25 PRINT N i računar će u svakom prolasku kroz ciklus prikazivati broj iz niza.

— Da probamo... Radi, ali brojevi prebrzo prolaze — i dalje ne mogu da proverim da li računar dobro radi.

• I to može da se reši i to na dva načina: ukoliko, u toku izvršavanja programa pritisnemo i držimo DEL, sve će se „zamrznuti“. Ako ne želimo da se mučimo na ovaj način, dodaćemo naredbu 26 FOR I=0 TO 1000:NEXT I i računara će čekati posle prikazivanja svakog broja. Od interesa bi, međutim, bilo i prebrojati koliko je prolazaka kroz ciklus bilo potrebno da se dođe do kraja posla. Kako bi to moglo da se ostvari?

— Trebalo bi dodati jednu promenljivu koja bi se povećavala za jedan pri svakom prolasku kroz ciklus. Neka to bude promenljiva P, dodajmo, dakle, naredbu 27 P=P+1 i na kraju programa 85 PRINT P. Neka, N bude 5... Dobili smo da je računar prošao kroz ciklus, 4,5 puta! Da li je možda poslednji prolazak računat polovino?

• Naravno da ne. Jednostavno, promenljiva P na početku rada nije imala vrednost 0 već 0,5 (kod drugih računara bi možda bila prijavljena greška „No such variable at line 27“). Grešku lako ispravljamo dodajući naredbu 5 P=0.

— Da li je neko već eksperimentisao sa većim brojevima? Da li se uvek dobija rezultat 1?

• Pomoću velikih kompjutera isprobani su brojevi sa deset, petnaest i, u poslednje vreme, dvadeset cifara. Pokazalo se da svi

oni, posle duže ili kraće konvergencije, dolaze do broja 1. To, naravno, ne znači da jednog dana neće biti pronaden broj koji se ne računa više u 1 na opisani način, ali daje realne osnove za pretpostavku da je nagađanje prof. Ulama tačno.

— Da li naš program radi za tako velike brojeve (do 1 E38)?

• Veliki broj je, možda, neparan. U tom slučaju računar ga množi sa 3 i rezultatu dodaje 1. Ukoliko je zbir opet neparan, operacija se ponavlja, što znači da je program već udesetostručen. Može, dakle, lako da se desi da broj izide iz opsega u kome računar radi i da se pojavi greška pri zaokrugljivanju. Zato broj koji kucamo treba da bude bar 20 puta manji od maksimalnog celog broja koji „galaksija“ može da prihvati (vama ostavljamo da pronađete koji je to broj).

— Znači li to da računar „galaksija“ ne može da nam pomogne ako želimo da ispitujemo većne brojeve?

• Naravno da ne znači. Mogli bismo da smeštamo svaku cifru broja u element numeričke matrice A(I) i da sami sintetiše-mo računске operacije. No, to je problem kojim ćemo se baviti nekom drugom prilikom. Shvatili smo šta su program i algoritam, pa smo spremni da se uhvatimo u koštac sa problemom koji ćemo rešavati u ovoj „školi“.

## Čas 010 — Tablica množenja

Prethodni problem je bio jednostavan zato što nam je algoritam bio unapred dat i treba ga samo prevertiti u program. Došlo je vreme da se probamo i u sastavljanju algoritma: treba da napišemo program pomoću koga računar testira korisnikovo poznavanje tablice množenja. Ovaj program, naravno, nije namenjen nama koji ćemo ga pisati — znanje bejzika podrazumeva i neka najosnovnija matematička znanja. No, za osnove iz naše okoline ovaj program može da bude zabavan i, što je još važnije, koristan.

Pve svega, treba jasno da definišemo problem i odlučimo šta računar treba da radi. Testiranje tablice množenja se obavlja tako što računar zadaje probleme (problemi su tipa 2\*7=?), a zatim očekuje od korisnika da ih reši. Korisnik kuca odgovor, računar proverava njegovu ispravnost i izdaje odgovarajući izveštaj. Proveravanje rezultata ne bi trebalo da predstavlja poseban problem — svaki računar zna da množi, a naredba IF je kao stvorena za testiranje rezultata. Pravi problem je, dakle, postavljanje problema. Kako računar može da odluči koje će brojeve prikazati korisniku?

Prvo što ćemo pomisliti je da se u memoriji računara stave svi mogući parovi brojeva. Obzirom da će naš program testi-

rati poznavanje tablice množenja 10x10, „pamćenje“ 100 parova brojeva bi odnelo 800 bajta memorije, što i nije previše. U memoriju bismo upisali i redosled postavljanja problema: prvo, na primer, računar treba da prikaže osmi par brojeva, zatim pedeset šesti i tako dalje. To bi oduzelo još izvestan prostor, ali bi utrosak memorije i dalje bio prihvatljiv. No, računar bi zadavao probleme uvek istim redom, što znači da bi test, posle izvesnog vremena, postao nepouzdan. U takvom trenutku bismo morali da menjamo čitav program ugrađujući u njega novi redosled brojeva. To, po mogućstvu, treba da izbegavamo — jednom napisani program bi morao da bude u stanju da pouzdano radi bez ikakvih modifikacija.

Ideja koju smo imali, dakle, nije dobra. Kako da dodemo do dobre ideje? Najbolje je da razmislimo o načinu na koji bi čovek rešavao isti problem. Kako, na primer, učitelj testira poznavanje tablice množenja? Tako što kaže neka dva broja i od učenika zahteva da ih pomnože. Kako on dolazi do ta dva broja? U većini slučajeva kaže prve brojeve koji mu „padnu na pamet“ — ti brojevi su, dakle, slučajni.

Pojam 'slučajan broj' se vrlo teško može definisati. Što se naziva slučajnih brojeva tiče, njihova definicija je prilično jednostavna: to je grupa brojeva koji nisu povezani nikakvim međusobnim relacijama. Ukoliko ste proučili uputstvo za upotrebu vašeg računara i našu „Osnovnu školu bezjeka“, svakako ste primetili da je i računar „galaksija“ snabdevan naredbom RND koja služi za generisanje slučajnih brojeva. Kako je uopšte moguće da računar generiše slučajne brojeve kada su svi rezultati koje on daje stroga funkcija ulaznih veličina?

Slučajni brojevi koje računar generiše su, u stvari, pseudoslučajni. Među pseudoslučajnim brojevima postoje neke „tajne veze“, ali su one toliko složene da korisnik na osnovu prethodnih ne može nikako da pogodi dalje članove niza — za njega su, dakle, pseudoslučajni brojevi „dovoljno slučajni“. Formula koja povezuje članove niza pseudoslučajnih brojeva je najčešće tako konstruisana da se stalno dobijaju brojevi između 0 i 1 i to tako da je njihova srednja vrednost praktično 0,5 (što više brojeva generiše, to će njihova srednja vrednost biti bliža ovoj cifri). Za formiranje ovoga niza potreban nam je njegov prvi član da bi iz njega bio formiran drugi, iz njega treći i tako dalje. Taj prvi član niza se u stranoj literaturi slikovito naziva „seed“ (seed = seme) i generiše tako što računar, u nekom trenutku, analizira sadržaj svog internog časovnika ili nekog drugog registra koji se često menja. Računar „galaksija“ nije nikakav izuzetak — svaki put kada ga uključite, PRINT RND će dati drugi rezultat.

Slučajne brojeve u programu, dakle, generišemo naredbom RND. Možemo da napišemo A=RND i promenljiva A će dobiti neko slučajnu vrednost između 0 i 1. Naša tablica množenja treba, međutim, da testira množenje jedni jednocifrenih brojeva. Zato ćemo dobiti slučajni broj pomnožiti sa devet, uzeti ceo deo rezultata (tako dobjiamo ceo slučajan broj između 0 i 8) i dodati mu 1 (broj je sada između 1 i 9), na isti način ćemo da generišemo i drugi slučajan broj, a zatim ćemo, pomoću naredbe

10 I TABLICA MNOZENJA

20 I

30 T=0

40 N=0

50 P=0

60 HOME

70 PRINT AT 256, "TAČNI"; AT 265, "NETAČNI"; AT 276, "BEZ ODGOVORA"

90 PRINT AT 320, T; AT 331, N; AT 345, P

100 PRINT AT 0, "KOLIKO JE"

110 A=INT(RND\*99+1); B=INT(RND\*99+1); C=A\*B; PRINT

120 PRINT AT 9, A; " \*"; B; " ?"

130 INPUT X#

140 IF VAL(PTR X#)=C GOTO 300

150 N=N+1; PRINT AT 0, "NIJE TAČNO!"

240 PRINT AT 32; A; " \*"; B; " ="; C

250 FOR I=1 TO 500: NEXT I

260 GOTO 90

300 PRINT AT 0, "B R A V O !"

310 T=T+1

320 GOTO 250

PRINT, ispisati problem na ekranu. Evo i programa:

```
10 A=INT(RND*9)+1
20 B=INT(RND*9)+1
30 PRINT "KOLIKO JE":A;"*";B;"?"
40 INPUT C
50 IF C=A*B GOTO 90
60 PRINT "ODGOVOR NIJE TAČAN"
70 PRINT "POKUSAJ PONOVO"
80 GOTO 30
90 PRINT "BRAVO! TAČAN ODGOVOR"
100 GOTO 10
```

Ovaj program se neprekidno izvršava: računar zadaje probleme i daje informacije o tačnosti odgovora. Kako sve a naročito uvežavanje tablice množenja mora na kraju da dosadi, treba znati kako se prekid izvršavanje programa. Dovoljno je pritisnuti taster BRK (BREAK) i računar će ispisati uobičajeno READY. Nešto radikalnije rešenje predstavlja pritisak tastera RESET (posledice su iste) ili isključivanje računara iz mreže za napajanje, čime će i to program biti izguđen. I nije neki program, radićete, ali ga ipak nemojte brisati — radićemo još na njemu. Pitanja?

— U programu se dva puta ponavlja naredba I (ili B)=INT(RND\*9)+1. Da li može da se upotrebi potprogram?

— Ne može da se upotrebi potprogram koji bi racionalnije trošio prostor. Neki računari omogućavaju definisanje funkcija, što znači da bi naredbe 10 i 20 mogle da se zamene sa

```
10 A=FNANDN
20 B=FNANDN; trebalo bi, jasno, dodati i
1000 DEFNANDN=INT(RND*9)+1, ušte-
da memorije, dakle, ne bi bila značajna, a
program ne bi bio ništa razumljiviji.

```

— Čini mi se da program, ovako kako je napisan, nije dovoljno animirajući za dete i da ne daje realne rezultate koji bi govorili o njegovom znanju. Program, pre svega, postavlja više problema ali ne prebrojava tačne i pogrešne odgovore. Osim toga, vreme za razmišljanje nije nikako ograničeno, pa onaj ko sedi pred računarem ne mora ni da odgovara na pitanja — ne postoji način da se ustanovi koliko je problema bilo postavljeno!

— Tačno. Međutim, ovaj program predstavlja samo početak. Pri pisanju nekog složenog programa možemo da se opredelimo za jedan od dva pristupa: da pišemo onakav program kakav on na kraju i treba da bude ili da ga razvijamo „korak po korak“. Prvi pristup primenjuju iskusniji programeri koji cene svoje vreme. Drugi je pristupačan korisnicima koji tek počinju:

razvijajući program, oni imaju priliku da rešavaju probleme na koje nailaze jedan po jedan kako nailaze na njih i da se, kada ne mogu dalje da napreduju, zaustave na nekoj verziji programa i proglase je za konačnu. Radićemo, dakle, dalje na programu za testiranje poznavanja tablice množenja rešavajući probleme iz postavljenog pitanja obrnutim redom.

## Čas 011 -- kako ispisati tabelu?

Poslednja primedba je bila da program ne formira tabelu tačnih i netačnih odgovora i rubriku koja govori o vremenu razmišljanja korisnika. Tu tabelu ćemo brzo napraviti, ali moramo da se odlučimo na koji deo ekrana da je smestimo i kako da je formiramo. Najjednostavnije je da se u prvom redu ekrana nalazi problem koji dete treba da rešava, u drugom odgovor koji dete kuca a, nešto ispod toga, tabela koja ilustruje njegovu znanje. Neka ona počinje od devetog reda na čiji početak, kao što vidimo sa mape ekrana (strana 11 uputstva za upotrebu „galaksije“), dolazi sa PRINT AT 256.

Na samom početku programa moramo da, kako bi to programeri rekli, inicijalizujemo brojače. To će jednostavno biti promenljive T, N i P. Promenljiva T će brojati tačne odgovore; promenljiva N netačne a promenljiva P probleme koje dete nije rešilo u predviđenom vremenu. U početku problema nije ni bilo, pa će sve tri promenljive dobiti vrednosti nula za šta su nam potrebne tri naredbe...

— Da li su te naredbe zaista neophodne i onima koji nemaju „galaksiju“? Zapravo, T, N i P, obzirom da nisu korišćene, već nemaju vrednost nula?

— Kod „galaksije“ ne sme da se računa sa početnim vrednostima promenljivih (iskusniji korisnici će primetiti da sve promenljive na samom početku rada imaju vrednost 0.5i da se ona ne menja naredbom RUN). Neki računari (TRS 80, na primer) automatski dodeljuju vrednost nula svakoj promenljivoj kojoj nije dodeljena vrednost. Čak i ako posedujete računar kod promenljivih kojoj nije dodeljena vrednost. Čak i ako posedujete računar kod koga dodeljivanje vrednosti promenljivima na početku rada nije neophodno, programski „bon-ton“ zahteva da ga izvršite, jer će tako onaj ko analizira program na samom početku znati koliko promenljivih ima, koliko će one memorije zauzeti i, uopšte, šta da

```

30 T=0
40 N=0
50 P=0
60 HOME
70 PRINT AT 256,"TACNI";AT 265,"NETACNI";AT 276,"BEZ ODGOVORA"
80 FOR L=1 TO 20
90 PRINT AT 320,T;AT 331,N;AT 345,P
100 PRINT AT 0,"KOLIKO JE"
110 A=INT(RND*99+1);B=INT(RND*99+1);C=A*B;PRINT
120 PRINT AT 9;A;" *";B;
130 Y#="00:00:00";DOT#
140 INPUT X#
200 S=VAL(PTRY#*6)
210 IF S<25 GOTO 270
220 PRINT AT 0,"PREDUGO RAZMISLJAS!"
230 P=P+1
240 PRINT AT 32;A;"*";B;"=";C
250 FOR I=1 TO 500:NEXT I:NEXT L
260 GOTO 260
270 IF VAL(PTRY#)=C GOTO 300
280 N=N+1;PRINT AT 0,"NIJE TACNO!"
290 GOTO 240
300 PRINT AT 0,"B R A V D !"
310 T=T+1
320 GOTO 250

```

radi ako njegov računar ima „strožiji bejzik“.

Da nastavimo. Svaki put kada dete da tačan odgovor, računar treba da izvrši naredbu T=T+1 i tako poveća sadržaj promenljive T za jedan. Kada dete pogreši, računar će povećati sadržaj promenljive N, a kada prekorači predviđeno vreme — promenljiva P. Posle toga računar će izvršiti naredbe kojima će stari sadržaj tablice biti zamenjen novim koji je aktuelan.

— Da li je potrebno da najpre izbrisemo stari sadržaj tablice?

\* Nije. Koristećemo naredbu PRINT AT koja štampa ono što želimo na proizvoljnoj poziciji ekrana, ne obračunajući pažnju na to što se ovim ispisivanjem menja sadržaj koji se na toj poziciji već nalazio. U ovom slučaju ne može da se javi problem koji bi u nekim sličnim programima bio neprijatan: obzirom da se broj tačnih, pogrešnih i zakasnelih odgovora nikada ne smanjuje, ne može da se desi da od prethodnog sadržaja preostane neko slovo koje bi, zajedno sa novoispisanim, promenilo smisao poruke (ako preko broja 10 štampamo broj 9, dobijemo 90 a ne 9).

Program sa prve slike omogućava, osim svega što u prethodni, delimično ispisivanje tabele. Broj zakasnelih odgovora je, međutim, uvek nula, jer još nismo ni govorili o naredbama kojima bismo obezbedili da „galaksija“ meri utrošeno vreme.

— To bar nije problem. Pa „galaksija“ poseduje ugrađeni časovnik.

\* „Galaksija“ zaista poseduje veoma tačan časovnik koji ćemo u ovilj prilik svaka-koristiti. Da se podsetimo: časovnik je usko povezan sa promenljivom Y\$ koja, na početku, treba dodeliti vrednost „00:00:00“. Da aktiviramo časovnik, koristećemo naredbu DOT#, a da ga zaustavimo UNDOT#. Ove dve naredbe ne treba nikako mešati sa naredbama za crtanje po ekranu — zvezdica na kraju im je sasvim promenila smisao. Sve, bar za sada, izgleda

vrlo jednostavno: na početku aktiviramo časovnik i dodelimo mu nultu vrednost, a zatim, pošto dete da odgovor, proveravamo da li je prošlo više od, na primer, 25 sekundi. Problem, međutim, leži baš u ovom poređenju: kako da uporedimo broj preteklih sekundi sa 25?

— Ništa lakše — upotrebimo naredbu IF Y\$25 P=P+1...

\* Taman posla! Pre svega, Y\$ je alfanumerik, a 25 je broj, pa nema nikakvog smisla porediti ih. Mogli bismo, naravno, da napišemo i IF Y\$>.25" ali ni to ne bi bilo ništa bolje. Pre svega, u Y\$ nije smešten samo broj sekundi već i broj časova i minuta u formatu koji smo maločas opisali. Čak ni IF Y\$>.00:00:25" ne bi mnogo pomoglo, jer „galaksija“ ne omogućava poređenje alfanumerika uz pomoć simbola > i < — moguće je samo proveravati da li su sadržaji dve alfanumeričke promenljive jednaki i to pomoću funkcije EQ.

— Problem, dakle, ne može da se reši...

\* Naravno da može. „Galaksija“, reklosmo, ne može da poredi stringove, ali sasvim lepo poredi brojeve. Potrebno je, dakle, da pretvorimo sadržaj Y\$ u broj sekundi...

— Pa da! Za to služi naredba TIME opisana u poglavlju koje je namenjeno iskusnijim vlasnicima „galaksije“. Samo...

\* Samo što se radi o programu na mašinskom jeziku koji čak ni onima koji čitaju to poglavlje nije baš sasvim jasan! Ne, mi ćemo problem rešiti koristeći „galaksijin“ bejzik. On poseduje naredbu VAL koja pretvara alfanumerik u konstantu. Ukoliko, kako piše u uputstvu za upotrebu, upotrebimo naredbu A=VAL(PTRY Y\$) nećemo uraditi ništa naročito. Promenljiva A će zaista dobiti vrednost konvertovanog stringa u Y\$, ali samo do simbola : koji, za „galaksiju“, služi kao oznaka kraja izraza. Dobijemo, dakle, broj sati koji će uglavnom uvek biti nula. Nas interesuju sekundi koji su sedmi i osmi simbol promenljive Y\$. Zato ćemo upotrebiti VAL (PTR Y\$+7).

— Zar smemo da sabiramo Y\$ (alfanumerik) sa 7 (broj)?

\* Naravno da ne smemo. Međutim, ispred Y\$ se nalazi naredba PTR koja, kao

što znamo, daje adresu promenljive u memoriji (u našem slučaju, to je 10880, što bismo videli iz mape sistemskih promenljivih da znamo da pretvorimo broj 2A80 u dekadni zapis) tj. broj. Tom broju dodajemo sedam, što je sasvim regularno, a onda izračunavamo vrednost izraza koji je na tu adresu smešten.

— Mogli smo, dakle, da upotrebimo i naredbu A=VAL(10887)

\* Mogli smo, ali to nećemo da učinimo. Ovakvom izmenom bismo, doduše, uštedeli dva bajta, ali tada niko ne bi znao šta takva naredba radi a da ne računamo što bi, čak i opremljen mapom sistemskih promenljivih, pomislilo da se nalazi na pogrešnom putu, jer ni jedna sistemska promenljiva ne počinje od 10887. Ovakvo je svakome ko se upoznao sa „malim tajnama“ naredbe PTR sve sasvim jasno.

Program sa slike 2 ilustruje ono što smo do sada uradili. Treba obratiti pažnju da se program ovoga puta ne izvršava neprekidno, već se, zahvaljujući naredbi broj 80, postavlja samo 20 problema. Posle prikazivanja poslednjeg, „galaksija“ nalazi na naredbu 260 koju neprekidno izvršava ne kvareći tako izgled tabele. Rezultate koje je postiglo dete, jasno, ne može da „uništi“, bar ne po pravilima „fer-pleja“.

— ????

\* Pod fer-plejom ne podrazumevamo način koga bi se setilo čak i dete koje je dalo 20 pogrešnih odgovora — isključivanje računara iz napajanja. Osim toga, pritisak na BRK prekida izvršavanje programa. Čak i ako pretpostavimo da ovaj taster nije naročito upadljiv, taster sa RESET na zadnjoj strani — prosto mami da bude prisutan.

Testirajmo program.

— Pokušajmo, najpre, da damo tačan odgovor. \* Dobro je, kaže BRAVO. A po grešan? I to je u redu. Da sačekamo više od 25 sekundi... Nikada 25 sekundi nije ovoliko trajalo... Prošao je i minut i po i još ništa. Izgleda da program ne radi!

\* Pokušajmo sada da damo tačan odgovor. Evo, kaže da smo predugo razmišljali!

— U pravo vreme. Zašto to nije ispisao ranije?

Zato što mu nismo rekli da to uradi. Koristili smo naredbu INPUT koja nalaze računaru da čeka sve dok korisnik ne otkuca traženi podatak i da onda nastavi sa radom. Vreme, u međuvremenu, teče, što interni časovnik lepo registruje, ali to ni na koji način ne možemo da registrujemo dok računar ne nastavi sa radom. Pre nego što počnemo da radimo na ovom problemu, treba da primetimo još jedan: pokušajmo da sačekamo minut i deset sekundi.

— Računar kaže da je odgovor dobar, iako je vreme prekoračeno više nego dvostrukoo!

\* Problem je u tome što smo, ispitujući vreme, testirali samo sekunde. Kada minut prođe, sekunde ponovo počinju od nule. Mogli bismo, jasno, da proveravamo i minute, pa i časove, ako hoćemo da budemo sasvim dosledni, ali se ovim nećemo baviti. Ovaj problem će se rešiti sam od sebe kada rešimo onaj mnogo veći — kako da registrujemo sam trenutak kada vreme istekne. O tome, naravno, sledećeg časa.

## Čas 100 — Naredba KEY

Od naredbe INPUT, na zalost, moramo da odustanemo, jer ne možemo da je promenimo tako da vraća kontrolu našem programu da bismo kontrolisali vreme.

— Naredba INPUT je previše zgodna! Zar problem kontrole vremena ne može baš nikako da se reši?

\* Može, ali ne na nivou kojim se bavimo (iskusnijim korisnicima koji čitaju ovu školu a žele da provere svoja znanja o interaptu predlažemo da pokušaju da sa-stave odgovarajući mašinski program za modifikovano INPUT). Moraćemo, kao što rekodmo, da se odredimo INPUT-a i proučimo jedinu preostalu naredbu za testiranje tastature — naredbu KEY. Ona ima veliku prednost nad naredbom INPUT utoliko što se izvršava samo jedan trenutak — kada naiđe na naredbu IF KEY (1) GOTO 1000 računar proverava da li je taster A pritisnut i ako jeste ide na liniju 1000, a ako nije na sledeću: čekanja, i ni u kom slučaju, nema.

Moramo, pre svega, da razmotrimo koju grupu tastera treba testirati. Dete, po prirodi stvari, treba da otkuca neki broj, što znači da tastere sa slovima i specijalnim znacima možemo sasvim lepo da ignorise-mo. Ostaje da treba testirati samo tastere 0-9, odnosno, prema tabeli sa strane 21 uputstva. KEY (32) — KEY (41). Njih ćemo najbolje da testiramo u petlji, i to jedan po jedan, na primer ovako:

```
10 XS=""
20 FOR I=32 TO 41
30 IF KEY (I) ELSE NEXT I : GOTO 20
40 XS=XS+CHR$(I+16): GOTO 20

Najteže je shvatiti liniju 40. Na početku smo promenljivoj XS dodelili vrednost „praznog stringa“, tj. alfanumerika koji ne sadrži nijedan simbol. Cim otkrijemo da je dete pritisnulo neko slovo, treba da ga „prilepimo“ uz taj string koji će tako, iz trenutka u trenutak, postajati sve duži. Ovo dodavanje slova je, zapravo, sabiranje alfa- numerika. Ako, na primer, XS ima vrednost „GALAKSI“, Y$ vrednost „JA“ a mi izvrši-mo XS=Y$+YS, PRINTXS će dati rec GALAKSIJA koja je dobijena nedovezivanjem JA na GALAKSI.
```

Nadovezivanjem alfanumerika „0“ na XS bi moglo da se realizuje sa XS=Y\$+.0 ili, prema tabeli sa strane 18, XS=Y\$+CHR\$(48). Primitimo da, za testiranje tastera na kome je nula, koristimo IF KEY (32) i da je 48=32+16. U ciklusu koji smo maločas dali, promenljivoj I ima ulogu brojača koji ispuštuje tastere koji su pritisnuti. Cim naiđe na neki od njih, „galaksija“ na promenljivoj XS nadovezuje CHR\$(I+16) što predstavlja upravo ASCII kod pritisnutog tastera. Može da neće biti loše da ovo sami isprobate na nekoliko primeri i uverite se da program korektno radi. Da biste, međutim, ovo upošte mogli da uradite, treba da otkrijete trenutak u kome je korisnik završio sa kucanjem broja. Signal za to je, naravno, pritisak na LENTER, pa ćemo program dopuniti naredbama:

```
30 IF KEY (I) ELSE NEXT I : IF KEY (48)
GOTO 10 ELSE GOTO 20
100 PRINT XS : GOTO 10

Startovaćemo program u kucati proizvoljne brojeve. Po pritisku na ENTER, računar će prikazati broj koji smo otkucali i početi — Zašto ovaj program treba da bude ovoliko „temeljan“? Da bi neka cifra bila registrovana, treba pritisnuti neki taster i poduže ga držati pritisnutim. Sa druge strane, ovo dugo pritisicanje može da ima i neprijatnu posledicu da računar neki taster
```

```
10 ! TABLICA MNOZENJA
20 !
30 T=0
40 N=0
50 P=0
60 HOME
70 PRINT AT 256,"TACNI";AT 265,"NETACNI";AT 276,"BEZ ODGOVORA"
80 FOR L=1 TO 20
90 PRINT AT 320,T,AT 331,N,AT 345,P
100 PRINT AT 0,"KOLIKO JE"
110 A=INT(RND*99+1):B=INT(RND*99+1):C=A*B:PRINT
120 PRINT AT 9,A;":":B;":":C;"?"
130 M=0:X$="":Y$="00:00:00":DOT$
140 IF KEY(48)**(M>0) GOTO 270
150 IF KEY(29)**(M>0) GOTO 330
160 FOR I=32 TO 41
170 IF KEY(I) ELSE NEXT I:GOTO 200
180 X$=X$+CHR$(I+16):M=M+1:IF M>10 GOTO 220
190 PRINT AT 19,X$
195 NEXT I
200 S=VAL(PTR Y$+6)
210 IF S<25 GOTO 140
220 PRINT AT 0,"PREDUGO RAZMISLJAS"
230 P=P+1
240 PRINT AT 32;A;":":B;":":C
250 FOR I=1 TO 500:NEXT I:NEXT L
260 GOTO 260
270 IF VAL(PTR X$)=C GOTO 300
280 N=N+1:PRINT AT 0,"NIJE TACNO!"
290 GOTO 240
300 PRINT AT 0,"S R A V O !"
310 T=T+1
320 GOTO 250
330 M=M-1
340 BYTE PTR X$+M,0
350 PRINT AT 19,X$
360 GOTO 200
```

registruje dva puta, a mogućnost da stvar korigujemo pritiskom na ← izgleda, ne postoji.

■ Problem je u tome što „galaksija“ nije mnogo brz računar, a bežik nije mnogo brz jezik. Dok „Galaksija“ ne prođe kroz čitav ciklus i ne naiđe na pritisnut taster, može da prođe dosta vremena. Rešenje ovoga problema ne postoji — mogao bi jedino da se napiše poseban program na „mašincu“ koji, kao što rekodmo, izlazi iz okvira ove Škole. Što se tiče korigovanja — tu pomoći ima. Treba, najpre, da testiramo taster — (KEY (29)). Ako je on pritisnut, treba da promenljivoj XS „oduzmemo“ jedno slovo, što i nije tako jednostavno kako na prvi pogled izgleda. Treba, pre svega, da naučimo da „galaksija“ smešta alfanumerike u standardnom ASCII, zapisu pri čemu se, iza poslednjeg slova, nalazi 00 koja označava njegov kraj. Cilj je da tu nulu pomerimo za jedno mesto ulevo.

Da pristupimo memoriji treba nam, jasno, naredba BYTE i promenljivoj koja će nam govoriti koliko smo slova do tog momenta otkucali. To će biti promenljiva M koja će na početku dobiti vrednost 0 i koja će, zatim, biti povećavana kad god na XS nadovežemo neko slovo. Kada poželimo da „odsečemo“ poslednje slovo, izvršićemo M=M-1: BYTE PTR X\$+M,0 i tako skratiti XS.

— Ali ovom naredbom mi nismo „po-merili“ nulu za jedno mesto ulevo! Jednostavno smo na to mesto upisali nulu, ali se ona i dalje nalazi i na sledećem mestu — kraj stringa, tako, označavaju dva nula-bajta!

■ Tačno, ali to ovdje nije mnogo bitno. Prvi nula bajt je za „galaksiju“ oznaka kraja stringa, dok drugi ni u kojoj prilici ne

biva trefiran, jer, jednostavno, nije deo korisnog sadržaja promenljive XS.

Na trećoj slici je prikazan program koji ćemo nazvati konačnim. U njemu ima još nekoliko mesta koja vrede prokomentarisati. Promenljivoj M smo, osim za redakturu, upotrebili za neke druge stinice: „galaksi-ja“ najpre, proverava da li je otkucani string duži od deset slova. Ukoliko bi se dete igralo kucajući broj duži od 16 slova (po prirodi stvari, besmislen kao rezultat množenja dva dvoциfrena broja) kolika je maksimalna dužina „galaksijinoj“ alfanumerika, mogli bi da nastupe razni problemi. Uz ovaj test računar će svaku ovakvu igru proglasiti za „predugo razmišljanje“ i preći na sledeći problem.

Testirano je, osim toga, da li je taster ENTER ili — pritisnut pre nego što je otku-can neki broj. Ukoliko jeste, pritisak ovog tastera bila ignorisan, jer bi mogao da izazove nove probleme. To smo učinili logičkim množenjem iskazka koje je opisano u 22 poglavlju Uputstva za upotrebu „galaksije“ (strane 22-23).

I poslednje pitanje: — Ovih sto lekcija je nekako mnogo brzo prošlo. Čini mi se da je njihovo numerisanje bilo u najmanju ruku čudno...

■ Lekcija nije bilo sto, nego 100. A binarni sistemi programeri često koriste...

**Spectrum**

## Kako postići besmrtnost

Sve video igre, na ovaj ili onaj način, ograničavaju broj „života“ glavnog junaka. Često je to i istovremeno ograničavanje raspoložive količine energije, kiseonika, hrane ili vremena. Posledicu znamo: igrač veliku većinu vremena provodi u prvih nekoliko nivoa (koji mu ubrzo postaju nezanimljivi). Tek što uspe da pređe u sledeći (najčešće nepoznatu i težu) fazu igre, igrač izgubi nekoliko života, ne stigavši da uoči sve prepreke i razradi novu taktiku.

(Istini za volju, neke video igre, kao što je „Penetrator“, pored uobičajenog moda za igru, pružaju i mogućnost uvežbanja (training mode) igrača — igrač uvežbava pojedine faze igre, ili, jednostavno, bira samo ono što voli.)

Naravno, bez ovih otežavajućih okolnosti većina igara bi izgubila svaku draž. Ipak, da bi ispravili ovo „nepravdu“, vredni ljubitelji igara su upornim radom otkrili ključne lokacije u programu koje je potrebno izmeniti da bi igrač postao „besmrtni“. Iako oduzimaju mnogo vremena, ovakvi poduhvati omogućuju sticanje novih znanja iz oblasti programiranja.

Za ovu priliku smo odabrali nekoliko popularnih profesionalnih igara, o čijim višim nivoima mnogi iscrpljeni spektromani još uvek maštaju. Navešćemo lokacije koje je potrebno izmeniti naredbom POKE neposredno pre startovanja programa. Većinu ovih lokacija je otkrio Eliša Kabiljo iz Beograda. Za program Pyramid zaslužan

Program	Adresa	Sadržaj	Posledica
COOKIE	28698	0	bezbroj života
PSSST	24984	0	bezbroj života
Manic Miner	35136	0	bezbroj života
	34269	n	n = broj života, n <= 32
Pyramid	44685	0	neograničena energija
Arcadia	25776	0	bezbroj života
Hunchback	26998	0	bezbroj života
Transversion	23020	0	bezbroj života
Jet Set Willy	34483	195	eliminisanje zaštite
	35899	0	bezbroj života
ZIP-ZAP	53750	0	bezbroj života
	53751	0	bezbroj života
	53752	0	bezbroj života
	53753	0	bezbroj života
	54141	0	bezbroj života
	54142	0	bezbroj života
	54143	0	bezbroj života
	54144	0	bezbroj života
	54144	0	bezbroj života
	54038	0	bezbroj života

je Zoran Strumberger, a za program Arcadia Slobodan Vujanović, obojica iz Zemuna.

Iz tabele se vidi da Manic miner može da se prekroji i tako da igrač ima fiksni broj života, ali ne veći od 32, jer se u tom slučaju događaju čudne stvari na ekranu. Na primer, ako želite da igrate sa deset života, ukucajte POKE 34269.10.

Unošenje ovih izmena (u žargonu: patch, zakrpa) vrši se na sledeći način:

- 1 Učitava se bezik program koji učitava ekran i mašinski deo, ili delove programa. To se postiže komandom „MERGE“.
- 2 Pronalazi se linija u kojoj se izvršava naredba RANDOMISE USP xxxxx (ili PRINT USP xxxxx, ili nešto slično). Ovom naredbom bezik nepovratno skraćuje u mašinski program na adresu xxxxx, pa se izmene moraju uneti pr njenog izvršenja.
- 3 Liniju ćemo izmeniti tako što ćemo neposredno pre ove naredbe ukucati POKE adresu, sadržaj iz tabele.
- 4 Sada možemo da kažemo RUN ili GOTO ... da bi se nastavilo normalno učitavanje ostalih delova programa. Još je praktičnije sačuvati ovaj izmenjeni program sa, na primer, SAVE „NOVI“ LINE ... i pomoću njega kasnije vršiti učitavanje programa.

Za izmenu programa Jumping Jack potrebno je ubaciti tri nove linije i program sačuvati sa SAVE „JJ-El“ LINE 10:

```
10 POKE 26034:0: POKE
26035,91: RESTORE 30: FOR
a=23296 TO 23309
20 READ d: POKE a,d: NEXT
a: GO TO 2
30 DATA 33,0,0,34,141,38,24,
34,143,117,195,66,111
```

Nedavno se u literaturi pojavilo još nekoliko zahvata koji omogućavaju ulazak u bilo koji nivo Manic Minera. Postupak je prilično čudan: učitati normal-

no program, zatim pritisnuti ENTER i sledeći niz brojeva: 6031769. Na drugu ekrana bi trebalo da se pojavi cipela. Pritisnjajući ISTOVREMENO brojeve iz tabele koja sledi, ulazimo u odgovarajući nivo igre.

nivo	Kombinacija
1	6
2	61
3	82
4	621
5	630
6	631
7	632
8	6321
9	64
10	641
11	642
12	6421
13	643
14	6431
15	6432
16	64321
17	64
18	651
19	652
20	6521

S. Vujanović

## Spectrum 64 znaka u redu

Ovaj program vam omogućava da na računaru ZX Spectrum, ostvarite ispis teksta na ekranu u širini od 64 slova u jednom redu. Program je u potpunosti rođen u bejziku. Zbog ograničenja koja nameće rezolucija na ekranu, slova su nešto manje čitljiva nego originalna, ali uz malo podešavanje osvetljenja i kontrasta na vašem TV aparatu, to ne bi trebalo da predstavlja problem.

Da bi se razumelo kako program radi, potrebno je upoznati se sa značenjem sistemske varijable CHARS. Ta varijabla stoji na adresi 23606 i 23607 i

pokazuje Spectrumovom operativnom sistemu gde treba da traži definiciju oblika slova koje ispisiše. Prvobitna vrednost za CHARS je 23606=0 i 23607=60 i ona pokazuje na ROM (fiksnu memoriju), pa stoga i dobijamo slova odmah po uključenoj računaru. Ta sistemska varijabla se može postaviti na bilo koju drugu vrednost u okviru RAM-a. Ako se na novoj adresi nalazi neka druga, nova definicija slova (na primer, cirilica). Spectrum će je prihvatiti kao svoju sopstvenu.

Da bi ta osobina mogla da se iskoristi za dobijanje 64 slova u redu na ekranu, potrebno je definisati dva nova seta znakova. Oba dva seta će sadržavati iste slova, samo što će se u jednom setu slova nalaziti u levoj polovini Spectrumovog standardnog znaka, a u drugom setu u desnoj polovini. Sada je još potrebno da navedemo računaru da ta dva znaka odštampaju u jednoj standardnoj poziciji na ekranu, što se postiže naredbom OVER 1, koja, inače, ima zadatka da preštampa dva slova jedno preko drugog.

Niz znakova može imati do 64 slova i smešta se u niz B\$, koji može biti dimenzionisan na novu širinu ekrana. Ta dimenzija ne može biti veća od 64 i ne manja od 1. Program će prihvatiti niz slova bilo koje dužine (do 64).

Potrebno je, takođe, objasniti funkciju promenljive F. Ona služi programu da za svako parno slovo zadrži poziciju za pisanje na ekranu i tako omogućava da, preko OVER 1, budu napisana dva slova umesto jednog u jednoj standardnoj slovnoj poziciji.

Ako vam kojim slučajem program krahira sa novim setom znakova, povratak na standardni set se vrši sa: GO TO 9999

Program omogućava štampanje na bilo kom mestu na ekranu, i to na način koji je sličan Spectrumovoj PRINT ad instrukciji. To znači da pe poziva podprograma treba postaviti dve promenljive: „r“ za red i „p“ za kolonu u kojoj želite da štampate. Takođe treba biti obazriv da dužina teksta koji štampate ne pređe krajnju desnu kolonu jer će se tada javiti greška.

Sam program je veoma kratak i obuhvata samo linije 9000 do 9999. Ceo potprogram se nalazi u liniji 9000. Linija 9988 vrši automatsko snimanje potprograma i novog seta znakova. Linija 9990 je za automatsko učitavanje seta znakova, a 9999 vam pomaže da se vratite u Spectrumovom definiciju znakova. Sama, definicija dva nova seta znakova je prilično obimna i zahteva 1540 bajtova na kraju memorije (iznad RAMTOP-a).

# MOŽETE ZNAČAJNO POVEĆATI VASU PRODUK TIVNOST

Izašla je iz  
štampe knjiga

Uz knjige koje su alat za stabilizaciju

## D.S. Arandjelović 1. KOMPJUTERSKA NOJ PRIVREDI

D.S. Arandjelović, dipl. ing. stručnjak za organizaciju rada, naučni radnik, profesor, saradnik (IV i VI kategorije) H.B. Maynard INC. Management Consultants (USA), autor 14 knjiga iz oblasti organizacije, industrijskog inženjeringa i statistike radnih prirodnosti. Tehničko-ekonomsko knjige Kompiuterska organizacija u savremenim privrednim organizacijama i postupcima koji se primenjuju u savremenim privrednim organizacijama. dati su i kompjuterski algoritmi automatizacije i konstantne sposobnosti na međunarodnom tržištu. Harčedito su obradene metode upravljanja procesa proizvodnje i preciznih normativa (MTP), kvaliteta i primeni iz prakse uspešno veštinskih sistema rešenja kompiuterske automatizacije postrojenja koja se stručnjaci pri modernizovanju radnih organizacija moraju da znaju. Metodološki organizaciju sprovedite tako da je mošće u svako doba pravim i ekonomičnim sredstvima. Strana 322, format 18x25 cm, pozov broj sa zaštitnom vizebojnom omlatnicom, latinitica. Cena 1.100.— dinara

## Frederick W. Hornbruch 2. POVEĆANJE PRODUKTIVNOSTI

Deset praktičnih primera kažu kako je u deset američkih proizvodnim povećana produktivnost 25—50%, pa čak i 100% u organizacijama koje proizvode i rekorodnog dobrotina. Data je analiza sledećih istaknutih proizvodnih programa: a) fabrike maslaca, b) fabrike metalnih proizvoda, c) održavanje i popravka mašina, d) fabrike tekstilnih proizvoda, e) fabrike prehrambenih proizvoda, f) fabrike elektonskih mašina, g) Banka (sveinik troškove) za 25%, h) fabrika elektonskih mašina, i) Banka (sveinik troškove) za 85%, j) fabrika elektonskih mašina. Povećanje produktivnosti na samo 2% ili 10% nego za 50—100%. Strana 365, format 18x25 cm, pozov broj, vizebojna omlatnica. Cena 750.— din.

## H.B. Maynard 3. SAVREMENA ORGANIZACIJA PROIZVODNJE

Priručnik modernog upravljanja proizvodnjom. Jedno od najpraktičnijih savetnih izdanja, na tome je saradivao 85 evropskih naučnika, stručnjaka i privrednika, na osnovi njihove prakse. Ovaj priručnik je namenjen savremenim rukovodnim i izvršnim radnicima i pomoćnicima u njihovim organizacijama. On daje savete o stvaranju i održavanju produktivne kvalitativne proizvodnje, povećavanju u stalnim i stalnim uslovima, format 18x25cm, pozov broj, vizebojna omlatnica. Cena 1.500.— din.

## L.D. Miles 4. PRIVREDNA I TEHNIČKA VREDNOSTI

Knjiga upoznaće tehničke i ekonomske stručnjake u praksi i radnike na primeni industrijskih nauka kako se boljim rešenjima i vrednoti proizvoda i posredstva bolje produktivnosti. Knjiga je namenjena praksi u radnim organizacijama. U njoj se proučavaju metode rešenja problema povećanja vrednosti proizvoda bilo da se radi da kvalitativno, ili i jedno i drugo. Knjiga je namenjena praksi u radnim organizacijama. U njoj se proučavaju metode rešenja problema povećanja vrednosti proizvoda bilo da se radi da kvalitativno, ili i jedno i drugo. Analiza vrednosti pruža objektivno merilo vrednosti analognih vrednosti. Strana 340, format 18x25 cm, pozov broj, vizebojna omlatnica, latinitica. Cena 1.100.— din.

## HITVOI NAJVEĆEG SVETSKOG IZDAVAČA

- MC GRAW
- HILL
- NALJORK

Nezavisno posredništvo koje izdaje pod svojim imenom  
knjige sa sledećim adresama: Organizacija i privreda  
za organizaciju i privrednu kulturu (iz prve knjige)  
1/2-3/3 za organizaciju i privrednu kulturu (iz druge knjige)  
Ustanovljena delovodna organizacija (iz treće knjige)

Narudbjenica — Računari 2  
Kulturni centar RJ „Privredna knjiga“  
Zbirno Gornji Milanovac, Gornji Milanovac

Ime i prezime radu nove organizacije \_\_\_\_\_ Datum \_\_\_\_\_  
Broj pošte, mesto, adresa i telefon narudbice \_\_\_\_\_  
Broj pošte, mesto, adresa i telefon narudbice \_\_\_\_\_ Datum \_\_\_\_\_  
Preporučeni pošta \_\_\_\_\_  
Pošta odmahovog lica \_\_\_\_\_  
Ova štamparija za preporučeno odobro čuva  
sve sa knjigama u oštari i štampi štampa  
redovno je odgovarajuće na sve  
M.P. \_\_\_\_\_

Kulturni centar — RJ „Privredna knjiga“  
Izdavač: Gornji Milanovac





BIBLIOTEKA  
PROGRAMA

kasetu. Od sada pa dalje  
možete ga učitati sa LOAD  
„64 slova/1“ ili samo LOAD“.

Osnovna vrednost ovog  
programa je što umesto 704 slova  
mesta (ako isključimo donje  
dve sistemske linije), omogućava  
korišćenje 1408 slovnih me-  
sta na ekranu, što znači i dva  
puta više informacija u isto  
vreme.

Jedan predlog bi bio da se  
kao formirani ekran smesti kao  
SCREENS datoteka na traku, a

```

1:
2: "VENTILATOR" I "GALAKSIJA"
3:
4: 7.4.1984.
5:
100 M=RND*(PIR-I+0 TO 3)
110 BYTE E=SCREEN, I=INT (RND*6)
120 A(1)=INT (RND*6)
130 NEXT I
135 HOME
140 A=USR (A2D537)
150 CALL 4000
170 R=H+I*8
180 R=H+I*8
182 PRINT "JA SAM RACUNAR GALAK
SIJA" P: "A": I: I: I: X.
185 HOME
190 PRINT X$; AT 16; "GALAKSIJA"
200 P=96
210 IF M<0.5 G.510
220 PRINT AT P; INPUT Y$
230 FOR I=0 TO 3
240 BYTE C=23A+I, A(1)
250 W=BYTE (PTR$+I)-40
255 IF (D<0)+(D+I)>6.220
270 BYTE E=C+D+I
280 NEXT I
290 B=USR (E2C46)
300 PRINT AT P+4; B;
310 IF B=4 G.1000
320 B=USR (E2C40)
330 P=P+B
510 FOR I=0 TO 3
520 BYTE E2800*P+I+1, BYTE (A+I)
530 NEXT I
540 PRINT AT P+24; I: Y$
550 B=BYTE (PTR$+4)-16
560 IF (D<0)+(D+I) 4.510
570 IF 0=4 G.2000
580 BYTE A+4
590 B=BYTE (PTR$+4)-40
600 IF (D<0)+(D+I) 6-10
700 BYTE A+5
710 A=USR (F.3000)
720 P=P+32+I*220
1000 ROVY
1010 PRINT AT 362, "V R A U O"
1020 PRINT: PRINT X$, AT 430, R
1030 PRINT "GALAKSIJA", AT 462, K
1040 PRINT AT 499, "PRITISNI RET
":
1050 IF KEY(48) ELSE G.1050
1060 IF KEY(48) G.1060
1070 FOR I=0 TO 3
1080 BYTE E2800+I, INT (RND*6)
1090 A(1)=INT (RND*6)
1100 NEXT I
1110 M=I-M*6.165
2200 K=K+1
2210 PRINT AT 362, "POBEDIO SAM!"
2020 GOTO 1020
3000 PRINT AT 362, "V R A R S !"
3010 K=K+1
3020 G.1020
4000 HOME
4010 PRINT " * MASTERMJAJN
D *": P.
4030 PRINT " PRAVILA IGRE SU,
NADMO SE,"
4040 PRINT " OPSTE POZNATA, OVAJ
PROGRAM JE ZA VAS."
4050 PRINT " OHDUGUVA PARELNUJ
IGRU PRO-".
4060 PRINT " TIJ RACUNARA. MOZE
SE IZVRSITI."
4070 PRINT " BEDIŠ?": P.
4080 PRINT " IGRANU CIFRE 0-5
NA ČETIRI:"
4090 PRINT " MESTA. SVI PODACI S
E KUCAJU:"
4100 PRINT " NORMALNO (KOMANDA I
NPOV):"
4110 PRINT AT 499, "PRITISNI RET
":
4120 IF KEY(48) ELSE G.4120
4130 HOME
4140 RET

```

```

M2C4A: 02 00 05 05 FF FF 02 01
M2C4A: 01 02 FE FE 21 3A 2C 11
M2D4A: 3E 2C 0D 77 2C 0D 7D 20
M2C52A: 0C 06 2C 23 13 78 7E 0D
M2C5A: 20 07 07 07 07 07 07 07
M2C62A: 2C 11 42 2C 0D 60 13
M2C6A: 7B FE 4E 20 67 23 70 FE
M2C72A: 42 20 0E 1E 05 01 04 0E
M2C7FA: ED 00 09 C9 06 00 21 3E
M2C82A: 11 42 2C 0D 77 2C 0D 7D
M2C8A: B9 08 04 3E 00 00 00 00
M2C92A: E1 3E FE 77 01 C9 77 01
M2C9FA: C9 00 00 00 00 00 00 00
M2CA2A: 06 2E 07 07 07 07 07 07
M2CA0A: 00 00 00 00 00 00 00 00
M2C2B2A: 2E 22 40 2C 2A 0A 2C 09
M2C2BA: 2A 0A 2C 11 06 00 19 E5
M2C2C2A: F9 D1 2A 0A 2C 01 84
M2C2D2A: E1 20 03 19 00 00 22 0E
M2C2E2A: E1 3E FE 77 01 C9 77 01
M2C2F2A: 21 62 09 22 4A 2C 0A
M2C3A2A: A4 2C 28 7C 05 22 4A 2C
M2C3E2A: 21 FFF FF 0A 2A 0A 2C 34
M2C3FA: 7E 7E 87 00 65 00 23 0E
M2C4F2A: 1F 05 21 00 0E 22 42
M2C2FA: 11 3A 2C 2A 0A 2C 0D 77
M2D02A: 2C 11 42 2C 2A 0A 2C 0D
M2D05A: 77 2C 0D 4A 2C 2A 0A 2C
M2D12A: 11 04 08 19 00 00 00 00
M2D15A: 23 E5 0C 60 2C 01 E7 0E
M2D22A: 28 05 23 22 4A 2C 0D E5
M2D23A: 00 2C 37 3F 0D 52 02 FA
M2D24A: 2C 23 2E 2E 2E 2E 2E 2E
M2D30A: 0C 05 2D 3E 2A 32 06 28
M2D32A: 32 2A 0A 21 A6 2D 0E 00
M2D44A: 2D 21 0B 20 11 00 20 0E
M2D50A: 00 00 00 00 21 06 29 11
M2D62A: 0D 06 15 0C 00 00 00 00
M2D62A: F2 21 11 0A 2D 0A 0C 0D
M2D66A: AC 2D 3A 33 2D 0A 01 20
M2D72A: F9 21 FF 26 20 2B 08
M2D76A: 52 FF C9 03 2C 4E 42
M2D82A: 54 55 44 42 42 4B 4F
M2D8A: 45 44 46 42 42 4F 21 53
M2D92A: 44 54 55 42 46 57 4A
M2D94A: 5D 21 32 3A 29 35 51 51
M2D94A: 4A 55 4A 4A 4F 4A 21 45
M2D9A: 4D 4A 1A 3D 57 05 23 13
M2D9C2A: 2D 0B C9 05 15 3A 23 0A
M2D9D2A: 2D 0B C9 05 15 3A 23 0A
M2D9E2A: 00 00 00 00 00 00 00 00
M2DCA: 00 00 00 00 00 00 00 00
M2DD2A: 00 00 00 00 00 00 00 00
M2DD4A: 00 00 00 00 00 00 00 00
M2DD6A: 00 00 00 00 00 00 00 00
M2DD8A: 00 00 00 00 00 00 00 00
M2DDE2A: 00 00 00 00 00 00 00 00
M2DE4A: 00 00 00 00 00 00 00 00
M2DE6A: 00 00 00 00 00 00 00 00
M2DE8A: 00 00 00 00 00 00 00 00
M2DEA: 00 00 00 00 00 00 00 00
M2DE2A: 01 02 02 01 01 01 01
M2E12: 03 02 02 01 01 04 03
M2E1A: 01 02 00 04 01 02 04 03
M2E22: 00 00 00 00 00 00 00 00
M2E24A: 00 00 00 00 00 00 00 00
M2E26A: 00 00 00 00 00 00 00 00
M2E32A: 00 00 00 00 00 00 00 00
M2E3A: 00 00

```

```

10 REM
20 REM program 64 znakU u LINIJI (C)BORIVOJ PERCIC
30 REM
40 CLEAR 63394: LET G=63395: LET I=6425
100 NEXT I: LET I=I+1: LET G=I+16: GOTO 100
80 READ P: IF I=INT (M/16): LET G=I+16: I
90 POKE G+1: I=INT (M/16)+8: POKE I+8: G=I+16: I
100 NEXT I: LET I=I+1: LET G=I+16: GOTO 100
120 REM
130 DATA 0.2: 2.0: 0.0: 0.0: 0.25: 0.7: 5.6: 7.5
140 DATA 0.2: 0.8: 1.4: 0.5: 1.4: 0.0: 1.3: 0.3: 0.4: 0.6: 0.3: 0.6: 1.7: 2.0: 0.0
150 DATA 0.2: 0.6: 0.6: 0.5: 0.5: 0.6: 0.3: 0.0: 0.2: 0.6: 1.3: 0.3: 0.6: 0.2
160 DATA 0.0: 0.0: 0.0: 0.2: 0.2: 0.2: 0.4: 0.0: 0.1: 0.1: 0.2: 1.0: 0.0: 1.0: 0.0
170 DATA 0.1: 1.4: 0.6: 0.2: 0.2: 0.4: 0.0: 0.3: 0.1: 0.2: 0.3: 0.4: 0.1: 0.0
180 DATA 0.2: 0.2: 0.6: 0.8: 1.1: 0.1: 0.2: 0.0: 0.5: 0.6: 0.9: 0.2: 0.4: 0.3: 0.6
190 DATA 0.1: 1.4: 0.5: 0.7: 0.3: 0.1: 0.1: 0.0: 0.2: 0.2: 0.0: 0.4: 0.0: 0.4
200 DATA 0.0: 1.6: 0.4: 0.6: 0.0: 0.0: 0.2: 0.0: 0.2: 0.0: 0.3: 0.1: 0.2: 0.6
210 DATA 0.9: 0.2: 0.2: 0.6: 1.3: 0.1: 0.0: 0.0: 0.0: 0.0: 0.0: 0.0: 0.0: 0.0: 0.0
220 DATA 0.1: 0.0: 0.4: 0.6: 0.6: 0.6: 0.0: 0.0: 0.0: 0.0: 0.1: 0.4: 0.2: 0.3: 0.6: 0.0
230 DATA 0.8: 0.7: 0.2: 1.1: 0.1: 0.1: 0.1: 0.0: 0.1: 0.1: 0.2: 0.2: 0.5: 0.7: 0.0
240 DATA 0.1: 0.0: 0.5: 0.5: 0.5: 0.0: 0.1: 0.1: 0.1: 0.1: 0.1: 0.1: 0.1: 0.1: 0.1
250 DATA 0.1: 0.1: 0.1: 0.1: 0.1: 0.1: 0.1: 0.1: 0.1: 0.1: 0.1: 0.1: 0.1: 0.1: 0.1
260 DATA 0.1: 0.1: 0.1: 0.1: 0.1: 0.1: 0.1: 0.1: 0.1: 0.1: 0.1: 0.1: 0.1: 0.1: 0.1
270 DATA 0.1: 0.1: 0.1: 0.1: 0.1: 0.1: 0.1: 0.1: 0.1: 0.1: 0.1: 0.1: 0.1: 0.1: 0.1
280 DATA 0.1: 0.1: 0.1: 0.1: 0.1: 0.1: 0.1: 0.1: 0.1: 0.1: 0.1: 0.1: 0.1: 0.1: 0.1
290 DATA 0.1: 0.1: 0.1: 0.1: 0.1: 0.1: 0.1: 0.1: 0.1: 0.1: 0.1: 0.1: 0.1: 0.1: 0.1
300 DATA 0.1: 0.1: 0.1: 0.1: 0.1: 0.1: 0.1: 0.1: 0.1: 0.1: 0.1: 0.1: 0.1: 0.1: 0.1
310 DATA 0.1: 0.1: 0.1: 0.1: 0.1: 0.1: 0.1: 0.1: 0.1: 0.1: 0.1: 0.1: 0.1: 0.1: 0.1
320 DATA 0.1: 0.1: 0.1: 0.1: 0.1: 0.1: 0.1: 0.1: 0.1: 0.1: 0.1: 0.1: 0.1: 0.1: 0.1
330 DATA 0.1: 0.1: 0.1: 0.1: 0.1: 0.1: 0.1: 0.1: 0.1: 0.1: 0.1: 0.1: 0.1: 0.1: 0.1
340 DATA 0.1: 0.1: 0.1: 0.1: 0.1: 0.1: 0.1: 0.1: 0.1: 0.1: 0.1: 0.1: 0.1: 0.1: 0.1
350 DATA 0.1: 0.1: 0.1: 0.1: 0.1: 0.1: 0.1: 0.1: 0.1: 0.1: 0.1: 0.1: 0.1: 0.1: 0.1
360 DATA 0.1: 0.1: 0.1: 0.1: 0.1: 0.1: 0.1: 0.1: 0.1: 0.1: 0.1: 0.1: 0.1: 0.1: 0.1
370 REM
380 PRINT "GOSPODAR NIS IZBRISHE RET: *": LET M=0: GOTO 3000: STOP
390 REM
4000 POKE 23005, 1531: LET M=1: FOR W=1 TO LEN B$ POKE 23007, 245+M*240*W: P
4100 POKE 64, 64: W=W+1: GOTO 4000: IF M=1: LET M=0: GOTO 4000: IF M=0: GOTO 4000
4990 POKE 64, 64: W=W+1: LINE INPUT B$: W=1: GOTO 4000: IF M=1: GOTO 4000
9990 CLEAR 63394: LOAD: "CODE": STOP
9990 POKE 23005, 0: POKE 23007, 0

```

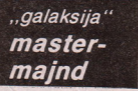
Da bi se olakšalo ukucavanje, broj bajtova je sažet na 1/4 prvobitnog, iako da treba ukucati samo 384 broja u DATA linijama. To sažimanje je zahtevalo jednu malu rutinu koja će izvršiti potrebne transformacije i prenesti karakter set na njegovo pravo mesto. Rutina za premještanje seta znakova je u linijama 60 do 110, a definicija znakova je u DATA linijama 130 do 360. Linija 380 stupa za demonstraciju rada programa.

Posle ukucavanja programa i njegovog starovanja, mogu se izbrisati sve DATA linije, kao i rutina za premještanje seta znakova.

Posle ukucate kao program, otkucajte RUN da izvršite definisanje novog seta znakova. Program će tada izvršiti malu demonstraciju rada. Zatim otkucajte GO TO 9999 i izbrisate linije 10 do 380 jer vam više nisu potrebne. Sa GO TO 9988 izvršite snimanje podprograma i definisanog seta znakova na

pre nekog vašeg programa, iako da u toku učitavanja programa na ekranu već imate sve informacije i uputstva za taj program. Ista tehnika se može iskoristiti da smestite sadržaj trake na njen početak. Učitavanje datoteke SCREENS traje oko 40 sekundi. Dovoljno je, znači, da toliko ostavite na početku svake trake, pa da kasnije i to snimite njen sadržaj koji ste napisali uz pomoć ovog programa. Datoteka tipa SCREENS se učitava sa LOAD "\_\_\_" CODE.

Borivoj Perčić, dipl. ing.



2020 GOTO 1020  
3000 PRINT AT 362, "V R A R S !"  
3010 K=K+1  
3020 G.1020  
4000 HOME  
4010 PRINT " \* MASTERMJAJN D \*": P.  
4030 PRINT " PRAVILA IGRE SU, NADMO SE,"  
4040 PRINT " OPSTE POZNATA, OVAJ PROGRAM JE ZA VAS."  
4050 PRINT " OHDUGUVA PARELNUJ IGRU PRO-".  
4060 PRINT " TIJ RACUNARA. MOZE SE IZVRSITI."  
4070 PRINT " BEDIŠ?": P.  
4080 PRINT " IGRANU CIFRE 0-5 NA ČETIRI:"  
4090 PRINT " MESTA. SVI PODACI S E KUCAJU:"  
4100 PRINT " NORMALNO (KOMANDA I NPOV):"  
4110 PRINT AT 499, "PRITISNI RET":  
4120 IF KEY(48) ELSE G.4120  
4130 HOME  
4140 RET

drugi igrač saopštava koliko se ta kombinacija slaže sa onom koju je na zamislio. Za svaku pribadaču koja je postavljena na dobrom mestu u dobroj boji igraču se daje jedna crna, a za svaku koja je pogodena po boji ali ne i po mestu — jedna bela pribadača (crne i bele pribadače ne mogu da se nađu u zadataj kombinaciji; on su samo informativne). Na osnovu svih podataka igrač pokušava da, u nekom od sledećih pokušaja, dobije četiri crne pribadače — znak da je pogodio kombinaciju.

Posle „galaksija“ nema kožu grafiku, boje su zamenjene brojevima 0-6. Da bi igra bila zanimljivija, igrač i računar paralelno pokušavaju da pogode kombinaciju koju je protivnik zamislio — onaj kome to pre uspe dobija jedan poen.

Najpre treba, sa NEW 512, rezervirati prostor za mašinski program i uneti ga pomoću UTM-a ili nekog drugog hex-loadera. Zatim treba ponovo otkucati NEW 512 i uneti bajzik program. Snimićemo sve ovo

```

10 :
20 : "VENTILATOR" I "BALAKS13A"
30 :
40 : 31.3.1984.
50 :
60 :
70 HOPE
80 P.AT165,"*****
**
85 P.AT177,"*
**
90 P.AT229,"* EVOLUC13A
**
95 P.AT261,"*
**
99 P.AT293,"*****
**
100 FOR I=1 TO 2000:N,I
110 HOPE
120 PRINT "KOMANDE"
130 PRINT
140 P:
141 PRINT "STRELICE ZA PO
KRETAJZE"
150 PRINT "SHIFT STRELICE ZA BR
ZO"
160 PRINT "KRETA"
170 P:
178 PRINT "ENTER ZA OS
TAVANJE"
180 PRINT "CE
190 PRINT "SPACE ZA BR
ISANJE"
195 P:
200 PRINT "L ZA PO
DETAK"
2000 A=USR(K2CC3A)

```

na kasetu sa SAVE, a zatim startovati program sa RUN. Posle zaglavlja, računar će zatražiti da se predstavimo, a zatim će početi igra. U toku predstavljanja treba da zamislimo broj koji će računar pogodati; to je, kako rekoh, četvorocifreni broj, u kome nijedna cifra ne sme da bude veća od 6, dok se cifre, naravno, smeju ponavljati. Kada je na nas red, treba da otkucamo pretpostavku (npr. 1234) a pritisnemo RETURN (ENTER) i računar će prikazati koliko bismo crnih, odnosno belih pridačica dobili (21 označava dve crne i jednu belu), a zatim će on pokušati da pogodi našu kombinaciju. Odgovarajući mu na isti način — kucajući 21, 03 ili neki sličan broj i pritisnujći ENTER. Igra se nastavlja svaki put dok neko ne pogodi protivničku kombinaciju (skor 40); računar tada ispisuje trenutni skor i igra se nastavlja.

Dejan Ristanović

**„galaksija“  
Evolucija**

„Evolucija“ („Life“) je poznata simulacija života. Na nekom staništu korisnik zadaje početni organizam koji se sastoji od određenog broja živih ćelija, dok su preostala polja „mrtva“. Taj organizam se razvija u strogim evolucionim uslovima — usamljene ćelije i ćelije koje žive u prenaseljenim područjima brzo umiru, dok se u umereno naseljenim predelima iz časa u čas radaju nove ćelije. Neke početni organizmi, posle određenog broja generacija umiru dok drugi prelaze u neke stabilne forme. Za mnoge početne organizme se, verovali ili ne, posle određenog broja generacija javlja struktura koja, čak i bez mnogo mašte, podseća na čoveka.

Da biste otkucali ovaj program, treba najpre da rezervišete prostor za njegov mašinski deo — to činite tako što kucate NEW 454. Posle toga treba da unesete mašinski deo pomoću UTM-a ili nekog drugog programa. Ponovo otkucajte NEW 454 da obrišete UTM, pa unesite bežik koji, jednostavno, štampa uputstva. Sve to snimite na kasetu sa SAVE. Posle toga ćete program uvek startovati jednostavnim RUN i slediti uputstva koja računar ispisuje.

Dejan Ristanović

**BBC Electron  
Kako  
presnimiti  
program?**

Acornovi računari imaju sistem snimanja programa i podataka na kaseti koji slobodno možemo da nazovemo genijalno zamišljenim. Podaci se upisuju u blokovima od po 256 bajtova koji su jasno razdvojeni jedan od drugoga — ispred svakoga bloka se nalaze po dve informacije koje obuhvataju i ime samog programa. Zbog čega je ovo zgodno? Pre svega, možemo da uzmemo svaku traku, smestimo je u kaseto i prematamo na bilo koje mesto. Ako otkucamo "CAT i pritisnemo RETURN a zatim i PLAY na kasetofonu, za trenutak ćemo videti na koji je program kaseto pozicionirana i koliko smo daleko od njegovog početka (računar, uz ime programa, ispisuje i broj bloka na koji smo naišli i to heksadecimalno).

Osim imena, u okviru zaglavlja svakog bloka nalazi se i druge informacije. To je, najpre, adresa u memoriji od koje program počinje da se učitava ukoliko korisnik nije naredio drugačije (označava se kao „load address“), zatim adresa od koje program počinje da se izvršava ukoliko ga učitavamo sa „RUN („excution address“), broj bloka od početka i (eventualno) bit koji govori o tome da se radi o zadnjem programu u bloku.

Najjednostavniji način da presnimimo program je da, najpre, otkucamo \*OPT 1,2, a zatim: „LOAD“... Učitavanje će započeti, potrajati izvesno vreme, a zatim će računar, iza imena programa, ispisati njegovu dužinu u bajtovima kao i „load“ i „excution“ adresu programa. Adresa od koje se program učitava i izvršava adresa imaju po

četiri mesta do bajta, što ne bi trebalo posebno da nas zabrine — oni koji kupu dodatni procesor sa 64 K (ili pola megabajta) RAM-a shvatit će da program može da se učitava i u ovaj novi memorijski prostor, koji na neki način treba adresirati. Drugo zapažanje će biti još jednostavnije i korisnije: pošto znamo odakle program počinje, koliko bajtova ima i koja mu je izvršna adresa, moći ćemo da ga snimimo na kasetu bez mnogo problema. Ovo je voljno je da otkucamo \*SAVE „ime“ FFFxxxx + nnnn FFFFeee, pri čemu nnnn označava dužinu programa u bajtovima, xxxx adresu od koje se on učitava a eeee izvršnu adresu.

Na žalost, ne mogu svi programi da se presnime na ovaj način — u cilju njihove zaštite adresa od koje se oni učitavaju je odabrana tako da, po učitavanju, kontrola ne može da se vrati bežik interpreteru. Te programe je neophodno učitati na neko drugo mesto, a zatim snimiti tako da kopija bude identična originalu, što možemo samo koristeći sistemski potprogram OSFILE.

OSFILE se poziva sa JSR &FFDD ili, ukoliko je poziv iz bežikica, CALL &FFDD. Pre toga je neophodno da u X i Y registre smestite adresu kontrolnog bloka koji kontrolise operaciju upisa odnosno čitanja (u X se smešta bajt manje, a u Y bajt veće težine adrese bloka), a u A kôd koji kontrolise operaciju koju treba izvršiti. Tačna struktura kontrolnog bloka je data na strani 455 uputstva za upotrebu BBC B, u akumulatoru se stavlja nula ukoliko želimo da upišemo na traku, a &FF ako želimo da čitamo sa nje.

Posle čitanja programa sa trake računar će u svoj radni prostor smestiti parametre programa. Adresa od koje se program učitavao će biti smestena od &3BE do &3C1, izvršna adresa od &3C2 do &3C5, dužina u &3C6 (MSB) i &3C8 (LSB), a ime programa počešći od &3D2 pa do prvog nula bajta koji predstavlja terminator. Treba primetiti da dužina nije uvek tačna — ako je njen bajt manje težine (MSB) nula, bajt veće težine (LSB) treba veštački povećati za jedan da bi se dobio tačan podatak o dužini. Uz pomoć ovih podataka (koji se ovde po prvi put javno objavljuju) može da se pripremi kontrolni blok za snimanje programa koji omogućava pravljenje kopija.

Čitava ova teorija je sažeta u programu PREPIS koji objavljujemo. Asemblirajte ga i snimite na traku (pre kucanja sors programa otkucajte PAGE=&3000, a posle asembliranja — \*SAVE „PREPIS“ E00 +&B0). Pro-

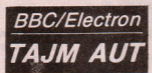
```

PROGRAM PREPIS
1 REM
2 REM
3 REM PPSINPC $E00
4 REM THE $E71
5 REM BLOK $E7C
6 REM BLOK 2H
7 REM PREPIS $F00
8 REM
9 REM Dejan Ristanovic 1904.
10 REM
11 REM
12 REM
100 FOR I=0 TO 3 STEP 3
200 P%=$E00
300 I=I+1
400 ENTRY_POINT
500 JSR $FEF0
600 ENTRY2
700 LDX $E7C
800 LDY $E6
900 LDR $E6F
1000 JSR $FDFD
1100 LDR $0
1200 STR $E6B
1300 LDR $E6F
1400 STR $E67
1500 LDX $0
1600 CYCLE1
1700 LDR $3E0,X
1800 STR $E7D,X
1900 DEK
2000 BNE CYCLE1
2100 CYCLE2
2200 LDR $3D2,X
2300 BNE $PREPARE
2400 STR $E71,X
2500 INK
2600 BNE CYCLE2
2700 $PREPARE
2800 LDR #13
2900 STR $E71,X
3000 CLC
3100 LDR $3CB
3200 STR $E6B
3300 BNE OK
3400 INC $3CC
3500 OK
3600 LDR $3CE
3700 RDC $E6F
3800 STR $E6B
3900 $RAWING
4000 JSR $FDFD
4100 CMP #13
4200 BEQ REINIC
4300 LDR $E6F
4400 LDY $E6
4500 LDR $0
4600 JSR $FDFD
4700 JMP $RAWING
4800 EOS (*C10,Ristanovic 8*)
4900 EOS $ASC"4"
5000 I
5100 P%=$E0E
5200 DOP1
5300 REINIC
5400 LDR $E0
5500 STR $E71
5600 LDR $0
5700 STR $E7E
5800 STR $E6E
5900 STR $E6B
6000 STR $E61
6100 LDR $E6F
6200 STR $E7F
6300 JMP ENTRY2
6400 I
6500 NEXT I
6600 ?$E71=$E0D
6700 IN$E7C
6800 RDC P IF A#$999 THEN 7400
6900 ?1=$E1=$1
7100 DTR# 571,$E0,$F,0,0,0,0,0,0
7200 DTR# 0,$F,0,0,0,0,0,0,0
7300 INT$999
7400 CALL ENTRY_POINT

```

gram se aktivira sa "RUN PREPIS. Po učitavanju pritiskate RETURN da učitate sledeći program sa trake, a SPACE da se memoriju učitani program snimate na drugu kasetu. Ovo snimanje možete da ponavljate proizvoljan broj puta — sve dok ne napravite dovoljan broj kopija. Zatim možete da pritisnete RETURN i učitate sledeći program.

PREPIS, na žalos, ima ograničeno dejstvo: pomoću njega ne možete da učitate programe Hopper, Starship Command i još nekoliko drugih. Osim toga, program „The Hobbit“ je toliko dugačak da ga PREPIS vrlo teško prepisuje. O načinu na koji se presnimavaju i najbolje zaključani (kada sa "LOAD pokušamo da pročitamo posebno zaštićen program, dobićemo poruku Locked) dugački programi govorimoće, ako za to bude interesa, u Računarima 3. Tada ćemo objaviti i poboļjša verziju programa PREPIS, koja se zove DCOPIY.



Toga dana sve vam je išlo savršeno — učitali ste svoju najmilijenu igru i, iz prvog pokušaja, postigli svoj apsolutni rekord — došli ste do 12 ekrana! Igra teče i dalje, nadate se da ćete stati i do sledećeg, kad čujete zvono na vratima (na telefon se, naravno, ne biste obazirali). Šta da se radi? Pomnjući sve lepe reči kojih se setite, pritisnete BREAK i odlazite do vrata da biste ustanovili da neko želi da pita gde je stan porodice... Vaš najbolji skor svih vremena je, naravno, otišao u nepovrat.

Naravno da je bade baš tako? Moralo bi da mora ako posedujete program „Univerzalna pauza“. Učitajte ga pre učitavanja bilo kog (ozbiljno mislimo bilo kog) drugog programa i, kada želite pauzu, pritisnite taster (⊙ „biznis a“ ili, kako ga naši programeri nazivaju, „majmumski znak“). Ekran će se potpuno „zamrznuti“ — jedino će se (eventualno) čuti kontinuirani ton. Kada poželite da nastavite, pritisnite bilo koji taster (na primer, onaj koji treba da označi vašu sledeću akciju) i produžite sa igrom.

Kako „Univerzalna pauza“ može da bude univerzalna? Principi ovog programa su prilično jednostavni, iako će mnogi pomisliti da oni odlaže predaletko u „mračne vode“ operativnog sistema. Vaš računar radi u režimu stalnih prekida (interrupt) — svaki put kada, na primer, pritisnete neki taster, mikroprocesor dobija maskirani interrupt da bi ga registrovao. Poseban vid prekida su takozvani događaji (events) koji su više informativne prirode i koje računar u normalnoj situaciji ignorise. Pomoću \*FX 14 (ili, kako je urađeno u ovom programu, odgovarajućeg OCP-YTE poziva) neki od događaja mogu da budu omogućeni. U tom slučaju mikroprocesor ko-

```

10 REM
20 REM PRUSE source
30 REM
40 REM Dejan Ristanovic 1904.
50 REM
60 REM
70 REM
80 REM
100 OSBVE=$FFFF4 FOR I=0 TO 3 STEP 3
200 P%=$B00: REM Ili neka druga adresa
300 C% OF I
400 .INIT
401 I
402 \
403 \ Promena sadržaja EVNT vektora
404 \
405 \ - kada god bude bio pritisnut neki taster računar ce skociti na PROG
406 \
407 \
408 \
409 \
410 \
411 \
412 \
413 \
414 \
415 \
416 \
417 \
418 \
419 \
420 \
421 \
422 \
423 \
424 \
425 \
426 \
427 \
428 \
429 \
430 \
431 \
432 \
433 \
434 \
435 \
436 \
437 \
438 \
439 \
440 \
441 \
442 \
443 \
444 \
445 \
446 \
447 \
448 \
449 \
450 \
451 \
452 \
453 \
454 \
455 \
456 \
457 \
458 \
459 \
460 \
461 \
462 \
463 \
464 \
465 \
466 \
467 \
468 \
469 \
470 \
471 \
472 \
473 \
474 \
475 \
476 \
477 \
478 \
479 \
480 \
481 \
482 \
483 \
484 \
485 \
486 \
487 \
488 \
489 \
490 \
491 \
492 \
493 \
494 \
495 \
496 \
497 \
498 \
499 \
500 \
501 \
502 \
503 \
504 \
505 \
506 \
507 \
508 \
509 \
510 \
511 \
512 \
513 \
514 \
515 \
516 \
517 \
518 \
519 \
520 \
521 \
522 \
523 \
524 \
525 \
526 \
527 \
528 \
529 \
530 \
531 \
532 \
533 \
534 \
535 \
536 \
537 \
538 \
539 \
540 \
541 \
542 \
543 \
544 \
545 \
546 \
547 \
548 \
549 \
550 \
551 \
552 \
553 \
554 \
555 \
556 \
557 \
558 \
559 \
560 \
561 \
562 \
563 \
564 \
565 \
566 \
567 \
568 \
569 \
570 \
571 \
572 \
573 \
574 \
575 \
576 \
577 \
578 \
579 \
580 \
581 \
582 \
583 \
584 \
585 \
586 \
587 \
588 \
589 \
590 \
591 \
592 \
593 \
594 \
595 \
596 \
597 \
598 \
599 \
600 \
601 \
602 \
603 \
604 \
605 \
606 \
607 \
608 \
609 \
610 \
611 \
612 \
613 \
614 \
615 \
616 \
617 \
618 \
619 \
620 \
621 \
622 \
623 \
624 \
625 \
626 \
627 \
628 \
629 \
630 \
631 \
632 \
633 \
634 \
635 \
636 \
637 \
638 \
639 \
640 \
641 \
642 \
643 \
644 \
645 \
646 \
647 \
648 \
649 \
650 \
651 \
652 \
653 \
654 \
655 \
656 \
657 \
658 \
659 \
660 \
661 \
662 \
663 \
664 \
665 \
666 \
667 \
668 \
669 \
670 \
671 \
672 \
673 \
674 \
675 \
676 \
677 \
678 \
679 \
680 \
681 \
682 \
683 \
684 \
685 \
686 \
687 \
688 \
689 \
690 \
691 \
692 \
693 \
694 \
695 \
696 \
697 \
698 \
699 \
700 \
701 \
702 \
703 \
704 \
705 \
706 \
707 \
708 \
709 \
710 \
711 \
712 \
713 \
714 \
715 \
716 \
717 \
718 \
719 \
720 \
721 \
722 \
723 \
724 \
725 \
726 \
727 \
728 \
729 \
730 \
731 \
732 \
733 \
734 \
735 \
736 \
737 \
738 \
739 \
740 \
741 \
742 \
743 \
744 \
745 \
746 \
747 \
748 \
749 \
750 \
751 \
752 \
753 \
754 \
755 \
756 \
757 \
758 \
759 \
760 \
761 \
762 \
763 \
764 \
765 \
766 \
767 \
768 \
769 \
770 \
771 \
772 \
773 \
774 \
775 \
776 \
777 \
778 \
779 \
780 \
781 \
782 \
783 \
784 \
785 \
786 \
787 \
788 \
789 \
790 \
791 \
792 \
793 \
794 \
795 \
796 \
797 \
798 \
799 \
800 \
801 \
802 \
803 \
804 \
805 \
806 \
807 \
808 \
809 \
810 \
811 \
812 \
813 \
814 \
815 \
816 \
817 \
818 \
819 \
820 \
821 \
822 \
823 \
824 \
825 \
826 \
827 \
828 \
829 \
830 \
831 \
832 \
833 \
834 \
835 \
836 \
837 \
838 \
839 \
840 \
841 \
842 \
843 \
844 \
845 \
846 \
847 \
848 \
849 \
850 \
851 \
852 \
853 \
854 \
855 \
856 \
857 \
858 \
859 \
860 \
861 \
862 \
863 \
864 \
865 \
866 \
867 \
868 \
869 \
870 \
871 \
872 \
873 \
874 \
875 \
876 \
877 \
878 \
879 \
880 \
881 \
882 \
883 \
884 \
885 \
886 \
887 \
888 \
889 \
890 \
891 \
892 \
893 \
894 \
895 \
896 \
897 \
898 \
899 \
900 \
901 \
902 \
903 \
904 \
905 \
906 \
907 \
908 \
909 \
910 \
911 \
912 \
913 \
914 \
915 \
916 \
917 \
918 \
919 \
920 \
921 \
922 \
923 \
924 \
925 \
926 \
927 \
928 \
929 \
930 \
931 \
932 \
933 \
934 \
935 \
936 \
937 \
938 \
939 \
940 \
941 \
942 \
943 \
944 \
945 \
946 \
947 \
948 \
949 \
950 \
951 \
952 \
953 \
954 \
955 \
956 \
957 \
958 \
959 \
960 \
961 \
962 \
963 \
964 \
965 \
966 \
967 \
968 \
969 \
970 \
971 \
972 \
973 \
974 \
975 \
976 \
977 \
978 \
979 \
980 \
981 \
982 \
983 \
984 \
985 \
986 \
987 \
988 \
989 \
990 \
991 \
992 \
993 \
994 \
995 \
996 \
997 \
998 \
999 \
1000 \

```

risti EVNT vektor da odredi adresu na koju će „skočiti“ kada događaj nastupi. U našem slučaju omogućen je događaj 2 koji, kako piše u uputstvu, nastupa kada je pritisnut neki taster. U tom slučaju, izvršava se PROG; u akumulatoru je broj 2 (broj događaja), a u Y registru ASCII kod pritisnutog tastera. U okviru tretiranja događaja potrebno je, najpre, sačuvati sve procesorske registre na steku. Posle toga nastupa pauza koja je neophodna da bi korisnik otpustio taster (⊙). Zatim računar skanira tastaturu bez korišćenja rutine operativnog sistema — za vreme „događaja“ interrupt je omogućen pa ove rutine ne funkcioni-

šu. Ovo skaniranje je omogućeno testiranjem područja SHEILA koje je veoma nekompletno obradeno u okviru uputstva za upotrebu. Ukoliko je detektovan pritisak na bilo koji taster, računar sa steka skida potrebne registre i, pomoću RTS, vraća kontrolu programu koji se izvršavao pre pritiska na ⊙.

Program možete da otkucaete kao i svaki bejzik. Posle inicijalizacije program je neprekidno u računaru sa kojim možete da radite sasvim normalno; program će biti obvisan jedino pritiskom na CTRL BREAK, dok je posle BREAK potrebna nova inicijalizacija koja se postiže sa CALL & BOO

# olivetti M20

## OLIVETTI M20

Fabrika Olivetti ima dugu tradiciju u proizvodnji ličnih profesionalnih računara. Tako je još pre dvadesetak godina bila prva sa „programom“ P101 računarem, koji je imao 4 K memorije, zasnovane na irikolji opruge. Nastavljajući tradiciju u svojoj filijali u Cupertino u Kaliforniji, Olivetti je 1979. godine otpočeo razvoj novog ličnog profesionalnog računara. Rezultat, M20 zvanično je predstavljen u proleće 1982. godine. Proizvodi se u Italiji, a u nas ga zastupa RO DINARA iz Beograda.

## HARDWARE:

M20 je zasnovan na Zilog-ovom potpuno 16-bitnom mikroprocesoru, Z 8001, koji radi na frekvenciji 4 MHz. Za njega je napisan poseban operativni sistem PCOS (Professional Computer Operating System).

Mikroprocesori iz serije Z 8000 omogućavaju segmentirano adresiranje do 128 segmenata po 64 KB (odnosno 8 MB), virtuelnu memoriju, 32-bitnu aritmetiku, dinamičku dodelu memorije, multiprogramiranje, multiprocesorski rad i još dosta drugih svojstava do nedavno korišćenih samo kod većih sistema. Zajedno sa M20 paralelno je razvijena cela serija od M10 do M60 zasnovana na novoj tehnologiji.

Taj razvoj se i dalje nastavlja i gotovo svakodnevno svojstva i mogućnosti sistema rastu. Osnovni memorijski kapacitet je 128 KB RAM-a, a moguća su proširenja sa priraštajima po 32 KB ili 128 KB. Ukupno za sada M20 može imati do 0,5 MB, odnosno korisničkih 421 KB neto. Diskretne jedinice od 5 1/4 inča, mogu da budu od 160, 320 ili 640 KB neformatizovano.

Winchester disk sa 11,7 MB neformatizovano, moguće je priključiti kao dodatnu perifernu jedinicu ili ugraditi umesto jedne diske. Ekran od 12 inča može da bude crno beli sa standardnim ili zelenim tosfomom ili u boji sa 4 ili 8 boja. Grafika je sa 512 x 256 tačkica i 16 prozora. Struktura broja linija i znakova u redu je softvarski promenljiva i može da bude 16 x 64 ili 25 x 80. Tastatura ima poseban numerički deo a

funkcijski tasteri zamenjeni su kombinovanjem tastera, pa je moguće generisati do 256 znakova ili komandi. Postoje naši znaci i tastatura u skladu sa JUS-om.

Osnovna verzija M20 uključuje 1 paralelni priključak za štampač i jedan serijski RS 232. Kao dodaci mogu da budu ugrađeni: IEEE488 sa svojim najviše 14 perifera, kao i još najviše 4 RS 232 priključka, odnosno 4 strujne petlje. LAN (Local Area Network) priključak omogućava grupno korišće-

nje kapaciteta, na primer: datoteka na disku, za najviše 64 računara M20 povezanih u mrežu.

Od velikog broja raznih perifernih uređaja štampača, plotera, traka, kaseta, bušača čitača, itd., koje je moguće priključiti na M20, posebno treba izdvojiti matricni štampač PR 15 sa 120 zn/rišek širom od 8 inča 80 do 132 znaka u redu sa gustinom 10, 12 i 16 zn/inču ploterskim mogućnostima otiskom povećane vernosti i paralelnim Centronix ili serijskim RS 232 priključenjem.

Za povezivanje sa drugim sistemima na raspolaganju su protokoli: asinhroni npr.: TTY33, sinhroni BSC npr. IBM 2780, bit orjentisani npr.: IBM 3275 i X25.

## SOFTWARE:

Radi kompatibilnosti sa software-om drugih proizvođača, M20 koristi proširenje, elektronsku ploču sa INTEL 8086, 8 MHz, a postoje i verzije sa dva procesora na osnovnoj ploči. Tako je omogućeno korišćenje i operativnih sistema: MS/DOS (kao IBM PC), CPM/80 (kao i APPLE), CPM/86 i UCSD-p.

Sistemske Software pod PCOS-om obuhvata Assembler, Basic (Microsoft, interpreterski, za oko 50% brži od IBM PC-a), Pascal i Forth. Pod ostalim operativnim sistemima mogu da se koriste mnogi jezici, uključujući FORTRAN, kao i BASIC sa prevodjenjem.

Proizvođač nudi mnogo gotovih paketa od kojih posebno treba istaći:

- MULTIPLAN** — usavršena verzija VISICALC-A (MICROSOFT) za analizu i modelovanje
- OLIWORD** — obrada teksta (postoji verzija prevedena na srpskohrvatski)
- OLICHART** — grafički prikazi rezultata Multiplan analize
- OLISORT** — sortiranje
- OLITERM** — komunikacija asinhrona
- OLICOM** — komunikacija sinhrona
- OLEINTRY** — unos podataka i obrada
- OLISTAT** — statistički paket
- OLINUM** — numerička analiza
- OLIS275** — komunikacija IBM 3275

Aplikacione software pored korisnika radi i oko 400 specijalizovanih software-skih kuća, za koje Olivetti periodično publikuje katalog, a u nas je urađen jedan iz interaktivne grafike.

Dokumentacija za M20 u ovom trenutku kod zastupnika obuhvata biblioteku od 56 knjiga, koja raste po stopi od 2—3 mesečno, pa ovaj prikaz ličnog profesionalnog računara M20 treba shvatiti kao presek jednog vrlo dinamičnog procesa.

olivetti



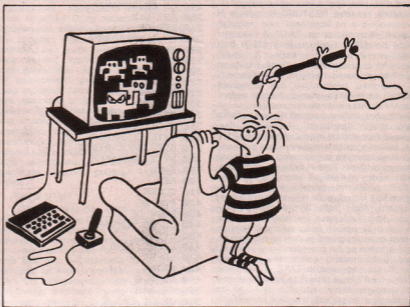
„DINARA-KOMERC“  
Vlajkovića 5, Tel.: 335-886  
BEOGRAD i 335-886.

# „galaksija“ bez tajni

Majstorije  
na računaru

Uputstvo za upotrebu računara „galaksija“ koje smo objavili kao podlistak specijalnog izdanja „Računari u vašoj kući“ bilo je namenjeno prevashodno početnicima. Osim njih, „galaksiju“ su napravili i mnogi koji su „prevazišli bežik“ i poželeli da pišu i mašinske programe. Njima, osim korisne mere sistemskih promenljivih, uputstvo nije pružalo ništa, nismo mogli da damo čak ni adrese korisnih rutina u ROM-u jednostavno zašto što one nisu bile fiksirane kada je uputstvo pisano. Za njih smo ovoga puta pripremili obilje programskih finesa za napredno programiranje računara „galaksija“.

Čim je razvoj ROM-a završen, ponudili smo svima zainteresovanima njegov disasembirani listing (i dalje možete da ga dobijate ako na žito-račun „Galaksije“ uplatite 200 din i pošaljete nam potvrdu o uplati uz napomenu da želite program GALA005052). On je praktično trenutno postao bestseler, ali smo ipak bili svesni da svi oni koji su ga naručili neće moći da izvuku mnogo koristi od njega. Da biste razumeli operativni sistem i bežik interpretator jednog računara (čak i ako, kao u našem slučaju, taj program ima samo 4 Kb), morate da poznajete mašinski jezik izuzetno dobro i uložite mnogo rada, strpljenja i, što je najvažnije, dosta vremena. Zato smo odlučili da vam u „Računarima 2“ omogućimo da podete prečicom: izdvojili smo sistemske adrese koje su nam se činile najkorisnijim, obradili kompletnu aritmetiku pokretnog zarez, proučili za vas korišćenje linkova za naredbe, video i (do sada nedokumentovanog) linka u interaptu i sve to sumirali u sledećih nekoliko strana. Obzirom da je sada pisanje mašinskih programa za „galaksiju“ postalo moguće na jedan od tri načina (o njima nešto doćnije), nadamo se da će podaci koje dajemo biti iskorišćeni na najbolji mogući način: za pisanje sistemskih i uslužnih programa i, naravno, efektnih igara.



## PTR

Dvadeset šesto poglavlje Uputstva za upotrebu računara „galaksija“ bilo je posvećeno naredbi PTR. Ona, da podsjetimo, daje memorijsku adresu neke numeričke ili alfanumeričke promenljive (PRINT PTR X\$, na primer, daje rezultat 10864, što je isto što i &2A70 prema mapi sistemskih promenljivih). No naredba PTR može da se koristi i bez adresnog dela, npr. A=PTR ili PRINT PTR (poslednja modifikacija „galaksijinog“ ROM-a pred njegovo „zaključavanje“ 3. januara 84.) U tom slučaju, naredba PTR daje memorijsku adresu na kojoj je sama smeštena. Šta će nam to? Ako se u memoriji

nalazi iole duži bežik program, nije ni malo jednostavno pronaći neko određeno slovo u njemu da bismo ga, pomoću BYTE, promenili. Ako pre toga slova stavimo jednostavno PRINT PTR i izvršimo program, direktno ćemo dobiti traženu poziciju (ne treba zaboraviti da će se ta adresa smanjiti za 9 bajtova ako izbrisemo naredbu PRINT PTR koja, računajući i blanko simbol, ima isto toliko slova).

Navedimo jedan primer upotrebe naredbe PTR. Dok lista program, računar „galaksija“ ne očekuje da naide na linijski broj 0. Ukoliko ipak naide na njega, računar smatra da je listanje završeno i vraća se u komandni mod prikazavši READY. Kako da kreiramo linijski broj 0? Ako želimo da prva naredba programa ima taj broj, ništa lakše: otkucaćemo WORD WORD(&2C36), 0, pa će „galaksija“ listati samo prvu liniju programa u kojoj može da piše ime njegovog autora ili, možda, A=USR(&2C3A) što bi stvorilo utisak da je program pisan na mašinskom jeziku. No, šta ako želimo da se listanje programa prekine posle neke druge linije? U prethodnu liniju ćemo, na samom njenom kraju, dodati naredbu PRINT PTR, a

zatim je i izvršiti. Dobićemo adresu prvog bajta iza slova R (od PTR). To je bajt &D koji označava kraj programske linije. Iza njega, kao što ćemo videti, stoji linijski broj sledeće naredbe koji možemo da promenimo u 0 primenom naredbe WORD. Ne treba da zaboravimo da obrišemo PRINT PTR, naredbu koja je obavila svoju funkciju. Ukoliko bismo ovu naredbu obrišali pre nego što promenimo linijski broj sledeće, morali bismo ponovo da preračunavamo adrese jer se sadržaj memorije pomerio.

Naredbu PTR ćemo mnogo više koristiti (opisani metod je prilično „opasan“ jer „galaksijini“ sistem traženja linijskih brojeva kod GOTO i CALL naredbi očekuje da ovi brojevi rastu) za pisanje programa koji menjaju samog sebe. Jedan od takvih je i „Telefonski imenik“, ilustracija Interfejsa 1, koji je objavljen u martovskoj „Galaksiji“. Ako se odlučite na sličnu „avanturu“, vodite računa da adresa početka i kraja jednog programa mora stalno da bude smeštena u sistemske promenljive &2C36 i &2C38. Odstupanje od ovoga može da ima neprijatne posledice po GOTO naredbe i, posebno, ispravke pojedinih linija.

# TAKE

Devetnaesto poglavlje Uputstva govori o naredbi TAKE koja je mnogim vlasnicima drugih računara poznata pod imenom READ. Radi uštede prostora u ROM-u, u bežikju uobičajena naredba RESTORE je zastupljena samo kao poseban slučaj TAKE. Ovoj mogućnosti smo do sada posvetili samo jedan pasus, propuštajući da spomenemo jedan interesantan trik koji je zamišljen još kada je TAKE koncipirana. Ako se iz TAKE nalazi neki broj n (npr. TAKE A,B,100,C), „galaksija“ će potražiti liniju sa tim brojem i postaviti pointer naredbe TAKE na njen početak. Ukoliko se u toj naredbi nalazi (DATA) LISTA, SLEDEĆE TAKE će uzeti podatak iz nje. Ukoliko se u naredbi uopšte ne nalazi lista, računac će potražiti sledeću naredbu, zatim sledeću...

Ovakva struktura naredbe TAKE je prilično fleksibilna i omogućava uštedu RAM-a: sve se dešava u istoj listi. Na prvi pogled, međutim, koncepcija ima veliku manu: simulacija naredbe RESTORE N, gde je N promenljiva a ne konstanta, nije moguća! Zaista, ako otkucamo TAKE A, računac neće potražiti listu u liniji čiji je broj smešten u A, već će promenljivoj A dodeliti vrednost iz prve sledeće liste. Šta da se radi? Mogli bismo, naravno, da smestimo N u sistemsku promenljivu čija je adresa &2A9D, ali je takvo rešenje „hleb sa sedam kora“: u ovu sistemsku promenljivu se ne smešta broj linije već adresa njenog početka posmatrano apsolutno u memoriji. Ovu adresu možemo da odredimo pomoću PTR, ali moramo da je menjamo svaki put kada pravimo ispravku u programu, što nije baš mnogo prijatno. Zato je bolje primeniti ovaj trik: kada „galaksija“ razmatra naredbu TAKE, bežik interpretator koristi potprogram koji određuje da li je sledeći element numerik. Ako jeste, biva pozvan potprogram koji izračunava vrednost izraza (ako je taj izraz običan broj, vrednost koja se vraća iz potprograma je sam taj broj, naravno) a zatim se on preračunava u vrednost koju treba smestiti u sistemsku promenljivu. A šta ako umesto TAKE A napišemo TAKE 0+A? Nula plus A je, kao što znamo iz osnovne škole, isto što i A, ali za „galaksiju“ to ne važi u svakoj konstelaciji događaja. Pri analizi naredbe računac naišao na broj nula koji prepoznaje kao numerik, podrazumevajući da se radi o simuliranoj naredbi RESTORE. Potprogram koji izračunava vrednost izraza, međutim, nije tako lako prevariti i on će svakako razumeti da je 0+A isto što i A, pa ćemo dobiti naredbu RESTORE A!

# KEY

Odmah iza naredbe TAKE govorili smo o naredbi KEY uz odgovarajuću tabelu u kojoj su se nalazila dva provokativna prazna mesta pa su se mnogi verovatno zapitali šta znači KEY(49) i KEY(51). Pripremajmo tabelu, imali smo puno razloga da izostavimo pomenuta dva koda jer u bežikju mogu samo da izazovu zabunu, ali ćemo sada govoriti o njima pošto mogu da budu od

```

10 !
20 ! CLEAR
30 !
40 ! DAJE VREDNOST NULA
50 ! NUMERICKIM PROMEN-
60 ! LJIVIMA A-Z
70 !
80 <
90 ORG &3000
100 OPT 3
110 LINK EQU &2BA9
120 KOMANDA EQU &75B
130 PREPOZNAJ EQU &39A
140 DFILE EQU &2B00
150 NUMVAR EQU &2A00
160 LD A,&C3 ! &C3 = JUMP
170 LD (LINK),A
180 LD HL,PROG
190 LD (LINK+1),HL
200 RET
210 PROG
220 EX (SP),HL
230 PUSH DE
240 LD DE,KOMANDA
250 RST &10
260 POP DE
270 JR Z,NAREDBA
280 EX (SP),HL
290 RET
300 NAREDBA
310 LD HL,TABLICA-1
320 JP PREPOZNAJ
330 TABLICA
340 TEXT "CLEAR"
350 BYTE CLEAR>B+&B0
360 BYTE CLEAR#&FF
370 BYTE BACK>B+&B0
380 BYTE BACK#&FF
390 BACK
400 RET
410 CLEAR
420 POP AF
430 PUSH DE
440 LD DE,NUMVAR
450 CYCLE
460 LD HL,ZERO
470 LD BC,4
480 LD IR
490 LD A,E
500 CP &68
510 JR NZ,CYCLE
520 POP DE
530 RST &30
540 ZERO
550 WORD &001B
560 WORD &4000
570 >
580 A=USR (&3000)

```

```

&3000: 3E C3 32 A9 2B 21 0C 30
&300B: 22 AA 2B C9 E3 D5 11 5B
&3010: 07 D7 21 28 02 E3 C9 21
&3018: 1C 30 C3 9A 03 43 4C 45
&3020: 41 52 B0 27 B0 26 C9 F1
&302B: D5 11 00 2A 21 3B 30 01
&3030: 04 00 ED B0 7B FE 6B 20
&303B: F3 D1 F7 18 00 00 40

```

velikog značaja pri pisanju mašinskih programa.

KEY(49) testira taster BRK, a KEY(51) taster DEL. BRK nema mnogo smisla (testirati jer pritisak na njega prekida izvršavanje programa (otkucajte 10 PRINT KEY(49);GOTO 10 i startujte program pokušavajući da pritisnete BRK samo na kratko,

toliko kratko da ga KEY registre, ali da se program ne prekine. Nije lako ali je moguće!

Testiranje DEL je još beznačajniji slučaj: pritisak na njega prekida izvršavanje programa i taj prekid traje sve dok korisnik ne otpusti DEL. U mašinskom programu, međutim, svakom od ovih tastera možete da dodelite proizvoljnu funkciju ukoliko ne koristite potprogram za testiranje tastature iz ROM-a. U mašinskim programima, štaviše, mogu da se testiraju i pritisnici na SHIFT BRK (npr. u „Zamku“ i „Jumping Jack-u“), programima koji su emitovani preko Ventilatora, ova kombinacija se koristi za prekidanje igre i novi start) i slične „nemoguće“ kombinacije.

Naredba KEY(0) koristi potprogram za rad sa tastaturom iz ROM-a, što znači da se javljaju isti problemi (sasvim očekivani, uzgred rečeno) sa testiranjem BRK i DEL, ali i jedna nova zagonetka: ako slučajno pritisnemo STOP/LIST kod se izvršava KEY(0), počeo listanje programa i samim tim prestati njegovu izvršavanje. Dalje ispitivanje bi nas uverilo da sličan artefakt postoji i kod naredbe INPUT. Ništa ozbiljno, ali o tome treba voditi računa u toku pisanja bežik programa.

# INPUT

Kada smo već kod naredbe INPUT, potrebno je da odgovorimo na jedno često postavljano pitanje: ako kristolimo naredbu INPUT X\$, tekst koji kucamo ne mora da bude pod navodnicima. Kada su, zapravo, navodnici obavezni? Najpre u  $\neq$  listama (što se vidi i iz spiska naredbi sa primerima) iz uputstva za upotrebu), a zatim i u naredbama za dodeljivanje koje bi mogle da izazovu zabunu. Možemo, dakle, da napišemo X\$=GALAKSIJA, ali ako u X\$ želimo da smestimo slovo Y i  $\neq$ , moraćemo ga iskoristimo X\$=Y\$, jer X\$=Y\$ ima drugi smisao! Kod „numeričkog INPUT-a“ imamo puno pravo da otkucamo neki aritmetički izraz umesto obične konstante. „Galaksija“ će ga izračunati i dodeliti promenljivoj koja je navedena u INPUT-u. U tekstu koji kucamo mogu da se koriste i promenljive i funkcije, ali tako da sintaksa bude ispravna; u protivnom, „galaksija“ će ispisati WHAT? ili HOW? (zavisno od prirode greške) i prestati sa radom. U toku INPUT-a, dakle, biva izvršena i naredba VAL.

# VAL

Naredba VAL je neobično moćna i treba je koristiti kad god je to moguće. Potreban nam je, na primer, program koji pretvara heksadekadne brojeve u dekadne. Možemo, naravno, da izdvajamo slovo po slovo iz promenljive X\$, pretvaramo ga u broj i primenjujemo uobičajeni algoritam množenja sa 16, ali postoji i daleko jednostavniji način: otkucamo:

```

10 INPUT X$
20 Y$ = &+X$
30 PRINT VAL(PTR Y$) i — to je sve. Ako u odgovoru na INPUT otkucamo 2A0, promenljiva Y$ će dobiti vrednost &2A0 a na ekranu će biti štampan broj 522, prevod 2A0 u dekadni sistem. Ako pišete program koji će vam omogućiti da unosite mašinski program dat kao „hex-dump“ u memoriju, program poput ovog će vam uštedeti čitavu trećinu kucanja podataka!

```

Postoji još mnogo načina da lukavo

iskoristite potprograme operativnog sistema bez potrebe da se pozabavite mašinskim jezikom. Za pripremanje pomenutih heks-dampova koji bi bili pogodni za listanje i objavljivanje u „Galaksiji“, bilo je neophodno da sastavimo program koji će dati mašinski program pretvarati u naredbe. Za to je bilo potrebno pretvarati numerike dobijene pomoću BYTE u ASCII vrednosti i od njih kreirati listu (PROGRAM koji samoga sebe menja). Za ostvarivanje toga poslužili smo se malim trikom: obrišemo ekran sa HOME, štampano vrednost koju treba pretvoriti u ASCII karaktere a zatim iz same video memorije (od &2801 pošto se na &2800 nalazi znak koji je za vrednosti koje su nam potrebne (do 255) uvek blanko jer su na brojevi pozitivni) čitamo jednu po jednu cifru običnim BYTE!

Korišćenje rutina operativnog sistema, naravno, nosi sa sobom i određene opasnosti da će nas računari „pogrešno shvatiti“. Svi oni koji su napisali više mašinskih programa počeli su da „misle heksadekadno“, tj. da koriste heksadekadni sistem za predstavljanje svih programerskih problema. Za njih će, dakle, biti sasvim prirodno da umesto NEW 1024 otkucaju NEW &400 rezervišu kilobajt za mašinski i program koji će biti smešten neposredno ispred bejzika. Računari „galaksija“ ih, na žalost, neće razumeti! Iza NEW i OLD se, naime, mora nalaziti numerik, a ne izraz. NEW &400 će biti shvaćeno kao obično NEW i računari, da stvar bude gore, neće to prijaviti ni na kakav način. Docijnje može da dode do mešanja bejzika i mašinskog programa sa katastrafalnim posledicama po oba. Ako, dakle, koristimo neku opciju koja nije dokumentovana, moraćemo najpre da proverimo njeno dejstvo.

## Minsko polje

Skijaše-početnike uču da se zaustave pre nego što nauče da krenu niz padinu, pa ćemo i mi, pre nego što počnemo da govorimo o mašinskim programima, izložiti koje memorijske zone i sistemske registre treba po svaku cenu čuvati ili, bar, vratiti u prvobitno stanje.

Postoje dve vrste mašinskih programa: jedni se koriste iz bejzika kao potprogrami, što znači da, posle njihovog izvršavanja, „galaksija“ treba da se vrati u bejzik. Da bi se takva kooperacija ostvarila, potrebno je, očigledno, dobro čuvati sve ono što je interpretatorov radni prostor. Postoje, sa druge strane, i mašinski programi sasvim nezavisni od bejzika i ROM-a koji, na primer, predstavljaju interpretatore drugih programskih jezika ili (mnogo verovatnije) kompletne igre. Reklo bi se da ovi programi ne moraju ništa da čuvaju i da mogu proizvoljno da se prostiru u memoriju. No, odmah ćemo videti da nije baš tako.

Treba, pre svega, da razumemo da računari ne izvršava naš program sve vreme. Svake pedesetinke sekunde procesor prekida rad na našem programu i pomaže video-stepenu da generiše sliku na ekranu monitora. Ovu rečenicu pišemo već po ko zna koji put ali za to imamo mnogo razloga: iz ličnog iskustva nam je poznato da je vrlo teško naći grešku koja je nastala kao posledica interaptja koji programer često

# KAKO SE UNOSE MASINSKI PROGRAMI

U okviru poglavlja koje je posvećeno malim tajnama računara „galaksije“ dajemo dosta primera koji treba da ilustruju opisane tehnike. Neki od tih primera, međutim, mogu da budu interesantni i kao uslužni programi koje biste rado uneli u računari i koristili u raznim prilikama (ovde najpre mislimo na programe RSAVE i TIME). Za to unošenje će poslužiti hex-dump svakog od tih programa i kratka rutina UTM koju ovde donosimo.

Mašinski program se u računari unosi na jedan od tri načina pomoću UTM-a ili nekog sličnog programa, pomoću asemblera ili pak krajnja solucija, preko asemblera nekog drugog računara koji je povezan sa „galaksijom“. Kada ćemo koristiti koju od ovih solucija? Drugi računari je neobično zgodan ako pišemo komplikovaniji sistemski program (npr. ROM 2) koji će, u toku razvoja, više puta „krahirati“. Posle svakog ovakvog „krah“, moraćemo da izvršimo „galaksijinu“ memoriju pri čemu izbrisemo, asemblerski, program u memoriji drugog računara neće biti izgubljen. Korišćenje asemblera programa je zgodno kada pišemo igre ili neke druge uslužne programe i, naročito, kada posedujemo ovakav assembler. Program UTM ostaje kao rešenje neobično zgodno za unošenje gotovih programa koji su objavljeni u nekom časopisu i čije nas modifikacije ne zanimaju — želimo samo da otkucamo program i da ga koristimo.

## Šta je UTM?

UTM je, naravno, Ultra Tiny Monitor. Kada pogledate koliko je dugačak, verovatno ćete pomisliti da ni ono „Ultra Tiny“ u nazivu ne ilustruje dovoljno njegove slabosti u odnosu na prave monitor programe. Ali nije baš tako — videćete da je UTM, za primene za koje je napisan, sasvim dovoljan. On omogućava unošenje programa u proizvoljni segment memorije, testiranje i promenu sadržaja proizvoljne memorijske lokacije i kontrolu sadržaja većih segmenata.

```

5 CALL 240
10 PRR*(4)
20 Y$=""
30 I=0
40 X$=""
50 H=I*CALL 160
60 PRINT " ";
70 H=BYTE(I)*CALL 160
80 PRINT " ";
90 INPUT X$
100 IF EQ X$,Y$ I=I+1:GOTO 50
110 IF <BYTE(PTR X$+1)=0>+(BYTE(PTR X$+2)=0) CALL 130:BYTE I,H:X$="" :GOTO 100
120 CALL 130:I=H:GOTO 40
130 X$(0)="E"+X$
140 H=VAL(PTR X$(0))
150 RETURN
160 FOR K=4 TO 1 STEP -1
170 G=H
180 H=INT(H/16)
190 G=G-16XH
200 IF G<10 X$(K)=CHR$(G+48) ELSE X$(K)=CHR$(G+55)
210 NEXT K
220 FOR K=1 TO 4:PRINT X$(K):NEXT K
230 RETURN
240 HOME
250 PRINT AT 400,"GALAKSIJA"
260 PRINT
270 PRINT "UTM .VERSION 1.0"
280 PRINT
290 RETURN

```

ta memorije. Evo uputstva za njegovu upotrebu.

Po startovanju programa, „galaksija“ ispisuje njegovo zaglavje i upitnik. Kada god se pojavi upitnik možete da izaberete jednu od sledećih opcija:

Ukoliko otkucavate neki četvorocifren heksadekadni broj, kome ovde ponašamo da prethodi oznaka &, računari će prikazati sadržaj memorijske ćelije čija je adresa data tim brojem i proglasiti tu adresu za „tekuću“.

Ukoliko otkucavate neki jedno ili dvoциfren heksadekadni broj, kome ne sme da prethodi oznaka &, računari će taj broj smestiti u memorijsku ćeliju čija je adresa u tom momentu označena kao „tekuća“. Zatim će automatski biti izvršena sledeća opcija.

Ukoliko jednostavno pritisnete RET odnosno ENTER, tekuća vrednost će biti povećana za jedan, pa će na ekranu biti prikazan sadržaj te memorijske ćelije koju zatim slobodno možete da menjate primenom opcije 2.

Kako da otkucate neki mašinski program pomoću UTM-a? Startuje UTM sa RUN (ne smete da zaboravite da, ukoliko mašinski program počinje od &2C3A, treba da pomerite UTM kucajući NEW NNNN pr kucanju UTM-a ili NEW NNNN OLD NNNN WORD &2C38, NNNN+&2E3E pr njegovog unošenja sa kasete; NNNN je bar za jedan veći broj od dužine mašinskog programa, pri čemu treba zaboraviti da NNNN mora da bude dekadni broj), otkucate početnu adresu programa a zatim, bajt po bajt, sadržaje svake memorijske lokacije. Koristiće stalno heksadekadni sistem, koji je uobičajen kod programera. Posle unošenja poslednjeg bajta, možete da pritisnete BRK i tako prekinete rad ili ponovite početnu adresu i, uzastopnim pritiscima tastera ENTER, proverite sadržaj memorije koji treba da odgovara hex-dampu. Ukoliko pronađete grešku, lako ćete je ispraviti kucanjem novog sadržaja.

potpuno zaboravlja jer se dešava van njegovog uticaja. Da bi računar korektno izvršio interapt, nužno je da ni po koju cenu ne menjamo sadržaj procesorskog IY registra. Ukoliko to učinimo (pa makar se restituirali samo nekoliko naredbi doncije), rizikujemo da računar posle interapta počne da izvršava program na nekom drugom mestu u RAM-u pa ćemo, pogotovu ako je prostor sistemskih promenljivih poremećen, izazvati siguran krah praćen reinicijalizacijom računara i gubitkom programa.

Ma koliko se trudili da naš program bude nezavistan od bejzika, za njegovo testiranje će biti korisno da se ipak oslonimo na njega. Nema toga ko će napisati sistemski program tako da proradi „iz prve“. Ukoliko se program razvija na „galaksiji“, posle svake probe ćemo izgubiti program ukoliko ne obezbedimo mogućnost da bejzik interpretator preuzme kontrolu. Čak i ako program razvijemo na nekom drugom računaru, poželjećemo da koristimo potprograme iz ROM-a, a zahtevaju određene predostojnosti.

Stek mora da vam bude „najveća svetinja“. Ukoliko vaš program ima, u nekoj grani, više naredbi PUSH nego naredbi POP, povratak u bejzik će biti nemoguć, jer RET nalazi adresu za povratak na vrhu steka. Isto tako, ako se stek stalno puni, računar će se garantovano „ugušiti“ posle izvesnog vremena. Simptomi ove greške su programi koji jedno vreme ispravno rade, a onda nastupa „krah“ praćen brisanjem čitave memorije.

Dakle, svaka operacija sa stekom, a naročito direktna izmena registra SP, treba da bude plod pažljivog planiranja (ovim, naravno, nikako ne želimo da kažemo da ne treba koristiti stek — bez njega se ne pišu ni najkraći programi) koje će imati za cilj da sve bude vraćeno u prvobitno stanje. Vodiće, na primer, računa o tome da poziv potprograma iz koga se ne vraćamo dodaje dva broja na stek. Kao i svako pravilo, i ovo ima izuzetaka: ako se ne vraćate u bejzik pomoću RTN već pomoću JP & 66, ne morate previše da brinete o steku; posle ovog skoka „galaksija“ ispisuje READY i reinicijalizuje registar SP (u programerskom žargonu ova adresa se zove „farma“, pa bi zagriženiji hakeri prethodnu rečenicu izgovorili kao „ne brini o steku ako računar posle ide na farmu“).

Osim steka adresiranog registrom SP, „galaksija“ ima još jedan stek adresiran registrom IX — aritmetički stek čiju ćemo strukturu izložiti kada budemo govorili o aritmetici u pokretnom zarezu. Ovaj stek, ukratko, sadrži argumente aritmetičkih operacija i ne treba ga previše remetiti. Ponekad će nam, naravno, zatrebati registar IX (pošto IY i tako ne smemo da koristimo), pa ćemo ga najpre sačuvati (sa PUSH IX), onda koristiti i najzad vratiti u početno stanje (POP IX).

```

&3000: 3E C3 32 A9 2B 21 0C 30
&3008: 2E AA 2B C9 E3 D5 11 77
&3010: 07 D7 D1 2B 02 E3 C9 21
&3018: 1C 30 C3 9A 03 44 49 53
&3020: 50 4C 41 59 B0 29 B0 2B
&3028: C9 F1 CD 6A 0A 25 11 03
&3030: 2B 19 D1 7E 6F 26 00 C3
&3038: BC 0A
  
```

```

10 :
20 :          DISPLAY (PPP)
30 :
40 :  DAJE ASCII KOD KARAKTERA
50 :  U VIDEO MEMORIJI CIA JE
60 :  ADRESA <PPP> PREMA MAPI
70 :
80 <
90 ORG &3000
100 OPT 3
110 LINK EQU &2BA9
120 FUNKC EQU &777
130 PREPOZNAJ EQU &39A
140 DFILE EQU &2B00
150 KRAJHL EQU &ABC
160 IZRAZ EQU &A6A
170 LD A,&C3 ! &C3 = JUMP
180 LD (LINK),A
190 LD HL,PROG
200 LD (LINK+1),HL
210 RET
220 PROG
230 EX (SP),HL
240 PUSH DE
250 LD DE,FUNKC
260 RST &10 ! POREDIL HL I DE
270 POP DE
280 JR Z,NAREDBA
290 EX (SP),HL
300 RET
310 NAREDBA
320 LD HL,TABLICA-1
330 JP PREPOZNAJ
340 TABLICA
350 TEXT "DISPLAY"
360 BYTE DISPLAY#&FF00>B+&B0 !
370 BYTE DISPLAY#&FF
380 BYTE NEGDE#&FF00>B+&B0
390 BYTE NEGDE#&FF
400 NEGDE
410 RET
420 DISPLAY
430 POP AF
440 CALL IZRAZ
450 PUSH DE
460 LD DE,DFILE
470 ADD HL,DE
480 POP DE
490 LD A,(HL)
500 LD L,A
510 LD H,0
520 JP KRAJHL
530 >
  
```

savim specijalan način, o kome ćemo govoriti nešto doncije) pointera početka i kraja bejzika. Promena drugih sistemskih promenljivih ne može da ima katastrofalne posledice, ali može da izazove male neprijatnosti (ako, na primer, promenite sadržaj sistemskih promenljive &2BA8, slika će se pomeriti što ćete lako ispraviti).

Ne bismo želeli da pomislite da je najbolje ne dirati sistemske promenljive: pisanje mašinskih programa, pogotovu ako se radi o rutinama sistemske prirode, ne može ni da se zamisli bez intervencija u ovom bloku memorije. Hoćemo samo da vas upozorimo da svaka promena neke od sistemskih

promenljivih mora da bude rezultat razumevanja njene funkcije i situacija kada se ta funkcija ostvaruje.

Pažljiv pogled na tabelu sistemskih promenljivih otkriva da u njeemu ima neiskorišćenog prostora — bajtovi &2A97, &2A98, &2AAA i &2A90 nisu korišćeni, pa bi mogli lepo da se iskoriste za smeštaj nekih podataka koji su našem programu potrebni. Da li je pametno koristiti ih? Nismo sigurni u to — računar „galaksija“ će biti i dalje proširivan, pa sasvim moguće da će neka od varijanti ROM-a ili iskoristiti prazan sistemski prostor. U tom slučaju, moraćete od preradjuje vaše programe kako bi ostali kompatibilni i sa proširenjima računara. Teško da će vaše programi biti baš toliko dugačak da mu zafale tačno četiri bajta, pa vam savetujemo da prostor za podatke potražite na nekom drugom mestu. A jedno od takvih mesta je već poznato — bafer koji se prostire od &2BB6 do &2036. Ovih 126 bajtova su prostor koji se u toku izvršavanja programa svakako ne koristi, pa može zgodno da posluži kao zona u koju bi se privremeno smestali podaci. Zašto kažemo privremeno? Ukoliko se izvršavanje programa prekine, svakako će biti potrebno da korisnik otkuca nešto da bi program nastavio da radi. To nešto biva smešteno u bafer, čime jedan deo njegovog sadržaja nepovratno strada. Treba, takođe, biti vrlo oprezan sa upotrebom bafera ako se koristi naredba INPUT za primanje podataka.

Mašinske programe i podatke koji su za njih potrebni možete slobodno da smestate i na kraj RAM-a, ali je korisno pre toga promeniti sadržaj sistemske promenljive &2A6A. Područje iznad RAMTOP-a je bezbedno i na njega ne utiču ni naredbe za editovanje programa ni operacije sa numeričkim i alfanumeričkim nizom. Potrebno je ipak da obratimo pažnju na probleme koji mogu da nastupe ako jednog dana proširimo memoriju „galaksije“ ili ako program damo nekom prijatelju koji je to već učinio.

## Nove naredbe

ROM računara „galaksija“ je koncipiran tako da omogućava relativno jednostavno dodavanje novih naredbi i funkcija. Da bismo dodali novu naredbu ili funkciju nije, jasno, neophodno da razumemo čitavu strukturu registar interpretera (dovoljan je „recept“ koji ćemo, korak po korak, dati u okviru ovoga poglavlja), ali nije loše da uočimo nekoliko osnovnih stvari.

Pri analizi bejzik programa računara „galaksija“ u registru DE čuva adresu bajta koji se trenutno obrađuje. Kada pišemo običan mašinski potprogram koji se poziva sa USSR registru DE, će biti automatski sačuvan na steku. Ako, pak, želimo da dodamo novu naredbu, potrebno je da se uklopimo u sam interpreter, tj. da po svaku cenu čuvamo sadržaj DE registra. Ukoliko naša naredba ima i neki adresni deo, potprogram koji je podržava mora da primi i taj deo, vodeći računa da registar DE, po povratku u bejzik, „pokazuje“ na prvi neobrađen bajt iz.

Kao i DE, registar IX permanentno pokazuje stanje aritmetičkih akumulatora i treba ga, ukoliko nije vršeno računanje koje vraća rezultat bejziku, ostaviti nepremnjen. Naš program mora da se pobrine i za sve moguće sintaksne greške. Ako pogledate listing ROM-a, videćete da je dosta prostora posvećeno proveru sintaksne i rutinama za tretiranje grešaka. Tako, jednostavno, mora da bude. Ukoliko interpretat



## VAŽNE ADRESE „GALAKSIJINOG“ OPERATIVNOG SISTEMA

Funkcija	Poziv: CALL &
a) Aritmetika pokretnog zarezca	
Izraz adresiran sa DE · (IX)	AB2
HL · (IX)	ABC
(HL) · (IX)	A45
(IX) · HL	A6D
(IX) · (HL)	73B
Izraz u zagradi adresiran sa DE · (IX)	781
Sabiranje	B32
Oduzimanje	B1E
Množenje	A6E
Deljenje	AF7
Poređenje elemenata na vrhu aritm. steka	B10
RND · (IX)	C8F
b) Adrese bezik interpretatora	

Izraz adresiran sa DE (celobrojno) · HL	008:RST &8
Izraz u zagradi adresiran sa DE · HL	A6A
Preskakanje blankova adresiranih sa DE	104
Ako DE adresira zarez, kao RST &8 inače WHAT?	5
KEY(O) uz podršku BRK, DEL i LIST	CF5
Karakter iz A na ekran uz &D, &C i &1D	20:RST &320
Štampanje HL na ekranu kao ASCII niz	8FB
Štampanje alfanumerika adresiranog sa DE;	
terminator je OO (nema novog reda) ili &D	
(izdaje se novi red)	937
Lociraj varijablu čije ime pokazuje DE. U HL se	
smesta PTR; ako je C setovan nije varijabla; ako	
je Z setovan numerička varijabla a u protivnom	
alfanumerička	125
Prepoznavanje sledeće naredbe	30:RST &30
Stanje aritmetičkog steka	8F6
Slobodna memorija (od ARRS do bezjika) · HL	183
Snimanje bajta na kasetu. Bajt je u A	E68
Bajst sa kasete · C	EDD

(a dodajući naredbe i vi pišete taj interpretator) ne reaguje na grešku, može lako da se desi da računar krahira i da se program izgubi. Kada god u vašoj naredbi sintaksa koju je korisnik otkucao nije ispravna, treba da predvidite naredbu JP &78F (poruka WHAT?) ako je sintaksa ispravna ali argumenti ne odgovaraju — naredbu JP &65A (poruka HOW?), a ukoliko je memorija premla za ono što je korisnik želeo da uradi — JP &153 (SORRY). Pri tome ne morate mnogo da razmišljate o steku (SP će biti inicijalizovan pre nego što „galaksija“ napiše READY), ali morate da obratite pažnju na registar DE i sistemsku promenljivu &2A9F. Ukoliko se u &2A9F (i &2AA0, naravno) nalazi nula, računar će jednostavno izdati poruku WHAT?. HOW? ili SORRY. Ukoliko je u &2A9F nešto drugo, računar smatra da mora da prikaže i liniju u kojoj se nalazi greška. Trebalo bi da DE pokazuje na prvi bajt koji nije bio razumljiv kako bi na tom mestu bio štampan upitnik. Ukoliko DE pokazuje neki bajt ove naredbe, sve će biti u redu i upitnik štampan na pravom mestu. Ukoliko DE ne pokazuje ispravno, biće štampana linija označena promenljivom &2A9F sa upitnikom na kraju. To se, na primer, dešava ako TAKE listi ne odgovaraju podaci u # listama.

Već smo u prethodnom pasusu videli koliko za nas mogu da učine programi koji već postoje u ROM-u. Moramo da naučimo da se njima služimo što više, jer će nam tako nove naredbe biti mnogo kraće, pisaćemo ih naupoređivo lakše i brže i jednostavnije ćemo ispravljati greške koje smo napravili. Čitanje „recepta“ koji sledi će vas, nadamo se, konačno uveriti u korist od sistemskih programa.

Link za naredbe zauzima tri bajta čije su adrese &2BA9, &2BAA i &2BAB. U prvom od ovih memorijskih ćelija se normalno nalazi C9, što je kod mašinske naredbe RET dok druge dve nisu važne. Ukoliko želimo da dodajemo nove naredbe, treba da promenimo bajt čija je adresa &2BA9 u C3 (kod naredbe JP), a u sledeća dva da smestimo adresu na kojoj se nalazi naš program koji prepoznaje nove naredbe. Ukoliko taj program počinje od &3000, upotrebicemo naredbe:

LD A,&C3

LD (&2BA9), A

LD HL,&3000

LD (&2BAA), HL

Program koji počinje od &3000 treba najpre da prepozna da li je vreme da se naide na naredbu ili na funkciju. U čemu je razlika? Jednostavno rečeno funkcije se razlikuju od naredbi po tome što se nalaze desno od znaka jednakosti. Tako je, na primer, PRINT naredba a INT funkcija. Postoje neke reči koje mogu da budu i naredbe i funkcije (BYTE, WORD, DOT), ali to ne treba da vas zbuni: pogled na listing ROM-a će vas uveriti da je svaka od ovih reči u memoriji smeštena po dva puta i da se potpuno različito tretira u zavisnosti od toga da li se trenutno traži naredba ili funkcija. Mi ćemo, najpre, pokušati da uvedemo jednu naredbu.

### Tabela naredbi

Kada očekuje da prepozna naredbu, „galaksija“ će na link doći pomoću naredbe CALL i to sa adrese &75B. Obzirom na to, na vrhu steka će se u tom slučaju nalaziti bajtovi &07 i &5B. Pošto nam je i tako potrebno da sačuvamo HL registar, prva naredba našeg programa će biti EX (SP). HL čime smo u HL registar doveli adresu sa koje je dođeno na link. Sada treba da proverimo da li se u HL registru nalazi broj &75B. To ćemo uraditi tako što ćemo najpre sačuvati registar DE (PUSH DE), zatim u njega uneti ovu konstantu (LD DE, &75B) a onda izvršiti RST &10. Time je pozvan sistemski program koji upoređuje registre HL i DE upotrebom svega, jednog bajta. Po povratku sa ovoga programa Z i C flagovi su setovani u zavisnosti od rezultata poređenja. Ukoliko je, dakle, Z flag resetovan registri HL i DE nisu bili jednaki, pa treba vratiti kontrolu bezik interpretatoru. To radimo tako što, najpre, restauriramo DE i HL i izvršimo jednostavno RET.

Ukoliko je Z setovan, treba da pokušamo da prepoznamo naredbu. Najpre ćemo, za svaki slučaj, vratiti DE u normalno stanje (POP DE), a zatim ćemo u HL smestiti adresu tablice umanjenu za jedan i preći na sistemski program čija je ulazna adresa &39A.

LD HL, TABLICA-1

JP &39A

Potrebno je još da formiramo tablicu koja se sastoji od naziva bezik naredbi koje dodajemo, pri čemu se iz svake od njih

nalazi ulazna adresa mašinskog programa koji je ostvaruje. Tekst se, jednostavno, smesta u memoriju slovo po slova, ali je ulazna adresa priličan problem. Ona se odnosi na neki program koji se nalazi bilo gde u prvih 16 Kb memorije, pa su za njeno smještanje potrebna dva bajta. Prvi od njih daje značajniji bajt adrese (veoma je važno da učite da ovo odstupa od standardne Zilogove mnemonike kod koje najpre dolazi manje značajan bajt adresa), pri čemu mu je najznačajniji bit setovan. Ovo setovanje bita je neophodno da bi računar razlikovao kraj imena od početka ulazne adrese.

Posle drugog, manje značajnog bajta adrese, sledi ime sledeće naredbe koju dodajemo, zatim sledeće i tako dalje. Posle poslednje naredbe stavljamo još jednu ulaznu adresu kojoj ne prethodi ni jedna reč. Ta adresa pokazuje bilo koji bajt prvih 16 K memorije na kome se nalazi naredba JP &75B. Potrebno je, dakle, da računar nekako prepozna kraj tabele kako bi znao da nema više novih naredbi.

Naredbu koju dodajemo treba da započnemo jednim POP AF koje će rešiti problem steka koji je poremećen CALL-om na link koji nije završen sa RET. Ovaj POP treba da zaboravimo čim smo ga stavili u program: ne treba ga kompenzovati nikakvim daljim PUSH, niti, smo njim u AF registre doveli neke podatke od značaja.

Slede instrukcije koje izvršavaju naredbu i njen završetak sa RST &30. Posle ovoga bezik interpretator će pokušati da prepozna sledeću naredbu koja je adresirana DE registrom.

Koje korisne sistemske programe još možemo da upotrebimo? Potrebno je, pre svega, da prenesemo neke argumente nove naredbe. Ukoliko je naredbi potrebno samo jedan argument koji sledi iza nje u zagradi i po prirodi predstavlja ceo broj, stvar je vrlo jednostavna: u okviru tabele ćemo setovati bit 6 prvog bajta ulazne adrese (sada znamo zašto se sve mora nalaziti u prvih 16 Kb — najstarija dva bita ne služe za adresiranje). Izraz u zagradi će biti izračunat, računar će naci ceo deo tog izraza i poslati ga našem programu kao sadržaj registra HL. Register DE će u tom slučaju pokazivati prvi bajt iza zagrade.

Ukoliko iz nekih razloga ne želimo da setujemo bit 6 u tabeli (to će se, na primer, dogoditi ako realizujemo naredbu koja ima različite moguće tipove adresnog dela), možemo da započnemo naredbu sa CALL &A6A, što ima istu funkciju. Ukoliko se argument ne nalazi u zagradi, iskoristićemo

RST &8 koji će izračunati vrednost izraza i njen code deo smestiti u HL registar. Posle toga, naravno, može da sledi zarez koji odvajaj prvi argument od drugog. Za detektovanje toga služi CALL &5. Ako DE pokazuje zarez, on će biti preskočen i sledi drugi izraz vraćen kao sadržaj registra HL, a ako ne pokazuje biće prikazano WHAT? prema pravilima koja smo ranije definisali.

Možda ćemo poželeti da jednostavno preskočimo blankove i dodemo do sledeće značajne bezijk konstrukcije. To radimo sa CALL &104. Ova rutina će nam koristiti jedino ako želimo samo da detektujemo ono što se nalazi iza naše naredbe; u protivnom, do sada pomenuti potprogrami obavice i ovo zaobljaženje blanko simbola.

Pošto dodajemo naredbu, nije nam dovoljno samo da prenesemo argumente — naša nova naredba treba da ima i neko dejstvo. To će retko biti neko jednostavno pomeranje memorije — obično treba kontaktirati sa ekranom ili tastaturom. I tu mogu mnogo da pomognu sistemski pozivi.

RST &20 štampa na ekranu karakter čiji je kod smešten u registrar A. Pri tome će se, ako je taj bajt &D, preći u novi red, obrisati čitav ekran ako je poslat bajt &C, a obrisati poslednji karakter ako je u A &D.

Ako se ne zadovoljavamo štampanjem jedne jedine cifre, poželedimo da koristimo CALL &937. Ovaj poziv na ekranu štampa čitav alfanumerik adresiran DE registrom i to sve do koda &D (izdaje se novi red na ekranu) ili do koda &00 koji ne izaziva novi red. Debugovanju programa će pomoći i mogućnost štampanja vrednosti registra HL kao ASCII niza slova — to se postiže sa CALL &8FB.

## Očitavanje tastature

Tastaturu možemo da testiramo na više načina. Filozofiji mašinskog jezika najviše odgovara da se poslužimo memorijskim prostorom &2000 - &2037. Svakom bajtu ovoga prostora dodeljen je po jedan taster prema tablici KEY(N). Najmanje signifikantan bit ovoga bajta je setovan ako je taster pritisnut, a resetovan ako nije. Ako, na primer, želimo da detektujemo da li je taster A (kod 1) pritisnut, izvršićemo program:

```
LD A,&(2000+1)
AND 1
JP Z,PRITISNUT.
```

Mnogo je, međutim, komfornije koristiti potprograme za tastaturu. CALL &CF5, na primer, vraća u A registar kod zadnjeg pritisnutog tastera (kao KEY(0)), pri čemu, ukoliko je BRK pritisnut, računar prikazuje READY a ako je pritisnuto DEL privremeno prekida sa radom. Najneprijatnije je to što STOP/LIST započinje u takvoj prilici najverovatnije neželjeno listanje programa. Sistemski potprogram &CF5 podržava, pored ostalog, i naredbu REPT za ponavljanje teksta.

Ukoliko želimo samo da računar prekine sa radom ako je pritisnuto BRK, a malo pričekaj ako je DEL u akciji, izvršićemo samo CALL &2FF.

Interakcija bezijk sa mašinskim programima može da se odvija i preko promenljivih A-Z, X5, Y5, A(I), i X5(I). Za to je najbolje upotrebiti CALL &125, što locira varijablu na čije ime pokazuje DE. Po završetku rada

```
10 !
20 ! REC (RECIPRODNA VREDNOST)
30 !
40 ! ILUSTRACIJA FLOATING
50 ! POINT ARITMETIKE
60 !
70 <
80 ORG &3000
90 OPT 3
100 LINK EQU &2BA9
110 FUNKC EQU &777
120 PREPOZNAJ
130 NUMIX EQU &A45
140 IZUIX EQU &7B1
150 DELI EQU &AF7
160 LD A,&C3 ! &C3 = JUMP
170 LD (LINK), A
180 LD HL,PROG
190 LD (LINK+1),HL
200 RET
210 PROG
220 EX (SP),HL
230 PUSH DE
240 LD DE,FUNKC
250 RST &10
260 POP DE
270 JR Z,NAREDBA
280 EX (SP),HL
290 RET
300 NAREDBA
310 LD HL,TABLICA-1
320 JP PREPOZNAJ
330 TABLICA
340 TEXT "REC"
350 BYTE REC>B+&B0
360 BYTE REC<FF
370 BYTE NEGDE>B+&B0
380 BYTE NEGDE<FF
390 NEGDE
400 RET
410 REC
420 POP AF
430 LD HL,JEDAN
440 CALL NUMIX
450 CALL IZUIX
460 CALL DELI
470 RET
480 JEDAN
490 BYTE 0 ! BROJ 1
500 BYTE 0
510 BYTE &B0
520 BYTE 0
530 >
540 A=USR(&3000)
```

```
&3000: 3E C3 32 A9 2B 21 0C 30
&300B: 22 A8 2B C9 E3 D5 11 77
&3018: 07 D7 D1 2B 02 E3 C9 21
&3019: 1C 30 C3 9A 03 52 45 43
&3020: B0 25 B0 24 C9 F1 21 33
&3028: 30 CD 45 0A CD 81 07 CD
&3030: 7F 0A C9 00 00 80 00
```

u HL registru se nalazi apsolutna pozicija promenljive (poput naredbe PTR u bezijku). Carry flag je setovan ako DE ne pokazuje na nešto što bi moglo da bude promenljiva. Ukoliko je C flag resetovan, Z flag govori o vrsti promenljive: setovan Z označava numeričku, a resetovan — alfanumeričku promenljivu.

Sve što smo govorili o dodavanju naredbi ilustruje i program na slici 1. Unesite ga u računar (pomoću hex-loadera) i startujte, pa će set naredbi vašeg računara biti dopunjen naredbom CLEAR. Ona dodeljuje svim numeričkim i promenljivima (A-Z) vrednost nula, što je neobično kori-

sno pri inicijalizaciji programa (kada normalno otkače RUN, promenljive zadržavaju vrednosti koje su dobile u prethodnom programu ili, ako od uključivanja računara nisu korišćene, sadrže broj 0).

Zašto smo se odlučili za naredbu koja ne opšti sa ekranom, ne prima ništa sa tastature niti zahteva neke argumente? Jednostavno smo želeli da ilustrujemo sam princip dodavanja naredbi dok za prenošenje argumenta ima vremena — njega ćemo ilustrovati u sledećim redovima.

## Nove funkcije

Funkcije se, kao što rekohmo, uvek nalaze sa desne strane znaka jednakosti kao deo aritmetičkog izraza. Najveći deo funkcija (ovde nije reč samo o računaru „galaksija“) zahteva jedan argument koji se da bi se poštovala logika sintakse bezijka, obavezno stavlja u zagradu (kod nekih novijih kompjutera ovaj argument može ali ne mora da se stavi u zagradu, ali u njenim korišćenjem eliminiše mogućnost zabune u slučajevima kada je argument čitav izraz). Kada naiđe na funkciju, dakle, računar napre izračunava vrednost izraza u zagradi, zatim na tu vrednost primenjuje funkciju i dobijenim rezultatom dalje operiše u aritmetičkom izrazu.

Program za dodavanje funkcija je u mnogo čemu sličan programu za dodavanje naredbi, mada postoji i nekoliko razlika. Prva i najvažnija razlika je činjenica da se na ovaj program ne stiže sa lokacije &75B već sa lokacije &777. Na početku programa, dakle, ne treba u DE registar smestiti &75B već &777 pre nego što se izvrši RST &10. Isto tako, na kraju tablele ne treba ići na lokaciju RAM-a na kojoj se nalazi JP &75B, već negde treba formirati JP &777, pa se sa kraja tablele uputiti na to mesto.

Možda je najvažnije da se program koji izvršava funkciju završava sa RET ili JP &ABC, a ne sa RST &30. Kada koristimo RET a kada JP &ABC? Kada je rezultat neki code broj koji se nalazi u HL registru, korišćićemo JP &ABC da bi taj broj bio smešten na aritmetički stek. Ukoliko smo već računali sa brojevima u pokretnom zarezu tako da se rezultat nalazi na aritmetičkom steku, upotrebićemo obično RET. Kako za sada znamo da radimo samo sa celim brojevima, naše funkcije će se završavati sa JP &ABC.

Program na slici 2 će, nadamo se, učiniti da vam dodavanje funkcija bude znatno jasnije. Odlučili smo se za naredbu DISPLAY koja ima jedan argument koji daje poziciju na ekranu. Naredba DISPLAY tada vraća ASCII kod karaktera koji se nalazi na toj poziciji. Ovo, jasno, uvek možemo da zamениmo sa BYTE (&2800+I) ali nam, kao što znamo, nije cilj da po svaku cenu dodajemo korisne naredbe koje bi bile složene. Kada razumete princip, moći ćete da se okušate i na nekoj drugoj funkciji.

Za sada, dakle, znamo da dodajemo samo funkcije čiji su argumenti i rezultati celi brojevi. Ovakve funkcije mogu da budu korisne u igrama i sličnim primenama, ali su matematički nastrojani čitaoci ovih redova svakako poželeti da svoj računar dopune nekom trigonometrijskom ili eksponencijalnom funkcijom koja im mnogo nedostaje. To nije jednostavno kako se u prvi mah čini, ali verujemo da će ovakve funkcije veoma brzo opravdati trud uloženi u razumevanje principa njihovog dodavanja. Taj trud će najvećim delom biti utrošen na razumevanje tzv. obrnute poljske notacije.

## Obrnuta poljska notacija

Obrnuta poljska notacija, Lukasijeviĉeva notacija ili, jednostavno, RPN je sistem notiranja koji na prvi pogled izgleda sasvim neprirodno (odatle valjda i ono „obrnuta“ u nazivu), ali koji je sa aspekta raĉunara i posebno mašinskih jezika nešto sasvim normalno i, skoro bi se moglo reći, jedino prihvatljivo. Oni koji su posedovali ili poseduju neki Hewlett-Packardov džepni raĉunar (bio on programabilan ili ne) svakako su upoznati sa principima RPN-a pa mogu slobodno da predu na ĉitanje sledećeg poglavlja. Preostalima savetujemo da poseite nekog prijatelja koji ima ovakav raĉunar i u praksi provere koliko će im ovo poglavlje biti jasno. Proba sa džepnim kalkulatorom je daleko manje naporna od pokušaja da se na mašinskom jeziku (pogotovu ako ne posedujemo asembler) nauĉi RPN.

Za RPN notaciju je stek neobiĉno vaŹan pojam. Stek možete da zamislite kao orman sa mnogo police koje se nalaze jedna iznad druge. Na svakoj polici može da se nađe samo jedan predmet (u našem sluĉaju samo jedan broj). Zato kada god Źelimo da stavimo broj na donju policu broj koji se u njoj nalazio mora da bude prenesen na prethodnu, broj iz prethodne na policu iznad nje i tako dalje. U teoriji bi stek morao da bude beskonaĉan, što znaĉi da se dodavanjem broja ništa ne gubi iz njega. No raĉunari teško mogu da posluŹu sa beskonaĉnom memorijom koja bi morala da bude upotrebljena da se saĉuvava beskonaĉni stek, pa naš orman ima gornju policu. Kada stavimo neki broj na stek, broj iz prethodne police će preći u poslednju, dok će broj iz poslednje police biti za sva vremena izgubljen.

NajniŹe dve police u steku su posebno privilegovane jer se u njih smeštaju argumenti aritmetiĉkih operacija. Ako, na primer, treba da saberemo brojeve 2 i 3, najpre ĉemo na stek staviti broj 2 a zatim i broj 3 (time će, naravno, broj 2 biti pomaknut u sledeću policu), a zatim pozvati program za sabiranje. Da li će on izraĉunati zbir i staviti ga na stek? Stvar, na Źalost, nije tako jednostavna: po sabiranju nam je potreban zbir, a ne i argumenti koji bi na ovaj naĉin ostali zauvek na steku. Zato će program za sabiranje najpre spustiti stek za jedno mesto (u gornju policu upisati nulu, sadržaj gornje police preneti na policu ispod nje itd. sve dok sadržaj druge police ne pomeri u prvu a sadržaj prve ne skloni sa steka), sabrati argumente koji su se ranije nalazili na najniŹe dve police i smestiti zbir na najniŹu policu „zaboravljajući“ njen prethodni sadržaj. Ako su se, dakle, na steku nalazili brojevi 2, 3, 4 i 7 (tim redom; 2 je na najniŹjoj polici), posle sabiranja će na steku biti brojevi 5, 4, 7 i 0.

O mnoŹenju teško vredi reći išta novo, ali oduŹinjavanje i deljenje zahtevaju nekoliko reĉi objašnjenja. Kod ove dve operacije, naime, nije svejedno da li će se raĉunati X-Y (X/Y) ili Y-X (Y/X). Zato je u RPN-u uvedena konvencija koja kaŹe da je broj koji se nalazi na najniŹjoj polici steka uvek umanjilac (delilac), a da je broj iznad njega uvek umnoŹitelj (deljenik). Ako se, kao u malopredređenom primeru, na steku nalaze brojevi 2, 3, 4 i 7 posle izvršavanja oduŹinjanja će on dobiti oblik 1, 4, 7 i 0.

&2C3A: 3E C3 32 AC 2B 21 46 2C  
&2C42: 22 AD 2B C9 F5 21 00 10  
&2C4A: 2B 7C FE 00 20 FA F1 C9

10 !  
20 ! ZAMENA READY  
30 !  
40 ! ILLUSTRACIJA LINKA ZA  
50 ! VIDEO  
60 !  
70 !  
80 <  
90 ORG &2C3A  
100 OPT 3  
110 LINK EQU &2BAC  
120 LD A,&C3 ! &C3 = JP  
130 LD (LINK),A  
140 LD HL,PROG  
150 LD (LINK+1),HL  
160 RET  
170 PROG  
180 PUSH AF  
190 CP 64  
200 JR Z,MIPRO  
210 LD A,(RADNI)  
220 CP 0  
230 JR Z,NOPRINT  
240 DEC A  
250 LD (RADNI),A  
260 BACK  
270 POP AF  
280 POP HL  
290 POP HL  
300 RET  
310 NOPRINT  
320 POP AF  
330 RET  
340 MIPRO  
350 LD A,6  
360 LD (RADNI),A  
370 PUSH DE  
380 LD HL,PORUKA  
390 LD DE,(&2A6B)  
400 LD BC,11  
410 LDIR  
420 LD (&2A6B),DE  
430 POP DE  
440 JR BACK  
450 PORUKA  
460 BYTE 64  
470 BYTE 39  
480 TEXT "GALAKSIJA"  
490 RADNI WORD 0000  
500 >  
510 A=USR(&2C3A)

Osim ĉetiri osnovne raĉunske operacije, potrebne nam je i poredenje brojeva. Poredenje se, naravno, brojevi na najniŹe dve police steka, ali mogu da postoje dva pristupa ovakvoj operaciji. Po jednoj poredenju brojeva ne remeti stek ni u kom smislu (ovako nešto vaŹi kod Hewlett-Packardovih programabilnih kalkulatora kao što je HP41C), a po drugoj poredeni brojevi nestaju sa steka dok se ĉitav stek spušta za dva mesta uz popunjavanje poslednje dve police nulama (ovakav pristup je zastupljen na raĉunaru „galaksija“).

## Broj u kondenzovanom formatu

Do sada smo smatrali da se na svakoj polici steka u jednom trenutku može naći samo jedan broj, što nam je u mnogome olakšalo ĉitavo rezonovanje. Svaki vlasnik „galaksije“, medutim, zna da svaki broj u pokretnom zarezu zauzima ĉetiri bajta, a zabuna će biti kompletna kada kaŹemo da se svaki broj na steku sastoji od pet bajtova!

Da bismo razumeli ĉitavu zbrku, moramo da upoznamo organizaciju „galaksijinog“ pamćenja brojeva. Svaki broj u pokretnom zarezu zauzima ĉetiri bajta (32 bita) memorije nalazeći se u takozvanom „kondenzovanom formatu“. Kondenzovani format, medutim, nije pogodan za raĉunavanje, jer je eksponent broja pomešan sa njegovom mantisom. Zato je pred svake aritmetiĉke operacije neophodno broj iz kondenzovanog prevesti u normalizovani format. Takav broj će zauzimati 40 bitova od kojih će osam biti neiskorišćeno. Ako po

&2C3A: 3E C3 32 AC 2B 21 46 2C  
&2C42: 22 AD 2B C9 F5 FE 40 2B  
&2C4A: 11 3A 00 2C FE 00 2B 0B  
&2C52: 3 32 80 2C F1 E1 C9  
&2C5A: F1 C9 3E 06 32 80 2C 05  
&2C62: 21 75 2C ED 5B 68 2A 01  
&2C6A: 0B 00 ED 5B 68 2A  
&2C72: D1 1B E1 40 27 47 41 4C  
&2C7A: 41 4B 53 49 4A 41 00 00

završetku rada treba „zapamtiti“ neki normalizovani broj, pretvorimo ga ponovo u kondenzovani format i smestimo u memoriju. Ćemu ova komplikacija kada su svi brojevi mogli da se ĉuvaju u normalizovanom formatu? Jednostavno zbog uštede skuĉenog memorijskog prostora.

Za programera bi bilo previše komplikovano da stalno misli na to koliko bajtova zauzima koji broj na steku, koliko je ovaj potpun, Źta pokazuje registar i sliĉno. Zato su obezbeđeni sistemski programi koji omogućavaju da stek zamislimo kao orman sa policama opisan u prethodnom poglavlju. Ukoliko se sluŹite njima, ĉinice vam se

da se na svakoj polici nalazi jedan broj, nećete misliti na to koliko on ima bajtova, u kom je formatu i tome slično. Zato ćemo sledeće redove posvetiti sistemskim potprogramima za operacije sa stekom.

Najčešće će procesorski registar HL pokazivati na neku memorijsku lokaciju počevši od koje je smeštena neka konstanta koja zauzima četiri bajta. CALL &A45 će ovu konstantu prevesti u normalizovani format i smestiti je na aritmetički stek uz njegovo obavezno pomeranje nagore. Inverznu funkciju (smeštanje broja sa dna steka u četiri bajta memorije na koje pokazuje HL) obavlja CALL &A6D.

Kako se, međutim, konstante uopšte stavljaju u memoriju? Ako se radi o celobrojnoj konstanti, možemo da je smestimo u HL registar i izvršimo CALL &ABC. Ovak poziv će sadržaj registra HL pretvoriti u broj u normalizovanom obliku i smestiti ga na aritmetički stek uz uobičajeno pomeranje. Inverznu funkciju (smeštanje celog dela broja sa dna aritmetičkog steka u HL uz spuštanje steka) obavlja CALL &A6D.

Kako, međutim, da smestimo konstantu koja nije cela tako da je donjije prenesemo na stek naredbom CALL &A45? Mi namerno nismo trošili prostor na objašnjenje smeštanja brojeva u kondenzovanom formatu (možete li da dedukujete način na koji „galaksija“ smešta brojeve? — nagradićemo najbolje obrazložene odgovore) jer ovo prevodenje može za nas da obavi operativni sistem primenom jednog lepog trika. Potrebno nam je, na primer, da ustanovimo kako se u memoriju smešta broj 1 (on je ceo, ali na sasvim isti način možemo da postupimo i ako nije). Tada ćemo najpre izvršiti A=1, a zatim: FOR I=0 TO 3: PRINT BYTE (PTR+I): NEXT I. Tako ćemo, bajt po bajt, videti da se broj 1 smešta u memoriju kao (posle prevodenja u svak pogodnije heksadekadne brojeve): &0, &0, &80 i &0.

Krajnje je vreme da damo i adrese sistemskih rutina koja obavljaju osnovne operacije. CALL &B32 sabira, CALL &B1E oduzima, CALL &AE6 množi, CALL &AF7 deli, a CALL &B10 poredi brojeve (rezultat poredenja biva smešten u Z i C flagove na uobičajeni način). Da ne ostanemo dužni onima koji žele da pišu programe za igre, sistemski program koji na stek stavlja slučajni broj između 0 i 1 može da se pozove sa CALL &C9F. Ukoliko nam je, najzad, potrebno da izračunamo vrednost čitavog izraza i smestimo je na stek, postavićemo DE registar tako da dokazuje na početak izraza i izvršiti CALL &AB2. Ukoliko je taj izraz u zagradi (npr. ako je argument neke funkcije), upotrebićemo CALL &781.

Ukoliko, najzad, želimo samo da utičemo na znak broja, setovaćemo ih resetovati u 1 i 0 u četiri na koju pokazuje IX-1 u zavisnosti od toga da li je broj negativan ili pozitivan.

Kao ilustraciju svega ovoga i još jedno podsećanje na mogućnosti dodavanja funkcija dajemo program sa slike 3. On dodaje računaru funkciju REC koju prati argument u zagradi. Funkcija izračunava, kao što joj ime i govori, recipročnu vrednost ovog argumenta, što i nije nešto naročito korisno ali zato odlično ilustruje rad sa aritmetičkim stekom. Namerno se

```
10 <
20 ! PERMANENTNI SAT
30 !
40 ! ILLUSTRACIJA LINKA U
50 ! INTERAPTU
60 !
70 <
80 ORS &2C3A
90 OPT 3
100 DFIL EQU &2B00
110 INTER EQU &FD
120 Y EQU &2A80
130 HALT
140 LD IV,ENTRY
150 RET
160 ENTRY
170 LD A,(&2B00)
180 CP 0
190 JR NZ,BACK
200 LD A,(Y+2)
210 CP " "
220 JR NZ,BACK
230 LD A,(Y+5)
240 CP " "
250 BACK
260 JP NZ,INTER
270 LD DE,DFILE+20
280 LD HL,Y
290 DISP
300 LD A,(HL)
310 AND Z
320 JP Z,INTER
330 LD (DE),A
340 INC HL
350 INC DE
360 JR DISP
370 >
380 A=USR(&2C3A)
```

```
&2C3A: 76 FD 21 40 2C C9 3A B0
&2C42: 2B FE 20 00 2C 3A 82 2A
&2C44: FE 3A 20 05 3A 85 2A FE
&2C52: 2E C2 FD 00 11 14 2B 21
&2C54: 80 2A 7E A7 CA FD 00 12
&2C62: 23 13 18 F6
```

nismo poslužili sistemskim programom &ABC koji je mogao da prenese jedinicu iz HL na stek — smeštanje četvorbajne konstante u memoriju i njeno prenošenje na stek pomoću CALL &A45 je mnogo instruktivnije.

## Link za video

Pogled na mapu sistemskih promenljivih pokazuje da „galaksija“, osim linka za naredbe, ima i link za video na adresama &2BAC, &2BAD i &2BAE. Pre nego što će na ekranu ispisati bilo koje slovo (podrazumeva se da to slovo treba da bude ispisano sistemskim pozivom RST &20 a ne direktno smešteno na ekran pomoću LD ili BYTE) računaru izvršava CALL &2BAC. Ukoliko po pozivu naide na RET, računara će se vratiti u ROM i ispisati karakter. Ukoliko, međutim, na ovim adresama nađe neki JP, naš program će preuzeti kontrolu i karakter će biti ispisan tek posle naše obrade, kada računara nađete na kraju našeg programa u RAM-u.

Link za video je, u stvari, obezbeđen da bi uopšte bilo moguće podržati rad sa štampačem. Ukoliko je ovaj priključen, trebalo bi sastaviti program koji, kada je neki flag setovan, prenosi svako slovo koje treba

da bude ispisano na ekranu na papir posredstvom porta i štampača koji je sa njim povezan. Videćemo, međutim, da se sa ovim linkom mogu poslati i neke druge stvari koje će interesovati i one koji ne poseduju štampač. Počevšom, naravno, od najjednostavnije i ne naročito korisne primene koja će nam pomoći da shvatimo drugi primer programa.

Da li vam se nekada učinilo da vaša „galaksija“ radi prebrzo? Naravno da nije (to se ne dešava čak ni vlasnicima daleko brzih kompjutera), ali ćemo vam prikazati program koji će usporiti ispisivanje poruka. Ideja je u tome da nije ni malo teško sastaviti bežik program koji će na ekranu usporeno ispisivati neki tekst da bi se, na primer, računala neki EPF poruka. Ipak, retki su računari kod kojih se može intervenisati tako da se uspori ispisivanje poruka bežik interpretatora. Pokažite program sa slike 4 nekom vlasniku ZX81 ili Spectruma — računara koji ne omogućavaju nikakav izlazak iz ROM-a!

Kako radi ovaj program? Ništa lakše — pre ispisivanja svakog karaktera „galaksija“ izvesno vreme izvršava „mrtvu petlju“ koju može da skraćujete ili produžavate menjanjem broja, koji se smešta u HL registar na početku njenog radnog dela. Po završetku „mrtve petlje“ jedno RET vraća kontrolu nad računarem bežik interpretatoru u ROM-u. Primetite da je bilo neophodno sačuvati registre AF.

Program sa slike 5 predstavlja daleko složeniju primenu linka za video i znatno veće posezanje u rad operativnog sistema. Umesto READY, računara će ispisivati GALAKSIJA sve dok je ovaj program u memoriji. Kako smo to postigli? Pre svakog READY „galaksija“ ispisuje znak koji se sastoji od dva karaktera čiji su kodovi 64 i 39 (označeni u tabeli na strani 18 kao spec). Mašinski program najpre ispituje da li se u registru A nalazi kôd 64 koji označava da upravo treba da bude štampana leva polovina znaka. Ukoliko nije stigao kôd 64, izvršava se obično RET i dozvoljava ispisivanje karaktera. Ukoliko, međutim, program prepozna karakter čiji je kôd 64, biva proglašeno „vanredno stanje“: u jednu memorijsku lokaciju se upisuje podatak da sledećih 6 bajtova ne treba štampati a zatim se, na sasvim standardan način, na ekranu ispisuje tekst GALAKSIJA.

Najslabije je bilo „zabrani“ računaru da štampa karaktere. Da bi ovo bilo moguće, proučili smo rutinu za štampanje karaktera u ROM-u i utvrdili da „galaksija“, posle RST &20, poziva dva nivoa potprograma od kojih je drugi link. Zato naša rutina skida sa steka prva četiri bajta koje predstavlja dve adrese za povratka sa potprograma (trebalo bi da svi znamo da adresa za povratka sa potprograma ima 2 bajta, a da isto toliko sa steka „sklanjanje“ jedno POP HL). Računar, dakle, na završetku našeg programa nailazi na RTN, ali sa ono ne vraća na mesto na kome se nalazio CALL koji je pozvao link već u rutinu koja je pozvala RST &20; karakter, tako, nije ni štampan što smo i želeli.

Link za video može da se koristi i za mnogo što-šta drugo (sačekajte samo malo pa ćete naći na program koji se automatski startuje) pa i za jednostavno intervenisanje u video igrama. Ima, na primer, igara kod kojih se igračev „broj života“ ispisuje samo na ekranu, pa bi u principu bilo moguće beskonačno produžavati igru zabranjujući brisanje karaktera koji ga predstavlja. Čini nam se, ipak, da je ovakva primena

sličnih linkova daleko zahvalnija kod drugih računara, kod kojih je struktura ekrana komplikovanija (link za video kod Spectrum-a bi mogao divno da se koristi da postoji). Autori igara za „galaksiju“ će, svakako, smatrati da je lakše prosto „poukovati“ nešto u ekran nego koristiti sistemski i program koji to radi!

## Link u interaptu

Na početku poglavlja koji je posvećeno računaru „galaksija“ rekli smo da nikako ne treba koristiti registar IY, jer je on korišćen kao link u interaptu. Pošto smo do sada, nadamo se, stekli sva potrebna znanja o pojmu linka, posvetimo se podrobnije ulozi koja je dodeljena registru IY. Moramo, međutim, da upozorimo da su linkovi koje smo pominjali opasni prema redosledu izlaganja, što znači da je najopasniji bavi se ovim poslednjim: pre startovanja svakog programa treba „sors fajl“ (izvorni program) snimiti na kasetu. Kada pogrešno završen interapt „uğuši“ računar, RESET najverovatnije neće pomagati — biće potrebno da isključite i ponovo ga uključite.

Pošto je pomogao video stepenu da generiše sliku i povećaj brojač sekundi u časovniku (ako je ovaj uključjen, naravno), mikroprocesor je pri kraju interapta — treba još samo uzeti sistemske registre sa mesta i izvršiti RETI. Pre nego što to uradi, međutim, mikroprocesor nailazi na naredbu JP (IY). U registru IY je u toku inicijalizacija računara smešten broj &00FD, pa mikroprocesor izvršava naredbe koje se tamo nalaze. To su, naravno, četiri POP naredbe i RETI.

Mi, međutim, možemo da promenimo sadržaj registra IY tako da on pokazuje neku lokaciju u RAM-u. Tamo treba da se nalazi program koji će se izvršavati svake pedesetine sekunde i koji će se završavati sa JP &FD. Pri izradi takvog programa, međutim, treba da budemo neobično pažljivi; on, pre svega, mora da se izvršava vrlo kratko jer svaki mašinski ciklus njegovog izvršavanja skraćuje iskorišćenje mikroprocesora za izvršavanje redovnog programa. Osim toga, za vreme interapta ne sme nikako da se desi da mikroprocesor dobije novi interapt, ako ništa drugo zbog toga što bi se stek tako stalno punio. Da bi to bilo sprečeno na samom početku interapta, mikroprocesor automatski izvršava DI. Ukoliko naš program koji se izvršava u interaptu, samostalno ili nehotičnim pozivom neke rutine iz ROM-a, izvrši EI, vrlo je verovatno da će nastupiti veliki problemi. Ako se ovo ne desi, ali se interapt ne omogući u razumnom vremenu, slika se više neće videti i jedini spas će biti isključenje računara.

Osim kratkoće, savetovali bismo vam da u okviru programa koji se izvršava u interaptu ne pozivate potprograme iz ROM-a, jer većina njih u nekim slučajevima može da poremeti normalan tok stvari (odlaskom na rutinu za grešku, na primer). Ako, dakle, želite da testirate neke tastere u interaptu, moraćete da testirate čelije &2000-&2037, a ne da koristite CALL &CF5.

Na slici 6 dajemo primer korišćenja linka u interaptu. Radi se o rutini koja se nekada nalazila u ROM-u, ali je morala da bude

```

10 <
20 ORG &3F00
30 OPT 3
40 LINK EQU &2BA9
50 Z&B3 EQU &75B
60 X343 EQU &39A
70 DFILE EQU &2B00
80 NUHVAR EQU &2A03
90 BYTEA EQU &E6A8
100 WORDHL EQU &E6A3
110 ZAREZ EQU 5
120 LD A,&C3 ! &C3 = JUMP
130 LD (LINK),A
140 LD HL,PROG
150 LD (LINK+1),HL
160 SET
170 PROG
180 EC (SP),HL
190 PUSH DE
200 LD DE,Z&B3
210 RST &10
220 POP DE
230 JR Z,NAREDBA
240 EC (SP),HL
250 RET
260 NAREDBA
270 LD HL,TABLICA-1
280 JP X343
290 TABLICA
300 TEXT "RSAVE"
310 BYTE RSBUB>+&B0
320 BYTE RSBUB<+&FF
330 BYTE BACK>+&B0
340 BYTE BACK<+&FF
350 BACK
360 RET
370 RSBUB
380 POP AF
390 LD HL,&66
400 PUSH HL
410 RST &8
420 PUSH HL
430 CALL ZAREZ
440 INC HL
450 CALL PUSH HL
460 CALL ZAREZ
470 LD B,H
480 LD C,L
490 POP HL
500 POP DE
510 PUSH BC
520 LD B,&66
530 DI
540 LEADER
550 XOR A
560 CALL BYTEA
570 DJNZ LEADER
580 LD A,&AS
590 CALL BYTEA
600 POP BC
610 PUSH HL
620 PUSH BC
630 LD H,D
640 LD L,E
650 ADD HL,BC
660 LD B,&AS
670 CALL WORDHL
680 LD A,B
690 POP BC
700 POP HL
710 PUSH HL
720 ADD HL,BC
730 LD B,A
740 CALL WORDHL
750 POP HL
760 JP &E57
770 >
780 A=USER(&3F00)

```

```

&3F00: 3E C3 32 A9 2B 21 0C 3F
&3F08: 22 AA 2B C9 E3 05 11 5B
&3F10: 07 07 D1 2B 02 E3 C9 21
&3F18: 1C 3F C3 9A 83 52 53 41
&3F20: 56 45 BF 27 BF 26 C9 F1
&3F28: 21 66 00 E5 CF E5 03 05
&3F30: 00 23 E5 C0 05 00 44 4D
&3F38: E1 D1 C5 06 60 F3 AF CD
&3F40: 68 0E 10 FA 3E A5 C0 69
&3F48: 0E C1 E5 C5 62 6B 09 0E
&3F58: A5 CD 63 0E 7B C1 E1 E5
&3F68: 09 47 CD 63 0E E1 C3 57
&3F70: 0E

```

žrtvovana zbog nekih rešenja rada sa alfanumericima: neprekidnom prikazivanju vremena. Kada otkucamo i izvršimo ovu rutinu, neće se, bar u prvi mah, ništa dogoditi. Da bismo videli njeno dejstvo, otkučaćemo YS=12:10:15:10"DOT" (zapazite da su mi nulti i sekundni odvojeni tačkom, a ne uobičajenim dvema tačkama). Od toga momenta u desnom gornjem uglu ekrana videćemo tekuće vreme. Bez obzira da li listamo program ili brišemo ekran, vreme će biti prikazivano sve dok, na primer, ne otkučamo YS="12:10:15:10".

Kako je ovaj efekat posignut? Prilično jednostavno: svake pedesetine sekunde računar prenosi sadržaj promenljive YS na ekran ukoliko su ispunjena dva uslova. Prvi je da se u YS uopšte nalazi vreme (treći simbol treba da budu dve tačke), a drugi da je peti simbol stringa YS tačka. U ovom programu se javlja i naredba HALT koju obično prenebregavamo u školama programiranja na mašinu, pa smo odlučili da joj ovde posvetimo par reči.

Naredba HALT izaziva prekid rada mikroprocesora, što obično izaziva pomisao da je ona destruktivna. Mikroprocesor, međutim, neće zaugdo stajati: čim naiđe prvi interapt, njegova aktivnost će se nastaviti. To znači da se HALT može prevesti kao „sačekaj prvi interapt, izvrši ga, pa nastavi sa radom“. Kao takva, HALT je neobično korisna naredba: ukoliko želimo da radimo nešto sa IY registrom ili, čak, da prenosimo veće blokove ekrana, vrlo je korisno upotrebiti naredbu HALT kao neku vrstu sigurnosnog ventila. Ukoliko to ne učinimo, moguće je da program počne da greši s vremenom na vreme u naoko besmislenim trenucima, a iskustvo nam govori da se takve greške najteže otkrivaju.

Šta bi se desilo da je, u momentu kada je mikroprocesor naišao na HALT interapt bio onemogućen (DI)? U tom slučaju bi HALT zaista blokirao računar, ali bi pritisak na RESET povratio sve u normalno stanje bez štetne po sadržaj RAM-a. Dolazak na „farmu“ bi, ujedno, i omogućio interapt, pa bi se pojavila i slika koje do toga momenta nije bilo.

Obzirom da govorimo o interaptu, neophodno je da sa par reči obradimo NMI — nemaskirani interapt. On nastupa kada korisnik pritisne dirku RESET koja se nalazi sa zadnje strane „galaksije“. Nemaskirani interapt se razlikuje od maskiranog u tome što računar, kada ga dobije, u istom trenutku prestaje sa izvršavanjem naredbe koja je trajala i prelazi na program koji se nalazi u ROM-u počevši od &66. Tamo najpre biva reinicijalizovan stek pointer, a zatim ispisano READY. Posle nemaskiranog interapta, međutim, „galaksija“ neće reinicijalizovati sistemske promenljive, linkove i sadržaj IY registra, što za praktičnu posledicu ima činjenicu da računar može ali i ne mora da se vrati u radno stanje — ako je, na primer, interapt pogrešno organizovan, slike neće biti ni posle njega.

## Kako se „pakuje“ bejzik

Došlo je vreme da se posvetimo peterijskoj jedinici koju svi vlasnici „galaksije“ redovno koriste — kasetofonu. Pre nego što objasnimo rad sa njom i navedemo adrese dva sistemska potprograma koja će nam omogućiti daleko više od naredbi OLD i SAVE, posvetimo pažnju načinu na koji „galaksija“ pakuje bejzik program u svome RAM-u.

Kod „galaksije“ nije zastupljeno takovano tokenizovanje naredbi, što ima dobro

loše strane. Pošto naredba nije pretvorena u poseban kôd (kod Spectruma se, na primer, PRINT pretvara u &F5), lako je i jednostavno analizirati program bajt po bajt, ali se zato njegovo izvršavanje unekoliko usporava. Program je, dakle, u memoriji smešten onakav kakav jeste, slovo po slovo, uz dva izuzetka.

Linijijski broj se kodira kao dva bajta i to u obliku koji proizvodi naredba WORD. Ukoliko je, na primer, prva naredba bajzik programa 1500 GOTO 1550, PRINT WORD (&2C3A) će dati rezultat 1500. Blanko koji računar prikazuje prilikom listanja programa između linijskog broja i prvog slova naredbe nije, u stvari, upisan u RAM-u (tako se šteti memoriji), dok se ostali blankovi u okviru linije kodiraju kao i svaki drugi ASCII karakter.

Posle poslednjeg znaka neke programske linije u memoriji je obavezno smešten bajt &D (13 dekadno), koji predstavlja terminator linije (znak da je ona završena) i pokazuje računaru da su sledeća dva bajta broj sledeće linije. Ukoliko se radi o poslednjoj liniji bajzik programa, iza njenog &D se ne stavlja ništa posebno (kod drugih računara neki bajzici označavaju jedna do tri nule ili neki drugi specijalni kodovi). Kod „galaksija“ bi ovaj terminator bio sasvim nepotreban; sistematska promena linija čija je adresa &2C38 u svakom trenutku pokazuje gde se nalazi prvo slobodan bajt iza bajzik programa. Ukoliko smo, na primer, u memoriju stavili samo bajzik naredbu 10 PRINT A, memorijske ćelije počevši od &2C3A imaju redom sadržaje 0,10,80,82, 73, 78, 84, 32, 65, 13, a promjenljiva &2C38 ima vrednost &2C44.

Dobar primer svega ovoga je program „Telefonski imenik“ koji je objavljen u „Galaksiji“ broj 143. Ovaj program modifikuje samoga sebe dodajući nove naredbe i kodirajući ih na odgovarajući način. On, naravno, stalno vodi računa o sistemskoj promenljivoj &2C38 koju povećava posle dodavanja svakog bajta. Pri njegovoj analizi treba jedino imati u vidu da bajzik program ne počinje od &2C3A (tu je smešten mašinski potprogram), već da je sa NEW 60 početak bajzika pomenar za 60 bajtova. Računar to registruje promenom sistemske promenljive &2C36.

## Snimanje na kasetu

Dolazimo, najzad, do snimanja programa i podataka na kasetu. Da bi računar mogao korektno da učita program, moraju da mu budu saopštene informacije o njegovoj dužini i (kada nije reč o bajzik programu) adresi od koje ga treba upisivati. Zato se ispred svakog programa nalazi takozvani „heder“ (zaglavak) u okviru koga su upisane ove informacije. Pre hedera na traku biva snimljen takozvani „lider“ od 96x8 nula bitova koji nemaju nikakvu posebnu ulogu — oni jednostavno saopštavaju računaru da sledi program i omogućavaju da se kasetofoni koji imaju automatsku kontrolu nivoa reprodukcije prilagode specijalnim tonovima. Računar pri učitavanju ne proverava da li je učitano baš 768 „praznih“ bitova pa učitavanje programa ne morate da počnete sa trakom premlanom tačno na mesto na kome se nalazi početak „lidera“, tolerancija je, naznao, nije velika.

```
&3BAC: 00 00 21 FF 29 36 20 2B
&3BB4: 0B 5C 20 F9 21 CD 2B 11
&3BBC: CB 5C 20 F9 21 CD 2B 11
&3BC4: E7 2B 01 11 00 ED B0 1B
&3BCD: FE 41 55 54 4F 53 54 41
&3BD4: 5E 54 20 50 52 4F 47 52
&3BDC: 41 4D
```

Posle „lidera“ nailazi sinhronizacioni bajt &A5 koji se takođe ne unosi u memoriju; on, jednostavno, govori računaru da je došao kraj „lidera“, i da sledi „heder“. Heder ima samo četiri bajta: prva dva označavaju adresu od koje program treba da bude smešten u memoriju, a druga dva adresu prvog bajta koji će biti slobodan posle unošenja programa. Sledi sam program, bajt po bajt. Na kraju snimka nailazi takozvani „čeksum“: jedan bajt koji omogućava računaru da sa velikom sigurnošću (ova sigurnost ni kod jednog kompjutera ne može da bude stoprocenatna) utvrdi da li je program ispravno upisan u memoriju ili je u toku učitavanja neki bajt ispušten ili deformisan. Šta je, dakle, „čeksum“? U toku upisa programa u memoriju, „galaksija“ sabira pročitane bajtove i to u modulu 256 (100+20=120, 120+111=231, 231+30=50+261=256) i tome slično, a čeksum ovaj zbir popunjava do 00 (u gornjem primeru čeksum bi bio 256 - 5 = 251).

Ukoliko bi se po učitavanju pokazalo da zbir, dopunjen čeksumom, nema vrednost nula, bilo bi sasvim sigurno da je program pogrešno upisan, pa bi računar ispisao WHAT? & zatim i READY. U protivnom, vrlo je verovatno (moguće je, naravno, da je neki bajt povećan za jedan, a neki drugi smanjen za 1, što se u konačnom zbiru ne prepućuje) da je program ispravno prepisan u memoriju, pa će „galaksija“ ispisati samo READY.

Čak i kada je program pogrešno upisan sa kasete, računar „galaksija“, zahvaljujući originalnom metodu kodiranja bitova, olakšava njegovo „vraćanje na normalu“ uz minimalne intervencije.

Na kasetu se upisuju četvrtasti impulsi koji predstavljaju binarne jedinice i nule. Svaki impuls se sastoji od pozitivne i negativne poluperiode od kojih svaka traje po 0,22 ms. Ukoliko se na kasetu upisuje binarna nula sledi pauza duga 2,46 ms, a ukoliko se upisuje jedinica — kratka pauza od 1,1 ms. Iza svakog „dvostrukog“ impulsa koji odgovara jednom bitu slede drugi, treći i ostali bitovi bajta, koji su kodirani na isti način. Po završetku učitavanja bajta sledi dodatna pauza od 5,36 ms koja predstavlja značajno osiguranje: ako se desi da računar, zbog oštećenja trake ili nekog drugog problema ispusti neki bit, pauza od 5,36 ms će se „podsetiti“ da je bajt završen i on će biti spreman da primi sledeći. Tako će biti ispušteno jedno slovo, ali će sva dalja biti ispravna ukoliko nije bilo daljih oštećenja. Računar će, jasno, po učitavanju ispisati WHAT?, pa će obično RUN verovatno naš sintaksnu grešku koju ćemo lako ispraviti. Postoji verovatnoća (mada vrlo mala) da program posle greške u učitavanju bude sintaksno ispravan, ali da nekorektno radi. U tom je slučaju mnogo lakše pokušati novi upis sa kasete nego raziti grešku u imenu promenljive ili konstanti.

Da biste sami upisali bajt na traku treba, najpre, da mašinskim programom isključite interapt (DI), a zatim da pozovete sistemski potprogram čija je ulazna adresa &E68. Na traku će biti poslat bajt iz registra A.

Učitavanje bajta sa kasete se postiže na prilično sličan način — CALL &EDC će navesti računaru da sačekava sledeći bajt („galaksija“ neće imati ništa protiv ako do nalaska tog bajta protokne daleko više od 5,36 ms) i da ga prenese u registar C.

Program sa slike 7 je jedan od primera rada sa kasetofonom: on omogućava relokirano snimanje programa. Rutina sa slike 7 se smešta na sam kraj memorije kako bi bilo moguće upisati programe koji se nalaze na njenom početku. Kada izvršite program, vaš računar će posedovati novo naredbu RSAVE koja ima tri argumenta: prva dva odgovaraju argumentima naredbe SAVE koja omogućava snimanje segmenta memorije, a treći daje adresu počevši od koje će se na kasetu upisani program donjije upisivati. Naredba RSAVE &3000, &3500, &500 će snimiti sadržaj memorijskih ćelija &3000-&3500 na traku, ali će se ovaj program ubuduće učitavati počevši od &3500 (= &3000 + &500) kada upotrebimo obično OLD.

Ovaj program predstavlja dobar uvod za sledeće poglavlje. Želeli smo da navedemo dve primene koje bi trebalo da budu ne samo poučne nego i korisne. Prva od njih će biti automatsko startovanje programa!

## Automatsko startovanje programa

Sinclairovi računari omogućavaju da program koji se učitava sa kasete počne da se izvršava bez ikakvih daljih intervencija korisnika. Obrazložimo potreba za ovim automatskim startovanjem je obično tvrdnja da je za početnika teško da pamti naredbu RUN što i nije neki razlog (ako već mora da pamti naredbu LOAD) — RUN valjda ne predstavlja toliki problem. Da su oni koji propovedaju automatsko startovanje iskreni, rekli bi da se predstavlja dobar način zaštite od neovlašćenog presnimanja programa. „Galaksija“ nikada nije bila pristalica nekih zahteva programa, najpre zbog želje da softver što više kruži među njenim čitaocima. Dakle, razlog zbog koga navodimo informacije vezane za autostart programa nije naša želja da stimulisemo čitaoca da nam šalju zaštićene programe (na kraju poglavlja ćemo reći i kako se ovakve zaštite „probijaju“), već verovanje da je autostart „lepa, čaka“ koja ilustruje mnogo stvari o kojima smo do sada govorili.

Ideja je prilično jednostavna: kada računar „galaksija“, primenom naredbe OLD, učita program, on će pokušati da ispiše READY. Ukoliko je u link za video upis predizdati neki program, kontrola će biti predata tom programu koji će se, dakle, startovati odmah po učitavanju. Ostalo je još pitanje kako da postignemo da se sadržaj bajtova &2BAC-&BAE promene u toku učitavanja programa. Samo na jedan način — naš program treba da počne da se učitava od adrese &2BAC, a njegova ulazna adresa treba da bude na samom početku. Posle ovakve intervencije nećemo, jasno, više moći da se oslanjamo na RST &20, sistemsko skrolovanje slike i slične pomoći operativnog sistema. Zauzvrat, čak ni pritisak na RESET neće prekidati naš program i posle nemaskiranog interapta „galaksija“ će pokušati da ispiše READY, ponovo započeti izvršavanje našeg programa.

Sve ovo ilustruje program sa slike 8 dat kao hex-dump. Unesite ga, u memoriju a zatim (pod uslovom da se u memoriji već nalazi startovan program sa slike 7), otkucajte RSARE &3BAC, &3BDD, — &1000

uključite i uključite računar pa otkucajte OLD i startujete kasetu na koju ste malčas snimili program. Po njegovom učitavanju, na ekranu ćete vidjeti jednu kratku poruku koju će RESET ponoviti.

Zamislite da vam je neko dao program sličan ovome i to na kaseti. Kako da ga presnimite ako ne želite da se upustite u metod koji uvek daje dobre rezultate pre-snimavanje uz pomoć dva kasetofona. Ne brinite, nije mnogo teško. Najpre utvrdite njegovu početnu (MMMM) i krajnju (NNNN) adresu. Otkucajte OLD 200 i učitajte program koji će ovim biti relociran za 200 bajtova. Zatim, koristeći program sa slike 7, otkucajte RSAVE MMMM+200, NNNN+200, —200 i mirno snimite program na novu kasetu.

Može li da se napravi bolja zaštita? Može ako sastavite program koji će biti toliko dugačak da za njegovo relociranje upisivanje nema mesta (npr. program koji bi se proširao od &2BAC do &3FFF) što možete da učinite jedino ako posedujete memorijsko proširenje. Takav program će biti prilično dobro zaštićen od onih koji nemaju toliku memoriju, ali oni sa hardverom poput vašeg i dalje neće imati problema, a snači će se i jedan deo „lukavijih“ sa RAM-om od 6 Kb: sastavite program za prepisivanje koji će počinjati od &2A00 i smestiti ga u područje numeričkih varijabli koje u toku učitavanja programa nije mnogo bitno.

Ukoliko, najzad, sastavljate program uz pomoć nekog drugog računara sa mnogo memorije ili disk jedinica, nema prepreke da napišete rutinu dugu nekih 54 kilobajta koja će, samom svojom dužinom, biti prilično dobro zaštićena (možete li da stavite program koji bi presnimio i ovakvo nešto? Ne zaboravite da iza svakog bajta na kaseti može da postoji i pauza duž od 5.36 ms), ali takva rutina neće biti baš mnogo komercijalna jer će njeno učitavanje trajati.

## Funkcija TIME

U akcionim igrama na kraju se igrači obično rangiraju po osvojenim poenima koji su, pored ostalih parametara, funkcija utrošenog vremena. Računar „galaksija“, kao što znamo, ima časovnik koji je vrlo pogodan za prikazivanje vremena, ali ne toliko pogodan za merenje broja sekundi koje su protekle između dva događaja. Program sa slike 9 dodaje vašem računaru funkciju TIME koja pretvara vreme iz Y\$ u sekunde. Ovaj program ilustruje skoro sve o čemu smo govorili u poslednjim poglavljima: dodavanje naredbi, prijavljivanje grešaka (ako sadržaj promenljive Y\$ nije vreme, biće prijavljeno HOW?) rad sa brojevima u pokretnom zarezu uključujući i smeštanje konstanti...

## Svi „galaksijini“ bagovi

Redovni čitaoci naših napisa u „Galaksiji“ su se u poslednje vreme upoznali sa pojmom „bag“, ali neće biti loše da, u cilju oštrog kompjuterskog obrazovanja, još jednom kažemo nekoliko reči o njemu.

Bag je greška u operativnom sistemu računara koja se ispoljava u nekim specijal-

nim situacijama kao reakcija na neku nepredviđenu akciju korisnika. Ne treba misliti da je bag trenutak kada računar ispiše SYNTAX ERROR ili WHAT? — tada je sve pod potpunom kontrolom operativnog sistema. Posledice бага su obično daleko teže: gubitak linije, programa ili sadržaja čitave memorije, pogrešan ispis na ekranu ili gubitak njegovog sadržaja, remećenje bitnih sistemskih promenljivih...

Svi računari imaju bagove koji predstavljaju manju ili veću smetnju njihovim korisnicima. Izuzetno, kod nekih računara sa „zaborenim“ operativnim sistemom (bez PEEK i POKE) bagovi mogu da budu korisni, jer otvaraju pristup područjima memorije koja su inače tabu za korisnika. Proučavanjem i sistematizacijom bagova računara HP41C članovi američkog PPC kluba su razvili sintetičko programiranje koje im je omogućilo kompletnu kontrolu ovog čuvenog džepnog računara.

Što se kućnih kompjutera tiče, njihovi bagovi prave same probleme: BBC mikročunar i dalje ima bagova (najozbiljniji od njih praktično onemogućavaju korišćenje zarez-a u naredbi PRINT), premda njegov operativni sistem predstavlja dvanaesti verziju istog programa, Spectrum (pogotovo stariji modeli) ima pregršt bagova od kojih je, na svu sreću, malo „pogubnih“, naredbe za editovanje kod TRS 80 izazivaju bezbrojne probleme kod neiskusnih korisnika koji teže da primene sve što su našli u

uputstvu za upotrebu. Nije ništa bolje ni kod drugih kompjutera.

Iako smo uložili dosta truda da pronađemo i eliminišemo sve „galaksijine“ bagove, nismo se nikada zavaravali nadom da ih neće biti. Teši nas jedino činjenica da su sve „galaksijine“ bagove pronašli oni koji su radili na njegovom razvoju — sudeci po vašim pismima nijedan čitalac „Galaksije“ nije naišao na neki bag svoga računara. Mi ćemo, naravno, navesti pronađene bagove da bi se čitaoci ovih redova čuvali situacija koje mogu da izazovu probleme.

BUG 1: Otkucajte punu programsku liniju (obeležite 10 i slova A sve dok računarev bafer ne bude pun, na primer). Zatim otkucajte EDIT 10, i kada se linija pojavi na ekranu, pritisnite DEL nekoliko puta a zatim i ENTER. Pritisnite LIST — ko je rekao da se u memoriji „galaksije“ nalazi neki program?

BUG 2: Otkucajte liniju koja će glasiti 10 A= zatim tridesetak otvorenih zagrada, pa neki broj, pa isto toliko zatvorenih zagrada, a zatim startuje program. Posledice su iste kao kod baga 1. Napominjemo da će „normalan“ broj od dvadesetak otvorenih zagrada izazvati „normalnu“ reakciju HOW? BUG 3: Ovaj bag je toliko dobro sakriven da još nismo uspešli da ga lociramo. U nekim retnim situacijama (uzroci, da ponovimo, još nisu poznati) naredbe za rad sa numeričkim nizom A() će izazvati poruku SORRY. PRINT MEM daje neki negativan broj. Problem se rešava kucanjem naredbe ARRS(-), posle koje sve postaje normalno.

Svi ovi bagovi bi, jasno, mogli da se isprave odgovarajućim promenama ROM-a, ali ove promene verovatno neće nikada biti izvršene. Odgovor na pitanje zašto? Je sadrži jednostavnost: ako promenimo ROM adrese svih sistemskih potprograma (pa i onih koji su pomenuti u ovom poglavlju) će se promeniti, što znači da programi neće biti kompatibilni sa obe verzije ROM-a. Tako ćemo najverovatnije ostati kod 28 verzije ROM-a koju smo proglasili konačnom (PRINT BYTE (55) će vas uveriti da vaš računar ima 28. verziju ROM-a).

I uputstvo za upotrebu „galaksije“ nije imuno na bagove: u mapi sistemskih promenljivih na strani 26 postoje dve sistemske promenljive na adresi &2BB1, ali će svakome iole pažljivom čitaocu biti jasno da „registar za REPT“ zauzima delju &2BBA. Osim toga, „horizontalna pozicija teksta“ zaista ima inicijalnu vrednost 11, ali dekadno, a ne heksadekadno. Za autora ovoga teksta je najboljnije da kaže da program sa dva strane 7 ima grešku i to ne štamparsku nego logičku: on dobro radi za vrednosti iz primera, ali čim bi se putovanje odložilo preko jednog časa... Obzirom da nam se čini da svaki programer početnik (baš tako) može da napiše ovakav program tako da radi ispravno, ovde ne dajemo novu verziju.

Time smo došli do kraja ovoga prilično opširnog poglavlja koje je posvećeno računaru „galaksija“. Autor ovoga teksta duguje veliku zahvalnost Voji Antonicu, konstrukturu „galaksije“, za pomoć pri izboru tema koje će biti obrađene, saradnju na pripremi primera upotrebe raznih tehnika i, posebno, za sve podatke o ROM-u koje nikom drugi nije ni mogao da zna. Zahvaljujući ponajviše ovim podacima, naša „galaksija“ je, čini nam se, postala jedan od vrlo retnih računara koji, kako rekomo i u naslovu poglavlja, „nema tajni“!

Dejan Ristanović

# ČETRNAEST GODINA ELEKTRONIKE

Zagrebačka Tvornica računskih strojeva jedan je od najstarijavih pionira u primjeni mikroelektroničkih logičkih sklopova u biroelektroničkim i AOP uređajima. Već 1969. god. razvijen je na bazi DTL int. sklopova prvi elektronički stolni kalkulator; razvijeni modeli koristili su feritnu memoriju, ali u proizvodnoj seriji ugrađivani su već, tada još sasvim novi, MOS shift-registri. Razvoj novih tipova uvijek je pratio najnovija kretanja na svjetskom tržištu komponenti i gotovih proizvoda. Među prvim firmama u Evropi i TRS je počeo ispitivati mogućnosti mikroprocesora prve generacije; kao rezultat takvog rada pojavio se i fakturirni stroj TRS-701 (kasnije 711) sa INTEL-ovim 4-bitnim mikroprocesorom. Na području 8-bitnih procesora u ranoj je fazi sagledana važnost „single-chip“ tipova za masovnu primjenu u biro-opremi, pa je jedan niz serijskih proizvoda, gotovo cijeli kalkulatorski program, zasnovan na FAIRCHILD-ovom „F-8“ kojeg su za TRS-ove potrebe u velikim količinama izrađivali u posebnim varijantama 3 svjetska proizvođača (sigurno jedinstven pothvat u Jugoslaviji).

Polovicom sedamdesetih godina nabavljeni su razvojni sistemi za 6800-familiju i za Z80, a rezultat rada s tim, još i danas aktuelnim mikroprocesorima, su računski sistemi TRS-702 i 703, kao i periferni uređaji: matični štampači TRS-825/835 i video-terminali TRS-828/838. Danas mogućnosti 8-bitnih procesora još ni izdaleka nisu iscrpljene i mnogi novi TRS-ovi proizvodi koriste hardware i sistemski software za njih razvijen tokom vremena, ali već je započeo i rad na 16-bitnim sistemima, jer su u okviru srednjoročnih razvojnih planova definirani i proizvodi u kojima će procesna „snaga“ 16-bitnih mikroprocesora doći do punog izražaja. Paralelno s razvojem elektroničkih sklopova odvija se razvoj programske podrške; grupa za podršku u okviru teh. sektora broji dvadesetak ljudi i trajno širi područja primjene naše AOP-opreme.

U vezi s novim malim sistemima TRS-703 i 901 donesena je (većim dijelom već i realizirana) odluka da se izrade vlastiti sistemski programi. Operativni sistemi za „703“ i „901“ su već gotovi, a sada se radi na PL-1 compileru. Na ovu varijantu smo se odlučili ne samo zbog uštede deviza, koje bi morali trošiti za nabavku u svijetu rasprostranjenih OP-sistema CP/M ili MS-DOS i pripadnih compilera; nego i u želji da korisnicima pružimo sistemski software optimiran za naše male sisteme i za profesionalne aplikacije.

U skoroj budućnosti treba pristupiti realizaciji DBM (Data Base Management) software-a, imajući u vidu i neke specifične aplikacije i našu viziju daljnjeg razvoja računarske tehnike u smjeru distribuirane obrade i lokalnim mrežama.

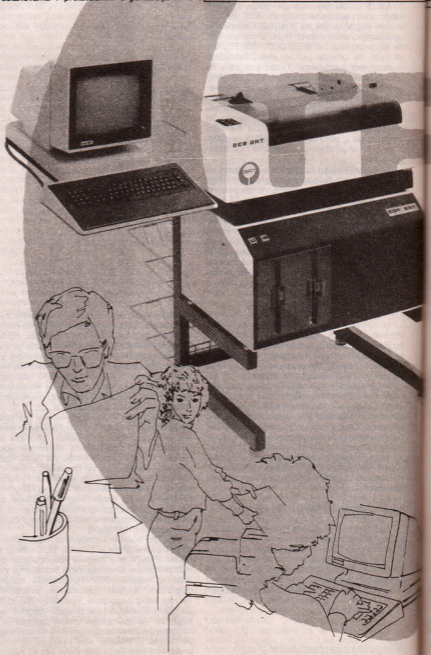
Potpuna modularnost na nivou hardware-a i sistemske i programske podrške omogućava TRS-u da svoju liniju proizvoda prilagodi potrebama naručitaca, a u izgradnji je i takva servisna organizacija koja će na zaista profesionalan način moći održavati isporučene uređaje u cijeloj zemlji. Pored dva glavna pravca; malih računskih sistema i perifernih uređaja, TRS će i nadalje proizvoditi uredsku opremu (kalkulatori), a s mnogo ambicija ulao je i u područje elektromedicinske opreme (za sada proizvodom „TRS-GLUKOTEST“ — uređajem za mjerenje koncentracije lukoze u krvi). S posebnim ponosom u TRS-u se gleda na činjenicu da su svi naši proizvodi rezultat vlastitog razvoja; smišljena i dosljedno provedena kadrovska politika omogućava TRS-u da trajno okuplja mlade stručnjake i da ostane jedan od rasadnika stručnih kadrova u zagrebačkoj regiji. Ironi-

ja je našeg vremena da se dotedavno vlastiti razvoj; manje isplatio od kupnje licenci ili putok sastavljanja uvezanih podsistema. U TRS-u se nadaju da će sadašnji napori za stabilizaciju nase privrvece animirati one proizvođače koji s vlastitim resursima dolaze do novih proizvoda i tehnoloških rješenja.

Na tom putu — uzimajući u obzir i činjenicu da bi autoritija bila društveno neprilvatljiva i štetna — TRS će i dalje surađivati s naučnim ustanovama i proizvodnim organizacijama u

SRH i šire, podupirući napore da se između proizvođača elektroničkih uređaja; a posebno AOP opreme, postignu dogovori o usklađivanju proizvodnih programa i da se što više ovih tehnički i tehnološki vrlo složenih uređaja realizira kao zajednički proizvod. U tome i kroz

## TRS 70





kooperaciju s odabranim stranim partnerima dugoročno vidimo jedinu šansu za plasman na inozemno tržište i zadovoljenje potreba za devizama. Učešće uvoznog repromaterijala u TRS-ovim proizvodima je oko 10% (po vrijednosti), a budući da se opravdano može govoriti o supstituciji uvoza, trebalo bi o toj činjenici voditi računa u gremijima u kojima se raspodjeljuje strana sredstva plaćanja.

Najatraktivniji proizvod Tvornice računskih strojeva svakako je:

### 3 MALI POSLOVNI SISTEM

Preteča je cijene obitelji suvremenih malih kompjutera opće namjene. Hardware je modularan i zasnovan na 8-bitnom mikroprocesoru Z80 i snažnim perifernim upravljačkim jedinicama. U osnovnoj konfiguraciji on poseduje 64 K RAM ima i dvije diskovne jedinice (Floppy-disketa) s kapacitetom od 2,2 MByte. Bitna svojstva tog kompjutera određuje njegova sistematska i aplikativna programska podrška: „703“ razvijen je i vlastiti operacioni sistem (svakako jedinstveni slučaj u Jugoslaviji), a dovršava se i prevodilac (compiler) za PL-1 viši programski jezik. U asoriranu korisničkih programa najvažniji su programi za vođenje materijalnog i finansijskog knjigovodstva, osobnih dohoda i salda-konti, kao i programski paket za organiziranje i praćenje proizvodnje i skladišnog poslovanja. Tu su zatim programi za obradu teksta i za emuliranje nekih IBM-ovih sinhronih terminala. U osnovnoj konfiguraciji mali sistem „703“ obuhvaća: centralnu jedinicu s oba diska, jedan štampač TRS-835 i video-terminal TRS-838. U pripremi je model „901“ sa četiri upisna mjesta i „multi-task/multi-user“ načinom rada.

**TRS 703**  
**TRS 703**  
**TRS 703**  
**TRS 703**  
**TRS 703**  
**TRS 703**  
**TRS 703**  
**TRS 703**  
**TRS 703**  
**TRS 703**  
**TRS 703**

## TRS 835

SERIJSKI MATRIČNI ŠTAMPAČ

Serijski matični štampač TRS-835 je izlazna jedinica za ispis podataka. Predviđen je za rad kao štampač — terminal isključivo u prijemnom modlu, odnosno kao RO štampač (RECEIVE ONLY).

Sve su funkcije štampača mikroprocesorski upravljane i nadzirane radi postizanja i održavanja zadanih tehničkih karakteristika. Kod projektiranja i proizvodnje serijskog matičnog štampača TRS-835 korištena su suvremena teorijska saznanja i najmodernija tehnološka dostignuća, što štampaču osigurava vrhunsku kvalitetu, veliku trajnost, sigurnost i pouzdanost u radu.

#### TEHNIČKI PODACI

Brzina ispisa	180 zn/ssek
Broj znakova u retku	132 znaka
Gustina ispisa	2,54 mm/zn
Ispis u oba smjera s optimiziranim puta	
Matrična ispisa	9 × 9
Veličina znaka	2,54 × 1,97 mm
Inkrementalni vertikalni pomak formulara	0,529 mm (u oba smjera)
Brzina pomaka formulara	101,6 mm/ssek
Gustina ispisa redaka	6 redaka na 25,4 mm ili 6 redaka/inču
Fond znakova	96 zn (JUS ili ASCII)

Vrpca velike trajnosti u kaseti  
 Veličina prihvatne memorije 274 zn (do 4 K opcion.)

Interface prema izvoru podataka:

- CENTRONICS paralelni
- V 24 (RS 232 c) serijski
- DEC LA 180 paralelni

#### REFERENCE:

Štampači TRS-815, TRS-825, TRS-8250 J i TRS-835, proizvedeni u Tvornici računskih strojeva provjereni su i ispitani s uspjehom u radu sa sistemima slijedećih proizvođača: ISKRA DELTA, IBM, OLIVETTI

## TRS 838

ASINHRONI VIDEO TERMINAL

TRS-838 je suvremeno oblikovani univerzalni video-terminal koji se zahvaljujući primjeni modernih mikroprocesorskih komponenti i koncepta po kome se sve funkcije uspostavljaju programski, a ne kruto ožičenom logikom, može primijeniti u svim aplikacijama u kojima se traži univerzalni asinhroni terminal. Pored ostalih interesantnih svojstava posebno treba istaknuti:

- mogućnost prikjučka hard-copy štampača (preko RS 232 C međusklopa)
- mogućnost povećanja video RAM-a do 12 K znakova (8 potpunih stranica) opcija
- mogućnost prikazivanja tzv. „business graphics“ (ne u lokalnom modlu)
- mogućnost emuliranja drugih rasprostranjenih asinhronih terminala (IBM 3101 standardno, DEC VT 100 po posebnoj narudžbi)
- mogućnost izravno povezivanja s host-om udaljenim do 1200 m preko RS 422's međusklopa.

Terminal se sastoji od jedinice s ASCII tastaturom i ekstrakod/logičke jedinice s kodnom cijevi dijagonalne 30 cm (12 inča) sa zelenim fosforom srednje perzistencije. Raspored znakova u tastaturi izmjenjiv je u dogovoru s naručiteljem. TRS-838 s različitim sistemima komunicira preko asinhronog međusklopa po RS 422 ili RS 232 C za izravnu ili udaljenom (remote) vezu, odnosno preko 20 mA strujne petlje za izravnu vezu.

#### Ostala svojstva TRS-838:

- radi u znakovnom ili blok modu
- brzina prijenosa podataka je od 110—9600 Bita/ssek
- korisnik može odabrati lokalni mod rada bez intervencije sistema: to se može koristiti za učenje operatera, kreiranje formata ili editiranje podataka na ekranu, adresiranje cursor-a funkcije brisanja, samostalni pomak na novi red, scrolling-up, prijenos bloka podataka i formatiranje polja
- atributi formiranih polja mogu biti:
  - a) zaštićeno / nezaštićeno
  - b) vidljivo / nevidljivo
  - c) trapnjače / netrapnjače
  - d) pojačani intenzitet / normalni intenzitet
  - e) modifikirano / nemodifikirano
- znakovi su u matrici 7 × 9, 24 redka s po 80 znakova; 25. red je za poruke operateru u radu terminala (sistemski red)
- tastatura je ergonomski optimalno riješena, odvojena od sistema i operater je može prilagoditi svom položaju
- ekran se može zakretati oko horizontalne osi i tako postaviti u najugodniji položaj za operatera
- ESCAPE sekvence za izvršavanje specifičnih funkcija koje se ne mogu aktivirati preko kontrolnih znakova.

Terminal svojim funkcionalnim karakteristikama emulira IBM 3101 (model 2 ×) terminal i može ga u potpunosti zamijeniti.

Terminal je razvijen za rad s IBM S/370 (uključujući 303 ×), 4300, 8100 i S/1 programskom podrškom.

## TVORNICI RAČUNSKIH STROJEVA - ZAGREB

41000 ZAGREB — Braće Kavurića 21 — p. p. 02-846 — Brojlov: TRS-Zagreb  
 Telex: 21434 — Centrala: 447-111, 447-001, 447-592 — Prodaja: 411-302

Sektor podrške: Poslovnica Zagreb, Šerferova 2, 41000 Zagreb, tel. 212-900, 212-911, — Poslovnica Beograd, 27. marta 39, 11000 Beograd, tel. 327-747, — Poslovnica Ljubljana, Linhartova 11a, 61000 Ljubljana, tel. 327-470, — Poslovnica Rijeka, Hotel „Royal“, Opatija, Drazice 3, tel. 712-544

# Komponente za memorijsko proširenje

## Konstruktorski hod po mukama

Za konstruktore i hobiste, čini nam se, nikada nisu bila teža vremena — čipova je, čak i na svetskom tržištu, sve manje, a cene vrtlogu raste. Neslavni rekord u skokovima cena drži Intelov čip 8255 koji je sa nekih tri funte za dva meseca skočio na 19 funti! On, na žalost, igra ključnu ulogu u našem programatoru EPROM-a, a vrlo brzo ćemo mu se, kada budemo radili paralelni interfejs za štampač, vratiti još jednom. Utešno je jedino što cene dinamične memorije stabilne već mesecima i što postoje dobri izgledi da se do kraja godine neće menjati.

Iako u ovim uslovima nije bilo nimalo jednostavno sklopiti aranžman za organizovanu nabavu delova iz inostranstva — u trenutku kada svojim kupcima mere čipove „na gram“, strani trgovci nisu baš previše raspoloženi za naše devizne propise i „štosove“ sa paketicima od 1500 dinara — redakcija je ipak pošlo za rukom da postigne odgovarajući sporazum. Ovog puta smo se upredili za London, iz dva razloga. Prvi je da se memorijski čipovi 4416 trenutno mogu naći jedino u Engleskoj (ne sumnjamo da će ih uskoro nabaviti i trgovci iz Austrije i Italije), a drugi je da je naš dopisnik Anđelko Zgorelec obećao punu pomoć, pa čak i kontrolu isporuka.

Iz organizacije akcije sa računarom „galaksija“ mogli smo puno toga da naučimo, ali, na žalost, malo šta možemo da promenišmo. Redakcija, naime, nema nikakvog uticaja na domaće privredne tokove, a pogotovu ne na kretanja na svetskom tržištu čipova. Zato se naši uslovi, verovatno, neće dopasti konstruktora koji gore od nestrpljenja da povećaju snagu svoje „galaksije“. Ali, ovog puta, ne možemo da garantujemo da se samo cenu nego ni — rok isporuke. Možemo jedino da obećamo komponente — štampana kola i čipove — i to samo za one koji pošalju narudžbenice do 10. avgusta. Tada će isporučeni biti zaključeni i na osnovu njih organizovana proizvodnja štampanih kola, odnosno nabavka čipova. To, praktično, znači da isporuka ne može početi pre kraja septembra, ali i to da će biti okončana u relativno kratkom roku. Naravno, ako nas i ovog puta ne zatraži 4000 narudžbenica, kao za računar „galaksija“. Prilikom isporuke će biti poštovan redosled prijavljivanja — ovgva puta nije bilo prelinarnih narudžbenica, pa ne može doći ni do zabave. Ljubitelji računara imaju gorko iskustvo iz januarske akcije da je jedan dan razlike u stanju narudžbenice značio i do tri meseca čekanja više.

Čitaoci koji su dobro upoznali mehanizme akcije sa računarom „galaksija“ — znamo da su oni, mimo naše volje i uticaja, mnogima dobro „samili“ živce — primetite dve novine: znatno nepovoljnije ali realnije rokove isporuke i nešto drugačiju organizaciju. Isporuka štampanih kola vršice se, najverovatnije, iz redakcije „Galaksije“. Time ćemo postati mnogo bolji uvid u njenu dinamiku a čitaoci će lakše dobiti do potrebnih obavestjenja. Isporuka čipova biće, kao i do sada, vršena iz inostranstva.

Cene koje navodimo za štampana kola i komplete čipova nisu, razume se, konačne. To, međutim, ne bi trebalo previše da brine ljubitelje računara. Tačne cene za štampana kola objavićemo petnaestak dana pre početka isporuke. Oni kojima to bude previše skupe moći će da se predomisle i otkazuju narudžbnu. Tačna cena kompleta čipova biće navedena u predračunju koji će svaki zainteresovani dobiti iz inostranstva. Isporuka čipova biće vršena u skladu sa našim carinskim propisima.

Nijedan čitaop ne može da preuzme na sebe sve konstruktorske probleme svojih čitalaca, a većina, najosobe, nije raspoložena da preuzme nijedan. Provereni projekat, zaista, predstavlja najveću vrednost koja se upotreb može ponuditi čitalcima. U našim uslovima, međutim, važe nešto drugačiji zakoni — provereni projekti ne znače ništa ako se ne obezbede uslovi za njihovu realizaciju. Računar „galaksija“ bi, bez obzira na sve svoje kvalitete, najverovatnije ostao samo „mrtvo slovo na papiru“ da nije organizovan servis za nabavku delova. U njemu nije uvek sve funkcionalno kao treba

— rokovi su se razvlačili kao gladna godina, a cene skakale kada smo se tome najmanje nadali — ali sada ipak ima u pogonu najmanje 1500 „galaksija“ i svakoga dana ih je sve više. Redakcija se još uvek nije umorila balansirajući između vatre svojih čitalaca i vatre svojih saradnika u ovoj akciji. Nadamo se da ne biste ni vi. Strpljenje je, uostalom, majka mudrosti. Ako nam ono ne pomogne, zaista ne znamo šta će.

### Koliko šta košta

1. Štampano kolo za memorijsko proširenje (dvostruko štampa na vitroplastu) sa konektorskom pločicom oko 1300 dinara
2. Štampano kolo za programator EPROM-a (jednostruka štampa na vitroplastu) oko 1000 dinara
3. Komplet čipova za memorijsko proširenje 48 K (čipovi, podnožja, konektor, potširina)
 

48 K	oko 40 funti
32 K	oko 30 funti
16 K	oko 20 funti
4. Komplet delova za programator EPROM-a (integrisana kola, podnožje od 40 nožica i ZIF podnožje od 28 nožica, konektor)
 

oko 35 funti
--------------

Važi do 10. avgusta

#### Narudžbenica (štampana kola)

Molim vas da mi pošaljete pouzdem:  
 1) štampano kolo za memorijsko proširenje (vitroplast, dvostruka štampa) sa konektorskom pločicom po orijentacionoj ceni od 1300 dinara  
 2) štampano kolo za programator EPROM-a (vitroplast, jednostruka štampa) po orijentacionoj ceni od 1000 dinara

Mesto

Ulica i

broj

Ako mi konačna cena ne bude odgovarala, zadržavam pravo da otkážem svoju narudžbnu deset dana pre početka isporuke. Štampana kola ću platiti poštaru prilikom preuzimanja više pošiljki. Narudžbenicu (original ili fotokopiju) dostaviti na adresu: Redakcija „Galaksija“ 11000 Beograd, Bulevar vojvode Mišića 17, ZA ŠTAMPANA KOLA

#### Narudžbenica (komplet integrisanih kola)

Molim vas da mi pošaljete predračun sa uslovima isporuke za sledeći komplet datova:  
 • memorija za „galaksiju“ 16 K  
 • memorija za „galaksiju“ 32 K  
 • memorija za „galaksiju“ 48 K  
 • programator EPROM-a za „galaksiju“  
 • programator EPROM-a za ZX Spectrum

Mesto

Ulica i

broj

Narudžbenicu (original ili fotokopiju) dostaviti na adresu: INTRAPRESS, 16 Garway Road, LONDON W2 4NH, England



Džozef Vajzenbaum:

# MOĆ RAČUNARA LJUDSKI UM

Knjiga koja na izuzetan način osvetljava neke aspekte jednog od najslonjenijih fenomena savremene tehnologije — kompjuterizacije. Pisana popularno i duhovito, ali bez narušavanja tehničke i naučne discipline, poseduje i tu nesvakidašnju vrednost što pruža odgovore na neke često nepoznate probleme, a istraživačima postavlja nezaobilazna pitanja.

Format 14x20. Str. 307. Broširano. Latinka. Cena.....350.-

**RAD — IZDAVAČKA RADNA ORGANIZACIJA**  
 11000 Beograd, Moše Pjase 12. Tel.: 404-765 i 422-517

#### NARUDŽBENICA — RAČ/105

Ovim nepozivno naručujem knjigu: **MOĆ RAČUNARA I LJUDSKI UM**, po ceni od.....350.-  
 Knjigu ću uplatiti pouzdem, prilikom preuzimanja od pošle.

(Poručilac)

(Broj pošle, mesto

i adresa stana)

(Datum)

(Potpis poručioca)

**RAD**

poziv  
na  
pretplatu

# Hemijско tehnološki principi



(u 6 knjiga)

**HEMIJSKO-TEHNOLOŠKI PRIRUČNIK** pokriva specifična područja hemije i hemijske tehnologije. Završavajući koncepciju ovog višestrukog izdanja, sve pojedinačne knjige čine celinu, čime poprimaju trajnu vrednost i postaju stalni saradnik hemičara u njihovom radu.

Prva knjiga  
**HEMIJSKI I FIZICKI  
PODACI I VELIČINE**

Autori: M. Kramanović, M. Obradović, S. Đorđević, G. Dimić, P. Todorović, D. Šepa, P. Bojović, K. Buboški, N. Radosavljević, J. Čirić, S. Mladenović, V. Draško, O. Tatić-Janić, S. Đorđević, V. Valent, S. Šerbanović, Z. Dizdār. **Redaktor: Z. Dizdār.**  
Knjiga sadrži izbor podataka iz više osnovnih oblasti koje se najčešće koriste u laboratorijama istraživačkih i industrijskih organizacija. Daje su numeričke vrednosti veličina koje se odnose na fizičko-hemijske osobine elemenata, jedinjenja i tehnoloških materijala.

- Iz sadržaja:
- Matematika
  - Definicije pojmova i jednačini fizika hemije
  - Međunarodni sistem mernih jedinica i preporuka
  - Fizičke osobine elemenata i jedinjenja
  - Fizičko-hemijske tablice
  - Osobine tehničkih materijala
  - Opihi, trivijalni i komercijalni nazivi materijala i hemikalija
  - Podaci za proračun doza i zaštite od zračenja radioaktivnih izvora

Druge knjiga  
**ANALITIKA**

Autori: J. Čirić, D. Stojanović, V. Vajand, D. Pešić, J. Dobrosavljević, V. Antonijević, S. Kolčki, M. Koblarov, G. Hajduković, Z. Krušević, A. Gogala. **Redaktor: J. Čirić.**  
Ova knjiga je obrađena i sistematizovana tako da daje ne samo teorijske osnove na kojima počiva analitika, otm. analitička hemija nego i postupke u praktičnoj primeni. Obrađene su osnovne klasične i savremene metode organske i neorganske analize. Metodološki je sistematizovan veliki broj analitičkih podataka, koji su neophodni za sveobuhvatno izvođenje pojedinih analiza.

- Iz sadržaja:
- Ravnoteže u rastvorima elektrolita
  - Kvalitativna neorganska analiza
  - Gravimetrija i volumetrija
  - Elektroanalitičke metode
  - Spektrohemijska analiza
  - Radio-hemijska analiza
  - Organska analiza
  - Spektrografska analiza organskih jedinjenja

- Knjige će izlaziti iz štampa sukcesivno, do kraja 1984.
- Mogu se otplatiti u najviše 10 mesečnih rata
- Po izlasku iz štampe cena će biti znatno veća
- Pretplatičnicima koji odloženim uplatom kao iznos preplate odobravamo POPUST 20%

Treća knjiga  
**FIZIKOHEMIJSKE  
METODE**

Autori: S. Đorđević, I. Dorostolovski, D. Pešić, A. Gogala, M. Marinković, M. Pavlović, A. Muk, S. Ribnikar, A. Antić-Jovanović, M. Jeremić, N. Javorčević-Halaj, M. Roguljić, I. Dragičević, Z. Draganić, K. Zrnubov, G. Hajduković, N. Čvrtičević, D. Čvrtičević. **Redaktor: S. Đorđević i D. Pešić.**  
Knjiga tretira fizičko-hemijske metode merenja koje se najčešće susreću u praksi.

- Iz sadržaja:
- Opšta fizičko-hemijska metode
  - Fizičko-hemijske metode
  - Optičke metode
  - Razne metode

Četvrta knjiga  
**ŠTETNE I OPASNE  
MATERIJE**

Autori: O. Stojanović, N. Stojanović, D. Kosanović. **Redaktor: O. Stojanović.**  
Knjiga sadrži tabelarni pregled oko 3.000 raznovrsnih hemikalija sa zapaljivim, eksplozivnim i toksičnim karakteristikama, kao i procenom opšte opasnosti i toksičnog delovanja. Pored ovih karakteristika navedene su i određene neophodne fizičke osobine za isia jedinjenja, kao što su temperatura topljenja i kućavanja, gustina, indeks refrakcije, napon para i gustina para.

- Iz sadržaja:
- Direktno dejstvo štetnih materija
  - Indirektno dejstvo štetnih materija
  - Transport opasnih materija
  - Skladističenje i pakovanje opasnih materija
  - Rad sa opasnim materijama i štetnim
  - Mere zaštite na radu
  - Opasne otpadne materije
  - Uplivost za prvi pomoć
  - Tabelarni pregled karakteristika opasnih materija

**RAD - IZDAVAČKA RADNA ORGANIZACIJA**  
11000 Beograd, Moše Pijade 12,  
tel. 404-785 i 422-517

**NARUČBENICA - RAČUNAR/103**

Ovim se neopozivo pretplaćujem na  
**HEMIJSKO-TEHNOLOŠKI PRIRUČNIK (u 6 knjiga),**  
po pretpl. ceni od ..... 12.600,-  
Uplatiti uz iznosen  
**ODJEDNOM** - Po prijemu „Radova“ uplatnica, sa  
**POPUSTOM od 20% (10.080,- din.),**  
U ..... RATA - Uplatiti broj (najviše deset), po  
prijemu uplatnica od „RADA“  
(Nepotrebno preostati)

U slučaju spora nadležan je Pesti optički sud u Beogradu.

(Poručilac) \_\_\_\_\_

(Broj pošte, mesto) \_\_\_\_\_

(Adresa stana) \_\_\_\_\_

M.P. \_\_\_\_\_

(Datum) \_\_\_\_\_ (Potpis poručilaca) \_\_\_\_\_

Peta knjiga  
**HEMIJSKO  
INŽENJERSTVO**

Autori: G. Popović, B. Đorđević, V. Valent, S. Šerbanović, F. Zdenaki, G. Vunjak-Novaković, S. Končar-Đurđević, Z. Grbačević, A. Tasić, A. Mihajlović, S. Joksimović-Tapacki, S. Pajunović, A. Turović, D. Simonović, Dž. Vasiljević, A. Toić, I. A. Oroz, R. Radosavljević, M. Mitrović, S. Pribitković, S. Cvijović, P. Pujanov, M. Ganova, G. Boloković, A. Deoljter, Lj. Mitasavljević, M. Jevđić, Z. Popović. **Redaktor: S. Končar-Đurđević, D. Simonović, J. Mitić.**  
U ovoj knjizi dati su teorijski osnovi kao i podaci koji će ko-rištano poslužiti u radu na tehnološkim i drugim projektan-askim poslovima. Ode god se smatrato korisnim i celishodnim, dati su primeri koji bolje ilustruju iznetu materiju.

- Iz sadržaja:
- Primenjena matematika u hemijskom inženjerstvu
  - Termodinamika
  - Principi i projektovanje tehnoloških operacija
  - A. Operacije prenosa količine kretanja
    - Fluidi u mirovanju
    - Fluidi u kretanju
    - Strujanje u cevima
    - Mekanje
    - Disperzni sistemi
    - Disperzna faza
  - B. Operacije mehaničkog tretiranja materijala
  - Transport čvrstog materijala
  - Sijanje materijala
  - Prosejavanje
  - C. Operacije prenosa toplote
    - Konvekcija toplote
    - Zračenje toplote
    - Toplotni bilansi
    - Cevni razmenjivači toplote
    - Uključavanje
    - Kondenzacija i kondenzatori
    - Radišni procesi
    - Goriva i sagoravanje
  - D. Operacije prenosa mase
    - Osnovni principi prenosa mase
    - Kontaktirni uređaji za prenos mase sistema gas-tečnost
    - Apsorpcija
    - Destilacija
    - Ekstrakcija tečno-tečno
    - Lužanje
    - Puhisanje i kondicioniranje
    - Suljenje i suljenje
    - Apsorpcija i jonaska razmena
    - Kristalizacija
    - Ostale difuzne operacije
  - Osnov teorije i projektovanje hemijskih reaktora
  - Katalize i katalizatori
  - Dinamika sistema, instrumentacije i automatska regulacija u hemijskom inženjerstvu
  - Osnove projektovanja proizvodnih sistema hemijske (procesne) industrije
  - Hemijsko-inženjerska ekonomika

Šesta knjiga  
**KOROZIJA I ZAŠTITA  
MATERIJALA**

Autori: S. Mladenović, M. Petrović, G. Riković. **Redaktor: O. Tatić-Janić.**  
Knjiga obrađuje teorijske osnove korozije i tehničko-tehnološke principe zaštite materijala.

- Iz sadržaja:
- Elektrohemija i hemijska korozije metala i legure
  - Korozije materijala neorganskog i organskog porijekla
  - Koroziona agresivnost raznih sredina
  - Zaštita materijala od korozije

Knjige su štampane latinicom, formata 15,5x22,5 cm, povezane u celo platno sa zlatotiskom, na oko 5.000 strana.

**Cena u pretplati . . . 12.600,-**

1949  
1984

Majstorije na računaru

# softverska veza

Struje i kanali — najmoćnije  
spectrumovo tajno oružje

Dobro je o periferijskom uređaju razmišljati tako što će se on podeliti u dva dela: jedan odgovara softveru a drugi hardveru koji prima ili generiše podatke. Kod ZX bezjeka softverska komponenta naziva se „struja“, a hardverska komponenta „kanal“. Ključna razlika je u tome što je struja bezličan protok podataka u program ili iz njega, dok kanal odgovara određenom uređaju, kao što je, na primer, ZX štampač. Struju treba shvatiti kao zbir podataka na njihovom putu ka nekom komadu hardvera ili od njega.

## Struje . . .

Struja je određena nekim brojem od 0 do 15; njene osnovne operacije su čitanje i pisanje podataka. Naredba:

```
INPUT #s; „ulazna lista“
```

čitae podatke iz struje „s“ i prebacivati ih u promenljivu koju smo nazvali „ulazna lista“. Na primer:

```
INPUT #0; A; B; A$
```

čitae podatke iz struje 0 i skladištiti ih u promenljive A, B i A\$. Na isti način, naredba:

```
PRINT #s; „print lista“
```

stae podatke strujom „s“ iz promenljivih u „print listi“. Na primer:

```
PRINT #0; TOTAL; A$
```

Obratite pažnju na to da se i INPUT i PRINT mogu koristiti na potpuno isti način kao i obične naredbe INPUT i PRINT. Svaki parametar koji se može koristiti kao i deo PRINT i INPUT može se uključiti i kao deo struje. Na primer, u važnosti su obe donje linije:

```
PRINT #2; „RAČUNAR“; TAB (10); „GALAKSIJA“
```

INPUT #0; „Kako se zoveš?"; NS  
Naredba PRINT šalje strujom 2 slovni niz „RAČUNAR“; zatim nekoliko praznina i, najzad, slovni niz „GALAKSIJA“. Obratite pažnju na to da podatak koji je poslat struji tačno odgovara podatku koji bi bio poslat ekranu. Naredba INPUT je nešto složenija, utoliko što ne zahteva samo podatak iz struje 0, već i šalje podatke u obliku slovno niza „Kako se zoveš?“. Svaki broj struje je, u stvari, vezan sa dve struje podataka: jednu ulaznu i jednu izlaznu struju. Podatak upisan u struji bilo naredbom PRINT ili naredbom INPUT, šalje se izlaznom delu struje, a svaki podatak koji se čita iz struje dobija se iz njenog ulaznog dela.

Sa „nerazvijenim“ (tj. bez mikrodajve) Spectrumom, mogu se koristiti još jedino naredbe LIST i INKEY\$. Potpuni oblik naredbe LIST je:



```
LIST #s.n
```

gde je „s“ broj struje kojom se program mora izlistati, a „n“ je broj linije od koje počinje listanje. Na primer:

```
LIST 1
```

izlistae ceo program pomoću struje 1.

Druga naredba orijentisana ka struji, INKEY\$, može se koristiti za prihvatanje pojedinačnog bajta iz bilo koje struje koja je vezana sa uređajem koji podržava ulaz.

Naredba:

```
INKEY$ s
```

vratice kod jednog karaktera iz struje „s“.

## . . . i kanali

Lako je shvatiti ideju o struji podataka, ali se postavlja pitanje kako se određena struja povezuje sa određenim periferijskim uređajem. Odgovor je: da bi se neki podatak poslao ili primio određenom strujom, ona mora biti najpre otvorena. Otvaranje struje ima dva cilja: ono povezuje potrebnu struju sa određenim periferijskim uređajem i, u isto vreme, signalizira koji će uređaj biti upotrebljen. Često, uz signaliziranje da će jedan uređaj neposredno biti korišćen, otvaranje struje obuhvata i uputstvo instalisanom uređaju kako da dođe u stanje u kome će se koristiti; ovo upućivanje, međutim, vrlo mnogo zavisi od samog uređaja. Otvaranje struje ostvaruje se naredbom:

```
OPEN #s, c,
```

gde je „s“ broj struje koja se otvara, a „c“ slovni niz koji određuje kanal sa kojim je ona povezana. Prema toj naredbi, određite

svakog podatka poslanog strujom „s“, biće kanal „c“, koji će, isto tako, biti i izvor svakog podatka pročitano iz te struje. Pre nego što se može dati praktičan primer korišćenja naredbe OPEN, treba znati kojim kanalima SPECTRUM raspolae.

Spectrum baz mikrobrojeva raspoznaje samo tri različita kanala:

- K — kanal tastature
- S — kanal ekrana
- P — kanal štampača

Tako

```
OPEN #5, „K“
```

otvara struju 5 i vezuje je sa tastaturom. Na osnovu naredbe:

```
INPUT #5; A; B
```

dobije se podaci sa tastature na isti način kao i pri normalnoj naredbi INPUT. Međutim, naredba:

```
PRINT #5; „GALAKSIJA“
```

šalje sada podatke izlazno strani struje 5, koja je povezana sa dnom ekrana. Na taj način, slovni niz „GALAKSIJA“ upisuje se u donji deo ekrana koji je normalno rezervisan za INPUT poruke. Ako to pokušate, nije verovatno da ćete moći videti niz, jer se donji deo ekrana briše čim se program zaustavi ili čim računar naiđe na naredbu INPUT. Ako hoćete da vidite efekat slanja podataka u „ulaznu oblast“ ekrana, pokušajte sa:

```
10 OPEN #5; „K“  
20 PRINT #5; RND  
30 GOTO 20
```

Videćete na ekranu nasumične brojeve, koji beže odzdo naviše. Program će se završiti sa porukom o grešci OUT OF SCREEN.

Prvi put kada poželite da na svoj Spectrum vezete neki periferijski uređaj, sudaricete se sa problemom kako mu slati ili kako od njega primati podatke. Obično se misli da je lakše konstruisati hardversku vezu sa Spectrumom nego „softversku vezu“ sa ZX bežikom. Međutim, Spectrum ima veoma elastična i sofisticirana metoda rukovanja periferijskim uređajima, zasnovana na korišćenju „struja“ i „kanala“, pa je razmerno lako povezati periferijske uređaje sa ZX bežikom tako da ih računar tretira kao svoj sopstveni hardver. Laksteci veoma jednostavan solver u mašinskom jeziku moguće je slati podatke svakom uređaju naredbama PRINT i LIST i primati podatke od svakog uređaja naredbama INPUT i INKEY. To, praktično, znači da se pirmenjeni programi mogu pisati u standardnom bežiku bez prečestog pozivanja mašinskog koda. Čudno je da se, uprkos tome što standardni Spectrum koristi tako elastičnu tehniku, ona jedva i pominje u priručniku!

## Mapa kanalskih zapisa

**kanal za tastaturu**  
 CHANS — adresa print rutine za donji deo ekrana  
 +2 — adresa input rutine za tastaturu  
 +4 — K kod kanala K  
**kanal za ekran**  
 +5 adresa print rutine za ekran  
 +7 — adresa rutine za tretiranje grešaka  
 +9 — S kod kanala S  
**kanal za redaktorski bafer**  
 +10 — adresa bafers input rutine  
 +12 — adresa rutine za tretiranje grešaka  
 +14 — R kod kanala R  
**kanal za ZX printer**  
 +15 — adresa rutine za ZX printer  
 +17 — adresa rutine za tretiranje grešaka  
 +19 — P kod kanala P

Mada u principu svaka struja ima ulaznu i izlaznu stranu, u praksi jedini kanal koji može da primi i ulaz i izlaz jeste kanal koji tastature. Druga dva kanala, ekrana i štampaća, samo su izlazni kanali i svaki pokušaj da se sa njih očitavaju podaci završicе prijavljivanjem greške J. Treba primetiti da je ovo ograničenje isključivo karakteristika hardvera za koji je vezana struja.

Za jedan kanal je moguće vezati više od jedne struje, ali ako se želi promeniti kanal sa kojim je struja vezana, veze se najpre mora ukloniti sa CLOSE. Dakle:

CLOSE # s  
 ukloniće svaku postojeću vezu između struje „s“ i kanala. U ovom smislu, CLOSE je za kanal suprotno od OPEN. Zatvaranje struje može se koristiti i zato da se komponenta hardvera jednog kanala obavesti da struji više nije potrebna.

## Formati kanala

Četiri struje, 0 do 3, automatski se otvaraju prilikom inicijalizacije računara:

struja	kanal
0	K
1	K
2	S
3	P

Tako će, čak i bez OPEN, naredba:

PRINT # 2; „GALAKSIJA“

ispisati tekst na ekranu. Spectrum koristi ove struje za upućivanje programskih podataka u odgovarajući uređaj. Na primer, LPRINT šalje podatke ka struji 3. Ovi odnosi struja i kanala mogu se pomeniti korišćenjem naredbe OPEN, ali same struje se ne mogu zatvoriti. Pokušaj da se zatvori jedna od ovih struja završava njenim ponovnim otvaranjem ka svom prvobitnom kanalu.

Ako struje i kanali treba da se koriste za povezivanje dodatnih uređaja, tada je očigledno važno znati unutrašnje formate koji su korišćeni za skladištenje informacije o kanalima, kao i koji je kanal vezan sa kojom strujom.

Informacija koja definiše svaki kanal uskladištena je u zoni kanalskih informacija polazeci od CHANS i završavajući sa PROG-2 (gde su i CHANS i ZPROG sistemске promenljive). Svaki kanal ima poseban „kanalni zapis“ sledećeg formata:

n — adresa početka kanalskih informacija  
 n+2 — broj adresa izlaznog programa  
 n+1 — broj adresa ulaznog programa  
 n+1 — broj bajta kanala

gde su ulazni i izlazni programi sabrutine mašinskog koda. Izlazni program mora da prihvati Spectrumove kodove karaktera koji mu se „doturaju“ preko A registra. Ulazni program mora da vrati podatke u obliku Spectrumovih kodova karaktera i da signalizira da su podaci dostupni setovanjem tzv. flega za prenos (carry flag). Ako nikakvi podaci nisu dostupni, to se signalizira resetovanjem Z i C flega. Ako kanal ne može da prihvati podatak sa ulaza ili izlaza, o tome treba da bude obavešten program koji tretira greške. Standardni način tretiranja grešaka u ZX bežiku jeste putem jednog redovnog partiza na adresi 8. Na mašinskom jeziku to izgleda ovako:

```
ERROR RST 0008
DEFB errocode
```

gde je „errocode“ numerički kod poruke o grešci koja se daje korisniku.

Kanalni zapisi za tri standardna Spectrumova kanala i jedan dodatni kanal koji do sada nije opisan prikazani su u Tabeli 1.

Obratite pažnju na to da je svaki kanalni zapis standardnog formata i da jedino K kanal može da prihvati i ulaz i izlaz. Novu, R kanal Spectrum koristi interno za slanje podataka redaktorskom baferu. Naredba OPEN neće dozvoliti korisniku da poveže struju sa R kanalom, pa je tako njegovo korišćenje ograničeno.

Važna primedba: format jednog kanalnog zapisa direktno je kada se koriste mikrodraji i interfejsi.

## Tablica struja

Veza struja sa kanalima uskladištena je u tablici struja, u oblasti sistemskih promenljivih, na prostoru od 36 bajta, polazeci od STRMS (adresa 2356). Tablica struja i svaki par bajtova u toj tablici nose jedan broj „x“ koji predstavlja adresu početka jednog kanalnog zapisa. Naredno prostog skladištenje adresa kanalnog zapisa, „x“ je „rastojanje“ na kome se kanalni zapis nalazi od početka zone kanalskih informacija:

početna adresa = adresa kanala + x - 1  
 zone kanalskih zapisa

Pošto postoji maksimalno 16 struja, moglo bi se pomisliti da su dovoljna 32 bajta (tj. jedna kanalna adresa po struji) za skladištenje svih asocijacija kanala i struja. U stvari, dodatnih 6 bajta koriste se za skladištenje kanalnih informacija za tri unutrašnje struje koje odgovaraju brojevima struje 255, 254 i 253. Te tri unutrašnje struje su automatski povezane sa kanalima R, S, odnosno K i nisu dostupne iz bežika. Međutim, prisustvo te tri unutrašnje struje mora se uzeti u obzir kad se pokušava naći adresa kanala koji odgovara na kojoj od struja 0 do 15. Prva tri ulaza u tablicu struja odnose se na unutrašnje struje 253 do 255, a četvrti ulaz daje adresu kanala koji će se koristiti sa strujom 0 itd. To znači da je početak tablice struja, što se spoljnih struja tiče, na adresi:

STRMS+6 ili 23574

Adresa kanala koji povezan sa strujom s (s je u okviru 0 do 15) skladišti se u dva memorijska mesta sa početkom na

23574+S\*2

Kada se jedna struja otvori za jedan određeni kanal, ispituje se tablica struja da bi se našla adresa kanala. Kad je struja zatvorena, njena adresa za ulazak u tablicu struja podešena je na nulu. Tako se nulti ulazak u tablicu struja koristi za otkrivanje pokušaja da se koristi struja koja još nije bila otvorena.

Ovaj sistem kanalnih zapisa i tablice struja proširen je i na korišćenje mikrodraja, ali njegove bitne karakteristike ostaju iste. Svaki kanal je opisan kanalinim zapisom, a struje su povezane sa kanalinim zapisima pomoću tablice struja.

Pre nego što pređemo na razmatranje kako praktično iskoristiti poznavanje struja i kanala, vredi pomenuti jednu sistemsku promenljivu: CURCHL. U njoj se čuva adresa tekućeg kanala. Na taj način, nakon naredbe kao što je PRINT # S, CURCHL sadrži adresu kanala povezanog sa strujom „s“.

## Kanalni po meri

Bežik naredbe IN i OUT predstavljaju najjednostavniji način za slanje, odnosno primanje podataka direktno sa odgovarajućeg ulazno-izlaznog porta. Na primer, ako je generator zvuka dodeljen port 31 za kontrolu njegove frekvencije, tada će

OUT 31,

slati podatke (f = 0 do 255) koji će direktno uticati na visinu tona. Za jednostavne uređaje i uređaje koji se kontrolišu pojedinačnim bitovima, IN i OUT su vrlo pogodni naredbe. Međutim, ako je uređaj „znakovno orijentisan“, tj. ako prima i odašilje podatke u obliku znakova, tada su IN i OUT nepodesni. Na primer, paralelni štampači i modem su znakovno orijentisani uređaji, i najpode-

```

10 DATA 01,00,254,237,121,201
20 FOR a=23297 TO 23301
30 READ d
40 POKE a,d
50 NEXT a
100 GO SUB 1000
110 FOR i=0 TO 7
120 LPRINT i:
130 NEXT i
140 GO TO 110
1000 LET c=PEEK 23631+256*PEEK 2
3632
1010 LET c=c+15
1020 POKE c,23296-INT (23296/256
)
1030 POKE c+1,INT (23296/256)
1040 RETURN

```

sniji način da se sa njima radi jeste putem običnih PRINT i INPUT naredbi. Jasno je da najbolji put vodi preko struja i kanala.

Uključivanje novog perifernog uređaja u sistem struja i kanala može se ostvariti na jedan od dva načina: ili promenom adresa uskladištenih u nekom postojećem kanalnom zapisu ili stvaranjem jednog potpuno novog kanalnog zapisa.

Prvi metod se sastoji u ubacivanju (POKE) novih adresa u postojeći kanalni zapis. Kanal će, tako, biti usmeren na vaše spovštene programe na mašinskom jeziku, smeštene negde u memoriji. Na primer, pretpostavimo da želite da povežete standardni štampač umesto ZX štampača. Umesto postojeće adrese na prvá dva mesta kanalnog zapisa ZX štampača, (CHANS + 15), treba da postavite adrese spovštenog programa za pogon štampača, i naredbe LPRINT i LLIST, kao i svaka druga naredba koja se odnosi na struju otvorene ka kanalu P, slaće podatke novom štampaču.

Pisanje novog programa za upravljanje štampačem je, u principu, lako: sve što treba da se uradi jeste dovesti ASCII kodove u A registar i poslati ih u štampač. Međutim, Spectrumov set znakova obuhvata i mnoge kodove koje standardni ASCII set karaktere ne poznaje. Pre nego što se pošalju preko štampača, ove kodove treba konvertovati. Na primer, svi kontrolni kodovi unutar PRINT naredbe biće poslani prema Dodatku A Spectrumovog priručnika. To znači da LPRINT TAB (10) šalje ASCII kodove 23, 10 i 0; 23 je Spectrumov kontrolni kod za TAB, a sledeća dva koda su najmanji i najvećnji bajt parametra TAB funkcije. Ovi kodovi, međutim, za štampač mogu da znače nešto sasvim drugo. Isto tako, kodovi za bejzik naredbe ne znače za štampač ama baš ništa. Oni se, zato, moraju prevesti u niz ASCII znakova, tako da se postigne efekat kao kada se svaka naredba kuca slovo po slovo.

Primer ovog metoda povezivanja nekog novog perifernog uređaja predstavlja sledeći program koji umesto izlazne adrese uskladištene u P kanalu postavlja adresu mašinskog programa koji je uskladišten u bafaru za štampač. Ovaj sistem funkcioniše samo stoga što je ZX štampač van upotrebe! Nov izlazni program ne radi ništa korisno sa podacima, osim što ih šalje ka portu 254 koji kontrolishe zvučnik i ivičnu

Tabela 2

23296	outdrv	LD	BC, 254	— unesi u BC registar 254
23299		OUT	(C), A	— pošalji sadržaj registra A na port 254
23301		RET		— vrati se u bejzik

Jednostavni program za opsluživanje perifernih uređaja u mašinskom jeziku

boju; to obezbeđuje da se njegovi efekti vide i čuju: Rutina kojom se postizu ovi efekti data je u tabeli 2. Ona je korišćena u bejzik listinju prikazanom u Programu 1.

Mašinski kod izlaznog programa skladišti se u DATA naredbi u liniji 10 i unosi u memoriju pomoću linija 20 do 50 (23296 = početna adresa bafara za štampač). Subrutina 1000 menja adresu u kanalnom zapisu za kanal P. Linija 1000 dobija adresu početka kanalne oblasti u C (CHANS), a zatim linija 1010 nalazi početak kanalnog zapisa za kanal P(CHANS + 15). Linije 1020 i 1030 ubacuju (POKE) adresu novog izlaznog programa u prvá dva mesta u kanalnom zapisu. Ako stvarujete ovaj program, videćete ivični blesak i veoma divnu promenu. Ako zaustavite program, dobićete dalji dokaz da novi izlazni program šalje podatke kontrolnom portu za border i izlavljanjem (LLIST) programa. (Pažnja: pre primene ovog programa treba odvojiti ZX štampač).

## Novi kanali

Menjanje adresa uskladištenih u postojećim kanalnim zapisima predstavlja lak metod za povezivanje novih uređaja, ali, na žalost, on onemogućuje jedan od Spectrumovih postojećih I/O uređaja. U praksi je nemoguće modifikovati kanal K, jer se njegove adrese ponovno uspostavljaju kad god se izvrši naredba INPUT. Kanali S i P ostaju, tako, kao jedini kandidati za modifi-

PROGRAM 2

```

10 DATA 0,91,11,91,69,1,0,254,
237,121,201,207,18
20 FOR a=23296 TO 23308
30 READ d
40 POKE a,d
50 NEXT a
100 LET s=5: GO SUB 1000
110 PRINT "RND:"
120 GO TO 110
1000 LET a=23574+2*s
1010 LET c=PEEK 23631+256*PEEK 2
3632
1020 LET r=23296+c-1
1030 POKE a,r-INT (r/256)*256
1040 POKE a+1,INT (r/256)
1050 RETURN

```

23296	chanrec	BEFB	5	— bajt manje težine u izlaznoj adresi
23297		DEFB	91	— bajt veće težine u izlaznoj adresi (91*256+5=23301)
23298		DEFB	11	— bajt manje težine u ulaznoj adresi
23299		DEFB	91	— bajt veće težine u ulaznoj adresi (91*256+11=23307)
23300		DEFB	„E”	— kod novog kanala („E”=69)
23301	outdrv	LD	BC, 254	— unesi u registar BC 254
23304		OUT	(C), A	— pošalji sadržaj akumulatora na port 254
23306		RET		— vrati se u bejzik
23307	indrv	RST	8	— pozovi rutinu za greške
		DEFB	18	— kod za poruku „Invalid device”



Pod velom tajni: lako se smatra da se o njemu već odavno sve zna. Spectrum ne pre-staje da iznenadjuje programere

linije 110 do 120 šalju kodove koji odgovaraju nasumičnim brojevima zvučnom i vidnom kontrolnom portu. Zaustavljanjem programa i unošenjem naredbe LIST # 5 može se proizvesti i blesak boje i zvuka, što znači da je program izlistan ka portu 254! Ako promenite liniju 110:

110 INPUT # 5; i

dobićete poruku o grešci, što znači da se taj kanal ne može koristiti za ulaz. Naravno, umesto 254 može se postaviti adresa nekog štampača, programatora EPROM-a ili gene-

ratora zvuka.

Ovim što se moraju pisati obuhvatni i specijalizovani programi, nema nikakvih teškoća u dodavanju novih kanala u zapis. Jedino treba voditi računa da gornji program neće funkcionisati ako su spojeni mikrodrajvovi.

## Ulazni kanal

O problemima pisanja izlaznog drajvera već je ranije bilo reči. Na kraju, treba

pomenuti dodatne zahteve za ulazni drajver. Ako jedan ulazni kanal treba, poput osmobičnog A-D konvertora, da prihvata pojedinačni znakovni kod, tada je najbolja bežična naredba INKEYS, # koja će vratiti pojedinačni znak. Međutim, ako planirate da koristite INPUT da biste učitali čitav znakovni niz, tada morate biti svesni dve stvari. Prvo, naredba INPUT ima i izlaznu struju, i nije dovoljno zameniti izlazni program programom za tretiranje greške; mora se obraditi svaki podatak koji šalje naredba INPUT, čak i u slučaju da se ona prosto zanemaruje! Drugo, naredba INPUT prihvata podatke kao da su otkucani na tastaturi. To, praktično, znači: ako koristite INPUT # s; B, da biste učitali neki broj u promenljivoj B, program mora da šalje niz ASCII kodove koji odgovaraju brojevima i da se završi sa jednim ENTEH kodom, kao da broj ulazi sa tastature. Na kraju, treba obratiti pažnju i na ovo: čak kada se podaci očitavaju i sa specijalnom komanda hardvera, naredba INPUT će tačno interpretirati sve kodove — ona uvek funkcionise kao da prima struju znakovnih kodova koji odgovaraju direkta-  
noje su pritisnute na tastaturi.

Spectrumov sistem struja i kanala predstavlja neočekivanu premiju za programere na ZX bežičnu. Korisćen unutar programa, on obezbeđuje nezavisnost uređaja i uopšte povećanje elastičnosti bez ikakvih negativnih strana. Programeru na mašinskom jeziku struje i kanali pružaju idealnu alatu za softverske interfejse sa novim uređajima. Primeri dati u članku predstavljaju polaznu tačku za sve realne softverske interfejse koje biste možda želeli da napišete.

(Electronics and Computing Monthly)

## Mali oglasi

- Spectrumovci — veliki izbor naj programa po najnižim cenai zveni novi popusti: hobit, penetrator, horace, psst i drugi. Cena svih programa je 200 ND Tel. 345-480
- Disk-drive za Apple (II, II+ i Ie), potpuno nov, prodajem. Tel. 011/4443-351

### software club

Garantujemo najniže cene u zemlji. Bilo koji program po vašem izboru za 100 din, — de-set programa za samo 600 dinara

- Manic Miner
- Penetrator
- Hobit.

i rncigji drugi programi koji stižu iz Engleske. Svi programi su snimljeni i verifikovani direktno iz Spectruma. Pišite ili se javite telefonom: Andrić Dejan, Dančeva 64, 11060 Beograd, tel. 011/784-041

- GALAKSIJA — komplet čipova samo 7500 din. (2K verzija). Sva-kih dodatnih 2K samo 1750 din. Isporuka odmah. SHARP 1500 i printer 80.000 din. Sezovic J. Jakov. Pive Karamatjevića 31, Beograd. Tel. 011/496-665

ZX SPECTRUM 16/48K i dalje najjeftiniji software u gradu SAMO 50 DINARA PC PROGRAMU. Puno novih programa za 16 i 48K.

Skripte za mašinic. Super brza usluga. Specijalni programi po porudžbini.

TRAŽITE NOVI BESPLATNI KATALOGE

Kostić Vladimir. Varvarinska 35, 11000 Beograd, tel. 011/4 00-823

### COMMODORE 64

Programski jezici kompajleri asembleri word procesori najnovije igre kućni programi posivni programi muzički sintesajeri sekvenceri. kompozeri edukativni programi kurs BASIC-a 130 KB govor-voice generator super grafika spreid sheet razmena programa Veselin Milosavljević, 11000 Beograd, Vitanovačka 42, stan 45, tel. 011/462-659 (po podne)

I dalje sa Vama COMPUTERLAND

Naša specijalnost je ZX SPECTRUM Sistemski programi Namenski programi igre Besplatan blic-katalog (na- i progr. i cene) Katalog sa opisima svih progr. i cenama 100 - din. „COMPUTERLAND“ J. Gagarina 120/62 11070 Novi Beograd TEL. (011) 162-774

ZX-Spectrum programi na kaseta-ma. Za 80 din. nudim pas-cal, forth, devpac, arcadia, pen-etrator, time-gate, jet-pac, flight simulation, football ma-nager, 3D-tunnel, hobbit, cyrus chess, manic miner, monopoly, kong, mr. wimpy, pool, pinbox, anti-attack, high-noon, bridge, battle 1917, još 250 d rubig. Za katalog pošaljite 20 dinara. Razmenjujem VHS-video materijal. Marko Marković, Dž. Bjedica 27/XI, 71000 Sarajevo

Za comodore-64 VIC-20 preko 200 fabričkih programa prodajem. Bes-platan katalog? Moguća razme-na? Jeftino? Andrija Kolundžić, Vojvode Brane 31/IV, 11000 Beograd, tel. 011/424-325

Comet SOFTWARE

Sve za ZX SPECTRUM

— video igre

— ili glske igre

— namenski programi

— knjige

SPECIJALNO!!!

Ako vam trebaju usluge na printeru obratite nam se. VRSIMO:

— p programski listing

— as emblerski listing

— kopiranje crteža u vizuelnoj rezoluciji.

Iskoristite LETNJI PUSTI!!!

MILIVANOVIĆ LUBIŠA

Petka Lekovića 57

1100 0 BEOGRAD

tel: 01 1/558 007

Skripte za ZX81 i spectrum na

sprekovratakom

1 basic (800 N. din)

2 mašinski jezik (800 N. din.)

3 najbolji i najnoviji listinzi pro-grama iz stranih časopisa (800 N. din.) 50 str.

4 seme i uputstva za hardver

dodatka u samogradnji (A/D ko-vertor, 1/0 port, RS232C interfejs,

light pen i mnogi drug...) na 50 strana (800 N. din.)

Visanici kompajlera

Comodore 64 i Commodore

VIC 20

Javite se radi razmene programa

i literature. Kolundžić Andrija,

Vojvode Brane 31/4, Beograd, tel. 011/424-325

# škola Igre na računaru

## Akcionijah igara

Cilj škole je da vas, kroz dosta primera, uvede u neke programerske veštine koje će vam pomoći da kreirate sopstvene akcione igre. Neki od primera programa predstavljaju interesantne igre koje, doduše, ne mogu da se mere sa komercijalnim programima, ali koje, obzirom da ćete ih sami otkućati i shvatiti njihovu realizaciju, mogu da vam pruže mnogo časova zabave, naročito ako spadate u one vlasnike računara koji ne žele samo da se igraju.

Svi znamo da su dobre akcione igre moguće samo na mašinskom jeziku, a ipak smo ovu školu propratili bezik primerima. Zašto? Čini nam se, pre svega, da je nedovoljan broj čitalaca ovih redova dobro upoznat sa mašinskim jezikom, pa bi mašinski programi odvlačili pažnju od onoga što program treba da ilustruje. Kada savladate mašinski, pogotovu ako posedujete knjigu „Disasembirani Spectrum ROM“, neće vam biti teško da primere bukvalno prevedete na mašinski jezik, ili da, u programima koje ćete sastavljati, koristite tehnike naučene u ovoj školi.

Škola je, u osnovi, namenjena vlasnicima Spectra, koji će imati privilegiju da sve date primere isprobaju na svom računaru bez ikakvih prepravki. I za vlasnike drugih računara, međutim, može da bude korisno da je pročitaju: tehnike koje ćemo objasniti su univerzalne, dok naredbe, jasno, mogu da budu izmenjene. Dovoljno je da držite uputstvo za upotrebu vašeg računara na dohvat ruke dok čitate ove redove i sigurno ćete se snaći bez većih problema.

### „Pritisni bilo koji taster...“

Nekada su računari bili namenjeni uglavnom obradi brojeva koji su se unosili primenom naredbi koje su vrlo slične dobro nam poznatoj naredbi INPUT. Jasno je da na ovaj način (korisniku je ostavljeno skoro beskonačno vreme za razmišljanje i prijem podataka kvazi izgled ekrana) ne mogu da se primaju podaci potrebni za akcione igre, pa je u bezik uvedena naredba INKEY\$ koja daje kod pritisnutog tastera. Ako se, na primer, u programu nalazi naredba 5 LET XS=INKEY\$, promenljivoj XS će biti dodeljeno slovo a ako je igra pritisnuta s, a ako je pritisnuto SHIFT A i tome slično. Kod Spectra je važno primetiti da naredba INKEY\$ ne registruje trenutak kada je taster pritisnut kao kod mnogih drugih kompjutera. Ukoliko, na primer, računari neprekidno izvršava naredbu 5, a korisnik stalno drži taster A pritisnut, X će neprekidno dobijati vrednost „A“.

Naredba INKEY\$, i pored ogromnih prednosti nad INPUT, nije pogodna za akcione igre. Zašto? U mnogim igrama je neophodno da se korisniku omogući istovremeno pritisnjanje dva ili više tastera (kreiranje i pucanje, na primer). Zato Spectrum poseduje moćniju i, na žalost, komplikovaniju naredbu IN. Ona ne daje kao rezultat alfanumerik nego broj. Ova naredba, osim toga, ne testira jedan nego čitavih pet tastera u isto vreme; tih pet tastera ćemo nazivati blok.

Svaki blok tastera ima svoj broj koji nalazimo u tabeli 1 i stavljamo iza IN naredbe (ako, na primer, želimo da testiramo taster 1, 2, 3, 4, i 5, koristećimo naredbu LET A=IN 63486). Ukoliko ni jedan od tastera iz bloka nije pritisnut, IN će vratiti broj 255 (uz jednu malu ogradu ali o njoj nešto donje). Svakom pritisnutom taster smanjuje ovaj broj za 2 dignuto na stepen n, gde se n takode čita iz tabele. Neka su, na primer, pritisnuti tasteri 1, 2 i 4. PRINT IN 63486 će tada dati broj 255-2<sup>0</sup>-2<sup>1</sup>+2<sup>3</sup>=244. Oni malo bolje upoznati sa binarnom logikom, svakako će zapaziti da je svakome tasteru dodeljen po jedan bit koji je „setovan“ ako je taster otpušten i „resetovan“ ako je on pritisnut. Zato je sasvim jednostavno ispitati da li su, na primer pritisnuti istovremeno tasteri 1 i 2, pri čemu se ne obraća pažnja na stanje preostala tri i naredba kojom se to postigne je:

```
IF (IN 63486 AND 3)=0 THEN PRINT „Pritisnuti su 1 i 2“
```

Čini nam se da će vam biti od koristi da uložiti određeni napor u razumevanje ove naredbe, jer ćete slične imati prilike da koristite u mnogim drugim prilikama. IN 63486 daje broj između 0 i 255 koji možemo da zamislimo kao niz od 8 binarnih jedinica i nula. Desna dva bita odgovaraju tasterima 1 i 2 respektivno ukoliko je neki od njih 0, odgovarajući taster je pritisnut. Preostali bitovi, međutim, imaju sadržaj koji zavisi od toga da li su tasteri 3, 4 i 5 pritisnuti, što nas, kao što rekoh, ne interesuje. Treba, dakle, da pronađemo način da zanemarimo vodećih 6 bitova dobijenog broja odnosno da ih prevorimo u nule. Upravo je tome namenjena konstrukcija... AND 3. Zapazimo, najpre, da se broj 3 funkcija binarno kao 0000011. Logička funkcija AND („i“) nalaze računaru da proredi brojeve IN 63486 i 0000011, bit po bit. Ako su oba bita koji se poredje jedinice, rezultat će takode biti 1, u protivnom rezultat je nula. Obzirom da broj 3 ima šest vodećih nula, prvih šest bitova rezultata će takode biti nula, što smo i želeli. Poslednja dva bita neće promeniti vrednost jer ih logički množimo (AND se ponekad naziva logičko množenje a OR logičko sabiranje) jedinicom. Ukoliko su tasteri 1 i 2 pritisnuti, ta dva bita (i, samim tim, čitav rezultat) će biti jednaki nuli, dok će u protivnom rezultat uvek biti različit od

nule. Ovakvo „maskiranje“ je vrlo često u upotrebi, pogotovu kada pišemo mašinske programe jer predstavlja najcelishodniji način izdvajanja nekoliko bitova i jednog bajta.

Kao primer svega izloženog, dajemo jednostavni program za pisanje i crtanje po ekranu. Iako to nije akciona igra, u svakoj akcionoj igri mora da se crta po ekranu zavisno od onoga što korisnik kuca, pa će tehnike naučene analizom ovoga programa svakako biti od koristi. Primetite da se u programu na više mesta koristi naredba INKEY\$ — pokazuje se da je ona jednostavnija od IN, pa zato ovaj drugi treba koristiti samo u slučajevima kada postoji mogućnost — u našem primeru u naredbama 50 i 60.

Obratimo, najzad, pažnju na liniju 20 u kojoj se skriva naredba BEEP koja, naravno, ne smeta, ali nije ni mnogo potrebna. Došlo je, dakle, vreme da izložimo „ogradu“ koju smo pomenuli: Spectrum operativni sistem ne mora uvek brojeve pritisnutih tastera oduzimati od 255 — u upotrebi su i druge vrednosti. U nekim slučajevima, međutim, računari se sigurno vraća u „normalizovano“ stanje i usvaja ovu konstantu. Najjednostavniji recept za ovo „normalizaciju“ je izvršiti BEEP pre ispitivanja tastera pomoću IN. Ovo objašnjenje morate, na žalost, da primite kao i mi — zdravo za gotovo. Naredba IN, jednostavno, ima mnoge druge namene osim ispitivanja tastature, pa se pojavljuju i neki artefakti.

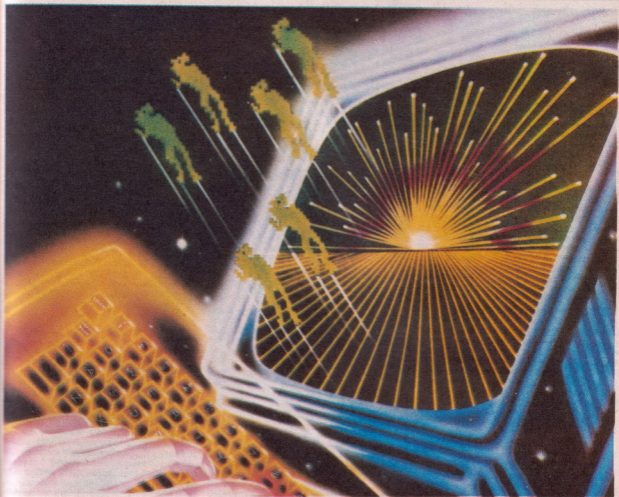
### Marsovc dolaze

Sve akcione igre mogu da se kreiraju korišćenjem naredbe PLOT, tj. crtanjem figure svaki put kad se ukaže potreba tačku po tačku. Ovaj način je mukotpan i, što je mnogo važnije, ne daje dobre rezultate kada je potrebna animacija. Često su oblici koje treba prikazivati relativno mali (figure čoveka u „pakmenu“, svemirski brod u „invejdjersima“, vočka kod „Horacija“...), pa možemo da ih svvedemo na veličinu jednog karaktera, koji ćemo, jednostavnim PRINT, prikazivati na ekranu gde god je potrebno. Karakteri se kod Spectra (i kod većine drugih sličnih kompjutera) definišu na matrici 8x8 kao što ćemo videti iz sledećeg jednostavnog primera.

Pretpostavimo da želimo da definišemo karakter u obliku figure čoveka sa slike. Najpre smo ga, kao što se vidi, upisali u matricu kvadrata tako da u njoj nema crna a neka bela (često je lako uklopiti neki složeniji oblik u prilično malu matricu — za ovo umetnost bivaju angažovani i profesionalci kada se pišu komercijalni programi). Zatim ga pretvaramo u brojeve i to red po red (vlasnici drugih računara će ušeteti sebi dosta napora ako provere da se to na njihovom kompjuteru ne radi kolonu po



Svaki vlasnik Spectruma je sigurno nabavio neki komercijalni program za akcionu igru (neki ih imaju na desetine, a možda i na stotine) i, upotrebljavajući ga, pozavideo njegovom autoru: kako je moguće da je igra ovako dobro zamišljena, da je ekran ovako efekatan, da su zvučni efekti ovako dobri... Dobru akcionu igru, zaista, nije lako sastaviti — ne toliko zbog toga što je teško napisati program koliko zato što nije jednostavno zamisliti dobar i originalan scenario koga se neko drugi već nije setio. Kada zamislite scenario, realizacija programa je stvar tehnike, koja, kao što rekosmo, nije teška, ali je lakša ako sami ne morate da je otkrivete od samog početka. To je i smisao ove škole akcionih igara.



kolonu, što je vrlo čest slučaj) pri čemu "zacrtnjeno" polje zamenjujemo jedinicom, a ono koje to nije — nulom. Konstruktori Spectruma su nas snabdeli funkcijom BIN koja ovako dobijene binarne brojeve pretvara u dekadne. U našem primeru dobijamo:

red 1: BIN 00001000  
red 2: BIN 00010100  
red 3: BIN 00001000  
red 4: BIN 00111110  
red 5: BIN 00001000

red 6: BIN 00011100  
red 7: BIN 00100010  
red 8: BIN 00100010

Najbolje je da, kada smo ispisali ove brojeve, uposlimo računar na njihovom prevođenju u dekadne. Za prvi red otkucamo, jednostavno, PRINT BIN 00001000 i dobijemo broj 8, za drugi broj 20 i tako dalje. Zatim unosimo jednostavan program kao što je sledeći, startujemo ga i, od tog momenta, karakter „A“ postaje čovečuljaska slike.

Da bismo vam olakšali pisanje sopstvenih akcionih igara, dajemo tabelu sa nekoliko predloženih karaktera. Tu četa naći sadržaje koje treba uneti u DATA linije ha kraju programa da biste imali definisane

oblike koji se često koriste. Jasno je, naravno, da korišćenje karaktera koje će poznavati i hiljade drugih nije neka prevelika originalnost, ali ćete lako, promenivši neke detalje, dobiti mnogo originalnije oblike. Kada se radi na maloj matrici (u ovom slučaju 8x8), čak i promena jednog polja iz temelja menja čitav utisak o karakteru. Dajte, dakle, na volju vašoj umetničkoj mašti!

### Zvučni efekti

Nijedna komercijalna igra ne vredi mnogo bez interesantnih zvučnih efekata. Konstruktori Spectruma su, na žalost, ostavili jedino ne naročito fleksibilnu naredbu BEEP: koja ima dva argumenta razdvojena

IN63406	: keus 1,2,3,4,5
IN64510	: keus Q,W,E,R,T
IN65022	: keus A,S,D,F,G
IN65278	: keus Caps Shift,Z,X,C,V
IN61438	: keus 6,7,8,9,0
IN57342	: keus Y,U,I,O,P
IN49150	: keus H,J,K,L,ENTER
IN32766	: keus B,N,M,Space Shift,Break/Space

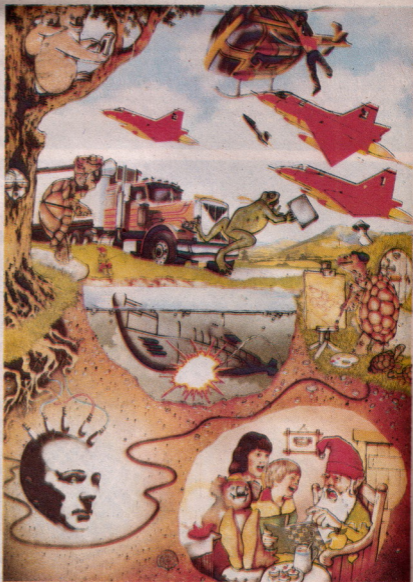
## SLIKA 2

```

10 GO SUB 1000
15
20 BEEP .1,10
25
30 IF m=1 THEN OVER 1
40 PLOT INK ink;x,y: LET a=x:
LET b=y
50 LET x=x+(IN 65022=251 AND x
<254)-(IN 65022=254 AND x>1)
60 LET y=y+(IN 64510=253 AND y
<166)-(IN 65278=251 AND y>1)
70 PLOT INVERSE (n=2): INK ink
: a, b
80 LET a$=INKEY$
90 IF a$="0" AND a$<"7" THEN
LET ink=VAL a$
100 IF a$="0" THEN PRINT AT 0,m
#11: FLASH 0: OVER 1: INVERSE 0:
" : LET m=m+1-(3 AND m
#2): PRINT AT 0,m#11: FLASH 1: O
VER 1: INVERSE 0: "
110 OVER 0: GO TO 30

120
1000 LET x=127: LET y=97: LET in
k=0: LET m=0
1010 BORDER 0: PAPER 7: INK 0: C
LS
1020 PRINT "Kretanje: "" H-ovor
e," "X-dole", " A-levo", "D-desno"
1030 PRINT " + dijagonalne komb
inacije"
1035 PRINT " 0-Prohena opcije"
1040 PRINT " "" "" Pritisni ne
ki taster."
1050 IF INKEY<>" THEN GO TO 10
50
1060 IF INKEY$="" THEN GO TO 106
0
1070 CLS
1080 PRINT FLASH 1: "CRTRANJE," : F
LASH 0: " POMERANJE, BRISANJE,
"
1090 PLOT 0,167: DRAW 255,0
1100 RETURN

```



Važnije od tehnike: Maštovitost i inventivnost predstavljaju ključ za dobru akcionu igru — sve ostalo se postiže gotovo samo po sebi

uz minimum memorijskog utroška proizvedemo druge melodije. Na slici je dat jedan jednostavan primer zvučnog efekta koji bi mogli da stvaraju napadači iz vasiona u toku spuštanja.

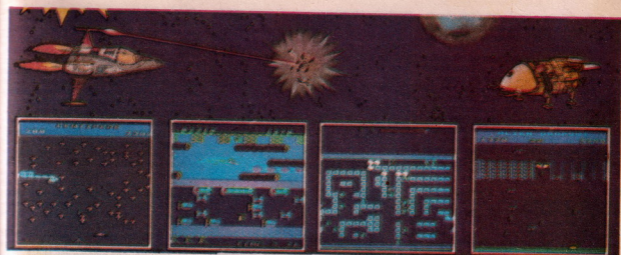
Vidimo da za svaki ton moramo da navedemo dva člana DATA liste, što nije samo veliki utrošak memorije već i gubitak vremena (READ nije ni izdaleka jedna od najbržih Spectrumovih naredbi). Zato je dobro odrediti se promenljive dužine tonova i nadoknaditi je „kompozitorskom veštinom“, kao u drugom primeru.

Na kraju poglavlja o tonovima jedan praktičan savet koji je više hardverske prirode. U prodaji se nalaze takozvane zvučne kutije za Spectruma („Sound boxes“). Njihova uloga je da pojačaju tonove koje računar generiše, čim zvučni efekti postaju

interesantniji ili, bar, privlače pažnju širem auditorijumu. Ovakve kutije možete sasvim uspešno zameniti bez ikakvog posebnog troška. Povežite, jednostavno, MIC ili EAR na Spectrum sa MIC na kasetofonu, pritisnite REC i PLAY (bez umetnete kasete) — to je sve. Zbog ekonomičnosti nekih rešenja Spectrumovog hardvera, kasetni ulaz i izlaz i generator tonova su tesno povezani, što omogućava da se kasetofon koristi kao pojačalo.

## Crtač u računaru

Svi znamo kako u bioskopu vidimo „pokretne slike“: svake sekunde preko ekrana „proleće“ 24 nepomične slike, ali mi, zbog tromosti oka, imamo utisak kretanja. Jasno je da nešto slično možemo da pokušamo kod računara, ali jednostavni eksperimenti pokazuju da su šanse samo teorijske: računar, pre svega, ne može da u jednoj sekundi prepíše 10 (još manje 24) sadržaja ekrana koji, inače, zaprema 8 Kb RAM-a. Osim



SLIKA 4

```
9000 RESTORE 9100
9020 FOR n=0 TO 7
9030 READ a
9040 POKE USR "a"+n,a
9050 NEXT n
9070 RETURN
9100 DATA 8,20,8,62,8,28,34,34
```

SLIKA 5

```
9000 RESTORE
9010 FOR i=1 TO 7
9020 READ duzina,nota
9030 BEEP duzina,nota
9040 NEXT i
9100 DATA ,2,7,,1,2,,1,2,,2,4,,4
2,2,2,6,,2,7
```

loga, memorija od 48 Kb može da primi samo šest ekrana, što nije dovoljno ni za pola sekunde „televizije“. Pa, kako je onda kompjuterska animacija uopšte moguća? Tajna je, naravno, u tome što najveći deo ekrana ostaje statičan, a pokreću se samo mali delovi (na primer, ranije definisani karakteri) i to ne proizvoljno nego tako da se njihov položaj u svakom trenutku može izračunati iz prethodnog položaja. Programer treba da bude toliko vešt da, promenom boja, istakne delove ekrana koji se pomću, a neutrališe statičnu pozadinu. To ne mora uvek da bude mnogo teško iako, svakako, zahteva određeno iskustvo. Mi samo početi od najjednostavnijeg: karaktere koji se kreću po ekranu.

Posmatrajmo primer sa prve slike koji predstavlja najjednostavniju primenu promenive ideje. Ako otkučate program, videćete samo tačku koja neprijatno treperi. Zašto? Naredba 30 se izvršava odmah posle naredbe 20 što znači da kvadratić biva brisan praktično trenutno. Sa druge strane, da bi kvadratić bio ponovo ispisan, računara treba da obavi jedno sabiranje i jednu GO TO naredbu što, naravno, traje puno vreme. Tačka je, dakle, mnogo teže isključena nego što je uključena i to proizvoljno neprijatno efekat.

*Zadovoljstvo bez premca: Za mladog programera nema uzvišenijeg trenutka od momenta kada na ekranu oživi njegova sopstvena igra*

Ako već ne možemo da ubrzamo sabiranje i GO TO naredbu, možemo bar da produžimo interval koji tačka provodi na ekranu dodajući jednu „mrtvu petlju“. Stvar možemo da začiniimo i sa malo muzike:

```
24 FOR N=1 TO 5 : NEXT N
27 BEEP .01.RND 20
```

Sada je već mnogo bolje, ali se ispoljava novi problem: čim kompjuter pokuša da štampa karakter izvan ekrana (na to ne treba tako dugo čekati), biva prijavljena greška. Prva ideja koju bismo mogli da primenimo je IF naredba koja ispituje da li se stiglo do kraja ekrana. No, Spectrumova IF naredba je spora i bolje bi bilo iskoristiti FOR-NEXT petlju (9 bi išlo od 1 do 31), ali postoji i elegantnije rešenje: da karakter koji se kreće, kada dođe do jednog kraja ekrana, nestane i pojavi se na suprotnom. Na sledećoj slici je dat program koji to omogućava. Početne linije izgledaju sasvim jasno sve dok ne stignemo do linije 40. Tamo nas očekuje neobična konstrukcija:

```
LET x=x+(y=31) Šta je to?
```

Malo bolji poznavaoци Spectruma znaju da se i logički izrazi izražavaju brojevima; (y=1) je logički izraz koji može da bude tačan ili lažan. Ako je tačan, računara će mu dodeliti broj 1, a ako je netačan — broj nula (kod mnogih drugih računara situacija je potpuno ista osim što se tačnom iskazu dodeljuje vrednost -1; ovo ćete najlakše ispitati ako otkučate PRINT (3=3) i pogledate rezultat). Kada, dakle, y postane 31, x će biti povećano za jedan, što znači da će računara početi da piše od početka sleded reda.

Drugi deo naredbe 40 izgleda još zagonetnije jer je ponovo iskorišćena naredba AND. Smisao ove naredbe je, da najpre to raščistimo, da računara ne prijavi grešku kada tačka, liniju po liniju, stigne do kraja ekrana. U tom slučaju bi trebalo da x ponovo postane nula tj. da se tačka pojavi u gornjem uglu ekrana. Da vidimo šta se dešava u oba kritična slučaja: kada je x jednako 22 i kada nije.

Ako je x jednako 22, onda i x=22 (obratite pažnju na redosled operacija) ima vrednost 1 koja, logički pomnožena sa 22, daje opet 22, što smanjuje x na nulu. Ako, pak, x nije 22, x=22, će dobiti vrednost nula, pa će nula logički pomnožena sa 22

dati ponovo nulu koja neće menjati vrednost X. Ako ste shvatili ovu logiku, ne bi trebalo da imate problema sa razumevanjem linije 50 koja proverava da li je stiglo do kraja ekrana. Tu liniju ćemo vam, dakle, ostaviti za vežbu.

## Glavom o zid

Tako smo, na relativno zadovoljavajući način, rešili problem kretanja tačke po jednoj liniji. Jasno je, međutim, da nam to nije dovoljno — potrebna nam je tačka koja će se mnogo slobodnije kretati po ekranu i odbijati se od drugih objekata na njemu. Za to moramo uvesti promenljivu koja će govoriti o pravcu kretanja. Za slučaj da se kretanje obavlja po jednoj pravci, ta promenljiva će se posle svakog štampanja i brisanja dodati trenutnoj vrednosti iksa (ako promenljiva ima vrednost 1, karakter će se kretati po jednoj polje udesno), dok se promena smeru postiče jednostavnim promenom znaka promenljive. Ovaj slučaj je, međutim, previše jednostavan, pa ćemo se pozabaviti kretanjem u svim smerovima. Njega karakteristično promenljiva d, koja će imati vrednost 1 za kretanje dole, 3 za gore, 0 za desno i 2 za kretanje levo (za trenutak ćemo videti zašto su brojevi tako raspoređeni), a x i y će, kao i do sada, označavati položaj tačke na ekranu:

```
1000 LET X=X+(d=1 OR d=2)-(d=0 OR d=3)
```

```
1010 LET y=y+(d=2)-(d=1)
```

Okoliko karakter „putuje“ preko ekrana i udari u horizontalni zid, d treba da promeni vrednost. To, na primer, možemo da postignemo naredbom:

```
2000 LET d=d+1-(2 AND (d=3 OR d=1))
```

Ako ste početnik, ove naredbe mogu da vas zaplaše. No, ne brinite, nije teško razumeti njihovo dejstvo: dovoljno je da se odlučite za neke konkretne vrednosti x, y i d i ručno, korak po korak, izračunate kako će se one promeniti posle izvršavanja ovakvih naredbi. Posle najviše dva primera sve će biti sasvim jasno!

Krajnje je vreme da izložimo nekoliko načina na koje može da se utvrdi da li je došlo do sudara dva objekta koja se kreću po ekranu. U nekim slučajevima možemo, jednostavnim IF, uporediti promenljive u koje su upisane njihove pozicije. U mnogim slučajevima, međutim, nije ekonomično trošiti memorijski prostor na koordinate svakog objekta (zamislimo samo da pišemo

program u kome igrač treba da prođe kroz „prebukirani“ pojas asteroida) kada je on već upisan na ekran, odakle ga, uz malo veštine, možemo pročitati.

Prvi način da to učinimo je naredba SCREEN (x, y) gde su x i y koordinate pozicije na ekranu koju želimo da „pročitamo“. Naredba će vratiti karakter koji se nalazi na tom mestu uz dva ograničenja koja su veoma ozbiljna: predefinisani i inverzni karakteri se nekorektno detektuju.

Drugi način je naredba ATTR koja, istina, ne daje obaveštenja o karakteru na nekom mestu, ali zato govori sve o njegovoj boji. Ukoliko uspemo da svaku vrstu objekta obojimo na određeni način ATTR će nam pomoći da se orijentišemo. Kao i SCREEN, ATTR ima dva argumenta koji određuju mesto na ekranu. Kao rezultat se dobija jedan bajt „spakovan“ kao zbir sledećih sabiraka:

128 ako karakter blinkuje

64 ako je karakter posebno svetao

8 puta boja „papira“ (PAPER)

1 puta boja „mastila“ (INK) kojim je karakter „pisan“.

ATTR ne obraća pažnju na to da li je karakter standardan ili ga je korisnik sam definisao iz jednostavnog razloga što i ne analizira taj karakter — kao što joj i ime govori. ATTR analizira atribute karaktera (o atributima smo dosta govorili u „Računari- ma 1“ kada smo objašnjavali kako Spectrum kodira sadržaj ekrana).

Poslednji način predstavlja naredba POINT koja nije u velikoj upotrebi iako je neobično moćna. Prate je dva argumenta koja daju poziciju na ekranu, ali ne poziciju karaktera nego poziciju tačke, kao kod naredbe PLOT x, y. POINT vraća broj 1 ili 0 — jedan ako je tačka čije su koordinate date osvetljena, a nula ako je zatamnjena. Zašto je ova moćna naredba kojom možemo da ispitamo status bilo koje tačke na ekranu manje u upotrebi? Jednostavno zato što su bezig programi sporiji i ne mogu sebi da dozvole luksuz rada sa tačkama — radi se isključivo sa karakteristikama. Primenom POINT možemo, jasno, da ispitamo svaki karakter tačku po tačku, ali tako ispitavanje oduzima previše vremena da bismo mogli da ga preporučimo.

Da ilustrujemo sve ovo, objavljujemo program „Lopta i reket“. Obzirom da se kreću samo dva objekta (lopta i reket), igra je prilično brza i dinamična. Kao i obično, ne treba samo da se zabavljate njime (za zabavu ima i mnogo boljih igara), nego da pokušate da ga analizirate, u čemu će, nadamo se, pomoći i komentari uključeni u REM linije.

## Skrolovanje

Skrolovanje je trenutak kada se svi objekti (slova, specijalni karakteri i crteži) pomeraju nagore za jedan red da bi se oslobodio prostor za novu liniju. Za normalan rad i pisanje programa (posebno kod kompjutera kod kojih se naredbe kucaju po čitavom ekranu: kod Spectruma su za naredbe odvojene samo poslednji redovi), skrolovanje je svakako neophodno. Spectrum, međutim, posle ispisivanja svake 22 linije očekuje od korisnika da potvrdno odgovori na pitanje Scroll? (za potvrdan odgovor se smatra pritisak na bilo koji



Iz jedne avanture u drugu: Prizori iz igračkog života

SLIKA 6

```
2000 FOR n=10 TO -10 STEP -.5
2010 BEEP .02,n
2020 NEXT n
```

SLIKA 7

```
10 LET y=0
20 PRINT AT 10,y;" "
25 PAUSE 1
30 PRINT AT 10,y;" "
40 LET y=y+1
45 IF y>31 THEN GO TO 10
50 GO TO 20
```

SLIKA 8

```
10 LET x=0: LET y=0
20 LET a=x: LET b=y
30 PRINT AT x,y;" "
40 LET x=x+(y=31): LET y=y-(x<22
AND x=22)
50 LET y=y+1-(32 AND y=31)
60 PRINT AT a,b;" "
70 GO TO 20
```

taster osim na „N“ i BREAK), što može da bude vrlo neprijatno za akcione igre. Jedan od načina da izbegnemo ovu sistematsku poruku je da, na samom početku programa, upotrebimo POKÉ 23692. — 1.

Skrolovanje je vrlo pogodno ako se neki objekt kreće prema vrhu ekrana uz umereno pokretanje po x osi. Jedan od najjedno-

stavnijih načina da isprobate skrolovanje je da štampate upravo definisani karakter u obliku malog svemirskog broda u poslednjem redu, a zatim koristite „prazne“ PRINT naredbe koje će ga pomeriti prema vrhu ekrana. Videćete da je pokretanje broda daleko brže nego u ranijim primerima gde smo koristili PRINT AT. Razlog je u tome što skrolovanje obavlja mašinski program koji je mnogo brži od svakog bajka.

Kada smo se uverili u korisnost skrolovanja, poželećemo da ga upotrebimo i po x dimenziji (sleva na desno ili sdesna nalevo). Na žalost, mašinski potprogrami koji bi ovo omogućili ne nalaze se u ROM-u, a svaki bajzik koji treba da pomeri 8 K video memorije bi bio toliko spor da ga ne treba ni pominjati. Ipak, kao vežbu, pokušajte da napišete bajzik program koji pomeri ekran za jedno mesto ulevo. Ako uspete, zaslužujete čestitku i kompliment da je krajnje vreme da definitivno pređete na mašine.

Ako nemate volje za pisanje mašinskih programa, dajemo vam dva koja će vam lepo poslužiti: prvi omogućava levi, a drugi desni skrol. Kao što se vidi, ova dva programa ukupno imaju samo 46 bajtova, što znači da su toliko optimizovani da se početnici teško mogu upustiti u njihovu analizu: zato i ne dajemo disasembirani listing.

Glavna nevolja sa skrolovanjem je što se ne pokreće samo ono što bismo mi želeli. Na ekranu se, naime, obično nalazi i neki broj bodova, maksimalni skor, abstrakt uputstva za upotrebu ili nešto slično. Sukcesivno pomeranje ekrana izaziva, jasno, i gubljenje ovoga teksta, što je prilično neprijatno. U komercijalnim igrama se često koriste mašinski potprogrami koji skroloju samo deo ekrana ali se, u najvećem broju slučajeva, može pronaći i bez njih: jednostavno koncipirajte sadržaj ekrana tako da su informacije koje stalno treba prikazivati u prvom ili eventualno prvih dva reda, a zatim ih ponovo štampajte svaku puta kada iskoristite PRINT da bi ekran skrolovalo. Na taj način, ujedno, rešavate i problem promene skora u toku igre, a Spectrum je dovoljno brz da se ovo stalno ispisivanje mnogo ne primeti.

## Strogo kontrolisani bajtovi

Izgleda da smo se približili mašinskom jeziku — došli smo do naredbi kojima možemo da pristupamo čitavoj memoriji našeg Spectruma. One su neobično moćne ali imaju i loše strane: dok se do sada operativni sistem našeg računara brinuo o svim greškama i čuvao naš program, PEEK i, posebno, POKÉ su naredbe koje mogu da budu destruktivne. Zato svaki program koji ih koristi treba, pre startovanja, svakako snimiti na traku — to je mera predostrožnosti koja nikada nije suvišna.

Necemo govoriti o sintaksi i funkcijama ovih naredbi (objašnjene su na samom početku „Malih tajni velikih majstora programiranja“ u „Računari- ma 1“), ali moramo da kažemo nekoliko reči o specifičnostima Spectrumove memorijske mape. Treba, naime, da nađemo memorijske prostore u koje možemo slobodno da smestimo sadržaj ne plašeći se da će naš program biti oštećen. U nekim programima česte načini na korišćenje REM linija u tu svrhu što je direktan znak da su autori tih programa, pre Spectruma, imali ZX81. Iako vam ne bismo mnogo preporučili ovaj način, rećemo nekoliko reči o njemu pre nego što pređemo na naredbu CLEAR.

Obrišite program koji se nalazi u raču-

naru i otključajte prvu naredbu i REM AAA a zatim POKE 23760, 127. Listanje programa će otkriti da se prvo slovo A pretvorilo u znak za kopiranje? Pogled na tabelu ASCII karaktere otkriva da je 127 kod tog znaka, što znači da smo pronašli mesto na koje je smešteno slovo A i promenili ga pomoću POKE! Adresa 23760 odgovara prvom karakteru iza REM naredbe (podrazumeva se da ispred REM-a nema drugih naredbi). Ne moramo da se ograničimo na jedan karakter: prva REM linija može da ima mnogo slova i tako posluži kao akumulator za čitav mašinski program ili tabelu podataka.

Zašto smo rekli da ne preporučujemo ovaj metod? Kada jednom počnete da listate program, prve linije će biti pune svakakvih karakterata. Rešenje ovog „estetskog“ problema bi moglo da bude smeštanje mašinske a u poslednje linije programa, ali je njih mnogo teže pronaći u memoriji računara i, što je još gore, njihova početna adresa se menja svaki put kada se program produži ili skrali.

Najjednostavniji način da rezervišete prostor u memoriji je naredba CLEAR za Spectrum 16 Kb CLEAR n će rezervisati prostor od adrese (n+1) do 32599 za mašinski program i podatke; ukoliko je n veće od 32600, neki ili svi karakteri koje ste definisali mogu da budu izgubljeni. Sve ovo se odnosi i na Spectrum 48 Kb, s tim što svi sve cifre veće za 32768. U memorijskim prostor rezervisan na ovaj način ćete, za sada, smeštati samo sadržaje delova ekrana koje treba brzo i dinamično razmenjivati da bi se dobila animacija, ali ćete ubrzo naučiti da je mnogo rentabilnije izbeći korišćenje matrica i nizova kada su svi njihovi elementi između 0 i 255. Takve brojeve je bolje smeštati u rezervisani deo memorije, pri čemu svaki od njih zauzima po jedan bajt. Vreme utrošeno za pisanje programa koji će, po potrebi, preračunavati poziciju nekog bajta i pozivati ga će se višestruko isplatiti — najpre zbog utroška memorije, a zatim, što je još važnije, zbog mnogo bržeg izvršavanja programa.

Razbijanje nizova na bajtove je samo jedna (i to marginalna) upotreba naredbi PEEK i POKE. Mnogo je zanimljivije koristiti ih za promenu sadržaja sistemskih promenljivih. Govorićemo samo o onima koje su od posebnog značaja za akcione igre.

Već smo videli da smeštanje broja različitog od 1 u memorijsku ćeliju 23692 sprečava računar da ispisuje Scroll? kada se ekran popuni. Slično tome, POKE 23609,100 (probajte i druge brojeve umesto 100!) daje kratak ton po pritisku svakog tastera umesto uobičajenog klika, što je zgodno u toku INPUT-a u nekoj igri.

Promena sadržaja memorijske ćelije 23624 utiče na boju kojom se poruke ispisuju na ekranu. Ova memorijska ćelija, zapravo, čuva atribut slova ovih poruka na način koji smo upoznali kod naredbe ATTR. Slično tome, memorijska ćelija 23693 je „odgovorna“ za sadržaj viših redova ekrana i atribute tog sadržaja, ali se koristi dosta teže: pokušajte, na primer, da izvršite POKE 23659,1 a zatim i CLS.

Ćelije 23677 i 23678 su korisne za one koji koriste naredbu DRAW koju je, kao što znamo, „relativna“ — pomeranje se obavlja u zavisnosti od položaja poslednje „pome-

```

9 REM Inicijalizacija
10 GO SUB 5000
19 REM Stanpanje lopte,cuvanje
  starih vrednosti
20 PLOT x,y: LET ax:=LET by:=
30 LET x:=x+d: LET y:=y+d
39 REM Provera polozaja lopte
40 IF x<6 THEN GO SUB 2000
50 IF x<6 OR x>248 THEN GO SUB
  3000
60 IF y>h THEN GO SUB 4000
69 REM Pomeranje reket
70 PLOT p,6: DRAW 20,0
80 LET pep<5 AND INKEY#="9" A
ND p<(235)<5 AND INKEY#="5" AND
  p>0
90 PLOT p,6: DRAW 20,0
99 REM Obrisi loptu
100 PLOT a,b
110 IF true THEN GO TO 20
119 LET dx=<RND#SGN dx#4+SGN dx#
  2
120 LET ball=ball-1: LET s=-1
130 PRINT AT 0,10-LEN STR# s: 0
  VER 0:s
139 REM Da li je kraj?
140 REEP 1,0: IF ball=0 THEN GO
  TO 8000
199 REM Priprema za sledecu
  loptu
200 PLOT p,6: DRAW 20,0
210 GO SUB 5300
220 GO TO 20
299 REM Posodjen reket
2000 LET dx=<RND#SGN dx#4+SGN dx#
  2
2010 LET dy=<RND#4+2
2020 LET true=<x>=p AND x<p+21:
2060 LET ue#7
2070 LET s:=+1: PRINT AT 0,10-LE
  N STR# s: OVER 0:s
2100 BEEP .01,20
2200 RETURN
2999 REM Posodjen zid
3000 LET dx=<RND#SGN dx#4+SGN dx
  #2
3010 LET dy=<RND#SGN dy#4+SGN dy#
  2
3020 LET x=<6+242 AND x>248:
3100 BEEP .01,0
3200 RETURN
3999 REM Posodjen vrh
4000 LET dy=<RND#4-2
4010 LET dx=<RND#SGN dx#4+SGN dx#
  2
4020 PLOT 6,h: DRAW 243,0
4030 LET h=h-(2 AND h>38)
4040 PLOT 6,h: DRAW 243,0
4050 LET y=h-1
4100 BEEP .01,40
4200 RETURN
4999 REM Potprogram za
  inicijalizaciju
5000 LET ball=0: LET s=0
5010 LET h=166
5100 BORDER 4: PAPER 1: INK 7: C
  LS
5300 LET x=127: LET y=h-1
5310 LET dx=<RND#10-5: LET dy=<-R
  ND#4+2
5320 LET p=115
5330 LET true=1
5600 GO SUB 6000
5700 RETURN
5999 REM Crtanje ekrana
6000 OVER 0
6010 PRINT AT 0,1:"SKOR=000
  OSTALO LOPTI 000"
6020 PRINT AT 0,10-LEN STR# s:=:
  PRINT AT 0,31-ball: FLASH 1,"0"
6100 PLOT 0,5: DRAW 0,167: DRAW
  0,167
6150 PLOT p,6: DRAW 20,0
6200 OVER 1
6400 RETURN
7999 REM *** KRAJ IGRE ***
8000 PRINT AT 10,11: FLASH 1:"KR
  AJ IGRE"
8010 BEEP 1,-12: BEEP 2,-24
8020 OVER 0: BORDER 7: INK 9
8030 STOP

```

```

5 REM Prostor za "masinac"
10 CLEAR 31999
11 REM 64999 umesto 31999
  za 48K
20 GO SUB 6000: REM "masinac"
8000 RESTORE 8100: FOR i=3:2000 T
  O 32645: READ a: POKE i,a: NEXT
  i: RETURN
8005
8010 REM SCROLL levo
8019 REM RANDOMIZE USR 32000-
8020 REM 65000 umesto 32000
  za 48K
8040 REM SCROLL desno
8049 REM RANDOMIZE USR 32023-
8050 REM 65023 umesto 32023
  za 48K
8100 DATA 1,255,2,17,0,88,33,1,8
  8,237,176,1,0,24,17,0,64,33,1,64
  ,237,176,201
8110 DATA 1,255,2,17,255,90,33,2
  54,90,237,184,1,0,24,17,255,87,3
  ,254,87,237,184,201

```

SLIKA 11

```

1 REM Za 16K SPECTRUM
2 CLEAR 32592: REM
  (65300 za 48K)
3 LET num chars=1
4 LET bytes reserv=num chars
  #8
5 LET n=32599-bytes reserv:
  REM (65367 za 48K)
6 LET chr before first=CODE
  #"-32
7 LET byte before first=chr
  before first*8
8 LET char=n+1-byte before fi
  rst-256
9 LET chr in 23606=char-256#I
  N(char/256)
10 LET chr in 23607=INT(char/
  256)
20 FOR i=n+1 TO n+bytes reserv
  e: READ a: POKE i,a: NEXT i
30 DATA 255,36,36,24,24,36,36,
  255,0
50 POKE 23606,chr in 23606
60 POKE 23607,chr in 23607
70 PRINT AT 11,15;"#
80 POKE 23606,0
90 POKE 23607,60

```

nute" tačke. Ako, na primer, najpre izvršimo PLOT 100,80 a zatim DRAW 20,60, linija koju dobijemo se neće završavati u tački 20,60 nego u 120,140! Promenom sistemskih promenljivih 23677 i 23678 možemo da simuliramo apsolutni DRAW. U prethodnom primeru bilo bi dovoljno zameniti naredbu DRAW 20,60 naredbom: DRAW 120-PEEK 23677, 140-PEEK 23678 ili, alternativno, zameniti ovo sa dve naredbe, od kojih bi jedna izračunavala a druga crtala, što je način koji će početnici uvek rado prihvatiti.

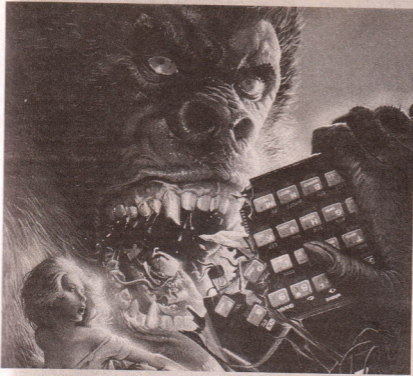
Peščani sat

U mnogim igrama je potrebno meriti vreme koje je igraču bilo potrebno da obavni neki zadatak. Poneko to radi tako što u pravilnim intervalima, dodaje i nekoj promenljivoj, ali se ubrzo pokazuje da je ovakav sat više nego netačan, što izaziva ne malo zaprepašćenje. U čemu je stvar? Vaš Spectrum ne radi neprekidno non stop koji ste mu vi zadali: vrlo često (50 puta u sekundi) on

```

5 REM **** RELI II ****
10 GO TO 3000
8998 REM Definicije karaktera
8999
9000 CLEAR 32399: RESTORE 9200
9020 PRINT AT 11,10: FLASH 1;"SR
CEKATI"
9030 FOR i=32400 TO 32599
9040 READ a: POKE 1,i: NEXT i
9050 FOR i=1 TO 10: FOR n=0 TO 7
9060 READ a: POKE UR CHR# (1+14
3)*n, a: NEXT n: NEXT i
9070 CLS : PRINT AT 9,7."STARTUJ
TRAKU"
9080 LOAD ""
9199 REM Znaci za definisanje
karaktera
9200 DATA 0,24,36,36,126,102,102
,102
9210 DATA 0,124,68,68,126,70,70,
126
9220 DATA 0,124,64,64,96,96,96,1
26
9230 DATA 0,126,66,66,98,98,98,1
26
9240 DATA 0,126,64,64,126,96,96,
126
9250 DATA 255,129,189,165,165,18
9,129,255
9260 DATA 0,126,64,64,96,102,98,
126
9270 DATA 0,68,68,68,124,76,76,6
5
9280 DATA 0,16,16,16,24,24,24,24
9290 DATA 204,204,51,51,204,204,
51,51
9300 DATA 0,0,0,0,0,0,0,0
9310 DATA 0,32,32,48,48,48,48,62
9320 DATA 0,126,86,86,86,86,86,8
6
9330 DATA 0,124,76,76,76,76,76,7
6
9340 DATA 0,126,66,66,98,98,98,1
26
9350 DATA 0,124,68,68,124,64,64,
64
9360 DATA 0,0,0,0,0,0,0,0
9370 DATA 0,124,66,68,126,70,70,
78
9380 DATA 0,124,64,64,124,12,12,
124
9390 DATA 0,126,16,16,24,24,24,2
4
9400 DATA 0,68,68,68,68,68,68,12
4
9410 DATA 0,126,66,2,39,16,16,0
9420 DATA 0,86,86,86,86,86,86,12
6
9430 DATA 0,24,24,0,0,24,24,0
9440 DATA 0,68,68,124,16,16,16,1
6
9499 REM Podaci za "kola"
9500 DATA 0,73,73,93,127,93,63,6
5
9510 DATA 33,66,188,68,60,29,2,4
9520 DATA 254,16,56,63,56,16,254
0
9530 DATA 4,2,29,68,68,188,66,33
0
9540 DATA 65,63,93,127,93,73,73,
8
9550 DATA 32,64,184,68,68,61,66,
132
9560 DATA 127,8,28,252,28,8,127,
8
9570 DATA 132,66,61,68,68,184,64
,32
9599 REM Podaci za brojeve
9600 DATA 0,60,66,66,98,98,98,68
9610 DATA 0,8,8,8,12,12,12,12
9620 DATA 0,124,2,2,68,96,96,62
9630 DATA 0,124,2,2,68,6,6,126
9640 DATA 0,64,96,108,108,126,4,
4
9650 DATA 0,62,64,64,68,6,6,126
9660 DATA 0,62,64,64,124,70,78,1
26
9670 DATA 0,128,9,8,12,12,12,68
9680 DATA 0,68,66,66,60,70,70,68
9690 DATA 0,68,66,66,60,6,6,126
9999 POKE 23686,0: POKE 23687,68
: RETURN
9300 REM Definicije karaktera
5399 REM Sledeca kola
5400 LET x=20: LET y=10: LET dr=
6: LET q=1
5410 PLOT 12,66: DRAW 0,5
5420 OVER 1: GO SUB NORM
5430 POKE 23672,0: POKE 23673,0:
BEEP 1,0
5440 GO TO LOOP
6499 REM Podaci za ekran
6500 DATA 34,10,2,17,36,11,1,5,3
,4,7,12
6510 DATA 1,2,2,5,1,2,6,4,7,2,1,
2,8,2,6,3,2,5,1,2,1
6520 DATA 2,2,5,2,2,5,2,9,2,1,2,
1,3,1,2,2,3,4,2,3,5
6530 DATA 1,2,1,2,1,3,1,2,3,2,4,
4,7,2,1,2,1,3,1,3,2,2,4
6540 DATA 8,3,2,1,2,2,2,2,2,3,2,
4,8,1
6550 DATA 2,3,2,1,3,1,3,2,2,8,4,
1,2,3,2,1,3,2,2
6560 DATA 2,2,3,4,3,2,1,3,1,3,1,3,2,
2,2,2,3
6570 DATA 4,2,3,1,6,2,3,2,2,3,
7,3,4,2,4
6580 DATA 1,3,2,3,6,3,9,4,2,3,3,
2,2,1,2,3,9
6590 DATA 3,3,3,4,2,2,1,2,3,7,4,
3,3,5,2,2,1,2,13,4,5,1,4,6,11,6,
8,34
6597
7000 PLOT 4+e8,56: DRAW 0,5
7010 OVER 0: LET s=INT ((PEEK
23673+256*PEEK 23672)/50)
7020 LET s=s+INT (s/60): LET s=s
-n*60: LET a=8: LET b=6: GO SUB
TIME
7030 GO TO CRASH+FINISH+CRASH A
ND ATTR (x,y)*56)
7497
7498 REM *****CRASH*****
7499
7500 FOR i=0 TO 25: LET d=nd+i-
1: LET d=7: PRINT AT x,y:CHR#
(144+d): BEEP .02,1: NEXT i
7510 GO SUB ALTER: PRINT AT x,y:
b*(1): GO SUB NORM
7520 PRINT AT 16c,1:" ": LET c=
c-1
7530 IF c>0 THEN GO TO NEXT CAR
7540 GO TO NEW GAME
7597
7598 REM *****FINISH*****
7599
8000 PRINT AT x,y:CHR# (144+d):
LET f=60*8:
8010 GO SUB ALTER
8020 INPUT "
8030 PRINT AT 1,0,0:"JIJIJIJKKRA
VD,-USPELI STEK JIJIJIJIJ": PRUSE 2
00
8040 INPUT "
8050 IF t<h THEN LET h=t: LET a=
1: LET b=7: GO SUB TIME: GO SUB
ALTER: PRINT AT 1,0,0:"fffffki T
0 JE NAJBOLJE VREME:fffff": GO TO
8070
8050 PRINT AT 1,0,0:"ALI VREME
NIJE NAJBOLJE"
8070 PAUSE 200
8080 PRINT AT x,y:"J"
8090 GO SUB NORM
8097
8098 REM *****NOVA IGRA*****
8099
9000 INPUT "JOS JEDNOM (D/N)": L
INE z#
9010 IF (z#=" "X1)=N THEN GO TO
BYE
9020 INPUT "pritisci ENTER ": LI
NE z#
9030 GO TO START
9900 BORDER 1: PAPER 1: INK 7: C
LS
9990 PRINT AT 10,13: FLASH 1: IN
K 6: PAPER 2:"ZBRAVOOO"
9999 STOP

```



dobija takozvani interapt koji mu nalaže da pređe na izvršavanje jednog programa u ROM-u koji će, pre nego što računar nastavi sa radom, obaviti neku vrstu „generalnog spremanja“. Bilo kakav vremenska petlja koju stavite u program će, dakle, biti često prekidana i taj broj prekida nećete moći lako da izračunate.

Zato je za merenje vremena najbolje iskoristiti sistemski časovnik koji računar uvećava za jedan u toku svakog interapta (to je jedna od komponenti pomenutog „generalnog spremanja“). Za upotrebu ovog časovnika treba upoznati memorijske ćelije čije su adrese 23674, 23673 i 23672. Svake pedesetinke sekunde sadržaj ćelije 23672 biva uvećan za jedan. Kada se on poveća na 255, računaru povećava sadržaj ćelije 23673 dok se nula smešta u 23672. Slična operacija se ponavlja i kada sadržaj 23673 postane 255, samo što se tada sadržaj 23674 povećava za jedan. Sasvim je, prema tome, prosto izračunato vreme u sekundama od kada su sva tri ova broja bila nula (možete ih sami postaviti na nulu pomoću POKE na početku igre): LET sekundi = (65536 \* PEEK 23674 + 256 \* PEEK 23673 + PEEK 23672) / 50.

Na žalost, Spectrumov sat je daleko od savršenstva. Pre svega, vreme biva izgubljeno svaki put kada se izvrši BEEP (od sada je ponovno inicijalizovanje nekih sistemskih promenljivih posle BEEP bilo od koristi), ali kada se koristi neka periferijska jedinica (npr. štampač). Pored toga, ponekad izračunato vreme može da bude netačno: možda se sadržaj 23674 nije bio promenio dok je računar izvršavao prvo PEEK. Zato se u on povećan sadržaj 23673 anuliran pre nego što je izvršeno drugo PEEK. Zato se u uputstvu za upotrebu savetuje da se uvek izračunaju dva broja sekundi, pa da se veći uzima za čvan.

## Tenk na reliju

Jeste li znali da možete da redefinišete više od 21 karaktera (čak do 96) i tako povećate raznovrsnost vaše flote napadača iz svemira? Evo kako se to radi na najlakši mogući način:

Najpre iscrtajte sve nove karaktere i izračunate dekadne vrednosti njihovih linija.

— Odlučite koji originalni karakteri treba da budu predefinisani i na koji način.

— Upotrebite CLEAR da rezervišete prostor za definicije. Za svaki karakter koji treba definisati rezervišete osam bajta

— Prebrojte koliko ima karaktera pre onoga koji treba redefinisati (ako, na primer, redefinišete znak „\*“ koji ima kod 42, naći ćete da pre njega ima 42—32=10 karaktera). Zatim pomnožite taj broj sa 8 i nazovite rezultat m

izvršite naredbe:  
POKE 23606, m-256 INT (m/256)  
POKE 23607, INT (m/256)

— Okončajte definiciju karaktera na uobičajeni način.

Sve će ovo postati mnogo jasnije kada pogledate primer koji predefiniše karakter u grafični blok.

Gotovo sve što smo do sada prešli sadržano je u igri „Reli“. Treba, naravno, voziti automobil po kružnoj stazi. Automobil, doduše, više liči na tenk, ali je to zato što je potrebno da se na neki način vidi kuda se krećemo iz oblika pokretnog predmeta; tenk je za to neobično pogodan zbog „cevi“. Ukoliko vam se ovaj oblik ne sviđa, ili ste miroljubivo raspoloženi, možete da ga promenite bez ikakvih problema.

Program koristi IN (naredbu opisanu na početku teksta) da omogućiti pritisak dva tastera u isto vreme, na primer taster za ubrzanje i promenu pravca. Automobil, naime, ima četiri brizge. Pritisak na O vas ubacuje u sledeću, a pritisak na A u prethodnu brzinu. Automobil se kreće u svim

pravcima; taster P je namenjen skretanju u pravcu kazaljke na časovniku, a O skretanju u obrnutom pravcu.

U igri je korišćeno merenje vremena, ali se vreme prikazuje tek kada nastradate ili kada obidete čitav krug — igra bi inače bila prepora. Za ispunjenje zadatka imate tri automobila; svaki put kada stradate, računaru vas vraća na početak, što je dodatno otežanje.

Glavna karakteristika igre je predefinisani set karaktera; praktično su sva slova i brojevi predefinisani uz korišćenje ideja iz prethodnog poglavlja. To je, na žalost, učinio program predugačkim, pa je morao da bude podeljen u dva dela: prvi definiše karaktere a zatim učitava drugi koji predstavlja pravu igru. Ovakva podela programa, jasno, ne odgovara vlasnicima Spectruma 48 Kb koji mogu, uz malo veštine, ujediniti programe u jednu celinu.

Ukoliko kucate program iz dva dela, otkucajte najpre prvi snimite ga sa SAVE „Reli“ LINE 10, verifikujte snimak i zatim otkucajte drugi deo programa i snimite ga sa SAVE „C“ line 10

## Sedam zlatnih pravila

Ovim smo se približili kraju ove škole akcionih igara na Spectrumu. Ipak, pre nego što završimo poslednji čas i pozovemo vas da sami prionete na posao, nekoliko saveta koji će, nadamo se, pomoći da vaše buduće vido-igre budu dinamičnije.

Što više humora u komentarima koje računar ispisuje. Neka korisnik programa vidi da igru ne treba shvatiti baš previše ozbiljno. Ipak, treba paziti da sve šale izazivaju simpatiju prema kompjuteru.

Ukoliko se pojavljuje situacija u kojoj je kompjuteru potrebno duže vreme da se pripremi za sledeće akcije (npr. da „zamisliti“ lavirant na početku igre), pokušajte da stvorite utisak da kompjuter razmišlja. Neka se, na primer, na ekranu s vremenom na vreme smenjuju komentari u stilu „Samo što nisam spreman“, „Hmmm... nije baš lako“ i slično. Isti komentari nipošto ne smeju da budu isti pred svaku partiju, jer će postati dosadni. Pripremite više poruka nego što je potrebno, pa ih slučajno birajte pomoću RND.

Uvek koristite karaktere koje ste sami definisali umesto zvezdica ili nečeg sličnog. Sigurno sata da igrači neće ni malo smetati što je bomba velika koliko i avion ili Messc koliko i Sunce ako su istaknuti bitni detalji i ako se na prvi pogled vidi šta se dešava.

Ekspertizite što više sa bojama; koristite čak slučajne boje koje, ipak, moraju da budu kontrolisane u izvesnoj meri (ako PAPER i INK budu isto obojeni, neće se videti šta se dešava). Promena boje pozadine neverovatno menja utisak o prizoru, a „košta“ samo jednu kratku naredbu.

Eksplozije su neobično efektni i ne treba štedeti trud na njihovoj realizaciji. Treba, samo, paziti da eksplozije ne traju predugo jer one najčešće blokiraju sve druge događaje.

Pošto ste završili igru, utrošite još pola sata da smislite nešto što će biti originalno. Predložili bismo da uvedete neki događaj koji je izuzetno malo verovatno i dešava se na primer, jednom u 200 partija. Za korisnika će biti izuzetno prijatno iznenađenje da otkrije da igra koja mu je možda skoro dosadila ima još malih tajni.

Iako ste tek završili prvi deo škole akcionih igara na bejziku, razmislite o prelasku na mašinski jezik!

Dejan Ristanović

# Servis računara

## „Galaktičari“ u nevolji

Pre nego što vas obuzme panika, proverite još jednom štampane veze od video izlaza preko integralnih kola 74LS38, 74LS00, 74LS123, CD 4017, CD 4040 do 74LS93 i 74LS04. Uključite ponovo računar i pokušajte da na ekranu registrujete bar neki signal. Možda će to biti slika bez ikakvih znakova (slova ili brojeva), ali sa karakterističnim rasterom. Ako se i sada na ekranu ništa ne pojavljuje, ostaje vam mogućnost „hladnog lema“ (postoji vizuelni ali ne i električni kontakt) na jednom od gore pomenutih integriranih kola. Zagrejte lemnicu i osvežite lemove na ovim kolima. Ako i to ne pomogne, biće vam potrebna logička sonda da među ovim integralcima pronađete krivca.

Lociranje i otklanjanje greške je mnogo jednostavnije ako se na ekranu pojavi bar nekakva slika. Za brzo lociranje kvara pomoći će vam tabela 1, sa pregledom integriranih kola u čijoj okolini treba tražiti kvar.

### Pročitajte još jednom

Ako još uvek niste dobili tastaturu ili kompletni ras ve delove, pročitajte sledeće savete pre početka rada.

• dok ne „oživate“ računar, dovoljno je da ugradite samo jednu memoriju 6116; kada računar proradi, lako ćete ugraditi i ostale, a ako ste u nečemu drastično pogrešili, bar ćete sačuvati skupu memoriju.

• koristite RF modulator firme ASTEC UM 1111 E 36; ovaj se modulator napaja sa 12 volti, a rasposed nožica je isti kao na štampanoj ploči; R16 zamenite kratkospojnikom, a otpornik R15 izostavite; za druge tipove modulatora potrebno je eksperimentalno odrediti vrednosti ovih otpornika.

• vaš računar može, u nevolji da radi i bez kondenzatora za blokadu Cn; ako ih još niste nabavili, probajte da „oživate“ svoj računar i bez njih.

• posebnu pažnju obratite lemljenju najveći broj kvarova potiče upravo odatle; prilikom lemljenja integriranih kola, napravite pauzu posle svake naočice dok se kolo ne ohladi, kada se na njemu pojave nečistoće, vrh lemlice čistite vlažnom krpom.

• masku za tastaturu pre upotrebe treba malo obraditi; ivice svih otvora za tastere malo sastružite turpijom i prvo postavite tastare na uglovnima, a zatim po jedan u sredini svakog reda.

Ako ste pažljivo sklupali svoj računar i ako su vam sve komponente ispravne, onda ćete prilikom prvog uključivanja imati sliku na ekranu sa dugo željenim READY. Međutim, može se desiti da je slika iskrivljena i

### SIMTOM

SIMTOM	MOGUĆI UZROK
1. Ekran bez teksta i rastera	80A, 6116, 2732, 74LS04, 74LS93, 74LS74, 74LS123
2. Raster postoji ali nema teksta	2716, 74LS04, 74LS74, 74LS156
3. Slika jako iskošena	C3, C4, R12, R13
4. Ne dokuđuju se određeni karakteri	74LS156, 74LS251
5. Karakteri se ispisuju ali nedostaju nek njihovi delovi ili su deformisani	280A, 2732, 6116, 74LS174
6. Parafalno se ispisuju prvi i peći red istim karakterima	74LS174
7. Svi se karakteri ispisuju ali neki delovi nedostaju	74LS166
8. Slova presečena po vertikalni a na BRK se javljaju crte na desnoj strani	74LS174
9. Nevezani karakteri na raznim delovima ekrana	280A, 74LS32, 74LS156
10. Pifinčani raster po celom ekranu	280A, 4LS156
11. Promenljivi raster po celom ekranu	280A, 74LS156
12. Dijagonalni raster sa jednom vertikalnom štrafom	280A, 74LS156
13. Nevezani tekst, ne prima komande	74LS38, 4LS74
14. Vertikalne šare bez teksta	74LS04, 74LS38
15. Crte po celom ekranu	74LS38
16. Pored READY povremeno se javlja nevezani tekst	74LS38
17. READY na desnoj polovini ekrana zatim karakteri po celom ekranu	280A, 74LS74
18. Slika se kreće po vertikali	74LS04, 74LS123
19. Slika jako iskošena i kreće se po vertikali	C3, R12, 74LS123
20. Kosi tačkasti karakteri	74LS123
21. Par belih kosih crta	74LS123
22. Vertikalno postavijeni ili iskošeni karakteri	74LS93, CD4040
23. Vertikalno kretanje slike bez teksta	74LS93
24. Na PRINT MEM pokazuje manje memorije	6116, 74LS156

da se slova pojavljuju u više iskošenih redova. Pokušajte prvo da povećate kondenzator C3 na oko 10 nF do 30 nF i otpornik R12 na 1 do 2 koma. Slika će početi da se ispravila. Početak reda možete pomerati smanjivanjem kondenzatora C4 do 50 nF i R1 do 15 koma.

### Neopetni karakteri

Umirili ste sliku i sada proveravate da li se svako slovo koje otkucate pojavljuje na ekranu. Ako neki taster ne reaguje, odmah proverite njegovog suseda sa vertikalne i horizontalne linije na shemi. Ako ne reaguje čitav red, proverite veze ka kolima 74LS251 i 74LS156. Možda je ta linija u prekldu, ili nožica koja nije dobro zalemljena. Posle ove intervencije, tastatura će reagovati, ili je neispravno kolo ka kome vodi linija koja ne reaguje na komande sa tastature.

Na ekranu se mogu pojaviti i nepotpuni karakteri (recimo, slovo A bez horizontalne crte po sredini). Grešku treba tražiti na nožicama 4 i 5 kola 74LS174 ili na nožicama 2, 3, 4, 5, 10, 11, 12, i 14 kola 74LS166. U na njihovom putu ka epromu 2716. U slučaju da su slova dvostruka, proverite

nožice 6 i 7 kola 74LS174, kao i kola ka kojima ona vode. Loše zalemljene nožice 2 i 3 ovog kola daju deformisane i neprepoznatljive karaktere, a pritiskom na taster BRK na desnoj strani ekrana pojavljuju se vertikalne linije. Ako su karakteri presečeni po horizontalni, proverite nožice 10 i 11, a ako se karakteri ispisuju istovremeno u prvom i petom redu, odgovore su nožice 14 i 15 kola 74LS174.

Neispravno kolo 74LS32 ili 74LS156 (u adresnom dekodoru) prouzrokuje sliku pirinčanog rastera ili će se na raznim delovima ekrana pojaviti nevezani karakteri koji mogu biti i nestabilni. Ako se pojavljuje dijagonalni raster sa jednom vertikalnom štrafom, nožica 15 kola 74LS156 ima kratk spoj sa masom. Isti se simptomi javljaju i kad je neispravan mikroprocesor 280A.

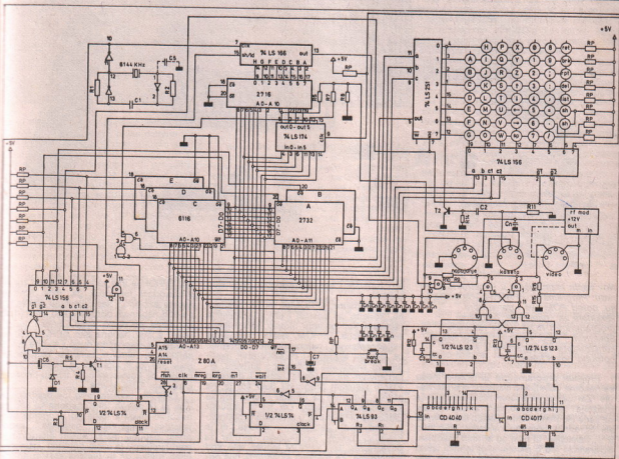
### Nevezani tekst

Nevezani karakteri, ali sa ispisanim READY na početku prvog reda, javljaju se ako nožica 11 kola 74LS38 ima spoj sa masom. Ovaj deo kola, kada je ono neispravno, prouzrokuje vertikalne šare po ekranu ili nevezani tekst. Potpuno jednaku



Najzad je sve gotovo! Posle svih muka sa nabavkom delova, čekanjem tastature i čipova, vaš računar je, najzad, sklopljen. Uključili ste ispaljač, povezali kablove i počeli da pretražujete frekventne opsege na svom prijemniku, a na ekranu se ne pojavljuje ama baš ništa. Baš nemate sreće! Ne treba, međutim, gubiti glavu — to nije najgora stvar koja je mogla da vam se desi. Do sada je, osim toga, proradilo oko dve hiljade „galaksija“, mnoge i u znatno neveselijim rukama nego što su vaše. Zašto biste baš vi bili izuzetak?

Računar „galaksija“ odlikuje visoka pouzdanost u radu — do sada se redakciji javilo samo nekoliko čitalaca, u svakom slučaju manje od deset koji nisu mogli sami da izadu na kraj sa svojim računarom, tako da praktično nemamo nikakvih serviserskih iskustava. Zato smo jedan od redakcijskih primeraka stavili „na muke“, izazivajući na njemu, naravno po određenom sistemu, sve kvarove koji su mogli da nam padnu na pamet. Računar je ovu torturu glatko preživeo, a rezultati su sumirani u tabeli i u ovom tekstu. Ako vam ni on ne pomogne da se izvučete iz nevolje, javite se redakciji. Vaš računar, budite sigurni u to, neće ostati samo gomila skupih čipova.



**Električna shema računara „galaksija“:**  
Na shemi nema nikakvih izmena — jedino su obeleženi izvodi na integrisanim kolima, što znatno olakšava traženje za varovima

se manifestuje prekid signala za osvežavanje memorije (koji inače ovdje ima sasvim drugu ulogu), nožica 28 mikroprocesora, ako je neispravan Z80A, nožice 3 i 4 kola 74LS04 ili nožica 13 kola 74LS74.

Može se desiti da, neposredno posle uključivanja računara, na desnom delu ekrana dobijete READY, a da se odmah zatim pojave nevezani karakteri po celom ekranu. Grešku potražite na nožicama 2, 3 ili 5 kola 74LS74 ili na nožicama 20, 24 ili 27 mikroprocesora.

Vertikalno kretanje slike izaziva neka nepravilnost u lancu delitelja učestanosti kristala — kola 74LS93, CD4040, CD 4017 ili sam monostabilni multivibrator 74LS123. Tačne učestanosti signala na ovim kolima ne možete izmeriti bez specijalnih instrumenata (osciloskopa ili frekvencmetra), pa vam jedino preostaje da kola zamenite novim.

Kad računar potpuno proradi, ugradite preostale memorije 6116 i otkucajte PRIN MEM. Ako računar pokaže da nemate onoliko memorije koliko ste ugradili integrisanih kola, proverite nožice 4 i 5 kola 74LS156 ili promenite mesta memorijama i opet istestirajte kapacitet memorije.

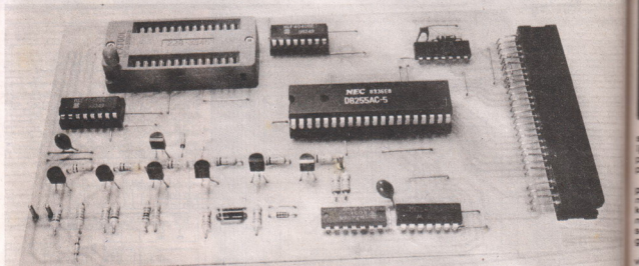
Sve navedene kvarove mnogo lakše ćete otkloniti ako imate mogućnosti da koristite direktan ideo ulaz u svoj televizor, ili vam je na raspolaganju monitor. U tom slučaju izbegavajte greške koje unosi RF modulator.

Tekst: Miša Hadži Đorđević

Crtež: Mira Todorović

# programator eproma

računari  
u domaćoj radinosti



Nov način razmišljanja u konstruktorskom radu: Programator EPROM-a, na kome su, pored ROM-ova i generatora karaktera za „galaksiju“, uspešno programirani i ROM-ovi za računare Spectrum i BBC, može da uvede mlade konstruktore i programere u krug potpuno novih ideja

Projekat programatora EPROM-a može se nekomе učiniti kao svojevrsno preterivanje. Zar zaista ne postoji ništa zanimljivije i važnije? Ne! Naravno, za one koji ne žele da sede skrštenih ruku i čekaju šta će im za samogradnju ponuditi najesei „Računari 3“, „Galaksija“ ili neki od stranih časopisa. Stvaralacki rad u računarskoj tehnici teško je ostvariv bez jedne ovakve naprave. Mikroprocesor je potpuno bespomoćan bez uputstava za rad koja dobija od ROM-a, a ROM možete imati jedino ako ga sami napravite. Uredaji za povezivanje računara sa periferijskom opremom, interfejsi za procesno upravljanje, višekanalni generatori tonova, uređaji za sintezu glasa i sve ostale računarske sitnice koje život znače rade znatno jednostavnije ako je operativni program u nekom ROM-u nego ako se učitava sa kasete. I ne samo to! Kakva je korist od neke nove rutine ili naredbe koja povećava snagu računara ako ona prilikom svake upotrebe mora iznova da se učitava sa trake? Šta tek da se radi kada se u biblioteci programa nakupi tridesetak novih nared-

bi i funkcija i stotinak korisnih mašinskih rutina?

Programator EPROM-a može, očigledno, korisno da posluži i onim ljubiteljima računara koji nemaju nimalo konstruktorskih ambicija. EPROM, zaista, predstavlja najjednostavniji, najpouzdaniji, najtrajniji, pa, sasvim sigurno, i najelegantniji medijum za smeštanje sistemskog softvera, alternativnih programskih jezika i izuzetno važnih uslužnih programa (kao što je, na primer, program za obradu teksta) koji se svakodnevno koriste. On se u računaru uključuje ili direktno, u predviđena podnožja, kao kod „galaksije“ i BBC-ja, ili je dovoljan sasvim jednostavan hardver za povezivanje, tzv. EPROM loader, kao kod Spectruma. Za njegovo aktiviranje nije potreban nikakav dodatni softver (BBC) ili je dovoljno nekoliko prostih mašinskih rutina (Spectrum). Pristup programu je, povrh svega, trenutan i, što je najvažnije, on se nalazi u računaru čim ga uključite ili čim, određenom naredbom, pozovete potreban program.

## Kako radi

Rad sa programima iz EPROM-a je komforniji čak i onda kada imate disk-jediniću! Na ovom svetu, međutim, ništa nije savršeno, pa nisu ni EPROM-i: brisanje i upisivanje podataka u njih se ne vrši u računaru nego u specijalnim uređajima, kao što je ovaj čiju shemu objavujemo, a čitava procedura, naročito brisanje, može da potraje i nekoliko časova. Pošto se programira jedanput a koristi bezbroj puta, ovaj nedosta-

tak EPROM-a ima samo akademski značaj. Proizvođači memorija ga, ipak, shvataju veoma ozbiljno i već najavljuju čipove koji se pod naponom ponašaju kao RAM (može se, dakle, po njima pisati i brisati do milion volje) a bez napona kao EPROM (čuvali podatke beskonačno dugo i to one koji se zateknu u njima u trenutku isključenja). Ipak, sasvim je sigurno da naredne dve-tri godine na komercijalnom tržištu pripadaju klasičnim EPROM-ima.

EPROM-i, sigurni smo, ne predstavljaju više ni iz daleka takvu zagonetku kao pre nekoliko meseci kada su, u vreme prvih tekstova o računaru „galaksija“, izazvali pravu uzburku među našim čitaocima. U međuvremenu je svaki graditelj „galaksije“ imao priliku da ih upotrebom sasvim dobro uperi se da se ne radi o nimalo nežnim čipovima. Malo teorije, ipak, neće biti ne odmet.

EPROM predstavlja skraćenicu engleskog izraza „Electrically Programmable Read Only Memory“ i označava memoriju koja se programira električnim putem i čije se podaci mogu samo čitati. Procedura programiranja EPROM-a je gotovo bašta jednostavna: svaki bajt se upisuje pod „visokim naponom“ TTL impulsom koji traje 50 milisekundi. Ovaj „visoki napon“ kreće se između 25 V (kod 2716) i 13 V (kod 27256). Programiranje ne mora da teče sukcesivno — može se programirati bit koji memorijska lokacija u bilo koje vreme — pojedinačno jedna za drugom ili ne preskok. Jedanput programiran, EPROM će sačuvati upisani sadržaj najmanje deset godina — ako se ne izloži sunčevoj svetlo-

Računar „galaksija“, bez sumnje, predstavlja veliki podsticaj i za programera i za sagraditelja računara. On razobličuje kućni računar kao složen tehnički proizvod i ohrabruje ljubitelje računara da i sami ugrade poneku ciglu u tu građevinu. Ozbiljniji programerski i konstruktorski rad na razvoju i, naročito, „doradama“ računara (najomiljenija sportska disciplina domaćih hakera) ne može se, međutim, ni zamisliti bez alatke koja se naziva programator EPROM-a i koja omogućuje jednostavnu produkciju ROM-ova. Zato smo za treću sagradnju posle „galaksije“ pripremili projekat univerzalnog programatora EPROM-a (adaptacija i dorada sheme objavljene u nemačkom časopisu MC) za računare „galaksija“ i ZX Spectrum. Zašto baš za njih? Zato što su to dva najpopularnija računara na domaćem tržištu i zato što u njima ima najviše prostora za dorade. Uređaj programira najznačajnije savremene tipove EPROM-a — 2516, 2716, 2732, 2732A, 2764 i 27128 — i radi pod punom programskom kontrolom korisnika. Pomoću njega se može iskoprirati ROM bilo kog kućnog računara — od „galaksije“ do BBC-ja — ili isprogramirati alternativni ROM sa nekim komercijalnim ili svojim sopstvenim programom.

## Lična karta EPROM-a

tip	(kapacitet kilobajta)	potrošnja mirno/akt.	brzina odziva
2716	2 Kb	25/100 mA	350 ns
2732A	4 Kb	35/125 mA	200 ns
2764	8 Kb	40/100 mA	200 ns
27128	16 Kb	45/150 mA	200 ns

\* Intel

sti ili nekom još intenzivnijem izvoru (ultraljubičastih zraka. Broj ciklusa pisanje (impulsom) — brisanje (ultraljubičasti zraci) praktično je neograničen.

I pored sve jednostavnosti, procedura programiranja i brisanje EPROM-a može manje iskusnim konstruktorima izgledati krajnje opasno: šta se dešava kada se kroz nežnu strukturu čipa propusti 25 V ili kada se ona izloži razornim UV zracima? Očigledno, upravo ono što je potrebno!

Memorijsku ćeliju u EPROM-u predstavlja MOS tranzistor sa dva gejta: jedan je slobodan za prijem signala, a drugi obložen košuljicom silicijum-dioksida; pošto na njega ništa nije vezano, najčešće se naziva plivajući gejti. Silicijum-dioksid je izvanredan izolator — elektroni ne mogu ni da zadu napolje iz košuljice na plivajućem gejtu niti da uđu unutra. Kada se, međutim, na drejn dovede visoki napon a na slobodni gejti nizak, tako da potencijalna razlika između njih iznosi 20 do 25 V, elektroni dobijaju dovoljno energije da probiju barijeru silicijum-dioksida i naelektrisu plivajući gejti. U ovoj „košuljici“ ostaju zarobljeni praktično zauvek — u svakom slučaju najmanje deset godina. Ćelija sa naelektrisanim plivajućim gejtom ponaša se kao prazna memorijska lokacija.

Ultraljubičasto zračenje slabi izolaciona svojstva silicijum-dioksida. Ako je dovoljno intenzivno, „košuljica“ će se toliko „razmekšati“ da će elektroni „iscuriti“ kao da ih nikada nije ni bilo — EPROM će biti obrisani i spreman za upis novog sadržaja. Ćelija sa razelektrisanim plivajućim gejtom ponaša se kao puna memorijska ćelija. Zato se u praznom EPROM-u u svakoj ćeliji nalazi logičko „1“, odnosno na svakoj adresnoj lokaciji broj 255 (heksadekadno FF). Prilikom programiranja u EPROM se, dakle, upisuju samo logičke nule, a prilikom brisanja te nule se vraćaju na jedinice.

## Familije

Na tržištu se, trenutno, mogu naći dve nekompatibilne familije EPROM-a: jedna nosi oznaku 27XXX i dolazi iz proizvodnih hala čuvenog Intela, a druga 25XX i pripada Texas Instrumentsu. Kao i sa mnogim drugim stvarima, Texas Instruments ne ovide nije imao sreće: Intelove standarde i Intelovu familiju, koja, povrh svega, ima i nešto više članova, prihvatilo je znatno više japanskih i evropskih proizvođača memorija i TI je i ostao gotovo usamljen. Jedan tip EPROM-a (2516) kompatibilan je, doduše, sa Intelovim 2716, ali su zato TMS 2532 i TMS 2564, sa drugačijim rasporedom pinova i nešto drugačijom procedurom programiranja, ostali sasvim izvan „serije“. Njihova uloga se, izgleda, svela na to da plaše kupce visokom cenom i do besvesti kompliciraju univerzalne uređaje za programiranje EPROM-a!

Naš programator EPROM-a urađen je prema Intelovim standardima. On može da programira sve članove Intelove familije — 2716, 2732, 2732A, 2764 i 27128 — osim najnovijeg 27256, jer se on pojavio kada je uređaj već bio završen. Iako, u principu, ne bi bilo naročiti teškoća da se uređaj opredeli da programira i EPROM-e iz familije 25XX, ova pogodnost ipak nije ugrađena, jer bi programator previše iskomplikovao i softverski i hardverski. Pošto su čipovi po Intelovom standardu jeftiniji i, uz to, pristupačniji od EPROM-a TI, ovakva komplikacija zaista nije bila potrebna. Za one koji ne mogu bez familije 25XX, dajemo uporednu tabelu sa rasporedom izvoda (sl. 1) i procedurom programiranja (sl. 2) za EPROM-e iz obe familije, iz njih se može videti kako se on „galaksijom“ programator EPROM-a mogu, uz malo veštine i nekoliko hardverskih rezova, programirati i TI čipovi.

Svaki EPROM ima četiri kontrolne funkcije — CS = izbor čipa, OE = dozvola izlaza, Vpp = programski napon i PGM = programski impuls — i, u zavisnosti od logičkih stanja na kontrolnim ulazima, pet različitih modova rada: mirovanje (stand-by), čitanje, programiranje, provera i zadržavanje (inhibit). Procedura za mirovanje, čitanje i proveru je krajnje jednostavna i,

reklo bi se, spontana, i otuda izbor koncepcije programatora treba da vodi računa prvenstveno o tome kako da se na najbolji način zadovolje kriterijumi za proceduru programiranja. (Zadržka se koristi samo kod istovremene programiranja više EPROMa različitim sadržajem.)

Procedura programiranja ne bi, takođe, bila vredna pažnje da upis jednog bajta, sa stanovišta mikroprocesora, ne traje čitavu večnost: mikroprocesor drži adresu i podatak koji joj pripada stabilnim svega nekoliko stotina nanosekundi i odmah prelazi na sledeću, dok je EPROM-u za prihvatanje podatka potrebno čitavih 50 mikrosekundi. Zato okosnicu svakog programatora čini sklop koji „zaustavlja“ adresu i podatak dok se ovaj ne upiše u EPROM i, razume se, obezbeđuje programski impuls u trajanju od 50 ms. Klasična koncepcija programatora EPROM-a zasniva se na tzv. osmootbitnim „lečevima“ (latch) — primitivnim memorijska koje su u stanju da „upamte“ jedan jedini bajt. Njihov zadatak je da „zamrznu“ adresu i podatak u trajanju od 50 ms. Uz izlazne bufere, adresni dekoder koji nešto malo hardvera, ovakva rešenja obavezno kras i vešpolni preklopnik za izbor tipa EPROM-a.

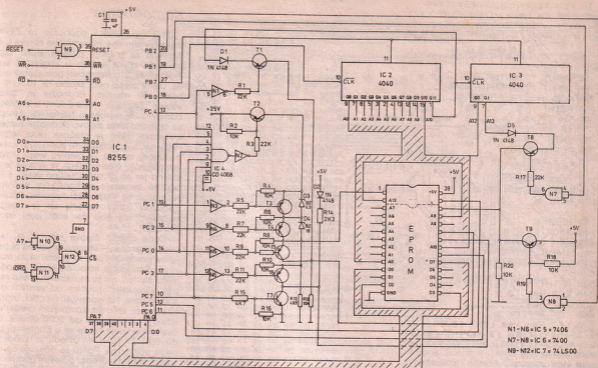
## PIO 8255

Sva svojstva koja se očekuju od ovakvih sklopova objedinjena su u jednoj posebnoj vrsti čipa poznatoj pod imenom PIO (Parallel Input Output Port — grubo: paralelna ulazno-izlazna kaplja). Ova vrsta čipa je specijalno projektovana za dvosmernu paralelnu komunikaciju mikroprocesora sa periferijskim uređajima, pri čemu se njihove funkcije određuje isključivo programskom kontrolom. Programator EPROM-a, bez sumnje, predstavlja najjednostavniji uređaj koji pomoću PIO čipa može da se spregne sa mikroprocesorom. Zašto se ta pogodnost ne bi iskoristila da se samo sa nekoliko čipova napravi uređaj profesionalnih mogućnosti i — karakteristika?

Svaka porodica mikroprocesora mora, nezostavno, da ima i svoj PIO čip. On se u Zilogovoj familiji Z80 zove, jednostavno, Z80 PIO. Mada je namenjen za spregu sa računarsima izgrađenim oko procesora Z80, u ovom programatoru EPROM-a je i ipak upotrebljen Intelov čip iz familije 8085 sa oznakom 8255AC-5. Zašto? Pored toga što je njegova i mašinska i programska sprema čak i sa „tuđim“ procesorima jednostavnija od sprege Z80 PIO sa svojim sopstvenim procesorom, Intelov PIO čip ima čitav jedan port više, što u ovoj praksi predstavlja presudnu prednost. Da budemo do kraja jasni: Z80 PIO se ne može koristiti u ovoj shemi bez ozbiljnijih hardverskih izmena jer nema dovoljno izlaza da se ostvare sve potrebne funkcije!

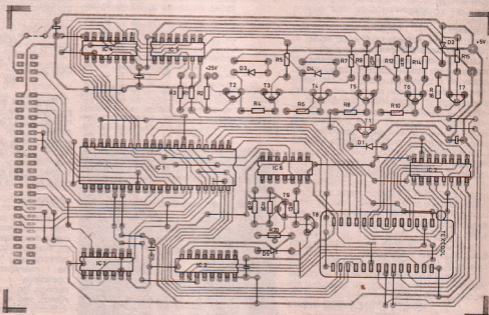
Sprega PIO čipa 8255 sa mikroprocesorom





Savršeno jedinstvo  
 mašinskog i programskog  
 dela (hardvera i softvera  
 Električna shema  
 programatora EPROM-a  
 uređaja:

Montažna shema: Kod  
 složenijih uređaja ne  
 jednostrukoj štampi  
 kratkospojnici  
 predstavljaju uobičajenu  
 sliku; ima ih i ovde  
 podosta, ali ne toliko da  
 usporu samogradnju ili  
 pokvare estetski izgled  
 programatora; prilikom  
 programiranja čipovi sa  
 24 nožice postavljaju se  
 na podnožje kao što je  
 prikazano isprekidanom  
 linijom na električnoj  
 shemi.



#### KOMPONENTE

##### Integrirana kola

IC 1	8255AC-5
IC 2, IC 3	CD 4040
IC 4	CD 4068
IC 5	SN 7406
IC 6	SN 7400
IC 7	SN 74LS00

##### Tranzistori

T1-T6, T8	BC 177
T7	BC 197
T9	BC 287

##### Diode

D1, D2, D5	1N 4148
D3	BZ 4, 3
D4	BZ 15

##### Otpornici

R1, 3, 5, 7, 9	11, 17	22 koma
R2, 4, 6, 8, 10	13, 16, 18, 20	10 koma
R14	2k2	
R12, 15	4k7	
R19	680 koma	

##### Kondenzatori

C1-C4	100 nF (blek)
	-1 uF (tantal)

##### Podnožja

ZIF 28 nožica (Textool)	1
40 nožica	1
16 nožica	1
14 nožica	2

##### Konektor

2 x 22 razmak 3.95 mm (galaksija)	1
2 x razmak 2.5 mm (Spectrum)	1

```

10 REM *****HEXLOADER*****
20 DEF FN h(x)=CHR$(INT (a/1
6)+48+7*(a/159)+CHR$(a-16)*INT
(a/16)+48+7*(a-16)*INT (a/16)>>9
))
30 DEF FN h(x)=CODE h(x)-48-7*(
h(x)/9)
40 INPUT "STARTNA ADRESA:";:x
45 INPUT "ZAVRSNA ADRESA:";:f
50 FOR n=0 TO f STEP 8
60 PRINT n;"; ";
65 INPUT a;
70 IF a#="S" THEN STOP
75 PRINT a;
80 FOR b=0 TO 7
90 LET z=FN h(a#)#16+FN h(a#&2
))
100 POKE (n+b),z
110 LET a#=#a&(4 TO 2)
120 NEXT b

```

```

130 NEXT n
140 STOP
499 REM *****HEXLIST*****
500 INPUT "STARTNA ADRESA:";:X
510 INPUT "ZAVRSNA ADRESA:";:Y
520 CLS
530 PRINT X;"; ";
540 FOR Am1 TO B
550 LET Am=" ";
560 LET B=PEEK X
570 LET Am#CHR$(INT (B/16)+48+
7*(B/159))+CHR$(B-16)*INT (B/16)
+48+7*(B-16)*INT (B/16)>>9))
580 PRINT Am;"; ";
590 LET X=X+1
600 IF Y=X THEN STOP
610 NEXT A
620 PRINT
630 GO TO 530

```

**Program 1**  
Program za unošenje mašinskog koda (tzv. hex-loader): Bajtovi se ne unose jedan po jedan, kako je to uobičajeno, nego red po red ENTER, dakle, treba pritisnuti tek na kraju reda

```

10 CLEAR 24999 :LORD "ERROR:"
CODE 25000,724
200 PRINT AT 3,6;"PROGRAMATOR: E
PROGRAM"
30 PRINT AT 11,10;"1 = 2710"
40 PRINT AT 12,10;"2 = 2720"
50 PRINT AT 13,10;"3 = 2730"
60 PRINT AT 14,10;"4 = 2764"
70 PRINT AT 15,10;"5 = 27120"
80 INPUT "TIP ERRORA";:B
90 IF B#0 OR B#5 THEN PRINT AT 21,8;"NETRONI PODRUKU";:GO TO
200
100 POKE 23300,B
110 CLS :PRINT AT 11,7;"1 = PR
OGRAMERINJE"
120 PRINT AT 12,7;"2 = KOPIRIRAJ
E"
130 PRINT AT 13,7;"3 = LISTIRAJ
E"
140 PRINT AT 15,7;"4 = KRAJ"
150 INPUT C
160 IF C#4 THEN PRINT AT 21,9;"
KRAJ"
170 CLS :INPUT "POCETNA ADRESA
U RAMU";:U
180 POKE 23297,U-256*(INT (U/256)
)
190 IF C#1 THEN GO TO IPR 25150
200 IF C#2 THEN GO TO IPR 25155
210 IF C#3 THEN GO TO 230
220 LET #="(123456789ABCDEF"
230 LET #0#INT (PEEK (U/16)) :LET
#PEEK (U+16*INT (PEEK (U/16)
))
240 PRINT U;TAB 7;#(C#1);#(C#2)
)
250 LET U#U+1
260 GO TO 230

```

**Program 2**  
Program u bejziku ispisuje meni i prosledjuje parametre mašinskom programu — ovo rešenje je prihvaćeno da bi se izbeglo pisanje novog od

25000	CD	EC	62	01	C2	63	CD	34
25008	63	CD	58	63	0D	7E	01	D3
25014	3F	08	1F	FE	FF	20	64	CD
25020	80	62	20	F5	0D	7E	00	D3
25026	3F	CD	18	63	3E	98	03	7F
25030	CD	29	63	0D	7E	02	03	7F
25036	7E	FF	28	13	33	1F	0D	00
25042	7E	05	03	3F	F3	3E	32	0E
25048	8D	0D	20	3F	30	20	F8	FB
25054	0D	7E	02	03	3F	CD	80	62
25060	20	DE	CD	18	63	CD	24	63
25066	CD	29	63	0D	7E	01	03	3F
25072	1F	4E	88	20	25	CD	60	34
25078	62	20	F9	91	95	64	CD	34
25112	63	CD	58	63	CD	5F	63	81
25120	14	00	C9	81	D3	63	CD	34
25128	63	CD	58	63	CD	5F	63	81
25136	14	00	C9	81	D3	63	CD	34
25144	63	CD	58	63	CD	5F	63	81
25152	14	00	C9	81	D3	63	CD	34
25160	63	CD	64	63	CD	58	63	0D
25168	7E	01	D3	3F	08	1F	77	CD

25176	20	35	20	36	33	20	43	
25184	44	20	35	46	20	36	33	20
25192	30	31	20	36	33	20	35	31
25200	32	30	31	34	20	30	30	
25208	20	43	39	20	31	20	44	
25216	33	20	36	33	20	43	44	
25224	33	34	20	20	32	35	31	
25232	32	30	31	36	33	20	43	
25240	20	35	20	36	33	20	43	
25248	44	20	35	46	20	36	33	20
25256	30	31	20	36	33	20	35	31
25264	33	36	31	34	20	30	30	
25272	20	43	39	20	31	20	31	
25280	44	20	36	34	20	43	44	
25288	33	34	20	36	33	20	43	
25296	34	31	36	33	20	43	44	
25304	20	35	20	36	33	20	43	
25312	44	20	35	46	20	36	33	20
25320	30	31	20	36	33	20	35	31
25328	35	32	31	34	20	30	30	
25336	20	43	39	20	43	44	20	45
25344	43	20	36	32	20	30	31	20

25352	33	30	20	20	32	35	33	
25360	33	36	30	32	30	34	33	
25368	20	33	39	20	32	30	24	
25376	33	20	34	34	20	32	30	24
25384	34	35	20	36	33	20	35	33
25392	34	34	34	33	20	32	30	
25400	20	33	36	20	33	32	30	
25408	30	33	20	33	20	33	31	20
25416	32	30	20	20	20	00	16	
25424	00	00	CD	68	01	E1	69	
25432	01	96	00	CD	63	01	59	81
25440	30	64	CD	34	63	81	59	64
25448	CD	34	63	CD	40	63	CD	50
25456	63	C9	41	81	07	47	84	04
25464	05	90	90	5E	1E	84	04	05
25472	50	90	50	1E	84	04	05	70
25480	30	34	14	84	04	05	70	80
25488	34	14	82	82	16	09	80	80
25496	48	48	50	49	52	41	4E	48
25504	45	08	16	09	08	48	4F	50
25512	49	52	41	4E	48	4F	50	48
25520	4F	54	4E	56	4F	16	9C	08

25528	43	45	48	53	55	40	20	3D
25536	20	08	16	09	09	50	52	4F
25544	47	52	41	40	49	52	41	4E
25552	48	45	08	16	07	06	12	01
25560	45	50	52	4F	40	20	4E	49
25568	48	45	20	50	52	41	50	54
25576	45	21	08	16	09	04	53	54
25584	41	56	49	20	45	50	52	4F
25592	40	20	55	20	50	4F	44	09
25600	4F	50	48	45	08	16	09	04
25608	50	52	4F	47	52	41	40	50
25616	52	41	4E	48	45	20	55	49
25624	50	45	4C	4F	16	07	03	3
25632	12	01	50	52	4F	47	52	41
25640	40	49	52	41	4E	48	45	20
25648	4E	49	48	45	20	55	53	50
25656	45	4C	4F	21	08	16	09	04
25664	49	50	56	41	44	20	20	45
25672	50	52	4F	40	20	45	50	20
25680	40	4F	44	4E	50	52	4F	41
25688	08	16	09	02	49	20	50	52
25696	49	54	49	53	4E	40	20	42

25704	49	4C	4F	20	48	4F	4R	49
25712	20	54	41	53	54	05	52	12
25720	01	20	08	00	00	00	00	00

## Komponente

PIO čip 8255 i podnožje za EPROM predstavljaju dve najobilnije stavke u troškova za gradnju programatora. Čip je do kraja aprila koštao u Engleskoj svega 3—4 funta, a u Nemačkoj petnaestak maraka, da bi preko nekoliko nedelja (19 funti!) i preko 30 maraka. Podnožje je, takođe, veoma "papreno" oko 8 funti u Engleskoj, odnosno 35 maraka u Nemačkoj. Unapred se odrecite ideje da ovo podnožje zamenite klasičnim, iako je stotina puta jeftinije. Radi se o specijalnom podnožju koje se proizvodi upravo za ovakvu vrstu uređaja. Postavljanje čipa u njega je veoma jednostavno jer su otvori dovoljno veliki, a kontakt se ostvaruje mehaničkom kočnicom. Na tržištu se može naći nekoliko tipova sa približno istom cenom, ali je ubedljivo najbolji onaj koji na sebi nosi oznaku „Textool“.

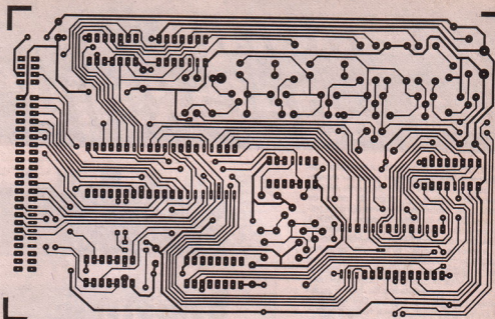
Cene ostalih čipova niksu ni iz daleka tako šokantne. Ukipan iznos ne bi trebalo da prede 35 funti sa poštanskim troškovima. Štampano kolo, na kome takođe ne treba sedeti, može se nabaviti preko redakcije „Galaksije“ po ceni od *verovatno oko* 1000 dinara. Na ispravljak će otići, otprilike, ista suma, tako da kompletan uređaj ne bi smeo da košta više od 9000 dinara. Cene ovakvih uređaja kreće se u Engleskoj između 60 i 70 funti — ušteda je, dakle, više nego očigledna. Pošto se očekuje da će se tržište smiriti u drugoj polovini godine i cene krenuti blago nizbrod — veruje se da se one više nikada neće vratiti na staru meru — čitava kalkulacija može da se pokaže i znatno povoljnijom.

- OUT 0, 255 — uključuje sve izlaze na portu A (255 = 11111111)
- OUT 1, 0 — isključuje sve izlaze na portu B (0 = 00000000)
- OUT 2, 129 — uključuje prvi i poslednji izlaz na portu C (129 = 100000001)

Portovi, a suština i jeste u tome, neće promeniti stanje sve dok mikroprocesor novom OUT instrukcijom ne uputi novi zahtev i novi — podatak. Port, drugim rečima, može da ostane „zamrznut“ beskonačno dugo, a da to nimalo ne omete mikroprocesor — umesto da čeka da periferijski uređaj završi posao i vrli se u mrtvoj petlji, on može da radi na drugim zadacima.

## Adrese

U našem projektu za povezivanje PIO čipa i aktiviranje portova primenjene su nešto drugačije adrese. PIO čipu je, razume se, svedejno koja ga adresa poziva (koje će adrese linije biti vezane na CS, AO i A1), ali ne i mikroprocesoru, pa ni računarnu „galaksiju“ i Spectrum, iz jednostavnosti razloga što ta adresa ne sme biti u memoriji, već se mora nalaziti u tzv. ulazno-izlaznoj (I/O) adresnoj mapi. Iz te mape se uzimaju adrese za rad sa svim periferijskim uređajima, pa se mora uzeti i adresa za programator EPROM-a. Procesor Z80 u ovom mapi ima samo 256 adresa (AO—A7), a toliko ih ima i računar „galaksija“. Spectrum, doduše, i ovdje ima 65536 adresa, ali je najbolje koristiti adrese linije AS, A6 i A7, jer ih sam računar ni za šta ne koristi. Ove adrese linije, dakle, odgovaraju u svakom slučaju. Kod Spectruma valja zadvojiti još jedan uslov: adrese linije AO—A7 moraju biti na visokom nivou da bi se



zaštittile neke već postojeće adrese u I/O mapi.

Za prozivanje PIO čipa odabrana je adresna linija A7 i to za uslov kada se nalazi na niskom nivou. Ona, sama za sebe, međutim nije dovoljna. PIO čip sme da se aktivira samo kad se mikroprocesor obraća svojoj I/O mapi. On to čini jedino kada dobije IN ili OUT instrukciju. Da bi periferni uređaj uopšte znao da mu se obraća, mikroprocesor mora u tom času da spusti nivo na svom izlazu IORQ (=tebi kažem). Jednostavan sklop sa NI kapijama omogućuje mikroprocesoru da prozove PIO čip samo ako su A7 i IORQ u isto vreme na niskom logičkom nivou. Ova situacija je moguća u 128 slučajeva (sve adrese od 0 do 127), ali to nije ni od kakvog značaja, jer u tom času ne radi nijedan drugi periferni uređaj i ne može da dođe ni do kakve zabune. Apsolutno prozivanje PIO čipa, jedinom jedinstvenom adresom, samo bi nepotrebno iskomplikovalo uređaj. Na kontrolne ulaze A0 i A1 vezane su adrese A6 i A5 (A0 = A6, A1 = A5 — logičnije bi bilo obrnuto ali ovo malo pojednostavljuje štampu). Pošto sa prozivanjem bilo kog porta mora istovremeno da bude prozvan i PIO čip, u adresu mora da ude i taj uslov.

A5	A4	A3	A2	A1	A0	dekadno adresa
0	0	1	1	1	1	31 PORTA
0	0	1	1	1	1	63 PORT C
0	0	1	1	1	1	93 PORT B
0	1	1	1	1	1	127 kontrolni registar

Pošto između programatora EPROM-a i mikroprocesora u računaru ne postoji nikakva interakcija („handshaking“ — signali kojima mikroprocesor i PIO čip razmenjuju informacije da li su završili određenu operaciju ili ne), PIO čip radi u modalitetu 0. Preko porta A prenose se podaci koji se upisuju u EPROM ili čitaju iz njega, preko porta C impulsi i naponski navoji koji brinu o proceduri programiranja, verifikacije i čitanja za različite tipove EPROM-a, a preko porta B impulsi za resetovanje i okidanje programskog brojača, kao i za elektronski

preklopnik koji na nožicu 24 dovodi ili napajanje (do 2764) ili adresnu liniju A13 (27128). Očigledno je da portovi B i C uvek rade u izlaznom, a port A prilikom programiranja u izlaznom, a prilikom čitanja u ulaznom modalitetu. Potrebna su, dakle, samo dva kontrolna bajta:

128 — modalitet 0, port A izlaz, port B izlaz, port C izlaz

144 — modalitet 0, port A ulaz, port B izlaz, port C izlaz.

Adresiranje memorijskih lokacija u EPROM-u prilikom čitanja i programiranja izvedeno je, takođe, na krajnje neobičan način. Bilo bi logično da se taj posao poveri mikroprocesoru — adresiranje bi se tada

odvijalo automatski. Tada bi, međutim, bila potrebna još dva „leća“ za privremeno „zamrzavanje“ adresa. Štampano kolo bi se toliko iskomplikovalo da bi, najverovatnije, bila potrebna dostruka štampa, a programiranje postalo znatno složenije. Zato je u programator ugrađen poseban programski brojač sa dva čipa 4040 (brojač sa modulom 16 — deli sa 4096). Ovakva koncepcija je elegantna i u mašinskom i u programskom smislu, ali ima i jednu slabu tačku: ako se EPROM ne programira sukcesivno, od početka do kraja, nego na preskok, treba dopisati petlje za okidanje brojača u prazno vreme kada se prekače neka adresa.

Hardverska koncepcija programatora EPROM-a omogućava potpunu programsku kontrolu svake funkcije prilikom čitanja, testiranja, programiranja i verifikacije EPROM-a i, samim tim, ostavlja korisniku potpuno određene ruke da odredi funkcionalnu koncepciju i prilagodi uređaj svojim videnjima i svojim potrebama. Opisano hardverskom rešenju nedostaje, možda, samo jedna stvar: sopstveni ROM sa programom za rad uređaja koji bi se „aktivirao“ čim se programator priključi na računaru. Time bi se izbeglo dosadno učitavanje programa sa trake svaki put kada treba isprogramirati neki EPROM, ali bi se programator ipak nepotrebno iskomplikovao. Rad sa programatorom je, inače, u svemu ostalom veoma komforan: nakon startovanja programa dovoljno je uneti tip EPROM-a i početnu adresu u RAM-u. Sve ostalo se dešava gotovo samo od sebe.

Programator EPROM-a, mada mu cena nije mala, spada u onu vrstu uređaja koje se isplaćuje praktično čim proradi i po tome je u kompjuterskom svetu potpuno jedinstven. Razlike u ceni programa na kaseti i u EPROM-u su toliko drastične (prazan EPROM 27128 košta oko 25, a napunjen i do 60 funti), da se programator ponekad isplaćuje samo jednim programiranjem! Ovakvi uređaji, međutim, imaju i jednu mnogo dragocenniju vrednost: oni u programerskom i konstruktorskom radu otvaraju vrata za sasvim nove ideje.

Jova Regasek

**Ovo nije sve!  
U „RAČUNARIMA 3“  
SOFTVERSKA KONCEPCIJA I  
PRAKTIČNA REALIZACIJA PLUS**

• **Kompletan program za računaru „galaksija“ sa sledecim opcijama**

- **testiranje eproma i ...**
- **parcijalno testiranje između dve adrese**
- **kopiranje EPROM-a u RAM i ...**
- **... parcijalno kopiranje između dve adrese**
- **programiranje EPROM-a iz RAM-a i ...**
- **... parcijalno programiranje između dve adrese**
- **verifikacija i ...**
- **... verifikacija između dve adrese**

• **usavršeni program za računaru Spectrum sa opcijama za testiranje, kopiranje, programiranje i verifikaciju bit po bit**

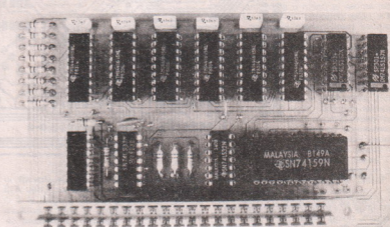
# memorija „Galaksija za „galaksiju“

„Galaksija  
48 K“

Da bi jedan računar funkcionisao kao skladna celina, svi njegovi sklopovi moraju da budu primereni jedni drugima. Moćan operativni sistem ne može da dođe do izražaja sa malom radnom memorijom (računaru QL nije, izgleda, dovoljno ni 128 K!), niti će velika memorija biti dobro upošljena sa skromnim operativnim sistemom. Kolika je memorija, onda, primerena računaru „galaksija“? Skeptici bi rekli da je on već sada ima dovoljno, ali u to ih može razuveriti svaki vlasnik „galaksije“ koji je pokušao da na svom računaru napiše barem jedan ozbiljniji program u bejziku. Oni koji tvrde da će se „galaksija“ sa 54 K radne memorije „osećati kao u tuđem odelu koje joj je, uz to, za tri broja veće“ verovatno su nešto više u pravu. Učitavanje programa u toliku memoriju, slažemo se, može da potraje čitavu večnost. Nije, međutim, sve ni u brzini učitavanja čak i ako ona ne bi mogla da se poboljša.

Obilna memorija možda izgleda nezapрно uz računaru „galaksija“, ali ona u mnogome može da nadoknadi skromnije osobine operativnog sistema. Domaći ljubitelji računara, koji silom prilika nabavljaju mašine koje su znatno ispod njihovih potreba, u doradama operativnog sistema često vide jedini izlaz. Operativni sistem „galaksije“ je savršeno otvoren za ovakve dorade — računaru nedostaje samo malo više memorije. Veća memorija, pored toga, predstavlja osnovni uslov za hardverska poboljšanja računara, kao što je, na primer, finija grafika i, naročito, za programski zamah. Ko će se prihvatiti da napiše „ozbiljan“ program za „galaksiju“ kad se sa 6 K radne memorije ne mogu pisati tekvi programi?

Iako trenutno potrebe, po pravilu, ne mogu da budu pouzdano merilo za veličinu memorije jednog računara — nju će diktirati razvoj i hardvera i softvera koji za sada niko ne može da predvidi — svaki vlasnik „galaksije“ će, očigledno, morati sam da odredi svoju meru. Uz one kilobajte koji već postoje u računaru, 16 K, sasvim sigurno, predstavljaju najrazumniju količinu. Ako vreme pokazuje (verovatno i hoće, ali ne tako brzo) da je i to nedovoljno, memorija se koristi da bude jednostavno, uz dodatak samo dva ili četiri čipa, proširiti za još 16 ili 32 K. Od potrošačke groznice niko nikada nije imao koristi, pa nećete ni vi i nemojte se plašiti da kasnije nećete naći potrebne čipove — doći ćete do njih sigurno, i to, verovatno, znatno jeftinije. Realno, 32 K predstavlja gornju razumnu granicu.



„Galaksija 48“: Memorijsko proširenje povećava radnu memoriju računara „galaksija“ u zavisnosti od broja ugrađenih čipova, za 16, 32 ili 48 K; zahvaljujući fleksibilnom adresnom dekoderu, proširenje se može pomerati po memorijskoj mapi u skokovima od po 4 K

## Statičke ili dinamičke?

Ako je, u trenutku kada je koncipiran računar „galaksija“ i moglo da bude dilema oko tipa čipa koji će se koristiti u radnoj memoriji, sada nedoumici nema ni najmanje mesta. Statičke memorije, koje se ugrađuju u „galaksiju“ toliko su poskupele na svetskom tržištu da ih preporučujemo samo u izuzetnim prilikama i — mikroskim količinama. Prema trenutnom paritetu, dinamički kilobajt je gotovo osam (!) puta jeftiniji od statičkog, što praktično znači da se za „galaksijinih“ 6K u osnovnoj verziji može dobiti gotovo 48 K u „dinamičkim“ bajtovima! Na izbor statičkih memorija za osnovnu verziju „galaksije“ najviše je uticao bojazan da dinamičke memorije mogu konstruktorima da zaboravaju život više nego što je to nužno. Kakva je razlika između statičkih i dinamičkih memorija i da li je zabreda konstruktora „galaksije“ bila opravdana?

Memorije čelija u statičkoj memoriji izgrađena je na bazi tzv. flip-flopa — dosta složenog elektronskog sklopa koji se ostvaruje sa pet-šest tranzistora. Jednom upisan, sadržaj ostaje u ovakvoj čeliji sve dok se preko njega ne upiše novi ili dok se memorija ne isključi iz struje. Svi sklopovi potrebni za rad sa ovom vrstom memorije ugrađeni su u sam čip, pa je procedura upisivanja i čitanja podataka potpuno spontana. Statičke memorije su, međutim, veoma glomazne — iole značajnija memorijska konfiguracija zahtevala bi mnoštvo čipova i ogromno štampano kolo. Proizvođačima je do sada pošlo za rukom da u jedan čip

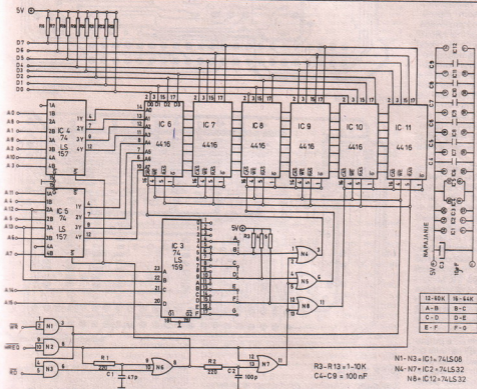
smeste najviše 8192x8 čelija (osam kilobajta), ali takav čip košta 50 dolara!

Kod dinamičkih memorija čelija je izgrađena na bazi kondenzatora. Pošto zauzima manji prostor od čelije sa tranzistorima, kondenzatori obezbeđuju znatno veću gustinu pakovanja i — manju cenu po bajtu. Mikrometerske dimenzije, međutim, povlače da sobom i „mikrometerski“ kapacitet — pet do deset stotih delova pikofarada! — pa se čelija brzo prazni i njen sadržaj se nepovratno gubi. Najbolji tipovi dinamičkih memorija mogu da pamte podatke najviše četiri milisekunde. Zato se njihov sadržaj mora stalno obnavljati — to se postiže takozvanim „osvežavanjem“. Osvežavanje dinamičkih memorija, koje uteruje strah u kosti neobaveštenim ljubiteljima računara, predstavlja trivijalan proces koji procesor Z80 obavlja automatski na kraju svakog mašinskog ciklusa. Osvežavanje se postiže prozivanjem memorijskih čelija „na prazno“.

Druga specifičnost dinamičkih memorija, međutim, nije toliko bezazlena i zahteva i nešto malo dodatnog hardvera i određenu proceduru za upisivanje i isčitavanje podataka. Memorijske čelije u dinamičkoj memoriji su organizovane u redove i kolone, ali je direktan pristup omogućen samo redovima. Zato se adresa ne saopštava memoriji odjednom nego — kao i kod ostalih procesora — u dvaput. U memoriji se najpre upiše adresa za redove a potom — u istom mašinskom ciklusu! — za kolone. Čelija tek tada postaje spremna da primi ili oda svoj sadržaj. Nešto malo komplikovanija procedura rada i zavisnost



**Radna memorija, bez sumnje, predstavlja jedno od osnovnih merila snage jednog računara. Bez obzira na snagu i kapacitet njihovog operativnog sistema, u savremenim kućnim računarima se za memorisanje, po pravilu, koristi sav prostor koji dopušta mikroprocesor. Takva koncepcija se, razume se, nije mogla očekivati i od računara „galaksija“, ali nju mogu (i, rekli bismo, moraju) da sprovedu svi oni koji nameravaju da naš računar koriste iole ozbiljnije. Iako su za hobbiste, zbog osudnice na svetskom tržištu čipova, došli veoma teški dani, u redakciji smo razvili memorijsko proširenje „galaksija 48“, koje „galaksiju“ značajno približava „ozbiljnom“ računarskom društvu. Proširenje omogućava povećavanje memorije u blokovima od po 16 K, a može se priključiti na verzije računara sa 2 i 6 K.**



### KOMPONENTE

**Integrirana kola**  
 TMS4416 2 4 ili 6  
 74LS157 2  
 74LS08 1  
 74LS32 1 ili 2  
 74159 1

**OTPORNICI**  
 R1, R2 220 oma  
 R3-R13 1-10 koma

**Kondenzatori**  
 C1 56 pF  
 C2 100 pF  
 C3-C7 100 nF  
 C8 10-47 uF (tantal)

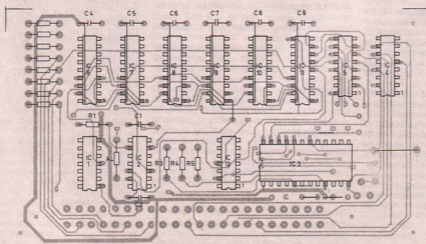
**Predzočja**  
 16 nožica 2 4 ili 6  
 14 nožica 2  
 16 nožica 2 ili 3  
 24 nožica 1

**Konektor**  
 S x 22 razmak 1  
 3,95 mm

12-80K	18-64K
A-B	B-C
C-D	D-E
E-F	F-G

N1-N3 = IC1 = 74LS08  
 N4-N7 = IC2 = 74LS32  
 N8 = IC12 = 74LS32

**Električna shema proširenja galaksija 48. Memorija je podeljena u tri nezavisna bloka od po 16 K**



**Montažna shema proširenja „galaksija 48“: lako je primenjena dvostruka štampa, nekoliko kratkospojnika se nije moglo izbeći; kratkospojnici koji prolaze ispod memorijskih čipova služe da skrate vezu kondenzatora za blokadu sa masom, čime se povećava njihova efikasnost u „peglanju“ napajanja za dinamičke čipove; kratkospojnici ispod i iznad čipa 74159 služe za programiranje adresnog dekodera — na slici je prikazana situacija kada se u „galaksiji“ nalazi 6 K RAM-a (kada su ugrađena dva tri memorijska čipa)**

od spoljašnjeg hardvera predstavlja, praktično, jedinu komplikaciju u radu sa dinamičkim memorijama. Ova komplikacija je, međutim, prilično trivijalna i nije vredna pažnje ni sa stanovišta cene, ni sa stanovišta složenosti, ni sa stanovišta pouzdanosti rada uređaja. Dinamičke memorije, uz to, rade doslovce u milionima primeraka kućnih računara — nema nikakvog razloga zašto ne bi mogle i u „galaksiji“

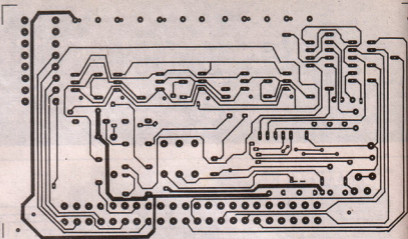
## I sve i nešto

Na tržištu postoji četiri tipa dinamičkih memorija. Prvi među njima, 4416, organizacija 16384 bita  $\times$  1, predstavlja najjeftiniji i, možda, najlogičniji izbor. Osam čipova, koliko je potrebno za 16 K, koštaju svega oko 2000 dinara. Ove memorije, na žalost, zahtevaju trostruko napajanje, a za veće kapacitete strahovito složeno štampano kolo i mnoštvo — čipova! Po svojoj konfiguraciji i tehničkim rešenjima, ove memorije predstavljaju prevaziđeni tip. Memorije dvostruko većeg kapaciteta, 32768  $\times$  1, koje se ugrađuju u računar ZX Spectrum, mogu da budu znatno privlačniji izbor. Ovaj tip memorije — dugo je trebalo da se to sazna! — ne proizvodi se nigde u svetu. Radi se o memorijama 4164 (65536  $\times$  1) kojima nije ispravna jedna sekcija!

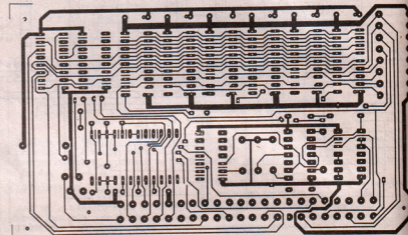
Tako, praktično, za izbor ostaju samo dva tipa. Osam čipova tipa 4164, za četrdesetak funti, obezbeđuju memoriju kapaciteta 64 K! lako se u „galaksiji“, kao ni u jednom računaru, ne može direktno iskoristiti čitav adresni prostor (nešto se mora ostaviti i za ROM, mada RAM, pod određenim uslovima, može da radi i „ispod“ ROM-a), ova konfiguracija predstavlja najjeftinije „rešenje“. Memorija jednostavno pokriva sve prazne prostore u memorijskoj mapi, uključujući i prostor za ROM 2, i uspešno zamenjuje jedan ili dva statička RAM-a (D i E). Oni vlasnici „galaksije“ koji još nisu nabavili ove čipove mogu, tako, za isti novac da vrate gotovo pola memorijskog proširenja.

Za ljubitelje potpunih rešenja — a potpuna rešenja su uvek i elegantnija i jeftinija od polovičnih — ova varijanta predstavlja jedini moguć izbor. U redakciji je razvijen prototip ovog rešenja koji se odlično pokazao i koji će dobro poslužiti za razvoj sistemskog softvera, ali smo se ipak odlučili za drugu varijantu. Zašto? Ovo rešenje ima jedan dosta ozbiljan nedostatak filozofsko-finansijske prirode — ono pred vlasnike „galaksije“ postavlja dilemu „sve ili ništa“. Plašimo se da bi većina ljubitelja računara pre bila prisiljena da izabere „ništa“ nego „sve“. Za njih bi to bila „supa za ždrala“ — lepo ali skupo i nepristupačno rešenje.

Četvrti tip memorije, 4416, organizacije 16384  $\times$  4 bita, možda nije toliko fleksibilan u popunjavanju rupa u memorijskoj mapi, ali objedinjuje prednost potpunih i polovičnih rešenja. Memorija se, takođe, može proširiti do punog formata, ali se to ne mora uraditi odjednom — moguće je i u triput u blokovima od po 16 K. Jedan ovakav blok, koga čine dva čipa 4416, nije, čak ni danas, skuplji od jednog jedinog čipa 6116! Ovaj tip memorije je relativno nov i malo se teže nalazi na tržištu — drže ga,



gornja strana



donja strana

*Štampano kolo: lako objavljujemo crteže obe strane štampanog kola u razmeri 1:1, preporučujemo vlasnicima „galaksije“ da se stripe do septembra i sačekaju fabričku verziju*

praktično, jedino u Engleskoj — ali tako neće ostati zadugo.

Memorijski čip 4416 ima osam adresnih ulaza, AO—A7, i četiri ulaza za podatke, DO—D3, što, praktično, znači da jedan čip može da pamti samo pola bajta (četiri bita). Zato se za svaku polovinu bajta koristi po jedan čip. Preko kontrolnih ulaza WE (upisivanje/čitavanje), RAS (upisivanje za redove i osvežavanje), CAS (upisivanje adrese za kolone i izbor čipa) i G (dovozila izlaza) upravlja se procedurom upisivanja i iščitavanja podataka. Čip može da radi u nekoliko različitih modaliteta i veoma složenim memorijskim shemama. Mi smo, razume se, odabrali najjednostavniji način rada.

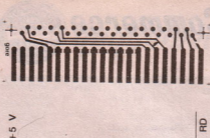
## „Dinamička“ procedura

Električna shema memorije „galaksija 48“ izvedena je klasično za ovakvu vrstu sklopova, bez ikakvih ekstravaganantnih zahvata i komplikovanih vremenskih koha. Adrese za redove i kolone obezbeđuju dva

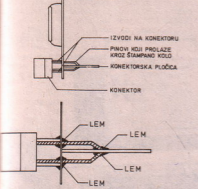
multipleksera tipa 74LS157. U zavisnosti od stanja na kontrolnom ulazu S, multiplekseri na izlaze Y postavljaju ili adresu koju dobija preko ulaza A (A8—A13) ili adresu koju dobija preko ulaza B (AO—A7). Na početku ciklusa upisivanja ili čitanja multiplekseri postavljaju na adresne ulaze AO—A7 memorijskog čipa adresu za redove, a stotina nanosekundi kasnije na ulaze A1—A6 adresu za kolone. Adresa za redove upisuje se u memorijski čip silaznom ivicom impulsa RAS (row address select — izbor adrese za redove), a adresa za kolone silaznom ivicom impulsa CAS (column address select — izbor adrese za kolone). Stotina nanosekundi nakon kompletiranja adrese aktiviraju se izlazni baferi i podatak može da se pročita.

Procedurom upisivanja adrese upravlja se signalom MREQ preko nekoliko I/Li kapija sa odgovarajućim kolima za kašnjenje. Na adresnim ulazima memorijskog čipa stalno se nalazi adresa za redove i oni se pozivaju svaki put kad se mikroprocesor signalom MREQ obraća memoriji — bez obzira da li želi da upiše podatak, da ga pročita ili da osveži memorijske ćelije. Ako želi da upiše ili da pročita, WR signal će pripremiti memorijski čip, a RD ili WR otvoriti I/Li kapiju N6. Pošto je već upisao u čip adresu za redove, MREQ će, nakon

Konektor i konektorska pločica: Sandvič-sistem montaže konektora omogućuje da se na računaru „galaksija“ istovremeno sa memorijskim proširenjem priključi još neki uređaj, na primer programator EPROM-a ili generator zvuka, raspored izvoda na konektorskoj pločici na proširenju dopunjen je novim signalima (RD, CS1, CS2 i CS3 i +5V)



Konektorska pločica na računaru „galaksija“: Da bi se omogućilo jednostavnije povezivanje periferijskih uređaja preko porta za proširenje, konektorska pločica mora da pretrpi izvesne izmene; sve izmene se vrše sa gornje strane pločice, a za njih su dovoljni skalpel, lemljica i dva parčeta žice



zvesnog kašnjenja (R1, C1, N6) „prekopčan“ multiplexer i dovodi na A1—A6 adresu za kolone. Nakon što se stabilizuje ova adresa, na izlazu kola za kašnjenje oko N7 (R2, C2) će se pojaviti signal CAS i upisati je u memorijski čip. Prilikom upisivanja podataka čipu se istovremeno sa adresom za kolone saopštava i podatak (tzv. early write modalitet — upisivanje impulsom CAS). Prilikom čitanja podatak se pojavljuje stotinak nanosekundi nakon upisivanja adrese za kolone.

Signal CAS, istovremeno, upravlja i izlazno izlaznim baferima — podatak se ne može ni upisati ni pročitati ako ulaz kojim on upravlja nije na niskom nivou. To, praktično, znači, da se on može koristiti i za izbor određenog memorijskog bloka. Adresni dekodler 74159 određuje prema kom će bloku ovaj signal biti usmeren. Dekodler pokriva čitav memorijski prostor u blokovima od po 4 K. Pošto se radi o modelu sa tzv. otvorenim kolektorima, izlazi se mogu napajati paralelno bez ikakvog ograničenja. Za blok od 16 K moraju se spojiti četiri izlaza. Shema proširenja „galaksija 48“ dopunjava dve mogućnosti — nadovezivanje na postojeća 2 K radne memorije (spojiti tačke A—B, C—D i E—F) i 6 K (B—C, D—E—F—G). Šta da uradi oni koji su u „galaksiji“ ugradili srednju varijantu od 4K? Moraju da dokupe još jednu statičku memoriju 8196 ili da, presecajući voda CS koji vodi prema memoriји D, isključe ovaj memorijski čip. U prvom slučaju memorija se može proširiti do 64K, a u drugom „samo“ do 32 K.

Upotrebljeni dekodler se nekomе može ubiti previše glomazan za ovakvu primenu. To je, međutim, bio jedini način da se postigne izvesna fleksibilnost sistema. Od toga, osim toga, mogu imati koristi čak i oni koji ne nameravaju da proširuju memoriju do kraja — na konektor su izvučeni signali CS za poslednjih 12 odnosno 16 K. Oni se elegantno, bez potrebe za novim dekodrom, mogu iskoristiti u nekom drugom proširenju — na primer proširenju

ROM-a u kombinacijama 1 × 16 K, 2 × 8 K ili 4 × 4 K! Oni koji se unapred odriču poslednjih 16 K radne memorije, treba na štampanom kolu da preseku veze između nožica 14 i 15 i 15 i 16 i time razdvoje CS signale u poslednjih 16 K adresnog prostora.

### Štampano kolo...

Štampano kolo je, zbog dosta složene strukture, urađeno u dvostrukoj štampi. To će, znamo, razljutiti one hobiste koji sve vole da rade sami, naročito ako na ono što i sami mogu da urade moraju da čekaju dva-tri meseca. Ovakvi sklopovi, međutim, podrazumevaju štampano kolo profesionalne izrade — inače se može desiti da nikada ne prorade. Za one koji imaju razumevanja za domaće prilike i znaju da se neke stvari moraju i čekati, pripremili smo servis za nabavku štampanog kola. Za one koji gore od nestrpljenja i koji imaju više poverenja u sebe objavljujemo crtež obe strane u razmeri 1:1. Sa jedne na drugu stranu veza se prenosi u svim tačkama koje se nalaze na gornjoj strani — najčešće preko pinova na integriranim kolima ali ima i posebnih prenosioca. U tim tačkama (treba li to reći?) element se lemi sa obe strane. Oni koji se odluče za samogradnju štampanog kola moraću da nabave specijalne pinove za štampano kolo, poput onih na našim fotoaparatima, ili da zaleme čipove direktno. Ovo, naravno, važi samo za štampano kolo iz amaterske radionice. Na štampanom kolu koje obezbeđuje redakcija sve rupice će biti metalizirane i hobisti neće morati da brinu gde se veza prenosi, a gde ne. Na ovakvom štampanom kolu lemi se, naravno, samo sa donje strane.

### ... i štampani konektor

Na štampanom konektoru na računaru „galaksija“ nisu, iz neobjašnjivih razloga, izvedena dva priključka bez kojih memorijsko proširenje ne može da radi — priključak za +5 V za napajanje proširenja iz računara i RD priključak sa signalom za isčitavanje memorije. Zato se, pre nego što se proširenje priključi, na konektoru moraju izvršiti izvesne dorade: napajanje, komadom žice, treba dovesti na priključak 1, koji je, inače, prazan, a RD signal, direktno sa nožice 21 na mikroprocesoru,

na priključak 22; ovaj priključak je, inače, vezan sa masom računara i na sa veza, na konektorskoj pločici, mora preseći. Da bi se oslobodio prostor za CS signale, treba, takođe, preseći i sve veze između priključka 6 i 15. Sa zadnje strane memorijskog proširenja postavljena je konektorska pločica. Ona omogućuje da na računaru, istovremeno sa proširenjem, bude priključen još jedan dodatak — na primer, programator eproma, generator tonova, interfejs za štampač ili — nešto treće.

### Memorijski bagovi

Testiranje sagradnog proširenja je krajnje jednostavno — ovo testiranje, zapravo, „galaksija“ obavlja automatski prilikom svake inicijalizacije — i rezultat je vidljiv čim se uključi računaru. Dovoljno je, dakle, otkucati naredbu PRINT MEM, koja izdaje količinu slobodne memorije, i na ekranu će se, u zavisnosti od broja ugrađenih blokova, pojaviti brojka od koje će se konstruktor zavrtiti u glavi. Na žalost, nije tako. Ma koliko to čudno zvučalo, operativni sistem „galaksije“ ne obezbeđuje potpunu podršku za memorijsko proširenje — u njemu se potkralo nekoliko bagova koji u izvesnim situacijama otežavaju rad sa spoljnjim memorijom. Najozbiljniji bag se odnosi na pun memorijski format. Umesto da u takvoj situaciji račun sa 54 K slobodne memorije, koliko stvarno i ima, „galaksija“ u promenljivoj „kraj memorije“ upisuje nulu i ponaša se kao da nema ni jednog jedinog slobodnog bajta — odbija da radi u programskom modalitetu! Kako je to moguće? Prilikom testiranja memorije, „galaksija“ pristupka promenljivoj „kraj memorije“ na prvu lokaciju sa koje nema odziva — umesto na poslednju sa koje ga ima — a prva lokacija posle FFFF je 0000!

Ova teškoća će zagorčavati život samo onima koji na postojećih 6 K dodaje još 48 K radne memorije. Rešenje je jednostavno ali neelegantno i dosta zamorno: ako se pre početka rada u promenljivoj „kraj memorije“ (82A6A) upiše FFFF, „galaksija“ će ponovo postati poslušna do sledećeg — uključanja. Ručno postavljanje tzv. „ramtapa“ predstavlja u ovoj situaciji, na žalost, jedino rešenje. U svim ostalim situacijama „kraj memorije“ se postavlja kako treba, ali to, praktično, znači da se moramo odrediti čitavog memorijskog bloka (kako „odseći“ samo šesnaest bajtova?) koji, u našem slučaju, iznosi 4 K.

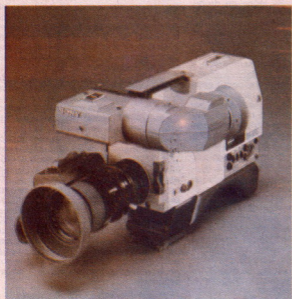
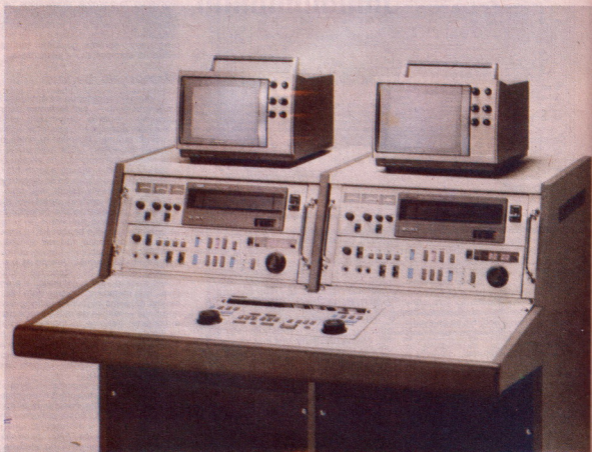
Naredba PRINT MEM, da stvar bude teža, sa više memorije počinje da izdaje sve čudnije rezultate. Ona ispravno funkcioniše samo do 22 K radne memorije, odnosno do adrese 32767. Nakon toga daje negativan broj iz logičan se ne može izvući ama baš nikakav logičan zaključak o stvarnoj količini slobodne memorije. Sa proširenjem od 16 K PRINT MEM daje rezultate 21445, od 32 K — 27706, a od 48 K — 11322! Ispitivane promenljive „kraj memorije“ — PRINT WORD (82A6A) — daje takođe negativne brojeve — 32768 (16 K), — 16384 (32 K) i 0 (48 K). U prostoru iznad 32 K RAM-a „galaksija“ u naredbama tipa BYTE i WORD prihvata samo heksadekadske ali ne i decimalne pozitivne brojeve. Sva ova ograničenja, međutim, nemaju nikakvog uticaja na uobičajeni rad sa memorijom — ona potpuno normalno prihvata i bejzik i mašinske programe.

Projekat i tekst: Jova Regasek  
Crteži: Mira Todorović

*Yugoslavia Commerce*



**SONY**



# SONY



## Yugoslavia Commerce

"YUGOSLAVIA COMMERCE" je preko 20 godina generalni zastupnik "SONY"-a za našu zemlju. Predmet zastupništva je celokupni proizvodni program SONY-a iz oblasti Hi-Fi i video opreme, kao i kompletni program U-MATIC opreme. Kompletan servis je obezbeđen u Beogradu, Sarajevska broj 5.

Od marta ove godine u Sarajevskoj br.3 saloni je izložen salon SONY-a. Za sada, "salon" nije kompletiran, ali će do kraja godine doći svi konkretni fiziolozi. Trenutno je izložena Hi-Fi i video oprema. To je tek prva naza na putu pronalazača stalne koncepcije izloženog salona.

U drugoj fazi koju očekujemo krajem maja, biće izložena U-MATIC video oprema. U-MATIC oprema je posebno interesantna za RO iz oblasti turističke privrede, mada je njena primena daleko šira: u bankarstvu, u osiguranju, školstvu, institutima, ... Na primeru turističke RO objasnimo vam jedan od načina primene U-MATIC video opreme. U-MATIC je pogodan:

- Za snimanje i prezentiranje kompletne turističke ponude na turističkim berzama i sajmovima. Poslovnim partnerima možete da prikazete dinamiku turističke sezone tokom cele godine. Vaše mogućnosti nisu više ograničene sezonom.

- U turističkim agencijama, pogotovo u onim koje se bave plasmanom turističkih aranžmana i prodajom izleta. Obogatite vašu ponudu video spotovima. Rezultati sigurno neće ostatati.

- Za projekciju filmova, muzičkih, sportskih i drugih emisija. Ponudite vašim gostima video, oni su na to već navikli. Možda im baš to nedostaje. Loše vreme ne sme da vas iznenadi.

- Za organizovanje interne televizije

U trećoj fazi ustanovljavanja SONY-evog salona biće predstavljeno uvođenje kompjutera u video sistem. Ova faza očekuje se krajem novembra '84 god. Kompletiranjem sve tri faze izložbeni salon mogao bi da postane ogledni centar jedne banke podataka o trenutno naj-savremenijoj audio, video i kompjuterskoj opremi.

Ako vam naš salon nije usputan, posetite nas na jednom od sledećih sajмова:

- Sajam tehnike u Beogradu: od 22 — 26.V. '84 god.
- Sajam elektronike u Ljubljani: od 3 — 10.X '84 god.

- Interdisk u Zagrebu: 18 — 24.IV '84 god. Da vas ukratko upoznamo sa našim poslovnim partnerom: Kompanijom SONY.

Malop je firmi u svetu koje su poput SONY-a, za tako kratko vreme, izmislile naše svakodnevnije navike. Još je manje firmi koje su to uradile tako "munjevit".

Ali krenimo od samog početka.

Firma SONY osnovana je u maju 1946 godine. MAJARU IBUKA i AKIRO MORITA, sa početnim kapitalom od samo 500 (USA) dolara, osnivaju malu firmu-radionicu: TOKYO TELECOMMUNICATION ENGINEERING CORPORATION, jednu od mnogobrojnih, koje su nastale odmah posle II svetskog rata, u Japanu. Za nepune četiri decenije SONY je prevalio put od male radionice do svetskog giganta. U prvoj deceniji lansiran je prvi japanski tranzistor (po američkoj licenci), a ubrzo prvi u svetu SONY izbacuje tranzistorski radio prijemnik. Kasnije sve je bilo jednostavnije. Pokrenuta je lavina pronalazaka koji će lagano osvajati fabrike, institute, domove, ... Ostalo su učinili sposobni inženjeri-elektroničari i spretni menadžeri.

SONY svake godine lansira na svetsko tržište nove proizvode koji postaju sastavni deo života savremenog čoveka. Kao ilustraciju nabrojimo samo neke SONY-ove proizvode:

- pored pomenutog tranzistorski radio prijemnika, za SONY je vezana i pojava prvog tranzistorski TV prijemnika.

- 1958 prvi tranzistorski video rekorder.
- 1969 prvi tranzistorski portabi video rekorder.

- 1984 prvi kućni video rekorder.
- 1969 razvijen U-MATIC sistem (3/4 inča).
- 1975 razvijen BETAMAX sistem (1/2 inča).
- 1976 razvijen OMEGA sistem (1 inc).

Pored ovog ne možemo a da ne spomenemo:

- TV sistem trinitron.
- usavršavanje video i audio traka (Fe i Cr trake su SONY-ev izum).
- WALKMAN (personalni Hi-Fi)
- NAVICU (prvi elektronski fotoaparati).
- COMPACT DISK (prvi digitalni gramofon u saradnji sa PHILIPS-om).
- BETAMUVIE (prva video-kamera rekorder).
- WATCHMAN (prvi džepni televizor).
- BETA Hi-Fi (prvi video rekorder sa Hi-Fi zvukom).

SONY-ev izuzetan doprinos razvoju Hi-Fi i video tehnike ovim nije iscrpljen. Spomenimo i trenutno najmanju kameru na svetu: SONY CCD-65. Očekuje se njena svetska premijera.

Organizacionu strukturu kompanije SONY čine tri njena dela:

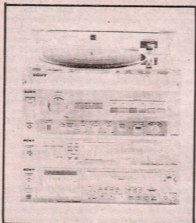
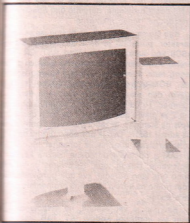
Prvi deo, koji je i bio osnovan za razvoj ostala dva, radi na razvoju opreme široke potrošnje. Tu se proizvode, Hi-Fi i video oprema.

Drugi deo, osnovan 1965. godine, je takozvani "INSTITUTIONAL VIDEO" baziran na sistemu U-MATIC.

I na kraju treći deo firme "SONY-BROD-KAST" nastao je tek sedamdesetih godina razvojem video-rekordera od 1 inča.

SONY-ovi ciljevi do 1990. godine su da se zadrži pozicija vodeće video kompanije u svetskim razmerama. Stratešku tužište je na razvoju novih media kao što su: teletekst, videotekst, kablovska televizija, satelitska televizija.

U svojoj orijentaciji SONY nije zapostavio ni razvoj mikrokompjutera. Krajem godine biće predstavljen SONY-ev personalni kompjuter (uglavnom za poslovnu primenu) sa kompletnim perifernom opremom. I u ovoj deceniji može se očekivati dinamičan razvoj, posebno ako se zna da SONY ulaze 5-6% godišnjeg profita u istraživanje i osvajanje novih proizvoda.



## Yugoslavia Commerce

OUR YUCOM TRADE  
Kneza MILOŠA 60/11000 Beograd  
011/668-166

IZLOŽBENI  
SALON

# SONY

11000 Beograd  
Sarajevska br. 3  
011/646-344

# svi spektarumov restarti Umetnost programiranja

Pri mašinskom programiranju procesora Z80 pozivanje potprograma se vrši instrukcijom CALL. No, ovaj procesor obezbeđuje još jednu instrukciju kojom se može izazvati izvršenje određenog potprograma. To je tzv. restart (RST). Prednost ove instrukcije je što u programu zauzima svega 1 bajt (u poređenju sa CALL — 3 bajta), pa samim tim zahteva manje vremena za njeno izvršenje. Ali, ovom instrukcijom je moguće pozvati samo potprogram koji počinje na jednoj od sledećih adresa: 0/H0/8B, 1/H0 1gh, 20H, 28H, 30H ili 38H. Odmah primećujemo da se ove adrese odnose na najniži memorijski prostor i to na tzv. nultu stranicu (zero page) memorije, tačnije prvih 256 bajta. Pogled na memorijsku mapu Spektruma uveriće nas da u ovom računaru svih 8 restart adresa „pripadaju“ ROM-u, tj. na njima počinju neki potprogrami operativnog sistema. Naravno, prostor između samih restart adresa je suviše mali za iole ozbiljniji potprogram, pa obično ovi potprogrami započinju skokom ili pozivom glavnog dela potprograma koji se nalazi na nekom drugom mestu u memoriji.

Pri kreiranju jednog složenog operativnog sistema, kao što je u Spektrumu, vodi se računa o tome da se na restart adresama postavе potprogrami koji se u toku rada najčešće pozivaju. To omogućava da se u glavnom delu operativnog sistema vrlo jednostavno, trošajući jedan bajt memorije, dake RST instrukcijom, pozove neki od tih potprograma. Zasto ovaj luksuz ne bismo ponekad priuštili i sebi pišući neki svoj mašinski program?

Pre svega, moramo upoznati tačnu funkciju svakog od ovih restart potprograma. Mi ne možemo menjati postojeći program u ROM-u (osim ako neko nije raspoložen da piše sopstveni operativni sistem), pa se stoga, ukoliko želimo da koristimo njegove potprograme, moramo prilagoditi njegovim „bubicama“.

Pogledajmo, stoga, redom, šta i kako rade pojedini restart potprogrami u Spektrumu.

## RST0: počni od početka

Poziv ovog dela operativnog sistema ne postoji nigde u ROM-u. Razlog je jednostavan: po uključivanju računara, procesor započinje izvršavanje instrukcija upravo od adrese 0 — posle provere RAM-a i inicijalizacije sistema prestaje sa radom dajući na ekranu svima poznatu copyright poruku. To znači da bi instrukcija RST 0 u nekom programu imala apsolutno isti efekat kao da smo isključili i ponovo uključili računar. Stoga ova instrukcija ima smisla jedino ako je upotrebimo na kraju nekog programa, a želimo da on bude uništen i sistem reinicijalizovan.

## RST 8: prijavljuje grešku

Potprogram koji započinje na adresi 08 ima pune ruke posla dok preko tastature unosite neki

bežik program. On je, naime, zadužen da izvrši normalizaciju steka i nekih sistemskih promenljivih svaki put kada interpreter naiđe na grešku u vašem programu. Osim toga, on na pogodan način „doturi“ interpreteru kôd greške, tako da se kasnije ona i tekstualno prikaže.

U čemu je značaj normalizacije steka? Ukoliko pogledamo bilo koji složeniji mašinski program, vidimo da se on sastoji od manje ili više potprograma koji se pozivaju iz jednog glavnog dela. Ovi potprogrami mogu, takode, dalje da pozivaju svoje potprograme, itd. Ako radimo mašinski program koji, recimo, briše ekran ili tome slično, nećemo imati potrebe da prijavljujemo greške. Međutim, ako pišemo program koji treba da prepozna određene komande ili koji radi interaktivno, tj. zahteva od korisnika unosenje nekih podataka, moramo kontrolisati te podatke pri unosu i javljati određenom porukom da je došlo do greške jer, recimo, sintaksa ili podaci ne odgovaraju. Šta da radimo ako greška nastupi u nekom potprogramu koji poziva drugi potprogram, a njega, opet, poziva, pretpostavljamo, glavni program? Pri svakom uzastopnom pozivu na stek se ostavljaju povratne adrese i jednostavno naredbom za povratka iz potprograma (RET) ne možemo se tek tako lako vratiti tamo gde hoćemo. Ako vršimo direktan skok, (JP ili JR) u potprogram za „regulisanje“ grešaka posle svake greške imaćemo sve više podataka na steku, (ostavljenih usled CALL instrukcija) i on će se, naprosto, zažugusti, a program krahirati.

Svih ovih i nekih drugih briga može nas, vrlo elegantno, osloboditi instrukcija RST 08. Ona će pozvati rutinu iz operativnog sistema koja će vratiti stek u prvobitno stanje i izvršiti indirektan skok u naš potprogram koji će, dalje, regulisati grešku. Naravno, mi moramo na neki način „javiti“ ROM-u gde se ta naša rutina nalazi. To činimo tako što u sistemsku promenljivu ERR SP (23613) ubacujemo, recimo na početku rada našeg programa, adresu na kojoj se nalazi naš potprogram za prijavljivanje grešaka. Najjednostavniji način da se ovo uradi je da se na početku programa na stek ubaci adresa naše rutine i sadržaj tek pointera (SP) odmah sačuva u ERR SP. To bi, onda, izgledalo ovako:

```
LD HL, ERRENT   adresa rutine za greške
PUSH HL         idi na stek
LD (ERR SP), SP sačuvaj stek pointer
```

Nakon ovoga dovoljno je da posle detekcije greške u programu jednostavno izvršimo RST 08 i dalje rad se nastavlja u našem potprogramu za prijavljivanje grešaka, sa stekom na prvobitnoj adresi.

Ako u programu treba da prijavljujemo više različitih grešaka, onda iz instrukcije RST 08 postavimo jedan bajt koji će predstavljati kôd određene greške. Posle izvršenja, ovaj kôd će se, zahvaljujući potprogramu iz ROM-a, naći u sistemskoj promenljivoj ERR NR (23610). Naša rutina za greške će, tada, jednostavno preuzeti ovaj kôd, proanalizirati ga i prijaviti odgovarajuću grešku. Veoma je važno da po izvršenju ovog potprograma, ili na njegovom početku, vratimo

na stek njegovu adresu (PUSH instrukcijom) da bi RST 08 i dalje mogao funkcionisati ispravno još nešto: ako hoćemo da po izvršenju našeg skok programa nastavimo rad u bežik sistemu, koristimo RST 08 na opisani način, onda pak ikakvog postavljanja u ERR SP treba sadržati u sistemске promenljive negde sačuvati (ima jednog zgodno mesto od dva bajta u zoni sistemске promenljivih na adresi 23728), a pre povratka bežik vratiti joj originalni sadržaj. Ukoliko smirdiral ERR NR, pre povratka u bežik sistem treba da u ovu lokaciju ubacimo vrednost 255 (FFH).

Za sve ove probleme, naravno, postoje mnoga druga rešenja, ali je verovatno najbolje koristiti potprograme iz ROM-a kad god je to moguće. To će, pre svega, smanjiti veličinu našeg programa, čemu uvek treba težiti.

## RST 10: ispiši karakter

Od svih ovih restart potprograma u ROM-programeru će, verovatno, najviše koristiti oni koji započinje na adresi 10H. Njegova funkcija da izvrši ispisivanje znaka, doturenog u akumulatoru procesora, na ekran ili štampač. Pod znakom se ovde podrazumevaju ne samo alfanumerički i grafički karakteri, već i svi kontrolni kôdovi koje obezbeđuje Spectrumov set karaktere. Ovaj potprogram radi nešto sporije, upravo zbog toga što obuhvata sve ove znakove. Stoga, ako zahtevamo maksimalno brzo ispisivanje na ekran, memoriju u igrama, onda nam jedino preostaje da pišemo svoje rutine za ispisivanje. No, ako vrem ispisivanja nije kritično, onda će nas RST 10 rešiti dosta problema, naročito oko Spectrumov „nezgodne“ video memorije.

Kao što smo već rekli, da bi smo ispisali jedan karakter, potrebno je da u akumulator stavimo ASCII kôd karaktera i izvršimo RST 10H. Ali, u praksi stvari zahtevaju malo više pažnje i pripremnog rada. Pre svega, pre bilo kakvog ispisivanja, ROM-u treba javiti gde vršimo ispis, na glavni deo ekrana, donje dve linije ekrana ili štampača. To činimo tako što u akumulator ubacimo 2 za glavni deo ekrana, 1 za donje dve ili 3 za ispis na štampač, te zatim pozovemo potprogram na adresi 1601H. Ovo je potrebno uraditi da bi se zoni kanalskih informacija (channel informacija) u RAM-u postavila određena izlazna adresa za ispisivanje. Nakon toga, ispisivanje će se vršiti tamo gde smo odredili, sve dok ponovnim pozivom rutine na adresi 1601H ne promenimo postojeće stanje.

Kao što je već rečeno, na raspolaganju su nam svi kontrolni kôdovi (AT, TAB, kôdovi za boje, poziciju kursora, itd.), kako je predstavljeno u uputstvu za Spectruma na strani 183. Posle kontrolnih kôdova za INK i PAPER, treba da sledi kôd boje (0—7), a iz FLASH, INVERSE, BRIGHT i OVER 1 ili 0 zavisi od toga da li „uključujemo“ ili „isključujemo“ određenu funkciju. Iz AT i TAB moramo dati dva parametra: za AT istovetno kao u bežikju, dok TAB kontrola zahteva za sledeći podatak bode stvarni TAB parametar, on je drugi podatak nebitan, ali ga moramo „poslati“.

**Mašinsko programiranje na računaru ne može se ni zamisliti bez dobrog poznavanja operativnog sistema. Većina rešenja koja su ljubitelji računara potrebna čak i za najjednostavniji program već se nalazi u ROM-u — na programeru je samo da ih „pronađe“ i iskoristi na svoj način. Teško je naći kućni računar koji prati tako obimna, raznovrsna i detaljna dokumentacija kao ZX Spectrum. Pa ipak, njegov operativni sistem je još uvek pun tajni i one koji se spremaju da zarone u njega sasvim sigurno očekuju prijatna iznenađenja. Za prikaz Spectrumovog ROM-a i opis svih rutina u njemu teško da bi bilo dovoljno i čitavo ovo izdanje. Zato prikaz ograničavamo na osam najvažnijih rutina, koje u ROM-u zauzimaju sasvim posebno mesto i koje računar u toku rada najčešće proziva.**

Zbog greške u ovom potprogramu ROM-a, naročito moramo obratiti pažnju da ne dozvolimo „ispis“ kontrolnog koda 8 (cursor left) u momentu kada su koordinate za ispis podešene na 0/0 (gornji levi ugao ekrana), jer će se desiti vrlo čudne stvari: autoru ovog teksta se dešavalo da bezijk program bude oštećen, a moguć je čak i kraj programa. Na ostalim pozicijama nema nikakvih problema. Kao i za svaki pojedini znak, za svaki kontrolni kôd i njegove parametre potreban je poseban RST 10H. Naravno, u programu ćemo to ostvariti pomoću petlji ili sličnih rešenja.

Za one koji su zainteresovani za „izlazak“ iz ROM-a, ovde se pruža šansa za tako nešto. Pošto RST 10H pre ulaska u samu rutinu za ispis karaktera traži njenu adresu u RAM-u, mi bismo mogli na to mesto u RAM-u (channel information, „podvaliti“ adresu naše rutine i na taj način preuzeti dalju kontrolu. Tako bi bilo sasvim moguće na mesto neiskorišćenih kontrolnih koda dodati neke nove, na primer za brisanje ekrana, automatsku tabulaciju, i slično. Posle provere kôda, ukoliko nije „naš“, izvršili bi smo skok u rutinu u ROM-u (počinje na 09F4H) na odgovarajuće mesto, pa bi ona dalje dovršila posao. Ovo je sasvim izvodljivo ako pozivamo RST 10H iz našeg mašinskog programa. Stvari postaju bitno komplikovanije ako hoćemo da na RST 10H iskačemo u našu rutinu za vreme normalnog rada bezijk interpretera. Razlog za to je što jedna od rutina za brisanje ekrana u ROM-u direktno postavlja u RAM adresu potprograma za ispis 02BF4H. Znači, ako mi izmenimo adresu na koju prelazi RST 10H, već prvo brisanje ekrana uništiće našu adresu. Protiv ovoga se, donekle, može „boriti“ tehnikom interapta, o čemu će, ako bude interesovanje, biti reči drugom prilikom.

## RST 18 i RST 20: uzmi sledeći karakter

Dva kratka potprograma na adresama 18H i 20H bezijk interpreter intenzivno koristi za vreme izvršavanja bezijk programa. Funkcija im je da, shodno sistemskoj promenljivoj CH ADD (23645), u kojoj se čuva adresa do koje je interpreter stigao analizirajući liniju bezijk programa, vrate interpreteru kôd karaktera koji je adresiran preko CH ADD (RST 18H), odnosno kôd prvog sledećeg karaktera (RST 20H). Eventualna primena ovih restart rutina bi mogla biti u mašinskim programima koji na neki način „nadiru“ bezijk interpreter, ili za eventualna proširenja postojećeg bezijka u Spectrumu.

## RST 28: počni da računaš

Na adresi 28H počinje Spectrumov kalkulator. Kao što je poznato, on omogućuje rad sa pokret-

nim zarezom i obezbeđuje Spectrumu sasvim pristojan set matematičkih funkcija. Rutine kalkulatora zauzimaju dobar deo ROM-a, no ubičajeno je da se poziv kalkulatora vrši instrukcijom RST 28H. Iza ove instrukcije treba da slede sukcesivno bajti kôd da odredi kalkulatoru „šta da radi“, sve do bajta (38H) koji određuje kraj rada kalkulatora. Izvršavanje programa se dalje nastavlja od sledeće instrukcije.

Sve funkcije koje su na raspolaganju u bezijku možemo upotrebiti i u našim mašinskim programima. Dobro upoznavanje ovog kalkulatora može biti od ogromne koristi svima koji su zainteresovani za rešavanje složenijih matematičkih problema u mašinskim programima. Kalkulator raspolaze sa 66 operativnih funkcija i 5 konstanti. Izračunavanje nekog matematičkog problema se svodi na to da se on „razbije“ na tekve delove koji odgovaraju operativnim funkcijama kalkulatora i zatim reši u sukcesivnom izvršavanju ovih funkcija. Table i objašnjenja u vezi ovih funkcija i konstanti bi nam odnele previše prostora u ovom tekstu, te preporučim čitaocima koji su zainteresovani za ove probleme sledite knjige: „Understanding your Spectrum“ od Dr lana Logana i „The Complete Spectrum Rom Disassembly“ od istog autora.

## RST 30: napravi malo mesta

Namena ovog kratkog potprograma, koji se inače vrlo često poziva iz operativnog sistema, jeste da u određenom prostoru u RAM-u (tzv. work space) „napravi mesta“ za onoliko bajta koliko mu se doturi u BC paru registra. S obzirom da funkcioniše preko određenih sistemskih promenljivih, čini mi se da ovaj potprogram za sada nema neku „upotrebu vrednost.“

## RST 38: prekini i očitaj tasteru

Restart potprogram na adresi 38H se bitno razlikuje od svih do sada spomenutih: specifičnošću lokacije 38H je u tome što procesor, ukoliko mi je naredeno da radi u režimu interapta (interrupt — prekid) i to u modu 1 (IM1), svakih 20 mikisekundi (dakle 50 puta u sekundi) prekida izvršavanje programa, „zapamti“ gde je stao i prelazi upravo na adresu 38H, da odatle izvrši potprogram koji na njoj započinje. Kada ga završi, vraća se na glavni program i to tačno na mesto na kome je bio prekinut. Iz ovoga proizilazi da, u stvari, rad dva programa paralelno: glavni, koji interpretira bezijk, itd., i drugi na adresi 38H. Postavlja se pitanje šta to radi taj potprogram kad mu je već dodeljena „čast“ da radi uporedo s bezijk interpreterom? Odgovor je: očitava tasteru i „pokreće“ Spectrumov interni časovnik. No, svakako, glavni zadatak mu je da očitaj tasteru i u sistemsku promenljivu LASTK (23560) ubaci kôd tipke koja je eventualno pritisnuta. Očitavanje tasterature se ne vrši pomoću ovog potprograma za vreme izvršavanja bezijk

naredbi LOAD, SAVE, i BEEP zbog isključenja interapta, mada SAVE i LOAD imaju svoje instrukcije za direktno očitavanje tipke BREAK.

Što se tiče naših mašinskih programa koji koriste tasteraturu, sve dok ne naredimo isključenje interapta instrukcijom DI (disable interrupts), u sistemskoj promenljivoj LASTK ćemo dobiti kôd pritisnute tipke uz prethodnu proveru kontrolnog bita u sistemskoj promenljivoj FLAGS (23611) — ukoliko je bit 5 jednak nuli, nikakva tipka nije pritisnuta, a ako je jednak jedinici, onda možemo da preuzememo kôd tipke iz sistemske promenljive LASTK. Jedno takvo tipično očitavanje bi izgledalo ovako: WAIT HALT; sačekaj interapt BIT 5, (Y+1); tipka pritisnuta? (Y=5C3AH) JR Z, WAIT; ne — skok nazad RES 5, (Y+1); mala predostrožnost LD A, (LASTK); uzmi kôd tipke RET; izlaz.

(Naredba HALT se može izostaviti. Labela WAIT tada ide na BIT 5, (Y+1).)

Ako nam je za rad programa neophodno isključenje interapta (naredbom DI), moraćemo sami da očitavamo tasteraturu. Ukoliko se radi o igri, gde se očitavaju 3—4 tipke, to ćemo uraditi direktnim očitavanjem sa „porta“ pomoću IN instrukcija. Lepo objašnjenje ovog postupka je dato u knjizi „Spectrum machine code for absolute beginner“. Međutim, ako nam je potrebna kompletna tasteratura, možemo pozivati, kad je to potrebno, potprogram iz ROM-a na adresi 02BFH ili, još zgodnije, sa RST 38H, s tim da odmah sledi naredba DI, pošto pri izlazu iz rutine na adresi 38H postoji naredba EI (Enable Interrupts) za omogućenje prekida. Posle ovakvog očitavanja, kôd eventualno pritisnuta tipke biće postavljen u LASTK, a bit 5 u FLAGS će javiti da je tipka zaista pritisnuta. Evo primera očitavanja tasterature u tom slučaju:

```
READ RES 5, (Y+1); mala predostrožnost (Y=5C3AH) CALL 02BFH; poziv potprograma iz ROM-a BIT 5, (Y+1); dali je neka tipka pritisnuta? JR Z, READ; ne — skok nazad LD A, (LASTK); uzmi kôd tipke RET; izlaz.
```

CALL 02BFH može biti zamenjen sa RST 38H a zatim DI.

U slučaju da pišemo program koji radi u režimu interapta u modu 2, možemo u servisnu rutinu, koja se izvršava po nastupu interapta, ubaciti RST 38H i na taj način imati očitavanje tasterature — kao u modu 1 (normalan rad bezijk interpretera i operativnog sistema), pošto interapt u modu 2 nastupa u istim vremenskim intervalima kao u modu 1 (svakih 20 ms).

Ovo bi bilo, u kratkim crtama, ono osnovno što moramo znati o „nultoj stranici“ Spectrumovog ROM-a i njegovim restart potprogramima da bi smo ih uspešno koristili u našim programima. Za detaljnije analize neophodan je disasembling ROM-a, a od velike pomoći mogu biti i knjige na ovu temu već spomenute u tekstu.

**UNIS****UNIS-RO promet birosredstava (p.o.) - Sarajevo****Olympia International****olivetti**

INFORMATIONS-UND KOMMUNIKATIONS-SYSTEME

**ELEKTRONSKI SISTEMI ZA PISANJE**

su na raspolaganju za oba, jednostrani ili dvostrani tok podataka.

**Kontrolna elektronika:** Kontrolna elektronika sastoji se od

1. Kontrolor štampača
2. Kontrolor modaliteta
3. Interface

**Interface:** IEC bus (slušač)  
 (kako izaberete) IEEE 488/ANSI MC 1.1.  
 Centronics kompatibilni, 7(8) bits  
 EIA RS 232 C (CCITT V 24)  
 TTY 20 ma current 100p, 50-19.200 Bauda

**Snabdevanje energijom:**  
 (kako izaberete) 220V 50Hz 110/220V 60Hz  
 240V 50Hz 110...127V 60Hz  
 110/220V50Hz 120V 60Hz  
 100V 50Hz 100V 60Hz  
 Sekundarno: UMG= +36V  
 UMK= +9V  
 Preko regulatora voltaže: UDD= +5V

**Osigurači:** Primarni: 1 termički osigurač (115 st. C)  
 Sekundarni: 1×1.6 amp sporo-eksplozivajući  
 1×4.0 amp sporo-eksplozivajući

**Potrošnja struje:** U radu — 70 watti  
 U pripremi — 35 watti

**Sigurnosna stipulacija:** \* Odgovara Evropskoj  
 sigurnosnoj stipulaciji BSI,  
 VDE, SEV, NEMKO itd.  
 \*\* Odgovara internacionalnoj  
 sigurnosnoj stipulaciji,  
 npr. IEC 380

\*\*\* Odgovara prekomorskoj  
 sigurnosnoj stipu  
 laciji CSA, UL.

**Štampač:**

Štampač sa lepe  
 zom (daisy wheel), 96  
 znakova, pozicioniranje šta  
 mpača i lepeze pomoću koraka  
 motor. Brzina štampanja 17 zna  
 kova/sekunda.

**Metoda štampanja:** Automatsko naprijed/na  
 zad, automatsko potiskivanje funkcije među  
 prostora.

**Stilovi štampanja:** Masni tip, prostorno kucanje, dupli udar

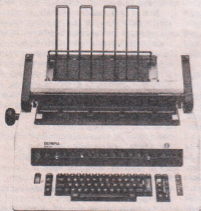
**Razmak slova (pitch):** Standardni stilovi 1/10", 1/12", 1/15",  
 proporcionalno pismo 1/60", osnovno kre  
 tanje 0.42 mm.

**Dužina linije:** 141 znakova sa razmakom 10 (pitch)  
 169 znakova sa razmakom 12  
 212 znakova sa razmakom 15

**Lista konstrukcija:** ASCII standard i sekvenca izbegavanja za  
 kontrolu štampača

**Mehanizam trake za pisanje:** Specijalne kasete

Ručni izbor priprema indiga i karbon trake,  
 automatski izbor multi-udarne karbon trake  
 štampanje u dva nivoa.  
 Standardna karbon traka, lift-off karbon  
 traka, dužina 130 m, širina 13 mm, broj  
 udara 100.000



**OLYMPIA INTERNATIONAL ESW 103  
 NOVA KLASA ŠTAMPAČA SA LEPEZOM  
 (typewheel)**

Gdje god su Vam potrebni pouzdanost i oštro-oblikovana  
 kucana strana ESW 103 je alternativa optimalnog odnosa  
 cena/performance.

Pisača mašina visokih performansi — naravno sa korekturnim  
 mehanizmom — za korespondenciju, ili savršeni štamp  
 pač za Vaš kompjuterski sistem, ESW 103 poseduje obe  
 funkcije. Selekcija između tipova operacija, bilo kao pisača  
 mašina ili kao štampač, izvršava se jednostavnim pritiskom  
 na taster.

Bilo standardno pisanje različitim stilovima, proporcionalno  
 pisanje, masna slova i/ili prostorno pisanje, od kojih svaki  
 ima svoj znak, svaki ima kvalitet koji odgovara  
 Vašem imidžu.

Intelegencija štampača oslobada Vaš kompuj  
 ter suvišnog opterećivanja. U fazi rada  
 na sistemu, primljeni tekst štam  
 pan je maksimalnom brzinom  
 i najbržim metodom. Lampe  
 na tastaturi su i pomoć  
 rukovaocu i simultano  
 ukazuju na odgo  
 varajući opera  
 cioni mod.

**ELEKTRONSKA  
 PISAČA MAŠINA  
 OLYMPIA ESW 103**

**Tehnički podaci**

Tip mašine: Tip 103 je mikroproce  
 sorom kontrolisana output pisača mašina sa  
 interface-om. Dodatno funkcijama pisače maš  
 ne, ova mašina može biti kao output ili input/ou  
 tput uređaj u on-line modalitetu. Ukupno 3 interface-a

**CENTRALNI DEO VASE KA**



Multi-karbon traka, dužina 115 m, širina 13 mm, broj udara 328.000  
Tkanena traka, crna, dužina 9 m, širina 13 mm, broj udara 1.080.000  
Tkanena traka, crno/crvena, dužina 9 m, širi na 13 mm broj udara 270.000

**Tastatura:** Tastatura sa 48 tastera tip TUMBLER. Svi tasteri sa funkcijom ponavljanja osim tastera akcenta. Tasteri funkcija: povrat štampača sa i bez prelaza na sledeći red, prostorni taster, polukorak, povrat za jedno mesto, ispravka, taster za pomeranje i fiksno pomeranje.

Kontrole desnog dela tastature: tasteri za desnu i levu marginu, taster ON/OFF-LINE taster tabulatora, prostor među redovima i taster za razmak (pitch) slova, tasteri za napred/nazad vertikalno pozicioniranje valjkorak, preko funkcije ponavljanja

Kontrole levog dela tastature: oslobađanje margine, taster tabulatora (tabuliranje una zad), taster šifre (code), taster ponavljanja.

Kontrole u kombinaciji sa tasterom šifre: masna slova, prostorno pisanje, dupli udar.

**Indikator lampe:** ON/OFF-LINE, greška, masna slova, prostorno pisanje, dupli udar.

**Valjak:** Dimenzije valjka: dužina 43,5 cm, prečnik 44,5 mm

**Buffer:** 2K i 4K byte buffer podataka u ON-LINE operacijama

**Memorija ispravki:** 256 byte (1 linija) u OFF-LINE operacijama

## ELEKTRONSKI SISTEM ZA PISANJE ETS 1010

Ukoliko već posedujete elektronsku pisaču mašinu Olivetti već ste načinili prvi korak u elektronski kancelarijski sistem Olivetti. Vašu elektronsku pisaču mašinu možete koristiti sa savremenim uređajem sa ekranom.

ETS 1010 ima CRT displej sa 25 linija, dve pogonske diskete i memorijski sistem sa 360 K-bajta memorije i mogućnošću povezivanja sa ET 121.

Za upis informacije jednostavno otkucati tekst (tastatura ima tradicionalni izgled). Informacija se prikazuje na displeju. (U ovoj fazi tekst na mašini se ne kuca), a kada se završi upis ili izmena teksta, štampa se konačan tekst brzinom od 20 znakova u sekundi. Ono što vidite na ekranu-doterani tekst, masni slog i podvučena slova sada dobijete otkucano na papiru.

Lako je naučiti i raditi sa ETS 1010.

ETS 1010 olakšava prelaz na upotrebu automatike uključivanjem u elektronsku pisaču mašinu OLIVETTI i održava blisku i komandnu vezu sa korisnikom. Mašina je još uvek najefikasnije sredstvo za pisanje kratkih obavesti, indeksnih kartica, poverata itd.

ETS koristi tehniku laganog pomeranja teksta. Kada se na displeju pomeru teksti ne vrše se preskoci od jednog reda u

određenom vremenskom periodu. Umesto toga, kreće se polako i omogućava jednostavno čitanje. Da bi se olakšala upotreba ETS ima funkcije UNDO i HELP. Ukoliko nešto radite (npr. umetanje paragrafa na pogrešnom mestu), a zatim uočite grešku nemojte brinuti, jednostavno poništite to (UNDO). ETS 1010 će sve dovesti u red.

Ukoliko niste sigurni u vezi sa funkcijom ili šta treba da se radi, tražite pomoć (HELP). ETS 1010 će vas provesti kroz sve funkcije. ETS 1010 takođe poseduje široki opseg novih funkcija za pomoć u upravljanju.

### Vanserijske funkcije.

Jedinstvena sposobnost vizuelnog sistema u ETS 1010 predstavlja novu standardnu odliku za lakšu obradu fusnota, naslova i brojeva kao predmeta kao i za prenošenje informacija od jednog dokumenta na drugi.

Vizuelni sistem omogućava da se ekran podeli na dva ili više površina za protok informacija. Veći dokumenti (duži od 80 kolona ekrana) se obrađuju u mnogim sistemima horizontalnim pomeranjem. Jedini problem je u tome što se gubi tekst. Upotrebom kapaciteta prosleđivanja u sistemu ETS 1010, leva kolona se može „zamrznuti“, dok se preostali deo dokumenta kreće.

### ETS 1010 pomoć u administrativnom rukovodstvu.

ETS 1010 poseduje „ladice za dokumentaciju“ i „registre“ za vaše dokumente. Možete staviti dokument u arhivu ili jednostavno referencu prema položaju dokumenta. Tehniku vođenja dokumentacije je lako razumeti jer ona obezbeđuje brzu dostupnost listi svih ladic za dokumentaciju, registrima i dokumentima kao i obaveštenje o datumu nastanka, autoru, daktilografu i određenom ključu. Identifikacioni sistem za automatsku reviziju štedi vreme administriranja filme što ubrzo va korekturu. Ukoliko je potrebno obaviti ovu funkciju ETS 1010 može emitovati ili štampati izbrisane reči precrtane linijom, ili dodate reči sa crtom iznad njih.

Sa ostalim vanserijskim elementima ETS 1010 će uskoro pomagati u proračunima, planiranju budžeta i periodičnim proračunima i pripremi izveštaja.

### ETS 1010 za pouzdanost

Kvalitet i pouzdanost su oznake izvedbe ETS 1010. ETS 1010 može podneti praženjanje statičkog elektriciteta koji se često javlja zbog suve klime i zagrejanih prostorija. Ukoliko je pad napona ili nestanak struje u radnim prostorijama česti, za zaštitu od uništenja informacija služi baterija (po izboru kupca).



**SVE DETALJNE INFORMACIJE KOD ZASTUPNIKA ZA JUGOSLAVIJU**

**UNIS**

**UNIS-RO promet birosredstava (p.o.)-Sarajevo**

UDRUŽENA METALNA INDUSTRIJA  
SARAJEVO

Dure Dakovića 45 d, 71000 Sarajevo  
telefon: 071/ 516-355  
39-342



# Honeywell

## INFORMACIONI SISTEMI



## PROGRES



Porodica računara „EI-HONEYWELL“

„EI-HONEYWELL“ NUDI OPREMU PO POTREBAMA I MOGUĆNOSTIMA KORISNIKA, ZATO, POTRAŽIMO REŠENJE ZAJEDNO!



Računski centar sa sistemom „EI-HONEYWELL“ DPS 6/95 Karakteristike tog modela su:

- najsnajzniji model iz serije EI-HONEYWELL DPS6
- 32 — bitni procesor visokih performansi za komercijalne i naučne obrade
- memorija kapaciteta 2 MB do 16 MB
- mogućnost priključenja do 112 terminalskih radnih jedinica i/ili komunikacionih linija
- kapacitet diskova do 3,6 GB (giga bajta)

EI — HONEYWELL  
INFORMACIONI SISTEMI

„EI-HONEYWELL“ Informacioni sistem je preduzeće sa zajedničkim ulaganjem (JOINT VENTURE) američke firme „HONEYWELL INFORMATION SYSTEM“ Inc. sa sedištem u Mineapolisu, poznatog svetskog proizvođača računara, **ELEKTRONSKE INDUSTRIJE**“, Niš jednog od najpoznatijih preduzeća u oblasti elektronike i digitalne tehnike u Jugoslaviji i „PROGRESA“, **OOUR 'PROGRES-INFORMATIKA**, Beograd, kao generalnog zastupnika i distributera opreme HONEYWELL, za koju pruža korisnicima svu stručnu pomoć, i kao takvo preduzeće bazirano na uspešnoj saradnji 3 partnera, obavlja sledeće delatnosti:

- **Proizvodnju** DPS6) modela mini, srednjih i velikih računara, video-terminala i serijskih štampača iz proizvodnog programa Honeywell-a
- **Plasman** mini, srednjih i velikih računara na jugoslovenskom tržištu
- **Export** mini, srednjih i velikih računara u više od 50 zemalja sveta, posebno u nesvrstane zemlje i zemlje u razvoju
- **Projektovanje** informacionih sistema
- **Razvoj** aplikativnih paketa
- **Distribuciju** sistemskog softvera i aplikativnih paketa
- **Instaliranje**, održavanje i poslove field-engineerig-a
- **Obuku** korisnika računarskih sistema.

Svi računari se proizvode pod znakom „EI-HONEYWELL“, koji garantuje kvalitet, proizvoda na svetskom nivou, sa automatski prenetim novim znanjima i otkrićima samog „HONEYWELL“-a

Danas na 5-godišnjicu potpisivanja JOINT VENTURE Ugovora, možemo se pohvaliti da smo sa otvaranjem nove fabrike u Nišu (1. maj 1981) uspeali da zadovoljimo velike potrebe jugoslovenskog tržišta, i dosada plasiramo preko 300 sistema koji su dobili mnoge pohvale i priznanja od strane zadovoljnih korisnika.

©ELEKTRONSKA INDUSTRIJA NIŠ  
RO „EI-RACUNARI“  
OOUR „EI-HONEYWELL  
INFORMACIONI SISTEMI“  
18000 Niš, Bui, Vojvoda Vukobratovića 89-82  
tel. 011-332-342, 334-090  
telex: 14286 YU EI-NIS

©EI Honeywell  
INFORMACIONI SISTEMI  
SEKTOR MARKETINGA  
11000 BEOGRAD, Terazije 3-4  
tel. 011-343-444, 323-530  
telex: 11937 YU EI-NIS

RO „PROGRES“  
OOUR „PROGRES-INFORMATIKA“  
11000 BEOGRAD, Bulevar Oslobođenja 1-1/118  
tel. 011-485-947, 541-565  
telex: 11820 YU PROGO  
Predstavništvo:  
41000 ZAGREB, Čačkovića 4  
tel. 041-447-478  
81000 LJUBLJANA, Cankarjeva 4  
tel. 061-324-783

NOVO U IZDANJU  
IZ "VUK KARADŽIĆ"

# ELEKTRONSKI RAČUNAR

Most u budućnost

SVE ŠTO STE  
HTELI DA ZNATE  
O RAČUNARIMA,  
PROČITAJTE  
U OVOJ KNJIZI

Elektronska obrada podataka, kao polje aktivnosti savremenog čoveka, i elektronski računar, kao tehnološki proizvod koji omogućava tu aktivnost, poslednjih godina se izuzetno brzo razvijaju i šire. U svakodnevnom životu su sve prisutniji i imaju sve veću ulogu.

Literatura iz ove oblasti je u svetu veoma brojna i bogata. Kod nas, na žalost, to nije slučaj. Postoji samo manji broj udžbenika i priručnika okrenutih prvenstveno programskim jezicima i teorijskim osnovama računara namenjenih studentima.

Nedostaju tekstovi koji daju celovit prikaz računara i njegovih mogućnosti na popularan, iako dosledno stručan način dostupan svima čije je poznavanje matematike i tehnike na elementarnom nivou.

OVA KNJIGA TREBALO BI DA  
POPUNI OVU PRAZINU

Stanko Popović

## elektronski računar

most u budućnost



Vuk Karadžić

Izdavačka radna organizacija „Vuk Karadžić“  
Beograd, Kraljevića Marka 9, tel. 620-371

### NARUĐBENICA RAČUNARI II

Nepozivno naručujem knjigu ELEKTRONSKI RAČUNARI po ceni od 900 –dinara, ZA GOTOVO — sa popustom od 20%.  
Plaćanje poštom u prilikom prijema knjige. NA OTPLATU u 2 mesečne rate. Rate ću otplaćivati redovno svakog meseca,  
po prijemu knjige, računa i odgovarajućeg broja uplatnica „Vuk Karadžić“. Kamata će mi se obračunavati prema  
broju rata i važećim zakonskim propisima.

(Nepotrebno precrtati)

U slučaju spora priznajem nadležnost Opatinskog suda u Beogradu.

Prezime, očevo ime i ime \_\_\_\_\_

Mesto i godina rođenja \_\_\_\_\_

Zanimanje \_\_\_\_\_ Pošt. broj i mesto stanovanja \_\_\_\_\_

Ulica i broj \_\_\_\_\_

telefon \_\_\_\_\_

Naziv i adresa radne organizacije \_\_\_\_\_

telefon \_\_\_\_\_

Oveva o zaposlenju za kupce na otplatu

Potpis kupca

br i karle \_\_\_\_\_ izdate od SUP-a \_\_\_\_\_

Penzioneri prilažu pretposljednji ček od penzije

Latinica,

130 strana,

format: 165 × 22cm,

60 fotografija i ilustracija,

tvrd, plastificirani povež.

Cena: 900 din.



# Iskra Delta



RO ISKRA DELTA je proizvođač kompletnih računarskih sistema sa uhodanim razvojem i proizvodnjom mašinske opreme, sistemske i aplikativne programske opreme, sa organizovanom prodajom, školovanjem, održavanjem i inženjeringom.

Osnovna koncepcija RO ISKRA DELTA temelji se na proizvodnji računarskih sistema i perifernih jedinica sa maksimalnom primenom domaće tehnologije i znanje, a na osnovu najnovijih svetskih dostignuća u ovoj oblasti.

Iskra Delta  
proizvodnja računarskih  
sistema i inženjering

LJUBLJANA, Parmova 41  
Telefon 061/312-988 h.c.  
Telex 31366 YU DELTA