

# 6

Izdaje  
BIGZ  
OOUR  
„Duga“

## računari

Specijalno izdanje časopisa „Galaksija“  
Cena 200 dinara juni/juli 1985.

---

„galaksija“  
**fina grafika**

---

„komodor“  
**recepti za nove naredbe**

---

„spektrum“  
**sažimanje ekrana**

---

periferijska oprema  
**monitori**

---

ekskluzivno  
**„atari“ u Jugoslaviji**

---

novi računari  
**„enterprajz“**

---

igre  
**besmrtnost i kako je steći**

---

programeri na ispitu  
**hakerski zakonik**

---

smešna strana računara  
**razvojni put bore pirata**

---

vodič za maturante  
**kako studirati računarstvo**



**UNIS**

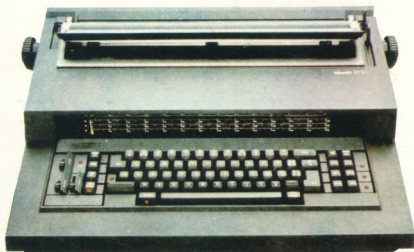
**RO TBM — TVORNICA BIRO MAŠINA** BUGOJNO

U saradnji sa vodećim svjetskim firmama  
**OLIVETTI** i **OLYMPIA** proizvodi  
elektronske programir pisace mašine

**UNIS**

**UNIS**

**UNIS**



**UNIS**

**UNIS**

**UNIS**



**UNIS**

**Plasman: RO PROMET BIROSREDSTAVA**  
**SARAJEVO**  
Đure Đakovića 45 d  
tel. 071/37-083





cena 200 D **Y** Izdanje BIGZ OOUR „Duga“  
**racunari**  
 junij/juli 1985.  
 Specijalno izdanje  
 časopisa „Galaksija“

**Izdavač**  
 Beogradski izdavačko-grafički zavod  
 OOUR Novinska delatnost „Duga“  
 11000 Beograd  
 Bulevar vojvođe Mišica 17  
**Telefoni**  
 650-161 (redakcija)  
 650-528 (propaganda)  
 651-793 (propaganda)  
**Generalni direktor**  
 Gojko Žegar  
**Direktor OOUR „Duga“**  
 v.d. Aleksandar Bađanjak  
**Glavni i odgovorni urednik**  
 Gavriilo Vučković  
**Urednik izdanja**  
 Jova Regasek  
**Likovna i grafička oprema**  
 Dušan Miličević  
**Redakcija časopisa „Galaksija“**  
 Tanasije Gavranović, pomoćnik  
 glavnog i odgovornog urednika  
 Esad Jakupović, zamenik glavnog  
 i odgovornog urednika  
 Aleksandar Miličević, urednik  
 Jova Regasek, urednik  
 Zorka Simović, sekretar redakcije  
 Srđan Stojanović, novinar  
 Gavriilo Vučković, glavni i odgovorni  
 urednik  
**Stručna saradnja**  
 Dejan Ristanović  
 Nevenka Spalević  
 Anđelko Zgorelec  
 Mihnajlo Tesević  
**Autori tekstova**  
 Predrag Bogdanović

Branko Đaković  
 Đorđe Janković  
 Vladimir Kostić  
 Ivan Nador, dipl. ing.  
 Damir Omrčen  
 Borivoj Perčić  
 Bogdan Petrović  
 Dejan Ristanović  
 Jelena Rupanik  
 Duško Šavić  
 Jovan Skuljan  
 Mr. Veljko Spasić  
 Nenad Dunjić  
 Dušan Veljković  
 Zoran Živočić  
**Crteži**  
 Miša Marković  
**Tehnička saradnja**  
 Ljubisa Milovanović  
 Ljilav Rjadčenko  
**Prevodioci**  
 Esad Jakupović  
 Ksenija Pješćić-Lebedinski  
**Izdavački savet „Galaksije“**  
 Dr Rudi Debijadi, prof. dr Branislav Dimitrijević (predsednik), Radovan Drasković, Tanasije Gavranović, Zvonar Glišić, Esad Jakupović, Velizar Maslić, Nikola Pajić, Željka Perunović, prof. dr Momčilo Ristić, Vlada Ristić, dr inž. Milorad Teofilović, Vidoklo Veličković, Velimir Vesović, Miloje Vučković  
**Stampa**  
 Beogradsko izdavačko-grafički zavod  
 11000 Beograd, Bulevar vojvođe Mišica 17  
 Ziro-račun kod SDK 60802-833-2463  
 Devizni račun kod Beobank  
 60811-820-8-82701-999-01066  
 Za inostranstvo cena dvostruka (400 D,  
 2,50 US \$, 6,50 DM, 45 Sch, 5,50 Sfrs,  
 20 Ffrs)  
 Na osnovu mišljenja Republičkog  
 sekretarijata za kulturu broj 413-77/72-03 i  
 „Službenog glasnika“ broj 26/72, ovo  
 izdanje oslobođeno je poreza na promet.

**sadržaj**

- 4/** šta ima novo u svetu računara
- 7/** load „pismo“
- 9/** računari u izlogu  
„enterprajz“
- 10/** promašaj na japansku način  
**računari izlazećeg sunca**
- 14/** računari u razgovoru  
**neki čudni klinici**
- 16/** izbor računara  
**hakeri to vole vruće**
- 17/** periferijska oprema  
**monitori**
- 20/** programeri na ispitu  
**hakerski zakonik**
- 22/** smešna strana računara  
**razvojni put bore pirata**
- 24/** računari iz mog ugla  
**obračun kod turingove mašine**
- 25/** programi koje treba imati  
**slikari bez kičice**
- 26/** obrada teksta  
**pero od osam bita**
- 31/** majstorije na računaru (komodor)  
**novе naredbe na nov način**
- 33/** biblioteka programa
- 38/** umetnost programiranja („spektrumi“)  
**nova kruna za starog kralja**
- 41/** put u sedište roma-a  
**velika video predstava**
- 46/** majstorije na računaru  
**ekran pod presom**
- 49/** programiranje u bezjiku  
**hamlet u računaru**
- 53/** vodič za maturante  
**kako studirati računarstvo**
- 54/** umetnost programiranja  
**svirka na komodoru**
- 54/** majstorije na računaru („komodor“)  
**novе naredbe na novi način**
- 56/** računari i igre  
**besmrtnost i kako je steći**
- 57/** hajde da se igramo
- 60/** računari u domaćoj radinosti  
**rom od sedam milja („spektrum“)**  
**fini hardver za finu grafiku („galaksija“)**  
**„kata“ na „galaksiji“**
- 66/** hakeri u nevolji

Atari u Jugoslaviji

## Više za klase nego za mase

Jugoslovensko tržište, barem do sada, nije bilo naročito zanimljivo za svetske proizvođače kompjutera. Od „Iskrinog“ ugovora sa Sinklerom domaći ljubitelji računara su samo dobili glavobolju, dok se BBC i Amstrad okrutno oglašuju o sve pozive. Prolećna ofanziva Atarija na Evropu zahvatila, međutim, i našu zemlju. Na Beogradskom Sajmu tehnike „Mladinska knjiga“ iz Ljubljane i reprezentativna ekipa iz Atarija — Sem Tramiel, potpredsednik kompanije Atari, Alvin Štumpf, predstavnik Atarija za Evropu i Les Plejer, stručnjak za softver iz Atarijeve londonske poslovne — obznanili su svoje planove Atarijevog pohoda na računarsku Jugoslaviju. Povoljni uslovi nabavke, obilje jeftinog softvera i, naročito u početku, damping cene predstavlja osnovno oružje kojim „Mladinska knjiga“ i Tramiel pokušavaju da osvoje naklonost domaćih hakera i male privrede.

Kao što je poznato, Džek Tramiel je sa saradnicima, među kojima je bio i njegov sin Sem Tramiel, prešao iz „Komodora“ u Atari. Atari je tada bio na ivici propasti, ali je ujedinenim naporima „Warner Communication“a i „Tramieljeve grupe“ podmladeni čitava kompanija. Započeto je sa radom na pravoj gomili novih računara. Jedan iz te gomile je i, ovde prezentirani, ST 520, koji u čitavom svetu izaziva ogromno interesovanje. ST 520 je šesnaestobitni računarski sistem, sa finom grafikom, „mišom“ proziorima i svim što ide uz mašinu takve klase. Ono što ga čini veoma interesantnim je, kao što je slučaj i sa ostalim proizvodima „novog“ Atarija, veoma niska cena za računarski tog kvaliteta i takvih karakteristika. Već je postala čuvena Tramielova kraljica: Radio za mase a ne za klase.

Jedna od prvih izjava Sema Tramielja, ne bez diplomatske kurtoazije, bila je da se nadaju da će uvođenjem ST 520 na jugoslovensko tržište početi uskoro da dobijaju i programe koje su radili stručnjaci iz Jugoslavije. Očito je da Atari pokušava da proširi biblioteku programa za 520. Na pitanje u čemu se sastoji nova „Atarijeva“ strategija, Tramiel je odgovorio da, kao i do sad, računaju na to da ponude izuzetno kvalitetne proizvode po izuzetno niskoj ceni, a da zaradu, za razliku od drugih proizvođača, ostvaruju velikim obrotom, a ne visokom cenom. Uputan, na koju vrstu softvera najvi-



Pažnja koja zbunjuje: Novi „atari“ i Atarijeva ekipa na Beogradskom sajmu

še računaju od domaćih programera, Tramiel je izjavio da za TS 520 očekuju programe u sve tri oblasti — obrazovanju, zabavi i poslovnim primenama. Pošto je poznata Atarijeva akcija da se britanskim školama računari ponude po nižim cenama, Sem Tramiel je upitan da li nameravaju da nešto slično probaju i u Jugoslaviji. Odgovor je glasio da za svaku zemlju imaju posebnu taktiku i da će o načinu nudenja računara školama tek razgovarati. Na pitanje da li Atari radi na otvaranju novih tržišta ili je ovo probaj samo u Jugoslaviji, odgovor je bio isto toliko efekatan i kratak: „Gospodin Tramiel sutra leti za Moskvu.“

Ako najavljuje cenu ST 520 budu tačne, to će biti izuzetno jeftin i atraktivan računarski sistem, sa obzirom na njegove mogućnosti i kvalitet. Predstavnik „Mladinske knjige“ je izjavio da će se računari Atarija prodavati preko konsignacije, ali da se radi na tome da se na neki način omogući i prodaja za dinare. Cena sistema ST 520 je negde ispod 3000 maraka, a tu je uračunat i crno-beli monitor i disk drajv. Osvrćući se na jugoslovensko tržište računara, Alvin Štumpf, predstavnik Atarija za Evropu, rekao je da se jugoslovensko računarsko tržište nalazi na istom stepenu razvoja kao i Nemačka pre dvanaest meseci, tako da, otkrili, znaju šta i kako treba činiti na njemu pošto već imaju iskustva sa nemačkom tržišta. Da li je to bilo kompliment ili laskanje jugoslovenskom tržištu teško je bilo utvrditi zbog izuzetno prijatnog i profesionalnog stava Atarijevih predstavnika. Alvin Štumpf je, takođe, izjavio da je u toku uspostavljanja mreže Atarijevih prodavaca u Nemačkoj i da su prvi primerici već krenuli ka svojim kupcima. Očekuje se da će najesen stići i prvi primerici za Jugoslaviju, izuzetno je zanimljiva izjava gosp. Štamfa da će prvi nekoliko stotina Atarijevih računara za Jugoslaviju, biti po izuzetno niskim cenama, nižim i od nemačkih. To bi mogao da bude baš onaj korak koji li Atarija doveo u lepu poziciju na jugoslovenskom tržištu. Na pitanje da li se radi na prevodenju poslovnih i drugih programa koji već postoje za Atarijeve računare, odgovoreno je da se još ne

radi, ali da su pripreme u toku. Objavljen je i Atarijev princip da sve popravke preuzimaju privatnici.

Zanimljivo je i to da Atari i „Mladinska knjiga“ nameravaju da ponude izuzetno jeftin softver koji će biti samo malo skuplji od piratizovanih programa koji se prodaju na „paralelnom tržištu.“ To bi bio prvi put da jedna kompanija tako drastičnim spuštanjem cena programa parira privatnim pretničarima. To je, verovatno, uslovljeno specifičnostima našeg tržišta (?).

Gospodin Štumpf je izjavio da se u dogovoru sa kompanijom Robotron već radi na printeru koji bi imao slovni set specifičan za našu zemlju. Hardver je već gotov, ali se još radi na softveru. Rečeno je da će zainteresovanima biti dostupna sva dokumentacija vezana za Atarijeve računare, kao i da će „Mladinska knjiga“ izdati knjigu o MC68000 čipu.

Sem Tramiel je još rekao da Atari vidi ST 520 kao ozbiljan računarski sistem, personalni kompjuter koji zadire i u oblasti kućnih i poslovnih mašina. Iako su to karakteristike svih njihovih novih mašina, to ne znači da se oni odriču tradicionalne Atarijeve naklonosti prema video igrama. On tvrdi da je to jedna od oblasti u kojoj su druge kompanije, uključujući i Komodor iz koga su došli, zakazale zato što nisu shvatili da je to izuzetno vitalna i jaka poslovna oblast. Atari to nikad nije prevideo.

Ostatak razgovora je posvećen tehnikama vezanim za pojedine Atarijeve računare. Pomenut je MIDI interfejs koji služi za povezivanje sa sintersajzerima. ST 520 je jeftiniji računarski sistem koji ima ugrađen takav interfejs.

Saznali smo da se u Atariju upravo radi na razvoju trideset-bitnog računara. Predstavnicima Atarija je, takođe, bilo značito veliko interesovanje za ST 520 pomalo bica ST 130 u senku, „iako je to veoma kvalitetan 8 bitni računarski sistem sa mnoštvom dobrih osobina (miš, prozor) i veoma niskom cenom.“

Ceo razgovor je protekao u znaku velike zainteresovanosti za novitete Atarija i veoma je verovatno da će Atarijev debi u Jugoslaviji biti izuzetno značajan — nadajmo se podjednako za Jugoslaviju koliko i za Atari.

Branko Đaković

## Čiji mozak truju računari

Kompjuteri su u poslednje vreme izuzetno atraktivna tema — nema časopisa koji nije, bez obzira na svoju tematsku usmerenost, doneo bar jedan članak o računaru. Pošto je to tema o kojoj „klasični“ novinari nemaju baš previše pojma, uglavnom se radi o prenošenju članaka iz nekog stranog časopisa.

U 101. broju „Intervju“ je učinjeno baš to — prenet je iz italijanske „Stampe“ intervju sa Džozefom Vajzenbaumom, izuzetno značajnim autorom iz oblasti odnosa računara i društva. Džozef Vajzenbaum zaslužuje mnogo više od ovog i ovakvog članka, ali to je već greška italijanskih novinara. Značajnija stvar je prezentacija kakvu on ima u „Intervju“. To je lako videti i iz naslova i nadnaslova koji su dati ovom članku: „Kompjuteri truju mozak!“ i „Džozef Vajzenbaum — disident tehnološke

...? Ne znam da li se i o računaru može pisati senzacionalistički, ali ovo je dobar pokušaj.

Nadnaslov „Džozef V. — disident tehnološke ere“ je olako ubačen iz tekst italijanskog novinara i nije ama baš ničim argumentovan. Znam da reč disident lepo zvuči i da ima prizvuk koji privlači čitaoca, ali ne treba preterivati. Naslov je još besmisleniji. On je dobijen, verovatno, pogrešnim navođenjem rečenice u kojoj Vajzenbaum tvrdi da „... bezik izaziva, da tako kažem, trovanje mozga“. To izgleda nije bilo dovoljno prerađivano ovog intervjua, pa je u podnaslov izvukao još jednu rečenicu svog italijanskog kolege, u kojoj je Vajzenbaumova najnovija knjiga okarakterisana kao „pohod protiv kompjuterske mode“. Velike reči znače uglavnom to da je Vajzenbaum protiv nakaradnog shvatanja računara. kakvo, uostalom, predstavlja i ovaj intervjua.

Ukratko, u naslov, nadnaslov i podnaslov su stavljena tri navoda koji najmanje imaju veze sa argumentovanim i konciznim izlaganjem Džozefa Vajzenbauma, koje je moguće uočiti u ovom intervjua, iako intervjua izgleda jako čudno, kao da je drastično skraćan ili vođen u hodu. Greške u njemu je najbolje i ne pominjati (Vajzenbaumova prva knjiga se zove „Moć Računara i Ljudski Um“ a ne „Moć Računara i Nemoć Ljudi“).

Nadam se da će „Intervju“ još puno svog prostora posvetiti računarima i sličnim stvarima, ali i da će tome posvetiti bar onoliko pažnje koliko i drugim temama koje obrađuje.

Branko Đaković

## Srećan rođendan, Meki!

Odprilike u ovo doba, pre godinu dana, i do nas su stigle vesti o novom spektakularnom 32-bitnom računaru, potpuno revolucionarne koncepcije, nazvanom „mekintosh“ (MACINTOSH).

Pošto su se prva euforična reaganovanja stišala, na videlo su izbili mnogi nedostaci nove mašine: samo jedan disk drajv, samo 128 K memorije (MC68000 procesor prosto guta memoriju), nema nikakve mogućnosti za dalja proširenja, nema hard disk, nema ugrađen programski jezik ili integralni softver, jako je spor, procesor nije, suprotno tvrđenju Apple-a, 32-bitni već 16-bitni, softverska podrška je ravna nuli...

U stvari, priča o „mekintoshu“ počinje par godina unazad. Tada je, uz ogromna finansijska ulaganja, Apple razvio stvarno revolucionarnu mašinu — „lizu“. Kompjuter sa famoznim 68000 procesorom, jednim megabajtom memorije, dva ugrađena floppy diska velikog kapaciteta, hard diskom, rezolucijom 720 puta 364 i velikim mogućnostima za dalja proširenja. Čim se pojavila, „liza“ je bila jednoglasno proglašena za računar decenije, a pratiči softver, mada ne broj, bio je, kažu, fenomenalan. Ipak, dva-tri programa, ma koliko bila dobra, ne čine softversku podršku, a u želji da što pre popuni isprazne kase, „Apple“ je za „lizu“ odredio astronomsku cenu — preko 10 000 dolara.



I, stvar bi nekako prošla da se u to vreme na tržištu mikračunara nije pojavio div sa godišnjim prihodom od preko 45 milijardi dolara koji je već decenijama iza sebe imao samo zadovoljne kupce — IBM. IBM-ov prvenac, jednostavno nazvan PC, mada nije bio tehnički tako savršen kao „liza“, počeo je da se prodaje kao alva, a softverska podrška je vrtoglavo rasta. Za rekordno kratko vreme, postao je najbolje prodavan računar u istoriji i sa najjačom softversko-hardverskom podrškom, a ljudi su ubrzo shvatili da je to, a ne „liza“, računar decenije.

Dok su „Appleovi“ tehničari i dalje prdali pohvale za svoj rad, ljudi iz majke nisu mogli čudom da se načude zašto se prodaju „lize“ broje na prste jedne ruke. Učinjen je očajnički pokušaj da se stvari izmene. Sa „lize“ je poskidano sve živo što se skinuti moglo — umesto dva disk drajva od po 860 K, stavljen je jedan prvoclajv od 400 K, umesto 1 M RAM-a, ostavljeno je samo 128 K, softver je malo prerađen, a cena spuštena na podnošljivu meru. Rodio se „mekintosh“ i, gle čuda, stvari su najzad krenule.

Stvari su, čak, krenule toliko dobro da ultra moderna fabrika u Kaliforniji nije stizala da zadovolji potražnju, mada nikome nije bilo jasno ko i zašto kupuje „mekintosh“. Sa „mekintoshem“ je bukvalno nemoguće uraditi bilo šta. Ako želite da obrađujete tekst, možete da napišete samo osam i po strana — više memorija nema. Ako vam se pišu programi, morate da za 130 dolara kupite bezik interpreter, a kad ga učitate u memoriju, ostaje slobodno manje od 20 K. Sa ostalim aplikacijama stvari stoje još gore.

Da bi nekako doskočilo problemu, „Apple“ je posle izvesnog vremena počeo da prodaje „FAT MAC“ — „mekintosh“ sa 512 K memorije. Naravno, tako to „Apple“ radi, cena je otišla nebu pod oblake. Onima koji su još prvih dana kupili „mekintosh“ sa 128 K, sada se nudi proširenje memorije za „samo“ 995 dolara!

Godinu dana kasnije nije mali period, pa su neke stvari stigle da se poprave. Najzad su se pojavili hard diskovi (nezavisnih proizvođača — „Apple“ ih još uvek ne pravi) ali, avaj, sada je problem kao ih povezati sa mašinom. U nekim slučajevima, potrebno je otvoriti kompjuter i prčkati po unutrašnjosti. Što se tiče softvera, tu je stanje još uvek kritično. „FAT MAC“ je bio dobar potez, ali sada treba vremena da softverske firme prilagode svoje programe „mekintoshu“. Majkrosoft se, između ostalog, sprema da lansira poboljšanu verziju WORD procesora, a LOTUS obećava softverski paket pod nazivom „JAZZ“ (poboljšana verzija slavnog LOTUS 1-2-3).

Jedino što u celom nastalom cirku su stvarno služi na čas „mekintosh“ je neverovatno lak način na koji se upotrebljava.

Ako je to bio cilj, onda je stvarno postignut. HAPPY BIRTHDAY!

Vladimir Kostić

## Soft-aid za Etiopiju

Posle velike pomoći koju su engleski rokeri, pod imenom Band-aid, prikupili za gladne u Etiopiji, izgleda da su i programeri na putu da urade nešto slično.

Projekt se zove Soft-aid. Radi se o jednoj kaseti sa 10 video igara (postoje dve varijante — za „spektrum“ i za „komodor“) koja se prodaje po ceni od 4,99 funti. Od svake prodate kasete 3 funte će biti uplaćeno u Fond za gladne u Etiopiji. Vredan pokušaj da se medijski veoma jaka stvar, kao što su to trenutno video igre, iskoristi u humane svrhe. Organizatorima Soft-aida najveću brigu predstavljaju pirati, pa je po engleskoj štampi objavljena molba u kojoj se od pirata traži da ne diraju ovo kasetu jer bi to moglo da upropasti čitav poduhvat.

Ako se ikad šta slično bude organizovalo i kod nas, to neće biti samo dokaz da smo humani i da brinemo za gladne, nego i da su naši programeri položili veliku (znanstvenu) maturu. A kako će naši potencijalni organizatori zamoliti pirate da ne diraju u programe, e to bih baš voleo da vidim.

BBC/Electron

BBC B+

Iz junskeg broja časopisa Acorn User saznali smo da se i Acorn pridružilo sve popularnijem trendu izdavanja „plus“ modela svojih popularnih kompjutera — zvezda predstojećeg Acornovog sajma je novi BBC B+.

Da smo ovu vest želeli da propratimo fotografijom, ne bismo imali mnogo problema da je pronađemo — novi model spojila izgleda potpuno isto kao i stari BBC B. Razlika se primećuje tek kada skinemo gornju ploču: raspored komponenta je sasvim promenjen. Novi računar je zasnovan na mikroprocesoru 6512 firme MOS Technology, čipu koji je po skoro svakom aspektu kompatibilan sa 6502; jedina razlika će biti primetna za one koji dopune računar dodatnim procesorom (280, još jedan 6502 ili 32016) u kom slučaju će čitava kombinacija raditi nešto brže. U sam računar je ugrađen disk interfejs zasnovan ponovo na čipu kompatibilnom sa dosadašnjim kontrolerom 8271. Umesto 32 Kb RAM-a, u novi računar je ugrađeno dvostruko više — 64 Kb. „Višak“ od 32 Kb može da se koristi na razne načine — 20 Kb možemo da rezervišmo za video memoriju, što znači da će osnovni RAM ostati potpuno slobodan čak i u najvišim grafičkim modovima; deo memorije možemo da koristimo kao „sideways RAM“, tj. da u njega učitavamo softver iz raznih ROM-ova (tekst procesor, data base itd); možemo da se opremimo čudnim RAM/ROM kombinacijama koje će verovatno biti vrlo popularne jer će omogućiti da specijalizovani programi koriste privatni radni prostor u RAM-u koji će biti pejdžovan zajedno sa njima; možemo, najzad, komandom da isključimo dodatnu memoriju i tako postanemo kompatibilni sa postojećim BBC B. Acorn, težeći kompatibilnosti, neće koristiti dodatnih 32 Kb za smeštanje programa i podataka ali je verovatno da će nezavisne kuće itekako



koristiti ovaj „dodatni bonus“ za pripremu atraktivnijih igara.

Dodatnom memorijom se ne iscrpljuju razlike između modela B i B+: novi kompjuter ima šest ugrađenih podnožja za ROM-ove (prema 4 kolutih je imao BBC B), od kojih su pet slobodni; šesto podnožje sadrži „OSBASIC“, ROM od 32 kilobajta u koji je smješten operativni sistem i bezik interpreter koji su manje-više identični sa onima koji su ugrađeni u BBC. U preostala podnožja mogu da se smestaju ROM-ovi i EPROM-I 2764 (8 Kb), 27128 (16 Kb) i 27256 (32 Kb), što pruža razne mogućnosti nezavisnim softverskim firmama i samom Acronsoftu.

Cena novog modela je, zajedno sa VAT-om, 500 funti. Suma je prilično nategnuta; ako sadašnji model B dopunite disk interfejsom, RAM tablom od 32 Kb i ROM tablom koja omogućava povezivanje 15 ROM-ova i još 16 Kb „sideways“ RAM-a, platite oko 520 funti i dobiti nešto što je vrlo slično modelu plus; svaki od ove dve mogućnosti ima prednosti i nedostatke. Da je Acorn pripremio ovaj model pre tri godine, napravio bi veliki posao. Ovako — videćemo!

Dejan Ristanović

## Više kuso nego brzo

Vlasnici računara „spektrum“, „komodor 64“ i „MSX“ mogu da nabave nov periferijski uređaj za smestanje podataka, nazvan „Quick Disk“. Proizvođač je firma TRITON, a cena je 129 funti, bez obzira na tip računara. U cenu je uračunat disk dray i, naravno, odgovarajući interfejs za povezivanje.

Quick Disk je zamišljen kao alternativa standardnim floppy diskovima i, razume se, nestandardnom spektromurom mikrodravju. Brzina kojom uređaj radi mogla bi se okarakterisati kao osrednja: nešto je manja u odnosu na standardne diskove, a nešto veća u odnosu na mikrodravju (ukoliko ova Sinklerova sprava uopšte može poslužiti kao neko merilo brzine).

Za čuvanje podataka koriste se diskovi kapaciteta 100 K i promera 76 mm (3 inča), a cena im je oko dve funte.

Čini se da sistem, u odnosu na svoju cenu, predstavlja dobar izbor, bar kad su u pitanju verzije za „komodor“ i „MSX“. Što se tiče „spektruma“, situacija je tu manje jasna. Izvesno je samo to da Quick Disk pruža veću pouzdanost u radu od mikrodravja, dok su ostale performanse tek neznatno bolje. Recimo, naredbe za rad sa ovim uređajem su jednostavnije od naredbi za mikrodravju, ali ni tu prednost nije uvidljiva. Dobra karakteristika je i to što sistemski softver ne zauzima memoriju računara, tako da se većina komercijalnih programa može preneti na Quick Disk, koristeći priložena uputstva.

Najviše, međutim, zabrinjava činjenica da prisustvo Quick Disk-a na „spektrumu“ u velikoj meri ometa rad ostalih periferijskih uređaja. Dovoljno je reći da se od štampača jedino može priključiti zastareli originalni ZX printer!

Pa, ko voli, nek' izvoli.

J. Skuljan

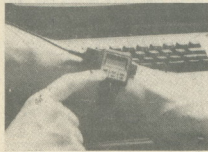
## Ručni mikro

Jedan od najnovijih proizvoda kompanije „Seiko“ sigurno može da ponese epitet najmanjeg kompjuterskog terminala RC-1000 kako se inače ova naprava zove, u suštini je ručni časovnik, sa displejom od tečnog kristala i nekoliko uobičajenih tastera za podešavanje i kontrolu rada. Međutim, pomoću posebnog kabela, RC-1000 se može povezati sa kućnim računarom (za sada, izgleda, jedino se „komodorom 64“ i programirati uz sve pogodnosti kompjuterskog sistema.

Ručni terminal je u stanju da prihvati dva kilobajta podataka sa računara i da ih prikaže na svom ekranu. Kapacitet displeja je 48 znakova (dva reda sa po 24 znaka). Moguće je isprogramirati veliki broj poruka koje će vlasniku biti saopštavanje kasnije, u toku godine, kada za njih dođe vreme (rodendani, važniji datumi, itd.).

Ipak, imajući u vidu cenu od 120 funti, čini se da ovo „zadovoljstvo“ i nije baš najpreče na spisku naših kompjuterskih prohteva. Možda bi za tu cenu „Seiko“ mogao uz svoju spravicu da ponudi i jedan „komodor“ kao periferijski uređaj...

J. Skuljan



6) šta ima novo

## Asembler u ROM-u za „amstrad“

Firma „Amor“ ponudila je vlasnicima „amstrada CPC-464“ hardversko-softverski dodatak „MAXAM“, koji u svom ROM-u sadrži kvalitetan assembler, disassembler i editor. Poziv assemblera se obavlja jednostavno iz bezjika, pomoću proširenog seta naredbi, a omogućeno je i lako prenošenje parametara između bezjika i mašinskih rutina. Uputstvo na 50 strana je kvalitetno, mada nedovoljno razjašnjava neke detalje (recimo, rad sa prekidnim tačkama). Dodatak se jednostavno priključuje na računar i omogućuje sva naknadna proširenja. Cena je nešto niža od 60 funti.

J. S.

## Vita jela, zelen bor — najbolji je „komodor“

Narod se podelio na šminkere i pankere, zvezdaste i partizanovce, a sada imamo i spektromove i komodoriste.

Prvih im je više, ali su zato drugi bolje naučili.

Ovaj rivalitet se preneo i na stranice kompjuterskih časopisa. Ako se, recimo, u nekom tekstu kao uzgred pominje katastrofalni „komodor“ bezik, gotovo je sigurno da autora kod vodećeg „spektruma“, jer niko toliko ne mrzi „komodora“ kao vlasnici „spektruma“. Obrnuto i nije baš izraženo. Komodoristi PREZIRU „spektruma“ kao neozbiljnu masinu nedostojnu čoveka.

Postoje i druge razlike na osnovu kojih možete lako utvrditi ko se s kim zabavlja. Dakle, spektromovci imaju velikih problema da zakumuliraju svoj osećaj inferiornosti, komodoristi su bili daleko samopouzdaniji dok se nisu namočili i prestali da budu ekshibicionisti. Dok su spektromovci uglavnom solidarni komodoristi su „brvi“ na davanje programa. I iz njihovih redova su ponikli najveći pirati koji jedu malu decu.

Na osnovu marke kućnog ljubimca može se ponešto saznati i o vlasniku!

Veliki broj „spektruma“ u našim krajevima rezultat je delovanja poznate tvrtke Šverco-komerc; „spektrume“ gaje oni koji su ga uspešno proširovali ispod kaputa, i oni koji su za male pare hteli da dođu do računara. Uskoro se, međutim, pokazalo da ta mala živuljica ima veliki apetit i da traži sve nove i nove dodatke. Ali, u međuvremenu, spektromovci su se emocionalno vezali za te male monstrume i sad muku muče da dokažu da njihov mali nije ništa lošiji od „komodora“.

„Komodora“ kupuju oni koji u startu mogu da odvoje više love; posle ga, uglavnom, koriste za razmatranje pred neupućenim komsijama. Među komodoristima ima manje hakera, ali je zato za velike pirate zva zgodna.

„Amstrada“ će kupiti oni koji su već šiznuli tipikajući po „spektrumovoj“ gumtenoj tastaturi, a ponos im ne dozvoljava da priznaju poraz i lepo kupe „komodora“ izgovor: oni žele da ostanu verni procesoru Z80.

Posebnu grupu čine zavedeni pa ostavljeni, naivčine koje je neko nagovorio da kupe mišolovke za koje ne može da se nabavi softver. Oni isto tako muku muče da dokažu da se nisu zeznuli.

Profesionalisti koji drže do sebe kupuju BBC, Epl, ili bar neku od njegovih istočnjačkih kopija, koliko da se ne bi izjednačavali sa amatima.

Posebnu grupu čine snobovi koji ne mogu sebi da dozvole da imaju isto što i ostali narod. Za njih su samo ekskluzivni modeli, kao što su Hewlett-Packard, IBM PC ili McIntosh.

U svakom slučaju, postalo je pitanje prestiža imati bar jedan računar u kući. Preko Radio-Sapca uskoro će se čuti i oglašiti ti:

„Domaćin Mitar ponosno javlja komsijama:

„Laje kera, imamo sinkler!“

Njemu odgovara kom Mlilav:

„Vita jela, zelen bor, stig'o novi komodor!“

ili:

„Psuje Mitar, na čelu mu bora, ne mo'ž „spektrum“ pokraj „komodora“!“

Jelena Rupnik





Load „pismo“

Priprema  
Branko Đaković

## Akcija oko „Velike akcije“

Poštovani druze urednice,  
Zeleo bih da kazem dve-tri reci o programu koji je dobio drugu nagradu na vasem Konkursu — „Velikoj akciji“ autora Aleksandra Radovanovic.

U propozicijama, objavljenim u „Galaksiji“ br. 150 od oktobra 1980, stoji da „program ne sme biti objavljen ili distribuiran pre isteka konkursa“. Medutim, ja sam dobio od prijatelja program „Velika akcija“ tri meseca pre nego sto ste objavili rezultate konkursa. Dakle, taj program je vec distribuiran — iako nelegalno, on ipak kruzi među vlasnicima racunara. U propozicijama, doduse, pise da se nece ulaziti u korespondenciju sa komisijom, ali kako ocekivati da se postuje ova, kad se nije postovala i prethodna propozicija? Stoga smatram, pozivajući se na VASE propozicije, da mogu da kazem: Komisija se ogresila o pravila konkursa i odluka o dodeli nagrada ne moze biti ispravna!

Da li je A. Radovanovic nehotice ili hotimice pruzio komisiji netacne podatke? Da li je Komisija namerno prekrasila pravila (jer je autor prilično poznat, ti, vodi skolu manjina u emisiji „Ventilator 2/73 ili se, mozda, desilo nešto treće!

Postovani druze urednice, toliko od mene. Na vama je (i autoru kojeg izuzim) cenim) da ovo pismo demantujete odgovarajucim dokazima. Vise bih to voleo, nego da na jednom tepom i korisnom takmicenju ostane mrtlja. Ako ovo pismo zavrsi u korpi za otpatke, ti, ako mi ne odgovorite na njega, necu vise imati leve misljenje o vama.

Pozdravlja vas vaš verni čitalac  
Čala Zoltan, Maksima Gorkog 61,  
21000 N. Sad

Dragi Zoltane, priznajemo ti da si, na žalost, u pravu... „Velika akcija“, zahvaljujući preduzljivosti domaćih preprodavača, već nekoliko kruži Jugoslavijom. Konkursna komisija, razume se, to nije znala sve dok nisu objavljeni rezultati konkursa. Nakon prvih telefonskih upozorenja, ustanovili smo da se radi o radnoj, primitivnoj verziji programa, koja je nekada davno data jednom prijatelju na poverenje i koja nema više od 50 dosto zajedničkog sa nagrađenim pro-

gramom. Možda je, formalno, i bilo osnovna da se Aleksandru Radovanoviću, ni krivom ni dužnom, odzme pravедno zaslužen nagrada. Pirati su, međutim, pokvarili već toliko stvari — zasigurno im dozvoliti da pokvare i ovo? Učesnici nekog budućeg konkursa će, očigledno, svoje radove morati da čuvaju u trezorima Narodne banke Jugoslavije. Tvoje ime, inače, nisam primetili među učesnicima konkursa. Hvala ti.

## Prava imena za ružne stvari

Drago mi je da sadržajni i pravovremeni interviu, koji su jedna od odlika „Galaksije“, počinju da se pojavljuju i u „Računarima“. A još više mi je drago što se trudiš da to ne bude „pričam ti priču“ interviu. Verovatno počešće da je ovo pišem povodom interviua sa Jozom Dujmovićem, koji je išao u petom broju „Računara“. Bilo bi zaista dobro da i u sledećim brojevima „Računara“ imate po jedan tako dobar i kvalitetan interviu.

Profesora Dujmovića ne poznajem lično (nisam jedan od njegovih studenata), ali cenim njegovu hrabrost da ohrvano kaže neke stvari koje su, inače, dobro poznate u tole stručnijim krugovima, ali se ljudi plaše da kažu šta misle zbog mogućeg zameranja moćnicima iz te oblasti.

Činjenica je da je bezik upravo onaj kako kaže profesor Dujmović, činjenica je da je velika greška nametati bezik kao početni jezik u programiranju svim generacijama daka koje dolaze. Ali, opet, to se događa. I to zato što ljudi koji podržavaju bezik imaju ne samo interesa da to čine nego i moći da to sprovedu. Smatram da je to osnovni kvalitet izlaganja profesora Dujmovića: hrabrost da nazove stvari imenima koje zaslužuju.

Mislim da je profesor u pravu i za jezik kojim treba počinjati. Ja sam veoma brzo sa bezika prešao na paskal i mogu vam reći da je razlika svemirska.

Na žalost, profesor Dujmović je u pravu i što se čine amatera-programera. Do ne tako davno i ja sam razvijavao zaštitu na kupljenim programima i piratizovao ih, ali to je trajalo jako kratko. Kad sam odlučio da probam da i programiram neke moje stvarčice, otkrio sam da nemam više vremena da se bavim kopiranjem tuđih. Na žalost, ubrzo sam i ja bio žrva. Spremam sam jedan program za vaš konkurs, ali kad sam ga završio otkrio sam da njegove radne verzije ima čitavo moje društvo, a verovatno i mnogi drugi. Program nisam poslao na vaš konkurs i to sve zbog mojih drugara koji su se igrali pirata na moj račun. Većina upravo ih programera-amatera-pirata nikad ne stigne dalje od igranja igara i pravljenja programa na nivou predškolskog uzrasta.

Vladimir Nikčević, Beograd

## Jozo, daj vode!

Redakciji Računara,  
Javijam vam se povodom razgovora sa Jozom Dujmovićem, koji ste objavili u petom broju „Računara“. Intervju sa Jozom Dujmovićem je upravo onakvo štivo kakvom sam se najmanje nadao u „Računarima“. Na stranu to što je taj članak pun korisnih podataka i razmišljanja, sam ton profesora Dujmovića, koji se može nazreti iz članaka je toliko svezanjuci i bar neki, argantan da je veoma teško pročitati članak do kraja.

Profesor Dujmović sve zna. On zna zašto bezik ne valja, a naročito zna što neki ljudi forsiraju bezik, on zna kako računarske novine teraju euforiju i senzacionalizam u vezi sa računarima, on u razgovoru čak dva puta pomogne to da kod nas nema „Computer science“ fakulteta, a na Zapadu ih eto ima! Šta sve još ima na Zapadu što nema kod nas? Bilo bi interesantno znati kako profesor Dujmović vidi sebe na nekom takvom fakultetu otvorenom kod nas? Sigurno ne kao vanrednog profesora.

Pri kraju razgovora profesor Dujmović nam razbija iluzije i tvrdi da je samo san to da je naša šansa u softveru. Kaže da bi „... naša devizno preglednata privreda odavno otkrila tu šansu da su stvari takve kakve“ novinarima na prvi pogled pričinjava i da mu predlažem da porazgovara sa mladincima iz Beograda, Zagreba i još par drugih gradova, koji su svoje programe prodali u inostranstvu. Jozo se hrabro obara na igre, jer se one sastoje samo iz lamanjenja marsovača. Druže Jozo, pare koje bi naša „ti o privreda“ zaradila na takvim „primitivnim“ programima su isto tako dobre i vredne kak i pare koje ste ti verovatno dobili za vaš „Data base management“. Isto tako smatram da je veoma sofisticisća vaša tvrdnja da „obzbiljno računarsvo mogu da razvijaju samo obzbiljno pripremljeni profesionalci“. Prilično je očito da se vi smatrate „obzbiljnim profesionalcem“ i da se ni malo ne libite da svima koji bi se računarima bavili objasnite da tu nemaju šta da traže.

Da stvari ostaju samo na profesor Dujmoviću i sličnim „obzbiljnim profesionalcima“, uskoro bismo u jednoj tako važnoj oblasti, kakva je razvoj računara, razvijali tehnokratski element. Stavovi profesora Dujmovića o školstvu neverovatno liče na stavove izvesnog Stipe Suvara. Slučajno?

Slazem se sa profesorom Dujmovićem da bezik nije onakav kakav je predstavljani, ali sve ostalo u tok interviua je toliko prožeto željnom da računara razvijati od običnih smrtnika, da je prosto neverovatno da profesor Dujmović upošte predaje. Zar mu nije žao da deli njegovo znanje sa „amaterima“ i „laicima“? Stvarno sam ožedneo od tog krasnog štiva. Jozo, daj vode!

Drugarski pozdrav,  
Saša Marković, Bulevar Lenjina  
201, Novi Beograd

Ljudi koji imaju šta da kažu imaju najčešće i način kako to govore. Pošto u „Računarima“ nastojimo

da se čuje što više stranih, neminovno je da pojedini „lični“ pogledi iritiraju pojedine čitaoca. Uzgred, ni vaš način nije nimalo naivan. Pišite nam i dalje. Na primer, šta mislite o tekstovima Jelene Rupnik.

## Dodite da se dogovorimo

Draga Redakcijo,

Vi ste treća redakcija računarskog lista koji se javijam povodom iste stvari. Nadam se da se nećete naljutiti što sam se vas setio tek na kraju. Evo o čemu se radi: računarski časopisi upošte neke da objavljuju priloge mladih ljudi koji im pišu. Verovatno je potrebno imati dobru vezu u redakciji pa da ti izade članak. Ja imam urađeni par tema vezanih uglavnom za primenu računara u različitim vidovima medicine. Razumee bih da su mi rekti da loše pišem ili tako nešto, ali redakcije prethodna dva časopisa nisu čak ni pokazale da su moje članke pročitale. Sada se javijam vama. Ako vas članici interesuju, molim vas da i makar javite čak i ako vas članici ne interesuju, molim vas, javite mi.

Vojvodić Danilo,  
Bulevar Revolucije 257, Beograd

Dragi Danilo, ne samo da smo zainteresovani za tvoje članke o primeni računara u medicini, nego je i pozivamo da svratiš do redakcije da se, kako bi to rekli u „Jugobancu“, dogovorimo. Otkrićemo ti jednu tajnu računarske redakcije OBOŽAVAJU mlade saradnike pod uslovom, naravno, da iz maternjeg jezika imamo najmanje dvoje, a neke ne traže čak ni toliko.

## Draga Saveta

Dragi „Računari,“

Javijam vam se da bih dobio savet Ja va se već pola godine spreman da kupim kojim računar. Glavni problem koji imam vezan je za izbor računara. Sakupio sam dovoljno para da ne moram da bavljam najjeftiniji, ali sada sam u situaciji da ne znam šta da kupim, iz računarskih časopisa sam kupio nekoliko ideja, a od mojih drugara koji imaju računare usmeren sam „spektrumu“ ili „komodoru 64“, zavisto šta ko ima. Ja se još uvek divim, šta mi je predloženo? Sad se dosta govori o „amstradu“, ali mi to baš nije sve jasno. Pomozite.

Petar Radovanović, Pančevački put  
bb

Kako bi bilo da sačekate još pola godine. Do tada će biti još puno novih računara, pa će biti zanimljivije za izbor. Toplo vam preporučujemo članak „Hakeri to vole ručice“. Možda ćete tako naći ono što vam je potrebno.

## Možda grešim, al' ja sam takvog dojma

Uredniku Računara,

Ja uopšte ne shvatam zašto vi i svi ostali izdajete časopis o računaru. Možda bi bilo mnogo realnije da ste svoj časopis nazvali Video Igral. Pretpostavljajući da će moje pismo da leti pravo u koš, pa mogu da vam kažem sve što mi je na srcu. Sestorica kojih poznanika imaju računare. Od te šestorice petorica ne razlikuje RAM od ROM-a, ni mikroprocesor od čipa. Njihovo strogo usmereno poznavanje računara je okrenuto ka učitanju igara. Mislim da bar nekoliko o njih ne znamo da se na računaru osim igranja video igara može raditi još nešto. Možda grešim. Nadam se da je tako. Možda zbog takvih „Poznavalaca“ svaki tri broja neke stvari objašnjavate od početka? Ako je tako, onda je najbolje da samo svakog meseca ponovo izdajete onaj mnogo puta doštampani prvi broj, umesto što se mučite da pišete istovremeno i za jadrnike koji noću ude paskali i za „entuzijaste“ kojim je važniji džojstik od disk jedinice i štampača zajedno. Ostavite takve stvari nekim drugim časopisima, a vi se lepo zadubite u to da odete što dalje u računarsnu makar naša industrija pravila galaksije“ siedeših sto godina, mislim na računar. Žao mi je ako izgledam ogorčen.

Pozdrav od Velimira Nešića, 29. Novembra 42, Beograd

Mislili smo da niko od čitalaca ne zna kako se poneki od nas u redakciji osećaju. Sad nam je lakše. Pozdrav od redakcije

## U Jelice oštre rečenice

Poštovani drugovi (ne i drugarice), Povremeni sam čitalac vašeg povremeno lista i, kao što je reč, na početku vam upućujem pregršt pohvala za sve što ste učinili, što činite i što ćete činiti za prosvetno-igromnih narodnih masa i njihovo bezbolno upućivanje u računarstvo, tj. računarsku budućnost, koja prvi put dođe bez našeg učesća. Dakle, ta vaša uloga zaslužuje samo čestitania, ali se slični komplimenti ne mogu podariti za vaše nerazumno ustupanje jednog od ionako malog broja stranica onima koji o pitanjima koja vas list dodiruje nemaju da kažu ništa, i to ništa kazuju na celoj stranici, a ponekad i na većem prostoru.

Reč je, naravno, o tekstovima vaše saradnice (?) Jelene Rupnik, objavljenim u 3. i 5. broju Računara. Izgleda da ste pali u zamku lažne medijske atraktivnosti i da ste, u skladu s tim, dali nezasižuje-

nu priliku jednoj osobi ženskog pola da na kvazi-duhoviti i ironičan način razmišlja (?) o nekim problemima društva koje iz informatičke predistorije sa mukom zakoračuje u rani (takode informatički) srednji vek. No, da bilo bi konkretniji, odmah ću izneti ono što može učiti svaki promućuniji čitalac „Računara“: iz tekstova J. R. proizilazi: a) Da je dotična potpuni, ako ne i kompletni, nepoznavalac tematike kojom se bavi. To što je čitajući Kompijutersku školu „Politike za decu“ i slušajući i gledajući razne Vjestice i Šćekice na radiju i Televiziji spoznala neke pojmove kao što su „bežik“, „kobil“, „sofver“, „disketa“ i slične, čini stvar još težom i opasnijom, jer, kao što kaže profesor Vajzenbaum sa Čuvenog Bita, „ne šteti nam ono što ne znamo, već ono što znamo pogrešno“.

b) I njen kvazi-sociološki a medijski pokušaj pristup pod brljijivom analizom doživljaja potpun i konačan krah. Služeći se metodološkim instrumentarjem koji su odavno odbacili i listovi poput „Bazar“ i „Nade“, vaša saradnica (?) pokušava da objasni probleme jednog društva koji se javljaju prilikom njegovog prelaska iz jedne ere u drugu. Promadnosť njenog pristupa proizilazi iz nepoznavanja nauke kojom pretenduje da se bavi. Možda ću biti malo vulgaran, ali ovom prilikom moram da kažem da se drugarica J. R. razume u sociologiju skoro isto toliko koliko i u kućne računare, dakle, otprilike kao Marica u džojstik.

c) Treć oblast: treće umede koje vaša saradnica (?) pretenduje da osvoji je polje duhovitosti na kome pretrpljuje najkonaciñni neuspah, jer, ako se nezanjan i može sakriti iza nekoliko nakanjenih poja-va i konstrukcija, nedostatak duhovitosti izbija iz skoro svakog reka njezinih umotvorina. Pred opažanjima tipa: „... dečki se zabavljaju puškama i pištoljima dok se nesretne devojčice igraju kuvanja ručka ... „ili“... živ čovek svašta može, a tek živa žena“, ili „računar treba staviti u peglu, u vikere za kosu ...“ ostaje se bez reči.

Da razimiram, za svaku je pohvalu pokušaj da se bude drugačiji, ali ako već hoćete da ocharate nove horizonte, neka ih onda otvaraju oni koji imaju šta da kažu i znaju kako to da kažu. To što Jelena Rupnik pokušava da postane računarska Slavenka lić Drakulic je njen problem. Zašto tu mešati „Računare“?

„Je toliko, hvala na pažnji i do čitanja.“

Saša D. Kovačević, Novi Beograd

Dragi Saša, prepoznali smo te iz „Kvarljivo leto“. Ne znamo da li Jelena izigrava računarsku Slavenku lić Drakulic — vidimo samo da ti hoćeš da budeš Jelena Igor Mandić. Napiši nam nešto lepo za „sedmicu“, Makar to bilo i na „Jelenin račun“

## Dobra duša iz „Računara“

Držte Uredniče,

Javljam vam se povodom nekolicina članaka vaše novinarke Jelene Rupnik. Kada sam prvi put videla članak „Računar je muškog roda“, bila sam zaista iznenađena i nisam baš bila уверена da su takvi članci u skladu sa koncepcijom lista kakvog ga ja znam. Međutim, od tada sam taj članak pročitala još nekoliko puta i drastično sam promenila mišljenje. Sada vidim da taj članak (pored „Pirati trče počasni krug“ od iste autorke) donosi ono što je računarska nedostajala još od prvog broja, a to je izvesna količina duše. Nadam se da nećete metalizovati da me shvatite — radi se o tome da tekstovi Jelene Rupnik otvaraju prostor za razmišljanja o računaru koja se blago dotiču sociologije. Ta je oblast, po meni, gotovo potpuno nedodiruta u najvećem broju časopisa, pa i u „Računaru“.

Vi verovatno dobro znate koliko je vrcavi i duhoviti stil Jelene Rupnik, pošto ste i uvrstili njene tekstove u časopis, ali bih ipak htela da kažem koliko me je iznenadilo toliko otvoren i svež, upravo zato što je duhovit, pristup temama koje su do sad razmatrane sa strogo tehničkog aspekta. Nemojte pomisliti da sve ovo pišem zbog mog posla, zato što Jelena Rupnik u svojim člancima ima blag feministički ton. Smatram da mogu da budem objektivna i čak bih volela da čujem šta neki muškarac misli o tim istim člancima. Mojih par kolega sa fakulteta nisu baš nešto oduševljeni tim člancima, ali ja sam уверена da je u pitanju čisto muška sujeta. Jedino što mi smeta kod članaka Jelene R. je njihova dužina. Oni su gotovo uvek na jednoj stranici. Zašto ne date Jelenu više prostora? Ne znam da li je to zbog toga što časopis preplave prilozii tehničke prirode, ili se plašite nekonzervativnog pristupa vaše novinarke. Što se mene tiče, ja bih najviše volela da u časopisu imate tri Jelene Rupnik i da su „Računari“ prepuni tekstova Jelene R. „Računar je muškog roda“ i „Pirati trče počasni krug“. Verovatno ste i sami to osetili, jer u petom broju „Računara“ se pojavljuje jedan mali tekst o zabludama vezanim za program Elizu. Ali, to ipak nije članak kao Jelenu. Nedostaje nam svežina i neformalan pristup. Kao da je moguće iz članska videti koliko su vam stari saradnici. Po tome izgleda da su vam tehnički saradnici stari po četrdeset godina, autor članka o Elizi oko trideset pet, a Jelena Rupnik oko osamnaest.

Mirjana Mihajlović, Ohridska 3, Zemun

U poverenju, Jelena ima nešto malo više, a svi ostali mnogo, mnogo manje godina. Puno pozdrava i — pišite nam opet

## Računari u izlogu

ENTERPRISE



Računar „enterprajz“ prvi put je predstavljen septembra 1983. U vreme njegove performanse su zvučale gotovo nestvarno, nudeći ljubitelju mikroručunara ono o čemu je mogao samo da sanja. Danas, kada je ova mašina ušla u masovnu proizvodnju, njene karakteristike, istina, više ne zvuče nestvarno, ali i dalje ostavljaju utisak veoma moćnog sistema, koji će se, bez sumnje, uspešno nositi sa ostalim računarsima u svojoj klasi. Da se pojavio onda kada je bio očekivan, „enterprajz“ bi postao računar bez premla.

# preduzimljivost sa zadrškom



Računar „bujnih“ oblina: Enterprajz

Na prvi pogled, „enterprajz“ deluje u najmanju ruku čudno. Neko će možda reći i da deluje sjajno, ali takve komplimente mu, sigurno, neće uputiti korisnik koji je navikao da radi sa profesionalnim uređajima.

Umesto „kutijastog“ oblika, kakav, navodno, imaju ostali računari, tvorci „enterprajza“ su se opredelili za nekakav, rekli bismo, amorfan dizajn, sa puno nepotrebnih krivina i oblina, kao Venera Miloska, tako da sve skupa deluje jako rasplintu.

## Gumče na računče

Tastatura, sama za sebe, predstavlja pravo razočarenje — prilično podseća na Sin-

klerova iznenađenja tog tipa. Mada su sami tasteri oblikovani sa dosta ukusa, mehanizam je rešen na bazi gumene membrane, što, sve u svemu, ostavlja na korisnika prilično „sunderast“ utisak dok kuca po tome.

Udarina karakteristika tastature je, svakako, odsustvo strelca za pomeranje kursora(?). Umesto toga, korisniku je ponuđen jedan fini mali džojstik u desnom donjem uglu(?).

Ne znamo šta je navelo tvorce ovog računara na ovakav, najblaže rečeno, nestandardan dizajn. Danas, kada više nije problem opremiti sistem velikom memorijom i dobrim interpreterom za bejzik, trebalo bi ipak više pažnje posvetiti izgledu računara i njegovoj tastaturi. Već je toliko puta potvrđeno da korisniku daleko najviše znači profesionalni osećaj koji ima dok sedi za mašinom profesionalnog izgleda. Dosta je bilo igraćaka na tržištu mikroročunara! Zašto izmišljati gumene tastature bez strelca, a sa nekakvom palicom-mrdalicom? Profesionalna rešenja nisu baš toliko skupa da bi ih trebalo izbegavati.

Ostavljajući sada po strani spoljašnje efekte, mogli bismo nešto reći o funkcionalnosti same tastature. Nju čine 69 tastera, veoma racionalno raspoređenih, sa izdvojenim grupama funkcijskih i komandnih tastera, u posebnim bojama. Funkcijskih tastera ima osam, ali se, koristeći uz njih i tastere SHIFT i ALT, mogu ostvariti čak 32 različite funkcije.

## Dejv i Nik

Sa zadnje strane računara može se naći veliki broj raznih portova i priključaka. Čini se da nema perifernog uređaja koji se tu ne bi mogao priključiti: dva džojstika (ako vam se onaj na tastaturi ne sviđa), monitor, printer, kasetofon, televizor, memorijsko proširenje, itd.

Unutrašnja organizacija donosi mnogo noviteta. Istina, sistem se bazira na popularnom procesoru Z80, sa klokom od 4 MHz, ali čitava preostala struktura nije baš uobičajena za taj procesor. U osnovnoj verziji računar ima ROM od 32K (samo sistemski ROM-interpretar za bejzik se posebno priključuje kao proširenje ROM-a) i RAM od 64K, ali se u toku ove godine očekuje i model sa RAM-om od 128K. Međutim, jednostavnim proširivanjem, memorija se može dovesti do fantastičnih 4 Mb (4096K). Pitanje je, naravno, kako procesor Z80, koji normalno adresira samo 64K, izlazi na kraj sa 4Mb? Trik je u tome što je adresa magistrala proširena sa 16 na 22 linije, a memorija izdvojena na „stranice“ od po 16K (ukupno 256 stranica). Čitavu brigu oko memorijske mape vodi posebno dizajniran ULA čip nazvan Dejv (Dave).

Generisanje slike kontrolishe jedan drugi

specijalan čip koji se, pak, zove Nik (Nick). Sistem raspolaže sa pet grafičkih modova. Tekst se može ispisivati u 25 redova sa 80 kolona, matricom 8×9. Maksimalna rezolucija iznosi 672×512, a moguće je generisanje čak 256 boja, doduše, pri smanjenoj horizontalnoj rezoluciji.

Posao oko zvuka je opet poveren Dejvu. Mogu se kontrolisati tri tonska kanala i jedan kanal sa šumovima, u stereo kombinaciji i uz mogućnost definisanja frekvencije i jačine tona kao funkcije vremena. Tu je, osim toga, i veliki broj specijalnih efekata koji se mogu dobiti filtriranjem, moduliacijom i izobličanjem zvuka.

## Bejzik za sve

Ipak, najjača strana „enterprajza“ je njegov bejzik. Praktično ne postoji naredba koja je ikada ustrelaba programerima, ili koja je ikada viđena u raznoraznim proširenjima bejzika, a da nije ugrađena u ovaj interpreter. Po formi, taj jezik je daleko bliži paskalu jedino što se ne prevodi, već interpretira.

Velika pažnja posvećena je strukturiranosti programa, uključujući naredbe: IF — THEN — ELSE, DO — LOOP, WHILE, UNTIL, EXIT, itd. Promenljive se mogu koristiti u potprogramima nezavisno od glavnog programa, a čak se i linijski brojevi mogu ponavljati.

Postoji i širok izbor grafičkih naredbi (uključujući ELLIPSE i FILL), pa naredbi za zvuk itd. Teško je zamisliti situaciju u kojoj bi programer osetio potrebu da pređe na mašinski jezik, osim, naravno, ako je nezadovoljan brzinom interpretacije bejzika.

Uz računar se dobijaju i dva priručnika. Prvi uvodi korisnika u način povezivanja sistema i njegovo korišćenje, a drugi se, na oko 230 strana, bavi samim bejzikom. Naravno ništa ne može biti savršeno, pa ni taj priručnik. Grafičke mogućnosti računara su prilično zastopajvene, tako da programeru u tom cilju jedino preostaje da analizira programe sa demonstracijom trake...

Interesantna je pojednostava da „enterprajz“ ima ugrađen tekst procesor, a listing bejzik programa se takođe tretira kao tekst, što omogućuje listanje u oba smera, pretraživanje, prepričavanje itd.

Sve u svemu, radi se o računaru vrednom pažnje. Njegova cena od 250 funti je, po svojoj prilici, realna. Verzija sa 128K koštaće 350 funti, što je, recimo, niže od cene QL-a.

Jedinu nezavisnost sa sada predstavlja programska podrška. Imajući u vidu prilično revolucionarna rešenja u strukturi računara, pitanje je koliko će vremena biti potrebno softverskim firmama da se upuste u proizvodnju programa za ovaj model. Sa nestrpljenjem očekujemo rezultate.

Jovan Skuljan



# računari *Promašaj* izlazećeg sunca *na japanski način*



Onome ko nikada nije imao kućni računar nije lako objasniti problem kompatibilnosti. Kome bi, na primer, uopšte palo na pamet da će razni proizvođači pisanih mašina na različite načine raspoređivati taste na svojim proizvodima jer na taj način troše manje žice? Ili, još gore, da će svaki autor neke knjige izmisliti slova koja se njemu sviđaju i usvojiti svoj pravopis? Kod kućnih računara nije tako: program napisan za „spektrum“ neće (na žalost) raditi na „galaksiji“, kasetu snimljenu na ZX 81 neće moći da učita vlasnik „spektruma“ (i pored toga što je „spektrum“ neposredni „potomak“ ZX 81), a i ako mu to uspe program će raditi samo u slučaju da je (krajnje paradoksalno) sasvim jednostavan i nekoristan! Kako je moglo da dođe do ovakvog stanja?

## Kao jaje jajetu

Da je na samom početku jedna snažna firma moćnom reklamom „pokupila“ tržište

*Zatresla se gora — rodio se miš: Mada je imao izvesnih preduslova da napravi revoluciju na tržištu kućnih računara, MSX se utopio u gomilu prosečnih pokušaja*

kućnih računara, ona bi nametnula svoje standarde, kojih bi se dodnije svi držali. Velike firme su, međutim, interesovali veliki i skupi kompjuteri, pa su prve kućne računare proizveli amateri u malim radionicama. U borbi za tržište, svaka firma je težila da izvuče apsolutni maksimum iz hardvera koji je koristila, ne pomišljajući da se odredne nekih raspoloživih mogućnosti. A standardizacija je upravo to: svesno odricanje od nečega što bi moglo da se postigne radi koordinacije sa drugima.

Problem kompatibilnosti se, pomalo paradoksalno, nije ispoljavao u početku razvoja kućnih računara. Do osamdesetih godina postojali su praktično samo Apple II, TRS 80 i daleko skuplji Hewlett Packardovi modeli. Svaki od ovih tipova kompjutera se zasnivao na drugom mikroprocesoru, pa je bilo praktično nemoguće razmišljati o kompatibilnosti, za svaki od njih je napisano sasvim dovoljno programa, a posao stvaranja tih programa je, u najgorem slu-

čaju, utroštručavan (pisanje verzije za svaki računar), što nije previše loše.

Za poslednje četiri godine se, međutim, na tržištu pojavljuju desetine (možda i stotine) modela kućnih računara koji su apsolutno nekompatibilni jedni sa drugima. Nije problem samo u tome što program pisan za „mekintoša“ neće raditi na BBC B, već i u tome što ova dva računara imaju potpuno različitu filozofiju, potpuno različitu grafičku rezoluciju, potpuno različiti sistem upisa na diskete i što, uopšte, prepravka programa sa jednog na drugi nije formalno sintakna nego logička pa čak i koncepcijska! Uz par časnih izuzetaka kao što je „Hewlett Packard“, nije obezbeđena ni „vertikalna“ kompatibilnost programa — čak i ako stalno kupujete kompjutere istog proizvođača. Posle svake promene modela moraćete da bacite 95% kupljenog i 80% softvera koji ste sami napisali.

Japanske firme su dugo izbegavale tržište kućnih računara, posmatrajući vrlo pažljivo događaje na Zapadu. Početkom 1983. šest japanskih firmi je tražilo od čuvene američke kompanije Microsoft (Japancima softver nikada nije bio jača strana) da njihovim potrebama prilagodi svoj širom sveta poznati bejzik. Rezultat postignutog ugovora je viđen već oktobra iste godine na sajmu u Oskai: MSX serija kompjutera.

## Hardverski . . .

Za osnovu nove serije kompjutera Japanci su izabrali čuveni ali već pomalo zastareli mikroprocesor Z80A koji radi na strogo fiksiranoj frekvenciji od 3.579545 MHz. Ovoj odluci je upućena čitava gomila zamerki, premda ima mnogo onih koji smatraju da je ona bila opravdana. Razlog za Z80A je veliki broj postojećih programa koji će, uz manje ili više problema, biti prilagođeni MSX-u. Jak argument protiv je činjenica da je MSX standard koji treba da traje, a ne nešto što će proći za daninu — dve. Z80 ima mnogo kvaliteta, ali danas ima znatno moćnijih i ne mnogo skupih šesnaestobitnih mikroprocesora. Pri formiranju standarda uvek je bolje izabrati konfiguraciju koja pripada budućnosti nego onu koja pripada prošlosti.

Čitav posao oko generisanja slike poveren je čipu TMS 9918A. To je, zapravo, mikroprocesor specijalizovan za generisanje slike i rada sa videom, koji omogućava grafičku srednje rezolucije 256x192 sa 16 boja i 24 reda teksta sa po 40 slova, uz zauzeće 16 K korisničkog RAM-a. Videće mo da 9918 podržava rad sa sprajtovima i vrlo brzu grafiku u bejziku.

Mikroprocesor je, takođe, oslobođen brige o generiranju tona koja je poverena čipu AY 3 8910, proizvodu firme General Instruments. Ovaj čip omogućava kontrolu tri tonska kanala i kanala za buku. Svaki kanal pokriva opseg od 8 oktava. Na bazi ovog čipa projektovan je i generator tona za „galaksiju“.



Kompatibilnost je najveći problem koji se isprečio na putu razvoja kućnih računara. Novi računari, zahvaljujući novim tehnologijama, postaju sve jeftiniji i sve moćniji ali je programa za njih sve manje jer softverska podrška svakog kompjutera mora da počne od apsolutne nule. Japanci su, zato, smislili ubojiti štos — svi njihovi računari proizvode se po jedinstvenom standardu. Iako je MSX-u proricana sjajna budućnost, nismo previše zurili da pišemo o njemu. Nakon skoro godinu dana na tržištu situacija je znatno jasnija — zatresla se gora rodio se miš.

CANON V-20  
£285.00



„Canon V-20“ je računar izvanrednog dizajna za koji je prizvedeno obilje dodataka. Posebno treba pomenuti da ovaj računar ima dosta video izlaza, tako da ćete lako pronaći način da ga priključite na televizor/monitor koji posedujete. Mane računara su pre svega usmerene na tastaturu: iako su tasteri sa kurzorima neobično pogodni, put koji dirke prevađaju da bi neka funkcija bila detektovana je prekratak za stroge daktilografske standarde. Osim toga, V-20 je relativno skup računar koji ne nudi praktično ništa izvan MSX standarda

JVHC-7  
£279.00



„JV HC-7“ krasi praktično sve dobre osobine „Canonovog“ kompjutera, uz bolju tastaturu (za razliku od Canonove, JVC-jeva tastatura zaista liči na deo jednog obiljnog kompjutera) koja je ipak daleko od savršenstva. Prvi primerci su pokazali dosta problema vezanih za generisanje tona i njegov kvalitet, ali bi moglo da se očekuje da ovi nedostaci budu otklonjeni u budućnosti

MITSUBISHI MLF-80  
£299.00



„Mitsubishi MLF 48“ je standardan MSX računar koji nema nikakvih posebnih karakteristika koje bi vredelo pomenuti. Tastatura mu je bolja nego kod prethodna dva modela ali zato praktično ne poseduje port za proširenja, što predstavlja izuzetno krupan nedostatak. Adut u prilog ovoga kompjutera može da bude cena od svega 250 funti za koju se dobija čak 48 K RAM-a

SANYO MPC-100  
£499.95



„Sanyo MPC 100“ je veoma solidno izrađen računar sa izvanrednom tastaturom i, za razliku od ostalih MSX kompjutera, reset tasterom koji će neobično obradovati one koji planiraju da pišu mašinske programe. Ovi kvaliteti su na žalost, kompenzovani nešto višom cenom i potpunim nedostatkom softvera koji se isporučuje uz njega

**SONY HIT BIT HB-75****£299.00****SPECTRAVIDEO SVI-728****£250.00**

Sony HB-75" je verovatno najjači MSX kompjuter koji trenutno može da se nabavi. Solidno izrađen hardver i profesionalna tastatura (kojoj smetaju mali tasteri za editovanje i nestandardno oblikovani funkcijski tasteri) su dopunjeni aplikacionim softverom koji je upisan u ROM. Po uključivanju računara ulazite u specijalni MSX meni iz koga možete da izaberete adresar, rokovnik ili program za rad sa bazama podataka, uz obaveznu opciju koja omogućava startovanje bežik interpretatora. Uz ove programe i disk jedinicu koja se jednostavno priključuje HB 75 predstavlja idealan izbor za poslovnog čoveka.

"SpectraVideo SVI-728" karakteriše, pre svega, izuzetno niska cena od svega 250 funti za koju se dobija čak 80 kilobajta RAM-a. Ovaj računar, za razliku od svih koje smo do sada pomenuli, ima numeričku tastaturu nezamenljivu za one koji planiraju poslovne primene kompjutera. Dizajn ostataka tastature je, na žalost, prilično nestandardan, što bi moglo da se tvrdi i za izgled čitavog računara. Za ozbiljnije poslovne primene je, na žalost, potrebno učiniti računar CP/M kompatibilnim, za šta će biti potrebno između 700 i 800 funti. U međuvremenu SVI-728 je interesantan za mnoge matematički nastrojene korisnike zbog dobre i brze aritmetike.

**TOSHIBA HX-10****£279.00**

"Toshiba HX-10" karakteriše tastatura koju bismo teško mogli da ocenimo više od ocenom — kod brzog kucanja se dešava da pritisak na taster ne bude registrovan, tasteri za editovanje su nezgodno smešteni a „Backspace" je premali i smešten preblizu dirke ENTER. Dobre strane kompjutera su jedino razumna cena i relativno dobar izbor hardverskih dodataka na tržištu.

Periferiju kontroliše Intelov čip 8255. Pod periferijom se ne podrazumeva kasetni interfejs ni sama tastatura koji se nalaze pod neposrednom kontrolom centralnog procesora. Postoji mišljenje da je velika greška što Z80A nije rasterećen brige o tastaturi uvođenjem još jednog čipa 8255, što bi vidno ubrzalo rad kompjutera (BBC mikroručačari deo svoje velike brzine duguju tome što je jedini posao centralnog procesora da računa).

Mnogim početnicima ovakva filozofija rada kompjutera izgleda skoro nemoguć: kako računar radi kada se sve aktivnosti događaju sinhrono? Odgovor leži u interaptima. Kada god, na primer, nastupi „sudar" dva sprajta, čip 9918A daje maskirani interapt glavnom procesoru. Z80, jasno, ne može da zna odakle je interapt stigao pa proverava kontrolne koji su dodeljeni svim ugrađenim čipovima. Kada otkrije uzrok interapta, mikroprocesor može, pod komandom programa, da preduzme nešto ili da ignoriše ovaj prekid. U bilo kom od ovih slučajeva, 9918A obavlja dalje svoj posao kod mikroprocesora radi nešto drugo, pa se, konkretno, sprajtovi i dalje kreću! Ovako organizovani interapti su sortirani po nekim prioritetima, a za specijalne primene (na primer, rad sa disk jedinicama) se koristi i nemaskirani interapt, koji može da „privuče pažnju" Z80 u trenutku nastanka.

Svi MSX računari imaju profesionalnu tastaturu (mada njen kvalitet varira u zavisnosti od cene računara) koja je, pored ostalog, opremljena sa pet tastera kojima mogu da se dodele proizvoljne naredbe i funkcije, kao i setom dirki za rad sa ekran-skim editorom. Tastatura je „osposobljena" za dodavanje „hiragane" i „katakane" (japanska pisma) i grafičkih simbola u budućnosti.

Za komunikaciju sa računarom se, uz tastaturu, koristi ugrađeni džojстик. Osim ekrana standardnog televizora ili monitora, izlazna jedinica može da bude štampač (uz dodatni „centronics" interfejs) a tu je i kasetofon koji služi kao spoljna memorija

ABS	INKEYS	POKE
AND	NO	POS
ASC	INPUT	PRESET
ATN	INPUT#	PRINT
AUTO	INPUT#	PRINT#
BASE	INSTR	PRINT# USING
BEEP	INT	PSBT
BINS	INTERVAL	PUT SPRITE
BLOAD	KEY	READ
BSAVE	KEY LIST	REM
CALL	KEY ON/OFF	RENUM
CBDL	KEY ON/OFF/STOP	RESTORE
CHR#	LEFTS	RESUME
CINT	LEN	RETURN
CIRCLE	LET	RIGHTS
CLEAR	LINE	RNRG
CLDAD	LINE INPUT	RUN
CLOAD?	LINE INPUT#	SAVE
CLOSE	LIST	SCREEN
CLSE	LLIST	SGN
COLOR	LOAD	SIN
CONT	LOCATE	SOUND
COS	LPOS	SPACES
CSAVE	LFPRINT	SPC
CSNG	LFORMAT	SPRITE ON/OFF/STOP
CSRLIN	LPRINT USING	SPRITES
DATA	MAXFILES	SQR
DEF DBL	MERGE	STEP
DEF FN	MIDS	STICK
DEF INT	MOD	STOP
DEF SNG	MOTOR	STOP ON/OFF/STOP
DEF STR	NEW	STR
DEF USER	NEXT	STRIG
DELETE	ON	STRING ON/OFF/STOP
DIM	OCTS	STRINGS
DRAW	ON GOSUB	SWAP
ELSE	ON GOTO	TAB
END	ON ERROR GOTO	TAN
EQV	ON INTERVAL GOSUB	THEN
EIO	ON KEY GOSUB	TIME
ERASE	ON SPRITE GOSUB	TO
ERL	ON STOP GOSUB	TROFF
ERR	ON STRIG GOSUB	TRON
ERROR	OPEN	USR
EXP	OR	VAL
FIX	OUT	VARPTR
FOR	PAD	VDP
FRE	PAINT	VPEEK
GOSUB	PDL	VPOKE
GOTO	PEEK	WAIT
HXS	PLAY	WIDTH
IF	PLAY	XOR
IMP	POINT	

sničajnici programe i podatke brzinama od 1200 ili 2400 bauda (po izboru korisnika) uz korišćenje standardne frekventne modula-cije. Izbor brzine je dobro primeren RAM-u koji, po standardu, mora da ima 8 K (ne računajući 16 K za video memoriju), mada se MSX računari uglavnom grade sa 32 Kb RAM-a.

Pošto je hardver standardizovan, tvorci MSX-a su odlučili da stvore i standardna proširenja. U prodaji je napre kartica koja omogućava pisanje 80 slova u redu i tako otvara vrata primene MSX-a za obradu teksta. Tu je, zatim, RS 232 interfejs, interfejs za povezivanje sa disk jedinicama (realizacija interfejsa je ostavljena pojedinim firmama ali je bitno da format bude kompatibilan i da se koristi isti DOS koji se, naravno, zove MS DOS) i „centronics“ interfejs za povezivanje sa štampačem, koja smo već pomenuli.

Hardverska koncepcija MSX-a je, dakle, bogata i moćna, ali u njoj nema ničega posebno originalnog ili bitno novoga. Verovatno su u pravu oni koji kažu da od standarda treba očekivati generalizaciju postojećeg, a ne uvođenje novoga.

### ... i softverski standardi

MSX bejzik zauzima ROM od 32 K i predstavlja, kao što se moglo pretpostaviti, klasičnu implementaciju „Majkrsoftovog“

bejzika. Nedostaje mu, dakle, grupa naredbi za rad sa procedurama i komplikovanim funkcijama i mnoge strukture koji su današnji superbejzici „nasledili“ od Paskala. Pomalo je paradoksalno da jezik koji treba da predstavlja standard za programere ne podržava danas opšte prihvaćene programerske norme nazvane strukturirano programiranje.

Klasičan „Majkrsoftov“ bejzik je prilično slab kada se radi sa grafikom i ne podržava sprajtove, pa su za ovu priliku na njemu izvršene značajne intervencije. Gleajući spisak naredbi, međutim, ne vidimo mnogo „noviteta“, što ne znači da oni nisu skriveni. Uzmimo, na primer, naredbu SCREEN čiji je puni format:

SCREEN <mod>, <veličina sprajta>,  
<ključtastra> <brzinapisanakasete> <opcija printera>.

Jednoj naredbi je, dakle, dodeljeno nekoliko značenja od kojih neka (brzina kasetofona, na primer) čak nemaju nikakve veze sa njenim imenom! Dok početnika ova nešta nepotrebno zbunjuje, iskusnijem programeru ova mana ne smeta — važno je da argumenti naredbe budu poredani tako da se češće korišćeni (i kucani) nalaze na početku što je Microsoft dosledno poštovao.

Naredbe za rad sa ekranom su, sve u svemu, veoma moćne i njihov set je kompletan. Posebno su interesantne naredbe ON SPRITE GOSUB i SPRITE ON/OFF/STOP čiju neverovatnu korisnost uočavate tek kada počnete da pišete akcione igre na bejziku.

Rad sa generatorom tonova je odlično rešen i „humanizovan“ u daljku većoj merni nego kod BBC mikrokomputera i IBM PC. Naredba PLAY definiše čitav muzički jezik pošto iza nje sledi niz kontrolnih kodova koji omogućavaju kako jednostavno kodiranje melodija tako i kompletnu kontrolu anvelope zvuka. Uloženo je dosta napora da „rečeniča“ uoženja PLAY liči na zapis u notnom sistemu, pa je čak dozvoljeno i korišćenje povisilica!

Iako se MSX računari prodaju najviše kao video igre sa pridohtom bejzikom, matematički nastrojani čitači ovih redova će sa zadovoljstvom uočiti naredbu DEFDBL u našem spisku. Primerenom ove naredbe definiše se promenljive u dvostrukoj tačnosti što znači da MSX računari mogu da vrše aritmetičke operacije sa 16 tačnih cifara. Namerno kažemo „aritmetičke operacije“, pošto „Majkrsoft“ već godinama tvrdoglavo odbija da napiše programe za izračunavanje 16 tačnih cifara elementarnih funkcija:  $\sin^{-1}$ ,  $\cos^{-1}$ ,  $\tan^{-1}$  i t. d. 16 cifara od kojih će prvih 8 biti tačne, dok preostale biti besmislene (na žalost, ne nule!).

Izgleda da nije bilo lako realizovati rad sa sprajtovima jer je softver očito „zbljan“ u ROM od 32 K. Tako su, kao i u svim sličnim prilikama, „stradale“ poruke o greškama koje su svedene na kodove 0-255 što je, kako za početnika tako i za iskusnog programera, prava katastrofa! Ukoliko vaš kompjuter napiše, na primer, ERROR 27, moraćete da pronađete vaše Uputstvo u kome ćete pročitati da je u pitanju „Unprintable error“!

Suprotno lošim japanskim običajima, dokumentacija koju dobijate uz MSX kompjuter je odlično napisana, uz odlično izabrane primere namenjene početnicima i izvode koji će biti sasvim dovoljni za one koji već imaju iskustva sa „Majkrsoftovim“ bejzikom. Na zapadnom tržištu se, paralelno

laganom prodoru MSX standarda, postepeno pojavljuju popularna i profesionalna literatura, koja, je, uglavnom, pisana na visokom nivou. Korist od standarda je i u tome što listing ROM-a, mapa memorije i spisak sistemskih promenljivih nisu nikakve tajne, premda se, jasno, od korisnika očekuje da pri pisanju mašinskih programa sledi određene standarde i ne piše kod tako da bude nekompatibilan sa nekim drugim MSX računarom.

### Hoćemo li standard...

Konačan zaključak o MSX-u treba da pruži odgovor na dva pitanja: da li nam je potreban standard i da li je MSX dovoljno dobar da bude taj standard. Saglasno uvodu ovoga teksta, reklo bi se da je odgovor na prvo pitanje apsolutno potvrđen. I pored toga, na engleskom i američkom tržištu MSX računari (i standardi uopšte) nemaju veliku produ u razloga koji postaju jasni tek posle zreljeg razmišljanja.

Pretpostavimo da je MSX standard usvojen širom sveta. To znači da bi računari svih proizvođača bili međusobno potpuno kompatibilni. No, zašto je tada potrebno da postoji više proizvođača? Oni trače vreme svojih inženjera na proizvodnju kompjutera-klonova sa originalnim hardverskim rešenjima! Osim toga, računari moraju da napreduju, a MSX da preraste u neki „moćniji“ standard. Ako su svi računari isti, novi standard neće nikada biti prihvaćen, jer neće moći da stekne popularnost!

Jedno od mogućih rešenja oba problema bi bilo da se proizvode računari koji će slediti MSX standard, ali i posedovati dodatne mogućnosti kao originalni doprinos pojedinih proizvođača. Ovakvo rešenje je, na žalost, samo prividno dobro: ili će autori softvera koristiti te dodatne mogućnosti ili neće. Ako ih ne koriste, niko nije morao ni da ih stvara da bi figurisale u literaturi. Ali ih koriste, gubi se kompatibilnost i sve se vraća na početak.

Filozofiji proizvođača kompjutera sa Zapada daleko je bliži drugi put rešenju problema kompatibilnosti: borbom će se na vrh tržišne piramide uspeha popeti jedna firma (ne treba biti mnogo pametan da ne bi pogodilo koja) koja će nametnuti standard i morati stalno da se bori protiv manjih firmi koje će nuditi svoje alternative. Tako će i vodeća firma morati neprekidno da unapređuje svoj standard čuvajući „vertikalnu“ kompatibilnost. Autor ovog teksta smatra da postojeće između različitih i različitim tržištima namenjenih operativnih sistema (CP/M, Unix, Xenix, QDOS i slični) predstavlja zdravije rešenje problema od uvođenja jednog sveopštog i apsolutnog standarda.

### ... i da li da to bude MSX

Kada bismo usvajali neki hardversko-softverski standard, MSX bi bio dobar kandidat što ne znači da on ne bi mogao da bude i bolji. Kreirati sveopšti standard je previše odgovoran posao da bi mogla da ga obavi jedna ili pet firmi; potrebno je da MSX računari budu na tržištu i u rukama ogromnog broja korisnika da bi se videlo šta u njemu treba menjati.

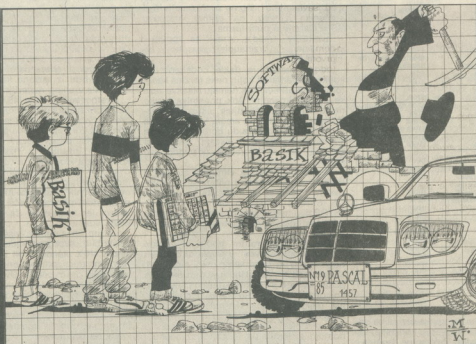
Japanski proizvođači tehničke robe (u našem slučaju računara) su uvek bili poznati kao vrlo fleksibilni i otkrenuti tržištu. Ne sumnjamo da će u bliskoj budućnosti MSX norme biti revidirane na putu prema uvek nedostiznom savršenstvu.

Dejan Ristanović

# neki čudni klinci

Računari  
u razgovoru

ilustracija: Miša Marković



Problemi koje je prof. dr. Jozo Dujmović pokrenuo u svom razgovoru sa stručnim saradnikom „Galaksije“ izvanredno su značajni i aktuelni u našem sadašnjem računarskom trenutku. Iстина, oni su bili još značajniji pre nekoliko godina, ali se tada dr. Dujmović i njegove kolege, ljudi od struke, nisu oglašili. A možda i jesu nekim (polu)otvorenim pismom (ili šapatom) prosvetnim ili drugim vlastima koje su ostale gluve i nisu nam prenale njihovu (polu)otvorenu misao.

No, bilo kako bilo, veoma mi je žao što nisam došao do pomenutog pisma koje je dr. Dujmović uputio našim prosvetnim organima, mada sam pokušao da ga dobijem. Još više mi je žao što do takve inicijative — do kritičkog razmatranja odnosa naših prosvetara prema jednoj izvanredno značajnoj naučnoj i tehničkoj disciplini — nije došlo pre 10 ili 15 godina već sada, jer to može da uzrokuje nova kašnjenja u odnosu na one sredine koje o važnim stvarima misle na vreme.

Koliko je meni poznato, za razvoj „YU računarskog pokreta“ najzaslužniji su zaljubljenici-amateri, popularni časopisi i

— nebudni carinici. Što se državnih organa tiče, oni su nešto više kasnili od ljudi iz računarskog esnafa. Tek nedavno ublazili su carinska ograničenja oko uvoza kućnih računara, pokrenuli inicijativu za uvođenje računara u škole i dr. Na drugoj strani, profesionalci koji znaju šta su računari, jezici koje računari koriste i njihove raznovrsne primene, spokojno su u miru svojih laboratorija i katedri razmišljali šta im valja činiti oko nekih manje praktičnih problema od spuštanja računarstva u narod.

## Nešto „naopačke“

I tako smo došli do neobične situacije. „Galaksija“ pali „računarska vatru“, a amateri (umesto profesionalaca) poput Voje Antonića (tvorac „galaksije“) i Mire Kocijana (tvorac „orla“), prave prve domaće kućne računare. Malo bolji poznavalac programerskih krugova kažaće vam da i među programerima glavnu reč vode, uslovno rečeno, „amateri“ — fizičari, inženjeri tehničke fizike, elektroničari i drugi. Dok programeri od struke čekaju na zaposlenje na berzi rada, „deca“ drže kurseve o programiranju vršnjacima i sedim glavama, a učenici uče svoje učitelje i nastavnike po školama! U toj opštoj inverziji stvari, niko živ ne vidi nešto što bi se moglo nazvati akcijom strukovnog saveza, a zato, priznaćete, najmanje su krivi hakeri, i dok se jedni

igraju računarima, drugi se igraju budućnošću računarstva kod nas. I ne samo njega.

I onda, odjednom, kao svi se prenu! Neki pišu predloge, neki proteste, a većina čeka. Hajde da se dogovoramo! Šest republika i dve pokrajine — to bi moglo da znači najmanje 8 domaćih ili, možda, 8 stranih računara. A tu su problemi materijalne prirode, obaveštenosti i dr.

Znajući to, mislim da je akcija koju je poveo prof. Dujmović protiv uvođenja računara „galaksija“ u naše škole u ovom trenutku pogrešna. Nesumnjivo je da od „galaksije“ ima boljih računara. Ali gde su ti računari? Kako oni treba da izgledaju, ko treba da ih napravi a ko da ih proizvede? Ne lipši magare od zelene trave!

## Računari na trepi

Čitaoci časopisa „Galaksija“ i njenog specijalnog izdanja „Računari“ znaju sve domaće kućne računare. Njih nema malo na broju, ali je, na žalost, retko koji od njih proizveden u „tiražu“ većem od 100 komada. Jedino je „galaksija“ prešla „kritični broj“ od 1.000 (prodato je preko 6.000 računara ovog tipa, što u delovima, što kompletniranih). Taj je broj vrlo mali prema broju prodatih „spektruma“, na primer, ali je astronomski u odnosu na druge domaće



**U prošlom broju „Računara“ objavljen je razgovor stručnog saradnika „Računara“ Nevenke Spalević sa dr Jozom Dujmovićem, vanrednim profesorom za računarsku tehniku i informatiku Elektrotehničkog fakulteta u Beogradu, pod naslovom „Sesnaestogodišnjaci za računarom — mladi genijalci ili izgubljeni generacija“. Razgovor se tiče računarskog opismenijavanja kod nas, pitanja izbora računara koji bi se koristio u osnovnim i srednjim školama, uloge amatera i profesionalaca u oblasti računarske tehnike i drugih aktuelnih problema vezanih za računare i njihovu primenu. Ovaj razgovor izazvao je polemična reagovanja naših čitalaca. Prvi se za „reč“ javio dr Vladimir Ajdajić, nuklearni fizičar iz Beograda, dugogodišnji saradnik „Galaksije“.**

## Računar na tavanu

Istraživanja koja je za proizvođače računara izvela Jenki grupa iz Bostona, SAD, pokazuje da je u trenutku kupovine 92% kupaca zainteresovana za kupovinu računara „da bi naučili jeziki — jezik budućnosti“. Međutim, samo 6 meseci kasnije 41% onih koji su kupili računare koriste ih samo za igre, a 42% za različite poslove koje obavljaju uz pomoć gotovih komercijalnih programa. U tom vremenu 8% korisnika već su „bacili svoje računare na tavan“, a 9% posle 6 meseci druženja sa računarima tvrde da su u stanju da sami pišu programe.

računare. „Galaksija“ je, nesumnjivo, prihvaćena, njena programska podrška iz dana u dan postaje sve bolja (konačno, i na njenom razvoju se učili), a i cena joj nije previsoka.

Na drugoj strani, dr Dujmović bira PMP koji je u Institutu „Jozef Stefan“, u Ljubljani, razvio ing. Marjan Miletić. Ovaj računar po svojoj ceni bio bi desetostruko skuplji od „galaksije“ (zahteva dva diska i terminala), pa time gubi svaku šansu za širu primenu u školama. PECOM Elektronske industrije iz Niša koristi totalno zastareli mikroprocesor COSMAC 1802, koji je za neke vojne svrhe dobrodošao jer može da radi u opsegu od -50 °C do +150 °C. To je važno u slučaju rata, ali je za primenu u školi. Drugi domaći računari postoje samo na papiru, pa o njima i ne vredi govoriti.

Znači, ostaje nam još alternativa da prihvatimo Dujmovićevu da: „dobar čarobnjak može od žabe da napravi princezu“ i da verujemo da će niko dinar da pretvori u dolar. Pa da, onda, krenemo u kupovinu pravog računara za malo „žaba“!

## Teorija ili praksa

Osnovni problem prof. Dujmović vidi u izboru računara i, u vezi s njim, s programskim jezikom kojim će se suočiti budući učenici računarstva u našim školama. Ja smatram da to nije glavni problem, već da je on u obučavanju učitelja i nastavnika za poziv predavača odgovarajućeg predmeta.

Kandidati za ovaj poziv se nalaze u sadašnjim učiteljima, nastavnicima matematike, fizike i osnova tehnike. Svako ko lole poznaje naše školstvo, znaće koliko su to druge stvari presudnije od izbora jezika ili nekog pseudojezika! Da sve ne i ostalo na teoriji, već da bi nešto postalo i praksa, trebalo bi što pre dovesti buduće učitelje u dodir sa računarima, jer sam sklon da verujem da će njima trebati više vremena

da savladaju novu „veštinu“ od njihovih učenika.

„Galaksija“ se možda poredi s jednim savremenim „komodorom“ ili „astradrom“ kao „fića“ s „mercedesom“. Ali, u prvom koraku najvažnije je preći iz taljiga (od abakusa u našoj školi) u auto (na računar), pa makar to bio i „fića“ („galaksija“). Jer mi želimo da naučimo vožnju, a ne da u njoj, još „nepismeni“, učestvujemo u trkama „formule jedan“.

Mi želimo da razvijemo generacije koje će biti sposobne da koriste računare, a ne specijaliste za programiranje. U tom pogledu može nam pomoći analiza naših suseda, Italijana. Prema njihovim istraživanjima, od 100 mladih ljudi koji se prihvataju kućnih računara, njih 70 ostavlja računare kada se naigra raznih igara. Samo 30 ostaje trajno privrženo računarima, pokazujući spremnost da usavršava svoje znanje, savladuje nove jezike i stvaralački radi u ovom polju. Verujem da u odnosu na „galaksiju“ bilo koji drugi računar s kojim bi se deca susrela u školi ne bi mogao značajno da utiče na broj i znanje ovih računarstvu najsklonijih mladih ljudi koji u sebi nose glavnu motornu snagu, želju i moć da se bore sa izazovima novog vremena.

## Profesionalci i amateri

Prenaglašeno reagovanje prof. Dujmovića na amateru — zaljubljenike u računare — podsetilo me je na jednu priču o popu koji se naljutio na svoje seljane zato što su u selu u njegovom odsustvu podigli crkvu. Zato nisam sklon da jednostavno zaboravim da, pored stotina onih koji zaslužuju prekore koje im Dujmović upućuje zbog prisvajanja tuđih programa, težnje ka brzji i lakoj zaradi, upućanja u tuđe banke podataka i sl., postoje i hiljade poštenih entuzijasta koji strazerno svojim sposobnostima i interesovanjima ostaju kraće ili duže, neki, čak, i trajno uz računare. Njima, u svakom slučaju, dugujemo jednu zahvalnicu — onu za otrežnjenje društvene zajednice, koje je, konačno, zbog prisutnosti tako brojnih propagatora računarstva, shvatila da mora nešto da preduzme i prikloni se duhu vremena. Možda je velika šteta što i u drugim oblastima ne može da proradi magična moć igara, jer bi i one postale šire poznate i priznate.

Profesor Dujmović s pravom traži meru u ocenjivanju postignuća amatera. Mi smo skloni da novim stvarima pripisujemo daleko veća značenja od onih koje uzujemo za stvari od juče. Ali ako hvalimo jednog „sedamnaestogodišnjaka koji završava svoj treći model kućnog računara“ (koji je, prema Dujmoviću, ekvivalent radio-amateru iz pedesetih godina koji sklapa prijemnik za pet radio-lampi), mi to činimo i zbog toga što takvih amatera nema mnogo, što bismo želeli da se njihov broj poveća; jer to se ne

može postići postavljanjem nedostiznih ciljeva.

## Sviraiči i muzičari

Ne mogu se složiti s prof. Dujmovićem da će naši potencijalni violinski virtuozii postati „kafanski svirači“, da će biti „izgubljeni računarska generacija koju će 21. vek iznenaditi koncepcijama veoma različitim od jezika i današnjih dečjih računara“. Ako bi se to i desilo, pre bi za to bili krivi oni koji su potpuno prevideli sadašnje tokove računarstva, od amatera koji bi posedovali pogrešne računare ili učili jezik.

Profesor Dujmović je u pravu kada programiranje poredi sa veštinom sviranja na violini. Ono, zaista, zahteva solidan, dugotrajn i strpljiv rad na ovladavanju „tehnikom“, kao i svakodnevnu vežbu. Ali, tu je profesor nešto zaboravio — neko mora da piše „muziku“, da brine o mladim „muzičarima“ i to na svim nivoima njihovog školovanja. A za to nije samo dovoljno znanje, potrebna je i dobra volja. A ona kod nas, pored upornosti, ponajviše nedostaje.

Pored dobre volje nedostaje i odgovarajuća organizacija. Ne bismo mogli reći da se problem uvođenja računarstva u naše škole može rešiti na delegatskom principu ili da je to stvar samo prosvetnih organa. Ovaj problem, kao i drugi njemu slični, zahtevaju angažovanje najvrjnih stručnjaka različitih profila. Njega ne može rešiti jedno „ad hoc“ formirano telo, već on mora postati briga najviših obrazovnih, vaspitnih i naučnih institucija, uključujući tu i akademiju nauka.

Profesionalci obično izbegavaju da rade sa masama, ali od njih traže profesionalnost. I popularizacija nauke i tehnike čini im se sumnjivo, jer ona je nužno „prizemlja“ od vrhunskog bavljenja naukom i tehnikom. Zato se profesionalci iz novijih, izglednijih disciplina obično oglašavaju kritikom, a manje konkretnim predlozima o tome šta treba raditi. Postoje i izuzeci. U određenoj meri jedan od njih je i prof. Dujmović, koji je u razgovoru sa Nevenkom Spalević dao nekoliko izvanrednih opaski u vezi ozbiljnosti s kojom moramo prići uvođenju računarstva u naše škole i njegovom daljem razvoju.

Vrednost Dujmovićevog istupanja, i pored izvesne nadmenosti koju je on kao profesionalac pokazao prema „novopridošlacima“ na računarskoj sceni — amaterima, leži u kritičkoj analizi jednog fenomena našeg vremena koji još dovoljno dobro ne razumemo. Zato ono može poslužiti kao podloga za obuhvatnije spoznavanje mera koje moramo preduzeti da bismo ovladali novom vrstom pismenosti — računarskom pismenošću. U suprotnom, možemo još jednom ostati slepi kod očiju.

Dr Vladimir Ajdajić

# hakeri to vole vruće

## Izbor računara

Izbor računara i periferijske opreme predstavlja, bez sumnje, još uvek izuzetno aktuelnu dilemu ne samo kod onih koji se nalaze pred kupovinom svoje prve mašine nego i kod svih onih kojima je kompjuter na tuđem stolu uvek — slađi. Do sada smo nekoliko puta davali savete kako izabrati „pravi“ računar. Jedan mladi saradnik računara — koji je nedavno i zvanično predstavljen na televiziji kao tipičan YU haker — gleda na čitavu stvar nešto malo drugačije. Objavljujemo i njegov radikalni savetnik u uverenju da ipak neće steći previše — istomišljenika.

### Procesor

Izbor kućnog računara obavezno počinje pogledom u tehničke podatke. Od životnog je značaja da nova mašina ima dobar i brz procesor. Recimo, Z80A na 4 MHz, 8086 na 8 MHz, 68000 takođe na 8 MHz ili, još bolje, 68020 na 16 MHz — u obzir nikako ne uzeti 6502 na 2 MHz a još manje 6502 na 1 MHz.

### Tastatura

Pošto se uverite da je procesor snažan a frekvencija visoka, bacite pogled na tastaturu. Loša tastatura, naročito gumenja, jednostavno deprimira korisnika — ma koliko bili zagriženi, ona vam neće dopustiti da provodite 25 časova dnevno za računarom. Pristojna tastatura obično ima izdvojene tastere za vođenje kursora i numerički set. Naravno, vrhunac svega je izdvojen heksadekadni set.

Pazite samo da kutija ne bude slučajno zelena, bordo, zebasta ili crna sa crveno-žuto-zelno-plavim prugama u donjem dešnom uglu. (uh! rugobe).

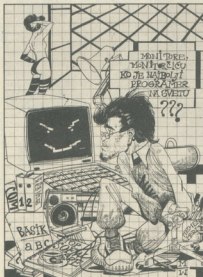
### RAM i ROM

Velika memorija obično olakšava posao programera: 64 K je pristojno, 128 K je dobro, 512 K je vrlo dobro, a 4 giga bajta ideal kome treba težiti (naknad se ne zna čemu sve može da zatreba). Zamislite samo ovo: probudite se jednog popodneva i kažete sebi: „Do ujutro treba da napišem gigabajt čistog mašinalca!“

ROM je, sa stanovišta dobrog hakera, potpuno nevažan, iz prostog razloga što se u njemu, po pravilu, nalazi bejzjik. Ipak, ako nemate disk jedinicu i ako se u ROM-u nalazi dobar i snažan operativni sistem ili, još bolje assembler i disassembler onda to može da bude presudno pri izboru.

### Hardver

Otvorite četvoro očiju! Nepouzdan hardver može da ostavi teške posledice na zdravlje pravog hakera. Ne verujete? Još uvek se nisam oporavio od šoka kada mi se



Ilustracija: Miša Marković

svojevremeno pokvario TI-99/4A. Ko zna gdje bih sada bio da mi se to nije dogodilo — možda bih i ja pisao za Engleze.

### Softver

Suprotno mudrovanju „pametnih“ glava, kvalitet softverske podrške uopšte nije bitan. Važno je samo da postoje dobar assembler i dibager. Svaki haker koji drži do sebe neće raditi ništa drugo osim pisanja mašinskih programa. Bejzjik, razne usluzne programe, igre, a naročito paskal ostavite konjoslavima koji ne prave razliku između bita, bajta i nibla.

Dobar assembler, između ostalog mora da ima ekranski editor i mogućnost definisanja macro naredbi, a dibager obavezno FRONT PANEL DISPLAY i mogućnost izvršavanja mašinskog programa korak po korak. Nema većeg zadovoljstva od debugiranja programa pomoću FRONT PANEL DISPLAY-a, pa pravom hakeru zaigra srce čim ga vidi na ekranu.

Ako, kojom nesrećom, ne raspolazete assemblerom i dibagerom navedenih mogućnosti, setite se da su se nekada kompu-

teri programirali prevezivanjem čitavog hardvera, i budite srećni.

### Disk jedinica i printer

Dva hard diska od po pedesetak megabajta su ono što treba svakom hakeru. U nedostatku, za početak može da posluži i jedan disk od 100 K. Dobar osamdesetkolonski matični štampač takođe nije na odmet. Ako vam to finansijske mogućnosti dozvoljavaju, STROGO IZBEGAVAJTE mikro drajvove disk jedinice koje rade brzinom kasetofona i, uopšte, sve što na sebi nosi naziv „SINCLAIR“ ili, još gore, „COM-MODORE“.

### Sitne gluposti

Od ostalih koještarija, grafika visoke rezolucije nije loša, ali su boje potpuno nepotrebne (idealno je imati zeleni monitor). Osamdeset slova u redu je vrlo lepo, ali ne mora da bude presudno pri izboru — mašinski program se obično sastoji od kratkih naredbi koje se pišu jedna ispod druge.

Trokanalni generator zvuka, pa čak i običan biper, predstavlja poslednju stvar koja treba da vam padne na um, a džojstike, svetlosne olovke, miševe, i ostale drangulije slobodno prepustite devojčicama.

### Prenosivost

Prenosivi računari nisu loša stvar. Preporučuju šokove koji nastaju kada morate da se ranastane, makar samo na pet minuta, od svoje mašine. Uz to, haker doživljava veliko zadovoljstvo svaki put kada stavi kompjuter pod jastuk. Na žalost, portabil mašine su obično nešto slabijih karakteristika i nešto više cene.

### Izbor je vaš

Najzad, koju mašinu izabrati? Ako ste skromni (a naročito skromnih mogućnosti), opredelite se za SPECTRUM (sa profi tastaturom, naravno) ili AMSTRAD. Manje skromnim savetujemo MEMOTECH MTX512, QL, CRAY 1, MACINTOSH (sa 512 K i bez onog pacova), HP-110, APRICOT, OLIVETTI M24...

Ako mene neko pita (pažite, ja uopšte nisam skroman!) najradije bih uzeo IBM sistem 370 sa dvadesetak terminala — jedan u kuhinji, jedan u hodniku, jedan u kupatilu i jedan na svaki kvadratni metar u mojoj sobi.

# monitori

## Periferijska oprema

Vremena u kojima smo radili sa ZX81 i njegovom rezolucijom od 3000 tačaka su, niko to ne spori, danas samo deo romantične istorije razvoja kućnih kompjutera — pojavili su se neki novi računari prihvatljivih cena koji omogućavaju veoma finu — grafiku —  $320 \times 256$  ili  $640 \times 256$  tačaka i rad sa par stotina boja i nijansi! Ovakve karakteristike ne vrede baš mnogo ukoliko ne raspoložemo ekranom na kojima bismo ih videli u punom sjaju — ekranom monitora. Monitor, na žalost, nije uređaj koji bismo nabavili „da kompletiramo sistem“ — njegova cena lako može da prevaziđe cenu samog računara. Moguće je, međutim, da već posedujete monitor, a da i niste svesni ove srećne činjenice — većina stranih kolor televizora koji su proizvedeni poslednjih godina imaju takozvani monitorski ulaz! Pre nego što shvatimo kako da upotrebimo ovaj ulaz, moraćemo da se pozabavimo osnovama funkcionisanja televizora i monitora i da upoznamo razlike između RGB, kompozitnog videa i RF ulaza u televizor.

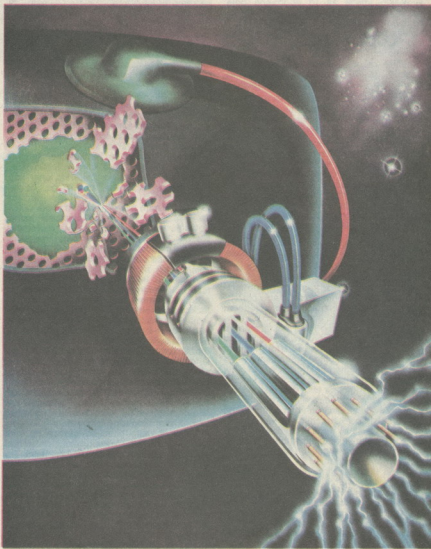
Zajednički imenilac svih monitora, televizora, osciloskopa i drugih uređaja koji imaju ekran je katodna cev ili CRT. Ona je zasnovana na osobinama nekih vrsta fosfora koji svetle kada na njih usmerimo elektronski mlaz. Zamislimo jednu staklenu površinu prekrivenu fosforom na koju je upravljen „top“ koji izbacuje elektronski mlaz. Sve tačke u koje ovaj mlaz pogodi će svetleti! — pomerajmo „top“ tako da skanira čitav ekran i sve će tačke na njemu biti sjajne. Ukoliko se „top“ pomera dovoljno brzo, naše oko neće biti u stanju da registruje gašenje pojedinih tačaka, pa će nam se činiti da svetli čitav ekran!

### Elektronska artiljerija

Da bi obišao sve tačke, elektronski top se kreće po cik-cak liniji: pošavši od gornjeg levog ugla, top iscrta 313 linija i stiže u donji desni ugao; zatim sledi povratak sličnom putanjom i ispisivanje 312 linija koje se nalaze između prvih 313, čime je postupak završen (američki TV standard je nešto drugačiji što YU vlasnicima američkih računara izaziva mnogobrojne probleme).

Pretpostavimo da je neko stavio kartončić kvadratnog oblika između topa i površine ekrana. Kvadratić će onemogućiti elektronskom mlazu da pogodi centralni deo ekrana, koji će tako ostati zatamnjen dok će okolina biti osvetljena; korisnik će, dakle, videti tamni kvadrat na svetloj pozadini. Ako sada zamislimo da našem topu nekako naređujemo kada da emituje mlaz a kada da ga isključ, postaje nam jasno da je na ekranu moguće dobiti neku statičnu sliku komplikovaniju od običnog kvadrata. Menjajmo tu sliku dvadesetak puta u sekundi i eto crtanog filma.

Dodajmo sada malo boje našim razmišljanjima. Pretpostavimo da je ekran premazan trima vrstama fosfora od kojih jedna, kada je pogodimo elektronskim mlazom, pocrveni, druga pozeleni, a treća poplavli. Dodajmo, zatim našoj „artiljeriji“ još dva „topa“ koji će u punoj sinhronizaciji gadati tačke ekrana osvetljavajući





tako svaku od njih u crvenu, plavu ili zelenu boju ili neku njihovu kombinaciju. Kao što će vam rado objasniti svaki slikar, kombinovanjem crvene, zelene i plave boje mogu da se dobiju sve ostale boje (ne pitajte nas zašto se u osnovnoj školi uči da su osnovne boje crvena, plava i žuta!) a sa njima i kolor televizor.

Recimo, potpunošti radi, da postoje i specijalni monitori kod kojih elektronski mlaz ne skanira čitav ekran već, krećući se poput olovke, iscrtava samo osvetljene tačke čime se dobija izvanredna statična slika i vrlo slaba animacija (čak ni brzina od četrdesetak hiljada kilometara na čas nije dovoljno velika za „pokretne slike“ dobijene na ovaj način).

Kako, međutim, da saopštimo mlazu koju tačku treba da osvetli, a koju ne? Kretanje mlaza elektrona je, jasno, strogo periodično, pa u svakom trenutku vremena treba na neki način saopštavati koji od topova „puca“ a koji se „odmara“. Potrebna su nam, dakle, tri ulaza koja odgovaraju crvenoj, zelenoj i plavoj boji a samim tim i tri žice koje će voditi od računara do monitora. Ovakav ulaz u monitor se, zbog engleskih imena za tri pobrojane boje, naziva RGB ulaz i pruža potencijale za najbolju moguću sliku. Osim R, G i B signala potrebno je obezbediti sinhronizaciju (sync impuls na shemama) kojom se nećemo opširnije baviti — za prosečnog korisnika računara je sasvim dovoljno da zapamti da RGB monitori omogućavaju izvanredno čistu sliku visoke rezolucije u koloru.

## Neki čudni ulazi

TV aparati su, na žalost, i danas najčešće lišeni RGB ulaza. Razlog nije u tome što je ovako nešto tehnički nelagodljivo (čitaoci ovih redova koji su verziraniji u elektroniku i TV tehniku znaju da je u svakom televizoru moguće pronaći R, G i B ulaze, mada ovakvo traženje može da bude otežano ili u praksi onemogućeno kod aparata koji koriste previše integrisanih kola), već u tome što do skoro nije bilo načina da se RGB ulaz upotrebi — televizor je, jednostavno, igrao ulogu televizorala!

Iako nemaju RGB, mnogi televizori imaju takozvani kompozitni video ulaz. Umesto tri žice, do monitora dolazi samo jedna (ne računajući masu) kroz koju stižu informacije o bojama pojedinih tačaka, kao i potrebnih sinhronizacionih impulsi. Kako se jednom žicom mogu zameriti četiri? Grubo porednje bi bila situacija u kojoj pošta montira dvojnika na vaš telefon. Na taj način dva korisnika mogu da se služe istom linijom, ali se pojavljuju i ozbiljno ograničenje — ne možete da koristite telefon dok vaš dvojnjak razgovara. Kada biste se, međutim, sinhronizovali sa dvojnikom tako da svako od vas govori u tačno određene sate i kada se ne bi pojavljivale situacije u kojima morate da koristite telefon kada nije na vas red, ni ovaj nedostatak se ne bi primedivao! Slično tome, nekoliko signala mogu da se modulišu na jedan signal-nosilac i tako upravljaju elektronskim topovima.

Jasno je da se kompozitni video signal mora „raspakovati“ kako bi CRT dobila informacije neophodne za normalan rad. Transformacija kompozitnog video signala u R, G i B signale obavljaju kola ugrađena u monitor, pri čemu ta transformacija ne može da proizvede signale koji bi bili

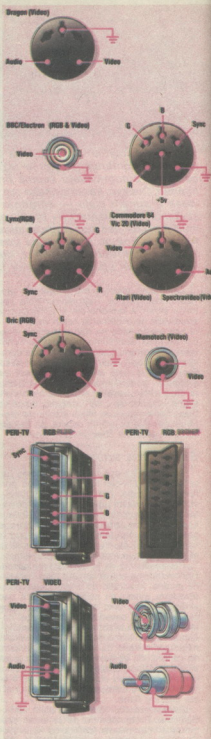
potpuno identični sa R, G i B signalima koji su „spakovani“ u kompozitni video. Pojavila su se, dakle, izobličenja koja ugrožavaju kvalitet slike.

Kompozitni video signal je, na žalost, krajnje nepogodan za radio difuziju — da se ljudi nisu dosetili nekih trikova, gledali bismo isključivo kablovsku TV, što se nikako ne bi isplatilo vlasnicima napravljenih aparata. Da bi se slika slala kroz etar, treba kompozitni video signal RF modulirati. Antena na krovu vaše zgrade prima RF signal, on se jednom žicom (druga je masa) dovodi do antenskog ulaza u vaš televizor gde najpre prolazi kroz tjuner i (de)modulator, a zatim biva dalje transformisan u R, G i B signale, koji jedini mogu da kontrolišu iscrtavanje slike

## Monitor i kako ga steći

Računar, jasno, nema nikakve potrebe da šalje sliku kroz atmosferu; on je sa monitorom vezan kablom. Da bi se obezbedila kompatibilnost sa starijim TV uređajima, u mnoge računare su ugrađeni RF modulatori koje ćete lako prepoznati — smešteni su u malu metalnu kutijicu negde uz samo zadnju stranu kućišta. Ovi modulatori transformišu kompozitni video signal u oblik koji bi antena primila iz etera. Čim tako dobijeni signal stigne u TV prijemnik, on biva pretvoren u kompozitni video — otprilike kao kada biste razgovarali sa komšijom tako što biste ono što želite da kažete prevodili na engleski, a zatim njemu prepuštali da tekst vrati na srpskohrvatski. Ma koliko dobri prevodioci radili ovakvo nešto, prevedeni tekst neće biti identičan polaznom. Slično tome, upotreba antenskog ulaza u televizor unosi izobličenje slike i onemogućava rad sa većim brojem znakova u redu i finom grafikom.

Bilo je teško ali se, nadamo se, isplatilo — sada, zar ne, znate kako radi televizor, pa je došlo vreme da izaberete onaj koji vam odgovara. Treba, pre svega, da se odlučite za kolor ili crno-beli monitor, što je, veruje nam na reč, prilično teška odluka. Reklo bi se da će onaj ko nema dovoljno novca kupiti crno-beli, monitor a srećnik koji se nalazi u manjoj nestašici dolara i maraka — monitor u boji! Ne mora, međutim, da bude baš tako — crno-beli (ili zeleno-beli) monitor obavezno daje kvalitetniju sliku, iako je daleko jeftiniji! Razlog leži u činjenici da je crno-belom monitoru potreban samo jedan delić fosfora za svaku tačku, dok kolor monitori zahtevaju tri, po jedan u svakoj boji. Na ekranu umereno visoke cene obično ima oko 360000 fosfornih tačaka, što znači da crno-beli monitor može da „opsluži“ rezoluciju 600x600, a kolor monitor sa istim brojem tačaka — svega 350x350! Za korisnike koji će se baviti obradom teksta, poslovnom obradom podataka i sličnim radnjama koje zahtevaju dugotrajno i koncentrisano gledanje u ekran, crno-beli (ili, kako se to obično kaže, monohrom) monitor predstavlja pravi izbor — nije ni malo slučajno što se na sve terminalne velikih računarskih sistema ugrađuju upravo ovakvi uređaji. Jeste li, međutim, nekada probali da igrate neku igru na velikom sistemu? Ako i niste, nemate za čim da žalite — igre bez boja kod kojih se po ekranu kreću slova i nisu nešto naročito. Kolor monitor bi, dakle, mogao da bude pogodan za igre, mada je za njih i običan televizor više nego dovoljan. Sve u svemu, za naš džep je crno-beli monitor upotpunjen kućnim televizorom u boji verovatno najprihvatljiviji izbor.



**Velikane veza:** Savremeni televizori u boji imaju znatno veće potencijale za priključivanje kućnog komputera nego što su njihovi vlasnici svesni — svi potrebni ulazi nalaze se na višepolnom priključku „peri“ (novi evropski standard za priključke na televizorima poznat pod imenom SCART); na slici su obeleženi samo oni pinovi koji su od interesa za jednog ljubitelja računara.





*Prozori u (računarski) svet: lako se na monitore najčešće gleda kao na svojevrsan luksuz, oni spadaju u najelementarniju opremu za lole ozbiljniju primenu — dobra i stabilna slika koja ne nadražuje vid poboljšava udobnost rada više nego bilo koja periferijska jedinica*

### **Dve muve jednim udarcem**

Ne treba biti mnogo praktičan pa pomisliti da nije teško spojiti korisno sa ugodnim — zašto da ne kupimo kolor televizor sa video ulazom? Takvi televizori se uveliko proizvode, s tim što video ulazi na njima obično nisu namenjeni komunikacijom sa računarom, već komunikacijom sa video rikorderom, što znači da ćemo imati manje ili više problema da ih povežemo sa našim računarom. U rešavanju ovih problema će od pomoći biti slika koju dajemo. Na njoj su prikazani video i RGB izlazi većine popularnih kompjutera na kojima su obeleženi pinovi. Ukoliko vaš televizor poseduje džek poput nekog od pobrojanih, biće dovoljno da ih povežete odgovarajućim brojem kablova i ispitajte čitavu skalereniju. Malo problem može da izazove činjenica da

većina ulaza sa video rikordera prenosi sliku i ton zajedno što znači da se, osim kompozitnog video signala i mase, vodi i žica koja „prenosi“ ton! Ta žica vam neće biti potrebna ukoliko vaš kompjuter ima sopstveni zvučnik, ali će „zlata vredeti“ ako se vaš računar ostanja na televizor da bi generisao zvučne efekte! Kada biste vaš „komodore 64“ ili VIC 20 povezali sa nekim ne-komodorovim monitorom, računar bi „zane-meo“, što bi moglo da se „izleči“ jedino istovremenim povezivanjem sa pojačalom. Ukoliko biste, međutim, iskoristili neki televizor sa monitorskim ulazom i potrudili se da sprovedete sve potrebne kablove, problema ovoga tipa ne bi bilo!

Moguće je da povežete računar sa televizorom, pritisnete taster AV (audio-video) na daljinskom upravljaču i vidite jedino sneg. U čemu je problem? Da bi detektovali priključeni video rikorder, neki televizori (npr. Grunding) zahtevaju da se na jedan pin video priključka dovede 12 V. Obzirom da taj napon treba jednostavno da učini neki tranzistor provodnim i tako isključi tjuner, potrošnja će praktično biti nula pa potrebnih 12 V možete bez grize savesti dovesti iz računara ili, ukoliko imate smisla za humor, iz samog televizora! Ako prevezivanje džekova zahteva svu hrabrost koju imate, možete da kupite ispravljač i sa njega napajate ovu kontaktanu tačku. Strogo pazite da vam se ne dogodi da pogrešite

polaritet napona, inače ćete osetiti neki čudan miris i — vaš TV putuje u servis! Na donjoj polovini slike vidite PERI konektore koji postoje na nekim televizorima i raspored priključaka na njima. Ukoliko se na vašem računaru nalazi DIN, a na televizoru PERI priključak, slobodno se upustite u operaciju poput one koju smo već opisali. Pinove priključaka koji vam nisu potrebni jednostavno ostavite nepovezane!

Iako je korišćenje televizora sa monitorskim ulazom privlačna ideja, uređaji ovog tipa nikada ne mogu da budu prava zamena za monitor. Razlog za to je takozvani protisni opseg — televizor ne može da izađe na kraj sa informacijama koje zahtevaju propusni opseg širi od 5,5 MHz, što je dovoljno za 40 slova u redu ali premalo za 80 — stroge tehničke specifikacije zahtevaju propusni opseg od 10 MHz! Monitori se prave tako da omogućavaju širok propusni opseg, što znači da će slika na njima, bez obzira koji ulaz televizor koristi, biti oštrija i prijatnija za oči. Zbog svega toga ćemo baciti jedan pogled i na tržište monitora.

### **Popularni modeli**

Treba, pre svega, da kažemo da su kod nas kolor televizori daleko skuplji nego u inostranstvu, pa može da vam se učini da kolor monitori predstavljaju prilično jeftine naprave. Opusov model JVC, za oko 180 funti, nudi sasvim prihvatljivu razoluaciju 370x470, dok se za tridesetak funti više dobija model sa rezoluacijom 580x470. Novex-1414-CL nudi rezoluaciju 640x256 (u početku je pripremljen sa BBC računarom u planu ali je donjeje prilagođen svim kompjuterima koji imaju RGB izlaze) za svega 200 funti. Nešto skuplji modeli kao što je Kaga (225 funti za model Vision 2 310 za Vision 3) nude veliki broj boja i neke druge privlačne karakteristike, dok je Microvitec 1431 rekordar sa cenom od samo 155 funti.

Monohrom (ili zeleno-beli) monitori, zajedno sa zaštitnim staklom protiv zrčenja, koštaju daleko manje — Digivision PM je jedan od kvalitetnijih sa cenom od 112 funti kod Chiltern Electronics prodaje uređaj koji, bar po reklamama, nudi veoma visoku rezoluaciju za samo 90 funti. Većina crno-belih monitora košta između 90 i 120 funti, dok je rekordar u niskoj cenii Zenith koga možete da dobijete za samo 66 LSTG. Svi podaci o popularnim modelima monitora dati su u tabeli.

Svi naši napisi o periferijskim uređajima se završavaju poglavljem u kome piše „... Sinclair Spectrum nema... ali...“ Neka tako bude i ovog puta — ZX81 i Spectrum nemaju video ni RGB izlaze, što znači da moraju da budu manje ili više modifikovani da bi se priključili na monitor. ZX81, jasno, nije računar koji bi trebalo povezati sa monitorom; to nije ni „spectrum“, ali će se možda neki od njegovih vlasnika odlučiti za ovu operaciju pogotovo ako im se pruži mogućnost da koriste isti uređaj kada nabave neki bolji kompjuter. Takve potencijalne pruža firma Microvitec koja za 225 funti prodaje paket pod nazivom 1431/MZRGB/TTL. U ovom paketu se, osim monitora, nalazi ekspanzijska kartica koja pretvara TTL izlaze na Spectrum u RGB signal potreban monitoru. Ista kompanija je model 1431/MO3 sa rezoluacijom 653x585 i propusnim opsegom od 18 MHz namenila vlasnicima QL-a.

*Dejan Ristanović*

# hakerski zakonik

## Programeri na ispitu

### Korektnost

Program je korektan ako radi ono za šta je napisan. Na žalost, postoji mnogo nijansi značenja izraza „korektan program“. Najjednostavnija je: (1) „Sintaksne greške nisu otkrivene“. Ovo je najslabija varijanta korektnosti: većina jezika upotrebe sintaksno ne analizira delove programa koji se nisu izvršavali. Nešto je bolji nivo (2): „Nema grešaka ni u tekstu programa ni u toku izvršavanja programa“. Program korektnosti nivoa (2) sigurno je korak napred u odnosu na nivo (1), no šta ako je neka formula pogrešno prepisana, a još je uvek sintaksno tačna? Zato uvodimo nivo (3): „Postoji neki skup ulaznih podataka za koje program daje tačne rezultate“. Ovo već nije loše, ali sad apetiti rastu i prirodno je zahtevali još veći kvalitet programa: (4) „Za tipičan (razuman ili slučajni) skup ulaznih podataka program daje korektan rezultat“. Većina početnika ostaje na ovom nivou shvatanja korektnosti, a neki dodaju još i nivo (5): „Za namerno zadate teške grupe ulaznih podataka program daje korektan rezultat“. Korektnost nivoa (5) utvrđuje se empirijski, testiranjem i neposrednom proverom programa na računaru. Sledeći nivo korektnosti je (6): „Za sve moguće skupove ulaznih podataka (moguće u odnosu na zahteve koje program mora da ispunji), program daje tačan odgovor“. Razlika između nivoa (5) i (6) suštinske je prirode: nivo (5) je najviši kojeg možemo proveriti metodom probe i pogreške. Nivo (6) je kvalitativni skok — tu programe treba dokazivati, tj. ubediti se nekim apstraktnim razmišljanjem da program radi dobro.

Evo banalnog primera. Zamislimo prost program koji čita jedan broj sa tastature. Prosečan kućni računar može da predstavi nekoliko milijardi različitih brojeva, pa je jasno da nikad ne bismo mogli proveriti da li tako jednostavan program radi za sve moguće ulazne vrednosti! Sledeći nivo korektnosti još je oštriji: (7): „Za sve moguće skupove valjanosti test-podataka, kao i za sve moguće kombinacije pogrešnih ulaznih podataka — program daje tačan (ili razuman) rezultat“. Na kraju, nivo (8) — „Za sve moguće podatke program daje korektno ili razumne odgovore“ — ostaje korisnikov ideal, ali ga je praktično nemoguće postići.

Pretpostavimo da imam dva „podjednako“ korektna programa, tj. na istom nivou korektnosti. Kako proceniti koji je program „bolji“? Kroz praksu se kristalizovalo nekoliko osnovnih kriterijuma.

### Čitljivost

Ovaj kriterijum je očigledan i ne bi ga ni trebalo navoditi kad bejzick ne bi prosto

### 20) programeri na ispitu

mamio programera da piše nepregledne programe. Programi u bejzicku se izvršavaju interaktivno, pa je prirodno da programer sedne direktno za tastaturu i istovremeno i smišlja i ukucava i proverava program. Greške su neminovne, jer mozak može svesno da obraduje samo do 7 objekata u isto vreme. Ovakve seanse pretvaraju se u mučenje editora umesto u kreativan rad, i sve ukupno, to je efikasan način da se izgube sati i sati u „programiranju“.

Ako osoba koja nije pisala program ne može da ga pročita, još manje može da ga shvati, a pogotovu ne da se izmeni. Dok stvara program, programer drži u glavi ogroman broj nepovezanih činjenica, ali je toliko skoncentrisan da ne oseća potrebu da bilo šta dokumentuje. Otrežnjen je dolazi kada kroz nekoliko dana (ili meseci) pokuša da promeni program. Sa iznenađenjem konstatuje da mu imena varijabli od svega dva slova ništa ne govore, da je zaboravio treba li koristiti trikovne umesto da piše jednostavno itd. Programeri se pišu „levom rukom“, jer programer smatra da je to njegova privatna stvar, da će program koristiti „sad i više nikad“ i slično. Sudbina takvih programa obično je znatno drugojačija. Teško je baciti gotov program samo zato što je „iznutra“ loše napisan: softver koji radi dragocen je! Još jedan loš program ulazi u svakodnevnu upotrebu — sa mogućim katastrofalnim posledicama...

### Prirodnost

Program je napisan prirodno ako naredbe programa slede logiku objekta programiranja, tj. procesi u programu odvijaju se na isti način kao i u problemu kojeg program rešava. Notorna je činjenica da je mnogim programerima takav pristup ispod časti! Takvo programiranje nije im dovoljan mentalni izazov, i trude se da donadu trikovne, prečiće, nestandardne upotrebe mogućnosti jezika za standardne potrebe itd. Pošteno govoreći, ima ljudi za koje je programiranje umetnost radi umetnosti, kreativan čin, umetničko delo, intelektualna vežba i slično. Oni teže da napravie SEBI sličan program, a ne program sličan problemu koji se rešava. Možda takvo programiranje može da postane novi oblik psihoterapije, ali u profesionalnom programiranju prihvataju se isključivo standardna rešenja.

Osoba koja čita tui program mora da savlada i problem koji se rešava i stil kojim je programer pisao. Ako je stil standardan (u bilo kom smislu), izmene u programu biće znatno olakšane.

### Pouzdanost

Pouzdanost programa je neka razumna mera odsustva grešaka u programu. Idealno bi bilo da su svi programi 100% pouzdani, ali je to u praksi nemoguće proveriti. Stoga se potpomažemo testiranjem, koje programeri-amateri obično shvataju kao

nužno zlo, a svaku pronađenu grešku kao ličnu uvredu. Profesionalci zauzimaju dijametralno suprotan stav: testiranje je proces izvršavanja programa sa ciljem da se pronađu greške! Budimo načisto sa jednim: programiranje je jedna od najtežih misaonih aktivnosti, pa je normalno da se javljaju greške. Nije, dakle, sramota greške praviti — sramota je ostaviti ih u programu.

### Prenosivost

Program je prenosiv ako se može lako i efikasno koristiti na računaru za koji nije prvobitno napisan. Prenosivost se postiže korišćenjem standardnih jezika (npr. Micro-soft bejzicku), standardnih operativnih sistema (CP/M, UNIX, MS-DOS), kao i korišćenjem standardnih osobina jezika na kojem programamo. U pogledu prenosivosti, brda raznih PEEK- i POKE-naredbi deluju sasvim kontraproduktivno. Ako ništa drugo, treba ih prokomentarisati. Neusaglašenost računara i slaba prenosivost programa jedan je od najvećih problema računarske industrije danas!

### Opštost

Dilema je: koristiti algoritam koji radi praprije, ali obuhvata više potencijalnih slučajeva, ili isprogramirati baš ono što se traži, pa makar kako? Ako to ne zahteva previše vremena i napora, treba se odlučivati za nešto opštija rešenja. To je jedan od načina da nivo korektnosti programa pređe „magičnu“ barijeru šestog nivoa.

### Dužina teksta programa

Jedan od kriterijuma kvaliteta programa može biti dužina teksta programa, sa idejom da je manji program automatski bolji. Ovo je opravdano ako računar ima malo memorije, npr. 1K ili 16K, no, to više i nije tako akutan problem. S druge strane, mali broj naredbi skoro uvek znači sporije izvršavanje programa.

### Fleksibilnost

Program je fleksibilan ako omogućava korisniku da kombinuje nekoliko od više datih mogućnosti. Još je važnije da program ne sprečava svog korisnika u tome. Zato se većina komercijalnih programa piše u vidu menija.

### Zaštita od pogrešnih ulaznih podataka

Ovo je izuzetno važno za svaki program koji pretenduje na neki kvalitet. Nije reč o zaštiti programa — to spada u razmatranja iz oblasti operativnih sistema — već on mora biti robustan kao buldog i spreman da se brani od svih podataka koji ne odgovaraju po specifikaciji problema (npr. ne sme se učitati negativna dimenzija niza ili matrice).

### Osmišljenost

Program je osmišljen ako njegova dokumentacija ne sadrži suvišne informacije.

**Svi smo čuli za greške koje potiču od računara: zakasnele penzije zbog nepravilno promenjenog programa, zahtev pretplatniku da uplati račun od 0.00 dinara, eksploziviranje rakete Saturn jer je u programu umesto zarezta stajala tačka... Krivci su, očito, programeri koji pišu programe. Poslednjih dvadesetak godina sve više pažnje se obraća na kvalitet programa, odnosno na kriterijume koje program treba da ispunjava da bi bio „dobar“. Ne prihvatamo arhitekta čije se zgrade RETKO ruše, i teramo u zatvor knjigovode čiji se završni računi MANJE-VIŠE slažu: ne bi trebalo prihvatati ni programere čiji programi UGLAVNOM rade!**

Npr. program nije osmišljen ako se na više mesta nalazi jednan te isti niz naredbi, ili ako je zaostao potprogram kojeg niko ne poziva. Programi u jeziku po pravilu nisu dovoljno osmišljeni, jer se pišu za računarem, a isuviše je lako rešavati nove probleme promenom broja postojećih programskih linija.

### Udobnost

Program je udoban za eksploataciju ako se može dopisati ili promeniti kada se pojavi potreba. Ovaj kvalitet programa je povezan sa shvatljivošću i čitljivošću programa, koji su mu predušolvi. Udobnost eksploatacije najviše se ceni među ljudima koji žive od programiranja.

### Korisnost

Program je koristan ako je zgodan za praktičnu primenu. Pojam korisnosti ima dva značenja — spoljašnje i unutrašnje. U prvom smislu, program je koristan ako se može primeniti (potpuno ili delimično) i u svrhe u koje nije napisan. Npr. da li dati program može da saraduje i sa nekim drugim programom, ili u nekom drugom projektu. Od dva procesora programa za obradu teksta, korisniji je onaj čije datoteke može da čita i bejzik (jer to omogućuje sortiranje, aritmetičke operacije itd.) U drugom smislu, program je koristan ako je prijateljski nastojen prema svom korisniku. Npr. kod unosa podataka, zgodno je da podaci budu u tzv. slobodnom formatu, (praznine se ne broje). Program uvek treba da sugeriše korisniku svoje mogućnosti, kao i da javlja greške, i to jasnim porukama. Bolje je prijaviti grešku kao „Division by zero in line XXXX“ nego porukom „E—2“. Da bi program bio korisniji, uvode se razna dodatna sredstva za komunikaciju čoveka i računara: svetlosna olovka, miš, palice za igru, boje na ekranu, grafičke mogućnosti itd. Izlazni rezultati moraju uvek biti dokumentovani, npr. program za obradu teksta treba da ispiše broj stranice i ime autora na svakoj stranici, itd.

### Struktuiranost

Program je struktuiran ako su njegovi delovi povezani u celinu na neki način. Struktuiranost podrazumeva da se autor programa tokom pisanja pridržava nekih standarda u pisanju. Npr. program koji je napisan po pravilima tzv. struktuiranog programiranja biće i struktuiran program. Programerova samodisciplina u pridržavanju usvojenog standarda je odlučujući faktor u stvaranju razumljivog programa. Ovo je moguće bez obzira na konkretan jezik programiranja, pa čak i na bejziku.

### Efikasnost

Program je napisan efikasno ako rešava postavljeni problem bez kršenja suvišnih resursa računara. Efikasnost je obično dijametralno suprotna prenosivosti i čitljivosti programa. Pojam resursa je fleksibilan i obuhvata centralnu i spoljašnju memoriju, broj izvršenih naredbi programa, brzinu izvršavanja itd. Tokom 60-tih godina glavni problem u praktičnom programiranju bio je odnos zauzeća memorije i brzine izvršavanja programa. Raniji računari bili su skupi i malog kapaciteta, pa su programeri veći deo vremena provodili praveći kompromis vreme ili brzina. Sa padom cena memorije ovo pitanje sve više gubi na važnosti. Brzina izvršavanja programa je kvalitet gde brže ne znači uvek i bolje! Ipak, većina programera spremna je da napusti sve dobre principe, poslovnice, pravila i preporuke samo da bi se program brzo izvršavao. Brzo izvršavanje programa bolje nego ista hrani programersku sujetu.

### Nezavisnost od računara

U oblasti mikoračunara ovo je samo san. Bejzik se javlja u toliko verzija da je napisana i knjiga kako prevoditi sa bejzika jednog računara na bejzik nekog drugog računara! Ako je ipak nezavisnost cilj (da bi se isti program plasirao na više računara i time profit povećao), onda treba sve ulazno/izlazne funkcije programa redukovati na tzv. primitivne funkcije, recimo na potprograme kojima se obraćaju svi ostali delovi programa. Najpoznatiji primer programa koji uspešno radi na više sasvim različitih računara je Hisoft Pascal, koji radi na „spektrumu“, „šarpu“, „naskomu“, CP/M-u, „nju brejnu“ itd.

### Dostupnost

Program ima svojstvo dostupnosti ako omogućava selektivan pristup pojedinim komponentama. U bejziku su bukvavno sve varijable dostupne svakom delu programa i to u svakom trenutku izvršenja (što je ujedno i najveća slabost bejzika).

### Komunikativnost

Program je komunikativan ako pruža mogućnosti da se ulazni podaci lako opisuju, odnosno ako su poruke koje šalje korisniku jednostavne za shvatanje i sadržajne. Većina komercijalnih programa sadrži tzv. Help-naredbu, kojom se korisnik u nedoumici obavestava koje programske opcije postoje. Neki programi za obradu teksta (Wordstar) sve vreme na ekranu prikazuju sve raspoložive naredbe. Dizajn protokola mora biti pažljivo odmeren. Korisniku treba omogućiti da unosi naredbe na dva načina: punom dužinom i skraćeno, tako da kad nauči da rukuje sa programom može de otkuca npr. F umesto FINISH. Izlazne poruke treba da budu informativne, ali ne i

predugačke: ništa ne nervira toliko kao sedenje pred ekranom i gledanje u dugačke, predvidljive poruke — koje se ne mogu ukinit!

### Otvorenost

Program je otvoren ako se njegove funkcije i smisao pojedinih naredbi lako shvataju čitanjem teksta programa. Imena varijabli bi trebalo da budu proizvoljne dužine, pa je bolje imati bejzik koji to omogućuje. Tu dolaze do izražaja komentari u programima, koje ne treba pisati posle, nego odmah u toku ukucavanja programa.

### Proširljivost

Program se može proširiti ako dozvoljava povećanje zauzeća memorije za podatke, odnosno ako se mogu proširivati postojeće računске mogućnosti u programu.

### Ljudski faktor

Program uzima u obzir ljudski faktor ako radi ne zahtevajući suvišan utrošak vremena od strane korisnika. Negativan primer je program koji zahteva da konvertujete stepene u radijane usred nekog astronomske proračuna. Osim takvih problema, programi treba da se odnose ljubazno prema svojim korisnicima.

### Izmenljivost

Program je izmenljiv ako njegova struktura dozvoljava lako unošenje izmena. Ne može se lako promeniti program u jeziku u kojem se nalazi desetak naredbi na istom broju linije. Ako je program napisan prirodno, tj. ako sledi strukturu problema kojeg program rešava, onda je to bitan preduslov za laku izmenljivost programa. Ljudi se menjaju, zahtevi se menjaju, programi se menjaju. Pardon — programi menjaju ljudi. Da promenimo program, potrebna su tri koraka: (1) razumeti program, i odrediti mesto izmene (2) efektivno izmeniti program, i (3) proveriti se u korektnost izmenjenog programa. Većina profesionalnih programera NE pravi nove, nego MENJA stare programe (što se stručno zove „održavanje programa“).

### Završenost

Program je završen ako sadrži sve delove neophodne za funkcionisanje. Npr. program nije završen ako jedna od opcija koja se nudi u meniju nije uopšte isprogramirana, ili ako nema obezbeđenja od unosa nepravilnih podataka i slično.

### Shvatljivost

Većina gore spomenutih kriterijuma ima za cilj da omogući lako shvatanje programa, te je ovaj kvalitet programa najvažniji. Ne zaboravimo da tekst programa služi prvenstveno za komunikaciju među ljudima, a ne među mašinama!

Duško Savić



Jedna  
tipična  
karijera

# razvojni put bore pirata

Smešna  
strana  
računara



U novinama su se za samo dvadesetak godina pojavila čak tri članka o piratima. Prvo je u „Svetu kompjutera“ Dragan Antić napao pirate zato što štete proizvođače i distributere... Odmah zatim, u intervjuu „Računarima“ profesor Jozo Dujmović izjavljuje da... (pirati) ugrožavaju ne samo pošten rad i privredivanje, već često i bezbednost društva“. U tom istom broju „Računara“ objavljen je i članak Jelene Rupnik, poznatije kao Piratska Majka, u kome ona brani jadne male pirate, koji su žrtve prave hajke samo zato što se lepo zabavljaju i zarađuju nešto malo za džepar. Robin Hud je još živ. Taj članak je po ceoj svojoj dužini isflekan suzama i po gde kojim gnevnom usklikom na račun velikih — najvećih pirata i inicijatora lova na pirate.

To je sasvim dovoljno da se vidi klima — vatreni napadi i materinska zaštita. Ukratko, pirati su džeparoši i novi tip filatelista.

## Uzmi pare i beži

Krada knjiga se kod nas ne smatra kradom, i to uglavnom zahvaljujući stavu autora koji su oduševljeni kad njihove knjige neko krađe. Pirati kradu ako je presnimavanje tuđih programa krađa. Presnimavanje tuđih programa je krađa, jer tako misle oni koji programe pišu. Prilično jednostavno, zar ne?

Pirati kradu. Ako već kradu bez ikakve grize savesti, zašto ne bi lepo i priznali da kradu. Možda bi se tako malo razvejali privid duhovnog bratstva koji se oko njih širi i malo nazreo njihov moto: „Uzmi pare i beži“. Da li će to ikada priznati dobre vile piratstva?

Ako su pirati kao kategorija nešto loše onda su najgori od lošeg, pravi negativci iz američkih ili Bulajičevih filmova, upravo mega-pirati! Ili dinosauri marketinga.

To su ljudi koji su se od piratstva već odavno obogatili. I to na sve moguće načine, osim duhovno. Jedan od najsajnijih primera tog king-size piratstva je zaštitni znak računarskog marketinga. Njegova titula je Veliki Majstor Prodaje, ali će u ovom tekstu biti poznat kao A. A. Priča o njemu

počinje još davno, dok su računari bili mladi, a naziv pirata se vezivao za Ridobrod i trgovce pornicima. A. A. je još tada shvatio kakav će rusvaj na tržištu zabave napraviti računari i video igre. Pošto je i sam bio zaražen napadačima iz svemira i sličnim stimulansima, imao je dve video igre, koje je prodao najboljem prijatelju da bi za tu lovu kupio treću. Pritisnut rastućim brojem prijatelja, poznanika i sasvim nepoznatih, koji su tražili da im on presnimi neke igre, a bili su sasvim voljni da plate za to, A. A. je počeo intenzivno da radi na presnimavanju i umnožavanju najpopularnijih igara, i tako je proširio svoje aktivnosti i koje su do tada uglavnom bile koncentrisane na prodaju šapirografisanih i fotokopiranih skripti za učenje računarskih jezika. I propagandom superpamčenja. Da, tajna Njegovog velikog uspeha je u Njegovom superpamčenju.

## Veliki majstor prodaje

A. A. je shvatio koliko će brzo računari (bolje rečeno — video igre) zaraziti široko pučanstvo. Zato je i poduzeo neophodne korake. U redakcije Televizije, Radija i svih ostalih medija je prodro putem svoje jake i

**Počelo je kao igra i kao dobar štos koji je sasvim lepo pothranjivao naš nacionalni ponos — naši računarski pioniri kao od šale doskaču najlukavijim engleskim programerima otvarajući njihove i najtvrdre Spretni da (ne uvek baš ni) za sitan džeparac podele rezultate svojih podviga sa manje veštih ljubiteljima računara, brzo su postali „demokratizatori računarskog znanja“; „lučonoše računarske pismenosti“; junaci godine, mas-medija i, malte ne, socijalističkog rada. Nisu, međutim, sve igre podjednako bezazlene. Uostalom, da li je sve to još uvek samo igra?**

neumorne ličnosti. Gomile programa koje je besplatno podelio urednicima tih redakcija su samo dokaz koliko su prijateljstva i poluprijateljstva koja su tako formirana čvrsta i iskrena. Skoro nijedna od gore navedenih redakcija nije propustila priliku a da malo svog dragocenog vremena ne posveti hit temi — računarima. Malo je takvih posveta u kojima se nije pojavio A. A. veoma glasno delujući svojom adresom i telefonom, verovatno zbog želje da što većem broju početnika u tim oblastima pomogne koliko može. Njegova kuća je bila uvek otvorena za momke željne novih programa.

Paralelno sa tim humanim aktivnostima, čitava organizacija malih pirata, sledbenika Velikog A. A., radi na reklamiranju i restauriranju njegovih kaseti sa piratizovanim programima. Skup genijalnih poteza jednog stručnjaka za marketing. Naravno, on je i pre toga bio legenda, jer je svojevremeno u jednom radio-klubu držao besplatna predavanja za velike prave prodavao skripta iz te oblasti. Makijaveli bi umro od zavisti, da nije već mrtav.

Pošto se ispojavljivao na svim mogućim radiotelevizijskim i ostalim programima sa svojim reklamama video igre i sebe, postao trajno vezan za pojmom računara. Živa legenda. Svoje čuvene instant kurseve bejzika je držao svuda. Možda čak i na seoskim svadbama. Tamo gde nije stizao lično, stizale su kasete sa njegovim govorima. Osim što je uspeo da se sasvim pristojno obogati baveći se svim poznatim i nepoznatim biznisima vezanim za računare, A. A. se smatra i računarskim prosvetiteljem naše sredine. Nešto kao elektronski Čirilo i Metodija. Kako stvari kod nas stoje, to bi možda čak i bilo moguće, da A. A. nije živi dokaz tvrdnje da onaj ko se na računarima obogatio ne mora obavezno da ima stvarne veze sa „tajanstvenim svetom programiranja“.

Priča o Čirilu i Metodiju se završava tako što je nakupio dosta para, izdao na jednoj kaseti svoja mucanja sa radija, napisao nekoliko knjiga za sve računare, svajči ukus i svajči džep, počeo da piše memoare i sabrane fotokopije iz stranih časopisa, ubacio se u velikom stilu u evropsku računarsku mrežu i ostavio u njoj gomilu malih oglaš, uzbuđeno je, oženio se i živio srećno do kraja života. Pale samo patuljci i vile. Ali, zato Majfije ima puno.

A. A. nije unikatan primer za najvišu klasu pirata, ima ih još prava gomila, ali pošto se manje eksponiraju i imaju manje talenta (i ega) za izgradnju javnog imidža, ostate zapamćeni samo po sitnim kakama

za prodaju piratizovanih programa, po gangsterskim mentalitetima i po uzrečicama tipa „100% mašinar“ i sličnim. Pitamo se, pitamo, koji je naš čuveni prosvetitelj osnovao prvu piratsku firmu i kupio jedan od prvih „spektruma“ u Beogradu prodajnog programa za ZX81. Pro... pro...

A šta je sa malim piratima? Mali pirati ne postoje. Mali pirati su mali samo kad započnu. Nema ništa malo u načinu na koji oni zaraduju pare. Kada presnimite svoje prve programe i prodate ih drugu iz klupe ili svojoj najnovijoj majci, vi ste mali pirat. Onda biznis kreće. Umesto malog pirata — velika poslovna akujka. Mali pirati spadaju u istu grupu fikcija kao i pravediti osvettnici i naivne lepoticke, znači, skoro da i ne postoje. Nešto kao Petar Pan.

### **Psi laju a karavan prolazi**

Jedna od stvari koja daje najviše boje piratizmu je legenda o piratskoj kreativnosti i spretnosti. Ta legenda potiče iz dugih sati traganja za cakom koja će pomoći da se razbije zaštića. Ako i to spada pod kreativnost, možda bi trebalo naći neku novu reč za kreativnost.

Šta pirati rade što je tako loše (osim što kradu)? To je teško iskopati iz govorinica antipirata. Pominje se neko siromašenje. Kao, distributeri i proizvođači siromaše zato što se njihovi programi piratiraju. To će ih i uništiti, pa će na šteti biti svi, pa i pirati. Jako zgodno. Šta oni nama MMFondom, mi njima piratima. Na žalost, nije tako. Ti milioni dolara koje su vredni pirati uskratili kojekakvim softver firmama ne postoje. Ne postoje zato što u našoj zemlji zabava živi dok je jeftina. Video igre su jeftine samo preko pirata. Ko bi plaćao za jednu igru 10 funti?

Drugi pirati, počnite da delite programe besplatno i ja ću postati prvi član Društva za Zaštitu Pirata i Ostalih Glodara, koje je osnovala Jelena Rupnik.

Zašto pirate boli uvo za sve što im se kaže? Zato što je sve to samo priča. Priča piratima da je to što oni rade, ... kao da džeparac nekog ... nanošenje neprocenjenih šteta itd. ... to je vandalizam sličan čitanju i zloupotrebi tuđih pisama ... je genijalan moralistički potez. Samo je malo zastareo. Jedno 100 godina. Fall još samo tvrdnja da piratstvo izaziva slepio i sušenje desnice ruke. Danas čak i poslovlode prodavnica iz principa ne krade ispod milijarde, ali će pirati sigurno prestatu sa svojim grešnim aktivnostima ako ih lepo zamolimo i, u najgorem slučaju, izgrdimo. Pirati nisu skupljači maraka ili fotografi amateri. Pirati su najsavršenija varijanta prodavca sa Zelenog Venca i kao takvi imaju razvijen čitav jedan moralni sistem. Citiraju taj sistem u potpunosti: „Meni je ovako dobro. Šta mi možeš?“ Malo razočarenje. Umesto gomile elektronskih Tesli u povuju samo uzvik: „Kupite farmerke, kafa,

žileti, dobri programi za 'spektrum'." Takav je život, rekli bi Francuzi.

### **Ne diraj u programere**

Kako pritegnuti pirate? Ovo se čak ni meni ne svida. Kada bi svaki pirat znao da će, ako da oglas u novine, biti nazvan od strane SUP-a i prijateljski upitan da nije možda svoje programe nezakonito presnimio (pod uslovom da postoji takav zakon) za mesec dana bi bilo 10% od sadašnjeg broja pirata. Kao što rekoh, ovo se čak ni meni ne svida, da SUP, pored toliko aktualnih stvari, gnjavi baš klince pirata. Možda bi trebalo čitavu stvar prebaciti tržišnoj inspekciji. Iako i ona varijanta sa SUP-om nije loša kad se setim Čirila i Metodija i još par drugih.

Drugi pirati, preorijentišite se. Ako baš želite da zaradujete na računarima, naučite nešto programiranja i pokušajte da zaradite radeći zaista na računarima. Znam da će biti teško odvići se od toga da se skoro ništa ne radi a da se zaraduje, i navidi se na to da se na „kućnom ljubimcu“, kako to lepo kaže jedna spisateljica, stvarno nešto radi. U poslednje vreme se u pametnim časopisima dosta piše o tome da je naša (jugoslovenska) šansa ne u razvoju hardvera nego u razvoju i proizvodnji softvera. Otkrivi vam veliku tajnu — pod razvojem softvera se ne misli na razbijanje softverske zaštiće nego na originalni, ili bar prilično originalne, programe. U poverenju, za računarski deo špice „Hita Meseca“ je plaćeno tri stara miliona. Tri računarske cigle.

Pirati su rodoljubi. Izuzetno vole da piratiraju domaće programe. Skoro svi programi koji su na „Galaksijinom“ konkursu za najboljeg Yu programera bili nagrađeni piratizovani su skoro trenutno i sad su toliko rašireni da se skoro ne isplati distribuirati ih. Isto to se dogodilo zimnu na Zimskim Čorlijama mladiću koji je uspeo da svoju igru proda jednoj engleskoj firmi. Tokom jedine demonstracije programa nepoznati maloletnik je uspeo da pomoću duplog izlaza sa kasetofona presnimati čitavu igru. Pobesneli autor je jurio anonimnog pirata duž čitave hale, ali nije uspeo da ga stigne. Da je uspeo, pirat bi, pored programa, dobio još nešto. Sada tu igru ima pola Jugoslavije i čitav Beograd.

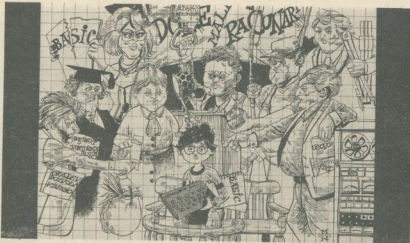
Naše tržište je ionako slablašno i siromašno. Još ako pirati počnu da potkradaju i frustriraju domaće autore, može da se dogodi da nikad i ne napravimo tržište softvera. Zato molim pirate da, dok ih ne zabrane, ne diraju domaće autore. Grehota je.

Pirati, čuvajte se. Ja sam gledao Osmog putnika. Svaki pirat u sebi nosi zametak jednog malog A. A. i kad-tad taj zametak će da naraste, pojede pirata i postane biznismen. Ostalo je legenda.

**Branko Đaković**

# obračun kod tjuringove mašine

## Računari iz mog ugla



Ilustracija: Misa Marković

Obični ljudi malo znaju o računarima, ali zato mogu da primete da računari znaju sve više. Dok je doskora vrhunac pretjne moderne tehnologije bio fizički nasrtać mašina na ljudsko telo, sada se taj sukob sve više prenosi u sferu imaginarnog. Veliki Brat će uz pomoć računara znati sve o vama: on će regulisati vaš život ili će vas ukinuti bez opomene ako se ponašate neregularno.

Po svemu sudeći, konfrontacija između naroda i računara nije u opštem interesu. Možda je mislija kućnih računara na Zemlji da celu stvar malo demistifikuju. Ali, čim su se pojavili, kućni računari stekli su otvorene i skrivene neprijatelje.

Proizvođači računarskih dinosaurusu bili bi najrešniji da mali računari nikad nisu ugledali svetlost dana. U dobra stara vremena oni su mogli da prodaju računare na kilo, i zajedno sa njima da uvaluju i opremu za održavanje specijalnih klimatskih uslova, neophodnih za opstanak njihove dece. A onda su došli mali, lepi, spretni i okretni računari koji prete da izjedu dinosaurusu i njihove roditelje.

Ni ovašnji proizvođači skupnih mišolovki ne mogu ravnodušno da posmatraju kako domaći miševi jedu sira na drugoj strani.

Oni koji se profesionalno bave računarima ne vole da im se po dvorištu umjavu tako neki hakeri. Samo prave gužvu i zagađuju im okolinu. Gubi se snaga misterije za ono što profesionalci rade. Oni strepe da se na kraju ne ispostavi da je njihovo čedo-dinosaurus u stvari maoumno.

Profesionalci bi da dokažu da veliki sistemi i mikroročunari nikako nisu isto. Izgleda im je zaista tako. Pojave se veliki stručnjaci za velike sisteme na TV da diskutuju o kućnim računarima. Čak i relativno nepučeni ljubitelji mogu da primete da ovi pojma nemaju o čemu govore. Propustili su priliku da se prethodno kod lokalnih hakera obaveste o čemu se tu radi.

Naučno plemstvo bi da se sve organizuje na strogo akademskom nivou, nikako samoinicijativno. To kvari nauku. Nema tu da se nekontrolisano vezamo: šta ako se neko drigne, pa umesto da se igra, počne da programira? Biće da su plemići ipak u pravu. Ako ništa drugo, titula je ono što daje snagu njihovim argumentima.

Jednog doktora nauka, koji postiže nirvanu teorizujući do besvesti, zaista može da iritira što se haker zeza sa računarom nego da dobri način. Šta mu drugo preostaje tako da sirotog hakera ubije u pojam, i to na svom terenu, gde je siguran u pobedu: zakazuje mu Veliki obračun kod Tjuringove mašine.

Zagovornici zvanične mudrosti oštri su protivnici malih računara koji pružaju mogućnost uspostavljanja kanala alternativne mudrosti za koju ne treba diploma. Nešto kao drugi program. Nezdugo je kad omladina može da bira; mogućnost izbora stvara anarhično situaciju, što je već samo po sebi opasno.

Neki koji ne vole male računare, a ne znaju kako to da objasne, okomili su se na bežik. Stvar je jasna: bežikomski mrze bežik zato što njime pričaju mali računari. Ah, ta preterana gadljivost izvesnih doktora nauka prema bežikju koji oštećuje mozak i ostavlja nepopravljive posledice na kičmenu moždinu. Ako već hoće da ogade deci računare, najbolje je da ih uvedu u škole, kao i seksualno vaspitanje.

Oni lukaviji još uvek borave u čeki i još nisu pripucali iz svog tajnog oružja. Oni su intimo protiv, ali neko njihovo šesto čulo govori im da trenutak nije povoljan da to ispolje. Oni se dodvoravaju, vode perfidnu politiku, sve dok im se ne ukaže prilika da u nekom mraku lupe po glavi kada se neće videti ko je to uradio. U međuvremenu, „sa puno simpatija gledaju na miade entuzijaste“.

Kovači sretnije budućnosti neguju podrživ stav prema malim računarima, koji mogu da poremete društvenu ravnotežu. Strah od promena loše utiče na njihovu lepotu i zdravlje. Oni pokazuju dobru volju, sve u nadi da će deca računare koristiti samo za igru. U njihovoj službi su raznorazni dežurni kočnari, ponosni na svoju funkciju. Odmerenost i umerenost su njihove glavne karakteristike. Oprezno su uvođenjem malih računara; blagi konzervativizam oduvek je bio na ceni. Od svih komandi na vozilu za 21. vek, oni još najviše vole da se drže za kočnicu.

Nostalgicari vremena prošlog, zagledani u daleko svetucanje svoje bivše mladosti, poznaju samo bisva zadovoljstva. Novim zadovoljstvima nisu dorasli. Zašto deca da se igraju sa računarom, kad im oni nude divotu prašine i krpenjaču u 21. veku? Zaboravili su samo da su pračke i krpenjače ukinule u socijalizmu, zajedno sa privatnom svojino.

Deo njihove svite su i „humanisti“, zabrinuti da se nove brzine pogubno ne odraze na čovekov metabolizam. Prastari naivni argumenti koji su tu samo da sakriju pravi razlog. Oni sa svojih humanističkih pozicija mogu da govore o otuđenju čoveka zbog gubitka kontakta sa drugim ljudima i prirodom. Računari su koncentrovana suština protivprirodno. Oni pokazuju gadeenje prema mašinama. Mašina je niži oblik od životinje; perversija je gora od sodomije. Mašina je vulgarna. Tehnologije je vulgarna, kažu humanisti, a ništa ne govore o otuđenju od budućnosti.

Protivnici malih računara su i oni koji su svoj razvoj kompletirali u 25-oj godini života. Oni su uhvatili određenu brzinu i sad funkcionišu po principu inercije. Goje se kod prepunog stola: veća masa — veća inercija. Čista fizika! Tehnologija zamajca, njima tako bliska, na žalost, nema nikakve veze sa računarima. Inerticari su zadovoljni svojim pozicijama (mala kućica, mala slobodica...) Znaju da nisu dorasli novim izazovima. Moraju po svaku cenu da štite ono što imaju (malu privilegiju). Ali, ma kako bili protiv računara, ipak će svoju malu dečicu poslati na mali kurs bežikja kod velikog pirata. Za svaki slučaj.

Među mrziteljke kućnih računara zalutale su i domaćice. One su samo deo folkora (ima toga još, ali baš da neću o tome da vam pričam). Njihovi muzevi beže u ekran. Rudar gore, rudar dole... I pritom zaboravljaju na svoje bračne dužnosti. Sirotim domaćicama nikad neće biti jasno šta je to što računari ima a one nemaju.

Mržnja prema računarima očigledno nema mnogo veze sa principijelnošću, nego sa tim kako sagledavate svoje šanse u tom kontekstu. Ako već niste u stanju da se sa računarima volite, ostaju vam dve mogućnosti: da ih mrzite, ili da ih ignorišete. Mnogo je bezbolnija varijanta da se trudite da ih uopšte ne primećujete: jednostavno zatvorite oči i nastavite da živite svoj dostojanstveni život.

Jelena Rupnik



# slikari bez kičice

Programi koje treba imati „Amstrad“/„Spektrum“

Mogu li računari da budu kreativni? Odgovor na ovo pitanje, kao uostalom i na pitanje o inteligenciji računara, za sada bi bio: mogu, ako se napravi program koji bi dovoljno dobro imitirao kreativnost (odnosno ponašanje). Reč „imitacija“, bez sumnje, zvuči prilično neuverljivo, ali treba imati u vidu da kopije ponekad umju da budu veoma bliske originalu. Program Brajana Džejmsa omogućuje popularnim računarima „spektrum“ i „amstrad“ da iskažu svoje skrivene umetničke kvalitete u slikanju pejzaža.

U video igrama, naročito u onima tipa avanture, sreću se sve uspešija grafička rešenja. Ima tu svakojakih predela, ukletih i prokletih šuma, napuštenih zankova, mračnih tunela, pećina, i naravno, tajanstvenih bića. Izbor je zaista veliki, mada se, doduše, u većini slučajeva od igrača traži prebujna mašta da bi neku od tih sličica doživio kao realistični ambijent.

Kreiranje slika na računaru, u principu, nije nikakav problem. Korišćenjem pogodnosti fine grafike (tu čak i naša „galaksija“ nije više izuzetak), mogu se ostvariti gotovo fotografski efekti. Jedini problem je kako konkretno ostvariti crtanje? Unositi tačku po tačku pomoću naredbe PLOT? To sigurno ne. Umesto toga, računari se može programirati da nacrtaju neku sliku.

## Sve iz svoje glave

Najjednostavnije je napraviti program koji će uvek crtati istu sliku, recimo jedan isti pejzaž, kad god se igrač u avanturi vrati na neko mesto. Eventualno, pojedini detalji se mogu pri tome menjati: na primer, obrušeno kamenje, ili oboreno deblo, kao prepreka na putu i slično.

Takođe, nije nikakav problem napraviti program koji će pri crtanju slike proizvoljno odabirati elemente i proizvoljno ih kombinovati: na primer, računari mogu izabrati jedan oblik planine od pet zadatih, ili jednu kuću od pet zadatih, itd., a onda sve to rasporediti na proizvoljan način. Međutim, teško da bismo to mogli nazvati kreativnošću.

Daleko veći izazov je sastaviti program koji će računaru pružiti maksimalnu slobodu u izboru — ne samo pri raspoređivanju standardnih elemenata na slici, već i pri samom kreiranju tih elemenata. Treba, dakle, navesti računaru da nešto naslika sam, onako, „iz glave“.

Naravno, u takav program se ipak moraju uneti neki osnovni podaci, neko početno „iskustvo“, na osnovu čega će računari stvarati svoja dela. Bez toga bismo dobili nekakve nadrealističke kreacije, sa lebdjećim građevinama, drvećem koje raste nadole, jezerima čija površina mirno stoji pod uglom prema horizontu i slično. Računari mora znati da građevine stoje na tlu, da drveće raste sa zemlje naviše, a da je površina vode uvek horizontalna. Mora zna-



ti i to da bliži objekti zaklanjaju dalje (a ne obrnuto), da udaljene planine imaju više plavičastu boju, dok one bliže mogu biti jačih boja. Objekti u blizini mirnih vodenih površina mogu se ogledati — to se naročito odnosi na obalske svetiljke, ako se radi o nekom „večernjem pejzažu“.

Sa svim ovim predrasudama, kao kakav umetnik koji je mnogo toga u životu iskustvo, računari mogu da pristupi stvaralačkom radu. Razume se, kreativnost će se imitirati, i to obilatim korišćenjem više matematike i teorije verovatnoće. Baš kao što je u stanju da simulira bacanje kocke, računari mogu, pomoću svojih generatora slučajnih brojeva, da simuliraju i sve druge slučajne radnje.

## Pejzaži po Furijeu

Na primer, treba izabrati oblik horizonta. Ukoliko je izbor pao na primorski pejzaž, deo horizonta može biti ravan, ali drugi deo može biti i brdovit. Oblik reljefa izabraće se metodama Furijeove sinteze: sabiranjem većeg broja sinusoidnih funkcija raznih amplituda i početnih faza, može se dobiti proizvoljno složena kriva linija. Dalje, na planinama može biti snega, što može biti stenovito ili pod bujnom šumom, nebo može biti sa ili bez oblaka, sa suncem na horizontu, ili na nekom drugom mestu... Detalj po detalj, računari će iscrtati čitav predeo, a da se praktično nikad ne ponavlja. Svaka slika će biti plod „trenutne inspiracije“.

Kvalitet slike, u mnogome, zavisi od karakteristika samog računara. Na „spektrumu“, recimo, veliki problem predstavlja ograničenost boja na čitave karaktere, tj. matrice 8x8. S druge strane, „amstrad“, koji ima mogućnost bojenja svake tačke ekrana posebno, daje neuporedivo bolji efekat. Zbog svoje „zraste“ strukture, slike dobijene na „amstradu“, korišćenjem programa Brajana Džejmsa, veoma podsećaju na vezele goblene, i to na one najuspešnije. Uostalom, tu ne treba mnogo govoriti. To treba videti.

Brajan Džejms je napravio dve verzije programa: za „spektrum 48K“ i „amstrad CPC-484“. Cena prvog je tri, a drugog pet funti. Program se može dobiti neposredno od autora: Brian James, 21 Lamond Place, Aberdeen AB2 3UT.

# pero od osam bita

„KOMODOR 64“

EASY SCRIPT

Lako, lakše,  
najlakše

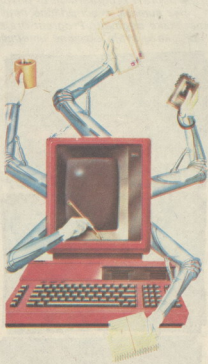
Objektivnost je, svakako, dobra osobina, ali kada je reč o mališanu „Komodor 64“ snabdevenim programom za obradu teksta „Easy Script“, nemoguće je ostati odmeren i ne priznati da taj program iznenađuje svojim kvalitetom. On nam, jednostavno rečeno, omogućuje da uradimo sa tekстом sve ono što nam je potrebno, pa i više od toga.

„Easy Script“ (tri skript) bi u prevodu značilo „lako pisanje“. Prvu verziju programa je napisao jedan čovek, Simon Tranmer, distribuira ga i štiti poznata softverska kuća Precision Software, a spada u jeftinije programe ove vrste. Cena mu je oko 60—70 funti na engleskom tržištu, a predstavlja verovatno najbolji program za obradu teksta namenjen piscima na „komodoru“.

## Osnovni parametri

„Easy Script“ radi kako sa disk jedinicom tako i sa kasetnim magnetofonom, pa za minimalnu hardversku konfiguraciju treba imati C-64, televizor, kasetni magnetofon i štampač. Naravno da je daleko pogodnije da spoljna memorija bude disketa. Predpostavimo, zato, da smo uključili račun, postavili disketu sa programom „Easy Script“ u malo da sačekamo dok se učita približno 23 mašinskog koda.

Na samom početku od nas se traži da izvršimo inicijalizaciju, odnosno da odredimo broj slova u redu na ekranu, da navedemo ili koristimo disketu ili kasetu kao spoljnu memoriju, kao i tip štampača koji smo priključili. Ako štampač većemo preko serijskog RS-232 interfejsa, onda unosimo i vrednosti za komandni i kontrolni registar koje određuju brzinu komunikacije, broj



korišćenih bitova u bajtu, i ostale parametre potrebne za serijsku vezu.

Broj simbola u redu na ekranu može biti od 40 do 240. Ako se opredelimo za, recimo, 80, to znači da će tekst koji unosimo posle osamdesetog otkucanog simbola u jednom redu, automatski biti nastavljen u sledećem. Mi, dakle, pišemo u tzv. slobodnom formatu ne pazeci na kraj reda, a račun vodi računa da se ne prekorači margina. Pri tome se reči lome na mestima na kojima ne bi smele, ali to ne treba da nas zabrinjava. Izgled teksta na ekranu ne određuje njegovu formu u kojoj će biti odtamparan na papiru. To se reguliše naredbama za formatiranje.

Šta se, međutim, dešava kada u kucanju dodemo do četrdesetog simbola koji već leži na desnoj ivici ekrana? Tada se ceo tekst pomera u levo tako da ekran „vidi“ i zonu desno od četrdesete kolone. Tako sve do desne margine. Vrlo je zgodno što se red pomera zajedno sa celim tekстом, tako da vidimo i njegovu „okolinu“.

Ali, nema razloga (osim ako ne unosimo široke tabele) da radimo sa 80 simbola u redu. Pri inicijalizaciji, zapravo, treba tražiti 40 slova u redu, i redovi teksta će nam

onda celom dužinom biti pred očima u svakom trenutku. Ako, kasnije, izvršimo formatizaciju na 80 simbola, moći ćemo da ureden tekst pregledamo i na ekranu na način koji je opisan. Tada na raspolaganju imamo tastere koji pomeraju tekst levo, desno i na dole.

Time je prilično elegantno prevaziđeno ograničenje „Komodor“ računara sa nedovoljnih 40 slova u redu.

Najviši red na ekranu rezervisan je za prikaz najvažnijih tekućih informacija, i naziva se statusni red. Tu se prikazuje režim rada u kome je trenutno „Easy Script“, položaj kursora, opcije koje su uključene i sl. Tu se takođe izdaju i poruke o eventualno učinjenim greškama. Ovaj red nestaje jedino u režimu komunikacije sa disketom. Preostala 24 reda slobodna su za tekst.

Najduži tekst može imati 764 reda sa po 40 slova, što predstavlja oko 17 standardnih kucanih strana.

## Operativne mogućnosti

Pri unošenju teksta kursorom se upravlja istim tasterima kao i u okviru ekranskog editora računara. Postoje još i dodatna pomeranja za reč, red, ili na određenu liniju u okviru teksta. Prikaz teksta olakšan je mogućnošću posmatranja ekran-po-ekran. Mogu se prema želji nezavisno postaviti boje iverice, pozadine i slova.

Pri ručnoj korekciji teksta možemo koristiti dva režima. Režim umetanja i režim prepisivanja novog sadržaja preko starog. Na raspolaganju su i automatsko traženje bez zamene, sa zamenom ili sa brisanjem. Brisati se može simbol, reč, red, rečenica, paragraf, ostatak teksta i ceo tekst. Mogu se kopirati ili premeštati delovi teksta sa jednog mesta na drugo.

Formatizovanje teksta se vrši umetanjem format naredbi u tekst na mestima gde želimo početak njihovog dejstva. Tako se mogu postavljati leva i desna margina, privremeno pomerati margine i time dobiti izbačen ili uvučen deo teksta, može se definisati prored, dužina teksta na papiru i fizička dužina papira. Takođe je moguće izvršiti automatsko poravnavanje desne margine, centriranje proizvoljnog broja redova, definisati gornju i donju oznaku stranica kao i stranice, itd.

Pri štampanju teksta, „Easy Script“ opet pokazuje fleksibilnost. Kada završimo formatizovanje i u tekst ubacimo sve format naredbe koje smo želili, možemo štampati uređeni tekst kako na papiru tako i na ekranu, gde se svaka stranica prikazuje u potpuno identičnoj formi kao što će biti na papiru. Nakon toga tražimo „pravo“ štampanje koje može biti bilo neprekidno, bilo stranica po stranica. U toku štampanja pojedine stranice se mogu lako usmeravati na ekran ili na papir i time vršiti selekcija stranica. Ako nam je potrebno više primera-ka istog teksta, navešćemo njihov broj i na

**Nijedan tehnički pronalazak posle Gutenberg nije toliko unapredio mučni računari pisanje kao što to čine (kućni) računari opremljeni specijalnim programima za obradu teksta. Iako su oni još uvek daleko od savršenstva „sedi-i-piši-ne-razmišljajući-kako-se-sprava-koristi“, računare prihvataju ne samo sekretarice, novinari i prevodioci, kojima je presudna lakoća i brzina pisanja, nego čak i pisci, možda najveći tehnički tradicionalisti, kojima je, izvan svega, bitan kvalitet teksta i koji su koliko juče bili spremni da prezru svako štivo koje nije napisano rukom i gušćim perom. Pri tome maglja računara nije u lakoci pisanja. Njihova glavna snaga dolazi od izražaja tek kada, nakon što tekst „odleži“, počne „muka sa rećima“ — kad se odmerava svaka nijansa u značenju, a pojedine stranice prekucavaju nbrojeno puta. „Elektronske makaze“ u računaru mogu u ovoj fazi da uštede i čitave mesece rada. U prošlom broju smo objavili opšti prikaz programa za obradu teksta sa rečnikom žargona. Ovoga puta predstavljamo na delu nekoliko najboljih programa za najpopularnije modele kućnih računara.**

štampanju se, jedna za drugom, pojavljuju kopije.

„Inteligentnim“ štampačima, „Easy Script“ šalje komande koje određuju kada štampati jednim tipom simbola, a kada preći na drugi. Komandne iskepe (escape) sekvence nisu zarobile internim kodom programa za obradu teksta, kao što se to, na primer, dešava u nekim sličnim programima za računare PDP. Lako se postiču podvlačenja, pisanje slova preko slova, korisnički definisani karakteri, kao i komanda „pomeći papir za jedan red“ koju treba poslati nekim štampačima posle kraja svakog reda, a koju računar C-64 sam ne generiše.

Iz „Easy Script-a“ možemo neposredno koristiti sve naredbe kod jedinici koje su na raspolaganju i iz bejzika, kao što su upis na disketu, čitanje sa diskete, formatizacija nove diskete, brisanje imena tekst datoteke, pregled spiska imena sa diskete (direktorijum), promena imena, kopiranje i sl. Možemo povezivati tekst datoteke u niz i time formirati lanac koji se na dalje tretira kao jedan dugačak tekst sa maksimalnom dužinom do kapaciteta diskete (približno 160 kućanih stranica). Ovo povezivanje je dragocena pomoć pri štampanju dužih tekstova.

Ovo je bio samo manji deo mogućnosti programa „Easy Script“. Podržano je još mnogo korisnih operacija sa tekstom kao što su, na primer, uslovno prenošenje dela reči iz reda u red, normalna i decimalna tabulacija, nezavisne margine za tekst i gornju i donju oznaku stranica, mehanizam popunjavanja označenih mesta jedne tekst datoteke sadržajem iz druge, sigurnosni metodi operisanja imenom teksta pri upisu i čitanju (neobično važno), i tako dalje.

### Neke (oprostive) mane

U radu sa „Easy Script-om“ ispoljavaju se i neke mane za koje on nije ni kriv ni dužan. To je, pre svega, prikaz od samo 40 slova u redu, nedostatak koji, zapravo, ima njegov domaćin — računar „Comodor 64“. Pored toga, da bi se počelo sa radom, treba dosta dugo čekati dok se program učita — opet mana sporih disk jedinica.

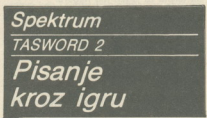
Od „ličnih“ propusta, treba pomenuti komplikovanost korišćenja pojedinih komandi. Kako je „Easy Script“ organizovan za rad u režimima (obrada, komande, disk ili trakica), vrlo često se za izdavanje jedne komande mora pritisnuti i do četiri tastera redom (da bi se prešlo u drugi režim, dao kod za komandu, unela potrebna vrednost parametra i pritisnuto RETURN). Mada, mehanički dobro zasnovane, neke komande se ipak, posle duže upotrebe, jednostavno izbegavaju.

Pomenimo i „tromost“ u reagovanju koji raste sa dužinom unetog teksta. Pri gotovo punoj memoriji, uočljivo je kasnije u reakciji kada se, recimo, imate neko ispušteno slovo. Ali, ako niste isteknući takmičar u brzom kucanju, ovu manu neće ni osetiti.

Čini nam se da je najveća mana „Easy Scripta“ to što nema makro naredbe. Ova mogućnost formiranja linijskog programa sastavljenog od elementarnih naredbi, i njegovog višestrukog izvršavanja pritiskom na taster, predstavlja najfleksibilnije oruđe pri korigovanju teksta, ili sistematskom formatiranju. Uz dodatak makro naredbi, ovaj program bi mnogo dobio na kvalitetu.

Pri odlučivanju o tome da li nabaviti „Easy Script“, treba imati na umu njegove mnogobrojne dobre karakteristike. Tu spadaju mogućnost horizontalnog i vertikalnog pomeranja ekrana po tekstu, kompletan skup format naredbi koje se fleksibilno koriste, povezivanje datoteka u velike tekst-celine, jednostavno upravljanje poslovnim disk jedinice, dobra podrška štampaču, i postojanje posebnih namena, kao što je ukupanje sadržaja dve tekst — datoteke. Kao se tome dodaju da ovaj jeftin program za obradu teksta nadmašuje svojim kvalitetima „starije kolege“ koji su i pet do deset puta skupiji, onda je jasno zašto je stekao veliku popularnost među korisnicima računara „komodor“.

Veljko Spasić



**Kada se razmišlja o ozbiljnoj obradi teksta na računaru, „spektrum“ bi, zaista bio jedna od poslednjih mašina na koju biste pomislili — nestaje mu profesionalna tastatura, prikazuje samo 32 znaka u redu na ekranu i nema ugrađen odgovarajući interfejs za štampač. Ovaj tekst, zaista, ne treba da čitaju oni koji žele savršenu mašinu za pisanje. S obzirom na cene, a i ostale uslove koji vladaju na našem tržištu, velika je verovatnoća da oni takav sistem neće imati tako skoro. Ovaj tekst je za one koji su spremni da, uz relativno male investicije i nešto malo programerskog umeća i**

**truda, dođu do pristojnog sistema za obradu teksta, kome, možda, ne treba previše gledati u zube, ali koji je, ipak, sposoban da obavi posao, i to zaista kvalitetno.**

Da biste „spektrum“ pretvorili u tekst procesor, potreban vam je program za obradu „Tasword 2“, interfejs za štampač i, naravno, sam štampač. Jedna od stvari, poželjna iako ne i obavezna, jeste dodatna profesionalna tastatura.

### Počelnica za pisanje

Program „Tasword 2“ je pravo malo programersko remek-delo i pokriva manje-više svaku osobinu koju bi trebalo da ima jedan tekst procesor. Prva i, verovatno, najvažnija osobina je što tekst prikazuje u širini od čak 64 znaka u redu — idealan format za obradu teksta.

Opis jednog programa za obradu teksta nije nimalo lak zadatak. Verovatno najbolji način da se njegove komande prikažu je opis onim redosledom kojim je to učinjeno u odličnom (kratkom i sadržajnom) uputstvu. Ono radeno baš u „Taswordu“ i odštampano na nekom od „Epson“ štampača.

Prva sekcija uputstva je „kako započeti“ i daje uvod u sam sistem, kao i opis upotrebe „Tasword tatora“ — datoteke koja stiče uz sam program i koja će vas, kada je učitate, voditi kroz sve komande i mogućnosti programa. Za one koji znaju engleski jezik to će, svakako, biti najlakši način da nauče da koriste sve mogućnosti ovog izvanrednog programa — i to samo za jedan dan!

Sledeće poglavlje uputstva objašnjava kako „Tasword“ radi i daje podatke o veličini teksta koja se može odjednom obradivati. Tekst može biti dug 320 linija po 64 znaka u svakoj štici, kada se izračuna, daje solidnih deset kućanih stranica. TV ekran predstavlja prozor u tekst koji se može pomerati gore i dole i prikazuje 22 linije samog teksta.

Pri normalnoj upotrebi, „Tasword 2“ daje 64 znaka u redu. Ipak, ako su vam slova isušila sitna, ekran se može tako podeliti da „prozor“ prikazuje samo 32 znaka u redu, ali se pri tom može pomerati i levo i desno da biste imali pregled nad sva 64 znaka. O ovom pomeranju vodi računa sam program, tako da nema problema oko praćenja teksta.

Kod unosa teksta koristimo tastere kao kod svake pišaće mašine. Zbog složenosti „spektrumu“ tastature, ponekad je potrebno otići u takozvani EXTENDED mode (ekvivalent E kursoru) da bi se pristupilo nekim znacima. Ovo se, na sreću, odnosi samo na one znake koji će se stvarno retko koristiti. „Tasword 2“ upotrebljava „spektrumove“ naredbe na tastaturi kao „spektr“ kontrolne komande (to su, uglavnom, ko-



```

EDIT  - - - HELP POKE
CAPS LOCK  CAPS LOCK Lock
TRUE VIDEO  cursor to word left
              VIDEO cursor to word right
ADDRESS - - cursor movement
ADDRESS - - printer control, chars >>>>
DELETE - - delete character
          - - more Line left
          - - cursor Line
          - - cursor Line character
          - - go to end of text
          - - go to start of text
          - - cursor character text
OFF - - delete line
          - - start of next line
          - - scroll down
          - - scroll up
ENTER - - scroll up
CAPS + SYMBOL SHIFT - enter or toggle extended mode
    
```

```

on Escape OFF
e enlarged 1 key 1
c condensed 1 key 2
w underLine 1 key 5
P italics 1 key 4
i emphasized 1 key 5
j double strike 1 key 6
i slide 1 key 7
P proportional 1 key 7
    
```

mande odštampane crvenom bojom, kao i belo označene komande iznad numeričkih tastera). I slova i kontrolni tasteri se ponajviše ako ih izvesno vreme držimo pritisnutu.

### Osnovne komande . . .

Prva komanda koju treba naučiti je izvedeno korisna komanda EDIT (CAPS SHIFT+1) koja prikazuje takozvani HELP ekran. Ovaj ekran sadrži kratak opis svake funkcije koju možemo ostvariti preko tastature u normalnom režimu. Tu je prikazano dvadesetak funkcija, od kojih ćemo opisati dejstvo najvažnijih.

#### EDIT (caps shift+1)

U normalnom modu prikazuje HELP stranicu (slika 1). Pritiskom na ENTER vraća se na tekst. Ako u isto vreme pritisnemo CAPS SHIFT i SYMBOL SHIFT, dobijamo stranicu za EXTENDED MODE. Treba napomenuti da ćemo se, ako u tom trenutku pritisnemo ENTER, naći u EXTENDED modu samog teksta, što je pogodno, jer tada obično i želimo da ostvarimo neku od tih funkcija.

#### CAPS LOCK (caps shift+2)

Menja velika i mala slova. Informacija o tome se prikazuje u statusnoj liniji na dnu ekrana.

#### TRUE VIDEO (caps shift+3)

Pomera kursor na desni kraj reči koja se nalazi ulevo od kursora. Ova komanda nam omogućava da se kursorom ubrzano krećemo kroz tekst reč po reč ulevo.

#### INVERSE VIDEO (caps shift+4)

Isto kao i TRUE VIDEO, samo što pokreće kursor udesno.

```

<<< T A S B O A R D T W O >>>
Extended Mode control keys

SCROLLING
F - fast scroll down
P - fast scroll up

TO PROMPT
P - print text file
L - large printing on marker
L - large printing off marker

MICROBARRING
C - change window on text
E - clear text file
Z - replace or find text
Z - insert mode on/off
EDIT - help page
ADDRESS - cursor movement

SYMBOL SHIFT and key to type these characters: E 3 0 ~ 1 x 2
    
```

### STRELICE (caps shift+5,6,7 ili 8)

Pomera kursor za po jednu poziciju u naznačenom pravcu.

### GRAPHIC (caps shift+9)

Ovom komandom se pristupa „spektrumovoj“ blok grafici. Ovakvo dobijeni grafički simboli se normalno prikazuju na ZX printeru, a kod profesionalnih štampača se koriste kao kontrolni znaci. DELETE (caps shift+0)

Briše znak ispod kursora i pomera ostatak linije za jedno mesto ulevo.

<= (symbol shift+q)

Pomera tekst koji se nalazi ispod i ulevo od kursora za jedno mesto ulevo.

<> (symbol shift+w)

Centrira tekst u liniji na kojoj se nalazi kursor. >= (symbol shift+e)

Pomera tekst ispod i udesno od kursora za jednu poziciju udesno.

AND (symbol shift+y)

Umeće znak, reč ili red u otkucani tekst.

OR (symbol shift+y)

Skok na kraj otkucanog teksta.

AT (symbol shift+i)

Skok na početak otkucanog teksta.

STOP (symbol shift+a)

Upotrebljava se za prikazivanje menija preko koga vršimo učitavanje, snimanje, štampanje i definiciju kontrolnih znakova za štampače.

NOT (symbol shift+s)

Briše liniju pod kursorom i pomera ostatak teksta za jedan red nagore.

STEP (symbol shift+d)

Preuređuje tekst u pasusu. Ako je u tekstu došlo do izmene, pa je ostalo praznih mesta, one se ovde uklanjaju.

TO (symbol shift+f)

Pomera ekran po tekstu za jednu liniju nadole.

THEN (symbol shift+9)

Pomera ekran po tekstu za jednu liniju nagore.

ENTER

Pomera kursor na početak sledeće linije. Ako je „insert“ (umetanje) uključen, umeće i praznu liniju u tekst.

### . . . i korisne finese

Pored nabrojanih osnovnih komandi, „Tasword 2“ koristi i „spektrumov“ EXTENDED mode za pristup dodatnom setu komandi. One omogućavaju korisniku da ostvari komande kao što su brzi pregled teksta na gore i dole (tasteri F i G), desno

poravnanje teksta (taster E), mogućnost da se reči, na prelazu u novu liniju, ne dele već prenose cele (tasteri J i H), izlaz teksta na ZX štampaču, postavljanje margina (tasteri A, S i D), manipulacija blokovima teksta (tasteri B, V, N i M) izbor broja znakova u liniji (taster C), brisanje teksta u memoriji (taster X), pronalazjenje i zamena reči u tekstu (taster R) i uključivanje i isključivanje „insert“ moda (taster I).

Opcija „Find and replace“ (pronadi i zameni) je posebno korisna jer omogućava na jednu komandu, laku zamenu neke reči u celom tekstu. Pretpostavimo da prilikom kucanja često upotrebljavamo reč „komputer“. Ako se iz nekog razloga predomislimo i poželimo da je zamenimo našom rečju „računar“, to ćemo u „Tasword“ programu lako ostvariti preko ove opcije. Druga, mnogo važnija primena je korišćenje skraćnica prilikom pisanja da bi se povećala brzina rada; skraćnice se kasnije lako zamenjuju punim rečima. Kod upotrebe ove opcije „Tasword“ automatski poravnava (formatira) tekst.

Još jedna zanimljiva opcija je i postavljanje margina teksta, što je posebno korisno ako želimo da istaknemo pojedine pasuse. Kada postavimo marginu, deo ekrana će promeniti boju iz bele u žutu da nam ukaže na njene dimenzije.

Sve vreme dok se nalazimo u EXTENDED modu donja statusna linija će flesovati. Za HELP ekran EXTENDED moda pogledajte sliku 2.

Ako ne odredimo drugačije, „Tasword“ će automatski formatirati tekst sa desnim poravnanjem i reči koje ne mogu da stanu u tekstu, prenosiće u novi red. Taster za novi red (ENTER) treba upotrebljavati zaista samo na kraju pasusa ili ako želite da otkučate praznu liniju.

Kada kucate poslednji znak u nekom redu, „Tasword“ će to prepoznati i ono, kada kursor pređe u novi red, otkučate slovo, cela reč će biti premeštena u novi red. Ako je poslednji znak u redu neki od znakova za interpunkciju, tada će se sledeće slovo u novom redu protumačiti kao početak nove reči i prethodna se neće prenositi.

### Nešto malo drukčije

Pritiskom na STOP taster prelazimo u meni u kome se, pored snimanja i učitavanja teksta sa trake, može ostvariti i povratka u bežik (povratka na tekst je RUN ili GO TO 10). Kada se nadete u bežiku, možete menjati program po volji, prevoditi meni, konvertovati rutine za snimanje i učitavanje za rad sa mikrodravom, a ako ste ambiciozni, možete dodavati i nove funkcije programu, jer ima preko 1K slobodne memorije. Takođe, možete prilagođavati program raznim interfejsima. Zanimljiva mogućnost je izmena HELP ekrana. Njihovi tekstovi se nalaze od adrese 52480 do 55296 (probajte neki POKE). Ako želite da menjate definiciju slova i, eventualno, kreiraite YU set slova, pogledajte adrese 61184 do 62079.

Moja iskustva u prilagođavanju ovog programa su apsolutno pozitivna — bez većih problema sam uveo YU set na štampačima BROTHER EP22 i EP44. Odne sam koristio mogućnost da grafičke simbole zamenim našim slovima i da za svaku njihovu pojavu u tekstu na štampaču pošaljem znak za odgovarajuće slovo praćen znakom za pomeranje glave štampača za jedno mesto unazad i odgovarajućim zna-

kom za interpunkciju, čime sam na pomenutu štampačima dobijao dosta dobre zamene za naša slova. Kod „spektrumovih“ slova definicija blok grafike se nalazi ne u ROM-u već na adresama 60928 do 61055.

Posebno interesantno iskustvo sam imao u povezivanju „spektruma“ sa interfejsom i 1 pisaae mašine sa lepezom OLIVETTI Praxi 41. Problem se sastojao u tome što je lepeza sa našim slovima imala potpuno različito raspored slova u odnosu na ASCII lepezu. U ovom slučaju je bila potrebna intervencija u delovima bezik programa (rutina za štampanje ispod linije 260), kao i definisanje tablice za konverziju kodova tako da za svaki znak na „Olivetiju“ bude poslat odgovarajući znak za YU lepezu. Trud se ipak isplatio, jer je, na kraju, rezultat bio veoma dobar. Gleđajući u izvanredno otkucani tekst, bilo je teško poverovati da je to urađeno na „spektrumu“.

Borivoj Perzić

## BBC B/VIEW

### Izvestan pogled

View je prvi i po mnogo čemu najbolji tekst procesor pisan za BBC B. Delo je Marka Koltona (Mark Colton), programera Acornove „zvanične“ softverske kuće koja se, naravno, zove Acornsoft i na tržištu je već više od dve godine. U međuvremenu je nastalo nekoliko dobrih programa ovoga tipa koje su sastavile nezavisne softverske firme, ali su mnogi vlasnici BBC B ostali verni View-u. Zašto?

„View“ se prodaje u ROM-u 16 K koji se priključuje u jedno od četiri podnožja ispod tastature vašeg računara. Pošto obavite uvek pomalo neprijatnu operaciju, možete da uključite kompjuter, otkucate „WORD“ i doživite prvo malo iznenađenje: za razliku od većine tekst procesora, „View“ nije organizovan „po menijima“.

### Komande umesto menija

Sistem menija je u poslednje vreme sve popularniji među autorima profesionalnog softvera pošto proširuje potencijalno tržište za neki program: za početnika je daleko jednostavnije da izabere jednu od deset opcija čija su imena ispisana na ekranu nego da pamti i otkucava čitave naredbe. Tako, na primer, tekst procesor „Wordwise“ koji je delo firme Computer Concepts po startovanju ispisuje meni koji nam nudi pisanje novoga teksta, učitavanje postojećeg sa trake (ili diska) ili prestanak rada. Izborom neke od ovih opcija dobijamo nove menije i, kada se opredelimo za sve bitne stvari, počinjemo sa pisanjem.

„View“ (=pogled) je koncipiran drukčije: po startovanju programa nalazimo se u takozvanom komandnom režimu u kome

možemo da naložimo računaru da nam pripremi, štampa, formatira i, uopšte, na neki način upotrebi tekst. Pritiskom na dirku „ESCAPE“ prelazimo u režim za pisanje — takozvani tekst mod. U toku rada možemo proizvoljno broj puta da prelazimo iz komandnog u tekst mod i obratno uzastopnim pritiskanjima na „ESCAPE“.

U komandnom režimu možemo da se poslužimo bilo kojom od 22 raspoložive komande. Okoliko zaboravimo sintaksu neke od njih, otkučamo „HELP CMODE“ i videti nešto poput slike 1: svaka naredba je proračena argumentima od kojih su neobavezni smešteni u zagrade. Ubedljivo najčešće korišćene komande su LOAD, SAVE, PRINT i FORMAT.

```
*help stored
VIEW A1.4
CE text          RJ text
DM xx           EM
PE (lines)      EP
OP              BM lines
FM lines        HM lines
TM lines        LM lines
LM margin       LS lines-1
FO 0/1          HO 0/1
HT type code    TS 0/1 margin
SR r value
DF /left/centre/right/
DH /left/centre/right/
```

OS 1.20

U režimu za pisanje ne možemo, jasno, da računamo ni na jednu od komandi sa gornje slike, ali nam one nisu ni potrebne. Potrebne su nam, međutim, neke druge komande, kao što je brisanje reči i paragrafa, umetanje teksta, rasporedavanje slika... Pri izvršavanju ovih zadataka svoju pravu vrednost pokazuju narandžasti tasteri kojima, kao što znamo, mogu da se dodeljuju proizvoljne funkcije. Svakom od tih tastera se, čim uđemo u tekst mod, dodeljuju po tri funkcije, koje izvršavamo pritiskom na taster, pritiskom na SHIFT i taster ili pritiskom na CTRL i taster. Samo se po sebi razume da se najčešće korišćene funkcije kao što je ubacivanje i brisanje slova, reči i redova kao i formatiranje tekućeg paragrafa dodeljeno „nešifrovanim“ tasterima.

Ulogu svake narandžaste dirke je, naravno, teško naučiti a lakob zaboraviti. Zato uz tekst procesor dobijamo specijalni umetak koji stavljam ispod plastične ploče na kojoj piše „BBC Microcomputer“. Ovaj umetak je od neverovatne pomoći u daljem radu; autoru ovoga teksta se događa da i danas, pošto ga intenzivno koristi skoro godinu dana, s vremena na vreme pogleda „šta ono beše CTRL F7?“.

Osim narandžastih tastera, specijalne funkcije su dobile i dirke sa strelicama kao i taster „COPY“. Dirke sa strelicama se, kao i kod svih tekst procesora na svetu, koriste za pokretanje kursora po ekranu i nalazejane nekog mesta u tekstu. Taster COPY omogućava kako kopiranje segmenata teksta tako i postavljanje željenih „lenjira“ u tekst.

### Rad sa lenjirom

„Lenjir“ (ruler) je pojam koji se veoma često koristi kada govorimo o tekst procesorima. Tekst koji pišemo treba, naime, na neki način rasporediti na budućim pisanim stranama: u svakom redu treba da se nađe određeni broj slova, neke redove možda treba uvući da se stvori prostor za sliku, treba vratnati desnu ivicu teksta ili je

ostaviti „slobodnu“... Zato na početku teksta navodimo jedan red koji se sastoji od tačaka. Okoliko, na primer, želimo da pišemo tekst sa 64 slova u redu (standardna kucana strana), otkučavamo red od 64 tačke a zatim prvu od njih zameniti znakom veće (<) a poslednju znakom manje (>). Ova dva znaka simbolično ukviruju tekst koji, opet simbolički, predstavlja tačke. Kada donnije budemo pisali, svaki red će počinjati od levog kraja ekrana dok će se, pošto otkučamo 65 slova, automatski prelaziti na novi red jer računaru (osim ako mu to izričito ne naredimo pritisnuvši SHIFT F2) zabranjuje izlazak „izvan margine“.

Neke od tačaka u lenjiru možemo da zamenimo zvezdicom koja predstavlja tabulator. Kada donnije budemo pritisnuli taster „TAB“, kursor će se pomeriti do sledeće pozicije tabulatora, pri čemu će biti zauzet samo jedan bajt memorije dok bi kucanje odgovarajućeg broja praznih simbola ponekad zauzelo i dvadesetak puta više.

U jednom tekstu ne mora, jasno, da se nalazi samo jedan lenjir. Okoliko nam je, na primer, potrebno da u sredinu nek stani- ce umetnemo sliku, jednostavno ćemo uvesti novi lenjir na kome će leva margina (znak veće) biti pomerenia u desno za onoliko slova koliko će buduća slika zauzmati. Pošto ostavimo potreban broj redova, ponovo u tekst ubacujemo početni lenjir i nastavljamo da pišemo. Čitav ovaj mehanizam (daleko jednostavniji kada ga isprobate nego kada čitate o njemu) je kod tekst procesora „View“ dosledno primenjen i dopunjen nekim pre zanimljivim nego korisnim opcijama. Jedna od njih je i mogućnost da neku od tačaka u lenjiru zamenimo malim slovom 'b'. Kada u daljem radu dođemo do te pozicije, računac će nas na to upozoriti „zvonom“ — baš kao na staroj, dobroj pisačkoj mašini. Na tekst procesoru, međutim, ovakvo upozorenje nema mnogo smisla jer se redovi automatski nastavljaju jedan za drugim, pa nema nikakve potrebe da ih planiramo i pritisakmo dirku RETURN posle svakog od njih.

### Naredbe za pamćenje

Komandni režim, narandžasti tasteri i raznorazni lenjiri nisu, verovali ili ne, bili dovoljni da se realizuju sve komande koje su zamislili „Acornsoftovi“ programeri. Zato su uvedene takozvane „komande za pamćenje“ (stored commands) čiji spisak dobijate kada u komandnom modu otkučate „HELP STORED ili pogledate našu sliku 2.

```
*help code
VIEW A1.4
LOAD file        SAVE file
READ file (1)   WRITE file (1 2)

EDIT file:n:lineout
More (1)        Finish
QUIT

SCREEN file:1...n
Print file:1...n SHEETS file:1...n

MODE            COUNT (1 2)
FIELD chr       FORMAT (1 2)
PRINTER file    MICROSPACE spacing
NEW

Search target (1 2)
Change target result (1 2)
Replace target result (1 2)
```

OS 1.20

„Komande za pamćenje“ se smeštaju u okviru samoga teksta koji pišemo: uvek



možemo da pritisnemo SHIFT f8 i kursor će se naći u levom uglu ekrana, tri znaka ispred početka tekućeg reda. U taj prostor možemo da upišemo dva slova koja u momentu kucanja teksta ne proizvode nikakvo dejstvo — ovi znakovi će postati značajni tek kada se tekst bude štampa!

„Komandama za pamćenje“ definišemo najpre oblik i budući izgled stranice određujući njenu dužinu, broj praznih linija koje predstavljaju gornju i donju marginu, zaglavlje i paginaciju. Kao i kod većine procesora za obradu teksta, naime, možemo da zahtevamo da se na početku svake strane ponavlja jedan, a na njenom kraju neki drugi tekst. Kod „View“-a ovaj tekst ne mora da bude jednak na svakoj strani — raspolažemo određenim brojem numeričkih promenljivih koje mogu da se štampaju kao u jeziku. Jedna od očiti primena ovih promenljivih je obeležavanje strana u tekstu.

PL 72  
 LS 1  
 SR P 1  
 FO 0  
 TM 2  
 HM 2  
 FM 0  
 BM 2  
 DH ///!P./

Na slici 3 vidimo primer komandi za pamćenje koje su neobično zgodne za početak svakoga teksta. Prva od njih definiše dužinu strane (72 reda: u ovaj broj su uračunate sve margine). Sledi određivanje razmaka između redova (LS 0 bi značilo kucanje bez proreda), dodeljivanje vrednosti i promenljivog P (ova promenljiva stalno pokazuje broj tekuće strane), naredbu da

se ??? footer ??? ne piše, definiciju četiri margine te, na kraju, oblik zaglavlja koji će u desnoj ivici svake strane ispisivati njen broj praćen tačkom.

„Komande za pamćenje“ se nalaze i u samom tekstu. Njima, na primer, možemo da postignemo da neki red bude uravnan prema desnoj, a ne prema levoj ivici papira (na primer, ono „S drugarskim pozdravom“ kojim završavamo neko pismo), da drugi tekst bude centriran, da kreiramo „makro tekstove“ (ponekad nam je potreban već broj pisama u kojima će se menjati samo par reči, kao što su ime i adresa onoga kome su upućena) i mnogo drugih sitnica na koje ovde nećemo trošiti previše vremena.

### Povezivanje teksta

Većina tekst procesora omogućava pisanje teksta koji je dugačak samo onoliko koliko dopušta memorija računara. Konstruktori „View“-a su vlasnicima disk jedinice omogućili mnogo više od toga — dužina teksta je ograničena jedino kapacitetom vaše diskete, što znači da on može da zauzima i celih 200 K (oko 140 šlajfni). Pod kontrolom tekst procesora sa diskete se učitavaju segmenti teksta koje onda prepravljamo i, po završetku te operacije, automatski odlažemo na disketu kako bismo oslobodili memoriju za njegov nastavak.

Ovakvo rešenje je imalo jednu veliku manu koju su posebno osetili oni koji rade sa kasetofonom — da bi mogli da se štampaju dugački tekstovi, bilo je neophodno da se svaki tekst koji treba štampati nalazi na nekom spoljnom mediju. To, dakle, znači da čak i pismo od pedeset redova morate najpre da snimate na disketu, pa da ga tek onda štampate. Ukoliko

nešto snimate na disketu, ova operacija će vam oduzeti svega par sekundi; ukoliko, međutim, snimate na traku, ona može da potraje nekoliko minuta ili, ako kasetofon počne da vas muči (što se redovno dešava u najneprijatnijem mogućem trenutku), čitav sat.

### Profesionalno za profesionalce

Razumevajući ovaj problem, „Acornsoft“ je pripremio novu verziju programa, nazvanu „View 2.1“, koja je zamenila postojeću verziju 1.4. U novoj verziji (koju autor ovoga teksta još nije imao prilike da vidi) omogućeno je štampanje teksta iz memorije i ispravljeno nekoliko ne naročito ozbiljnih ali neprijatnih bagova koje „View A1.4“ ima (pokušajte, na primer, da pritisnete f3, taster koji briše tekst sve do kraja reda, nalazeći se na početku nekoga reda).

„View“ je veoma fleksibilan — kao što smo videli u prošlom i ovom broju „Računara“, moguće je predefinisati neke tastere tako da predstavljaju naša slova i sastavi drajver za bilo koji matični štampač koji će ta slova preneti na papir. Dok je način na koji smo definisali slova prilično „prijav“ (u cilju uštede memorije korišćen je prostor koji je namenjen nekim sistemskim potrebama koje se ne koriste u toku rada sa „View“-om), drajveri za štampač su savršeno regularna stvar, koju korisnici ovog tekst procesora često kupuju. Drajver za štampač omogućava, osim predefinisavanja slova, i podvlačenje, upotrebu kosih (italic) i promenu veličine standardnih slova i još što-šta drugo — drajver vam, jednostavno, omogućava da iskoristite sve potencijale vašeg štampača. Jedna od tih mogućnosti se naziva mikrorazmicanje (microspacing) i nije na raspolaganju kod 95% drugih tekst procesora: kada računar uravnava desnu ivicu teksta on, jasno, umesto praznine između reči, čime tekst postaje prilično „razbijen“. Ako koristite mikrorazmicanje, računar će proširiti svaki prazan simbol za samo jedan ili dva reda tačaka, pa će tekst izgledati daleko ravnomernije, pri čemu će desna ivica i dalje biti uravnata.

Dobar drajver za štampač košta desetak funti, a može da se nabavi čak i program koji će, prema vašim specifikacijama, sam generisati drajver koji vam odgovara. Ni jedan od u Engleskoj raspoloživih drajvera ne omogućava, jasno, definisanje naših slova, pa našim čitaocima ostaje jedino da koriste onaj koji smo mi objavili u „Računaru 3“. Obzirom da definicije naših slova zauzimaju dobar deo od 256 bajta koliko drajver sme da ima, jedini specijalni efekat koji on podržava je podvlačenje teksta.

Sve u svemu, „View“ je izvanredan program za obradu teksta za koji vredi dati 60 funti. Treba, međutim, da kažemo da je ovaj tekst procesor namenjen poglavito onima koji imaju iskustva u korišćenju računara i kojima nije teško da koriste apstraktne komande koje ponekad imaju svega po dva slova. Osim toga, „View“ je naročito pogodan za one koji su se opremili solidnim štampačem, disk jedinicom, a možda i dodatnim 6502 procesorom (ovi poslednji treba da kupe takozvani „Hi-View“ čija je cena i dalje 60 funti; radi se o istom programu koji je pomenar na više adrese tako da korisnik dodatnog procesora raspolaže sa pedesetak kilobajta RAM-a u koji može da upiše svoj tekst). Za vlasnike BBC B koji koriste kasetofon i koji su, uz to, početnici verovatno je bolje da nabave „Wordwise“.

Dejan Ristanović



# novе naredbe na novi način

Majstorije na računaru Komodor 64

*Bejzik na računaru „Komodor 64“ je, u praksi, uglavnom ocenjen kao slab: 35 naredbi i 23 funkcije sigurno nisu fond kojim se jedan računar može ponositi. Iako su naredbe korektno izvedene i omogućavaju brz rad, hardverska koncepcija računara daleko prevazilazi ono što se osnovnim bejzikom može iskoristiti. Da bi bio uspešan, ovako zamišljen računar morao je biti potpuno „otvoren“ za proširenja — bejzik interpreter i operativni sistem su napisani tako da omogućuju krajnje jednostavne izmene i dopune. Kako je to postignuto?*

Osnovna ideja je da operativni sistem određene programske celine ne poziva direktno, već preko određene lokacije u RAM... U radnoj memoriji se nalazi stvarna adresa potprograma koji treba da obavi određeni zadatak. Kao ova adresa predstavlja pokazivač, pravak, u kome program treba da krene, to je simbolički nazvana VEKTOR.

Pri inicijalizaciji računara svi vektori se postavljaju na potrebne vrednosti. Kada to jednom uradi, računar više ne proverava koje su stvarne vrednosti vektora, već sa punim poverenjem kreće u smerovima na koje ukazuju. Izmavimo li vektor, a upravo iz tog razloga se njegova vrednost i nalazi u RAM-u, dobićemo jednostavnu mogućnost skretanja u mašinske programe koje sami kreiramo.

## Vektori bejzika

„Komodor“ je izabrao dovoljno veliki broj radnih vektora koje se na ovaj način pozivaju i tako pružio mogućnost gotovo potpune izmene ponašanja računara. U operativnom sistemu moguće je izmeniti editor i ASCII kodove koji se dodeljuju pojedinim tasterima, način komunikacije sa perifernim uređajima i obradu prekida (interapta). Broj vektora u bejziku je znatno manji, ali je njihov izbor izvanredno pogodan.

LOKAC.	VRED.	NAZIV	NAMENA
768.769	58251	PEM	Prijavlivanje grafičke Radni početak bejzika
770.771	42115	BWS	Kodiranje (tokenizacija) naredbi
772.773	42364	TBT	Ispis službenih reči za LIST
774.775	42778	LBT	Upućivanje na izvršenje naredbi
776.777	42930	BCD	Obrada alfa i numeričkih izraza
778.779	44678	BTE	

Lokacije se odnose na memorijske adrese u koje je smešten vektor. Imena smo dali prema početnim slovima engleskih naziva rutina koje se pozivaju odgovarajućim vektorom (Print Error Message, Basic Warm Start, Tokenize Basic Text, List Basic Text, Basic Character Dispatch i Basic Token Evaluation), pa ćemo im se u daljem tekstu, radi kraćeg pisanja, tako i obraćati.

Ovako napravljen izbor očigledno pruža mogućnost dodavanja novih naredbi ili kreiranja sopstvenog programskog jezika. Između vektora bejzika i vektora operativnog sistema postoji jedna razlika. Prvi se postavljaju samo pri uključivanju ili potpunom resetovanju računara, dok se ovim drugim vrednost obnavlja pri svakoj upotrebi STOP+RESTORE tastera.

Na nekoliko jednostavnih primera ilustrovaćemo kako se, promenom vektora, može izmeniti ponašanje računara i dodati mu se nove naredbe.

Mnogi vlasnici „komodor“ računara se žale na način kako je izvedeno listanje programa. Program suviše brzo prelazi preko ekrana, a taster CTRL, kojim se listanje usporava, nije od velike pomoći. Pokazaćemo jedno jednostavno rešenje ovog problema koje je objavljeno (ali sa greškom, nedostaje jedna instrukcija) u knjizi COM-MODORE 64 EXPOSED (autor Bruce Bayley).

Kada računar treba da izlista program, prvo ispisuje linijski broj. Zatim uzima red, bajt po bajt, jednu po jednu programsku liniju i preko vektora LBT poziva rutinu koja ispisuje da li se radi o kodu naredbe ili nekom argumentu da bi se izvršila detokenizacija naredbi, odnosno kodovi prikazali na ekranu kao službene reči. Dakle, pre ispisa bilo kog znaka, osim cifara linijskog broja, računar vrši indirektan skok preko vektora LBT. Vektor možemo izmeniti tako da pokazuje na naš mašinski program, čiji je zadatak da napravi pauzu, a zatim izvrši skok na „komodorovu“ rutinu. Za pravilnije pauze ćemo iskoristiti sistemsku promenljivu na adresi 653, čija se vrednost postavlja pri svakom očitavanju tastature, i to tako da prva tri bita, ako su 1, označavaju da je pritisnut taster SHIFT (bit 0), „Komodor“ znak (bit 1) ili CTRL (bit 2). Program je dat u prilogu 1.

Ako mašinski program upišemo od adrese 49152, onda vektor LBT menjamo sa:

POKE774,0:POKE775,192

Od tog trenutka će pritisak na bilo koji od tri navedena tastera potpuno zamrznu listanje. Može se iskoristiti i SHIFT LOCK, tako da pri dužim pauzama ne moramo da držimo taster. Ako nam ova osobina više ne treba, vektor vraćamo staru vrednost 42778:

POKE774,26:POKE775,167

Treba napomenuti da se promena vektora BWS,BCD i BTE ne može, kao u ovom primeru, vršiti iz bejzika.

## Nove naredbe

Vektor BCD upućuje na rutinu čiji je zadatak da na osnovu koda naredbe iz tablice u ROM-u uzme adresu i pozove potprogram koji tu naredbu izvršava. Ako otkucate: SYS(43680) „GALAKSIJA“, 10'200 biće isipan zadat tekst i rezultat izraza, jer smo naredbom SYS pozvali potprogram koji obrađuje naredbu PRINT. Upravo to radi i navedena rutina: na osnovu koda naredbe PRINT pronalazi u tablici adrese 43680 poziva potprogram na njoj. Sve naredbe bejzika su organizovane kao subrutine i završavaju se instrukcijom RTS, čime se program vraća na deo koji ispisuje da li sledi kraj linije ili znak i kojim se odvajaju naredbe. Zatim se ciklus ponavlja za sledeću naredbu.

U ovu proceduru se možemo umešati izmenom vektora BCD i tako proveriti da li se radi o naredbi koju smo sami kreirali. Potrebno je još upoznati i jedan važan potprogram. Počinje na adresi 115 i njegov disasembliрани listing je dat u prilogu 2. Računar ga poziva svaki put kad treba uzeti bajt iz bejzika. Adresa bajta u LDA XXXX menja direktno u samoj instrukciji uz pomoć tri prve naredbe potprograma. Iz tog razloga je čitav potprogram smeštena u RAM. I ovo se može iskoristiti za dodavanje nove naredbe, ali je put nešto složeniji.

Rutina ima i zadatak da preskoči dve razmake u bejziku, kao i da postavi CARRY zastavicu ako je uzeti bajt bilo šta drugo osim cifre (ASCII kod 48 do 57). Ako je potrebno da ponovo uzmemo isti bajt iz bejzika, možemo upotrebiti JSR 121, jer se tako adresa bajta ne menja.

## Naredba colour

Uvešćemo novu naredbu za čiju ćemo sintaksu izabrati sledeći oblik:

JC r,P,t

R,P i T će biti parametri koji određuju boju rama, podloge i teksta. Naredbu ćemo izvesti tako da se boja teksta menja na celom ekranu istovremeno, čime se mogu postići interesantni efekti.

Zašto smo izabrali da naredba počinje znakom uglaste zgrade? Bez obzira što uvodimo samo jednu novu naredbu, želimo

31) nove naredbe na novi način

da pokažemo opštiji slučaj, kada će ih biti više. Tada će biti dovoljno da u programu ispitamo samo jedan bajt da bi ustanovili da li se radi o našoj naredbi. Ako se ne radi o „J“, onda ćemo program odmah uputiti na standardnu „komodorovu“ rutinu i tako uštedeti znatno vremena. Uglavna zagrada je izabrana potpuno proizvoljno — može se upotrebiti bilo koji znak, ali treba voditi računa da se znaci matematičkih operacija i relacija takođe kodiraju kao i naredbe, pa u programu znak \* dobija kod 172, znak postaje 177 itd.

Samu reč naredbe COLOUR smo skratili na jedno slovo iz istog razloga. Svakako bi najelegantnije rešenje bilo korišćenje pune reči, ali bi tada, zbog brzine identifikacije, bilo neophodno izmeniti i vektor TBT i dopisati program koji će kodirati i naše naredbe. To za sobom povlači i izmenu vektora LBT da bi se kodovi pri listanju ispisivali kao cela reč. Ovaj posao je, svakako, vredan truda ako radimo na većim proširenjima bejzika, ali je preobiman, pa se njime ovde nećemo baviti.

Naredba COLOUR sadrži tri parametra, pa treba naći način da se oni prenesu u mašinski program. Za ovo već postoji

### Prilog 1

```

072          PHA
008          PHP
173 141 002  WAIT  LDA 653
008 251          BNE WAIT
040          PLP
104          JLA
076 026 167          JMP 42778

```

gotova rutina u ROM-u sa početkom na adresi 47006. Ona izračunava numerički izraz i njegovu vrednost, koja mora biti jednobajtna, vraća u X registru. Ako je izraz ili broj sintaksno pogrešno napisan, bejziku program će biti zaustavljen sa odgovarajućom porukom o grešci, tako da smo oslobodni tog dela posla u našem programu.

### Prilog 2

```

115 230 122          INC 122
117 208 002          BNE 121
119 230 123          INC 123
121 173 xxx xxx     LDA xxxxxx
124 201 058          CMP #58
126 176 010          BCS 138
128 201 032          CMP #32
130 240 239          BEQ 115
132 056              SEC
133 233 048          SBC #48
135 056              SEC
136 233 208          SBC #208
138 096              RTS

```

```

Prilog 3
// NAREDBA COLOUR //
//
//uzmi prvi parametar,boja rama
49152:          ,OPT P4
49152:          ** 49152
49160: 032 115 000 COLOUR JSR 115
49191: 032 158 183 JSR 47806
49194: 142 032 206 STX 53288
//
// IZMENA BCD VEKTORA //
//
49152: 169 011          LDA #NEMBCD
49154: 141 008 063 STA 776
49157: 169 192          LDA #NEMBCD
49159: 141 009 063 STA 777
49162: 086              RTS
//
// NOVA BCD RUTINA //
//
//preznak nova naredbe ?
49163: 032 115 000 NEMBCD JSR 115
49166: 201 093          CMP #*?
49168: 240 086          BEQ NEMCOM
//
//bajt basic-a ponovo u A
//pa nazad u basic ROM
49170: 032 121 000 JSR 121
49173: 076 231 167 JMP 42893
//
//sledeci bajt naredbe
//da li je "C" ?
49176: 032 115 000 NEMCOM JSR 115
49179: 201 067          CMP #*C
49181: 240 005          BEQ COLOUR
//
//ako nije, SYNTAX ERROR
49183: 162 011          LDX #11
49185: 076 055 164 JMP 42039
//
//nastavak komodor rutine
//provera kraja linije, programa i
//opet skok preko vektora BCD
49206: 076 174 167 JMP 42826
//
//proveri zarez, boja podloge
49197: 032 253 174 JSR 44797
49200: 032 158 183 JSR 47806
49203: 142 033 206 STX 53281
//
//proveri zarez, uzmi boju teksta
49206: 032 253 174 JSR 44797
49209: 032 158 183 JSR 47806
49212: 142 134 082 STX 646
49215: 138          TXN
//
//pocetak kolor RAM-a
49216: 162 216          LDX #216
49218: 160 000          LDY #0
49220: 132 251          STY 251
49222: 134 252          STX 252
49224: 162 004          LDX #4
//
//promena boje teksta
49226: 145 251          INX STA (251),Y
49228: 200          JSR 47806
49231: 206 251          BNE INK
49231: 230 252          INC 252
49233: 202          DEK
49234: 206 246          BNE INK
//

```

Pošto parametre razdvajamo zarezom, možemo upotrebiti još jednu rutinu iz ROM-a koja proverava da li je sledeći znak zaista zarez i, ako nije, zaustavlja bejzik i ispisuje SYNTAX ERROR grešku. Ovaj pod-program počinje na adresi 44797.

Program koji uvodi naredbu COLOUR dat je u prilogu 3. Nakon uošenja u memoriju, inicijalizuje se sa SYS 49152. Time je računar obogaćen novom naredbom i sve dok ga ne isključimo ili potpuno resetujemo (SYS 64738), ona će nam biti na raspolaganju.

### I malo drukčije

Na kraju nekoliko napomena koje mogu olakšati rad. Neke naredbe zahtevaju da budu izvedene isključivo u komandnom modu (na primer RENUMBER, RENEW i slične). Kako ispitati da li računar obrađuje program ili smo jednu komandu otkucanu direktno? Možemo iskoristiti sadržaj memorijske lokacije 123 (vidi prilog 2) koja predstavlja viši bajt adrese znaka koji se trenutno obrađuje. Ako je vrednost 2, znači da se tekst nalazi u INPUT baferu i linija izvršava u direktnom modu. U slučaju da se radi o programu, bajt će biti 8 ili veći, jer programske linije počinju od adrese 2049.

Opisani metod dodavanja novih naredbi se može iskoristiti i na drugi način. Ako vam se ne dopada što naredba GOTO zahteva da linijski broj bude zadat isključivo ciframa a ne numeričkim izrazom, na primer GOTO 1000+K\*100, ne morate smišljati novi naziv za naredbu koja bi imala ovu osobinu. U vašem programu koji se poziva vektorom BCD ispitajte da li je kod naredbe 137 (GOTO) i ako jeste, napravite skok na vašu rutinu koja će izračunati numerički izraz i onda izvršiti GOTO. Tako će, praktično, standardna „komodorova“ naredba biti izbačena iz upotrebe. Na isti način se može potpuno sprečiti izvršavanje neke naredbe.

Za nazive naredbi mogu se, u nekim slučajevima, upotrebiti i cele reči. Na primer, ako dopisujete program za definisanje funkcionalnih tastera možete upotrebiti naredbu:

JDEF n, „sadržaj“

gde bi bio broj tastera. DEF će u normalnoj proceduri kodiranja naredbi dobiti kod 150, tako da će i dalje biti dovoljno ispitati samo jedan bajt da bi se utvrdilo o kojoj je naredbi i reč.

Zoran Životić

### Ubrzani bejzik

Sporost „spektrumovog“ bejzika, moramo priznati, ne spada u naročito akutne probleme. Vlasnici „spektruma“ u tom pogledu uopšte nisu u beznačajnom položaju, mada, prema velikom broju sve učestalijih izjava, stvar izgleda prilično ozbiljno. Pitanje je, svakako, zašto je to tako, kada je činjenica da ovaj računari radi sa prilično brzim procesorom Z80, na 3,5 MHz? Očigledno, nije greška u samoj mašini, već u programu koji obavlja interpretaciju bejzika.

Može se u interperetu naći zaista mnogo mesta na kojima se baš i nije vodilo računa o brzini.

Treba, međutim, biti realan i racionalno prihvatiti neke stvari. Bejzik je bejzik i od njega ne treba očekivati čuda. A ako ipak, posle pažljive analize, zaključimo da je taj bejzik prosto stvoren za akcione igre(?), onda čemo se posvetiti, naravno, problemu njegovog ubrzavanja.

Najbolje rešenje je nabaviti kompjuter za bejzik. Oni kojima je to komplikovano mogu da „okušaju sreću“ sa programom Richarda Tajtora, koji ovdje objavujemo.

Uključite program „Ubrzani bejzik“ i startujte ga sa RUN. Posle nešto više od jednog minuta sve će biti spremno da na kasetu snimite formirani mašinski kod. Ako u nekoj od DATA linija program otkrije grešku, bićete o tome obavesteni. Ispravite grešku i ponovo izvršite RUN.

Čim se mašinski program snimi, prematajte kasetu i izvršite verifikaciju. Kasnije ćete program uzimati sa kasete pomoću: CLEAR 6405: LOAD "" CODE

Startovanje se vrši sa RANDOMIZE SUR 64082, što je, ustvari, ekvivalentno novoj naredbi NEW: biće izbrisana celokupna memorija, a kontrola se prenosi iz ROM-a u novi bejzik interpreter. Korisnik može da počne sa kucanjem svog programa na sasvim uobičajen način, ili može uzeti program sa kasete. Sve je potpuno isto kao da „Ubrzani bejzik“ i nije u memoriji. Jedino će se programi izvršavati brže.

Odmah da kažemo da ne treba očekivati spektakularne rezultate. Povećanje brzine će se, svakako, primetiti, naročito ukoliko u programu ima mnogo FOR — NEXT petlji i potprograma, pogotovo ako su potprogrami smešteni na kraju listainga, ali to ubrzanje ni iz daleka ne dostiže brzinu jednog mašinska.

Najbolji savet je: probajte — pa donesite odluku sami.

```

10 REM |
20 REM | FASTER BASIC |
30 REM | By Richard Taylor |
40 REM |
50 REM |
60 CLEAR 64057
70 LET a=64058
80 FOR n=0 TO 20: PRINT n+1
90 READ k,a$: LET s=LEN a$
100 FOR t=1 TO s-1 STEP 2
110 LET h=CODE a$(t)-48
120 LET l=CODE a$(t+1)-48
130 IF h>9 THEN LET h=h-39
140 IF l>9 THEN LET l=l-39
150 LET b=16*h+l
160 POKE a,b
170 LET s=s+b
180 LET a=a+1
190 NEXT t
200 IF s<k THEN PRINT "ERROR
IN LINE " ;1000+10*IN: STOP
210 BEEP 0,2,24 CLS : NEXT n
220 SHOW "fast-basic" CODE 64058
,1310
230 PRINT "VERIFY": VERIFY ""CU
DE 64058,1310
1000 DITH 80964,"0000003001323afa
cd50d4afed7b3d5c1fdcb019c3f8fa
c30dd4bb45ced5b385e2a7b5cd93e07d3
f3e3f4ed472ab25c5d5436002bbc20fa
d9e34d3b45ced5338"
1010 DITH 5547,"5e227b5cd9ed53b2
5c21083cc2235ceb363e2b792b2b223d
5c21b655cc24f5c11a15011500ebdb0
eb2b22575c2322535cc2245c36800322
59c3600d23368023"
1020 DITH 6321,"22615c22635cc2265
5c3e38328d5c328f5c32485c21230522
095cf3c3cf4d5ca21c61511105c010e
00ebd0afcd801cecdfd0ef336102fbcd
6b0d47323fa1104"
1030 DITH 5453,"fbcd0a0cfdcb02e2
18480027464153544552804241534943
27204150455241544944e72053595354
45420492792052696368617264205461
79c6c712720202041"
1040 DITH 7170,"75677573742c3139
3834a0fd363102cd951760016afcd01
16cd420cfd171bdfcb007e2000c2a595c
cda711fd3600ff18e42a595c225d5ccd
fb1970b1c237fcdf"
1050 DITH 8125,"fe0d28c7fcd3b046
5c410dcd6e0d3e19fd96473285cfdcb
01fef3600ff4d360a01cd93fc76fdcb
01aef3600c304ecfd8e3a3a53cf52100
00fa7537fd75262e"
1060 DITH 6576,"0b5c21010022165c
c0b01f1dcb37acdfe0dfdc020ef147
f0a380c2007cde1153e206711811913
cd0a0eaf113615cda0ced4b455ccd1b
1a3e3ad7f46d0906"
1070 DITH 6300,"00cd1b1acd97103a
5a3c5c281df092804fe152003f3d40d
01030011705c21445cfdcb0a7e280109
edbf8fd360afffdcb019ec34fb3e1001

```

Interesantnije bi ovde bilo da nešto kažemo o samom programu. Njegov autor je smatrao da najveći problem u pogledu brzine prave naredbe za skok (GO TO | GO SUB), kao i FOR — NEXT petlje. Zato prvo treba razjasniti kako to ove naredbe uopšte rade na „spektrumu“.

Recimo da u programu postoji naredba GO TO 9000. Ako pogledamo u deo memorije gde se ta naredba čuva, vidimo da je zapisana na sledeći način: prvi bajt sadrži 236, što je kodirana oznaka za GO TO. Sledeća četiri bajta sadrže 57,48, 48 i 48, što predstavlja cifre 9, 0, 0, 0. Konačno sledi bajt 1, a za njega pet bajtova za

```

0009e3affbed44349"
1000 DATA 6922,"5c2a5d5ceb212ffc
e52a615c3fed5e256069cde19200bcd
b19cde08193e01323afac1793d00282d
c503030302bed5b35cd5c5d55163e01
323afae122535cc1"
1090 DATA 7571,"c5132a615c2b2bed
b82a495cebc1702b712b732b72f1c345
fb21fe7f22455c2a615c2bed5b95c1b
3a445cc357fd7cdfb16fd340ddf0600
fe0d287afe3a28ee"
1100 DATA 10855,"217f7e2a54fe779f
dcb0c7e200ffecca937dfedcaaa8fd
ef3ca99fdfeea295f5efca81fd0fed52
90faee5ca52fafeefc2441b4f3e01323
afa75c3441bcd541f"
1110 DATA 7997,"d27b1bfdfcb0a7e20
71fdcb097e208a3a3aaf7c477fe2442
5ccdde193a445c282aa72c1b47fe76e
c079281fed7b3d5cfd3600ffc9c1fdcb
0c7e20073a3aaf7"
1120 DATA 7409,"c477fe2a555c3ec0
ac09afed1ce0056e35eed53455c235e
2305eb19232555ceb22d5d5c571e00fd
36aaf15fd720dcaab9f14cd8019e2002
cf16dfed082b7fe"
1130 DATA 9234,"3aca08fca38a1ccd
819cfdcfcc281cc1ef10238bcbde934
3029c3a8fcdfcd2fd9047ed53d5d5e23
5e234ced33afffd1fed600d24e333ed4b45
5cc5e5ed733cd5d5"
1140 DATA 6461,"011400cd05f118d5
2a5d5c7e23fe0e2807cd1b2d30f3e7c9
ebc10500197fe0fed28004fa320f0eb37
5c2d821c3671e2da5c237efe0e20fa
223bf435cd6c1c01"
1150 DITH 6270,"06002a3bfa340922
505c0c374e22e1c244d5ccbc7ead08
1d232269c5f0e20fc000038cdda1d09
2a5fa723e523237e235e23562a685c01
0f0094f7b28e201a"
1160 DITH 7986,"237abe2b21442323
799e20067ee1e52337f71e52364fc3
3aaf75e236c5237eebf0e619f1ed04b
535c7ed425eb1e1732c72d3727123
704fc33aff2a535c"
1170 DITH 8092,"2b11f300cdaeefe38
05c19037a2a5d5c18f011ce0037f52a53
5c0b5cdaeefe3809cdd0fed12a5d5c18
f1d113f13f300eaf323afac97efe3a28
15237eecc0837c023"
1180 DATA 7544,"23223b1fae2346e5
19444ade11600c5cd8b19c1d018e3e7cd
c2f2380611000c238b19fdcf3b2d2a5d
5c25c5cd991e6960b19ed4b3007f0e2
e042bbe173237223"
1190 DITH 5441,"3601232323225d5c
9e7237f7e0d2009f0e2802fe3a20f2
2b010600c0d55162323e5a3bfa5e2356
010600e09eb722b73e1360e06052336
001081b232255cc9"
1200 DATA 3445,"2a535c19cd541f38
5c06235eed53425cfd710acf14d13ec0
a5c079c345fd"

```

binarno reprezentovanje broja 9000. Ovi bajtovi se ne pojavljuju pri listanju programa. Računar ih koristi samo dok izvršava program, da ne bi svaki put prevodio niz cifara 9000 u odgovarajući binarni zapis.

Kada u fazi izvršavanja programa računar nađe na naredbu 236 (tj. GO TO), on prelazi odmah iza bajta 14 i čita linjski broj koji treba skočiti. Zatim počinje da pretražuje program. Kada nađe liniju 9000, tj. njenu adresu u memoriji, odlazi tamo i odatle nastavlja izvršavanje programa.

Slično bi bilo i u slučaju GO SUB naredbe. Jedino bi pre skoka računar, upamtio i mesto odakle je poziv potprograma.



ma usledio, da bi umeo tamo da se vrati kada naide na RETURN.

Ono što u svemu tome ne valja je sledeće: svaki put kada naide na naredbu GO TO 9000 ili GO SUB 9000, računac će iz početka tražiti liniju 9000 po memoriji... Prilično neinteligentno i, uz to, nepotrebno oduzima vreme. Linija 9000, kao i bilo koja druga, se ne pomera za vreme izvršavanja programa. Dovoljno je jednom naći njenu adresu i onda je pamtiti.

Program „Ubrzani bejzik“ radi upravo tako. On pre početka izvršavanja bejzik programa pronalazi sve GO naredbe i zamjenjuje naznačene linijske brojeve iz marker 14 odgovarajućim adresama. Otuda kratka pauza pri startovanju programa. Do predvođenja dolazi kad god se startuje program koji je prethodno podvrgavan izmenama ili učitavan sa kasete. Računar, prosto, vodi računa o tome da ne radi neki posao bez potrebe. Osim toga, GO naredbe iz kojih sledi čitav aritmetički izraz ne podvrgavaju se predvođenju, jer ionako nije jasno na koju će liniju skok uslediti sve dok program ne počne da se izvršava.

Drugi problem jesu FOR — NEXT petlje. Na primer, neka računac naide na naredbu FOR n=0 TO 20 STEP 2. Da bi se to izvršilo, ne treba mnogo raditi: rezervišaće se u memoriji 19 bajtova za promenljivu „n“, u prvi od tih bajtova upišeće se ime promenljive, a onda sledi tri puta po pet bajtova predviđenih za početnu vrednost, konačnu vrednost i korak. Preostala tri bajta čuvaće linijski broj i broj naredbe na kojoj petlja počinje (to je pozicija odmah iza FOR naredbe). Nakon toga, računac nastavlja sa izvršenjem programa.

Kad bude našao na NEXT n, uvećaće vrednost n za vrednost zadatog koraka i, ako treba, skočiti opet na početak petlje. Pre toga će pokupiti linijski broj i pretražiti memoriju: treba mu adresa linije na koju skače. Opet ispisanje vremena. Koliko god puta da se NEXT naredba izvršava, toliko će se puta tražiti programska linija koja je, jednom već nađena.

„Ubrzani bejzik“ omogućuje i naredbi NEXT da pamtí neposredno adresu linije. Pošto inače ne postoji neki „prirodni“ prostor za čuvanje te adrese, kao što je to bilo kod GO naredbi, ovde je taj prostor stvoren veštački: iz svake naredbe NEXT umetnuto je šest novih bajtova. Prvi od njih je 14, a ostalih pet služe za pamćenje adrese na kojoj počinje petlja. Problem je jedino u tome što u fazi predvođenja, pri startovanju programa, nisu poznate adrese petlji. Zato „Ubrzani bejzik“ ostavlja sve metnute bajtove praznim — adrese se preračunavaju tek kad naredbe NEXT počnu da se izvršavaju i kad postane jasno gde treba skočiti, računajući se adrese. Dakle, prvo izvršenje svake NEXT naredbe obavlja se na uobičajeni način, trajući za početkom petlje, dok se u ostalim navratima skok obavlja neposredno.

Bez obzira što je „spektrum“ krajnje nepodesan računac za proširenje ovog tipa, program „Ubrzani bejzik“ se, ipak, moraju staviti i neke zamerke. Na primer, on uopšte ne uzima u obzir naredbu RETURN, koja ima podjednako udela u tračenju vremena kao i GO SUB.

Program koji je jednom startovan u prisustvu „Ubrzanog bejzika“ više nikad neće raditi bez njega.

Sam autor programa ne preporučuje rad sa ekstremno dugačkim programima, jer postoji opasnost da ponestane memorije u fazi predvođenja, kada se iza NEXT naredbi umeću novi bajtovi. Naravno, ništa strašno se neće desiti, osim što ćemo dobiti potpuno neupotrebljiv napola preveden program sa kojim neće umeti da izadu na kraj ni „Ubrzani bejzik“ ni standardni interpreter iz ROM-a.

Problema ima i sa mikrodrajvom, jer svaka njegova upotreba vraća kontrolu osnovnom ROM-u, „Ubrzani bejzik“ se tada može ponovo pozvati sa RANDOMIZE USR 64061, a program će biti sačuvan od brisanja. Na žalost, ovaj trik se ne može koristiti u okviru programa. Naravoučenije: ne koristiti mikrodrajv iz bejzik programa.

Jovan Skuljan

## Spektrum Sintaksni teror (Micro drive syntax terror)

Ako imate mikrodrajv, onda ste sigurno već poludeli kucajući zvezdice, navodnike, tačka-zareze i ostale gluposti svaki put kada treba učitati ili snimiti neki program. Pošto sintaksa naredbi ser Klajvu nije jako strana, ovaj program se potrudio da ublaži posledice. Od sada će iza SAVE, LOAD, MERGE i ostalih naredbi za rad sa mikrodrajvom, automatski biti ispisano „m“; 1; ... Primera radi, za snimanje programa, dovoljno je pritisnuti taster S (SAVE), otkucati naziv programa i pritisnuti ENTER.

Program ima samo jednu liniju, ali datoteka, na žalost, nije kratka. Kada otkucate program, nemojte ga nipošto startovati (sam bi sebe obrisao), već ga prvo snimite na traku (kartrid) naredbom SAVE „m“; 1; „run“ LINE 1. Dakle, kada uključite mašinu i otkucate RUN, program se automatski učitava sa mikrodrajva, startuje, postavi tamnu pozadinu ekrana, smesti mašinic iza vrha memorije i, na kraju, izbrise prvu liniju bejzik programa — da se ona datoteka ne bi vukla kao rudiment. Posle toga, možete normalno da radite — program je potpuno providan za korisnika, osim što zauzima 160 bajtova na vrhu memorije.

Vladimir Kostić

```
1 CLEAR 65204: FOR I=65205 TO
65364: READ X: POKE I,X: NEXT I
: BORDER 0: PAPER 0: INK 7: CLS
: POKE 23658,8: RANDOMIZE USR 65
205: DATA 33,1,0,205,110,25,229,
33,2,0,205,110,25,209,205,229,25
62,254,237,71,237,94,207,255,0,
62,63,237,71,237,86,201,42,243,2
54,126,50,8,92,35,34,243,254,58,
242,254,61,50,242,254,58,59,92,2
03,239,50,59,92,24,94,0,0,0,42,3
4,77,34,59,49,59,34,34,0,1,255,2
45,229,58,59,92,245,255,241,203,
111,32,67,58,242,254,254,0,32,19
4,58,8,92,254,208,40,35,254,209,
40,31,254,210,40,27,254,213,40,3
0,254,214,40,26,254,239,40,22,25
1,2,38,40,18,254,207,32,25,62,1,3
3,250,254,24,12,62,9,33,246,254,
24,5,62,10,33,245,254,50,242,254
,34,240,254,225,241,251,237,77
```

## Spektrum

# Prenos podataka bejzik — mašinic — bejzik

„Spektrumova“ USR naredba, kao što je poznato, ne omogućava prenos podataka između bejzika i mašinskog programa, pa se programeri dovijaju na razne načine, najčešće pomoću PEEK i POKE naredbi. Ovaj kratak program (256 bajtova), smešten u bafuru za štampač, rešava problem. Cela stvar radi vrlo jednostavno. Sadržaj bejzik varijabli A, B, C, D, E, H, I, X, Y, CF (CARY FLAG), ZF (ZERO FLAG) i PC (PROGRAM COUNTER) smešta se u odgovarajuće registre procesora i prelazi na izvršavanje mašinskog programa počev od adrese koja se našla u PC registru. Kada se mašinski program izvrši, sadržaj A, B, C, D, E, H, I, X, Y, CF, ZF registra procesora prebacuje se u odgovarajuće bejzik varijable. Na primer, to izgleda još jednostavnije. Pretpostavimo da treba pozvati mašinsku rutinu na adresi 50000 i to za sledećim stanjem registra procesora: A=10, B=20, C=50 i CF=1. Ta se na izlazu pogledati vrednosti H i L registra. Program izgleda ovako:

```
10 LET PC=50000: REM adresa mašinske
rutine
20 LET A=10: LET B=20: LET C=50: LET
CF=1: REM početne vrednosti
30 LET D=0: LET E=0: LET H=0: LET
L=0: LET IX=0: LET IY=0:
30E LET ZF=C: REM ostale varijable moraju
da sadrže neku vrednost, inače će se
program u liniji 40 izvršiti porukom „Variable
not found“
40 RANDOMIZE USR 23296: REM poziva se
mašinski program
50 PRINT H: PRINT L: REM izlazni podaci
Prilikom upotrebe ovog programa, treba
paziti na jednu sitnicu: ako se pozivaju
rutine iz ROM-a, IY treba da sadrži 23610.
```

Vladimir Kostić

```
UTKURJATE PRVO HEX LOADER A ZA-
TIM POMOĆU NJEGA HEX LIST, KAD
KURJATE HEX LIST, ENTER PRITISNI-
TE TEK IH KRAJUJ SVAKE LINIJE.
PRVIM USRIM SRUJTOVA IZ SVAKE LI-
NIJE IDE U MEMORIJU, A DEVETI
SLUZI KRU CHECKSUR.
```

```
10REM CTRL CODE HEX LOADER
20 BURDER 1: PAPER 1: INK 7: P
UKE 23608,20
30 CLEAR 26599: POKE 23658,8:
PRINT AT 20,0
40 INPUT „START HDR:“:A
50INPUT „R:“:”/LINE A#IF A#
=“THEN STOP
60 LET C#0: IF LEN A#>18 THEN
GO TO 30
70 FOR I=1 TO 17 STEP 2: LET X
=16+CODE A#(I)-768-112*(A#(I)+9
)+CODE A#(I+1)-48-7*(A#(I+1)+9
): IF I<17 THEN POKE A#,X: LET
A#A#1: LET C#C#X
80 NEXT I: LET A#A#0: IF C#256
PRINT C/(256)*K THEN PRINT R:”
“A#” LET A#A#0: GO TO 50
90 PRINT #1: FLASH 1:”ERRKRI“:
BEEP 1,1: GO TO 50
```



```

40 REM PROGRAM DISK-LISTER
40 REM
40 REM PROGRAM LISTA SEKVENCIJALNE
60 REM DATOTEKE ILI PROGRAME SA DISKA
70 REM A PRII TOME SE NE SUDI PROGRAM
80 REM KOJI JE TRETNUTU U MEMORIJI
90 REM
100 REM *****
110 REM NAREDBE NE TREBA KUCATI !!!!!
120 REM
130 REM ***** BOSDAN PETROVIC *****
140 A=0
150 T=1
160 READ I:IF I=1 THEN PRINT:STARTI SYS
49152,"CHR$(I+1)";"MI I";CHR$(I+1);END
170 IF A=255 THEN T=T+1;POKET=49152,I;I
GOTO20
M=PEEK(64)+256+PEEK(64)
190 IF A<0 THEN PRINT "GRESKA U REDU "
I: A=I+1;GOTO160
200 PRINT "RED "I;" "OK="A;GOTO160
210 A=A+1
220 GOTO160
230 DATA32,253,174,32,158,173,32,163,182
,165,100,135,167,165,101,133,168,2331
240 DATA160,0,177,167,157,31,195,209,192
,3,144,246,173,31,195,201,20,174,2464
250 DATA30,201,4,144,36,160,0,173,33,195
,133,168,173,32,195,133,167,177,2154
260 DATA 16
270 DATA35,34,195,200,204,31,195,209,24
5,76,83,192,169,111,160,194,30,20,2679
280 DATA171,76,116,164,169,136,160,194,7
6,70,192,162,0,189,34,195,141,30,2275
290 DATA195,201,80,240,8,201,83,208,233,
160,4,208,2,160,232,189,34,195,2353
300 DATA201,76,240,8,201,48,208,215,200
,200,232,189,34,195,201,58,208,205,2937
310 DATA140,14,195,174,31,195,169,44,72
,157,34,195,232,173,30,195,157,34,2241
320 DATA195,104,232,157,34,195,232,157
2,157,34,195,169,0,232,157,34,195,2573
330 DATA238,31,195,173,40,3,141,15,195,1
73,41,3,141,16,195,169,34,141,30,195,16
340 DATA3,149,194,141,41,73,31,195,194
2,3,160,195,32,189,255,169,2,162,2313
350 DATA8,160,2,32,186,255,32,192,255,16
2,2,32,198,255,172,14,195,185,171,2508
360 DATA194,141,17,195,169,0,182,174,141,1
6,195,108,17,195,169,0,141,19,195,2276
370 DATA194,20,195,141,21,195,32,0,370
5,194,176,3,76,233,193,32,207,255,2291
380 DATA187,157,195,165,144,206,25,232,224
8,144,233,141,21,195,169,0,232,224,2719
390 DATA8,176,7,157,22,195,232,76,19,193
32,76,194,173,20,195,32,42,194,169,58,210
400 DATA19,195,32,42,194,169,58,32,210,2
55,162,0,189,22,195,32,35,194,232,2267
410 DATA224,8,144,245,32,76,194,169,18,3
2,210,255,162,0,189,22,195,41,127,2343
420 DATA201,32,176,4,169,46,208,3,189,22
,195,32,210,255,232,224,8,144,232,2582
430 DATA169,13,32,210,255,173,21,195,208
,17,24,173,19,195,105,8,141,19,195,2172
440 DATA144,3,238,20,195,76,149,192,76,2
33,193,32,81,194,32,81,194,162,255,2645
450 DATA32,81,194,165,167,232,157,179,19
4,224,3,144,243,32,81,194,232,165,219
460 DATA167,157,179,194,208,243,232,157
,179,194,232,157,179,194,169,179,135,3155
470 DATA167,157,194,169,135,3155,169,1
67,160,194,132,66,194,169,167,173,2212
0,3,72,173,1,3,72,169,34,141,0,3,1773
480 DATA169,194,141,1,3,32,195,166,104,1
41,1,3,104,141,0,3,32,15,194,176,1815
490 DATA177,76,233,193,32,207,255,32,210
255,148,144,208,8,32,189,174,26152
500 DATA241,76,233,193,32,204,255,169,2,
32,195,255,173,15,195,141,40,3,173,2627
510 DATA16,195,141,41,3,169,76,160,194,3
2,30,171,165,203,201,60,208,250,169,2504
520 DATA19,195,198,96,11,195,169,167,173
63,208,8,165,203,240,6,201,10,208,2562
530 DATA248,56,96,24,96,72,169,32,32,72
255,104,72,74,74,74,32,66,194,2094
540 DATA148,104,41,15,32,66,174,72,152,3
2,210,255,104,74,74,210,258,2,103,246,2361
550 DATA144,2,105,6,105,58,96,169,32,76
210,255,32,207,255,133,167,165,144,2361
560 DATA240,5,104,104,76,233,193,96,13,3
2,32,32,38,32,83,80,65,67,69,32,1638
570 DATA13,0,13,71,82,69,83,75,65,98,32,
68,85,71,65,67,75,79,32,73,77,69,1322
580 DATA32,33,33,13,0,13,71,82,69,83,75
65,98,32,75,79,32,73,77,69,83,75,1266
590 DATA32,70,79,82,77,65,84,32,85,76,65
90,65,32,33,32,13,71,82,69,83,75,1365
600 DATA192,193,233,192,1025
410 DATA =1

```

READY.  
36) biblioteka programa

## „galaksija“

### Struja

Momci iz Elektrodistribucije nelscrpni su u kreiranju novih ideja kako da vam na što jednostavniji način izvrše obračun utrošene električne energije ili, kako se to popularno kaže, „struje“. Do sada vam je za tu matematiku bilo dovoljno četiri razreda osnovne škole, a sada vam treba barem osam. Pri tome se verovatno polazilo od pretpostavke (optimističke) da je tek svaki dvadeset Beogradinac nepismen, a taj, ustalom, i ne troši puno, pa i nije tako bitan.

Program „Struja“ je raden za potrošače koji imaju jednotarifno ili dvotarifno brojilo. Vođeno je računa o „višoj“ (zimskoj) i „nižoj“ (letnjoj) sezoni. Vase je samo da odgovorite na tri pitanja koja vam bude uputio program počto pa upišete i startuje u na svojoj „galaksiji“. Možete praviti (preporučujemo vam i simulacije) teste: Šta ako samo noću uključujem bojler pa tako potrošim 100 kWh „jeftine“ struje više, ali istovremeno 100 kWh „skupe“ struje manje. Pouke koje bude izvukli biće veoma korisne za vas dteup. Program možete demonstrirati i komšijama i komšinicama — biće vam zahvalni.

U principu, dve su osnovne poruke ovog programa i elektrodistribucije: a) energija je veoma skupa (... b) energija je naročito skupa pre podne (kad radi industrija), i preduke, počto posle ručka odspavate, a domaćin krenu u kuvanje ručka za sutra, pranje veša i sl.

Na kraju, elementi za izradu ovog programa su uzeti sa malog belog lista koji se može dobiti na šalterima Elektrodistribucije Beograd a na kome su objašnjeni novi načini naplate „struje“ koji važe od 1. aprila 1985. Nije aprilska šala.

Ivan Nador

```

10 I *** STRUJA ***
20 I NADOR IVAN
30 I APRIL 1985
31 H.
32 P. *OVO JE PROGRAM KOJI VRSI
OBRAČUN UTROŠENE ELEKTRICNE
33 P. *ENERGIJE PO NOVOM
NACINU OBRAČUNA*
34 P. *ELEKTRODISTRIBUCIJE
BEDGRAD OD 01.04.1985*
35 F.H=1 TO 500:N.H
36 H.
50 P. *UNESITE PODATAK O VRSTI
BROJILA*
55 P. *JEDNOTARIFNO (1) ILI
DVOTARIFNO (2)*
60 I X
65 IF X=1 G.600
70 IF X=2 G.290
285 I DVOTARIFNO
290 G.1000
293 I.Y
295 IF Y=1 G.400
298 IF Y=2 G.300
300 INIZA SEZONA - DVOTARIFNO
310 S=247.94 I V=5.83 I H=2.91
315 P. *UNESITE BROJ POTROŠENIH
KILOVATA*
320 P. *VIŠE TARIFE *I:I.A
321 D=A*V
325 IF A<100 G.330 E.G.340

```

```

330 A=100
340 P. *NIZE TARIFE *I:I.B
350 G.460
400 I VISA SEZONA - DVOTARIFNO
405 S=371.92 I V=8.74 I H=4.37
410 P. *UNESITE BROJ POTROŠENIH
KILOVATA*
420 P. *VIŠE TARIFE *I:I.A
425 D=A*V
430 IF A<100 G.440 E.G.450
440 P. *NIZE TARIFE *I:I.B
446 P=(A/100)*S
480 I X
500 P. *VASE RACUN SE GASTOJI IZ
TRI DELA I TOI*
501 P.
510 P. *ZA ANGAŽOVANU SNAGU *I:P
520 P. *ENERGIJA (NIZA TARIFA) *I:D
530 P. *ENERGIJA (VIŠA TARIFA) *I:D
540 U=P+D*F
548 P. *-----
552 P. *STO UKUPNO IZNOSI *I:U
560 S.
560 I JEDNOTARIFNO HERENJE
620 C.1000
650 I X
660 IF Y=1 G.700
670 IF Y=2 G.680
680 INIZA SEZONA - JEDNOTARIFNO
685 S=247.94 I V=4.66
690 I X
700 I VISA SEZONA - JEDNOTARIFNO
710 S=371.92 I V=8.74
730 P. *UNESITE BROJ POTROŠENIH KILOVATA*
740 I X
750 IF K<100 G.760 E.G.770
760 K=100
770 P=(K/100)*S
780 G.900
900 P. *VASE RACUN SE GASTOJI IZ DVA DELA
I TOI*
902 P.
910 P. *ZA ANGAŽOVANU SNAGU *I:P
920 P. *UTROŠENU ENERGIJU *I:C
930 U=P+C
935 P. *-----
940 P. *STO UKUPNO IZNOSI *I:U
950 P.
1000 P. *UNESITE SEZONU*
1001 P. *VISA SEZONA (1) OD 01.X DO 01.IV
1002 P. *NIZA SEZONA (2) OD 01.IV DO 01.X*
1003 R.

```

## „galaksija“

### Korpa

Evo, u doba izražene inflacije, jednog programa pomoću koga možete odrediti proram troškova ishrane. Program ne treba mešati sa troškovima života, jer su, u tom slučaju, uključeni i drugi troškovi (stanovanje, odeća, obuća i sl.). Ova „korpa sa namirnicama“ slična je tzv. „Evropskoj korpi“ koja na sličan način ali malo drugačije popunjena služi zavodima za statistiku za određivanje troškova ishrane u pojedinim zemljama.

Treba, pri tome, znati da se pri određivanju cena jedne „korpe“ prave razne aproksimacije. Kilogram jabuka se razlikuje po ceni, na primer, u Portugalji, Švajcarskoj i Finskoj ne samo zbog troškova transporta, ponude i potražnje nego i zbog kvaliteta. Na cenu utiče podneblje, kupovna moć države i njenih stanovnika i sl.

Nemoćmo doći do istih rezultata ako ovu „korpu“ platimo na kasijedne samoposluge u Mariboru ili Tetovu. Cene kajmaka u Čačku i u Dubrovniku se, sasvim sigurno, dosta razlikuju.

Inače, u ovoj „korpi“ se nalazi:

```

hleba 6.3 kg
pšeničnog brašna 7.1 kg
testine 0.4 kg
pirinča 0.3 kg
mesa 2.9 kg
viršli (mesnih praradevina) 0.9 kg

```



10 I \*\*\* KORPA \*\*\*  
 15 I NADOR IVAN  
 20 I MART 1985.  
 50 H,  
 60 P \*TROSKOVI ZIVOTA - PRE SVE  
 GA HRA-NE STALNO RASTU\*.  
 80 P \*OVO JE METODA DA USTANOVI  
 TE KO-LIKO RASTU TROSKOVI ISHRA  
 NE\*.

100 P \*U PITANJU JE JEDNA PROSE  
 CMA KOR-FA HRANE NAFUKJENA ZIVOT  
 NIK NA-MIRNICAMA\*.  
 120 P \*OD VAS SE ZAHEVTA DA UME  
 SETE CE-NE ODRJEDJENI PREHRANEN  
 IH ARTIKALA KAKO VAS VEC BUDE P  
 ITAD NA-CINJA  
 130 F.X=1T015001.NX  
 140 P \*.  
 150 C.3000\*P."HLEBA"11.A  
 160 T.A(1)1:W=AA(1)1  
 180 C.3000\*P."BRANJA"11.B  
 190 T.A(2)1:W=AA(2)1  
 210 C.3000\*P."TESTEMINE"11.C  
 220 T.A(3)1:W=AA(3)1  
 250 C.3000\*P."PİRINCA"11.D  
 260 T.A(4)1:W=AA(4)1  
 280 C.3000\*P."GOVUDELJE HESA"11.E  
 290 T.A(5)1:W=AA(5)1  
 310 C.3000\*P."MESNIH PRERADJEVINA"11.F  
 320 T.A(6)1:W=AA(6)1  
 340 C.3000\*P."MASTI"11.G  
 350 T.A(7)1:W=AA(7)1  
 360 P.\*CENA LITRE JEŠTIIVOG ULJA"11.H  
 370 T.A(8)1:W=AA(8)1  
 390 C.3000\*P."MARGARINA"11.I  
 400 T.A(9)1:W=AA(9)1  
 410 P.\*CENA LITRE HLEKA"11.J  
 420 T.A(10)1:W=AA(10)1  
 430 C.3000\*P."PIŠTA"11.K  
 440 T.A(11)1:W=AA(11)1  
 450 P.\*CENA JEDNOG JAJETA"11.L  
 460 T.A(12)1:W=AA(12)1  
 470 C.3000\*P."KROMPIRA"11.M  
 480 T.A(13)1:W=AA(13)1  
 490 C.3000\*P."FASULJA"11.N  
 500 T.A(14)1:W=AA(14)1  
 510 C.3000\*P."KISELOG KUFUSA"11.O  
 520 T.A(15)1:W=AA(15)1  
 530 C.4000\*P."KONZERVIRANOG POVRKA"11.P  
 540 T.A(16)1:W=AA(16)1  
 550 C.4000\*P."CRNOG LUKA"11.R  
 560 T.A(17)1:W=AA(17)1  
 570 C.4000\*P."JABUKA"11.S  
 580 T.A(18)1:W=AA(18)1  
 590 C.4000\*P."DIZENA"11.T  
 600 T.A(19)1:W=AA(19)1  
 610 P.\*CENA LITRE VUČNOG SOKA"11.U  
 620 T.A(20)1:W=AA(20)1  
 630 C.4000\*P."ZIVINSKOG MESA"11.O  
 640 T.A(21)1:W=AA(21)1  
 650 C.4000\*P."ŠEĆERA"11.Y  
 660 T.A(22)1:W=AA(22)1  
 670 C.4000\*P."MEDA"11.Z  
 680 T.A(23)1:W=AA(23)1  
 700 P.\*CENA LITRE RAKLIJE"11.X  
 710 T.A(24)1:W=AA(24)1

720 P.\*CENA JEDNOG DNEVNOG LISTA"11.V  
 730 T.A(25)1:W=AA(25)1  
 1000 06.37.1.0.0.4.0.3.2.9  
 1001 10.0.0.0.0.0.0.1.7.7  
 1002 01.3.14.3.3.0.7.0.3  
 1003 00.1.1.1.0.0.5.0.3  
 1004 01.1.3.0.1.1.3.1  
 3000 P.\*CENA KILOGRAMA"1.R.  
 4000 P.\*CENA KILOGRAMA"1.Y  
 5000 P.\*CENA KORJE JE 1"1.M"1 DINARA"1.S.

crnog luka 1,0 kg  
 jabuka 1,8 kg  
 džemova 0,5 kg  
 voćnih sokova 0,3 l  
 živinskog mesa 1,0 kg  
 šećera 1,3 kg  
 meda 0,1 kg  
 rakije 1,0 l  
 dnevnih listova 31 komad

Prilikom pripreme podataka za računar potrebno je da saznate — zapišete cene pomenutih proizvoda i to za kilogram, litar, komad. Ostali deo posla obavlja ovaj program i računar „galaksija“, koja mora imati barem 4 K RAM-a.

Ovakva „korpa“ je na osnovu nekih prosečnih cena januara 1981. koštala 1553 dinara, a samo godinu dana kasnije 2116 dinara. Koliko to košta danas ostavljaj vama da proverite.

Podaci o količinama (6,3 kg hleba, 7,1 kg brašna itd.) gore navedenim redom nalaze se u linijama 1000 do 1004. Ako u vašoj porodici trošite više, na primer, hleba, a manje mleka, možete izmeniti podatak prvi u liniji 1000 i peti po redu u liniji 1001. Podaci su međusobno odvojeni zarezima. Ivan Nador

## Amstrad/Schneider Clockman

Ovo je prerađena ideja programa za „spektrum“. Radi se o dinamičnoj i efektivno ozvučenoj igri. Cilj igre je uništiti neprijateljske brodove koji se pojavljuju oko vas kao cifre na časovniku. Igra ima 5 nivoa, a svaki sledeći je brži, ti, brod kraće čeka da ga oborite. Morate oboriti najmanje 25 brodova, inače ne možete na viši nivo. Igra pruža velike mogućnosti dorade, u zavisnosti od mašte. Vlasnici kolor monitora mogu dodati boju. Vlasnici AMSTRADA: kod naredbi JOY umesto znaka = treba da stoji naredba AND, a linija 340 da glasi: 340 IF JOY(0) AND 0 THEN 360 ELSE 340. Program je dugačak 2108 bajtova. Pre ukucavanja stavite AUTO.

### A. Koreniji

```
10 SYMBOL AFTER 143
20 SYMBOL 144,2,13,50,196,196,50,13,3
30 SYMBOL 145,192,176,76,67,44,40,16,16
40 SYMBOL 146,24,24,36,36,66,90,165,195
50 SYMBOL 147,3,13,50,194,52,20,8,8
60 SYMBOL 148,192,176,76,34,34,76,176,19,19
70 SYMBOL 149,8,8,20,52,194,50,13,3
80 SYMBOL 150,195,165,90,66,36,36,24,24
90 SYMBOL 151,16,40,40,44,67,76,176,19,2
100 SYMBOL 152,24,60,60,90,255,24,165,255,1
29
```

```
110 MODE 1
120 qw=60:nv=1:ab=159
130 GOTO 460
140 go=1:sc=0:as=1:as=152
150 qw=1:cl=144
160 if cl=1
170 l=INT(RND*8+1):SOUND 1,ab,14,7
180 go=go+1
190 IF k=1 THEN d=11:c=6
200 IF k=2 THEN d=11:c=9
210 IF k=3 THEN d=11:c=16
220 IF k=4 THEN d=11:c=23
230 IF k=5 THEN d=11:c=26
240 IF k=6 THEN d=18:c=23
250 IF k=7 THEN d=21:c=16
260 IF k=8 THEN d=18:c=9
270 FOR =1 TO q
280 LOCATE c,d:PRINT CHR$(as):LOCATE 12,23:PRINT "vreme":x:LOCATE 13,25:PRINT "scot":isc
```

```
290 LOCATE 16,11:PRINT CHR$(j)
300 IF JOY(0)=8 THEN j=j+1:SOUND 1,30,7,7
7:1F JOY(1)=152 THEN j=144:GOTO 290
310 IF JOY(0)=4 THEN j=j+1:SOUND 1,30,7,7
7:1F j=143 THEN j=151:GOTO 290
320 IF JOY(0)=16 THEN THEN GOTO 470 ELSE GOTO 340
330 GOTO 350
340 IF JOY(0)<0 THEN GOTO 340
350 GOTO 360
360 IF q=30 OR sc=25 THEN THEN GOTO 370
370 IF q=30 THEN LOCATE c,d:PRINT " !510 0,170
380 NEXT
390 qw=20 AND sc=25 THEN MODE 0:LOCATE 3,15:PRINT "B R A V O !!!":END ELSE CLS:LOCATE 13,12:PRINT "S c o r !":S c
400 LOCATE 1,24:PRINT "pritisci paljbu z a dalje, dremc skraj"
410 IF JOY(0)=16 THEN SOUND 1,30,20,7:GOTO 330
420 IF JOY(0)=8 THEN THEN END ELSE GOTO 410
430 IF sc=25 THEN 450
440 GOTO 460
450 ab=117:IF nv=4 THEN qw=10:nv=5 ELSE qw=15:nv=mv+1
460 CLS:LOCATE 12,15:PRINT "I V O
"nv:FOR i=1 TO 2000:NEXT:CLS:GOTO 140
470 IF i=145 THEN ab=134:b=344
480 IF i=144 THEN ab=50:b=230
490 IF i=146 THEN ab=240:b=396
500 IF i=147 THEN ab=360:b=232
510 IF i=148 THEN ab=400:b=340
520 IF i=149 THEN ab=364:b=114
530 IF i=150 THEN ab=240:b=70
540 IF i=151 THEN ab=135:b=120
550 PLOT 240,230:DRAW a,b
560 IF k=143 THEN ENV 2,5,-3,30:SOUND 1,302,50,6,7,1,5:LOCATE c,d:PRINT CHR$(230):FOR i=1 TO 250:NEXT:LOCATE c,d:PRINT CHR$(253):FOR i=1 TO 250:NEXT:CLS:sc=c+170 ELSE as=152:ab=119:as=116:GOTO 170
570 FOR i=240 TO 119 STEP 8:SOUND 1,1,8,7:NEXT:PLOT 148,230:DRAW a,b:FOR i=1 TO 200:NEXT:CLS
580 GOTO 330
```

### Nastavak sa str 41

mora da bude maksimalno kratak (da ne bi došlo rad računara) i ne sme da bude destruktivan ni za jedan registar procesora. Takođe, RESET mora da bude uključen, da bi cela stvar radila. Drugi link se nalazi na adresi 33070. Taj link se poziva u slučaju da programer pritiskom na CAPS/ENTER aktivira RESET. Deaktiviranje CALL naredbe koja je stavljena u link vrši se tako što se u svu tra bajta linka postavlja instrukcija NOP (nikako ne RET).

Radni prostor u disasembleru je dovoljan da primi maksimalno 100 labela. Ako se slučajno desi da korisnik disasembleru program koji sadrži više od sto labela (što je vrlo malo verovatno), onda će se pri povratku u osnovni meni na dnu ekrana pojaviti poruka upozorenja. Tada je jedino bezbedna opcija SAVE/LOAD. Sve druge opcije mogu da dovedu do kraha sistema.

Rutina na adresi 33410 kao rezultat daje broj koji je smešten u AS. Ako je recimo AS=0EA60", onda će PRINT USR 33410 na ekranu odtampati broj 60000. U AS može da bude smešten neki dekadani, heksadekadani ili binarni broj.

Rutina na adresi 33228 deluje vrlo jednostavno kao gore opisana. Rezultat će na ekranu biti odtampan u dekadnom, heksadekadnom i binarnom obliku. Ova rutina daje lepše rezultate na ekranu ako se poziva sa RANDOMIZE USR a ne sa PRINT USR.

Na više puta, pomoću opcije LABEL ADDR\*/START PROGRAM, pozivate neki mašinski program sa jedim istim ulaznim podacima, onda ih možete staviti u datoteku (između navodnika u liniji 780) da bi izbegli stalno kucanje.

Vladimir Kostić

masti 0,8 kg  
 ulja 0,8 l  
 margarina (biljne masnoće) 0,1 kg  
 mleka 7,7 l  
 sira (mlečnih proizvoda) 1,3 kg  
 jaja 14 komada  
 krompira 3,3 kg  
 pasulja 0,7 kg  
 kiselog kupusa 0,3 kg  
 konzerviranog povrća 0,1 kg

# no va kruna Majstorije za starog kralja na računaru

Engleski „Devpak“ na jugoslovenski način

Kičma celog sistema su, naravno, dobro poznati GENS3M i MONS3M, obogaćeni softverskim resetom, jedan kratak 'bejzik' program i skoro tri kilobajta mašinskih rutina. Dine to čini jedan vrlo kompaktan blok dužine 20 K koji se sa trake učitava iz samo jednog dela — dakle maksimalno brzo.

Kad se program učita, na ekranu se pojavljuju osnovni meni, koji programeru nudi jednu od 18 mogućih opcija. Pre nego što krenemo na detaljan opis, neće biti na odmet da malo bolje razmotrimo organizaciju memorije.

RAMTOP je postavljen na 26500. Ispred RAMTOP-a je, naravno, bejzik program, koji je zbog malog prostora na raspolaganju, nafilovan VAL-ovima. Bejzik predstavlja neku vrstu komandnog programa. Sve to je, naravno, moglo da bude i na mašincu, ali bejzik ima tu prednost da se lako modifikuje — sa dve-tri intervencije određena opcija lako se prilagodava trenutnim potrebama. Tu leži glavna snaga ovog programa. Iznad RAMTOP-a, na 26501, počinje MONS3M. Zatim, na 31650 počinje blok mašinskih rutina, dužine 2600 bajta, koje se pozivaju iz bejzika da bi se izvršila neka opcija programa. Na 34252 (85CC heksadekadno, lakše se pamti) nalazi se prazan prostor dužine 200 bajta — to je radni prostor (workspace) koji MONS3M koristi pri disasembiranju nekog programa. Iza radnog prostora nalazi se još jedan blok mašinskih rutina, ovoga puta dužine samo 200 bajta i, najzad, na 34700 počinje GENS3M. GENS3M smešta izvorni kod počev od adrese 43572. U stvari, od te adrese pa sve do kraja — dakle ukupno 21 K — memorija je potpuno slobodna.

Izbor određene opcije vrši se pomoću strelice koja se pomera gore/dole pomoću tastera 6 i 7, a pritisak na nulu, odnosno ENTER, pokreće program. Mnoge opcije zahtevaju od korisnika da unese neke podatke. Ako se to ne uradi, to jest samo pritisne ENTER, program se automatski vraća na meni. I još nešto: svi brojevi koji se kucaju mogu biti u dekadnom, heksadekadnom (sa predznakom #) ili binarnom (sa predznakom %) obliku. Najzad eva i spiska svih opcija.

**ASSEMBLER (WARM START)** — ulazak u GENS3M. Sam GENS3M je obogaćen za dve nove komande. Pritisak na CAPS/ENTER briše ekran i postavlja kursor u gornji levi deo ekrana. Pritisak na CAPS/L trenutno prekida rad programa. Ponovni pritisak na L nastavlja rad. Ovo je uvedeno zbog jednog vrlo grubog бага skriveneog u GENS-u. Naime, kada se asemblira program pod opcijom 1, 3, 5, 7, itd, na kraju listinga biva ođtampan spisak labela. Ako

```
10 GO TO USR VAL "31650"
100 RANDOMIZE USR VAL "33160"
RUN
200 LET A$="SURE(Y/N)"; GO SU
B VAL "2200"; IF A$="" THEN RA
NDOMIZE USR VAL "33185"
210 RUN
300 RANDOMIZE USR VAL "33259"
400 RANDOMIZE USR VAL "33112"
500 RANDOMIZE USR VAL "33134"
RUN
600 LET A$="NUM"; GO SUB VAL "
2200"; LET N=USR VAL "32328"; GO
TO VAL "600"
700 LET A$="(X)XL/ADDR"; GO S
UB VAL "2200"; LET X=A$ SGN PI "="
X"; IF X THEN LET A$=A$ VAL "2
" TO "; INVERSE SGN PI; GO TO VA
L "720"
710 IF A$="A" THEN INVERSE SGN
PI; LET A$=USR VAL "32806"; GO
TO VAL "730"-VAL "30"(R)KRV AL "1
6384"
720 LET A$=USR VAL "32337"; IF
NOT X OR NOT A THEN GO TO VAL
"700"
730 PRINT "INPUT DATA"; LET A
$=VAL "32372"; LET X=SGN PI; REST
ORE FOR I=SGN PI TO VAL "4"; R
ERE X$;A$; IF X THEN IF A$="" T
HEN INPUT (X$); LINE A$; IF A$=""
" THEN LET X=NOT PI
740 IF A$="" THEN LET A$="0"
IF I=2 THEN LET A$=STR$ A$
750 PRINT (X$); RANDOMIZE USR 3
2631 GO SUB 2400; LET A$=2; NE
XT I; PRINT (SGN PI); READY?"; PR
USE NOT PI; IF INKEY$="E" THEN
RUN
760 CLS; LET A$=USR VAL "32300"
; RANDOMIZE USR VAL "34470"; PAU
SE NOT PI; CLS
770 RANDOMIZE USR VAL "34521";
GO TO VAL "700"
780 DATA "A=", "B=", "C=", "D=",
" ", "HL=", ""
900 GO SUB VAL "2000"; PRINT #S
GN PI; Text; "; RANDOMIZE U
SR VAL "33632"; PRINT #SGN PI; N
ORSPACE 34252 #85CC; PAUSE NOT
PI; RANDOMIZE USR VAL "33259"
900 LET A$=Text$end (#); GO S
UB VAL "2200"; LET A$=VAL "34754"
; GO SUB VAL "2400"; RUN
1000 GO SUB VAL "2100"; PRINT "
LEN"; USR VAL "33695"; GO TO VAL
"1E3"
1100 GO SUB VAL "2000"; GO SUB V
AL "2100"; RANDOMIZE USR VAL "33
715"; GO TO VAL "1100"
1200 INPUT "A$SAVE SE"; "I$SAVE M
C$"; "2$SAVE CODE"; "3$LOAD CODE";
OP; LINE A$; IF A$="3" THEN G
O TO VAL "1200"
1210 IF A$="0" OR A$="2" THEN R
UN
1220 LET Y=A$="0"; IF Y THEN RA
NDOMIZE USR VAL "32878"
1230 LET X=A$="2"; PUKE VAL "330
04"; A$="1"; LET A$=VAL "34223"; R
ESTORE VAL "1290"; FOR I=SGN PI
TO VAL "4"; LET A=A+VAL "2"; REA
D A$;N; IF Y AND I<VAL "4" THEN
NEXT I
```

```
1240 IF X OR N=PI THEN GO SUB V
AL "2200"; LET N=USR VAL "33410"
1250 GO SUB VAL "2500"; NEXT I
1260 INPUT "NAME"; LINE A$; RA
NDOMIZE USR VAL "34070"; PRINT #S
GN PI; VERIFY(Y/N); PAUSE NOT
PI; CLS; IF INKEY$<>"Y" THEN G
O TO VAL "1200"
1270 LET A$=VAL "34229"; GO SUB V
AL "2300"; VERIFY "CODE N"; GO T
O VAL "1200"
1280 INPUT "LOAD ADDR"; LINE A$
; IF A$="" THEN LOAD "CODE"; G
O TO VAL "1200"
1290 LET A$=USR VAL "33410"; LOAD
"CODE A"; GO TO VAL "1200"; DAT
A "LEN"; USR VAL "32091"; "LOAD A
DDR"; PI; "MEMORY ADDR"; USR VAL
"33015"; "HOW MANY"; PI
1300 RANDOMIZE USR VAL "33744";
RUN
1400 GO SUB VAL "2100"; FOR N=U
SR VAL "33410" TO VAL "1E6"; RAN
DOMIZE USR 32411
1410 IF INKEY$="" THEN GO TO VA
L "1410"
1420 IF INKEY$="S" THEN PRINT
GO TO VAL "1400"
1430 IF INKEY$="E" THEN RUN
1440 NEXT N
1500 LET A$="(I)ADDR"; GO SUB V
AL "2200"; IF A$ SGN PI "=" I" T
HEN PRINT "1 BYTE"; LET A$=A$ VA
L "2" TO "; RANDOMIZE USR VAL "3
2846"; GO TO VAL "1500"
1510 RANDOMIZE USR VAL "32585";
GO TO VAL "1500"
1600 LET A$="(I)ADDR"; GO SUB V
AL "2200"; LET X=A$ SGN PI "=" I";
IF X THEN PRINT "1 BYTE"; LET
A$=A$ VAL "2" TO "; RANDOMIZE U
SR VAL "32846"; GO TO VAL "1520"
1610 RANDOMIZE USR VAL "32585"
1620 LET A$=USR VAL "33410"; LET
A$="NUM"; GO SUB VAL "2200"; LE
T N=USR VAL "33410"; IF X THEN
PUKE A;N; RANDOMIZE USR VAL "328
56"; GO TO VAL "1600"
1630 GO SUB VAL "2500"; RANDOMIZ
E USR VAL "32645"; GO SUB VAL "1
600"
1700 INPUT "INK"; "R"; "PAPER"; N
; INK A; PAPER N; BORDER N; CLS
; PUKE VAL "23624"; ATTR (PI,PI);
RUN
1900 STOP
2000 INPUT "0$SF"; "1$SF"; "0P";
"1" LINE A$; IF A$="" THEN RUN
2010 IF A$="0" THEN LET A$="435
72"; GO TO VAL "2000"
2020 IF A$="1" THEN RUN
2030 LET A$=VAL "34754"; GO SUB V
AL "2300"; LET A$=STR$ N
2040 RANDOMIZE USR VAL "33410";
RETURN
2100 LET A$="ADDR";
2200 INPUT (A$); LINE A$; IF A$=""
" THEN RUN
2210 RETURN
2300 LET N=PEEK A+256#PEEK (A+1)
; RETURN
2400 LET N=USR 33410
2500 PUKE A;N;256#INT (N/256); P
UKE A+1; INT (N/256); RETURN
9000 REM USR PROG
9010 RANDOMIZE USR VAL "50000";
PAUSE NOT PI; RUN
```





je taj spisak veći od dvadesetak linija, višak će jednostavno ispasti izvan ekrana, bez mogućnosti da se to spreči.

**ASSEMBLER (COLD START)** — ulazak u GENS3M, ali se pri tome budi izvorni kod. Da se ova opcija ne bi slučajno pozvala, program postavlja pitanje: „SURE?(Y/N)”. Treba odgovoriti pritiskom na Y, odnosno CAPS/Y ako računar ne radi u CAPS LOCK modu. Uzgred rečeno, CAPS LOCK mod se automatski postavlja prilikom učitavanja programa, kad i boja ekrana koju je korisnik izabrao kad je kucao program. Negativan odgovor, naravno, vodi u osnovni meni.

**DISASSEMBLER** — poziva MONS3M. **RESET ON** — uključuje RESET. RESET je automatski uključen po učitavanju programa, a to svedoči mala crtica u gornjem desnom uglu ekrana.

**RESET OFF** — isključuje RESET i tada nestaje pomenuta crtica. Dok RESET radi, pritiskom na CAPS/ENTER prekida izvršavanje svakog programa, bilo bežička, bilo mašinska, naravno pod uslovom da interapt nije isključen mašinskom naredbom DI. Kada se upotrebi RESET, na dru ekrana se ispisuje poruka „RESET EXECUTED” i adresa (dekadno i heksadekadno) prekida ili, preciznije, adresa prve naredbe koja bi bila izvršena da nije aktiviran RESET. GENS3M za vreme svog rada isključuje interapt, pa RESET ne radi. Tada pritiskom na CAPS/ENTER ima sasvim drugu, već opisanu funkciju. Ove dve stvari ne treba mešati.

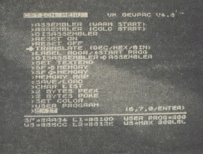
**TRANSLATE** — dovoljno je otkucati neki broj u dekadnom, heksadekadnom ili binarnom obliku (sa obavezanim predznakom #, odnosno % za hex. i bin. brojeve.) i na ekranu će se pojaviti taj isti broj u sva tri oblika. Ako je prvi karakter znak minus, onda se smatra da je broj negativan. Tada program postavlja pitanje: „8/16 BITS?” — što što, recimo, — 1 u osmobiitnom obliku daje 255, a u šesnaestobitnom 65535.

**LABEL ADDR/\*START PROG** — ovo je jedna od najvažnijih opcija u celom programu i postoje tri načina da se iskoristi.

Na pitanje „(\*) LBL/ADDR” otkucati naziv neke labele. Na ekranu će biti prikazana dekadna i heksadekadna adresa te labele. Ovo, naravno, podrazumeva da je pomoću GENS-a napisan i asembliran mašinski program koji sadrži tu labele. U protivnom, biće odštampan izveštaj: „NOT FOUND”.

Druga mogućnost je da se ispred naziva labele otkuca zvezdica. To znači da treba startovati mašinski program počev od navedene labele. Ipak, ne radi se samo o pukom pozivanju mašinske. Program omogućava da se postave početne vrednosti u A, BC, DE i HL registre procesora. Ovo omogućava da se mašinski program pozivaju sa različitim parametrima, što je vrlo korisno u fazi razvoja softvera. Ako nema potrebe da se postavljaju početne vrednosti, onda jednostavno pritisnuti ENTER. Sada se na dru ekranu pojavljuje fleširajuća poruka „READY?”. Pritiskom na neki taster i — startuje se mašinska. U desnom donjem uglu se pojavljuje mala kockica, znak da je mašinska obavio svoj posao. Još jedan pritiskom na bilo koji taster i na ekranu se ispisuje sadržaj BC registra pri povratku u bežik i koliko je dugo radio mašinska, u pedesetim delovima sekunde.

Treća mogućnost je, u suštini, isto što i druga, s tim što se umesto zvezdica labele,



## KUCANJE PROGRAMA

Da bi se formirao program, potrebno je posedovati programe GENS 3M i MONS 3M, koje „Galaksija”, normalno, nema prava da objavi niti bi imalo smisla kucati dodatnih 16 K mašinska. Prilikom formiranja programa treba vrlo strogo poštovati uputstva jer se može desiti da program ne radi kako treba:

- otkucati HEX LOADER i pomoću nje-ga mašinic koju sledi
- snimiti mašinska na traku komandama SAVE „MC 1” CODE 31650, 3050 i SAVE „MC 2” CODE 65000, 500
- PRINT USR 0
- otkucati bežik program (obavezno ceo program kucati velikim slovima)
- snimiti bežik program na traku komandom CLEAR: SAVE „BASIC”
- PRINT USR 0
- CLEAR 26500
- učitati GENS 3M komandom LOAD \*\*\*\* CODE 34700
- učitati MONS 3M komandom LOAD \*\*\*\* CODE 26501
- RANDOMIZE USR 34700; na pitanje „Buffer size” odgovoriti sa ENTER, zatim otkucati B i ENTER
- RANDOMIZE USR 26501; pritisnuti CAPS/ENTER
- učitati mašinska MC 1 i MC 2
- učitati bežik
- postaviti boje INK-a, PAPER-a i BORDER-a; te će boje biti automatski postavljene pri učitavanju programa.
- postaviti kasetofon u poziciju za snimanje i otkucati RANDOMISE USR 65000; time se na kasetu snima finalna verzija programa (pritisak na ENTER ponavlja snimak).

*Na kraju ostaje ono najprirodnije: učitati program (sa LOAD \*\*\*\* CODE) i isprobati sve opcije i mogućnosti. Za detaljnije informacije, svi zainteresovani programeri mogu da se jave autoru — preko redakcije.*

direktno kuka neka adresa, recimo 50000 ili 0C350, svejedno.

**DISASSEMBLER** → **ASSEMBLER** — ova opcija omogućava da se disasembliрани program prosledi GENS-u. Najpre treba odgovoriti na pitanje:

- 0) SF
- 1) SF+SF
- OP:

Nula znači da će disasembliрани program uništiti eventualni, već postojeći u GENS-u, a jedinica da će se spojiti iza onoga što već postoji u GENS-u. Zatim program daje vrednosti koje treba kucati kao odgovor na pitanja „Tekst” i „Work-space” koje postavlja MONS3M. Pritiskom

neki taster i sledi ulazak u MOS3M. Po izlasku iz ONS-a, program postavlja pitanje „Textend (?:)”. Treba otkucati broj koji je MONS naveo za Textend. Povisilica u zagradaima jednostavno poseđa na obavezni predznak za heksadekadne brojeve.

**SET TEXTEND** — postavlja vrednost TEXTEND-a za GENS3M.

**SF** → **MEMORY** — omogućava da se GENS-ov izvorni kod prekopira negde u memoriju. To znači da je moguće pisati sasvim nov program, a da se kasnije stari izvorni kod vrati GENS-u. Na pitanje „ADDR:” treba navesti adresu na koju će se izvorni kod kopirati. Posle toga program će odštampati koliko je izvorni kod zauzeo memorije.

**SF** → **MEMORY** — omogućava da se izvorni kod koji je prethodno bio prekopiran negde u memoriji vrati na korišćenje GENS-u. Program najpre postavlja pitanje 0) SF

1) SF+SF

koje je već ranije opisano. Zatim program pita na kojoj se adresi nalazi ranije prekopirani izvorni kod. Vrlo je važno navesti ispravnu adresu — u protivnom, program može da krahira.

**SAVE/LOAD** — program najpre postavlja pitanje:

- 0) SAVE SF
- 1) SAVE MC
- 2) SAVE CODE
- 3) LOAD CODE

Postoje, dakle, 4 mogućnosti. 0) Snima GENS-ov izvorni kod na kasetu. Na pitanje „HOW MANY?” treba navesti koliko puta treba ponoviti snimanje, a na pitanje „NAME”: treba navesti neko ime. Ovo poslednje pitanje program štampa inverzno da bi opomenu korisnika da snimanje počinje čim se pritisne ENTER. Prednost ove komande u odnosu na komandu P GENS-a je u tome što može da automatski snimi izvorni kod više puta, bez potrebe da posle svakog snimanja korisnik ponavlja naredbu.

1) Snima mašinski program počev od labele START pa do labele STOP. Mašinski program napisan pomoću GENS-a treba da ima, recimo, sledeći oblik:

```
ORG 50000
START LD BC, 1965
RET
STOP
```

pogrešno bi bilo napisati:

```
STOP RET
```

jer tada ne bi bio snimljen poslednji bajt. Ukoliko labele START i STOP nisu prisutne, ili mašinski program nije asembliran, onda će se program zauzvati sa porukom STOP STATEMENT, 1230:7. Ako je sve u redu, na ekranu će biti prikazane adrese labele START i STOP i dužina programa koji se snima. Program, zatim, postavlja pitanje „LOAD ADDR:”. Treba otkucati adresu na koju će se program učitati kada, kasnije, korisnik bude otkucao LOAD \*\*\*\* CODE. Sledeća dva pitanja („HOW MANY” i „NAME”) su već ranije objašnjena. 2) Omogućava da se neki deo memorije snimi na kasetu. Na pitanje „LEN” treba navesti dužinu bloka, na „LOAD LDDR:” na koju će se adresu program učitati (što smo već objasnili) a na „MEMORY ADDR” početnu adresu bloka. Slede još pitanja „HOW ANY” i „NAME”. Uzgred rečeno, na pitanje „NAME” može se samo pritisnuti ENTER, a poste snimanja (to važi i za opcije 0 i 1) pojavice se poruka

„VERIFY? (Y/N)“: 3) Vršiti učitavanje. Ako se na pitanje „LOAD ADDR“ pritisne ENTER, onda se izvršava naredba „LOAD“ „CODE“. Ako se navede neka adresa, onda se vrši naredba „LOAD“ „CODE ADRESA“.

**MEMORY MAP** — verovatno najefektnija opcija od svih. Omogućava da se, kao na dlanu, vidi koji su delovi memorije zauzeti, a koji slobodni. Program na ekranu crta grafički prikaz memorije, 256 bajta i biće vertikalna crta predstavlja 256 bajta i biće boje INK-a ako je bar jedan od tih bajtova različitih od nule. Ispod grafika je mala tačkica koja se može pomerati levo/desno pomoću tastera 5 i 8. Savim na dnu ekrana je adresa tastera. Ako se pritisne (i drži) taster X, onda će blok koji adresa pokrene tačkica biti „uvećan“ — svakom bajtu odgovara jedna vertikalna crta. Pritisak na ENTER vodi u osnovni meni.

**CHAR LIST** — ovo je, takođe, izuzetno korisna opcija. Omogućava da se „izlista“ neki blok memorije, pri čemu se za svaki bajt štampa njegova adresa i vrednost (sve, naravno, dekadno i heksadekadno) i ASCII kod koji predstavlja vrednost tog bajta, naravno, samo ako je vrednosti veća od 31. Treba prvo, na pitanje „ADDR“, navesti početnu adresu bloka koji se „lista“. Dalja uputstva nisu potrebna.

**2 BYTES PEEK** — daje sadržaj dva bajta memorije (po formuli prvi + 256 puta drugi). Ako se kao prvi karakter na pitanje „(!) ADDR:“ navede znak uzvika (na primer: 50000 ili !# C350), onda će biti ođštampan sadržaj samo jednog bajta.

**2 BYTES POKE** — stavlja neki broj u dva bajta memorije. Prvo, na pitanje „(!) ADDR:“, treba navesti adresu, a zatim na pitanje „NUM“ navesti vrednost koja se „pokuje“. Ako je prvi karakter u adresi znak uzvika, kao što je već opisano, onda se cela operacija odnosi samo na jedan bajt.

**SET COLOR** — postavlja boju slova i pozadine.

**USER PROGRAM** — startuje bežik program počev od linije 5000. Na to mesto programer može da stavi neki kratak bežik program koji mu je potreban prilikom razvoja nekog mašinskog programa. Pogled na listing bežik programa pokazuje da se na liniji 5010 već nalazi naredba RANDOMIZE USR 50000. Ako nema potrebe da se kontrolišu ulazni i izlazni parametri mašinskog programa, niti da se izmeri vreme, onda je mnogo lakše startovati mašinic na ovaj način, nego pomoću opcije LABEL ADDR/START PROG. Treba obratiti veliku pažnju na jednu sitnicu: ako ne postoji nikakav mašinski program na adresi 50000 i ako nije izbrisana linija 5010, onda poziv opcije USER PROGRAM može da dovede do one divne poruke: „ e 1982 Sinclair Research Ltd“.

Da bi se opisale sve mašinske rutine koje poziva bežik i sve skrivene mogućnosti jednog ovakvog programa, bilo bi potrebno pola ovog izdanja. Zato ćemo pažnju usmeriti samo na najvažnije detalje.

Program poseduje dva linka od po tri bajta. Prvi link se nalazi na adresi 33037. Ako se tu stavi jedna CALL naredba, recimo CALL 50000, onda će se potprogram na adresi 50000 pozivati pri svakom interaktu, dakle pedeset puta u sekundi. Pri tome treba poštovati sledeće uslove: program

Nastavak na str. 37

# velika video predstava

## Put u središte ROM-a (3)

*Priča o raspodeli memorije nekog kućnog računara kojoj smo posvetili prethodno poglavlje naše škole sistemskog softvera nije ni izbliza završena — ovoga puta govorimo o video memoriji. Pre toga ćemo naravno, ispuniti obećanje za kraja prethodnog napisa i posvetiti dužnu pažnju linkovima i vektorima operativnog sistema.*

### Linkovi i vektori

Iako o njima govorimo kao o dvema odvojenim stvarima, linkovi i vektori, u suštini, služe istoj svrsi — setu instrukcija nekih procesora (npr. Z80) više odgovara definisanje i korišćenje linkova, dok je kod drugih mikroprocesora (npr. 6502) korišćenje vektora racionalnije. Linkovi i vektori, jednostavno rečeno, omogućavaju programeru da „izade iz ROM-a“. Njihovu namenu ćemo pokušati da objasnimo na primeru nemaskiranog interupta kod Z80.

Posle izvršenja svake mašinske instrukcije, mikroprogram ugrađen u Z80 ispituje da li je na nožici NMI (Non-Maskable Interrupt), pri čemu koji se ne može zabraniti logički nisko stanje (0 V). Ukoliko nije mikroprocesor će nastaviti sa radom, ispitujući da li postoji zahtev za maskirani prekid (ako ovaj nije omogućen instrukcijom DI), a zatim preći na fazu pripreme sledeće naredbe. Ukoliko međutim, nemaskirani prekid postoji, biva proglašeno „vanredno stanje“. Mikroprocesor setuje poseban, spolja uglavnom nevidljiv flag koji označava stanje interupta, a zatim prenosi stanje brojača naredbi na stek i u brojač naredbi postavlja broj &66 (zašto baš ovaj broj? — na to pitanje mogu da odgovore jedino Zilogovi inženjeri) i počinje sa izvršavanjem programa koji se nalazi na toj adresi.

### Poziv preko linka

Pretpostavivši da naš računar ne koristi nemaskirani interupt za komunikaciju sa diskovima ili računarskom mrežom; poput „galaksije“, nemaskirani interupt će se koristiti samo za „oporavljanje računara“ u situacijama kada ga neki pogrešno napisan mašinski program zbuni. Počevši od &66, u ROM će, dakle, biti upisan program koji najpre reinitijalizuje sadržaj pokazivača steka (register SP), a zatim ponovnja egzistencijalno važne sistemske promenljive vrednostima koje će omogućiti normalan rad računara. Posle toga, kontrolu upravljanja treba predati „farmi“, tj. programu u ROM-u koji treba da ispiše READY i počne da skanira tastaturu da bi izvršavao korisničke naredbe.

Ako je stvar tako koncipirana (a zaista je tako koncipirana na „galaksiji“), korisnik nema nikakve mogućnosti da promeni NMI rutinu. Zamislimo da smo kupili interupta i prolog (ako čujete da se negde prodaje, obavezno nam javite!) i da smo ga startovali. U toku pisanja programa na prologu napravili smo kružnu definiciju i računar se

„zaglavio“. Pritiskamo RESET i na ekranu vidimo bežikovo READY. Prolog je prekinut da više nikada ne bude nastavljen a program koji smo pisali zauvek je izgubljen. Mora li da bude baš tako?

Zamislimo da je na &66 najpre inicijalizovan SP (zašto toliko insistiramo na tome da je neophodno inicijalizovati pokazivač steka pre nego što se radi bilo šta drugo? Nagradićemo prvi potpun odgovor!), a zatim izvršeno CALL &2C30, na &2C30 se, na primer, nalaze bajtovi &C9 00 00 pri čemu je &C9 kod mašinske naredbe RET. Mikroprocesor će, dakle, najpre izvršiti CALL &2C30, naći na RET i vratiti se u ROM. Besmislena šetnja? Šta, međutim, ako autor PROLOG interpretatora počevši od &2C30 u RAM upiše &C3 00 &30 ili, ako više volite mnemoniku, JP &3000? Uместo da se kontrola vrati ROM-u, počevši da se izvršava korisnikov program koji će, ako je to ikako moguće, „oporaviti“ prolog i sačuvati naš program, posle čega će, umesto „farme bežikova“, biti izvršena „farma prologa“. Jednostavno smo, dakle, postigli nešto što bi se na sadašnjoj „galaksiji“ postizalo samo uz mnogo programerskih trikova, ako bi uopšte i moglo da se postigne (ako je NMI link tako zgodna stvar, zašto ga „galaksija“ nema? Jednostavno, mesta za njega u ROM-u od 4 K nije moglo da bude).

NMI link je jednostavan kako za objašnjenje tako i za upotrebu. Drugi linkovi mogu da se koriste daleko komplikovanije, ali je princip na kome se zasnivaju potpuno isti: u ROM-u piše CALL LINK, gde je LINK adresa lokacije u RAM-u koju u toku inicijalizacije ponovljavamo sa &C9, pri čemu su memorijske ćelije LINK+1 i LINK+2 slobodne (npr. popunjene nulama). Ukoliko korisnik nije menjao sadržaj linka, kontrola će se vratiti ROM-u: ukoliko ga je promenio u JP SERVICE, biće izvršen njegov program SERVICE, posle čega se kontrola može (ali ne mora!) vratiti rutini iz ROM-a koja je pozvala link. Ukoliko odlučimo da se iz našeg potprograma ne vraćamo u ROM, treba da izvršimo jedno POP AF i tako eliminišemo adresu za povratka koju je generalisa instrukcija CALL.

Gde se treba postaviti linkove? Kao najnužniji minimum, moraju da postoje linkovi u maskiranom i nemaskiranom interuptu, link u rutini za greške, link preko koga prelazi sve što će se ispisati na ekranu, i link preko koga prelazi potprogram za skaniranje tastature (kako bi povezivanje sa nekom drugom tastaturom i prikjučivanje raznih džojstika bilo što jednostavnije). U „bogatijim“ ROM-ovima linkova će, jasno, biti daleko više, dok će u siroma-

šnjim slučajevima postojati samo jedan ili dva. Svaki link viška u neverovatnoj meri povećava fleksibilnost vašeg računara i spasava svakog ambicioznijeg korisnika od preturanja po ROM-u i njegovog prepisivanja u EPROM-e.

Nije, međutim, dovoljno predvideti linkove; programi iz ROM-a treba da prenesu programima u RAM-u sve podatke relevantne za njihovo korektno funkcionisanje. Posmatrajmo, na primer, link u rutini za obradu grešaka. Ovaj link će najverovatnije biti korišćen za dodavanje novih naredbi: detektujemo situaciju u kojoj je prijavljena greška, proverimo da li je naredba koja je izazvala grešku neka od naredbi koje smo dodali i izvršimo je ako smo je prepoznali. Jednostavno? Jeste ako znamo gde se nalazi naredba koja nije prepoznata i koja je greška nastupila (ako je, na primer, nastupila greška 'Negative root' nema mnogo smisla u traženju naše nove naredbe). Potrebno je, dakle, predvideti i dokumentovati lokacije u radnom prostoru operativnog sistema ili (još bolje) sistemske promenljive koje će sadržati podatke koji će eventualno zahtevati rutinu koja je izvršena posle skoka na link.

### Putokazi po RAM-u

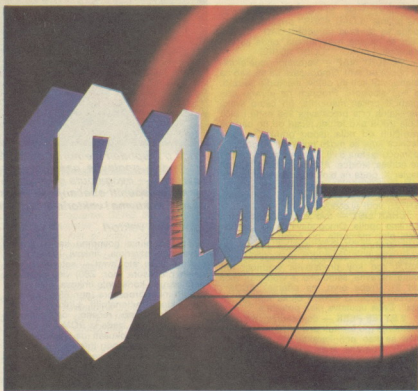
Mikroprocesor Z80 poseduje naredbu JP (HL) koja izaziva skok na naredbu čija je adresa u HL ali ne i naredbu JP (&2000) pri čemu bi adresa na koju treba skočiti bila upisana u memorijske ćelije &2000 i &2001; &502, nasuprot tome, nema ništa slično instrukciji JP (HL) (jer nema šesnaestobitnih registara), ali mu zato naredba JMP (&2000) ne prčinjava nikakve probleme (osim ponekad — jeste li čuli za to da i mikroprocesori imaju bagove; pokušajte JMP (&20FF)). Zato ćemo umesto serije naredbi &C9 00 00 &C9 00 00... koje predstavljaju linkove, u neku seriju memorijskih ćelija RAM-a upisati adrese rutina u ROM-u na primer:

```
&20A : 72 E7 I BYTEV
&27 : EB E7 I WORDV
&20E : A4 E0 I WRCHV
&210 : CE DE I RDCHV
```

Ovde je &E772 ulazna adresa rutina u ROM-u koji nosi ime BYTE (šta god ta rutina radi), &E7EB adresa rutine OSWORD itd. Na nekom fiksnom mestu u ROM-u (na primer pri njegovom kraju) će se nalaziti ovakva sekvencna.

```
&FFF4 .. OSBYTE JMP (&20A)
&FFF1 .. OSWORD JMP (&20C)
&FFEE .. OSWRCH JMP (&20E)
&FFF0 .. OSRDCH JMP (&210)
```

Kada god se u ROM-u pojavi potreba za rutinom OSRDCH, nećemo koristiti JMP &DECE nego JMP &FFE0. Na taj smo način dobili dve stvari: pre svega, korisnik može da promeni sadržaj RDCH vektora i tako „skrene“ rutinu iz ROM-a i potpuno je ili delimično zameni svojim programom. Drugi je dodatak u tome što možemo da proizvodimo nove verzije operativnog sistema u kojima će stvarna rutina RDCH biti na sasvim drugom mestu a da svi programi koji su propisno pisani (koji su, dakle, koristili JMP &FFE0 umesto JMP &DECE) i



dalje besprekorno funkcionišu. To, ipak ne znači da treba neograničeno često da menjamo jedan komercijalno raspoloživ ROM!

Kao i svaka stvar na svetu, i linkovi i vektori imaju loših strana. Pretpostavimo da, u prvom primeru, mašinski program koji je izazvao krah „izbombarduje“ memoriju i, pored ostalog, uništi i sadržaj NMI vektora. Pritisak na RESET nam tada neće mnogo pomoći — da linka nema računari bi se verovatno oporavio, dok je ovako nepovratno izgubljen. Rešenje toga bi bilo da NMI rutina najpre reinicijalizuje SP (nagrada čekal), zatim proveri da li je SPACE pritisnut, pa tek ako nije skoči na link. U protivnom skok na link biva ignorisan da bi se izvršila NMI rutina iz ROM-a koja, pored ostalog, inicijalizuje sve linkove koji su mogli da stradaju.

### Vide memorija

Pretposlednja (a, kao što ćemo videti, kod nekih računara i poslednja) stavka memorije koja „pripada“ operativnom sistemu je segment RAM-a u koji je ucrтана slika koju treba iscrtavati ne ekranu. Dok je raspodela zone sistemskih promenljivih uglavnom stvar autora ROM-a, lokacija i organizacija vide memorije je diktirana hardverom računara pa ćemo, da bismo je razumeli, morati malo da se zabavimo konkretnim rešenjima koje su primenili konstruktori nekih popularnih kućnih kompjutera.

Zu iscrtavanje slike na ekranu standardnog televizora ili monitora potrebna je savršena koordinacija hardvera i softvera računara; kod nekih (mahom jeftinijih) kompjutera, kao što su ZX81 i „galaksija“, u ovoj aktivnosti dominira softver, a kod drugih (npr. „amstrad“, „komodor“ i BBC) uloga hardvera je daleko veća. Ne treba biti

zavaran terminima „hardversko“ i „softversko“ rešenje vide — generisanje slike je proces toliko složen da bi hardver vrlo teško mogao da ga obavi bez softvera. Međutim, neki računari su opremljeni specijalizovanim mikroprocesorima koji su zaduženi isključivo za generisanje slike i koje zato nazivamo video kontrolerima. U te čipove su konstruktori upisali manje ili više kompleksne mikroprograme za obavljanje pojedinih mikrooperacija i za komunikaciju sa glavnim procesorom, tako da se, sa aspekta autora operativnog sistema, čitavo generisanje slike obavlja automatski.

### Dva jarca na brvnu

U poslednje vreme se video memorija sve češće odvaja u poseban blok koji ne mora da se nalazi čak ni u istoj memorijskoj mapi sa RAM-om računara. U čemu je smisao ovakvog komplikovanja? Treba, pre svega, da raščistimo pojmove memorijskih mapa. Svaki mikroprocesor može da adresira određeni broj memorijskih lokacija; ovaj broj je precizno određen dimenzijama njegovog adresnog basa. Osobitni mikroprocesori redovno imaju šesnaestobitni adresni bas (ili osobitni sa multipleksiranjem adresa), što znači da mogu da adresiraju  $2^{16} = 65536$  različitih osobitnih memorijskih lokacija. U tih 65536 lokacija konstruktor računara mora da smesti čitav ROM, čitav RAM i sve memorijski mapirane ulazno-izlazne jedinice.

Obzirom na sve veće zahteve koji se postavljaju pred moderne kućne računare, ovaj broj nije naročito velik, pa ga ponekad proširujemo nekim od metoda tzv. „pejzžovanja“, tj. paralelnog vezivanja više blokova RAM-a od kojih mikroprocesor u datom trenutku bira jedan. Video memorija, međutim, mora da bude pristupačna u svakom trenutku, jer se njen sadržaj ispisuje na



ekranu dvadesetak puta u sekundi. Segment video memorije, sa druge strane, može da bude prilično veliki: ako naš računar ima grafičku rezoluciju 256\*192, pri čemu svaka tačka može da bude „ofarbana“ jednom od osam boja, video memorija će zahtevati 256\*192\*3/8/1024=24 kilobajta ili gotovo 40% raspoloživog RAM-a; ako želimo veću rezoluciju ili više boja, ova brojka bi se drastično povećala. Da bi se RAM oslobodio za korisničke programe, moguće je video memoriju odvojiti u posebnu memorijsku mapu koja, dakle, nije pristupačna mikroprocesoru.

Logično rešenje bi bilo omogućiti video kontroleru da adresira svojih 32 (ili 64) kilobajta memorije i da sliku koja je upisana u nju u pravilnim intervalima iscrtava na ekranu. Jasno je, međutim, da mikroprocesor mora na neki način da opšti sa ovom memorijom, jer je on taj koji treba da odredi šta će se iscrtati na ekranu. Recimo, za sada, da se ta komunikacija obavlja posredstvom neke OUT instrukcije: ukoliko mikroprocesor želi da oboji tačku čije su koordinate (200, 100) crvenom bojom čiji je kod 010 odnosno 2 dekadno, pošalje na neki od svojih izlaznih portova redom bajtove 200, 100 i 2. Ove brojke će primiti video kontroler i upisati ih u video memoriju koja mu je pristupačna. Potrebno je ostvariti i komunikaciju u obrnutom smeru, jer mikroprocesoru može da zatreba da ispita boju nekog piksela, za šta će se koristiti neka IN mašinska naredba.

Komunikacije između mikroprocesora i video kontrolera se izvršavaju na principu takozvanog „rukovanja“ (handshaking): kada od video kontroler obradi prethodnu naredbu i postane spreman da primi novu, biće setovan neki flip-flop, koji mikroprocesor može da ispituje IN instrukcijom. Ukoliko mikroprocesor primeti da je flip-flop resetovan, slanje naredbe će biti odozno, a mikroprocesor će se vrteti u petlji čekajući da video kontroler bude spreman. Nevolja je baš u ovom čekanju — umesto da radi redovan posao, mikroprocesor gubi vreme zbog sprostosti periferije. Problemi takvog tipa se rešavaju uvođenjem bafera raznih kapaciteta i generisanjem interrapta kojim će mikroprocesoru biti stavljen do znanja da je posao obavljen, ali je korist od ovakvih rešenja umerena pri čemu ona povećavaju cenu računara. Pošto se kućni kompjuteri često planiraju tako da budu pogodni za igre, brzina je imperativ od koga se ne sme tek tako odustajati, pa konstruktori računara, u očekivanju šesnaestostnih mikroprocesora koji će imati daleko prostornije memorijske mape, pribegavaju alternativnim rešenjima: generatorima karaktera, grafičkim modovima ili takozvanim atributima.

Princip atributa ćemo objasniti na primeru popularnog „spectruma“. Kod njega je slika praktično crno-bela: svakoj tački na ekranu odgovara po jedan bit memorije koji je '1' ukoliko je odgovarajuća tačka u boji prednjeg plana („foreground colour“) a '0' ukoliko je tačka u boji pozadine („background colour“). Kako su onda ostvarene boje? Ekran je podeljen na slovna mesta dodeljen je atribut u vidu jednog bajta. U taj bajt su upisane boje prednjeg i zadnjeg plana koje važe za sve tačke tog znaka i, uz njih, par drugih informacija (na primer, da li

karakter blinkuje ili ne). To znači da sve tačke jednog slova moraju da budu obojene istom bojom, što je sasvim logično kada te tačke zaista pripadaju nekom slovu (nikome neće pasti na pamet da piše slovo A čija je leva polovina crvena, a desna crna, ali i vrlo nepotrebno kada na ekranu treba nacrtati neku sliku. Slika koju crtamo mora da bude logički podeljena na delove veličine slova koji moraju da budu obojeni istom bojom. Autori brojnih izvanrednih igara na „spektrumu“ su pokazali da je takva podela moguća ali ni jedan od njih nije rekao da je je princip atributa oduševio! Postigavši vrhunac svoje popularnosti sa „spektrummom“, atributi se polako sele u istoriju.

### Generator znakova

Generator znakova je bilo nekada veoma popularno rešenje video memorije, kome se, u manje ili više modifikovanoj verziji, često pribegava i danas. Da bi nam ovakvo rešenje bilo jasno, ponovo ćemo se poslužiti jednim primerom, ovoga puta primerom domaćeg računara „galaksija“. Pošto slova, cifra i specijalnih znakova ima relativno malo (jedva stotinak, čak i kada se uzmu u obzir mala slova), glupo je svako od njih iscrtavati na ekranu tačku po tačku. Mnogo je ekonomičnije rezervisati potreban ROM u koji će biti upisani oblici svih simbola koje neki računar može da iscrtava na ekranu, a zatim za video memoriju treba da izdvojimo 32\*16=512 bajta ili pola kilobajta. Ukoliko se, na primer, u gornjem levom uglu ekrana nalazi slovo 'A' čiji je ASCII kod 65, u prvoj memorijskoj lokaciji video memorije (njena je adresa &2800) će se nalaziti broj 65. Kako se ovaj broj pretvara u oblik slova A u trenutku generisanja slike? U generator karaktera je, bit po bit, upisan oblik slova A pa će video stепен, kada ga mikroprocesor pobudi, ispisati redove tačkica prema precizno određenoj šemi.

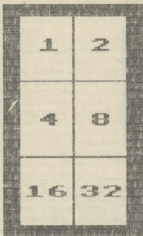
Osnovni nedostatak koncepcije sa generatorom karaktera je u tome što se tekst koji se ispisuje na ekranu može sastojati jedino od znakova koji su predvideli konstruktori računara. Znamo da se kućni računari mnogo koriste za igre, a igre bez grafike i nisu neke igre. Činjenica je, međutim, da računari sa generatorom karaktera (npr. TRS80 i „galaksija“) ipak imaju grafičku određene (niske) rezolucije koja je lepo primenljiva za igre. Kako je to postignuto?

bit	1	2	3	4	5	6	7	8
0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0
A	0	0	0	0	0	0	0	0
B	0	0	0	0	0	0	0	0
C	0	0	0	0	0	0	0	0
D	0	0	0	0	0	0	0	0
E	0	0	0	0	0	0	0	0
F	0	0	0	0	0	0	0	0

U „galaksiji“ generator znakova su upisana i 64 takozvana grafička karaktera od kojih su neki prikazani na prvoj slici. Vidimo da se svaki od njih sastoji od delića veličine šestine jednog karaktera; neki od tih delića („piksela“) su osvetljeni, a ostali su zatamnjeni. Bežik programeri nerado koriste ove karaktere i rado bi ih zamениli za mala slova ili dodatne specijalne znakove; bilo je čak i onih koji su nam pisali sa pitanjem „zašto su grafički karakteri uopšte upisani u ROM?“. Dobra prilika da im odgovorimo — bez grafičkih karaktera „galaksija“ ne bi imala naredbe DOT i UNDOT niti bilo kakvu grafiku!

Sa slike 1 smo videli da se svaki grafički karakter sastoji od dva bloka po horizontali i 3 po vertikali. Pomnoživši broj blokova po horizontali (3) sa brojem slova u redu (32) dobijamo 64; pomnoživši broj blokova po vertikali (3) sa brojem redova (16) dobijamo 48. Ako se setimo da je „galaksija“ grafička rezolucija 64x48, rad naredbi DOT i UNDOT bi trebalo da nam postane jasan; ako nije, objasnimo ga na primeru naredbe DOT 20, 30. Softver u ROM-u treba najpre da pronade karakter u koji je skrivena tačka sa koordinatama (20, 41). To je karakter čija je X koordinata INT (20/2)=10 a Y koordinata INT (41/3)=13 — sve u svemu, radi se od jedanaestom slovu u cetрнаestom redu (numeracija, obratimo pažnju, počinje od nule a ne od jedan) tj. o slovu koje bi bilo ispisano da smo upotrebili naredbu PRINT AT (13 \* 32 + 10) ili naredbu BYTE 32800 + 13 \* 32 + 10, XX.

Pošto je softver koji realizuje naredbu DOT utvrdio o kojem se znaku radi, treba da se opredeli za tačku tog znaka koju će da setuje. Tu je račun nešto komplikovaniji — treba naći ostatak pri deljenju broja 20 sa 2 i broja 41 sa 3; tu su ostaci respektivno 0 i 2. Sa slike 2 vidimo da se radi o tački u desnom donjem uglu pronađenog karaktera. Ovu, dakle, tačku treba setovati.

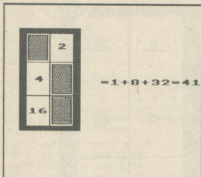


Tačka se, naravno, setuje tako što se odgovarajući bit video memorije postavi na jedinicu. Kako naš mikroprocesor može da adresira samo čitave bajtove, moraćemo da pozovemo prethodni sadržaj video memorije, iskoristimo naredbu OR da setujemo odgovarajući bit (da treba da ga resetujemo koristili bismo naredbu AND a da hoćemo da ga testiramo takođe naredbu AND pri čemu rezultat ne bi ponovo upisivali u memoriju) i promenjenu vrednost upisati na adresu &2800+13\*32+10. Vidimo (1) na-

redba DOT nema mnogo šanse da bude šampion u brzini — da bi se izvršila potrebno je nekoliko pristupa memoriji i dosta računanja. To je, međutim, cena koja je morala da se plati da bi se pomirile koncepcije postojanja grafike i postojanja generatora karaktera. Ukoliko vam je stalo da proverite svoje razumevanje te koncepcije, pokušajte da izračunate zašto ROM sa generatorom karaktera računara „galaksija“ ima baš 2 K i da li bi se u njega moglo upisati više znakova nego što je u ovom trenutku upisano.

### Crteži iz ROM-a

Postoji još jedna koncepcija rada sa generatorom karaktera koju ćemo opisati da biste znali kako ne treba raditi — koncepcija uvođenja grafike putem definisanja karaktera. Generator karaktera koji smo opisali u prethodnim pasusima može ali ne mora da se nalazi u osnovnoj memoriji mapy — proizvode ga video stepen u toku crtanja slike. Ako se, međutim, odlučimo da smestimo generator karaktera u memorijsku mapu, možemo da ga stavimo i u RAM u koga će se prepisivati posle inicijalizacije računara. U cilju štednje RAM-a možemo da se odlučimo da podelimo karaktere u grupe od po 16 pa da se za svaku grupu uvede flek koji će određivati da li su opisi njihovih karaktera u ROM-u ili u RAM-u. Bilo kako bilo, bezik programer može, koristeći usluge operativnog sistema, definisati svoje karaktere — ukoliko mu je potrebno grčko slovo ksi, lako će netu nepotrebnu srednju zagradu u našem setu karaktera zameniti njim. Da stvar bude još lepša, korisnik će moći da definiše naša latinčična slova i njima direktno zameni neke znakove sa tastature. Do sada je sve lepo. Šta, međutim, ako korisnik poželi da crta uz bliženje računara čiji su konstruktori prihvatili koncepciju generatora karaktera u RAM-u?



Jedina mogućnost za crtanje je da se najpre na nekom planeru ekrana pripremi slika, zatim se taj planer ekrana izdeli na delove veličine karaktera i preračuna kako se svaki od njih pretvara u računaru prihvatljiv niz brojeva (slika 3). Takve nizove brojeva treba uneti u memoriju i dobiti sliku. Ako pretpostavimo da bi ovakvo kodiranje i moglo da se izvrši (svakako ne ručno već primenom nekog programa), moramo da priznamo da bi sličan program mogao da se bavi i animacijom

— proračunao bi kretanje neke tačke, izmerno karakter u kome se ona nalazila na početku kretanja a zatim i onaj do koga je stigla. Ovde, međutim, ima dosta računara, pa bi akciona igra bila prilično spor. Da stvar bude još gora, utrošak memorije kod rada sa visokom rezolucijom drastično raste — za svako slovo na ekranu treba utrošiti dodatni prostor tako da bi se, za slučaj da je ceo ekran prekriven crtežom, upotrebljena memorija izjednačila sa potrebnom memorijom za iscrtavanje ekrana „bit po bit“. Do ovoga izjednačenja obično neke doći je su delovi ekrana pokriveni istim oblicima (tlo, nebo, prazan prostor itd) ali mogu da se pojave problemi ako se definicije karaktera „sudare“ sa bezjivim ili sa zonom numeričkih promenljivih.

Za eventualne početnike koji čitaju ovaj tekst reći ćemo da nije moguće najpre definisati karakter, ispisati x ga negde i onda ga ponovo predefinirati da bi bio ispisan na nekom drugom mestu — u video memoriju se upisujujedino ASCII kod slova koje treba ispisati na nekom mestu, dok se oblik tog slova nalazi u ROM-u ili RAM-u. Pošto se slika iscrtava mnogo puta u sekundi, promena jednog definisanog karaktera će trenutno promeniti sliku na svim mestima na kojima se taj karakter nalazi. Sve u svemu, koncepcija generatora karaktera u RAM-u može da bude pogodna jedino za terminale i računare koji se neće koristiti za crtanje i za igre; preostajima je ona dobar razlog za komercijalnu katastrofu koja je, na primer, zadela T199/4A i pored mnogih dobrih osobina.

### Grafički modovi

Došli smo, najzad, do koncepcije koja danas prevladava kod kućnih računara iz srednje klase — koncepcije grafičkih modova. O čemu se radi? Konstruktori računara su sasvim ispravno zaključili da se njihova remek dela koriste za više raznih stvari, ali da u jednom trenutku mogu da rade samo jednu od njih. Ukoliko neko koristi tekst procesor, treba mu 80 slova u redu, ali mu je grafika visoke rezolucije, mnogo boja i brz rad računara. Ukoliko neko koristi kompjuter za kreiranje crteža koji će dočnije biti dampovani, treba mu malo memorije i dve boje, ali i grafika veoma visoke rezolucije. Umesto da od svakoga zahtevamo da kupi računarski specijalizovan za ono čime se bavi, konstruisaćemo univerzalni računarski koji će, jednostavnim naredbom vršiti preraspodelu broja boja, grafičke rezolucije i zauzeća RAM-a.

Kod planiranja grafičkih modova treba se odlučiti za jednu od dve koncepcije i, kao posledica te odluke, odabrati neki od raspoloživih video kontrolera. Možemo da se odlučimo da video memorija zauzima konstantan memorijski prostor od, na primer, 16 kilobajta, a da korisnik pravi balans između boja, grafičke rezolucije i broja slova u redu (ovakva je koncepcija zastupljena kod danas sve popularnijeg „amstrada“), ili da u igru ubacimo i zauzeća RAM-a tako da različiti modovi zauzimaju više ili manje memorije (koncepcija BBC-ja i „elektrona“). Svaka od ovih koncepcija ima prednosti i nedostatke prva je, globalno posmatrano, jednostavnija za realizaciju, ali je druga potpunija. Osim toga, druga koncepcija omogućava postojanje veoma visokih rezolucija koje će se retko koristiti, dok prva stimuliše umerenije prelaze. Prva kon-

cepcija, međutim, ima jednu „psihološku“ prednost — stimuliše programera da izabere grafičke karakteristike prema potrebama programa a ne da piše program a zatim bira rezoluciju prema preostaloj memoriji.

Pošto smo izabrali video kontroler, dizajnera ćemo modove prema njegovim karakteristikama. Neophodno je da kreiramo jedan mod u kome će biti moguće ispisivati jedino tekst i to sa 80 znakova u redu, jedan mod sa relativno malom rezolucijom ali sa mnogo boja, mod sa visokom rezolucijom i četiri boje, te, na kraju, mod sa dve boje i veoma visokom rezolucijom. Za one koji se bave poslovnom obradom podataka možemo da pripreмимо i mod sa minimalnom rezolucijom (ili čak bez rezolucije) sa svega četrdesetak slova u redu koji će zauzimati najmanji mogući RAM — konstruktori BBC-ja su otišli tokom daleko da su uveli „mod 7“ u kome se koristi generator karaktera tako da ekran zauzima samo 1 kilobajt ali ovakve egzibicije komplikuju hardver računara i, samim tim, značajno povećavaju njegovu cenu.

Došlo je vreme da se pozabavimo pakovanjem grafike u video memoriju. Posmatramo najpre najjednostavniji slučaj u kome postoje samo dve boje — tačke su osvetljenje ili zatamnjenje. Za definisanje svake tačke potreban nam je po jedan bit, što znači da je za rezoluciju 256\*192 potrebno (256\*192)/8=6144 bajta ili tačno 6 kilobajta. Prirodno bi bilo očekivati da su tačke „logično poredane“: prvi bit video memorije odgovara tački u gornjem levom uglu ekrana, drugi bit tački do nje i tako dalje, red po red. Ovakvo, međutim, ništa ne mora da bude: vlasnici „spektruma“ i čitaoci „Računara“ verovatno znaju da je na ovom popularnom kompjuteru grafika realizovana u najmanju ruku čudno: najpre se u memoriju smesta prvi red tačaka a zatim osmi! Hir konstruktora? Naravno da nije — ovakva realizacija je posledica činjenice da je video memorija smeštena u dinamički RAM kome je potrebno stalno osvežavanje.

### Elektronska paleta

Posmatramo, na primer, mod sa tri boje. Za svaku tačku su nam sada potrebna dva bita, pri čemu će, na primer, kombinacija 00 označavati jednu, kombinacija 01 drugu a kombinacija 10 treću boju. Vidimo da nam je kombinacija bitova 11 ostala neupotrebljena, pa možemo da uvedemo i četvrtu boju koja neće zauzeti nikakvu novu memoriju. Iz ovoga možemo da zaključimo da nema nikakvog smisla projektovati mod sa bilo kojim brojem boja — njih mora uvek da bude 2<sup>n</sup> kako bi se do kraja iskoristilo n bita koji pripadaju nekoj tački. Povećanje broja boja, međutim, drastično povećava zauzeće memorije — rezolucija 640\*256 kod koje je svaka tačka „obojena“ jednom od 16 nijansi zahteva „samo“ 640\*256\*4/8=81920 bajtova ili 80 kilobajta, što teško mogu da podnesu čak i šesnaestobitni kućni računari ilie umerenih cena. Zato nemojte verovati specifikacijama računara u kome se pominju visoke rezolucije, mnogo boja i umereno visoke cene — gotovo je sigurno da su u svakom modu pristupačne samo neke od ovih grafičkih karakteristika. Šta, međutim, da kažete kada učitate neki računarski koji nudi 256 boja? Za umerenu rezoluciju od 256\*192, pri čemu svaka tačka može da bude obojena jednom od 256 boja treba odvojiti 256\*192\*8/4=96 kilobajta a ima računara koji nude 256 boja i svega 32 kilobajta

• 1) put u središte rom-a (3)

RAM-al! To je obično omogućeno primenom takozvane palete, ali izuzetno dobri efekti mogu da se dobiju i primenom raznih varki zasnovanih na interaktu.

Pozabavićemo se najpre paletom posmatrajći ekran na kome je svaka tačka obojena jednom od dve boje. Koje su to boje? Bez većeg razmišljanja smo skloni da kažemo: „crna i bela“. Nema, međutim, nikakvih prepreka da te dve boje budu, na primer, zelena, žuta, plava, magenta, braon i bela. Ove boje se zovu fizičke, jer se radi o bojama koje se zaista prikazuju na ekranu i obeležene su brojevima 0—7. Logičkih boja obično ima manje; u dvobojnom modu, na primer, postoje samo dve logičke boje koje su obeležene brojevima 0 i 1. U normalnoj situaciji logičkoj boji 0 je dodeljena fizička boja 7 (crna) a logičkoj boji 1 fizička boja 0 (bela). Primenom odgovarajuće naredbe koju autori ROM-a moraju da predvide i koju će primati bežik i izvršavati operativni sistem, korisnik može da definiše logičku boju 0 kao fizičku boju 2 (zelenu), što znači da će u daljem radu računar ispisivati bela slova na zelenoj pozadini (zavisno od kvaliteta kolor televizora ili monitora ovo može da bude vrlo prijatna kombinacija za oči). Što je najlepše, zamena logičke boje se događa praktično trenutno, u jednom jedinom frejmu, što pruža velike potencijale za animaciju. Kako se ovakva animacija izvodi? Koristimo neki višebojni mod, na primer mod u kome postoje četiri boje. Na početku rada definišemo da su tri od ove četiri boje (npr. 0, 1 i 2) jednake; to će biti boje zadnjeg plana. Preostalom bojom čiji je logički broj 3 nacrtamo statičan deo slike. Zatim, koristeći boju 0, nacrtamo jednu fazu kretanja objekta a zatim promenimo boju 0 u boju prednjeg plana. Počinjemo da crtamo u boji 1, koja je i dalje jednaka boji pozadine, tako da se crtež ne vidi. Kada završimo sa crtanjem, boju 1 proglašavamo za boju prednjeg plana a boju 0 za boju zadnjeg posle čega prvobitni položaj objekta postaje nevidljiv a novi vidljiv — dobili smo, dakle, brzu animaciju koja ne zahteva da se crtež priprema u odvojenom segmentu memorije a zatim, instrukcijom tipa LDIR, zamenjuje sa sadržajem video memorije. Ukoliko grafički mod koji koristimo ima više boja, efekte ovoga tipa možemo da eksploatišemo daleko efektivnije.

U poslednje vreme hit kompjuterskih časopisa su rutine koje „mešaju modove“ i omogućavaju ogroman broj boja uz vrlo umeren utrošak video memorije. Sve ove rutine se zasnivaju na pažljivoj upotrebi interakta pomoću koga se postiže da neke tačke menjiveto menjaju boju. Ukoliko se boja neke tačke naizmenično menja (npr. u svakom frejmu druga boja) naše oko dobija raznorazne utiske, neretko utiske postojanja novih boja. Ukoliko naš računar radi sa 16 osnovnih boja, može se zamisliti 16<sup>15/2</sup>=120 kombinacija boja koje će se međusobno menjati, dobar deo ovih kombinacija predstavlja nove boje. Znajući da se zgodni efekti postižu naizmeničnom izmenom tri ili čak četiri boje tačka, podnaslovi „1024 boje u našem „komodoru““ postaju sasvim razumljivi. Ovakvi efekti su počeli da se koriste u komercijalnim igrama (čuve-na Elita za BBC mikroracunara i „electron“ koristi ekran podeljen u dva dela od kojih je

svaki u različitom modu), ali nam nije poznat nijedan kućni računar koji ih koristi za normalan rad. Razloge za ovo verovatno treba tražiti u izuzetnoj osetljivosti slike dobijene na ovaj način na bilo kakve „zastoje u interaktu“ koji su, na primer, nastali zbog intenzivnog računarevog oštećenja sa diskom.

### Kontakti sa periferijom

Došli smo i do poslednje stavke memorije koja pripada operativnom sistemu — memorijski mapiranim periferijskim uređajima. Već znamo da mikroprocesor mora da koordinira rad video kontrolera, disk kontrolera, integralca koji kontroliše tastaturu, generatora zvuka i mnogih drugih manje ili više „inteligentnih“ čipova. Nekim od ovih čipova mikroprocesor treba samo da šalje podatke, od nekih treba samo da ih prima, a sa nekima treba da kontaktira u oba smera. Ta komunikacija se obavlja preko IN i OUT instrukcija koje smo već pominjali ili preko memorijske mape.

IN i OUT instrukcije nije teško razumeti: Z80, na primer, ima 256 I/O portova (uz malo dodatnog hardvera ovaj broj može da se poveća na 65536 ali je ovakvo povećanje vrlo retko zaista i potrebno). Ukoliko je štampaču dodeljen port 255, komunikacija Z80 sa ovim uređajem će se obavljati programom poput ovoga:

```
WAIT: IN A,(&FF)
      RLCA
      JR C,WAIT
      LD A,(DATA)
      OUT (&FF),A
```

Šta radi ovaj mali program? Najpre se u akumulator smešta broj koji je, da dopustimo sebi malo literarne slobode štampaču upisao u port 255 (ili &FF). Instrukcija RLCA prenosi najznačajniji bit akumulatora u carry indikator da bi se, sledećom instrukcijom, mikroprocesor vratio na tabelu WAIT u slučaju da je taj bit bio jedan. Petlja će trajati sve dok štampaču na port 255. Tek će tada mikroprocesor moći da pošalje podatke iz memorije na isti (ili neki drugi), port, a štampač će, čim ga primi, ponovo postaviti indikator u stanje „zaузet“ i držati ga u tom stanju sve dok postati karakter ne bude ispisao.

Pogledajmo ekvivalent gornjeg programa koji je pisan za mikroprocesor 6502:

```
WAIT: LDA &FC00
      ASL A
      BCS WAIT
      LDA DATA
      STA &FC01
```

Ovaj program nam, na prvi, drugi i treći pogled, izgleda u najmanju ruku čudno. Mikroprocesor najpre dovodi u akumulator sadržaj memorijske lokacije čija je adresa &FC00, a zatim, njenim šiftovanjem na levo i instrukcijom grananja, prelazi na tabelu WAIT ukoliko je najznačajniji bit ove ćelije jedinica. Reklo bi se da iz takvo dobijene petlje nema izlaza — ako je kritični bit bilo setovan, ne vidi se šta bi ga resetovalo. Možda interapt? Naravno da ne — rešenja koje koriste interapt retko zahtevaju od centralnog procesora da se vrli u mrtvoj petlji. Tajna je u tome što memorijska lokacija &FC00 ne pripada isključivo mikroprocesoru!

Za početak treba da shvatimo kako radi instrukcija LDA (lokacija). Prepoznajući se mikroprocesor na adresu magistralu stavija 16 bita (lokacije) i statusom kontrolnih

linija (jedna od njih je, na primer, R/W koja definiše da li mikroprocesor želi da upiše u memoriju ili da čita iz nje) zahteva pristup memoriji od koje se očekuje da na magistralu za podatke stavi pročitani podatak. Taj podatak se, rekli bismo, čita iz RAM-a ili iz ROM-a ali ne postoje nikakvi posebni razlozi da tako bude. U principu je sasvim ostvarljivo da LDA (lokacija) prenese na data bus sadržaj nekog flip-flopa ili niza flip-flova. Te flip-flobove će, sa druge strane, po želji setovati i resetovati neki periferijski uređaj prenoseći na taj način poruke mikroprocesoru.

Stvari su sada, nadamo se, postale malo jasnije. Instrukcija LDA &FC00 će, umesto da pročitati bajt iz RAM-a ili ROM-a, dostaviti mikroprocesoru sadržaj flip-flopa koji definiše spremnost štampača. Obzirom da je za odgovor na pitanje „da li si spreman“ dovoljan jedan bit informacija, vrlo je verovatno da će preostali bitovi magistrale za podatke biti primenom pull-up otpornika postavljeni na jedinice ili će prosto biti nedefinisani (uvek bismo vam preporučili prvo od ova dva rešenja kao „čistije“). Na sasvim sličan način instrukcija STA &FC01 će prenети sadržaj akumulatora na data bus da bi ga interfejs za štampač, pošto je dekodovao adresu „svoje“ memorijske lokacije, preuzeo i ispisao.

Posmatrajći memorijske mape nekih od popularnih mikroracunara verovatno ćete primetiti da se za neki periferijski uređaj odvajе na jedan bajt negde čitava strana memorije (256 bajta). Razlog za ovakvo raspisnoće je, paradoksalno, štednja! Ukoliko umesto 16 bita adrese dekodujemo samo prvih osam, štedimo na dekodorskim čipovima. Ukoliko projektujemo neki jednostavan kontroler zasnovan na mikroprocesoru možemo čak da se odlučimo na toliki luksuz da odustanemo od bilo kakvog dekodovanja i koristimo A15 kao signal koji će prozati periferijski uređaj! Na taj način se odričemo polovine čitavog adresnog prostora 32 K ali to i nije neka žrtva ukoliko se taj prostori i onako ne bi za šta koristio.

Koje su prednosti i mane memorijski mapiranih periferijskih uređaja u odnosu na IN i OUT instrukcije? Pokazuje se da neke velike razlike nema — jedna strana od 256 memorijskih lokacija nije neka silna žrtva, a na mikroprocesoru je uštedeno par pinova. IN i OUT instrukcije, sa druge strane, svakako doprinose čitljivosti mašinskog programa. Sve u svemu, rekli bismo da je korišćenje memorijski mapirane periferije prisutno kod jeftinijih i jednostavnijih mikroprocesora ali bi praksa demantovala tu izjavu — na Digitalovim sistemima (PDP, VAX) se koristi baš ova tehnika!

Ostalo je još jedno malo pitanje — šta sa memorijom koja je „pokrivena“ mapom periferijskih uređaja. Nema druge nego stisnuti zube i „baciti“ par strana u ROM-u koje će postati potpuno nepristupačne mikroprocesoru (zašto u ROM-u a ne u RAM-u? Steta je smanjivati korisničku memoriju a napisati ROM koji će imati tačno 32 (ili 16) kilobajta, kao što smo rekli, nije lako — zašto da ne iskoristite prazan prostor na njegovom kraju?). Tih 256 lokacija ćete, dakle, popuniti nulama? Samo ako nemate stila — mesto je kao stvoreno za vaš potpis, poruku vašoj devojci ili neku drugu tajnu koju će moći da pročitaju samo oni koji izvide ROM iz računara i analiziraju ga uz pomoć programatora EPROM-al

Dejan Ristanović



# ekran pod presom

Majstorije na računaru

spektrum

Poznato je da „spektrumov operativni sistem rezerviše 6912 bajtova za potrebe video memorije. Ti bajtovi se nalaze neposredno iznad ROM-a, počevši od adrese 16384. Da bi računar „zapamtio“ i prikazao sliku, potrebno mu je 6144 bajtova. Narednih 768 bajtova upotrebjeno je za skladištenje informacija o obojenosti pojedinih oblasti slike. Prva datoteka obično se naziva displej fajl, a druga atributi. Često se nam je potrebno da se u programu pojavljuje nekoliko vrlo detaljno nacrtanih slika. U grafičkoj obradi programa postoji vrlo jednostavno pravilo: veći broj slika — siromašniji detalji. S druge strane, korišćenjem nekog od programa za crtanje ili pomoćnog hardvera slika se može razviti do najfinijih detalja. U memoriju računara se tada može smestiti oko pet slika od po 6912 bajtova, a da nam za program još uvek ostanu oko 7 K slobodnog RAM-a. Korišćenjem jednostavnog trika u memoriju računara se može upakovati i znatno veći broj slika.

U mnogim igrama avanture (Hobbit, Urban Upristart, Eric, ...) računar iscrtaava svaki grafički prikaz. U RAM se upisuju ključne tačke crteža, a poseban podprogram ih spaja pravim ili krivim linijama. Prednost ovog metoda skladištenja crteža je ta što ih u programu možemo imati u velikom broju. Osnovne mane su sporost kojom računar crta i nedostatak detalja. Razvijeni su i drugi metodi za skladištenje video informacija. Podsetimo se izvanredne grafike u programu Mugsy, koja je dobijena kombinovanjem sprajtova sa prenošenjem datoteka atributa iz RAM-a u video memoriju. Slike dobijene na ovaj način su mnogo lepše i, što je možda važnije, prikazuju se gotovo trenutno. Mnogi programeri koriste i metode kompresije slika. Ova metoda odlikuje se velikom brzinom prikaza slike, a dopušta ilustratoru finiju obradu i veći broj detalja. (Pogledajte avanture HULK i Spider-Man)

## Slike iz kompresora

Kada se govori o kompresiji slike, ne misli se na njeno fizičko smanjivanje, već na sažimanje datoteke kojom je ona opisana. Za jednu te istu sliku često umesto uobičajenih 6912 bajta može biti dovoljno i 1K memorije. U svakom crtežu, naime, mora postojati ravnoteža između ispunjenih i praznih površina. Znamo da se video memorija sastoji od bajtova koji mogu imati vrednost između 0 i 255. Svaki postavljeni bit vidimo kao tačticu, a „pun“ bajt (255) predstavlja je crticom od 8 tačaka. Zašto trošiti memoriju na pamćenje bajtova koji

```

L
00010 *****
00020 * DEKOMPRESIJA SLIKE *
00030 *****
00040 ORG 30000
00050 LD A,2 :KANAL 2
00060 CALL #1601
00070 LD A,21 :-----
00080 RST 16 : OVER !
00090 LD A,1 : :
00052 LD A,1 : :
00100 RST 16 : PRINT
00110 LD A,22 : AT
00120 RST 16 : 0
00130 XDR A : :
00140 RST 16 : 0
00150 XDR A # 1 :
00160 RST 16 :-----
00170 *****
00180 * IZRACUNAVANJE POZICIJE *
00190 * DATOTEKE KARAKTERA *
00200 *****
00210 LD IX,(23296)
00220 POC LD A,(IX)
00230 CP 0
00240 JR Z,PET
00250 LD B,A
00250 P1 LD A,32
00270 RST 16
00280 DJNZ P1
00290 PET LD HL,(23684)
00300 JR PRP
00310 *****
00320 * PRENOŠENJE KARAKTERA *
00330 * NA IZRACUNATU ADRESU *
00340 *****
00350 POV LD B,8
00350 LOD INC IX
00370 LD A,(IX)
00380 LD (HL),A
00390 INC H
00400 DJNZ LOD
00410 *****
00420 * POMERANJE PRINT *
00430 * POZICIJE ZA JEDNO *
00440 * MESTO ULEVO *
00450 *****
00460 INC IX
00470 LD A,32
00480 RST 16
00490 *****
00500 * DA LI SMO STIGLI DO *
00510 * KRAJA DATOTEKE ? *
00520 *****
00530 PUSH IX
00540 POP DE
00550 LD HL,(23298)
00560 SBC HL,DE
00570 JR NC,PCC
00580 *****
00590 * AKO JESMO ONDA : *
00600 * OVER @:DECC.ATRIBUTA *
00610 *****
00620 LD A,21
00630 RST 16
00640 XDR A
00650 RST 16
00660 JR ATT
00670 *****
00680 * POPRAVKA ADRESE PRI *
00690 * PRELASKU IZ JEDNE U *
00700 * DRUGU TRECINU EKRANA *
00710 *****
00720 PRP PUSH HL
00730 LD DE,16640
00740 SBC HL,DE
00750 JR Z,N0
00760 POP HL
00770 PUSH HL
00780 LD DE,18588
00790 SBC HL,DE
00800 JR Z,N0
00810 POP HL
00820 JR POV
00830 N0 POP HL
00840 LD DE,1792
00850 ADD HL,DE
00860 JR POV
00870 *****
00880 * DEKOMPRESIJA ATRIBUTA *
00890 *****
00900 ATT LD HL,(23298)
00910 INC HL
00920 EX DE,HL
00930 LD HL,(23300)
00940 LD IX,22528
00950 P2 LD A,(DE)
00960 INC DE
00970 PUSH AF
00980 LD A,(DE)
00990 LD B,A
01000 POP AF
01010 PUSH HL
01020 PUSH DE
01030 SBC HL,DE
01040 POP DE
01050 POP HL
01060 RET C * IZLAZ *
01070 INC DE
01080 LPP LD (IX),A
01090 INC IX
01100 DJNZ LPP
01110 JR P2
01120 *****
01130 * PREBACIVANJE EKRANA *
01140 *****
01150 ORG 30200 :poc. adr
01160 LD BC,6912 :br.bajta
01170 LD DE,16384 :video m.
01190 LD HL,40000 :datoteka
01190 LDIR
01200 RET
01210 *****
01220 * A. Radovanovic (C) 1985 *
01230 *****

```

Novi programi, posebno iz domena avanturističkih igara i obrazovanja, ne mogu se ni zamisliti bez impresivne grafičke podrške — u avanturama slike predela izviru jedna za drugom kao marama iz madioničarevog šešira. Teško bi se mogao naći programer koji ne zna da je svaka slika u „spektrumu“ „teška“ preko šest kilobajta memorije. Gde su, onda smešteni svi ti silni pejzaži? Korišćenjem posebnih tehnika, svaka slika se, dok čeka na svoj red u memoriji, može sabiti na znatno uži prostor od onoga koji zauzima dok se prikazuje na ekranu. Aleksandar Radovanović, dobitnik druge nagrade na „Galaksijinom“ konkursu za najboljeg YU programera, piše o metodi koju je koristio za sažimanje u nagradnom programu „Velika akcija“.

```

30000 62 2 205 1 22 62
30005 21 215 62 1 215 62
30012 22 215 175 215 175 215
30018 221 42 0 91 221 126
30024 0 254 0 40 6 71
30030 62 32 215 16 251 42
30036 132 92 24 33 6 8
30042 221 35 221 126 0 119
30048 36 16 247 221 35 62
30054 32 215 221 229 209 42
30060 2 91 237 82 48 212
30065 62 21 215 175 215 24
30072 27 229 17 0 65 237
30078 82 40 12 225 229 17
30084 0 73 237 82 40 3
30090 225 24 203 225 17 0
30096 7 25 24 196 42 2
30102 91 35 235 42 4 91
30108 221 33 0 88 26 19
30114 245 26 71 241 229 213
30120 237 82 209 225 216 19
30126 221 119 0 221 35 16
30132 249 24 233 232 0 0

```

```

30200 1 0 27 17 0 64
30206 33 64 196 237 176 201

```

```

1 CLEAR 20000
5 PRINT "PRVI DEO KODA"
10 FOR N=30000 TO 30138
20 INPUT M: POKE N,M: NEXT N
30 PRINT "DRUGI DEO KODA"
40 FOR N=30200 TO 30211
50 INPUT M: POKE N,M: NEXT N
60 SAVE "C" CODE 30000,250

```

predstavljaju prazne površine? Pronađite netko igru koja ima naslovni ekran. Upišite zatim ovaj program:

```

10 LOAD "" SCREEN$: LET NUL=0
20 FOR N=0 TO 6144
30 IF PEEK (16384+N)=0 THEN LET
NUL=NUL+1
40 NEXT N:CLS
50 PRINT NUL; " BAJTA SU NULE"

```

Pustite da program učita ekran i prebroji nule. Videćete koliko je memorije otišlo za ovu svrhu.

Jedna od metoda kompresije zasniva se upravo na brojanju nula. Umesto da se pamti datoteka od 20 nula, u memoriju se može upisati: 0, 20. Kada naide na nulu, programer koji bude vršio „dekompresiju“ znaće da bajt koji zatim sledi predstavlja njehov broj. U ovom primeru uštedeno je čak 18 bajtova. To ne izgleda puno, ali takvih grupa sa manje ili više nula ima puno.

Ne moramo se, razume se, ograničiti samo na brojanje nula. Može se napraviti program koji će brojati bajtove koji se najčešće ponavljaju. To podrazumeva predhodnu analizu slike. Analiza video memorije tipa „bajt po bajt“ i kompresija i dekorativna primenljive su u slučajevima kada sažimamo datoteku jedne, dve ili sve tri trećine ekrana. Zbog (za ovu svrhu) nepovoljne organizacije video memorije, kompresiju moramo vršiti samo po trećinama.

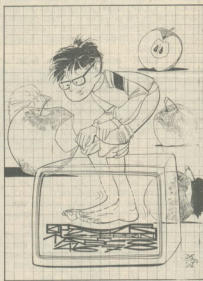
Mi smo se opredelili za metod sabijanja datoteke zasnovan na analizi grupa od po 8 bajtova. Svaka takva grupa formira karakter, a kompresija se vrši logičnim redom počevši od vrha ekrana. Predloženi program omogućava korisniku da sažima sliku bilo kog formata. Crtež je fiktivno podeljen na 22 reda sa po 32 karaktera, dakle saglasno PRINT koordinatama. Korisnik sam bira broj redova koje treba sažeti. Slike treba crtati u istom formatu, koristeći grafičku tablu ili program koji prikazuje mrežu atributa (Melbourne Draw). Program koji vrši kompresiju (listing 2) zasnovan je na analizi prikazanih karaktera. Funkcija SCREEN\$ vrši analizu sadržaja video memorije i ignoriše svaki prazan i svaki pun karakter (8x255). Oni donose memorijsku uštedu od po 8 bajtova. Datoteka se formira u grupama od 9 bajtova. Prvi bajt nosi informaciju o broju praznih, ili punih, znakova koji se nalaze ispred znaka čija je datoteka predstavljena so sledećih 8 znakova. Očigledno je da će ušteda biti veća ukoliko je slika manje ispunjena. Kako nama to nije cilj, pred dizajnera postavljamo zaista jednostavne uslove:

- 1) Kao podlogu za crtanje koristiti papir na kvadratice ili program koji prikazuje mrežu atributa.
  - 2) Svaki kvadrat, atribut, predstavlja po jedan znak — ne treba ulaziti u prazan karakter samo da bi se nacrtala jedna tačka.
  - 3) Ne koristiti „paint fill“ rutine, jer se svaki pun karakter može zameniti odgovarajućim atributom, odnosno INK, PAPER-om. Vizuelni efekat je potpuno isti. Program za kompresiju će opominjati na propuste ove vrste.
  - 4) Veće površine treba bojiti pomoću atributa odnosno PAPER-om, a samo ivice doraditi INK-om.
- Vidi se da je osnovni zahtev: što manje „Mastila“ (INK) u slici. Od crtača se, jednostavno, traži da mogućnosti računara iskoristi na pravi način, a njegove slabosti pretvori u prednosti.

Atributi se, takođe sažimaju. Datoteka se formira od parova bajtova. Prvi bajt označava atribut, a drugi broj njegovih ponavljanja.

### Sažimanje . . .

Program koji vrši kompresiju napisan je na bejziku. To znači da analiza slike može



potrajati i nekoliko minuta. Međutim, datoteka se sažima samo jednom, pa brzina, u osnovi, nije previše bitna. Program za dekompresiju napisan je na mašinskom jeziku i njega korisnik uključuje u svoj program zajedno sa datotekama kompresovanih slika.

Program pruža sledeće opcije:

- 1) LOAD učitava sliku koju želite da sažmete iznad adrese 40.000.
- 2) COMPRESS sažima datoteku. Pri izvršavanju ove opcije potrebno je malo sačekati.
- 3) SAVE snima na traku sažetu datoteku.
- 4) REST. COM. SCREEN vrši dekompresiju slike na osnovu datoteke dobijene sažimanjem. Ako se u slici pojave kvadratne „rube“, to znači da na tom mestu INK treba promeniti u PAPER.
- 5) FORMAT korisnik definiše format slike koju želi da sažme.
- 6) REST. ORG. SCR. jednostavno prikazuje ekran koji ste učitali u računar.
- 7) MEMORIA daje izveštaj o sažimanju — dužinu datoteke crteža i dužinu cele datoteke.
- 8) SAVE M. CODE snima na traku program za dekompresiju.

U slučaju da program zadržite sa BREAK ili se u opciji 1) pojavi „Tape loading error“, treba otkucati GOTO 7.000. Pomoću programa iz listinga 1 ukucajte navedenu datoteku i snimite je na traku. Resetujte računar i ukucajte listing 2. Program startuje sa GOTO 9.000 i učitaće ono što ste prethodno snimili. Kao program se može snimiti sa GOTO 9999.

## ... i „razvlačenje“ slike

Učitali ste nekoliko crteža, saželi ih, zapisali podatke o dužinama datoteka i snimili ih na traku. Kako uključiti crteže u neki svoj program? Opcija 8 snima mašinski program koji će vam biti potreban. Neka je kompresor dao sledeće podatke za neki crtež: ekran: 1000, ekran+atributi: 1200. Odučili ste se da datoteku stavite iznad adrese 60000, a program za dekompresiju iznad adrese 62000. Datoteku snimljenu opcijom 3 i mašinski program snimljeni opcijom 8 možete učitali pomoću instrukcija:

```
CLEAR 59999: LOAD "" CODE 60000:
LOAD "" CODE 62000. Program za dekom-
presiju će tražiti podatke o adresi početka
datoteke, o adresi kraja crteža i poslednjoj
adresu datoteke. Ti podaci se moraju nalaziti
u printer baferu:
23296 i 23297: adresa početka fajla. U
našem slučaju je to 60000.
23298 i 23299: adresa kraja crteža. Nama
ona iznosi 60000+1000=61000
23300 i 23301: adresa kraja datoteke. Za
nas primer to je 60000+1200=61200.
Bezijk instrukcije koju će te podatke preneti
u printer bafer mogu izgledati ovako:
RANDOMIZE 60000: POKE 23296, PEEK
23670: POKE 23297, PEEK 23671
RANDOMIZE 61000: POKE 23298, PEEK
23670: POKE 23299, PEEK 23671
RANDOMIZE 61200: POKE 23300, PEEK
23670: POKE 23301, PEEK 23671
U svakom slučaju, navedene adrese treba
razložiti na viši i niži bajt. Na kraju, startuje-
mo program za dekompresiju pomoću:
RANDOMIZE 59999: LOAD "" CODE 60000:
LOAD "" CODE 62000. Program za dekom-
presiju će tražiti podatke o adresi početka
datoteke, o adresi kraja crteža i poslednjoj
adresu datoteke. Ti podaci se moraju nalaziti
u printer baferu:
23296 i 23297: adresa početka fajla. U
našem slučaju je to 60000.
23298 i 23299: adresa kraja crteža. Nama
ona iznosi 60000+1000=61000
23300 i 23301: adresa kraja datoteke. Za
nas primer to je 60000+1200=61200.
Bezijk instrukcije koju će te podatke preneti
u printer bafer mogu izgledati ovako:
RANDOMIZE 60000: POKE 23296, PEEK
23670: POKE 23297, PEEK 23671
RANDOMIZE 61000: POKE 23298, PEEK
23670: POKE 23299, PEEK 23671
RANDOMIZE 61200: POKE 23300, PEEK
23670: POKE 23301, PEEK 23671
U svakom slučaju, navedene adrese treba
razložiti na viši i niži bajt. Na kraju, startuje-
mo program za dekompresiju pomoću:
```

Prenošenje podataka u printer bafer se
jednostavnije izvodi na mašinskom jeziku.
Numerički crteže brojevima 0, 1, 2, 3,
4... i neka registar A sadrži broj slike koju
želimo da prikazujemo. Neka se počevši od
labela SL nalaze podaci sa potrebnim adre-
sama. Pre početka dekompresije treba
izvršiti sledeći program:

```
5 REM K O M P R E S I J A
10 LET X=0: LET Y=0: LET P=0:
LET N=50000: POKE N,P
20 IF SCREEN$(Y,X)="" THEN L
ET P=P+1: POKE N,P: GO TO 30
30 PRINT AT Y,X: LET M=PEEK 2
3684+256*PEEK 23685
40 FOR K=0 TO 7
50 LET N=N+1
60 POKE N,PEEK M: LET M=M+256
70 NEXT K
80 LET P=0: LET N=N+1: POKE N,
P
90 LET X=X+1: IF X)=32 THEN LE
T X=0: LET Y=Y+1
100 IF Y)=INR THEN LET KR=N-1:
RANDOMIZE KR: POKE 23299,PEEK 23
670: POKE 23299,PEEK 23671: BEEP
.08,10: GO TO 200
110 GO TO 20
200 REM A T R I B U T I
210 LET AT=23228: LET K=INA
220 LET B=1
230 LET A=PEEK AT
240 POKE N,A: POKE N+1,B: LET A
T=AT+1: LET B=B+1: LET K=K-1: IF
K=0 THEN GO TO 300
245 IF B=255 THEN GO TO 260
250 IF PEEK AT=A THEN GO TO 240
260 LET N=N+2: GO TO 220
300 LET KRA=N+1: RANDOMIZE KRA:
PEEK 23300,PEEK 23670: POKE 233
01,PEEK 23671: GO TO 500
500 CLS: PRINT "Izvestaj o kom-
presiji:";"EKRA="ITAB 15: INV
ERSE 1:KRA-50000:" INVERSE 0:"EK
RAN+ATRIBUTI:" INVERSE 1:KRA-50
000:"Pocetak fajla je na adr.
50000:" PRINT 00:"Pritisnite nek
o dugme...": PAUSE 0: GO TO 7000
7000 CLS: PRINT INVERSE 1:" S C
R E E N C O M P R E S I O R ":"0
: INK 4: INVERSE 1" Radovanovic
Aleksandar 1985"
710 PRINT ""MENI:""1) LOAD
SCREEN""2) COMPRESS SCREEN""
3) SAVE FILE (50000,"IKRA-5000
```

```
01")""4) RESTORE COMP. SCREEN"
""5) SCREEN FORMAT""6) RESTOR
E ORG. SCREEN""7) MEMORIJA""
8) SAVE M. CODE (30000,140)""
7020 PAUSE 0: BEEP .008,-10
7030 IF INKEY=""1" THEN LOAD ""C
ODE 40000: GO TO 7000
7040 IF INKEY=""2" THEN CLS: RA
NDOMIZE 59999: LOAD ""CODE 60000:
LOAD ""CODE 62000: GO TO 10
7050 IF INKEY=""3" THEN GO SUB 7
300
7060 IF INKEY=""4" THEN CLS: RA
NDOMIZE 59999: PRINT 00:"Pri-
tisnite neko dugme...": PAUSE 0:
GO TO 7000
7070 IF INKEY=""5" THEN GO SUB 8
000: GO TO 7000
7080 IF INKEY=""6" THEN CLS: RA
NDOMIZE 59999: PRINT 00:"Pri-
tisnite neko dugme...":
PAUSE 0: GO TO 7000
7090 IF INKEY=""7" THEN GO TO 50
0
7100 IF INKEY=""8" THEN RANDOMIZ
E 59999: SAVE "MACHINE CODE
"CODE 30000,140: GO TO 7000
7200 GO TO 7020
7300 POKE 23517,158: INPUT "IME
FAJLA ":" LINE N#: IF LEN N#)10
OR N#="" THEN GO TO 7300
7310 POKE 23517,0: SAVE "MACHINE
CODE 50000,(KRA-49999): BEEP .08,10:
RETURN
8000 POKE 23517,158: INPUT "BROJ
REDOVA (1-22) : INR: IF INR)22
OR INR=0 THEN GO TO 8000
8010 POKE 23517,0: LET INA=32*IN
R: RETURN
9000 CLEAR 29999: LET KR=50000:
LET KRA=50000: POKE 23298,0: PO
KE 23300,0: POKE 23296,0: POKE
23297,195: POKE 23299,195: POKE
23301,195: LOAD ""CODE : BORDER
0: PAPER 0: INK 7: CLS: GO SUB
8000: GO TO 7000
9999 CLEAR: SAVE "COMPRESSOR" LI
NE 9000: SAVE "CODE "CODE 30000,
250
```

## Devpak 80

Poznata softverska firma HISOFT, koja se pruža, naročito među vlasnicima „spektruma“, po svom izvanrednom paskal kompajleru i sigurno najkvalitetnijem asembler i disasembleru trenutno na tržištu (DEVPAC), izdala je nedavno paket od tri programa, pod nazivom „DEVPAC 80“, namenjen računarnima sa CP/M operativnim sistemom, na bazi procesora 280. Paket obuhvata ekranски editor, asembler i disassembler, a cena mu je 40 funti. Sve tehničke pojedinosti se mogu dobiti neposredno od proizvođača: HISOFT, 180 High Street North, Dunstable, Beds. LU6 1AT.

J. S.

## Korak dalje

Pre prikazivanja slike potrebno je obrisati prethodni sadržaj ekrana, jer će se slike preklopiti. Ovo je urađeno namerno jer dozvoljava korisniku da kombinuje više crteža. Brisanje prethodnog sadržaja može se izvršiti naredbom CLS ili štampanjem praznih karaktera na željeno mesto.

FOR n=0 TO 320:PRINT ""; NEXT n
U mašinskom jeziku mogužetie upotrebiti svega jedan bit (npr. carry flag) za dojavu da li se prethodna slika briše ili ne.

Dalji razvoj programa bi se kretao u pravcu pisanja mašinskih rutina za kompresiju, kao i rutina za sažimanje već komprimovane datoteke (rutine za tokenizaciju). Kompresija datoteka crteža, kao i postupci sažimanja teksta, predstavljaju osnovu za pisanje bogato ilustrovanih programa sa velikim fondom reči.

```
LD HL, SL
LD B,5
LD C,A

LBL ADD A,C
DJNZ LBL
ADD HL, BC
LD DE, 23296
LD B,C
LDIR
```

dekompresija

SL DEFW 60000, 61000, 61200; prvi crtež

DEFW 61201, 63000, 63456; drugi crtež itd...

Na osnovu broja crteža program izračunava adresu na kojoj se nalaze traženi podaci i prenosi ih u printer bafer. Zatim se vrši dekompresija. Ulaz u program se vrši samo preko jednog registra.



# hamlet

Programiranje  
u bejziku

# u računaru

Kako kompjuteri  
donose odluke

**Mogućnost donošenja odluka je jedna od najznačajnijih prednosti računara nad kalkulatorima. Zahvaljujući upravo njoj, računarima se danas prepušta široko polje ljudske delatnosti — od izbora trase kojom će biti probijen novi put do donošenja tako suptilnih odluka kao koje osobe preporučiti za životnog saputnika klijentima bračnog savetovališta. Kako računar zna da donese određenu odluku i koje vrste odluka mogu da donose računari?**

Mogućnost donošenja odluka počiva na naredbama grananja koje obezbeđuju upravljanje redosledom izvršavanja instrukcija programa u zavisnosti od toga da li je ispunjen neki uslov. Na primer, naredba

```
IF G<18 THEN PRINT „MALOLETAN“
```

označava da će poruka „MALOLETAN“ biti štampana jedino ako je vrednost promenljive G, koja u ovom kontekstu označava godine, manja od 18. Ako je vrednost G jednaka ili veća od 18, odnosno ako uslov G<18 nije ispunjen, preskočiće se sve što u posmatranom programskom redu sledi iza službene reči THEN i upravljanje će se prenети na bejzik interpretator ili sledeći programski red, ako je odgovarajuća naredba sastavni deo bejzik programa.

**IF . . . THEN**

IF . . . THEN (ako . . . onda) je jedna od naredbi prenosa upravljanja koja može narušiti uobičajeni redosled izvršavanja instrukcija. Podrazumeva se, naime, da se instrukcije izvršavaju onim redom kojim su fizički smeštene u program. Na početku se izvršava prva naredba, kada se njena realizacija okonča prelazi se na sledeću i tako redom. Da bi se ovaj implicitni redosled mogao promeniti, u većini programskih jezika, pa i bejziku, imamo mogućnost da instrukcijama dodelimo adrese (leve adrese) na koje zatim eksplicitno prenosimo upravljanje instrukcijama bezuslovnog prelaska tipa GOTO ili instrukcijama uslovnog prelaska. Ovo je sasvim u skladu sa pravilima koja postoje kod jezika nižeg nivoa u kojima svaki registar koji sadrži komandu ima adresu, a u hardver su uključene komande prelaska koje predaju upravljanje prema zadatoj adresi. Osim u ovoj osnovnoj formi, instrukcije prelaska u nekim jezicima mogu da se koriste i na razne druge načine, tako da, na primer, leve adrese instrukcija ne moraju više biti numeričke već mogu biti i simboličke.

Kakvo je dejstvo instrukcije IF . . . THEN koja je sastavni deo većine verzija bejzika ilustrovaćemo sledećim primerom. Ovaj jednostavan program štampa prosečnu cenu artikala koje smo doneli iz samousluge i čije su cene ulazne veličine programa.



```

10 PRINT „UNESI JEDNU PO JEDNU
CENU ARTIKALA“
20 PRINT „OTKUČAJ -1 AKO NEMA
VIŠE PODATAKA“
30 LET S=0
40 LET B=0
50 INPUT C
60 IF C=-1 THEN PRINT „PROSEČNA
CENA IZNOSI“; S/B: STOP
70 LET S=S+C
80 LET B=B+1
90 GOTO 50

```

Napomena: Za račun „galaksija“ izostavite THEN u liniji 40 i LET u naredbama dodele.

Obratite pažnju da program neće regularno završiti sa radom ako prvi podatak koji unosite bude -1, jer u tom slučaju očkujete od računara da izračuna koliko je 0/0 što nije definisano. Podsetimo se još da bi program normalno radio na „komodor“ i „Acorn“ računarnima i bez linija 30 i 40 jer oni podrazumevaju da su početne vrednosti promenljivih nule.

Program radi na sledeći način. Na početku se odštampaju uputstva zapisana u linijama 10 i 20 i zatim se ulazi u ciklus u kome treba da unesete cenu artikla koja se dodaje ukupnoj sumi, a Brojač se uvećava za jedan. Ovaj ciklus se ponavlja sve dok ne unesete karakteristični podatak -1 koji znači da ste uneli cene svih artikala. Tada se, zahvaljujući ispunjenju uslova u liniji 40, izvršavaju naredbe koje u ovom programskom redu slede iz službene reči THEN, odnosno štampa se prosečna cena artikala i zaustavlja se izvršavanje programa.

### Složeni uslovi

Ako odvijanje programa zavisi od više uslova, onda se problem može rešiti pisanjem više uslovnih naredbi ili kombinovanjem uslova u jednoj IF... THEN naredbi pomoću logičkih operatora AND, OR i NOT. Ovaj drugi način pokazaćemo na primeru igre JACKPOT za koju vam prilazimo listinje za sva četiri računara.

```

1 REM *****
2 REM *
3 REM * JACKPOT ZA SPEKTRUM *
4 REM *
5 REM *****
6 REM
20 LET M=500
30 CLS
40 LET M=M-50
50 IF M<0 THEN PRINT"ZALIM, IZGUBIO SI."
: STOP
60 LET A=INT(RND*(12)+130)
70 LET B=INT(RND*(12)+130)
80 LET C=INT(RND*(12)+130)
210 PRINT PAPER0; INK4; AT10,14; CHR$A1
214 REM
215 REM PROVERA USLOVA ZA DOBITAK ILI
PREMIJU

```

```

216 REM
220 IF A=B AND B=C THEN PRINTAT13,2;"DOB
TO SI PREMIJU OD 500 DINARA";LETHM=500
230 IF (A=B OR B=C) AND A<>C THEN PRINT
AT13,9;"DOBIO SI 100 DINARA";M=M+100
240 PAUSE 25
250 PRINTAT15,8;"IMAS JOS";M;"DINARA";AT
16,8;"IDES LI DALJE?"(D/N)";
260 IF INKEY$="" THEN GOTO 260
270 IF INKEY$="N" THEN STOP
280 GOTO 30

```

```

1 REM *****
2 REM *
3 REM * JACKPOT ZA ACORN RACUNARE *
4 REM *
5 REM *****
6 REM
20 M=500
30 CLS
40 M=M-50
50 IF M<0 THEN PRINT"ZALIM, IZGUBIO SI
": STOP
60 A=RND(12)+224
70 B=RND(12)+224
80 C=RND(12)+224
110 PRINT TAB(17,10);CHR$A17;CHR$A;" "1
CHR$B;" "1CHR$C
214 REM
215 REM PROVERA USLOVA ZA DOBITAK ILI
PREMIJU
216 REM
220 IF A=B AND B=C THEN PRINTTAB(17,12);"
DOBIO SI PREMIJU OD 500 DINARA";M=M+500
230 IF (A=B OR B=C) AND A<>C THEN PRINTTAB
(14,12);"DOBIO SI 100 DINARA";M=M+100
240 FOR D=1 TO 1500: NEXT D
250 PRINT TAB(13,16);"IMAS JOS";M;"DINARA
";
TAB(15,16);IDES LI DALJE?PRINT"(D/N)";
260 LET K=GET$
270 IF K$="D" THEN GOTO30
280 END

```

```

1 REM *****
2 REM *
3 REM * JACKPOT ZA KOMODOR *
4 REM * 1.VERZIJA *
5 REM *
6 REM *****
7 REM
10 POKES3280,0: POKES3281,0: PRINT
CHR$(30)
20 LET M=500
30 PRINT""
40 LET M=M-50
50 IF M<0 THEN PRINT "ZALIM, IZGUBIO SI"
:STOP
60 LET A=INT(RND(11)+4)+1
70 LET B=INT(RND(11)+4)+1
80 LET C=INT(RND(11)+4)+1
90 IF A#1 THEN LET A=97
110 IF A=2 THEN A=115
110 IF A=3 THEN A=120
110 IF A=4 THEN A=122
110 IF B=1 THEN B=97
110 IF B=2 THEN B=115
110 IF B=3 THEN B=120
110 IF B=4 THEN B=122
110 IF C=1 THEN C=97
110 IF C=2 THEN C=115
110 IF C=3 THEN C=120
110 IF C=4 THEN C=122
210 PRINT"TAB(15)(CHR$(A)SPC(3)CHR$(B)SP
C(3)CHR$(C))
214 REM
215 REM PROVERA USLOVA ZA DOBITAK ILI
PREMIJU
216 REM
220 IF A=B AND B=C THEN PRINTTAB(5);"DOBI
D SI PREMIJU OD 500 DINARA";M=M+500
230 IF (A=B OR B=C) AND A<>C THEN PRINT
TAB(13);"DOBIO SI 100 DINARA";M=M+100
240 FOR D=1 TO 1500: NEXT D
250 PRINT"IMAS JOS";M;"DINARA.";PRINT"ID
ES LI DALJE?"(D/N)";
260 GET Q$; IF Q$<"D" AND Q$<"N" THEN
GOTO 260
270 IF Q$="D" THEN GOTO 30
280 END

```

```

216 REM
220 IF A=B AND B=C THEN PRINTTAB(5);"DOBI
D SI PREMIJU OD 500 DINARA";M=M+500
230 IF (A=B OR B=C) AND A<>C THEN PRINT
TAB(13);"DOBIO SI 100 DINARA";M=M+100
240 FOR D=1 TO 1500: NEXT D
250 PRINT"IMAS JOS";M;"DINARA.";PRINT"ID
ES LI DALJE?"(D/N)";
260 GET Q$; IF Q$<"D" AND Q$<"N" THEN
GOTO 260
270 IF Q$="D" THEN GOTO 30
280 END

```

Ono što ste verovatno primetili i posle letimičnog pogleda na listinje je da nedostaje listing za računar „galaksija“ i da je program za „komodor“ mnogo veći i različitiji od programa za „Acorn“ i „spektrum“. Stoga ćemo posle kratkog objašnjenja kako ovih priloženih programa rade pokušati da uvođenjem novih instrukcija sačinimo ekvivalentan ali elegantniji program za „komodor“ i, po ugledu na njega, da i za „galaksiju“ — koja ne samo što nema operatore AND, OR i NOT već ne može koristiti ni izvedene relacijske simbole kao <= na primer, a da i ne pričamo o njenim grafičkim mogućnostima — damo program koji bi omogućio simulaciju JACKPOT-a.

U navedenim programima koristimo nekoliko IF... THEN naredbi. Prva, u liniji 50, proverava da li imate dovoljno novca za

igru. Naime, u našoj verziji igru počinjete sa 500 dinara, svaki pokušaj košta 50 dinara i pri tome možete da dobijete 100 dinara, ako su dva susedna znaka jednaka, premiju od 500 dinara ako su sva tri znaka jednaka ili da ne dobijete ništa u svim ostalim slučajevima. Ukoliko imate dovoljno novca, upravljanje se prenosi na sledeći programski red, a ukoliko nemate, štampa se poruka zapisana u nastavku programskog reda i igra se prekida. Linije 60 do 80 biraju tri slučajna broja, koji se u liniji 210 konvertuju u karaktere i štampaju na centru ekrana.

Na „spektrum“ i „Acorn“ ti se brojevi direktno konvertuju u karaktere, a „komodor“ (vrlo nezgrapno) u dodatnih 12 linija konvertuje svaki slučajni broj od 1 do 4 u jednu od boja na kartama — herc, tref, pik ili karo i takođe linijom 210 štampa izabranu kombinaciju na centru ekrana.

U liniji 220 proverava se složeni uslov da li su sva tri znaka ista, pa ako jesu uvećava se vaša suma za 500 dinara.

U liniji 230 proverava se da li je ispunjen uslov da su samo dva susedna znaka jednaka i ako jesu proizvodi se uvećanje sume novca za 100 dinara.

Ukoliko nemate dobitak, dolazi se po proveru uslova u linijama 220 i 230, do linije 240 u kojoj se zavrti „prazna“ FOR... NEXT petlja koja je zapisana jedino da bi se isporučilo izvršavanje programa, a zatim u sledećih nekoliko linija nalazimo na DA/NE rutinu koja nam postavlja pitanje da li želimo da nastavimo igru i na osnovu odgovora donosi odluku da li da prekinemo ili nastavimo izvršavanje programa.

Ono što smo želeli da ilustrujemo ovim programima je korišćenje složenih uslova u IF... THEN naredbi. Uslov A=B AND B=C je zadovoljen ako je tačno i da je A=B i da je B=C, a da bi uslov A=B OR B=C bio zadovoljen dovoljno je da bar jedan od uslova povezanih operatorom OR bude ispunjen. Ako sa 0 označimo da uslov nije ispunjen, a sa 1 da jeste (u računarnima interpretacija „istine“ i „laži“ može biti i drugačija, recimo kod „komodora“ „istina“=-1, odnosno sve jedinice u bajtu), onda složeni uslovi imaju vrednost koja se formira prema sledećim tablicama istinitosti.

X	NOT X	X	Y	X AND Y	X	Y	OR Y
0	1	0	0	0	0	0	0
1	0	0	1	0	0	1	1
		1	0	0	1	0	1
		1	1	1	1	1	1

Ako u jednom složenom uslovu imamo više logičkih operatera, onda malim zagradama definišemo redosled izračunavanja istinitosne vrednosti uslova.

### Strukturirano IF

Vratimo se sada na opasku da program za komodor nije dovoljno „lep“. Ono što najviše smeta je to što se iz njegovog listinje ne može videti šta on, u stvari, radi, što nije strukturiran. (Ista primedba, uostalom, važi i za prethodna dva listinje, ali su oni, barem, kraći.) Strukturiranje nekog programa ujedno određuje i njegovu preglednost. Mada standardni bežik nije projektovan za strukturirano programiranje, dobro je da se pri pisanju programa pridržavamo njegovih pravila. Iako troše više memorijskog prostora i vremena, strukturirani programi nagradju korisnike lakom čitljivošću i preglednošću.

Dobro je da se potprogrami poredaju na kraju programa i označe REM linijama. Dalje, trebalo bi vidno naglasiti delove programa. Kod petlji se uvid u njihovo dejstvo postiže pomeranjem instrukcija koje čine telo ciklusa.

```
10 FOR I=1 TO 10
20 ::: A=A+1
30 ::: PRINT I,A
40 NEXT I
Struktuirano IF je takođe preglednije.
10 IF AS="STAMPATI" THEN 60
20 ::: PRINT "NIJE ZAHTEVANO STAMPANJE"
30 ::: PRINT "ZELITE LI IZDAVANJE NA EKRAN (D/N)?"
40 ::: INPUTXS:IF XS "D" AND XS "N" GOTO 30
50: IF XS="D" THEN AS="PRIKAZATI"
60 GOSUB 1000
```

### IF... THEN... ELSE

Sada se lakše može uočiti šta se preduzima u NE grani IF ispitivanja. Međutim, još bi bolje bilo kada bismo u repertoaru instrukcija imali naredbu

```
IF <uslov> THEN <da-grana> ELSE <ne-grana>
```

Iz koje se s lakoćom može videti ne samo šta se preduzima ako je uslov ispunjen, već i šta treba činiti ako nije. Ovakve instrukcije, na žalost, ne postoje u standardnom „spektrumovom“ i „komodorovom bejziku, ali i poseduju „...“, „Acorn“ i „galaksija“.

Zadržimo se na „galaksijinih“ naredbama uslovnog prelaska. Opšti oblik naredbe uslovnog prelaska razlikuje se unoekoliko od uobičajene IF... THEN naredbe ostalih računara. On glasi:

```
IF <uslov> <lista naredbi>.
```

Prva razlika je što ovde nema službene reči THEN posle uslova, a ako se napiše interpretator je neće prihvatiti. Druga razlika je u uslovu, odnosno relacijskom izrazu. Kod ostalih računara mogu se porediti dva numerička ili azbučna (string) izraza na isti način, s tim što se azbučni izrazi poredi tako da se proverava odnos ASCII kodova odgovarajućih karaktera, tako da i za njih može da se utvrdi da li su u nekoj od šest mogućih relacija: <=<=<=><=><=>. Kod „galaksije“ je, međutim, za dva brojna izraza moguće utvrditi jedino da li su u relaciji manje, jednako ili veće, jer „galaksija“ nema izvedene relacijske simbole, a za dve azbučne promenljive se jedino može utvrditi da li su jednake i to ako se uslov zapiše na sledeći način:

```
EQ <az.prom>, <az.prom>
```

Uz to, umesto uslova može se zapisati i tastaturni relacijski izraz

```
KEY <{br. izraz}>
```

koji ima sledeće značenje: izračuna se celobrojna vrednost izraza u zagradi i, ako je pritisnut taster sa odgovarajućim brojem, uslov je ispunjen. Međutim, zahvaljujući naredbama

```
IF <uslov> <lista naredbi> : ELSE <lista naredbi>
```

```
IF <uslov> [:] ELSE <lista naredbi>
```

„galaksija“ kompenzuje nedostatke svoje IF naredbe u odnosu na „...“ i „spektrum“ i „komodor“ i čak omogućava znatno bolje struktuiranje programa.

### Višestruko grananje

U programima se često traži da, u zavisnosti od vrednosti neke promenljive, upravljanje možemo da prenesemo na više od dve grane programa. Tako recimo u zavisnosti od toga da li je vrednost nekog brojnog izraza negativna, jednaka null ili pozitivna možemo preduzeti tri razne akcije, a u našem programu u zavisnosti od toga koji je slučajaj broj generisan možemo štampati četiri različita znaka. Ovakvo grananje se, razume se, može ostvariti i na do sada opisane načine kombinacijom uslova u jednoj IF... THEN naredbi ili korišćenjem više uslovnih naredbi, ali je elegantnije rešiti problem jednom novom

```
ON <brojni izraz> GOTO n1,n2,...,nk
```

Instrukcijom koja po izračunavanju celobrojne vrednosti brojnog izraza između ON...GOTO prenosi upravljanje na jedan od programskih redova čiji je broj naveden u listi Iza GOTO i to na prvi iz spiska ako je vrednost izraza jednaka 1, na drugi ako je 2, na k-ti, ako je vrednost po izračunavanju izraza jednaka k.

Dajemo sada listing programa za računar „komodor“ koji je ekvivalentan prethodnom, ali je zbog korišćenja pravila struktuiranog programiranja i instrukcije ON...GOTO znatno pregledniji.

```
1 REM *****
2 REM *
3 REM * JACKPOT ZA KOMODOR *
4 REM * 2. VERZIJA *
5 REM *
6 REM *****
7 REM
10 POKES3280,0: POKES3281,0: PRINT CHR$(30)
20 LET M=500
30 PRINT "M=500"
40 LET MH=50
50 IF KO THEN PRINT "ZALIM, IZUBIJO SI "
60 STOP
100 FOR I=1 TO 3
110 I=(I+1)*INT(RND(1)+.4)+1
120 I:ON A(I) GOTO 130,140,150,160
130 I:ON A(I)=97: GOTO 170
140 I:ON A(I)=118:GOTO 170
150 I:ON A(I)=120:GOTO 170
160 I:ON A(I)=122
170 NEXT I
180 A=(I+1) B=A(2) : C=A(3)
190 PRINT "TAB(15)CHR(A)SPC(3)CHR(B)SP(C)CHR(C)"
214 REM
215 REM PROVERA USLOVA ZA DOBITAK ILI PREMIZU
216 REM
220 IF A=B AND B=C THEN PRINTTAB(15):DOBI O SI PREMIZU OD 500 DINARA:MH=500
230 IF (A=B OR B=C) AND A=C THEN PRINT AB(13):DOBI SI 100 DINARA:MH=100
240 FOR D=1 TO 1500: NEXT D
250 PRINT "IMAS JOB" :M:"DINARA."PRINT ID EB LI DAL'JE? (D/N)
260 SET DR: IF DR<"D" AND DR<"N" THEN GOTO 260
270 IF DR="D" THEN GOTO 30
280 END
290 PRINT "IMAS JOB" :M:"DINARA."PRINT ID EB LI DAL'JE? (D/N)
300 SET DR: IF DR<"D" AND DR<"N" THEN GOTO 260
310 IF DR="D" THEN GOTO 30
320 END
```

U ovoj formi instrukcija višestrukog grananja mogu da koriste „Acorn“ računari i „komodor“, a „spektrum“ i „galaksija“ koriste instrukciju

```
GOTO b
```

gde vrednost promenljive b definiše broj programskog reda na koji se prenosi upravljanje. Ilustraciju primene ovako zapisane instrukcije višestrukog grananja možete videti u listingu za igru JACKPOT na računaru „galaksija“.

```
1 REM *****
2 REM *
3 REM * JACKPOT ZA KOMODOR *
4 REM * 2. VERZIJA *
5 REM *
6 REM *****
7 REM
10 POKES3280,0: POKES3281,0: PRINT CHR$(30)
20 LET M=500
30 PRINT "M=500"
40 LET MH=50
50 IF KO THEN PRINT "ZALIM, IZUBIJO SI "
60 STOP
100 FOR I=1 TO 3
110 I=(I+1)*INT(RND(1)+.4)+1
120 I:ON A(I) GOTO 130,140,150,160
130 I:ON A(I)=97: GOTO 170
140 I:ON A(I)=118:GOTO 170
150 I:ON A(I)=120:GOTO 170
160 I:ON A(I)=122
170 NEXT I
180 A=(I+1) B=A(2) : C=A(3)
190 PRINT "TAB(15)CHR(A)SPC(3)CHR(B)SP(C)CHR(C)"
214 REM
215 REM PROVERA USLOVA ZA DOBITAK ILI PREMIZU
216 REM
220 IF A=B AND B=C THEN PRINTTAB(15):DOBI O SI PREMIZU OD 500 DINARA:MH=500
230 IF (A=B OR B=C) AND A=C THEN PRINT AB(13):DOBI SI 100 DINARA:MH=100
240 FOR D=1 TO 1500: NEXT D
250 PRINT "IMAS JOB" :M:"DINARA."PRINT ID EB LI DAL'JE? (D/N)
260 SET DR: IF DR<"D" AND DR<"N" THEN GOTO 260
270 IF DR="D" THEN GOTO 30
280 END
290 PRINT "IMAS JOB" :M:"DINARA."PRINT ID EB LI DAL'JE? (D/N)
300 SET DR: IF DR<"D" AND DR<"N" THEN GOTO 260
310 IF DR="D" THEN GOTO 30
320 END
```

Napominjemo da računari „...“ Acorn imaju mogućnost primene i instrukcije GOTO b i da od sva četiri računara imaju bejzik koji najbolje zadovoljava zahteve struktuiranog programiranja. Ono što najviše smeta struktuiranju programa u bejziku su instrukcije GOTO.

### Koji se bagovi najčešće javljaju u programima?

Postoji mnoštvo mogućnosti da se na prave greške — bagovi u programu, a samo jedan način da program radi — da bude bez grešaka. Većina programera ima neprepušten talenat za komplikacije programa i pravljenje originalnih grešaka i bilo bi potrebno veoma mnogo prostora da se sve pobroje. Zadržaćemo se stoga na najčešćima.

- + zamena malog slova i ili velikog i sa cifrom 1
- + zamena slova O i cifre 0
- + zaboravljanje navodnika na kraju print naredbe
- + zaboravljanje na tipku RETURN ili ENTER na kraju programskog reda pri prelasku na sledeću instrukciju
- + zaboravljanje zarezu između dva broja u DATA naredbi može dovesti do broja koji se ne može registrovati
- + zaboravljanje znaka — (u nekim programima koji generišu grafiku ovo liči na uputstvo računaru da štampa nešto „izvan ekrana“



## Pitanja i odgovori

1. Da li je dobro smestati više IF... THEN instrukcija u jednom programskom redu?

Obično je to loša ideja. Princip rada IF... THEN u bejziku je takav da se uzima u razmatranje deo iza THEN jedino ako je zadovoljen uslov između IF i THEN, ali ako to nije slučaj računar naprosto prelazi u sledeći programski red i ignoriše sve iza THEN. Međutim, ova ideja ponekad može i korisno da se primeni kao u sledećem primeru:

```
70 IF X=Y THEN PRINT "Isteklo je vreme";
IF ZIVOTI>0 THEN
ZIVOTI=ZIVOTI-1
80 IF X=Y AND ZIVOTI=0 THEN PRINT
"Igra je završena"
```

Tu igrač gubi život jedino ako je zadovoljen prvi uslov, ali upozorenje da je vreme isteklo neće biti štampano onako ne bude zadovoljen uslov X=Y

2. Da li se dobija na brzini izvršavanja programa ako se umesto više jednostavnih IF... THEN instrukcija upotrebi jedna sa složenim uslovom?

Na nivou mašinskih instrukcija postoje samo proste naredbe prelaska s jednom adresom. Shodno tome, provera složenog uslova mora se svoditi na odgovarajući broj jediničnih mašinskih instrukcija prelaska. S druge strane, ako umesto složene provere u jednoj IF... THEN naredbi zapisimo više jednostavnih IF naredbi, odgovarajućim redosledom provera možemo značajno smanjiti vreme izvršavanja programa. Na primer, ako imamo zapise tri tipa sa kodovima 1, 2 i 3 i 90% ih pripada tipu 1, 9% tipu 2 i 1% tipu 3, onda bi najbolje bilo postavljati pitanja ovim redom:

```
100 IF KOD=1 THEN.....GOTO 150
110 IF KOD=2 THEN.....GOTO 150
120 IF KOD=3 THEN.....GOTO 150
140.....(identifikacija greške)
```

Sa ovakvim redosledom pitanja će se u 90% slučajeva proveravati samo jedan uslov, u 9% dva uslova i samo u 1% tri uslova. Srednji broj provera iznosiće

$$0.9 \times 1 + 0.09 \times 2 + 0.01 \times 3 = 1.1$$

a u slučaju ako bismo postavljali obrnut red pitanja bilo bi u proseku

$$0.01 \times 1 + 0.09 \times 2 + 0.9 \times 3 = 2.89$$

provera uslova. Od ovoga može i gore, ako na kraju IF... THEN linije izostavimo GOTO 150 kada imamo za svaki zapis po tri provere.

U primeru pomenutom na početku teksta, ako bi se klijent obratio elektronskoj prodavničkoj sa molbom da mu odabere nevestu koja bi bila mlada, privlačna, pametna, nevina i imala 1 000 000 dolara miraza, najbrži program postavio bi poslednju želju kao prvo pitanje.

## Sporovi oko GOTO

Instrukcije bezuslovnog prelaska prave problema kako programerima tako i onima koji treba da koriste njihove programe. Što

## Problemi sa uslovima

Ako prethodno niste koristili simbole "veće od" i "manje od", ovi operatori važe u prvom susretu mogu zbuditi. Evo, zato, potpune liste relacija u kojima se mogu naci numerički i izračunne promenljive (napomena; za računar, "galaksija" ne važe sve mogućnosti, vidi tekst!)

```
A>B se čita "A je veće od B"
A=B se čita "A je jednako B"
A<B se čita "A je manje od B"
A<=B se čita "A je manje ili jednako od B"
i suprotno je relaciji "veće"
A<>B se čita "A je različito od B" i suprotno
je relaciji "jednako"
A<=B se čita "A je veće ili jednako B" i
suprotno je relaciji "manje".
```

Na .... Acornovim računarima poslednja tri izvedena relacijska znaka moraju se kućati upravo ovakvim redom, a na "spektrumu" se mora koristiti samo jedan taster — ako otkucate, na primer, <p> onda =, računar to neće prihvatiti.

se programera tiče, već i vrapci znaju da mnoštvo instrukcija prelaska u jednom programu svedoči o njegovoj lošoj razradi. Program sa mnoštvom GOTO-va se teško testira i razume. Osnovnu teškoću predstavlja praćenje strukture takvih programa, jer nas pri analizi ovakvih programa svaka GOTO instrukcija stavlja na muku pošto može da prenese upravljanje u bilo koji deo programa, a odakle — to možemo videti jedino ako imamo ceo listing pred očima.

Međutim, mada se može programirati i bez GOTO instrukcija, proste forme ovakvih prelazaka dozvoljavaju direktnu hardversku realizaciju u klasičnim računarima, a ako ih nema mora se koristiti čitav niz drugih mehanizama. Zato, bar na početku, ne anatemišemo GOTO, ali preporučujemo da ih konzumirate u minimalnim količinama (sem ako vam se zamorio profesor programiranja, pošite mu tada programe sa puno divljih GOTO skokova po celom programu a algoritam izostavite "jer je vaše rešenje očigledno").

## REPEAT...UNTIL

.... Acorn računari imaju dve ekstra naredbe REPEAT... UNTIL koje se mogu uvek koristiti umesto IF... THEN. One su korisne kada želimo da ponavljamo deo programa uvek iznova sve dok se ne naruši uslov za ponavljanje. U igrama to može biti uslov da još imamo bombe, na primer. Uz korišćenje IF... THEN segment programa bi mogao da izgleda ovako:

```
50 .....(početak glavnog programa
...
200 IF bombe=0 THEN PRINT "Izgubio
si!"; END
210 GOTO 50
A uz REPEAT... UNTIL ovako:
45 REPEAT
50 .....(početak glavnog programa)
...
200 UNTIL bombe=0
210 PRINT "Izgubio si!"; END
```

čime dobijamo jasno strukturalan program koji se, uz to, i brže izvršava.

Nevenka Spalević

## Zvučno pero

Aparat izgleda kao naliv-pero na gajtanu. Međutim, ako se vrh postavi na arhitektonski crtež i povlači oko perimetra, za nekoliko minuta dobijte precizni proračun na koje bi u normalnim slučajevima bili utrošeni mnogi časovi.

Sistem se zasniva na sonarnoj tehnici, koja koristi zvučni za merenje razdaljine. Kada se pero, ili zvučni digitizator, postavi na crtež, ono emituje pucketave zvuke, koje prihvataju dva mikrofona unutar ultrazvučnog polja na crtaču koji ih pretvara u dimenzije do kompjutera koji ih pretvara i dimenzije i prikazuje na ekranu i štampa. Sa crtaču kuća, na primer, trenutno može da se izračuna koliko zemlje treba da se ukloni, koliko cigala i crepova da se utroši, koliko treba drveta za pod stakla za prozore, pa čak i kolika je zapremina kađe u kupatilu. Sve ovo može da se odredi bez pisanih nekoliko sekundi. Sistem može da se iskoristi i za dobijanje parametara dizajna: soba zasica može da dobije nove dimenzije. Posebno je koristan kada treba da se izmere površine nepravilnih oblika — na primer, puteva. Brzina proračuna zavisi od brzine kojom se pero povlači. Kompaktnog dizajna, jednostavno za korišćenje, pero može da uštedi 90% vremena potrebnog za ručne proračune.

Projektovala u firmi „Techsonix“ (Aston Science Park, Love Lane, Birmingham, B7 4BJ, England), pero predstavlja jedan od prvih proizvoda prvenstveno za potrebe građevinske industrije. Ono, međutim, može da se primeni i u mnogim drugim oblastima — prilikom eksploatacije nafte iz mora, u proizvodnji tepiha, u industriji plastičnih masa, za izradu mapa vazdušnih puteva i slično.

## Modeliranje života

Uz pomoć verovatno najsvršenijeg kompjuterskog softvera na svetu za tu vrstu istraživanja, na Kraljevskom društvu koleđa Birkboka u Londonu obavlja se proučavanje složene trodimenzionalne strukture proteina, od bitnog značaja kod izrade tradicionalnog dizajna lekova i vakcina.

Softver koristi grafički sistem za interaktivno ispitivanje ugrađivanja potencijalnih lekova u proteine, uzimajući u obzir ne samo potrebu da odgovara obliku nego i druga hemijska svojstva kao što je, na primer, naelektrisanje. Proteini koji se modeliraju za svrhe istraživanja na ekranu i njima se na razne načine manipuliše pomoću tastature.

Da bi se istakla precizno međudejstvo između složenih proteinskih struktura i drugih, često manjih, molekula kao što su lekovi koristi se analogija „brave i ključa“. Proteini su od suštinske važnosti za život i posreduju u mnogim biološkim procesima, koristeći svoje izvijene trodimenzionalne strukture koje dejstvuju kao „brava“ što precizno prepoznaje druge molekule, „ključeve“. Kod racionalne izrade lekova naučnici počinju da koriste svoja saznanja o strukturi proteina da bi predložili trodimenzionalni oblik lekova — „ključa“ koji tačno odgovara „bravi“, ali je ne zabavljuje.

# Kako studirati računarstvo

## Vodič za maturante

*Za školsku 1985/86 godinu na fakultetima širom Jugoslavije otvoreni su brojni odseci, smerovi i grupe na kojima je zastupljeno studiranje računarstva. Ovu novinu uvele su visokoškolske ustanove, počev od elektrotehničkih, prirodno-matematičkih, ekonomskih, fakulteta tehničkih nauka do pojedinih viših škola. Studije računarstva nalaze se u sklopu odseka, smerova i grupa koje imaju opšti naziv — informatika. Ovim prilogom dajemo osnovne uslove upisa i kriterijume za izbor kandidata na prvu godinu studija za školsku 1985/86 godinu. I pored velike želje da ovim pregledom obuhvatimo sve fakultete u Jugoslaviji, to nam, na žalost, nije pošlo za rukom, jer nismo uspjeli da na vreme dobijemo potrebne informacije.*

### BEOGRAD

**Priradno-matematički fakultet, Studentski trg 16, tel. 011/180-111**  
Na ovom fakultetu u okviru studijske grupe **matematika** postoji smer — računarstvo i informatika.

Uslovi upisa za studijsku grupu matematika su sledeći: Prijavu za upis mogu podneti kandidati koji su završili usmereno obrazovanje za IV stepen stručne spreme u struci koja je odgovarajuća za studije matematika i to: matematičko-tehničkoj, prirodno-tehničkoj, mašinskoj, mašinsko-energetskoj, elektrotehničkoj, građevinskoj, rudarskoj, geološkoj, hemijskoj-tehnološkoj, grafičkoj, staklarskoj, keramičkoj, vatrostalnoj, vatrosnoj, za proizvodnju građevinskog materijala, geodetskoj, tekstilnoj, kožarskoj, gumarskoj, dvoprenosivnoj, saobraćajnoj, hidrometeorološkoj, šumarskoj, metalurškoj, poljoprivrednoj, prehrambenoj, prosvetnoj (pripremi stepen pedagoške akademije) i veterinarskoj. Kandidati koji nemaju odgovarajući smer za ove studije polažu dopunski ispit iz matematike prema programu fakultetne nastave. Oni koji nisu završili III, odnosno IV stepen stručne spreme, odnosno srednje obrazovanje polažu prijemni ispit iz predmeta: matematika, fizika, hemija i pismeni ispit iz srpskohrvatskog jezika po programu usmerenog obrazovanja prirodno-tehničke struke.

Prijem prijava kandidata koji nemaju odgovarajuće prethodno obrazovanje obaveće se 17. i 18. juna 1985. godine.

Prijava kandidata sa odgovarajućim prethodnim obrazovanjem i sprovođenje konkursa obaveće se u sledećim rokovima:

#### Prvi konkursni rok:

Podnošenje prijave od 24. do 28. juna, klasifikacioni ispiti 1. i 2. jula, objavljivanje rezultata 3. jula i upis primljenih kandidata od 5. do 10. jula 1985. godine.

#### Dругi konkursni rok:

Prijave za ovaj konkursni rok će se primati samo ukoliko ne bude primljen dovoljan broj kandidata putem prvog konkursnog roka. Podnošenje prijave 13. jula, ispiti za sticanje opštih uslova za upis 15. jula, objavljivanje rezultata ispita 16. jula i upis primljenih kandidata 17. i 18. jula 1985. godine.

**Priradno-matematički fakultet u Beogradu za studijsku grupu matematika** — smer računarstvo i informatika prima 65 redovnih i 12 vanrednih studenata.

**Više škola za primenjenu informatiku i statistiku, Kneza Miloša 20, tel. 011/643-996, 659-397.**

Na ovoj višoj školi postoji osek informatike na koji će ove godine biti primljeno 190 redovnih i 40 vanrednih studenata.

Na prvu godinu studija, na osek informatike, mogu se upisati kandidati koji su završili IV stepen stručne spreme koja je tesno povezana sa matematikom, programiranjem. Zatim mogu konkurisati i kandidati koji su završili klasičnu gimnaziju i druge srednje tehničke škole.

Kandidati koji nemaju odgovarajuću srednju školu polažu dopunski ispit iz matematike. Ukoliko polože, tek tada mogu konkurisati za klasifikacioni ispit koji je u obliku odlog testa iz oblasti informatike.

Podnošenje prijave za konkurs 27. i 28. juna, klasifikacioni ispiti 1. jula, objavljivanje rezultata konkursa 8. jula, upis primljenih kandidata 9. i 10. jula.

Drugi konkursni rok, ukoliko ne bude popunjen predviđen broj mesta, počinje od 15. do 19. jula.

### NIŠ

**Elektronski fakultet, Beogradska 14, tel. 018/46-243, 46-286/875**

Za sticanje visoke spreme na fakultetu postoji sledeći osek — za prenos i obradu informacija sa smerovima — računarska tehnika i informatika i telekomunikacije. Za sticanje više stručne spreme postoji sledeći smer — za prenos i obradu informacija.

Na prvu godinu studija mogu se upisati kandidati koji imaju IV stepen stručne spreme, odnosno oni koji su završili IV stepen stručne spreme. Zatim gimnaziju ili

drugu četvorogodišnju školu: srednju, tehničku, elektro ili mašinskog smera i školu za KV radnike-radlo, elektro, mašinskog i hemijskog smera.

Kandidati koji nemaju odgovarajuće srednje obrazovanje prethodno polažu dopunski ispit iz matematike.

Podnošenje prijave kandidata bez odgovarajuće srednje školske spreme 15. juna, ispiti za sticanje opštih uslova za upis i objavljivanje rezultata ispita 18. i 19. juna.

Prvi konkursni rok: od 20. juna do 1. jula — podnošenje prijave za konkurs od 20. do 22. juna, klasifikacioni ispiti od 25. do 26. juna, objavljivanje rezultata 25. i 26. juna, upis primljenih kandidata 29. i 30. juna.

Ukoliko u prvom konkursnom roku ne bude popunjen predviđen broj mesta za upis u I godinu studija postoji i drugi konkursni rok: — Podnošenje prijave kandidata bez odgovarajuće školske spreme od 10. jula, ispiti za sticanje opštih uslova za upis i objavljivanje rezultata ispita od 11. do 12. jula. Drugi konkursni rok traje od 13. do 20. jula, podnošenje prijave za konkurs 13. jula, klasifikacioni ispiti 16. i 17. jula, objavljivanje rezultata do 19. jula, upis primljenih kandidata najkasnije do 20. jula 1985. godine.

### NOVI SAD

**Fakultet tehničkih nauka**  
Na ovom fakultetu postoji mašinski osek koji ima smer — upravljačko-informacioni sistemi u informaciji.

Na prvu godinu studija mogu se upisati kandidati koji su završili IV stepen stručne spreme (saobraćajni, mašinski, građevinski, elektrotehnički, informatički, smer matematički i fiziki).

Kandidati koji nemaju odgovarajuću srednju školu prijavljuju se od 5. do 7. juna, a od 11. do 14. juna polažu dopunski ispit. Ukoliko se prijavi više kandidata, polaže se klasifikacioni ispit iz matematike. Podnošenje prijave za konkurs od 20. do 22. juna, a polaganje je 24. i 25. juna.

Za maturovanje iz oblasti tehničkih nauka u Novom Sadu primiče dve godine 40 redovnih i 10 vanrednih studenata.

### SUBOTICA

**Ekonomski fakultet, tel. 024/41-312**

Na ovom fakultetu postoji smer — informatika (s tim što se studenti opredeljuju na pojedine smerove tek posle treće godine studija)

Na prvu godinu studija, smer informatike mogu se upisati kandidati koji su završili: gimnaziju, ekonomsku, trgovačku, zatim informatičku i matematičku srednju školu. Kandidati koji nemaju odgovarajuću srednju školu polažu dopunski ispit iz dva predmeta: Političke ekonomije i ekonomije OOUR-a.

Ukoliko se prijavi veći broj kandidata, polaže se klasifikacioni ispit koji se sastoji iz dva dela — obavezi koji se sastoji iz: osnova marksizma i osnova političke ekonomije. Drugi deo klasifikacionog ispita sastoji se od dva predmeta i kandidat se može opredeliti između deset kombinacija.

Podnošenje prijave za konkurs je od 20. do 24. juna, dok će klasifikacioni ispiti biti 5. jula.

Ekonomski fakultet u Subotici za smer informatike prima dve godine 30 redovnih i 10 vanrednih studenata.

### SARAJEVO

**Elektrotehnički fakultet, tel. 071/524-182.**

Na ovom fakultetu postoji osek za informatiku.

Na prvu godinu studija mogu se upisati kandidati koji su završili IV stepen stručne spreme. Polaže se klasifikacioni ispit iz matematike. Rok za prijavljivanje kandidata još nije određen, ali će verovatno biti u prvoj polovini juna, dok će polaganje klasifikacionog ispita biti između 20. i 24. juna.

### ZAGREB

**Elektrotehnički fakultet, Unaka bb, tel. 041/514-811**

Na elektrotehničkom fakultetu u Zagrebu postoji osek računarska tehnika — telekomunikacioni i informatika.

Na prvu godinu studija mogu se upisati kandidati koji su završili IV stepen stručne spreme. Klasifikacioni ispiti se polaže iz fizike i matematike.

Podnošenje prijave za konkurs je od 1. do 26. juna.

**Priradno-ovno-matematički fakultet, OOUR Matematički odelj, Socijalističke revolucije 8, tel. 041/414-079.**

Na prvu godinu studija mogu se upisati kandidati koji su završili IV stepen stručne spreme. Prilikom prijema polaže se klasifikacioni ispit iz matematike i fizike.

Podnošenje prijave za konkurs je od 1. do 28. juna, dok je klasifikacioni ispit 1. jula. Na Priradno-ovno-matematičkom fakultetu OOUR Matematički odelj za profesore matematike i informatike ima mesta za 60 studenata.

### SPLIT

**Fakultet elektrotehnike, strojarstva i brodogradnje, Rudera Boškovića bb, tel. 068/561-777.**

Na ovom fakultetu postoji usmerenje i računarstvo se studira u okviru odseka elektronika.

Na prvu godinu studija mogu se upisati kandidati koji su završili IV stepen stručne spreme. Kandidati polažu klasifikacioni ispit iz matematike i fizike.

Podnošenje prijave za konkurs je od 20. juna, dok je klasifikacioni ispit 1. jula. Ukoliko se ne prijavi dovoljan broj kandidata, postoji i drugi konkursni rok koji počinje 15. dana kasnije.

Za smer-elektroniku u ovoj godini ima mesta za 150 studenata.

# svirka na "komodoru" umetnost programiranja

U prvom delu su date samo osnovne instrukcije radi lakšeg upoznavanja procesa stvaranja zvuka. Osnovna karakteristika svakog zvuka je njegova boja. Ona zavisi od oblika talasa koji daje taj zvuk. Osnovni talasni oblici su sinusoidi, dok se složeniji (trouglasti, tasterasti, pravougaoni) mogu dobiti kombinacijom osnovnih. Kod „komodora 64“ generator raspolaže sa ova tri složena talasa i sa šumom. Drugi talasni oblici se mogu dobiti njihovim mešanjem ili korišćenjem filtera. Postoje tri osnovna tipa filtera: niskopropusni, opsežno — propusni i visokopropusni. Ostali se mogu dobiti njihovom kombinacijom. Svi oni rade na osnovu zadane odsadne frekvencije. Niskopropusni filteri oduzimaju, iz talasnog oblika, sve sinusoidne koje imaju frekvenciju iznad odsadne frekvencije. Opsežno-propusni filteri oduzimaju sve sinusoidne koje imaju frekvenciju izvan opsega određenog odsadnog frekvencijom. Visokopropusni filteri oduzimaju sve sinusoidne koje imaju frekvenciju ispod odsadne frekvencije.

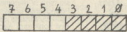
Pored boje, karakteristika zvuka je i njegova frekvencija. Ona predstavlja broj ponavljanja osnovnog talasa u jednoj sekundi. Ako za neki zvuk ne možemo da odredimo frekvenciju, tj. ne postoji osnovni talas koji se periodično ponavlja, onda taj zvuk predstavlja šum.

Karakteristika zvuka je i njegova jačina ili amplituda osnovnog talasa. Amplituda se ne može menjati u toku vremena što doprinosi bogatstvu zvuka. Zakon promene amplitude se naziva envelope. Jedna jednostavna envelope bita bi da se zvuk naglo pojavi pa postepeno utiša (kao kod klavira). Složena envelope bi se dobila praćenjem dinamike (jačine) orkestra koji svira delo iz klasične muzike. Kod „komodora 64“, envelope se određuje sa četiri parametra: attack, decay, sustain i release. „Attack“ predstavlja vreme potrebno da amplituda od nule dođe do maksimuma. „Decay“ predstavlja vreme potrebno da se amplituda od maksimuma spusti na nivo određenim parametrom „sustain“. „Release“ predstavlja vreme potrebno da se amplituda sa tog nivoa spusti do nule.

„Komodora 64“ poseduje tri odvojena glasa, što znači da istovremeno može proizvesti tri zvuka različitih karakteristika. Za svaki glas se može odrediti frekvencija, talasni oblik, envelope i puštanje kroz filtera. Tipovi filtera i jačina zvuka se određuju za sva tri glasa istovremeno. Posle ovog teorijskog uvoda, pogledajmo kako izgledaju nove instrukcije.

© CLR — resetuje generator zvuka tako što upisuje nule u njegove registre (54272—54296). Poželjno je koristiti ovu naredbu pre svakog rada sa zvukom.

© VOL <jačina zvuka> — određuje jačinu zvuka za sva tri glasa koja može biti ceo broj od 0 do 15. Dobienu vrednost upisuje u donja četiri bita (0—3) registra 54296.

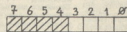


Najčešće se koristi jačina 15: VOL 15.

WAY <glas>, <talasni oblik>, <talasni oblik>, ... (ponavlja se onoliko puta koliko želite, dok ne pređe dužinu od 80 znakova) — određuje talasni oblik za svaki glas posebno. Glasovi se označavaju brojevima od 1 do 3, a talasni oblik po sledećoj konvenciji:

- 0 — trouglasti oblik
- 1 — tasterasti oblik
- 2 — pravougaoni oblik promenljive širine
- 3 — šum

Ukoliko date više vrednosti za talasni oblik, krajnji talas će biti jednak logičkoj disjunkciji (AND) pojedinih talasa. Vrednost za talasni oblik se upisuje u registre 54276 za prvi glas, 54283 za drugi i 54290 za treći. Svi ovi registri imaju oblik.



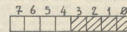
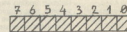
Ako su naznačeni bitovi setovani (postavljeni na 1), glas će proizvoditi odgovarajući talasni oblik

- bit 4 — trouglasti
- bit 5 — tasterasti
- bit 6 — pravougaoni promenljive širine
- bit 7 — šum

© WAV 1,0 će odrediti da prvi glas proizvodi samo trouglasti oblik.

© WAV 2,1,2 će odrediti da drugi glas proizvodi talas koji će biti jednak disjunkciji tasterastog i pravougaonog oblika.

© PUL glas (1—3) širina — određuje širinu pravougaonog talasa za svaki glas posebno koja može biti ceo broj od 0 do 4095. Dobienu vrednost upisuje u registre 54274—5 za prvi glas, 54281—2 za drugi i 54288—9 za treći. Svi ovi registri (za svaki glas) imaju oblik.



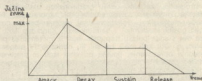
Od višeg bajta se koriste samo donja četiri bita (0—3)

© PUL 3,3115 će odrediti da širina pravougaonog talasa za treći glas bude 3115.

Ako je širina 0 ili 4095, pravougaoni talas neće davati nikakav zvuk.

© ENV <glas (1—3)>, <Attack>, <Decay>, <Sustain>, <Release> — određuje promenu jačine zvuka u toku vremena za svaki

glas posebno. U ovoj instrukciji, svi parametri, osim glasa, moraju biti ceo brojevi od 0 do 15. Proces generiranja zvuka počinje tako što bit 0 registra za talasni oblik (za svaki glas posebno) prvo resetujemo (postavimo na 0) kao pre nije bio postavljen na 0 pa setujemo. Tada će jačina zvuka krenuti od nula ka svojoj najvećoj vrednosti, određenoj instrukcijom © VOL, i postići je za vreme određeno parametrom „Attack“. Nakon toga će početi da se spušta i stići će do nivoa određenog parametrom „Sustain“ za vreme „Decay“. Ona će ostati na tom nivou sve dok ne resetujemo bit 0 registra za talasni oblik. Tada će se zvuk utišati do kraja za vreme „Release“. Setovanje i resetovanje bita 0 registra za talasni oblik odgovara pritisivanjem i puštanjem dirke na klaviru. Dobiene vrednosti za envelope se upisuje u registre 54277—8 za prvi glas 54284—5 za drugi glas i 54291—2D154—2 za treći glas. U gornja četiri bita prvog registra (za svaki glas) se upisuje vrednost za „Attack“, u donja četiri bita vrednost za „Decay“. U gornja četiri bita drugog registra se upisuje vrednost za „Sustain“, a u donja četiri bita vrednost za „Release“.



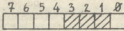
© ENV 1,0,9,0,5 će odrediti da prvi glas proizvodi zvuk sličan klaviru ili gitari.

© ENV 3,5,0,15,2 će odrediti da treći glas proizvodi zvuk sličan flauti ili violini.

© SPE <glas (1—3)>, <efekat>, <efekat>, ... (ponavlja se onoliko puta koliko želite, dok ne pređe dužinu od 80 znakova) — određuje specijalne efekte za svaki glas posebno. Efekti se označavaju po sledećoj konvenciji:

- 1 — sinhronizacija sa komplementarnim glasom (prvi sa trećim, drugi sa prvim, treći sa drugim)
- 2 — „ring“ modulacija (prstenasta modulacija) sa komplementarnim glasom
- 3 — isključivanje odgovarajućeg glasa

Da bi se dobila prstenasta modulacija, potrebno je, pored korišćenja instrukcije © SPE, odrediti trouglasti talasni oblik instrukcijom WAV. Efekti se dobijaju upisivanjem u bitove od 1—3 registra za talasni oblik:



Ako su naznačeni bitovi setovani, postići ćete odgovarajući efekat:

- bit 1 — sinhronizacija
- bit 2 — „ring“ modulacija
- bit 3 — isključivanje glasa



Računar „komodor64“ ima, u svojoj klasi, izvanredni generator zvuka, ali je, na žalost, njegova softverska podrška vrlo slaba (sve se mora raditi pomoću naredbe POKE). Zbog toga se javila potreba za dodavanjem novih bezijk naredbi koje bi podržavale generator zvuka, grafičku visoke rezolucije, sprajtlove... Ovaj kratak program predstavlja samo deo jednog većeg programa za upravljanje generatorom zvuka čiji će ostatak biti objavljen u narednim brojevima „Računara“.

©SPE 2.3 će isključiti drugi glas.

©WAV 1.0: SPE 1.2 će omogućiti efekat prstenaste modulacije na prvom glasu, kojim se dobija zvuk sličan zvonu ili gongu.

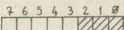
©FLT <glasovi>, <filteri> — određuje koji će filteri biti korišćeni i koji glasovi će biti propušteni kroz njih. „Glasovi“ je parametar koji određuje koji će glas biti propušten kroz filtere i može biti oblika 0 (nijedan glas ne proilazi kroz filtere), 1 (prvi glas prolazi kroz filter) 2,3,12 (prvi i drugi glas se propuštaju kroz filtere), 13, 23, 123 (svi glasovi se propuštaju kroz filtere). „Filteri“ je parametar koji određuje koji će filteri biti korišćeni. Ima isti oblik kao „glasovi“, ali je značenje cifara 1,2 i 3 drukčije:

1 — niskopropusni filter (propušta frekvencije koje su ispod odsečne frekvencije)

2 — opsežno-propusni filter (propušta frekvencije koje su u određenom opsegu)

3 — visokopropusni filter (propušta frekvencije koje su iznad odsečne frekvencije)

Dobijena vrednost parametra „glasovi“ se upisuje u donja tri bita registra 54295:



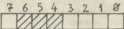
Ako su naznačeni bitovi setovani, odgovarajući glasovi će biti propušteni kroz filtere:

bit 0 — prvi glas će biti propušten kroz filtere

bit 1 — drugi glas će biti propušten kroz filtere

bit 2 — treći glas će biti propušten kroz filtere

Dobijena vrednost parametra „filteri“ se upisuje u bitove od 4—6 registra 54296, za jačinu zvuka:



Ako su naznačeni bitovi setovani, odgovarajući filteri će biti uključeni:

bit 4 — niskopropusni filter

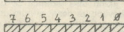
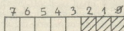
bit 5 — opsežno-propusni filter

bit 6 — visokopropusni filter

©FLT 1.3 će odrediti da se samo prvi glas propušta kroz visokopropusni filter.

©FLT 23,13 će odrediti da se samo drugi i treći glas propuštaju kroz filtere sačinjen od niskopropusnog i visokopropusnog filtera.

©CUT<odsečna frekvencija> — određuje odsečnu frekvenciju koja može biti ceo broj od 0 do 2047 za sva tri filtera. Promenom odsečne frekvencije dobijaju se razne boje tonova. Njena vrednost upisuje se u registre 54293—4 oblika



```

10 SFO=I=0: B=0
20 FORT=1: T09
25 READSM(1)
27 NEXT T
30 FORD=49169TD09BBSTFEB
35 FORT=XTDX+7
37 READA: S=S+A
38 POKEIT, A
40 NEXT T
45 B=B+1
47 IFB<310THENB=0
50 B=0: I=1+I: Z=S+0
55 IFB(1) = Z THENB=0
57 PRINT "GRESKA U BLOKU": I: STOP
60
65 SYB49B75
70 DATA8140,8460,9157,9539,9287
75 DATA7944,8144,9956,9686
80 REM *****
85 REM *****
90 REM *****
100 REM *****
101 REM ***** BLOK 1 ***
102 DATA169,0,133,13,8,216,24,165
103 DATA122,105,4,133,122,144,2,230
104 DATA123,40,86,32,17,192,32,158
105 DATA183,138,240,4,201,4,144,6
106 DATA32,72,178,76,174,167,202,134
107 DATA2,138,8,216,10,101,2
108 DATA133,2,40,169,0,133,251,96
109 DATA32,17,192,32,158,138,2,1
110 DATA16,144,6,32,72,178,76,174
111 DATA167,133,251,173,16,192,41,240
112 REM *****
113 REM ***** BLOK 2 ***
114 DATA5,291,141,16,192,141,24,212
115 DATA76,174,167,32,36,192,32,253
116 DATA174,32,235,183,165,20,10,10
117 DATA10,10,133,20,138,41,15,5
118 DATA20,164,2,153,3,192,32,253
119 DATA174,32,235,183,165,20,10,10
120 DATA10,10,133,20,138,41,15,5
121 DATA20,164,2,153,4,192,76,174
122 DATA167,169,0,160,25,153,295,211
123 DATA134,208,250,160,17,153,295,191
124 REM *****
125 REM ***** BLOK 3 ***
126 DATA136,208,250,32,115,0,32,115
127 DATA60,74,74,147,32,36,192,160
128 DATA0,177,122,201,44,208,29,32
129 DATA115,0,32,158,183,224,4,144
130 DATA6,32,72,178,76,174,167,232
131 DATA169,8,10,202,208,252,5,251
132 DATA133,251,144,219,164,2,185,2
133 DATA192,41,15,5,251,153,2,192
134 DATA76,174,167,32,36,192,160,0
135 DATA177,122,201,44,208,30,32,115
136 REM *****
137 REM ***** BLOK 4 ***
138 DATA60,32,158,183,224,4,144,6
139 DATA32,72,178,76,174,167,232,56
140 DATA169,0,42,202,208,252,5,251
141 DATA133,251,144,218,164,2,185,2
142 DATA192,41,240,5,251,153,2,192
143 DATA76,174,167,32,36,192,32,253

```

```

144 DATA174,32,138,173,32,247,183,165
145 DATA21,201,16,144,6,32,72,178
146 DATA76,174,167,164,2,165,20,153
147 DATA90,192,165,21,153,1,192,76
148 REM *****
149 REM ***** BLOK 5 ***
150 DATA174,167,32,17,192,32,158,183
151 DATA224,3,208,1,232,224,12,208
152 DATA2,162,3,224,13,208,2,162
153 DATA5,242,23,208,2,162,6,224
154 DATA23,208,2,162,7,138,41,248
155 DATA240,6,32,72,178,76,174,167
156 DATA173,15,192,41,248,134,2,5
157 DATA2,141,15,192,32,253,174,32
158 DATA158,183,224,3,208,1,232,224
159 DATA12,208,2,162,3,224,13,208
160 REM *****
161 REM ***** BLOK 6 ***
162 DATA2,162,5,224,23,208,2,162
163 DATA4,224,123,208,2,162,7,138
164 DATA41,248,240,6,32,72,178,76
165 DATA174,167,138,10,10,10,133
166 DATA2,173,16,192,41,15,5,2
167 DATA15,15,192,76,174,167,32,174
168 DATA192,32,138,173,32,247,183,165
169 DATA21,41,248,240,6,32,72,178
170 DATA76,174,167,165,20,14,7,141
171 DATA21,122,165,21,74,102,20,74
172 REM *****
173 REM ***** BLOK 7 ***
174 DATA102,20,74,102,20,165,20,141
175 DATA22,212,76,174,167,32,17,192
176 DATA32,158,183,138,41,240,240,6
177 DATA32,72,178,76,174,167,138,10
178 DATA15,15,192,76,174,167,32,174
179 DATA41,15,5,2,141,15,192
180 DATA174,167,32,36,192,32,253,174
181 DATA32,138,173,32,247,183,8,216
182 DATA165,2,201,5,208,2,105,1
183 DATA201,10,208,2,105,3,133,251
184 REM *****
185 REM ***** BLOK 8 ***
186 DATA168,165,20,153,0,212,165,21
187 DATA153,1,212,166,2,165,2,105
188 DATA5,233,252,40,189,0,192,153
189 DATA2,122,232,200,228,252,208,244
190 DATA173,15,192,141,23,212,173,16
191 DATA192,141,24,212,164,251,166,2
192 DATA89,2,192,9,1,153,2,41
193 DATA76,174,167,32,115,0,201,6
194 DATA240,3,76,231,167,160,1,177
195 DATA122,201,86,208,3,76,73,192
196 REM *****
197 REM ***** BLOK 9 ***
198 DATA201,69,208,3,76,108,192,201
199 DATA156,208,3,76,162,192,201,87
200 DATA208,3,76,189,192,201,83,208
201 DATA176,210,192,201,80,208,3
202 DATA76,44,193,201,70,208,7,36
203 DATA83,193,201,67,208,3,76,207
204 DATA193,201,82,208,3,76,254,193
205 DATA201,66,208,3,76,135,194,76
206 DATA174,167,169,194,141,9,3,169
207 DATA124,141,8,3,96,0,0,0
208 END

```

©CUT 720 će odrediti da odsečna frekvencija bude 720 za sva tri filtera.

©RES<rezonanca> — određuje vrednost rezonance koja može biti od 0 do 15 za sva tri filtera. Rezonanca predstavlja naglo povećanje amplitude na odsečnoj frekvenciji.

©RES 15 će odrediti da vrednost rezonance bude 15 za sva tri filtera.

©BOP <glas (1—3)><frekvencija> — određuje frekvenciju tona i generiše zvuk

za svaki glas posebno. Frekvencija može biti ceo broj od 0 do 65535. Generisani zvuk ima karakteristike koje su određene pomoću prethodnih naredbi.

©BOP 2, 6000 će proizvesti ton frekvencije 6000 na drugom glasu. Vrednost frekvencije se upisuje u registre 54272—3 za prvi glas, 54279—80 za drugi glas i 54286—7 za treći glas.

Đorđe Janković

# besmrtnost Računari i igre i kako je steći

Vlasnici kompjutera koji su u bilo kom trenutku bili zaraženi igrama znaju koliko je teško postati majstor čak i u igri koja se igra deset časova na dan. Problem je u tome što prve nivoe igre lako savladavamo a onda nailaze teži problemi za koje uobičajena tri života nisu dovoljna — za trenutak stradamo ne stigavši da upoznamo čak ni probleme koje treba da savladamo. Prinudjeni smo da počnemo od početka i da se dugo i dosadno probijamo kroz poznate ekrane da bismo se samo trenutak igrali na posljednjem. Vrio su reiki autori programa kao što su „Moon Rider“ i „Penetrator“ — po startovanju ovakvih igara na ekranu vidimo meni u kome biramo ekran od koga želimo da počnemo. Podrazumeva se da će svaki pravi haker pokušati da obezbedi sebi ono bez čega su ga autori igre ostavili — besmrtnost.



Iz stranih i domaćih kompjuterskih časopisa znamo da su se vlasnici „spektruma“ i „komodora“ potrudili da otkriju POKE-ove koji u komplikovanim igrama obezbeđuju

igračku neograničen broj života. U stranim časopisima posvećenim BBC-ju i „electronu“ koje imamo priliku da čitamo („Acorn User“, „A&B Computing“, „Beebug“ i, ponekad, „Micro User“) ovakvih POKE-ova nema ni za lek: ili su igre za Acornove računare predobro zaštićene, ili među njihovim vlasnicima nema pravih hakera, ili časopisi ne žele da pomažu nepoštenim igračima. Bilo kako bilo, autor ovoga teksta

**Snapper** za BBC: u drugom delu programa (Snap2) iza "load snapper ubacite ?&1344=&EA: ?&1345=&EA."

**Planetoid** za BBC: u drugom delu programa iza "load planetoid ubacite ?&1D9E=&EA: ?&1D9F=&EA ukoliko želite beskonačno života i ?&1AC9=&EA: ?&1ACA=&EA ukoliko želite i beskonačno bombi."

**Positron** za BBC i „electron“: ?&4967=&EA: ?&4968=&EA daje beskonačno života.

**Killer Gorilla** za BBC: četiri NOP-a (&EA) počev od &1E88 daju beskonačno života, dva NOP-a počev od &1E39 daju beskonačno vreme za svaki zadatak.

**Chuckie Egg** za BBC: na &2A45 treba ubaciti instrukciju JMP &2A70.

**Swoop** za BBC: na &2500 treba ubaciti JMP &243B.

**Swoop** za „electrona“: na &1B47 i &1B48 treba ubaciti dva NOP-a (&EA).

je preuzeo potragu za igračkom besmrtnošću i, uz manje ili veće probleme, došao od nekih zanimljivih rezultata koje bi rado podelio sa ostalim vlasnicima BBC-ja i „electrona“ kroz napise poput ovoga. Cilj tih napisa nije da nabrajamo POKE naredbe koje će vam подарiti beskonačno života ili ulazak u proizvoljan nivo neke igre (za one koje programiranje ne interesuje uradićemo i to na kraju ovoga teksta) — želimo, pre svega, da objasnimo kako se takvi POKE-ovi traže i nalaze i tako uputimo vlasnike svih računara u ovaj sport. Korist od ovoga je, dakako, mnogo veća — tehničko pretraživanje nepoznatog programa dobijaju svoj puni smisao tek onda kada pomoću njih počnemo da prilagođavamo ozbiljne komercijalne programe svojim potrebama

## Sa kasete na disk . . .

Stara poslovica kaže da bez alata nema ni zanata — za traženje beskonačnog broja života treba imati solidnu hardversku i softversku opremu, mada programerska veština može da dâ određene rezultate i bez nje. Bar jedna dika jedinica će učiniti da posao obavimo za deset minuta umesto za deset časova. Štampač će olakšati praćenje dugih disasembliраниh segmenata programa. Monitor program u ROM-u je praktično

neophodan i od njegovog kvaliteta neposredno zavisi uspeh operacije. Potreban je, najzad, softver i breinver (brainware) koji će izaći na kraj sa zaštitama koje su ugrađene u komercijalne programe na kasetama. Rezultati koji se objavljuju su dobijeni uz pomoć BBC B, 400 K disk jedinica, komercijalnog monitora programa Gremlin (Compucon Concepts) i Exmor II (Beebugsoft) i baterije programa za kopiranje DCOOPY domaće izrade.

Prva faza rada obuhvata prebacivanje programa sa kasete na disk i zahteva nešto znanja i dosta iskustva. Prvi problem je čitanje 'Locked' programa sa lažnim load adresama (jedna (pravda) load adresa u prvom bloku programa i druga (lažna) u preostalima). Pošto 'otključamo' program, reba da proverimo da li će on raditi u takvom stanju: neki autori su toliko lukavci da u toku inicijalizacije programa provere da li je ovaj pri uključivanju bio zaključan (po učitavanju zaključanog programa ćelije 33CA i 33DF sadrže 881, 8258 sadrži 03, segment 33BE-33C1 sadrži lažnu load adresu programa, a segment 33B2-33BD ime programa koje je, kod acoorsofthovih zaštićenih programa, obično neki kontrolni karakter (&O) koji se na ekranu ispisuje kao upitnik). Ukoliko program ustanovi da je učitao kao otključan, na ekranu će biti napisano nešto poput 'Software Protected' i sistem će se blokirati. Potrebno je disasembirati početni deo programa i pronaći mesta na kojima se ispituje odgovarajuća zaštita, pa testovima promeniti smisao. Sledeći pravac prebacivanje na disk, koje može da stvori daleko više problema od slične operacije na drugim računarima. Memorijaska mapa BBC-ja se, na žalost, bitno menja po priključivanju disk jedinica, pa se, "kaseni" programi moraju relocirati da bi radili. Ovim operacijama ćemo se baviti nekom drugom prilikom, pošto želimo da ovaj tekst bude od što veće koristi vlasnicima raznih računara koji žele da se upuste u potragu za besmrtnošću. Pretpostavićemo, dakle, da posedujemo igru na disku koja radi kada otkucamo "RUN ime.

## Izgubljene bitke ...

Najjednostavnija potraga koja će dovesti do rešenja u 90% slučajeva je traženje instrukcija tipa LDA #broj#života. Ukoliko u nekoj igri u startu imamo 3 života, pretražimo ćemo program (tu će operaciju, naravno, sprovesti naš monitor) i pronaći sve instrukcije LDA#3. Pošto notiramo njihove pozicije, prepravićemo prvih nekoliko ovakvih instrukcija u LDA#4 i startovni program. Postoji nekoliko mogućnosti — ili će se program trenutno porušiti (ova mogućnost je daleko manje verovatna nego što to može da izgleda; pronašli smo besmrtnost u petnaestak igara a da nam se takav slučaj nije dogodio) ili će raditi manje više normalno, pri čemu ćemo i dalje imati tri života ili će, u najpovoljnijem slučaju, pokazati da imamo četiri života. U poslednjem smo slučaju locirali grupu instrukcija od kojih je jedna ta koja postavlja broj života. Učitavaćemo program nekoliko puta, menjati jednu od fiksnih instrukcija LDA#3 u LDA#4 i startovni program sve dok ne pronađemo naredbu koja je odgovorna za broj života. Disasemblićemo njenu okolinu i svakako naći nešto poput:

LDA#3  
STA 8404

Ovakva sekvencna označava da memorijaska ćelija 8404, "čuva" broj života. Sada smo već vrlo blizu rešenju problema — pretražimo program i naći naredbu DEC 404 (ili LDA 8404 CBC #? ). Iza nje, svakako, sledi neka naredba tipa BNE dalje ili BEQ kraj. U prvom slučaju treba da zamenimo DEC 8404 BNE dalje za JMP dalje NOP kraj, a u drugom sekvencu DEC 8404 BEQ kraj sa šest NOP-ova. Posle ovoga je vrlo verovatno da ćete imati onoliko života koliko želite.

Ukoliko posle zamene jedne grupe LDA #3 instrukcija program i dalje daje tri života, pritisnemo BREAK i nastaviti sa zamena drugih grupa. Ako se desi da iscrpimo sve LDA 3 instrukcije bez efekata, izgubili smo prvi gem.

Sledeće pretraživanje sprovodimo za instrukcijama tipa LDA#2 obzirom da je moguće da se početni život podrazumeva. Njih, kao i u prethodnom slučaju, zamenjujemo sa LDA#3 i startujemo program, nadajući se uspehu (sada vidimo zašto je neophodno imati disk — uključivanje programa sa kasete za svaku probu bi izludilo i najzagriženijeg hekerja). Ukoliko izgubimo i drugi gem, pokušaćemo da tražimo instrukciju LDX#3 odnosno LDY#3 i da ih zamenjujemo sa LDX#4 odnosno LDY#4 respektivno, nadajući se da je autor programa iz nekih razloga pretpostavljao indeks registra akumulatru. Ukoliko ni ovakva potraga ne da rezultate moraćemo, da ostanemo kod teniskog rečnika, da se pomirimo sa gubitkom seta.

## ... i izgubljeni ratovi

Ako dođete do ove tačke, najbolje je da odustanete od uzaludnog posla i zaboravite igru — dalje rad je težak i mukotupan i ne garantuje uspeh, osim u slučajevima da imate zaista mnogo iskustva u problemima ovoga tipa. Potrebno je da manje ili više shvatite funkcionisanje određenih delova programa, obračunajući posebnu pažnju na inicijalizaciju i ispisivanje programa, obračunajući posebnu pažnju na inicijalizaciju i ispisivanje poruke Game Over. Ukoliko nađete segment naredbi koje ispisuju 'Game Over', pokušajte da otkrijete odakle se dolazi do njega i tako rešite problem. Ukoliko vaš monitor ne može da nađe string „Game Over“, vrlo je verovatno da ga autor programa negde sakrio (npr. povećao ASCII kodove slova za jedan), da se sprječio ono što upravo radite. Podstrek za dalje rad može da pruži jedino naša nacionalna osobina nazvana inat — autor ovoga teksta neće skoro zaboraviti četiri dana teškog rada koja su mu bila potrebna za pronalazjenje POKE-a za beskonačno života u izvrsnoj igri „Eagle Empire“.

Nakon što, uz manje ili više napora, pronađete put ka besmrtnosti, možete da snimite program na disketu tako da po normalnom učitavanju normalno radi, a da igrač dobija beskonačno života ukoliko u toku učitavanja drži pritisnut taster SHIFT ili TAB. Pošto ste utrošili nekoliko časova na traženje besmrtnosti, možete da utrošite i nekoliko minuta na njenu eksploataciju. Vrlo je verovatno da će vam i igra i besmrtnost brzo dosaditi i da ćete poželeti da na ovakav način „razvalite“ neku drugu igru — to traženje može da bude bolja zabava od svake igre!

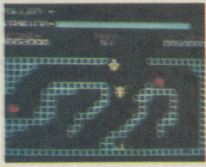
DA se  
STA 8404  
hajte da se  
igramo

„Spektrum“

ANTICS

ANTIKA

Antica, moramo priznati, neodoljivo poseđa na program „Birds & Bees“ koji se na našoj bogatoj igraćoj trpaži našao još u toku prošlog leta — to, u stvari, jeste nastavak ove proslavljene igre. Dakle, porovo ćemo se susresti sa malom vrednom platformi koju, ustovno, možemo nazvati Maja (svaka sličnost u daljem tekstu sa crtanim filmom „Rojstva Maja“ je namerna). A kada smo već kod ustovnosti, onda treba reći da će ona samo ustovno biti vredna, jer to, jasno, zavisi isključivo od vaše spremnosti (koja se povećava sa svakim satom igračkog sata). Treba, na samom početku, ponoviti i odlično urađenu muziku — šabnje zvudjenje svira poznate Barrove fuge. U drugom (proširenom i dopunjenom) izdanju, radnja se pronađe i na desetine igračkih zemlja, sa, pogledate vse, gomilom najrazličitijih hodnika, popunjenih još većom gomilom ispravica (obusavara, mraza, stonog... ) kojima se, izgleda, nimalo ne dopada vaš podzemni ter, pa će se svojizi truditi da on bude što kraći. A vaš zadatak je da pedantno zavirite u svaku prostoriju i opravite sve ovetve na koje naiđete: ako to obesmišlamo kao ispravu, onda bi vam igrači zadatak bio da pronađete Pavu i znesete ga na površinu. Pazite, možemo čitati su meksikan i nevidjivo na prvi pogled imajte na umu da autori programa nisu predviđali da se u toku Majing leta čuje zvudnje — stoga, ako vam se bude desilo da čujete zvudnje, budite sigurni da ste malo preterali sa igranjem.



„Spektrum“

Ser LANCELOT

SER LANSELOT

Evo, pravo, iz smrdnjeg sveta, malo podmanat i dotiran, u punoj ratnoj opremi, sa mačem i oklopi, na vaše male ekrane uskeke simpatični vitez iz ženskog sveta auditorijuma čije se uspeh dozvolilo mi da ga predstavim. Ser Lanseloti igra može da počne. Ser Lanseloti kreće u akciju bogovanja svoje nemoć najraznolikijim zlatnim predmetima. Očigledno je nešto naravno. Da, da, da, svo mač mu se sve vreme... trasa levo-desno. Sada, međutim, primjećujem da je u stvari, i kao Ser Lanseloti naravno, jer u drugu ekranu nemamo duksa nekoliko







ono što dolazi kasnije. Po nekima najverovatnijim proračunima, na ruru je trebalo savesti rati! Tu već vidimo savršeno oružje — avione, u vrednosti prodi onih jednih 15 kilometara ili oko 30 sekundi voljne. Ako preživite treći savesti rat, dolazi rat nuklearni koji imovni mogućim izgledom oduševljava onog ko ih stvora. Kada prođe mutanta, nalazi na prvom problemu, a posle toga i ono što je cilj igre — ali to ostvare samo za nam javite. Inače, kada startujete igru, prvi odlučujuće iznarednje. Betozenovno simfoniju odstranjuju onako kako to samo "komodor" ume i može. Posle toga dolazi drugi uređena garancija i avioni koji u početku izgledaju interesantni, ali čete ih od toga verovatno otkloniti. Nemojte se čuditi ali se nakon ovog naglo obratite i zakucava za zemlju — to se često dešava. Sve u svemu, jedna prava avantura za one prave, uporne igrače, koji žele da svoju igru dovedu do kraja.

**"Komodor 64"**

**BRUCE LEE**

**BRUS LI**

Svako od nas je ogledalo bar jedan film sa Brusom Lijom, bar čete je bilo mladi. Sada se u priči sa i igrati i oporaviti u kung-fu borbi, makar i samo otklopikom. U ovoj odličnoj uređenoj igri, vi ste Brus Li koji treba da uništi zlog čarobnjaka. Na nepokrepanje vam je nekakvo više udaraca — možete udarati rukom, nogom i lakozavrnim "magari" udarcima, koji je i naučitelji) to više neprijatelji (kojih ima dosta). Uz to, možete vrlo teško. Napada vas više vrsta neprijatelja — ma imi sa garinom mišom u rukama, ili debeli završni(?) karate borci koji su još opasniji od svih ostalih. Časim toga, da bi igra bila interesantnija i u igri dodati su odlični "trick" koji više ne vide strane (probate da skočite na neki, a uz njih i laseri, protiv kojih ni jedan Brus Li nije mogao da se bori, ali u vi ste, svakako, bolji borac od njega. Uz to, pričašete se kada se izložen uz illova — oni se obično zaveravaju oltim noževima, igra ima pravo neograničeno mnogo života, tako da čete teško proći sve sosa — iako čete naći na naku, nosu. Sve su vrlo različite, uređene dinamik grafikom u orijentalist stilu. Pored toga, vrlo je interesantno što igru mogu igrati i dva igrača u isto vreme, jedan se bori glavom, drugi rukama. Igra je vrlo zanimljiva, koji je va većno juriti kroz razne sote. To ovi odlični igru čini još interesantnijom za sve one "borce" koji imaju konkurenciju u nekom drugu, izratu ili komadi.

**"Komodor 64"**

**GRUDS IN SPACE**

**SVEMIRSKI TRANSPORT**

Još jedna od mnogih avantura, ali ova puta u svemiru. Vama, kao pilotu vasionog broda, siže poruka iz centra na našem trupu na Plutonu nedostaje gorivo... Mi vam ne možemo reći, ali čete dobiti milion dolara ako sa Saturna odnesete gorivo našim trupama na Plutonu. Milion zelenih — zvuči primamljivo, treba probati. Kroz pluća šasa, šestice za računarsom toznom izdubljeni, pokušavajući da se setite u kojoj ste pećini videli novčić na zemlji, ali se niste potrudili da ga uzimate (sajna lova). Sada ste u radnji, koja ima sve stvari koje su vam potrebne, možete ih kupiti, ali čete platiti. Krenite iz pećinica i skupljajte što što možete poret. Na početku igre, sa ekranu va više misati jednog divni mali zidovi sa još lepšim osmehom. Njega ne svesi još koji put. U većim vesionarima brod pro pronadite sote za komunikaciju, da biste primili poruku. Posle toga nadite "vindor" koji vam dodaje kao neki teleskop iz nekak toznu. Pravi problemi počinju tek na Saturnu. Priznajte odno da vas teleskop ne izbacuje još ne treba, jer de vam računari javiti da ste se smislili u izdatom svemiru, što nije baš tako izgledno. Vidite računari o svim stvarima ili navodnim stvarima — one obično oduševljava igra, zauzima oko 170 k i ima ogroman broj šasa koje su zaista savršeno uređene. Sa svojim vesionarima, igra donosi više od prvotne sve što vam padne na pamet. Probajte da kucate razne gruposti — računari de vam ukoliko nešto dubovito odgovoriti. Sve u svemu, ova avantura vam daje dosta zanimljivih mogućnosti isprobate svoje strpljenje i zaista joj vredi posvetiti nekoliko dana, pa i noći.

**Realizovali oglasni omet**



**SOFTWARE**

**izuzetna prilika po nabavnoj ceni nove, originalne knjige koje nijednom nisu kopirane**  
**Commodore 64 Programmers Reference Guide (486 strana)**  
 Liveness! 8060/8085 Assembly Language Subroutines (480 strana)  
 Colorful Commodore 64 Cookbook (250 strana)  
 Zaks: From Chips to Systems (550 strana)  
 Zaks: Microprocessor Interfacing Techniques (450 strana)  
 Zaks: CP/M User Guide (400 strana)

**knjige se prodaju samo u jednom primerku**  
 Ljubišta, 51/1558-007 (Petra Lekovića 57 11030 Beograd)

**COROS**  
**COMMODORE ROYAL SERVICE**  
**ŠTA ODLUČIJE COROS?**  
 — PRISTUPAJUĆE CENE (80—200 DIN)  
 — BRZ ROK ISPORUKE (1—3 DANA)  
 — 100% SIGURNOST ŠNIMKA  
 — BESPLATAN KATALOG  
 — OD NEKADANJI I NOVOSTI NA NAŠEM TRŽIŠTU: KARTIDŽI ZA OHLASNI RAD SA KASETOFONO- NIM, DISKOM, KARTIDŽI ZA KOPIRANJE I MONITOR

**OMRČEN DAMIR**  
**BULEVAR JNA 64/10**  
**11000 BEOGRAD**  
**011/682-044.**

**BYTE SHOP**

**package software**  
 Momentalno najpopularniji način nabavke programa — paketi programa. Dve su bitne odlike ovakvog načina nabavke: izuzetno pristupačne cene i brz rok isporuke. Nazovite nas i veš istog dana programi de brzinom umu, poštom krenuti u vaš dom!  
 1) 50 VELIČANSTVENIH...3990 din. Kupili ste nedavno Spectrum? Nije li najbolje da vašu kolekciju programa razproširite upravo sa onim najboljim (čije opise možete naći na stranicama Računara 8)  
 2) 10 SAHOVA+4 LOGIČKE IGRE...300 din.  
 3) 50 BASIC PROGRAMA...1500 din. Idealno za početnike! Igra u mogućnosti neposrednog učenja bazičica (Svaku igru možete prekinuti, analizirati a možda i poboljšati).  
 4) 5) 6) 16K PROGR...1500 din. svaki Vlasnik Spectruma 16K Byte shop misli na vas i nudu vam u tri paketa od po 41 program svaki. Jasno, moguće ih je koristiti sa spectrumu 48K, a od sada i većim sistemima i ostali programi u paketima po 14—16 programa za 1000 din. Možete ih i pojedinačno naučavati. Gore navedene cene uključuju cenu kasete i poštarine. Tražite novi besplatni katalog paketa.  
**OMRČEN DAMIR**  
**BULEVAR JNA 64/10**  
**11000 BEOGRAD**  
**TEL 011/662-044**

**DIAMOND-SOFT**  
 Vam nudu fantastičan izbor od preko 700 programa najboljih svetskih firmi za vaš "spectrum". Jedinstveni podaci najvećih hitova sa svetskih top lista među kojima izdajavamo:  
 nonn cresta, bruce lee, konn & s. hunteback, turmoli bangna nas eggs, airwout, technican, det plutone, lode runner 1 i 2, i još mnogo noviteta do izlaska ovog "računara".  
 Javite se pišom ili telefonom za najnoviji spisak ili katalog programa (30 din) na adresu: Veselino Dord, M. Gorkov 6, 21000 Novi Sad, tel. 021/20-742

**SVE ZA ZX SPECTRUM**  
 — video i logičke igre  
 — namerski programi  
 — knjige i prevodi  
 — uputstva za igre  
 — uputstva za namerske programe  
**GARANCIJA ZA SVE VRSTE USLUGA I I**  
**LITERATURA ZA SINCLAIR QL:**  
 — "INTRODUCING THE SINCLAIR QL" cena 1500,- din.  
 — "ADVANCED PROGRAMMING WITH THE SINCLAIR QL" cena 1500,- din.  
 — "PROGRAMMING THE M 68000" cena 1200,- din.  
 — "M 68000 16/32 BIT MICRO-PROCESSOR" Motorola cena 1800,- din.  
**VELIKI IZBOR LITERATURE ZA COMMODORE 64**  
 — BBC  
 Informacije i katalog možete dobiti kod:  
**MILOVANOVIĆ LUBIŠE**  
 Petra Lekovića 57  
 11030 BEOGRAD  
 tel: 011/658-007

**Direktno iz Engleske**  
**MEGASPECTRUM**  
**PRETVORITE VAŠ SPECTRUM U MEGASPECTRUM**  
 — 50 novih BASIC naredbi  
 — 76 novih I/O naredbi  
 — uvećane naredbi slovo po slovo (single letter)  
 — tri oblika slova  
 — tri veličine slova  
 — prozori u stilu QL-a  
 — imenovane procedure  
 — strukturo programiranje  
 — izboljšane branja ...  
**DD STUDIO**  
 Vrnjačka banja  
 19 April 5  
 telefon 036/64-619

**SPECTRUM HOUSE**

ZX specijalisti pripremili su za vas ekskluzivni DE LUX katalog — profesionalni vodič kroz hitove i klasične programe. U ovom enciklopedijskom katalogu — poklon program meseca (eng. listing) i semu light pusa sa uslobom 700 i više programa 100 din, za katalog vauca se pri prvom narudžbi.  
**VRCA MILAN, Zarija Vučević 79, 11000 BEOGRAD**

**\* NEW DATA \***

**KORAK DALJE SA SPECTRUMOM!**  
 Sopotvni programi za stručnjake: STATISTIKA, LINPROG, TRANSPORT, PERT.  
**NOVOSTI: GEOZEJUA — 12 programa.**  
**NEW DATA, D. Brašova 8/10, 21000 Novi Sad**

**SPECTRUMOVCI** — Specijalno postolje za vaš kompjuter od kvalitetne plastične mase koje omogućava strujanje vazduha i odvodnje njegove suvine toplote sa vašeg SPECTRUMA, pri čemu tastatura zauzima funkcionalan negib, za samo 1000 dinara. Dragun Dejko, ul. Beograd, Ljube Didića 40/4, tel. (011) 768-505.

**Spectrum 48K** — veliki izbor klasičnih novih i najnovijih programa, povojno na vašim liš nam kasetama. Tražite besplatni katalog, spisak literature. Bajic Goran, Stevana Filipovića 29/85, 11040 Beograd, 011/653-2685

• **Amstrad CPC 464** — veliki izbor najnovijih programa i literature. Tražite besplatan katalog. Češić Čip, Amruševa 7, Zagreb

**•• ATARI**

Veliki izbor programa. Besplatan katalog. Lamacanović Dejan, Sindeliceva 31/A, 23000 Zrenjanin, tel. 023/31-665

• **Amstrad — schneider i q Sinclair** — listinzi, testovi i informacije o softveru i hardveru. Sava Bošković, Vlade Živanovića 6, 21000 Novi Sad

**AMSTRAD**

Amstrad user club Beograd YU Prvi registrovani kompjuterski klub u Jugoslaviji nudi vam: Internacionalnu člansku kartu. Slobodan pristup u "Amsoft" — ovu banku podataka. Mogućnost preplate i uvid u "Amstrad user magazine" Novosti iz "Amsoft" produkcije Nabavku najnovijeg softwera — Nevrugite! — Proverite! Amstrad klub, G. Vučića 182/II, 11000 Beograd YU tel. 011/425-180, 425-181, 419-316

• Veliki izbor programa i gotovih "paketa" za komodor-64. Popusti. Katalog 011/135-950, Novak, li bulevar, 23/26, Novi Beograd

Imam samo najbolje i najkupije programe. Javite se radi kupovine ili razmene branislav Boždžević, J. Gagarina 206 11070 Beograd tel. 011/156-444

**KOMODORCI:** Od danas sve na jednom mestu: veliki izbor kvalitetnih programa, prevedena literatura, saveti, katalog, ANĐEŠIĆ ZENKO, Drugi Belevajr 34/53 11070 Novi Beograd, Tel. 011-131-641

**COMMODORE 64**  
 Veliki izbor literature i najkvalitetnijih programa  
 — štampane skripte — stručno prevedena literatura  
 — UPUTE uz programe  
 — sistemski i namerski programi, nove igre  
 Kvalitetno snimljeni programi uz GARANCIJU i prema vašem izboru!  
**BESPLATAN katalog** s opisom svakog programa  
**ENENAD RADOSAVLJEVIĆ, Ovljete Zuoćin 39/6, 41000 ZAGREB**

**GALAKTIČKI Navište "slapo"** — kucanje i izmeravanje sa sofisticiranom inteligencije novog obzičnih programa ne računju osnovi. Uz to i veliki izbor drugih igara. Informacije i katalog: 011-330-591, Aca Zelenko, Lenjinov Bulevar 139, Novi Beograd

# rom Računari u domaćoj radinosti

# od sedam miliji

## Spektrum

Problemi ovakve vrste obično se rešavaju raznoraznim „eprom loaderima“ i „kardidžima“, ali takva rešenja obično donose samo polovične rezultate — program u „loaderu“ se učitava svaki put kada se računar uključi ili resetuje, dakle i onda kada je taj program potreban i onda kada nije. Ljubitelji računara, osim toga, u početku izgleda sasvim dovoljno da u eprom stavljaju samo „svoj najomiljeniji program“, ali apetiti brzo rastu i skupi loaderi počinju da se nižu jedan za drugim. Polovično rešenje je, po pravilu, uvek skuplje rešenje. U želji da se preduprede iznenađenja ovakve vrste, razvijen je operativni sistem za rad sa alternativnim ROM-ovima koji, iako ima jedva nešto više od 400 bajtova, obezbeđuje veoma komforan rad:

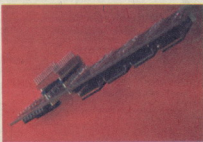
- program miruje u računaru sve dok se ne pozove sa tastature
- program se poziva iz bezjeka po svom imenu („MONS“, „GENS“, „WORD“, „EPROM“)
- programi se izvršavaju automatski ili sa unapred dodeljene adrese („GENS“) ili sa adrese koja se navede uz ime programa („GENS 33000“)
- poziv programa može biti „hladan“, kada zaboravlja sve svoje parametre i promenljive, ili, ako se stavi zvezdica iz imena programa, vruć, kada program nastavlja tamo gde je stao prilikom poslednjeg poziva
- pozvani program ne zauzima ništa, više memorije nego što bi zauzeo da se učitava sa trake.

### Hiljadu „kako“

U računaru se, bez ikakvih dopuna u softveru, može nalaziti do 256 eprova od 8 ili 16 kilobajta, razume se uz nešto malo jači hardver. Komercijalni programi se stavljaju u eprom bez ikakvih izmena, pri čemu broj programa u jednom epromu nije bitan — u njega se može staviti jedan ili više programa bez obzira na njihov tip: bejzik, mašinska, ili kombinacija i jednog i drugog. Programi koji ne mogu da stanu u jedan, mogu se završiti u bilo kom drugom epromu.

Mada „Računari 6“ izlaze prvog juna, ovakve specifikacije mogu nekome da liče na zaksnelu — prvoprilisku šalu. Kako je moguće da programi koji su predviđeni da rade iz RAM-a bez ikakvih izmena rade iz ROM-a i kako su u „spektrumu“ nagurani svi ti silni kilobajti, kada je njegova memorijska mapa popunjena do poslednjeg bajta?

Nijedan program u pomoćnom ROM-u ne radi, razume se, iz samog ROM-a — prepravka bilo kog komercijalnog programa da radi iz ROM-a ogadila bi život čak i



Ne više ni tako skupo: „Megarom“ za „spektrum“ od 64 kilobajta danas ne bi trebalo da košta više od 20000 dinara

najokorelijem hakeru — nego se, na poziv sa tastature, preslikava iz ROM-a u RAM i, odmah zatim, kontrola prenosi na njega. Operativni sistem, u suštini, tretira programe u epromu kao da se nalaze na traci. Jedina razlika je u načinu na koji se pozivaju (samo ime), brzini kojom ih pronalazi (u najgorem slučaju nekoliko desetina milisekundi), tehnički kojom ih učitava (LDIR) i — strahovitij brzini. Nakon pozivanja GENSA, karakteristični „Hajsoftov“ znak se javlja za manje od jedne sekunde (samo učitavanje traje sedamdesetak milisekundi), praktično ni gore što korisnik skloni prst sa tastera ENTER!

Rad operativnog sistema je, u osnovi, veoma jednostavan. Kada „spektrum“ dobije naredbu koju ne poznaje, prosljeđuje je novom operativnom sistemu. On isključuje osnovni ROM i pretražuje jedan po jedan eprom u potrazi za programom koji je pozvan. Ako ga pronađe, uzima iz zaglavija sve potrebne parametre, preslikava program u RAM, uključuje osnovni ROM i prenosi kontrolu na „učitani“ program. Ako u alternativnim ROM-ovima ne postoji ono što se traži, vraća „spektrumu“ lopticu sa primedbom „nepoznat“ i „spektrum“ prijavljuje sintaksnu grešku.

Po svojoj organizaciji, sistem je znatno bliži nečemu što bi moglo da se nazove eprom-disk nego onome što Sinkler naziva „bočni rom“ i onome što je ugrađeno u računare BBC. Svakom programu u epromu prethodi zaglavije sa ličnom kartom programa. U njemu se nalaze svi potrebni podaci za pozivanje, učitavanje i startovanje programa. Zaglavije zauzima 22 bajta za čisto mašinske programe i 30 bajtova za kombinacije mašinskih programa i bejzika, kao i u slučaju kada se program u jednom epromu nastavlja i u drugom. Organizacija zaglavija se, jednim delom, sliče bajt na bajt sa standardnom strukturom zaglavija u „spektrumu“. O organizaciji zaglavija pisaćemo u sledećim „Računarima“.

I druga dilema — kako uklopiti alternativne ROM-ove u prepunu „spektrumovu“ memorijsku mapu — ima takođe jednostavno rešenje. Osmobitni procesori (ima ili

nekoga da to ne zna?) mogu da adresiraju samo 65536 memorijskih lokacija. Memorija se, međutim, može organizovati na razne načine, između ostalog po tzv. stranicama. Kompletan adresni prostor jednog mikroprocesora može da se smatra jednom stranicom u knjizi od 65536 strana. Organizaciju memorije na ovakav način računari se može opremiti RAM-om neograničenog kapaciteta. Jedina je nevolja u tome što mikroprocesor može direktno da komunicira samo sa jednom stranicom — on ne ume bez pomoći programera da okrene list i pređe na sledeću stranicu i, što je još gore, potpuno zaboravlja sve što se dešavalo na prethodnoj stranici kada je toga nikada nije ni bilo. Malim softversko-hardverskim rezom mikroprocesor se, doduše, može navesti i da sam okrene stranice, ali samo da bi odložio ili prihvatio neki podatak — izvršavanje programa se ne može prenositi sa strane na stranu. Ovi nedostaci u ovakvoj primeni nemaju, međutim, nikakvog značaja.

Po stranicama se ne mora organizovati čitava memorija, već se to može učiniti i samo sa pojedinim njenim blokovima. Svi alternativni ROM-ovi smešteni su u „spektrumovom“ šesnaestercu, u prvih šesnaest kilobajta, na istom adresnom prostoru na kom se nalazi i osnovni ROM. Računar i dalje ostaje pod stalnom kontrolom osnovnog ROM-a, osim u trenutku kada je program iz RAM-a, osim u trenutku kada je program iz RAM-a.

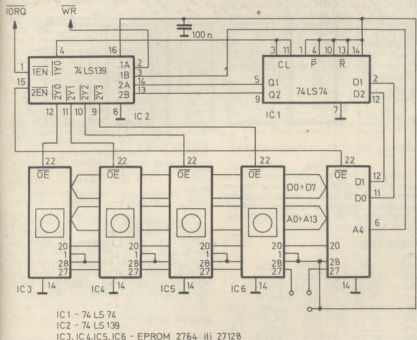
### Romski skretničar

Tehnika prelaska iz jednog ROM-a u drugi predstavlja, međutim, nešto drugačiji problem, koji se ne može rešiti bez pomoći hardvera. Svakim ROM-om, pa i onim u „spektrumu“, upravlja se pomoću dva kontrolna signala *CS* (izbor čipa) i *OE* (dovola izlaza). Da bi mikroprocesor mogao da pročita bilo šta iz ROM-a, oba signala moraju biti ništa. Osnovna ideja je u tome da se ovi signali preako pogodno elektronske skretnice usmeravaju prema onom ROM-u koji nam je trenutno potreban. Jednu ovakvu skretnicu predstavlja dekoader-demultiplikser 74LS139 koji je u stanju da niski logički nivo na svom ulazu *EN* distribuira u četiri različita pravca. Od stanja na adresnim ulazima *A0* i *A1* zavisi u kom pravcu će krenuti signal sa ulaza *EN* (tabela 2). Ako na ulazu nema signala, neće ga biti ni na jednom od izlaza što, praktično, znači da je sklop aktivan samo kada se mikroprocesor obraća ROM-u. Na svaki izlaz se može vezati po jedan eprom od 8 ili 16 K.

Elektronskom skretnicom upravlja kontrolni registar od dva bita koji je realizovan



Vreme „plovštine“, izgleda, nepovratno prolazi — velike kutije sa polupraznim štampanim kolima koja treba dopunjavati čitavog života ne interesuju više ni najveće zanesenjake. Računari najnovije generacije, koji su se pojavili na tržištu prošle i ove godine, donose, zaista, jednu sasvim novu koncepciju — oni, u specijalnoj primeni kao što je, na primer, obrada teksta, od svog vlasnika zahtevaju sve manje znanja. Novopečeni ljubitelji računara ne mora više da zna čak ni da učita program sa trake ili diska — svi programi koji bi mu mogli zabrati već se nalaze u ROM-u računara. Da li postoji mogućnost da se na sličan način napredni i popularni „spektrum“? Projekat koji objavljujemo omogućava da se „spektrumov“ ROM multiplicira u neograničeni broj primeraka i u njih smeste svi programi koji iole vrede — paskal, fort, C kompajler, bejzik kompajler, beta bejzik, tasvard i, razumljivo se devpak!



pomoću dva flip-flopa iz čipa 74LS74. Pomoću adresnog dekodera iz prve polovine čipa 74LS139, u kome učestvuju A4, WR i MREQ, kontrolnom registru je dodajena adresa 239 u ulazno-izlaznoj mapi. (Registar, zapravo, ima šesnaest adresa, među njima i 239, ali je izabrana upravo ta da bi se zaštitile relativne adrese koje koriste „spektrum“ i programator eproma za svoje potrebe). Uključivanje alternativnih ROM-ova postize se sekvencom.

LD A,n  
 OUT (239), A  
 pri čemu n predstavlja redni broj eproma i može imati vrednost 1, 2 ili 3, a povratka u osnovni ROM instrukcijama XOR A  
 OUT (239), A.

Prilikom uključivanja računara u kontrolnom registru se uvek nalazi stanje 0, što znači da prilikom buđenja računara kontrolno preuzima ROM koji se nalazi na poziciji 0. Na ovom mestu se, razumljivo, nalazi osnovni „spektrumov“ ROM, ali može da dođe i bilo koji drugi, pod uslovom da on preuzme na sebe inicijalizaciju računara. Kontrolni „slog“ ostaje u registru sve do

lako i jednostavno: Električna shema „margaroma“ za računar „spektrum“ — neobeleženi eprom predstavlja podnožje sa produženim nožicama preko koga se pločica priključuje na računaru umesto osnovnog ROM-a; nožica broj 27 na ovom podnožju može se povezati sa epromima na dva načina — preko jednog invertora (nisu potrebne nikakve prepravke u „spektrum“) ili direktno (linija koja povezuje nožice 27 se tada vuče na napon napajanja, a veza između nožice 27 i „spektruma“ obavezno prekida odsecanjem ili savijanjem pina)

upisa novog kontrolnog „sloga“, što znači da će jedan isti ROM biti uključivan sve dok mu se ne promeni adresa.

Od dva signala za upravljanje ROM-om, jedan može da ide direktno na sve ROM-ove, dok se drugi vodi na elektronsku skretalnicu. U ovom rešenju direktno se vodi signal CS. To, praktično, znači da se svi ROM-ovi nalaze u stanju pripravnosti (stand by), ali će biti pozvan samo onaj kome se, preko skretalnice, uputi i signal OE. Time se postize maksimalna brzina odziva.

Korišćenjem ove tehnike „spektrum“ se može proširiti neograničenim brojem ROM-ova. Dekoder 74LS138 omogućuje da se u sistem uključi osam ROM-ova, a 74LS154 čak šesnaest. Kontrolni registar, naravno,

#### ROM-OS CTRL CODE HEX LIST

38877:	217A3811C05C019293
38885:	01ED80C92A505C2B75
38893:	7EFE2AC0225D5CAFF0
38901:	32BC5C32B05CC42R4
38909:	5DFE84CA535DD021D7
38917:	01002A5D5CDD50177
38925:	23131A3C2D8143DBEC3
38933:	28F6DD7E133C28DEC6
38941:	DD5E11DD5612DD1987
38949:	18E0CD565D7EFE2A1E
38957:	CC215D225D5CFE0030
38965:	2804FE3A2017E1E15D
38973:	FDCB017E204D21B78C
38981:	12E5C3761B3EFF32BA
38989:	BD5C237EC9CD0821CEE
38997:	FDCB007EC8FE002841
39005:	03FE3A00E1E1FDCB85
39013:	017E201B180BF33AD7
39021:	8C5C3C32B05CC3EF60
39029:	C9F33ABC5C18F7223F
39037:	5D5CAFF3EFFBC9CDBB
39045:	991EED438A5CCD4D17
39053:	5D1813CD405DD07E5H
39061:	00FE023855DD6E00E5
39069:	DD6E0E228A5CCD6ED4
39077:	0F2CC23F5E212B2D13
39085:	E5DD6E095DD660ED73
39093:	5BB8A5C05D0D4E0B51
39101:	DD406C3AB05C3C28E6
39109:	02EDB03D32B05CCD094
39117:	7E114B7C4AF5DC0563C
39125:	5DC1C9D06E15DD668FA
39133:	16DD4E17DD4618DD70
39141:	5E19D0561AD3BEFD73
39149:	80C92A535CE5EB284C
39157:	595C2BDD4E0BDD4639
39165:	0CC0D5650DD220454
39173:	5BCDE519C1CD55161F
39181:	DD2A045B23C0405D00
39189:	DD4E0FDD4610092298
39197:	485CDD660E7CE6C01A
39205:	200AD7E00D22425C42
39213:	FD360A00D50E09DD6E
39221:	660AD13AB05C3C28F8
39229:	02EDB03D32B05CCD094
39237:	7E00FE012800C56D0
39245:	5D21761BE5C91500D3
39253:	1E0CDD19D06E09DD51
39261:	660ADD50E0DD5E0EF9
39269:	C3905D2DD6618225E
39277:	B2C36E2BF92B2BFC
39285:	22D35CC3815D00005C

Hexa-decni operativnog sistema za rad sa proširenim ROM-ov. Programi se unosi pomoću hexa-decitera koji objavljujemo na strani 33; u računar se unosi kompletna linija — prvih osam bajtova odziva u memoriju a deveti bajt predstavlja zbir prethodnih osam

tada treba da ima tri, odnosno četiri kontrolna bita (na primer, 74LS174). To, međutim, ne bi bilo sve — mikroprocesor u tom slučaju sasvim sigurno ne bi imao dovoljno snage da pobuđuje sve eprome i zato bi trebalo u magistrale za adrese i podatke ugraditi odgovarajuće drajvere (74LS245 za podatke i 74LS244 za adrese).

### Mali softverski rez

Opisana koncepcija mogla bi da se ostvari bez ikakvih izmena u osnovnom ROM-u. Na nultom položaju tada se ne bi nalazio osnovni ROM, već neka od alternativa. Njegov zadatak bi bio da izvrši inicijalizaciju računara, preslika samog sebe u RAM, pripremi „spektrum“ za prepoznavanje novih naredbi i vrati kontrolu osnovnom ROM-u. Svaki konstruktor, međutim, koji pretenduje da svoj računar proširi novim ROM-ovima trebalo bi, pre svega, da dovede u red onaj osnovni. U „spektrumovom“ ROM-u ima dosta posla za lovcu na bagove i dosta prostora za poboljšanje čak i da nije ostavljen prazan čitav jedan kilobajt (od 14446 do 15616). Svako, razume se, ima svojih ideja kakve bi prepravke trebalo izvršiti, ali bi se link za naredbe, bez sumnje, našao na vrhu svacije liste. Pošto je ROM već „otvoren“, u njega je ubačen i novi operativni sistem — čitava operacija ne bi imala mnogo smisla kada bi on morao da se učitava sa trake.

U osnovnom ROM-u „spektruma“ načine su, dakle, tri funkcionalne izmene:

- uveden je link za naredbe,
- pomeren je početak bejzika i
- ubačen je operativni sistem za rad sa alternativnim ROM-ovima.

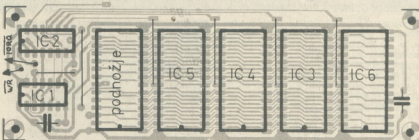
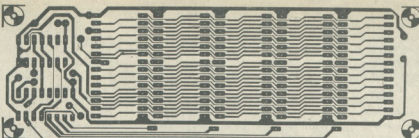
Operativni sistem je ubačen u osnovni ROM počev od adrese &387D, ali on ne može da radi iz ROM-a — prvom prilikom kada bi preneo upravljanje na neki od alternativnih ROM-ova, ukinuo bi i samog sebe! Zato se ovaj program prilikom inicijalizacije računara preslikava u RAM, ostavši, sa visine, rezbariva u „snesnaestercu“ računara. Gde, međutim, postavlja jednog ovakvog koordinatora. Sam vrh memorije može biti dosta bezbedno mesto, ali neki važni programi, kao što je beta bejzik, ne ostavljaju slobodan ni jedan jedini bajt. Jedino rešenje je bilo da se postavi ispred prostora za bejzik, naravno bez ikakvih REM linija. To je postignuto na sledeći način:

### izmena broj 1

**staro**  
1235 21 B6 5C LD HL,&5CB6

**NOVO**  
1235 21 46 5E LD HL, &5E53

Prilikom inicijalizacije računara, „spektrum“ koristi adresu &5CB6 za formiranje zone kanalskih informacija u RAM-u i svih sistemskih promenljivih na koje se bejzik oslanja u svome radu. Promenom ove adrese, zona kanalskih informacija i početak bejzik programa se mogu pomeriti za proizvoljan broj bajtova bez ikakvih posledica na kompatibilnost sa komercijalnim programima. Prostor između adresa &5CB6 i &5E53 (412 bajtova) ostaje potpuno slobodan da primi operativni sistem, linkove i nove sistemske promenljive. Isti ovaj prostor koristi i mikrodžaj za definicije novih kanala i bafera. Kompatibilnost sa mikrodžajom je, međutim, svesno žrtvovana već u hardverskom delu — adresa 239 pripada interfejsu 1.



Štampano kolo (1:1) i montažna shema

Formiranje linka za naredbe povereno je potprogramu koji je u ROM-u smešten na adresu &4AA. Na tom prostoru su, inače, konstruktori „spektruma“ zaboravili nekoliko bajtova iz ROM-a računara ZX 81 koji, razume se, ničemu ne služe.

### izmena broj 2

<b>NOVO</b>	5CB6	LINK_1	EQU	&5CB6
	5CB9	LINK_2	EQU	&5CB9
	04AA	CD DF OE	CALL	&0EDF
	04AD	3E C3	LD	A,&C3
	04AF	32 B6 5C	LD	(LINK_1), A
	04B2	32 C0 5C	LD	HL,&5CC0
	04B5	22 B7 5C	LD	(LINK_1+1), A
	04B8	3E C9	LD	A,&C9
	04BA	32 B9 5C	LD	(LINK_2), A
	04BD	CD 6E 38	CALL	&386E
	04C0	C9	RET	

### O istom trošku

U ovoj rutini, očigledno, ima i nekih drugih stvari. Ona, pre svega, formira dva linka za naredbe. Prvi je sistemski i nalazi se na adresi &5CB6, a drugi je potpuno slobodan i nalazi se na adresi &5CB9. U sistemski link za naredbe odmah se upisuje adresa novog operativnog sistema. Zahvaljujući tome nisu potrebne nikakve dodatne inicijalizacije. Potom sledi poziv potprograma sa adrese &386E. On se nalazi neposredno ispred proširenih operativnih sistema i ima jedan jedini zadatak — da ga preslika u RAM počev od adrese &5CC0(2374).

Odakle se, međutim, poziva ova potprogram? Iz rutine za inicijalizaciju „spektruma“. Gotovo pri samom kraju te rutine, na adresi &128B, nalazi se instrukcija CALL &0EDF — poziv potprograma za brisanje bafera za štampač. Zamenom adrese, na ovom mestu se sada poziva potprogram na adresi &4AA. Prva instrukcija u novom potprogramu je poziv rutine za brisanje bafera. Zatim se formiraju linkovi, kopira operativni sistem u RAM i, nakon povratka u glavnu rutinu za inicijalizaciju, prikazuje dobro poznata poruka.

### izmena broj 3

<b>staro</b>	128B	CD DF OE	CALL	&0EDF
<b>NOVO</b>	128B	CD AA 04	CALL	&04AA

Poziv linka za naredbe je ubačen u rutinu za obradu grešaka koja se kod „spektruma“ nalazi na adresi 8. Pošto na adresi &53, na koju se proizvava potprogram na adresi 8, ima nekoliko praznih bajtova, čitav potprogram je pomeren za šest bajtova naniže da bi se na adresi 8 ubacio jedno CALL LINK\_1(&5CB6).

### izmena broj 4

<b>staro</b>	0008 2A 5D 5C	LD	HL (&5C5D)
	000B 22 5F 5C	LD	(&5C5F), HL
	000E 18 43	JR	&53

<b>NOVO</b>	0008 CD B6 5C	CALL	&5CB6
	000B C3 53 00	JP	&53

0053 2A 5D 5C	LD	HL (&5C5D)
0056 22 5F 5C	LD	(&5C5F), HL
0059 E1	POP	HL
005A 6E	LD	L (HL)
005B FD 75 00	LD	(Y+O), L
005E ED 7B 3D	5C LD	SP, (&5C3D)
0062 C3 C5 16	JP	&16C5

Modifikacija „spektrumovog“ ROM-a ne bi trebalo da predstavlja teškoću čak ni programerima sa veoma skromnim iskustvom. Pošto će se u ovaj projekat, verovatno, upustiti i pravi poletarci, dajemo, u najbruljivim crtama, i ovu proceduru. Pomoću MONS-a i njegove naredbe I (=inteligentno kopiranje) kompletan ROM se, najpre, preslikava u RAM, na primer na adresu &8000 (32768). Lokacije u koje treba uneti izmene nalaze se sada na adresama koje su za &8000 više od originalnih — &8008, &8053, &84AA... Korišćenjem opcije za direktno unošenje sadržaja (M:8008/2A/ENTER/5D/ENTER) u memorijske ćelije, sve potrebne izmene se unose bajt po bajt. Operativni sistem, ranije unet pomoću heks-loadera, treba po završenom

poslu učitati sa LOAD " " CODE 47214 (&386E + &8000). Upisivanje modifikovanog ROM-a u eprom vrši se sa adrese 32768. Trud uložen u gradnju programatora epro-ma dobija, u ovom trenutku, svoj puni smisao — ljubitelji računara preuzima apsolutnu kontrolu nad svojom mašinom.

### Klasičan hardver

Hardversko rešenje je klasično za ovakvu vrstu proširenja. Mi smo, da bi se uštedelo trud oko dizajniranja štampanog kola, modifikovali jedan projekat iz časopisa „Elektron“. Detalji oko povezivanja sa računarom ne zaslužuju naročitu pažnju — njih će svaki vlasnik „spektruma“ rešiti na svoj način. Možda je najbolji način da se izvadi originalni ROM i umesto njega postavi pločica sa četiri epro-ma (detalji ovog rešenja vide se na fotografiji). Sve potrebne veze ostvaruju se preko kontakata u podnožju — nedostaju samo signali WR i IORQ koje originalni ROM ne koristi. Ovakvo rešenje je moguće na starijim verzijama „spektruma“, ali je već od verzije 3 nešto teže izvodljivo, jer čitav prostor iznad čipova, kuda bi, inače, prolazila pločica, zauzima hladnjak. Oni koji ne žele da previse rovale po svome „spektrumu“, mogu jednostavno da isključe CS na originalnom ROM-u, a čitav sklop povežu preko konektora.

Epromi od 16 kilobajta su potpuno kompatibilni sa ROM-om u „spektrumu“ osim u jednoj sitnici — na nožicu 27, koja inače služi za programiranje, treba, preko jednog invertora, dovesti signal MREQ. U principu, ne bi trebalo da bude nikakvih problema da se taj ulaz veže jednostavno na +5V, ali, iz dosta misterioznih razloga, neke verzije „spektruma“ ne podnose ovde statično vezane već zahtevaju dinamičan signal. Umesto tipa 27128, na svim mestima se može koristiti i tip 2764, ali, razume se, sa dvostruko manjim kapacitetom.

#### Kontrolni bitovi

kombinacija	uključuje osnovni rom
00	rom 1
01	rom 1
10	rom 2
11	rom 3

Uz minimalne izmene, umesto poslednjeg epro-ma može se staviti statički RAM koji, uz poseban program, može da radi kao tzv. ram-disk. U njega se, pre testiranja, mogu sklanjati mašinski programi — na primer, izvorni kod iz GENSA-a. Broj fajlova u RAM-u je ograničen samo kapacitetom čipa — u memoriju tipa 6264 može da stane čitavih osam kilobajta izvornog koda — a poziv bilo kog programa je trenutak. Krahiranje programa u razvoju, dakle, više ne predstavlja nikakav bauk — u računaru se uvek nalaze ne samo asembler i debugger nego čak i izvorni kod! Operativni sistem za ram-disk radi direktno iz GENSA-a. O ovome, međutim, možda nekom drugom prilikom.

Neko će pomisliti da nakon ovoga od „spektruma“ ostaju samo mikroprocesor i memorija. To — i nekoliko zaista dobrih komercijalnih programa koji na „spektrumu“, zapravo, najviše i vrede.

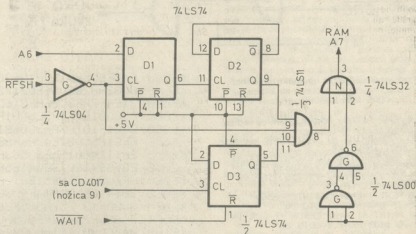
Jova Regasek

### 63) fini hardver za finu grafiku

# fini hardver za finu grafiku

## Računari u domaćoj radinosti „Galaksija“

Za „galaktičare“ koji su napravili memorijsko proširenje i generator zvuka izrada „grafičkog hardvera“ neće predstavljati veći problem — uz dodatak samo četiri TTL integrisana kola „galaksija“ će naučiti da crta sa 53248 tačaka! Hardversko rešenje je do banalnosti jednostavno — konstruktori će imati znatno više posla oko prilagođavanja „galaksije“, memorijskog proširenja i, eventualno, generatora zvuka, ne ako same grafike! Sasvim primereno računaru „galaksija“, softver i ovde igra ključnu ulogu.

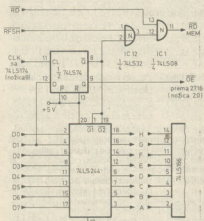


N - kolo neiskorišćeno u "Galaksiji"  
G - kolo se nalazi u "Galaksiji"

### Modifikacija memorijskog proširenja

Prilikom rada na razvoju fine grafike primećeno je da se sadržaj dinamičke memorije ponekad kvari, a generator zvuka „peva“ i onda kada ga niko ništa ne pita. Problem sa dinamičkom memorijom 4416 nastaje kao posledica nemogućnosti procesora Z 80 da je pravilno osvežava. Z 80 ima interni R registar i brojač za osvežavanje dinamičkih memorija dužine sedam bita, dok je za osvežavanje RAM-a 4416 potrebno osam bita. Procesor osvežava memoriju pri prihvatanju svake instrukcije. Ova činjenica je vešto iskorišćena na „galaksiji“ iz spisanje znakova, kao i crtanje slike visoke rezolucije. Tekst, odnosno grafička memorija se čita za vreme perioda osvežavanja posredstvom R registra i interapt registra I, čiji se sadržaji tada nalaze na adresnom vodovima procesora. Sadržaj R registra se pojavljuje na adresnim linijama A0 do A6, A7 je na logičnoj nuli, a sadržaj I registra se nalazi na vodovima A8 do A15.

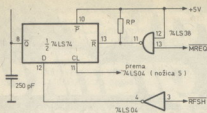
Početne vrednosti R i I registra zadaju se programski u zavisnosti od položaja teksta, odnosno grafičke memorije. Vrednost adresnog bita memorije A7 postavlja



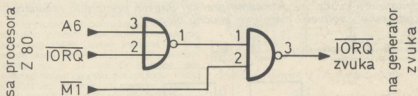
N - neiskorišćeno kolo na memorijskom proširenju

### Modifikacija računara „galaksija“





Sl. 3 — Električna shema grafičkog proširenja



Sl. 2 — Modifikacija generatora tona

CTRL CODE HEX LOADER

```
10 WORD &2A6A, &4000:PRINT "STAR
T ADDR:"; INPUT A
20 HOME:PRINT AT 44B;"ADDR "JA;A
T 44B; INPUT X#
30 S#0:FOR I#0 TO 8:Y#BYTEX I#2+1
65B9)-4B:Z#BYTEX I#2+10630)-4B:W#
(Y-X)(Y-Z)I#1#4Z-7X(W-Z) IF I#8
S#M#BYTE (A+1), W
40 NEXT I:C#S-256#INT(S/256)
50 IF C#M#A#B#GOTO 20
60 PRINT AT 44B;"ERROR I:"FOR I#
1 TO 1000:NEXT I:GOTO 20
```

\* HI RES GRAFIKA \*  
CTRL CODE HEX DUMP  
START:57344 LEN:1977

```
%E000: 3E00E73EC332AC2B3B
%E008: 2B292B2172E122AD49
%E010: 362174E722A82B28C8
%E018: 6A2A112000ED522226
%E020: 622AFD02131E40617E4
%E028: 36002310FB1134E58E
%E030: 722B732B2B2B2B2E7
%E038: 36BF09F105CD61E496
%E040: 2323237E3C2284E28C3
%E048: 6A2A7DE6E06F110057
%E050: 1AED52ED5B382CD7DC
%E058: DA5401226A2CD26D8
%E060: E023236FF1100D079
%E068: 19E57DCB7F1E022005
%E070: 02CBFB6056F3EDB2B
%E078: 060F0E01B02808D6E7
%E080: 20CB00CB0118F5E6A8
%E088: 7FE1542B722B72B1E
%E090: 732B702B713EFF3219
%E098: A82B3E0CE7FD212D4F
```

Nove reči u rečniku

GRAPH — grafika  
TEXT — tekst  
PLOT x,y — bela tačka  
UNPLOT x,y — crna tačka  
DRAW x,y — bela linija  
UNDRAW x,y — crna linija  
kursori se izvršavaju izmđu znaka navoda  
PRINT"AAA BBB"  
inicijalizacija  
A=USR (E000)

Objavljeno softversko i hardversko rešenje fine grafika nema nikakve veze sa projektom na kome autori Milan Tadić i Nenad Dunjić trenutno rade za potrebe Elektronike inženjering i Zavoda za Udžbenike i nastavna sredstva.

```
%E0A0: E476ED5ED1F7F13E9C
%E0A8: 0C32A82BED56FD2172
%E0B0: 31E4F7AF3C06AFC16D
%E0B8: F5CD01E4F1F95CD1F
%E0C0: 26E1F1D1F7AF3C06B1
%E0C8: AFC1F5CD01E42A6A8B
%E0D0: 2H11400195223563F
%E0D8: EB78941601300416058
%E0E0: FF7C904779951E157
%E0E8: 30041EFF7D914F903E
%E0F0: 300DF137F57A535F8E
%E0F8: 7C656F78414790C5HD
%E100: 040CD9D11C1DD928F4
%E108: B990300681087C8206
%E110: 6708087D836FF1F5CC
%E118: E530037C656FE3D924
%E120: CD26E10810DFE1E397
%E128: E52167E128032164FE
%E130: E1E37CFED63002A8FF7
%E138: 67E52FD630071DE64B
%E140: 070603CB3C0B1D100F
%E148: FAE3E5060F016AE1DD
%E150: 097E2A6A2A0114005H
%E158: 09C1712370C1090199
%E160: 080009C92FB06A6E6
%E168: 77C9F6E2B7FB7EFDFFB
%E170: BF7FCD52E4E1C1D1B4
%E178: E17DFEAS200409C3C1
%E180: 6600FEC1CA40E2E5F6
%E188: C5CD61E423237FE7A95
%E190: 2820FE06281C78FED6
%E198: 203835FEDF300CFEHA
%E1A0: DB38084604280CE67F
%E1A8: 3F1861FE2220047E7A
%E1B0: 2F7782B682A77E536
%E1B8: CD1BE4CDH2E404CDF0
%E1C0: 69E47E883801477073
%E1C8: E123CD98E4F1D9C9E0
```

```
%E1D0: FE4D2005CD81E4187A
%E1D8: F4FE0C202721E4E12B
%E1E0: E5CD1A0A2H6A2H0699A
%E1E8: 1036002310FB3A8856
%E1F0: 2B3C2009111000199A
%E1F8: 01001A545D1336FF14
%E200: EDB018C946042004EC
%E208: F6C018A7FE1B38BD83
%E210: 200311E0FFFE1C204D
%E218: 03112000FE1D200372
%E220: 11FFFFFE1E2003115F
%E228: 01002H682A197CFE50
%E230: 2A38022628FE283008
%E238: 002262922682A188D8H
%E240: D9CD61E43601CDF5E4
%E248: 0CCD52E42H682AF5C0
%E250: CD1BE4F1B7280CFE6E
%E258: 00CAE0E2FE5F283A58
%E260: E718DEE5E5CD69E4C1
%E268: 7E9028023004E1E1E2
%E270: 18CFB7280E47E32321
%E278: 7EE710FB3C06E23AF
%E280: 7E20EF2BE3CD1BE467
%E288: HFB32002062005E195
%E290: 3E20E770E122682H4A
%E298: 18A7E5CD69E4AF86F3
%E2A0: 3007E1CD92E4C35472
%E2A8: 01CB6E2320F1902826
%E2B0: 023003E118882B3418
%E2B8: E116005F1947115C0D
%E2C0: 29D7380CC5E5C29E4C
%E2C8: E3E1112000E052C1F5
%E2D0: 22682H2B7EE710F84C
%E2D8: 22682H3E20E/18850C6
%E2E0: CD69E40FB928070DBE
%E2E8: 28CB6E2HF70C2100A8
%E2F0: 2828071120004119E2
%E2F8: 10FDE5C069E41186D3
%E300: 2B46A8FB82818E37E79
%E308: 1223137BFE3420061B
%E310: CD81E4C3540110EF49
%E318: E3CB6E2320E33E008D
%E320: 1213E105CD92E401EF
%E328: C9ED586C2A21E001A9
%E330: 78E61F1320FA18ED05
%E338: 52234538434404B896
%E340: DACBEA0E5D21200000
%E348: 19ED00E1CD1BE4CD30
%E350: 69E47923462B7023ED
%E358: 3CFE0F38F63A88288A
%E360: 3C201CCD61E419E588
%E368: CD29E5C109E519D174
%E370: E305CD1BE4CD29E55F
%E378: D1444DE1EDB03EE11F
%E380: 21E029E5CD1BE4CD8H
%E388: 71E4CD61E42B77E1EH
%E390: C9F5C5D5E509C5D50E
%E398: E5017F202A6A2A1154
%E3A0: 18001956235E52111
%E3A8: 3E0D9E1DD05ED2188
%E3B0: C7E32356233E0BC5H
%E3B8: 3F3800CB3F47380000
%E3C0: 10FC46234A428C7825
%E3C8: 227F20CHF5E3ED4F9F
%E3D0: 79ED47ED537E207803
```

&E3D8: D980A1D9470000AFC9  
 &E3E0: D9C80A1D9814F006E  
 &E3E8: HFD9CB0B1F85D95762  
 &E3F0: 9225D9DDE9DDE1D934  
 &E3F8: E1D1C1D9C3C0009160  
 &E400: E3CFE5CD050045E18F  
 &E408: 4DE1F1D5F5E5C612FH  
 &E410: E42323237E3CE1C2AH  
 &E418: C2E0E93E1F9547CB9F  
 &E420: 0F4D71F1F1F1FE60A  
 &E428: 0F4FCB04C93EE3ED04  
 &E430: 47CD61E47EFE01380E  
 &E438: 0435CC40E4C3FD00E9  
 &E440: 36142346ED5B682A8D  
 &E448: 1A77E870CD1BE4C37B  
 &E450: A2E4F5CD61E436009C  
 &E458: 233EBF8EC443E4F18A  
 &E460: C92A6A2H11100019C1  
 &E468: C9ED586A2AEF691815  
 &E470: F63E2077E5CDA2E403  
 &E478: 04E1237DE61F20F19B  
 &E480: C92A682ACD1BE4E536  
 &E488: CD65E470E1CD71E48D  
 &E490: 1806237DE61F20FHD  
 &E498: 70CF2AD429E322680E  
 &E4A0: 2AC93A8283CC0C5C1  
 &E4A8: E5LD29E5484709E33B  
 &E4B0: 7EE1012000ED586A32  
 &E4B8: 2H1909FE5B3835F3E10  
 &E4C0: 5C3821281FFE5E3890  
 &E4C8: 1EF5E73814FE8F20A4  
 &E4D0: 043E58181FFED8B93E  
 &E4D8: 4EFEDF904H067F1812  
 &E4E0: 133E53113E43113E85  
 &E4E8: 5A36D7200236DF09A7  
 &E4F0: 36EF 180536F0936B8F  
 &E4F8: FF0905203829E51E62  
 &E500: 09EF67553E08293053  
 &E508: 01193D20F9E52A659A  
 &E510: 2H111600195E235641  
 &E518: E119EBE13E090081A2F  
 &E520: 77130983D20F7C1B0  
 &E528: C911A001EF793C3D5C  
 &E530: C81918FBFF0000FFFFF0  
 &E538: FFFF0000FFFFEFC8  
 &E540: EFEFEFEFEFEF9393D0  
 &E548: B7BDF0000000007D64  
 &E550: D701D7D7D7D7D7D7D7  
 &E558: EF83EDED836F6F8330  
 &E560: EFFF3B3D0FEFF79B84  
 &E568: 98FF83DDEBF7E85084  
 &E570: B05D63FEFCFAF6EF56  
 &E578: F6FAFCFEDFF7F72H6  
 &E580: F7F7F7EFDFF7EFD7F8  
 &E588: DFD0FD0FEFF7FFEF50  
 &E590: ABC701C7A8EFFF0FD2  
 &E598: EFEFEF01EFEFEFF98A  
 &E5A0: FFF000000000E7E7FB8  
 &E5A8: F7FFF00000001FFFFF2  
 &E5B0: FFFF00000000FFFFF8  
 &E5B8: FFE7E7F7F7FBDFEF58  
 &E5C0: F7FBDFDC7BB3D5D08  
 &E5C8: 6D7579B8C7E7E7E82A  
 &E5D0: EDEFEFEFC7C7BBF4  
 &E5D8: 7D7FBFL7FBFD01C742

&E5E0: BBBFCFB77F7DBBC786  
 &E5E8: BFDFEFF7BB8D1BFBC  
 &E5F0: BF01FDC5897D7F7DB4  
 &E5F8: BBC7C7BBFDC5B97DFC  
 &E600: 7DBBC7017DBDFEFA9A  
 &E608: EFF7F7C7BBB8C7D8  
 &E610: BB7D7DBBC7C7BB7D36  
 &E618: 7D3B477FBC7FBE7E6  
 &E620: E7FFFFFFE7E7FFF80  
 &E628: E7E7FFFFFFE7E7FF80  
 &E630: BFDFEFF7FB7EFD7F44  
 &E638: BFFF00001F0F01FFBC  
 &E640: FFFFFFFF7EFD7BFD5C  
 &E648: EFF7FBC7BB7DBDF7E  
 &E650: EFEFFFEFDFF7F0394  
 &E658: FF03F7EFD0C7BB7DC6  
 &E660: 7D7D017D7D7D0C1BD0F  
 &E668: 7DBDC1BD7D7D81C7FF  
 &E670: BB7DFD0FD7DDBCC72E  
 &E678: C1B7D7D7D7D7DDB8C  
 &E680: C101FD0FD0C1FD0F74  
 &E688: FD0101FD0FD0C1FDB4  
 &E690: FDFD0F877BFD0D000  
 &E698: 7D7D7B877D7D7D7D0F  
 &E6A0: 017D7D7D7D7D7E7E79A  
 &E6A8: EFEFEFEFEFC7017FF2  
 &E6B0: 7F7F7F7F7DBCC7BDB8  
 &E6B8: DDEDF59F95EDDD0B34  
 &E6C0: FDFD0FD0FD0FD0FDE8  
 &E6C8: 017D39556D7D7D7D0F  
 &E6D0: 7D7D7D7D79756D5D8C  
 &E6D8: 3D7D7D7C7BB77D7D30  
 &E6E0: 7D7DBBC7817D7D7D74  
 &E6E8: 81FD0FD0FD0C7BB7D74  
 &E6F0: 7D7D7D5DDB847817D04  
 &E6F8: 7D7D81EDDD0B7D8706  
 &E700: 7BFDB8C7BF7F78877A  
 &E708: 01EFEFEFEFEFEFEF8A  
 &E710: EF7D7D7D7D7D7D7D58  
 &E718: B8C77D7D7DDBBB882H  
 &E720: D7DFE7D7D7D6D0DEE  
 &E728: 6D6D55B87D7DB8D776  
 &E730: EFD7BB7D7D7D7D7D7E  
 &E738: 3B477F7FBBC7017F82  
 &E740: BDFEFFF7BFD01017E  
 &E748: 010101010101010198  
 &E750: 00001028441010009C  
 &E758: 00000010104428109C  
 &E760: 000000001008640084  
 &E768: 1000000001020418C  
 &E770: 20100000E305115854  
 &E778: 07D71280B7CFE207C  
 &E780: 200226E0E3C30F10E  
 &E788: 218DE7C39A0344528B  
 &E790: 4157A0C8554E445239  
 &E798: 4157A0C5504C4F543C  
 &E7A0: A086554E504C4F5438  
 &E7A8: A0834752415048A065  
 &E7B0: 3B54455854A0869056  
 &E7B8: 0F000000000000000F

tim, pokušate da snimate ili učitate sa  
 kasete program, izdžite „mašinic“ ili is-  
 kjučite generisanje slike, dinamička me-  
 morija može nenadano da stekne telepat-  
 ske sposobnosti. Autori „grafike“ su ovo  
 prilično gorko iskusili, naročito prilikom  
 snimanja ili učitavanja oko 16K izvornog  
 programskog koda za podršku grafike i  
 ekranskog editora. Rešenje je, srećom, vrlo  
 jednostavno (slika 1).

Trik se sastoji u tome da spoljnim kompo-  
 nentama „produžimo“ procesorski brojač  
 za osvežavanje memorije za jedan bit.  
 Flip-flop D1 zamrznuće stanje poslednjeg  
 bita R registra procesora Z 80. Flip-flop D2  
 promeniće stanje na svom izlazu pri svakoj  
 tranziciji adresnog bita A6 sa logičkog  
 nivoa 1 na 0. Troulazno i kolo omogućuje  
 pojavu simultarnog osmog bita R registra  
 samo za vreme perioda za osvežavanje memo-  
 rije koji se dešavaju od „galaksija“ ne  
 generiše sliku. O dojavu početka i kraja slike  
 stara se flip-flop D3. Signal RAM A7 spaja  
 se samo sa odgovarajućim vodom na me-  
 morijskom proširenju.

Kao što vidimo sa slike 2, promene na  
 generatoru zvuka su minimalne: • Uklinite  
 ulazno-izlazni zahtev za vreme dok pro-  
 sor javlja da je prihvatio interapt istovrem-  
 eno aktiviranjem signala IOR i M1.

• Adresa ulazno-izlaznih portova ostala  
 je neizmjenjena, ali su oslobođena ulazno-  
 izlazne adrese koje koriste interfejs za  
 štampač i assembler u koji je ugrađena i  
 mogućnost za prenošenje programa u me-  
 moriju spoljnog procesora.

Grafički „hardver“ je, u stvari, toliko  
 jednostavan da, gotovo, uopšte i ne postoji.  
 Princip njegovog rada je „zaobilaznje“  
 optima 2716 u kome su na određeni način  
 pohranjeni likovi znakova i direktno povezi-  
 vanje procesorske magistrale podataka sa  
 ulazima pomaćorskog registra 74LS166.

Video signal nastaje direktno iz poma-  
 ćorskog registra.

Na ovaj način se na ekran neposredno  
 prenosi sadržaj RAM-a.

Na slici 3. je data kompletna shema  
 grafičkog proširenja za „galaksiju“.

Flip-flop 74LS74 upotrebljen je kao proš-  
 irenje već postojećeg memorijskog porta  
 74LS174. Njime se upravlja programski i on  
 na svom nekomplementiranom izlazu sadr-  
 ži informaciju: 0-tekst, 1-grafika izvode  
 rezolucije. Na ovaj način optima 2716 sa  
 opismama znakova i oktalni neinvertujući ba-  
 fer 74LS244 se naizmenično stavljaju u  
 stanje izvode impedancne ostvarujući pome-  
 nuti princip rada. Po jedno ili i kolo  
 upotrebljeno je za čitanje grafičke memorije  
 za vreme intervale osvežavanja. Ova kola  
 se nalaze neupotrebljena na memorijskom  
 proširenju.

Red je da okončamo sa grafičkim „har-  
 dverom“. Uočite na shemi „galaksije“ flip-  
 flop 74LS74 čiji je osmi „pin“ povezan sa  
 nožicom 15 (SH/LD) pomaćorskog registra  
 74LS166.

Presjecite veze koje spajaju njegove no-  
 žice 11, 12 i 13. Takođe oslobodite veze na  
 „pinovima“ 11 i 13 int. kola 74LS38. Poveži-  
 te oslobođene nožice prema slici 4. Ova  
 izmena obezbeđuje punjenje pomaćorskog  
 registra tačno u trenutku kada su podaci iz  
 dinamičkog RAM-a spremni.

Završili ste povezivanje i sada imate  
 „galaksiju“ punu žica i novih mogućnosti.  
 Nedostaje vam samo optima 2716 napunjen  
 odgovarajućim programom koji se postavl-  
 ja na mesto za to predviđeno u generatoru  
 zvuka.

Nenad Dunjić

se preko LATCH-a 74LS174 i dve ni kapije.  
 Sve dok imate sliku na ekranu, vaša memo-  
 rija je bezbedna. Processor automatski osve-  
 žava sedam bitova A0 do A6, a programski  
 osmi bit A7 dovoljnom brzinom. Ako, među-

ima, zvuka, program teče sporo (a ionako ne radi brzo), pa opet nema zabave.

Problem, rekomsno, nije lako rešiti, ali je moguće. Pomoću interapt moda 2 može se naložiti procesoru da 50 puta u sekundi prekine izvođenje redovnog programa i uradi NEŠTO sprovedno. To NEŠTO može sasvim lepo da bude generisanje jednog vrlo kratkog zvuka. Pošto se cela stvar odvija jako brzo (50 puta u sekundi), korisnik ima utisak da sluša jedan staljan, kontinuiran zvuk. Pri svemu tome, izvođenje bezijk programa se vrlo malo usporava. Naravno, zvuk dobijen na ovaj način nije baš najboljeg kvaliteta, ali za akcione igre zadovoljava.

Program ima samo jednu programsku liniju i možete ga staviti ispred svakog programa koji zahteva „ozvučenje“. Zvuk se postize tako što se na adresu 65316 „poukuje“ njegova dužina, i to u pedesetim delovima sekunde. Recimo, POKE 65316,100 će proizvesti zvuk u trajanju od dve sekunde. Ako vam se izabrana frekvencija ne sviđa, možete da probate da je promenite „poukovanjem“ nekih vrednosti na adresi 65299 i 65302. Ako na te dve adrese „poukujete“ suviše velike brojeve, recimo, preko 100, primetno će se usporiti izvođenje bezijk programa.

```
1 CLEAR 65278: FOR I=65279 TO 65323: READ X: P OKE I, X: NEXT I: RANDOMIZE UST 65317: DATA 1, 2, 55, 245, 58, 36, 255, 254, 0, 40, 23, 61, 50, 36, 255, 197, 213, 229, 221, 229, 33, 30, 0, 17, 20, 0, 17, 20, 205, 181, 3: 221, 225, 225, 200, 200, 241, 255, 237, 77, 0, 62, 254, 237, 71, 237, 94, 20, 201
```

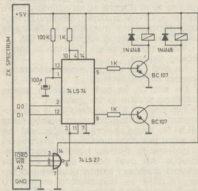
## ZX OUT

Zoran Hristić iz Kraljeva traži shemu pomoću koje bi „spektrum“ mogao da kontroliše dva releja.

Sema koju objavljujemo je krajnje jednostavna. Integralno kolo 74LS27 služi kao nepotpuni adresni dekodler. Njegov izlaz aktivira dvostruki FLIP/FLOP 74LS74. Samo kolo 74LS74, na osnovu onoga što se nalazi na data linijama D0 i D1, zna koji od dva releja treba da aktivira. Od materijala je potreban jedan EDGE konektor, integralna kola 74LS27 i 74LS74, dva releja od 5 volti, dva kondenzatora od 0,1 mikro farada, jedan elektrolitski kondenzator od 100 mikro farada, tri otpornika od 1K i jedan od 100K, dve diode 1N4148 i dva tranzistora BC107.

Releji se aktiviraju sa OUT 127, X, gde za X postoje 4 moguće kombinacije

- X=0 — dva releja isključena
- X=1 — uključeni relej 1
- X=2 — uključeni relej 2
- X=3 — uključena oba releja



## Zvuk, zvuk i samo zvuk

Dordević Mihajlo iz Beograda nas pita da li je moguće nekako izvesti da „spektrum“ „zuj“ a da pri tom ne staje sa izveštavanjem bezijk programa.

Naravno da jeste, ali to nije baš najjednostavnije. Pošto „spektrum“ nema specijalan čip za generisanje zvuka, njegovi konstruktori su rešili problem tako što su procesoru, pored svih muka, dodali i tu da periodično aktivira zvučnik onoliko puta u sekundi kolika je frekvencija, kada korisnik zadaje BEEP naredbu. Pri tome, da bi zvuk bio lep, a ne isprekidan, procesor mora da prekine izvođenje svih svojih poslova, pa otuda program staje dok „spektrum“ pišti. Time je pisanje akcionih igara na bezijkju jako otežano. Ako nema zvuka, nema ni dobre zabave, a ako

## Amstrad user klub

Jedan čitalac iz Vranja nas pita da li u našoj zemlji postoji neki klub vlasnika „amstrada“. U Beogradu je nedavno osnovan AMSTRAD USER KLUB (tel: 011/425-180). To je jedan od najprofesionalnijih klubova u našoj zemlji, sa izvanrednom softverskom ponudom, ali moramo da vas upozorimo da taj klub nema drugog cilja sem da što bolje finansijski olakša svoje članove.

## Z80B

Maksimović Dragoljub iz Nikića pita da li je moguće ugraditi u SPECTRUM Z80B procesor i kvarc kristal od 6MHz i na taj način povećati brzinu rada.

Nikom slučajaju! To je nemoguće iz gomile razloga, čiji spisak ne bi išao ni na dve cele strane, pa ga zato i ne objavljujemo. Isti odgovor važi i za one koji bi u „komodor 64“ hteli da ugrade 6502A.

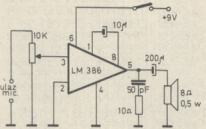
## Audio ulaz

Ostojić Radoslav iz Beograda pita čemu služi audio ulaz na „komodoru“. U pitanju je direktno vezan za SID čip, što omogućava da se na tri standardna generatora zvuka doda još i neki spoljni zvuk. Konkretno, ako se na audio ulaz dovede neki signal, recimo govor, onda će SID čip taj govor pomešati sa onim što daju generatori zvuka, sve to propusti kroz filtere, ako su uključeni, i na kraju poslato na zvučnik televizora.

Čim su šlozom otvorena vrata za razne muzičke efekte, ali prilikom eksperimentiranja treba paziti na sledeće:

— audio ulaz treba da bude mikrofonske jačine, ako kao izvor zvuka koristite EAR izlaz nekog kasetofona ili sličnog uređaja, potencijometar za jačinu stavite na najtiše moguće.

— masa audio izlaza se spaja na pin broj 2 AUDIO/VIDEO konektora sa zadnje strane „komodora“, a signal na pin broj 5.



## „Cela istina“ o 68008

Perić Dragoljub iz Niša se interesuje za procesor 68008 koji je urađen u QL-a — koja je „cela istina“ i gde da nabavi literaturu za taj procesor.

Procesor MC68008 nije ništa drugo slavni „Motorolin“ 16-bitni procesor MC68000, ali sa 8-bitnom magistralom za podatke. Unutrašnja arhitektura je ostala potpuno ista (sa izuzetkom dve-tri naredbe), tako da korisniku stoje na raspolaganju šesnaest 32-bitnih registra i izvanredno širok i moćan set naredbi. Pošto MC68000 obrađuje 16 bita odjednom, to, praktično, znači da MC68008 sastavlja naredbu iz dva poziva memoriji pre nego što je izvrši. To, naravno, usporava brzinu rada za oko 30 posto. Prednost osposobnog „deja basa“ je u tome što ne zahteva izuzetno skup i komplikovan hardver, kao što je to slučaj sa 16-bitnim magistralama za podatke. U tome i jeste tajna QL-ove vrlo niske cene. Računari sa procesorom MC68000 obično koštaju daleko iznad 1000 funti. Po našem mišljenju, MC68008, i pored 8-bitne magistrale za podatke, sigurno spada u grupu 16-bitnih procesora, ali ne i u grupu 32-bitnih — kao što to Sinciar tvrdi. Takvi procesori (kao što je MC68020) koštaju daleko više nego čitav QL! Galaksija će, verovatno, posvetiti više pažnje procesoru MC68000 u jednom od narednih brojeva „Računara“.

Što se tiče literature, SYBEX je izdao jednu knjigu na francuskom pod naslovom „MISE EN OEUVRRE DU 68000“. Vrlo je verovatno da SYBEX prodaje istu tu knjigu i na engleskom jeziku. Moramo samo da vas upozorimo da autor ne troši hartiju na objašnjavanje osnovnih pojmova, pa je za razumevanje ove knjige potrebno vrlo veliko predznanje iz oblasti mikroprocesorske tehnike.

## Olivetti M24

Više čitalaca nam se javilo se željom da kažemo dve tri reči o OLIVETTI-jevom trenutno najpopularnijem kompjuteru M24.

OLIVETTI M24 je jedan od najboljih, ako ne najbolji IBM PC kompatibilan kompjuter koji se može naći na tržištu. Procesor je pravi 16-bitni INTEL-ov 8086 na 8MHz, pa zato M24 radi skoro dva puta brže nego IBM PC (8088 na 4.77MHz). Operativni sistem je naravno, MS DOS, što će reći da se mogu koristiti gotovo svi programi napisani za IBM, a između ostalog i programski jezici BASIC, pascal, fortran, kobal, PL1, for i drugi. RAM memorija je kapaciteta 128K i može se proširiti da najviše 640K. Na ekranu se prikazuje tekst u formatu 25 linija sa 80 slova. Grafika u boji može da bude 640X200 (kao IBM) ili 640X400 tačkica. U mašinu su ugrađene dve disk jedinice od po 370K, a može se dodati i hard disk. Tastatura je, naravno, profesionalna sa 102 tastera i izdvojenim funkcijским tasterima, numeričkim setom i tasterima za vođenje kursora. Ono što najviše impresionira je tehnička savršenost, velika brzina i potpuna IBM kompatibilnost. Cena kompletnog sistema sa dva disk drajva i zelenim monitorom se kreće oko 2000 funti.

Priprema  
Vladimir Kostić

Hardverska rešenja  
Miroslav Milošević





Jure Špiler

# BASIC

za ZX SPECTRUM

## OPET RADOSNA VEST ZA SVE USMERENE, STARE I MLADE NIKAD NIJE KASNO!

**Ako želite da naučite više od onoga što pruža usmerena škola i imate računar ZX spectrum, onda je to prava knjiga za vas.**

Sinclairov ZX spectrum je bez dvoumljenja najrasprostranjeniji mikroručunar kod nas. Uprkos niskoj ceni i malim dimenzijama, koje so mu otvorile put u slovenačke domove, nije, naime, "mali" računar. Početniku nudi baš sve, što mu je potrebno da nauči osnove računarstva i programiranja.

Spectrum ima priličan broj specifičnosti (neke su zaista simpatične), a sve su obuhvaćene u navedenom priručniku.

Knjiga je namenjena svima koji imaju računar ZX spectrum, a sve do danas su tražili dobar i kompletan priručnik na domaćem jeziku koji bi detaljno, ali ipak jednostavno i očigledno opisao sve što spectrum može.

Dobro će doći i onima koji računar još nemaju ali uprkos tome žele da nauče programiranje u programskom jeziku BASIC, da se upoznaju sa ovim računarom i njegovom dodatnom opremom. Kod primera su uvek prikazani i rezultati, a kod svih važnijih primera detaljno je opisano šta se događa kad program aktiviramo.

Iz sadržine:

- osnove računarstva i programiranja
- uključivanje računara
- tastatura
- BASIC za ZX spectrum
- dodatna oprema - priključivanje i upotreba
- spisak sistemskih varijabli
- pregled svih slogova, naredbi, funkcija i obaveštenja

U knjizi je opisan i rad s posrednikom ZX interface i koji omogućava povezivanje više spectruma u mrežu, priključivanje mikrojedinica magnetnih traka i povezivanje spectruma sa računarima drugih proizvođača.

Knjiga ima 220 stranica. Sada i na srpskohrvatskom jeziku.

**KNJIGU MOZETE DA KUPITE U SVIM  
BOLJIM KNJIŽARAMA ILI DA PORUČITE NA  
ADRESU:  
BASIC, P. P. 302, 61001 LJUBLJANA**

cena: 1500 din