

izdaje BIGZ, QUR, ...

racunari

20

Specijalni izdavanje
novembra 1988.
Uključuje 100 stranica
cena 400 dinara

triling kečeva

„spektrum 128+2“

„amstrad pc1512“

„master kompakt“

birati pred licem pravde

**izvinite,
nismo znali**

racunari u akciji

**loto na
„spektrumu“**



Dr Radončić
A. Milinović



Razglednica iz Njujorka

Optičke memorije

Filips i Du Pon su nedavno otvorili marketing i tehnički centar za Sjedinjene Države sa ciljem da zajedničkim snagama nastupe na američkom tržištu optičkih memorija.

Laboratorija za specijalne tehničke aplikacije optičkih memorija sa kancelarijama za marketing smještena je u Wilmingtonu, država Delaver, u novozgrađenim prostorijama od 1200 kvadratnih metara. U istom gradu se nalazi i prva fabrika optičkih ROM diskova. U stilu svih velikih evropskih kompanija, koje zbog raspoloživosti reprodematizirala i tehničkog personala, otvaraju fabričke visoke tehnologije na teritoriji SAD, Philips je uspostavio identičnu proizvodnu liniju u Ajdohfenu, Holandija.

Svoju verziju optičkih disk jedinica Philips je nazvao CD-PROM. Za razliku od standardnih CD-ROM diskova, u koje se informacija upisuje laserskim graviranjem udubljena na disku, CD-PROM jedinice upisuju informacione biljege promenom faza površinskog materijala. Filipsov uređaj je u stanju da upisuje i očitava podatke kako sa CD-ROM tako i sa CD-PROM diskova.

Proizvodnja CD-ROM diskova je skuplja i isplati se samo u velikim količinama, dok će proizvodnja CD-PROM diskova biti ekonomski opravdana i u manjim količinama.

Očekuje se da Filipsova CD-PROM jedinica košta manje od 1.000 dolara, a sam disk manje od 100 dolara.

Glavni kupac Filipsovih optičkih diskova je, za sada Digital poznati proizvođač mini računara.

Laserske zdravstvene legitimacije

Firma Dreksler iz Mauntin Vjua, Kalifornija, pregovara sa poznatom firmom Kanon da zajednički započnu sa masovnom proizvodnjom laserskih zdravstvenih legitimacija. U pitanju je projekat vrednosti multimilijon dolara, a iniciralo ga je nekoliko dečaka, među kojima i dvojica jugoslovenskog porekla (Tesić i Sarić).

Interesantna ideja golobradih „mornaka sa čoška“ bila je da se kartica sa laserski upisanim podacima ogromnog kapaciteta iskoristi za upis kompletne zdravstvene istorije vlasnika kartice, donela je svakom od njih, uz malu pomoć advokata i komsije lekara, po preko četvrt miliona dolara. U



momentu kada je kriza sa kontrolom informacija o troškovima lečenja nekoliko miliona osiguranih postala evidentna, glavna kompanija za zdravstveno osiguranje u SAD, Plavi Krst i Plavi Štit, (Blue Cross and Blue Shield) prihvatila je sa oduševljenjem ideju dečaka o upotrebi laserskih legitimacija.

Primena novog načina identifikacije i evidentiranja pacijenata omogućile osiguravajućem društvu iz države Merilend da proširi broj osiguranih sa par miliona na 80 miliona. Osnovni uslov Plavog Krsta i Plavog Štita pre kompletnog prihvatanja ove nove tehnologije je bio da Dreksler izabere još jednu kompaniju koja bi, kao rezervna, proizvodila legitimacije u uređaju za očitavanje. U Americi je naime, praksa da se svaki projekat velikih dimenzija obezbedi sa više proizvođača istovetne opreme. Ovakvom poslovnom politikom se izbegavaju krizne nestašice reprodematizirala, a i sprečava se mogućnost ucene. Rezervni proizvođač je među inženjerima poznat popularno kao „drugi izvor“ (second source).

Odluka Kreksela, da drugi izvor bude Kanon, bazirana je na činjenici da je Kanon jedini bio u stanju da u potpunosti garantuje: totalnu raspoloživost bilo kog broja legitimacija u bilo kom momentu, zadovoljavajući kvalitet i spremnost da se brzo prilagodi novim zahtevima za modifikacijom legitimacija ako ovi kojim slučajem iskrnu u budućnosti.

U upotrebi je trenutno 15.000 eksperimentalnih legitimacija. Laserski čitači su proizvodnje Kanon, po licenci Drekslera. Pravi Krst i Plavi Štit takođe poseduje licencu. Glavni nosioci svih licencnih prava su, svakako, „dečaci sa čoška“ i Dreksler.

Teraatari

Mada je još uvek gotovo neverovatno zamisliti lični računar sa memorijom od jednog gigabajta (milijarda bajta), uskoro se mogu očekivati memorije hiljadu puta veće kapaciteta, reda veličine terabajta. TB, Holandska firma DCCdata planira laserski sistem sa optičkom trakom kapaciteta jednog terabajta. Prototipni uređaj, DOKtočak (DOCwheel), koristi 128 optičkih kaseti veličine popularnih audio kaseti i kapaciteta 8 gigabajta (8GB), aranziranih u vidu kruga površine 0,3 kvadratna metra.

Veruje se da će DOKtočak biti u stanju da pronade bit koji datoteku između astronomske velikog broja mogućih datoteka za samo 10 sekundi.

Komercijalni raspoloživost terabajtnih memorija je planirana za 1987. godinu. Mogućnost da se samo jednom upisuje na optičku traku DOKtočka ne predstavlja značajan hendikap u odnosu na izbrisivu magnetnu traku. Sekvencijalni karakter upisivanja na traku ne dozvoljava česte izmene upisanih datoteka, kao što je to slučaj sa magnetnim diskom (reorganizacija datoteka na traci uzima dosta računarskog vremena, tako da se i magnetna traka koristi gotovo kao nelizbrisivi medijum). Neće nas, naravno, iznenaditi ako jedan od prvih računara bude poneo ime „teragalaksija“ ili, barem, „teraatari“.

Mikroprocesor od galijum arsenida

Poznat proizvođač vojne opreme Mekdonel Daglas uspeo je da integriše 1860 galijum-arsenidnih tranzistora na četvorstruki četvorbitni mikroprocesorski čip MD-2901. MD-2901 je prvi mikroprocesor napravljen u novoj tehnologiji.

Optičke memorije ogromnih kapaciteta i komunikacioni kanali sa optičkim vlaknima ogromne informacione propusne moći zahtevaju velike brzine rada pratećih elektronskih (u budućnosti verovatno kompletno optičkih) uređaja. Zato se u Sjedinjenim Državama intenzivno traga za alternativnim rešenjima. Rezultat višegodišnjeg istraživanja, prvenstveno u SAD, predstavljaju galijum-arsenidni elektronski sklopovi sa velike brzine rada i male potrošnje električne energije. Primera radi, privenac MD-2901 troši svega 135mW. Radi se o procesoru tipa „bit-kriške“ (bit slice), koji uspešno izvršava programe napisane za poznati AMD-2901 mikroprocesor.

U laboratorijama Mekdonel Daglasa je u toku projekat 32 bitnog galijum-arsenidnog mikroprocesora.

Softverske pljeskavice

Kompanija Borland Internacional postala je preko noći jedan od glavnih proizvođača softvera za mikoročunare. Na Konferenciji za automatske i kompjuterske mašine, (ACM), u Ostinu Teksas, 1985. godine,

autor je imao prilike da se sretno sa osnivačima ove danas značajne kompanije. Ideja grupe relativno mladih momaka iz grada Skots Veli, Kalifornija, bila je da se piratima doskoči niskim cenama softvera, toliko niskim da se više isplati kupiti softverski paket sa garancijom nego ga kopirati. Broj pirata u odnosu na regularne kupce softvera pokazao se zapanjujući. Naime, broj prodanih prvih Borland-ovih Turbopascal paketa po ceni od 39 dolara bio je 10 prema 1 u odnosu na broj prodanih pascal prevodila bilo koje druge firme, čije su cene, do izlaska Borlanda na tržište, bile minimum 100 dolara. Borland Internacional je bila prva firma koja je sa parolom „što jeftiniji softver-to više kupaca“ u stilu slavnog Mekdonalda (proizvođač jeftinih lepinja i pljeskavica) postala softverski gigant.

Dosljedn svojoj poslovnoj politici, osnivači Borlanda su nedavno otkupili malu softversku firmu Singular-Softver sa svim proizvodima i projektima, da bi svim proizvođačima Singulara odmah dobili cenu za 33%. Singular je osnovan pre godinu dana. Jedini uspešni proizvod je bio paket „Interlejs“ za relaciuonu obradu baze podataka za Eptov „mekintoš“. Za nepunih godinu dana, Singular sa svega sedam zaposlenih zaradio je prodajom ovog paketa preko milion dolara. Borland je „interlejs“ prilego i za IBM-PC i prodaje ga pod nazivom „Refleks“. Cena „Refleksa“ je 99,95 dolara.

Prva elektronska biblioteka

Americki Institut Elektro i Elektronskih inženjera (IEEE) je juna ove godine instalirao u Nju Jorku prvu elektronsku biblioteku. Na konferenciji za štampu potpredsednik sektora za publikacije IEEE, Carls Haus demonstrirao je pretraživanje 40.000 stranica svje publikacija objavljenih 1984. godine uz pomoć običnog personalnog računara. Demonstracija elektronskih publikacija predstavlja prvi korak ka budućem totalno elektronicizovanom distribuiranju tehničkih informacija. Izgleda da je papirnim publikacijama Gutenbergovog tipa potpuno „odvonilo“.

Količina tehničkih informacija uvećava se svakodnevno neočekivanom brzinom. Kako sabrati sve raspoložive podatke o nekom naučno-tehničkom problemu u ekonomski opravdanom vremenskom periodu i kako razlučiti aktuelne od prevaziđenih činjenica: Na ovo pitanje je jedino moguće odgovoriti kompjuterizovanim pretraživanjem i obradom. Na drugoj strani su enormni troškovi klasičnog štampanja svega što naučno-tehnički autori žele da saopšte zainteresovanim čitaocima. Kuriozitet radi, za nedeljno izdanje Nju Jork Tajmsa potrebno je poseći gotovo neverovatnih dve stotine hektara šume?! Mnogobrojne publikacije IEEE, distribuirane po celom svetu, zahtevaju najverovatnije daleko više papira. Da se kojim slučajem ne recirkuliše stari papir, mogli bismo samo da zamislimo kako bi Amerika i šumovita Kanada izgledala za pedesetak godina.

Februara ove godine na sastanku generalnog rukovodstva IEEE odlučeno je da se započne sa „soft-publikovanjem“. Predložena su dva odvojena ali srodna projekta: izgradnja elektronske baze svih do sada podataka objavljenih i izgradnja sistema za



podnošenje novih članaka za publikovanje elektronskim putem, elektronskom poštom ili slanjem diske sa tekstom. Za sada samo kompjuterska sekcija IEEEa prima predložene tekstove u softveru. Nedostatak određenog standarda u vezi sa formatom članaka uz format računarskog komuniciranja, kao i nedovoljan kvalitet postojeće kompjuterske grafike sa prihvatljivom cenom koštanja osnovna su prepreka elektronskog sabiranja predloženih „rukopisa“.

Prenos „rukopisa“ sa jednog na drugi računar različitog tipa još uvek nije jednostavan zadatak. Pre tri godine je pri udruženju američkih Publicista (Association of American Publishers; AAP), započeto sa razradom standardnih procedura za elektronsko podnošenje članaka za štampu. Rezultat dosadašnjih napora je odštampan ove godine kao Standardni generalni jezik za označavanje (Standard Generalized Markup language; SGML). Korišćenjem SGMLa, autora sada mogu da lično koduju svoje rukopise. Prednost korišćenja SGMLa je ubrzo objavljivanje članaka bez nepotrebnih preklapanja i komplikovanog slaganja nepripremljenog teksta.

Na sastanku u avgustu ove godine, generalno rukovodstvo IEEEa je razmatralo detalje elektronskog, tzv. telekonferenacija. Razmatrane teme su bile: elektronsko publikovanje, biblioteka budućnosti, kompjuterizovano konfereniranje, grafičko usklađivanje i buduće metode kompjuterizovanog tehničkog obrazovanja.

Jedan od konkretnih projekata koji su u toku je kreiranje indeksa predavanja sa svih (nekoliko stotina) naučno-tehničkih konferencija održanih 1986. godine u SAD. Početna investicija je 100.000 dolara. Do januara ove godine, prema rečima Hausa, svi pretplatnici na računarsku mrežu „Kompjuterska pošta“ (Compa i I), ili na mrežu „Nalaženje svoga puta“ (Find-Your-Way), biće u mogućnosti da, korišćenjem personalnog računara i modema, pretražuju indekse za 1986. godinu.

Da bi se ispunila baza podataka od gore pomenutih 40.000 stranica teksta, upotrebljen je optički skaner za direktno učitava-

nje podataka koji su, zatim, upisani svi na samo jedan laserski disk od 12 inča. Demonstrirani sistem je razvijen u firmi „Univerzity Mikrofilm Internacional“ (UMI) iz En Arbra, država Mičigen, koja poseduje kopije svih mistarskih i doktorskih disertacija ikada napisanih u Sjedinjenim Državama. Predviđa se da će UMlev sistem biti povezan sa korisnicima faksimil mašinama visoke grafike rezolucije. Naime, predviđa se mogućnost kompjuterskog pretraživanja tehničke baze podataka i štampanje samo željenih tekstova i grafike.

Po zaduženosti gotovo svih zemalja sveta kod američkih banaka, da se delimično zaključiti od kolikog je značaja savršenost i efikasnost manipulisanja ekonomskim informacijama. Informacije o promenama situacije na svetskom tržištu su dostupne američkim biznismenima znatno ranije kašnjenjem nego bilo gde u svetu. Očigledno, na redu je automatizovana obrada ne samo ekonomskih već i naučno-tehničkih informacija. Po mišljenju mnogih, zemlje industrijalizovanog zapada na čelu sa SAD „informacionom eksplozijom“ eksponencijalnom brzinom uvećavaju tehnološki proces u odnosu na srednje i zemlje u razvoju. Ostavimo budućnosti da odgovori kakve će implikacije ovo imati na čovečanstvo.

Klon sa japanske strane

Svojom verzijom IBM-PC AT računara, sa oznakom MBC-990, Sanio se nedavno pridružio armiji proizvođača AT kompatibilnih računara. Za razliku od ostalih klonova, Sanion MBC-990 je računar minimalne konfiguracije, bez ikakvih ugrađenih opcionih ploča. Kupcu se ostavlja sloboda da ugrađuje prema potrebi ploče sa različitim funkcijama. Standardna konfiguracija sa 512K RAMa, floppy disk jedinicom kapaciteta 1,2 MB i operativnim sistemom MS-DOS 3,10 košta 2.599 dolara, što je više od cene po kojima ekvivalentne računare od 30MB, kolor monitorom Sanio-CRT-80 i kolor karticom, cena MBC-990 se penje do 4.000 dolara.

Računari
u izlogu

Spektrum 128
plus 2

„spektrumovo“ novo ruho

Na ovogodišnjem PCW sajmu u Londonu, koji je održan početkom septembra, „AMSTRAD“ je predstavio najnovije izdanje starog, dobrog „spektruma“, koje je kratko „+2“, i pustio ga u beli svet...

Novinari skoro svih engleskih računarskih časopisa su gotovo poludeli od sreće! Kompjuterske revije su prepune novih slika, prikaza, iznutrica, hvalospjeva, a sve je to posvećeno „Sink... pardon, „Amstradovom“ novom plenu, „+2“.

O. K. Idemo redom. Prvo...

TASTATURA

Može se reći da je oblik i raspored tastera upravo najbolja kombinacija starog „plusa“ i „CPC 464“. Svi tasteri su prilično zgodno raspoređeni; jedino je „edit“ previše blizu „caps shiftu“, pa ponekada možemo uleteti u editovanje i bez svoje volje. Takođe, taster „break“ je u gornjem desnom uglu, gde je prirodno mesto za „delete“, tako da ponekad prsti sami polete prema čošku. Ipak, oni koji su navikli na raspored kod „plusa“, ovdje će se naći na svom tlu.

Najvažnija karakteristika nove tastature je da nema štampanih tastera sa „keywordima“, što se dosada najviše i zamerilo svim Sinklerovim tastaturama. U stvari, ostala su samo tri, i to „run“, „code“ i „load“ — valjda zbog toga što se najviše upotrebljavaju. Ipak, treba imati na umu da će se novi „spektrum“ koristiti najviše u modu 128, a u mod 48 čemo se vraćati samo radi igara ili nekog specifičnog programa. Uostalom, mnogo je lakše kucati slovo po slovo, nego juriti šiftove naoklo. Zato sam najviše voleo „Beta Basic“.

Još jedna stvar — razmaknica je postala nešto veća, jer su i svi tasteri prošireni. Lepo, pogotovo za one kojima „spektrum“ služi za pisanje.

Ah, da, osećaj! Imam gadan utisak da se ispod krije membrana, ali ovoga puta sofisticiranija, sa osećajem kao kod „amstrada CPC 464“ ali sa blagim klikom. Perfektno. Idemo dalje...

KADA SE UKLJUČI...

... izgleda gotovo potpuno isto kao i običan „spektrum 128K“. Ipak, pada u oči natpis „© 1986, © 1982 Amstrad Consumer Electronic plc“ što gotovo nagoni suze na oči, a i nagoveštava velike probleme sa određenim programima koji testiraju čeksum ROM-a, jer je sada produžena reklamna poruka. Takođe, u meniju nedostaje opcija „tape tester“, jer ugrađeni kasetofon (verovatno) ne može da omane...

Sve ostalo, uključujući i attribute i ostale skalimerije, ostalo je po starom, tj. blesavo i drago (ljubi ga majka...).

KASETAŠ...

... predstavlja prvu stvar koja upada u oči čim se vidi novajlija. Da li će učitati sve programe, ne znamo, ali znamo da nema

4/računari u izlogu



pošeavanje jačine ili boje tona; ako stvari krenu naopako, jedino možete da uzmete šrafciغر i da pokušate da podesite glavu kroz mali otvor sa gornje strane. Dalje, prva i jedina spolja uočljiva MANA je ta da kasetofon NEMA, ponavlja, NEMA BROJAČ!!! Nazdravljaj! Zamislite situaciju da sve one brižljivo sortirane i obeležene kasete više ne mogu da se lako upotrebljavaju, jer... gde li ono beše „Tasword“ na prvog ili drugog trećini strane?... Zaista, „Amstradovci“ su mogli da se malo sažale i odvoje pare za lole bolju verziju kasetofona, koja bi imala brojač, pa makar to koštalo preko 150 funti. Inače, Englezi tvrde da je ovaj model bolji od onoga ugrađenog u „CPC 464“! Robusniji je i ima deblju pogonsku traku... Ako maier tera, onda ne pomaže ni automobilska guma...

UNUTAR KUČIŠTA...

... leži čudna zverka. Za onoga koji nikada nije otvorio stariji „128“, nova štam-

pana pločica će izgledati u prvi mah kao IBM-ov „motherboard“... dobro, dobro, preterao sam. Međutim, ima novosti. Hladnjak je sada opet iznutra, samo bolje smješten, a i famozni kalorifer „ULA“ je dobio hladnjak na leđa. Jedina razlika se vidi na čipovima. U sred srede leži novi čip koji pompezno nosi oznaku „Amstrad“ — to je, u stvari, novi ROM (sa novom kopirajnt porukom!) — a na mestu gde stoji oznaka za verziju pločice piše... dobro, sve već znate.

ULAZI/IZLAZI...

... na sve strane. Prosto je neverovatno koliko otvora je dobio novi „spektrum“. Ako se setimo 1982. godine i starog „gumenjaka“... Ovakvo, i dalje je tu konektor opšte namene, ali ima problema sa njim (čita) dalje). Tu su i RS 232/MIDI interfejs (vrlo slični konektori kao kod QL, u stvari britanski PTT džekovi), utičnica za „key-

pad", RGB monitor, TV prijemnik, izlaz za pojačalo (juhuuu!!!), tu su i reset taster i dva porta za džojstik.

E, pa baš sa tim džekovima nešto nije u redu... Na kutiji računara iznad ulaza lepo piše „koristite samo naše... SJSI džojstike“. Ovo već miriše na „Amstrad“ i njegove „dva u jednom“ džojstike, ali na drugi način. Naravno, džekovi su naizgled isti „stari“ standard, ali tek će pažljivo oko uočiti razliku u razmaku između kontakata. Originalni „SJSI“ džojstik liči na „Quickshot“, ali košta oko 15 funti i ni po čemu ne zaslužuje tu čast da bude kupljen. Podvala koju je zamislao Alen Šuger-Šečerko nije bila dugog veka:

Čim su se vratili sa PCW sajma, inženjeri firme „Cheetah“ su napravili adapter za „normalne“ džojstike i ugradili ga u svoj dobar džojstik 125, koji je tom prilikom dobio „+“ u imenu i sada se prodaje kao „125+“, i košta samo 9 funti! Prava ipak budeje... .

Da, i još jedna stvar. Džojstici rade po protokolu „interface 2“, tako da moramo reći zbogom starom, dobrom, univerzalnom „Kempstonu“.

PROGRAMIRANJE...

Je potpuno isto kao kod „128K“ modela. Jasnije je da svojim izgledom tastatura tera na programiranje u modu 128, ali i za one koji ne žele tu mogućnost, uz računare je priložena tabela „ključeva“, po kojoj možete da se setite gde... ono beše STRS, i ABSI?!

KOMPATIBILNOST

Evo, otprilike, šta se (tvrdi) oko tog fakta:

— svi priključni uređaji i periferije bi trebalo da rade bez problema. Jedino, zbog debljine plastike kod konektora opšte namene, mikrodrajveri bi trebalo da se pažljivo ugraju bračficerom ili silničim alatom. Isti problem će postojati i kod ostalih uređaja.

— Softver bi kompletno trebalo da bude kompatibilan, tj. ako je radio na „128K“, onda ni „+2“ ne bi trebalo da ga odbija. Jedini problem je kod programa koji testiraju čeksum ROM-a... već sam rekao.

— Stari džojstici neće pasovati na novu mašinu, dok se ne pojave (pojavi li su se) novi uređaji, prilagođeni novom obliku konektora.

U najgorem slučaju (zar?) , možete utaknuti svoj stari interfejs i koristiti ga kao i kod prethodnika. Sa konja na magarca, ali šta se tu može...

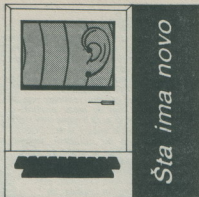
U SVAKOM SLUČAJU...

Za 150 funti dobijate zaista mnogo... Mašinu koja sigurno ima budućnost, jer iza nje ne stoje više mesečari koji svoje prve ideje daju na tržište, već ozbiljni poslovnjaci, koji svaki pene tripur prevrnu pre nego što ga ulože u nešto. Računar nasleđuje i ogromnu biblioteku od skoro (ili preko) 5000 programa, što predstavlja veliku prednost.

Jedna od najvažnijih činjenica je to da nijedan proizvođač računara ne nudi ovakav sistem za ovakve pare, ni „Atari“, ni „Amstrad“ a pogotovo ne „Komodor“. Držimo palčeve za starog kralja, jer njegovo novo ruho to zaslužuje!!!

Darko Stanojević

5/spektrumnovo novo ruho



Šta ima novo

Šetanja na daljinu

Sigurno ste čuli za Penmanov ploter. To je onaj ploter koji se umesto velike ploče i kompleksnog uređaja sastoji samo od jedne kutijice i dugačkog kabla. Kutijica se sama kreće po stolu (i po papiru) i ostavlja za sobom normalan trag. Zgodno? Sad je još zgodnije. Pojavila se varijanta Panmanovog plotera u koji je ugrađen modem. Šetajući ploter se lepo priključi preko telefona na vaš kompjuter, koji je kilometrima daleko, i crta šetajući po stolu. Za sad ne znamo cenu. (B.D.)

Amstrad vedri i oblači

Neverovatno ali istinito! Samo od marta ove godine „Amstrad“ je uspeo da proda oko 250.000 komada računara 8256 i 8512! Mnoge konkurentske firme su pokušale da prate ovaj uspeh obaranjem cena svojih računara, ali do danas niko nije uspeo da prestigne, ili samo dostigne, Amstradov bum.

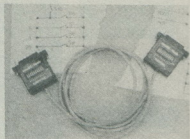
Dalje, firma je odlučila da svoje gorepomenute računare poskupi (!!) za 5 funti, naplaćivanjem priručnika za Malardov bežik, koji se do sada besplatno dobijao uz mašinu. Razlog za ovu dodatnu doplatu je, prema rečima službenika iz Amstrada, vrlo jednostavan: samo jedan od deset kupaca je koristio bežik za programiranje mašine, pa je odlučeno da se priručnik prodaje odvojeno ili po većoj ceni.

Verovatno zbog toga, bežik programi se još uvek nalaze na B strani sistemskog diska kao CP/M fajlovi. (D.S.)

Matrični ploter

Imate PCja. Želite da crtate i koštrirate na njemu. Zaljubiše se u mogućnosti nekog CAD programa. Nabavili ga od nekog prijatelja (naravno, besplatno). Šta vas tada preseče? Verovatno činjenica da bi vam za dostojno prikazivanje rezultata vašeg crtanja trebao ploter koji nije samo odvojeno kupiti jer ste poslednje para dali na matrični štampač. Jedini vaš izlaz predstavlja EOGraph ploča. EOGraph je ploča koja ne samo da služi kao jako koristan printerski bafer (iako bi samo za to bila malo preskupe) nego omogućuje da vaš matrični printer izbacuje crteže za koje je veoma teško reći da ih nije nacrtao ploter. Cena ploče je negde ispod 600 funti. (B.D.)

RS 232 KABEL SA SKLOPKAMA



Pri povezivanju računala sa vanjskim jedinicama može doći do raznih problema, od kojih su neki i nestandardni priključci. Da bi se takve smetnje izbegle tvrtka Lindy iz SR Njemačke proizvela je RS 232 kabl sa ugrađenim sklopkama, tako da kod eventualnih preinaka nije potrebno odmičivanje i ponovno lemljenje, već je dovoljno podesiti željenu komunikaciju samo uz pomoć sklopki. Dužina kabla je 2 m. Cijena je relativno niska — 70 DM. (Z.V.)

Ispomoz' za „amstrad“

Program „Sidekick“ za PC Kompatibilne računare već je postao prava legenda uprkos (ili možda upravo zahvaljujući) svojoj jednostavnosti i običnosti. Sada se na tržištu pojavio i njegov preprev za „amstradove“ računare pod imenom „Write Hand Man“. Program košta 30 funti i može da bude zaista koristan. Pored svih „Sidekick“-ovih performansi, ima i mogućnost da „amstradovu“ tastaturu privremeno tokenizujete.

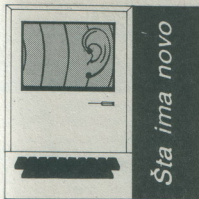
Rukavica u lice

Nekim kompanijama nikako ne polazi za rukom da postanu bestseleri. Međutim, to ih ne sprečava da i dalje usavršavaju svoje modele. Najnoviji primeri su firma „Tatung“ i „Saga Systems“, koje se trude da pariraju „Amstradu“.

„Tatung“ je proizveo novi računare, nazvan „Tatung 256“, koji kao, što mi i samo ime govori, ima 256 K memorije, 14 inčni color monitor i odvojenu tastaturu. Dalje, ima 512 boja, i stereo zvuk. Baziran je na Z80 procesoru i, naravno, CP/M kompatibilan. Firma ga ceni ravno 400 funti.

„Compliment“ je novi, prvi računare firme „Saga Systems“ i po rečima predstavnika te kuće, ima da bje „Amstrada“ na rođenom terenu! To je Z80 sistem (opet!) koji će koštati oko 300 funti, znači 100 funti jeftinije od PCW 8256. Sistem će obuhvatati 3,5 inčni disk-drajver, matrični štampač brzine 100 karaktera u sekundi i gomilu poklon-softvera. Monitor nije uključen u cenu, ali, ako se i on ubaci u računicu, cena opet nije viša od „Amstradove“.

„Saga Systems“ je poznatija po dodatnim tastaturama za „spektrum“ i bilo bi lepo kada bi ova nova mašina ličila na neku od njih. (D.S.)



Šta ima novo

Čavrljajte sa Tajancem

Ako slučajno imate tajanca (više nismo toliki optimisti da pretpostavljamo da neko ima čak i IBMovog PCja) možete ga naučiti da sluša vaše komande, zamerke, psovke i bilo šta što biste želeli da mu kažete. To vam omogućava Vocalink-paket koji se sastoji od pločice, mikrofona i diskete. Ova kombinacija omogućava vašem PC-kompatibilicu da razume oko 500 različitih komandi, od kojih već dobijate neke pripremljene za korišćenje Wordstara i Lotusa. Prava stvar za lenjivce. Cena? Prava sitnica. Njega oko 425 funti. (B.D.)

„Oric“ se ponovio!

Firma „WE Software“ je odlučila da produži samrtničke muke računara „ORIC“ (namerno ne kažemo „Oric-Nova“, pogodite zašto!), time što je ponudila njegovim vlasnicima nekoliko periferijskih uređaja. Da vidimo šta se tu kuva:

Prvo, disk-jedinica sa poboljšanim „Sedoric“ operativnim sistemom dvostrukog gustine pakovanja, po ceni od svega 239.65 funti, što predstavlja novost, jer je do sada njegova cena bila za 30 funti viša. Ah, da... format diskete je 3 inča... šmrnc, žalosno...

Drugo, modem. To je model V23 koji pristiže zajedno sa interfejsom i programom za podršku, treće, printer „kosmos“ i četvrto, programabilni interfejs za džozistik.

Mnogo poznatog (?) softvera će pristići sa redukovanim cenama, što će jako obradovati sadašnje vlasnike tog vremenom pregaženog, (ne)popularnog računara. (D.S.)

Nova cena „Računara“

Proizvodni troškovi „Računara“, naročito zbog poskupljenja papira, poslednjih meseci su toliko narasli da redakcija više ne može da ih podnosi sama. Od ovog broja, kao što je i red, delimo ih sa svojim čitaocima. Dakle, od sada za trećinu skuplje i, nadamo se, barem isto toliko bolje.

REDAKCIJA

DG/One 2

Tvrtka Data General, izbacila je na tržište, po karakteristikama i dizajnu vrhunsko prijenosno računalo. Pun naziv novog računala je DG/One Modell 2. Ono se može povezati na velike Data General uredske sisteme, kao i na IBM mainframe i PC kompjutore.

U osnovnoj varijanti računalo ima 256 K memorije koja se može proširiti do 640 K. Ugrađena je i 3,5 inčna disk jedinica, a operacijski sustav je MS DOS 2.1.

U računalo su ugrađeni serijski i paralelni interfejsi, a može se dodati i tvrdi disk kapaciteta 10 megabajta ili floppy disk jedinica za 5,25 inčne diskete. U samom računalu postoji još dovoljno mjesta za eventualna proširenja, npr. za procesor 8087.

Ekran je poboljšane LCD verzije. Cijena osnovne konfiguracije računala je cca. 5300 DM, a sa tvrdim diskom i procesorom 8087 cca. 9600 DM. Ukoliko se želi kvalitetniji EL-ekran (elektroluminescentni), treba doplatiti još 2600 DM.



Upravljanje riječima

U SAD je predstavljen novi sistem za upravljanje računala IBM PC AT riječima. Korisnik računala preko mikrofona i posebnog interfejsa unosi riječi (tekst) u računalo. Ciljeli postupak je znatno kvalitetniji od prvog ozbiljnog sistema te vrste koji se pojavio 1984. godine. Slične naprave su postojale i ranije, ali kvaliteta nije zadovoljavala visoke zahtjeve za profesionalnu upotrebu.



Zanimljivo je da je čip koji omogućuje spomenuto upravljanje računala riječima, razvijen u Švicarskoj. To je tzv. DSP-Chip (Digital Signal Processor) koji se sastoji od dva podsistema kojima upravlja AT procesor. DSP čip može izvesti oko 30 miliona operacija u sekundi.

Sistem rada je takav da korisnik prvo „unesu“ riječima probni tekst, računalo PC „snimi“ govornu karakteristiku te uskladi prijem u računalo. Tek potom se prelazi na unos glavnog teksta.

Prvi podsistem pretvara govorom uneseni tekst u znakove (fonetske simbole), a drugi podsistem od tih znakova tvori cijele riječi kojih u sistemu ima oko 5000. Za to je potrebno oko 2 megabajta memorije. Pouzdanost rada je vrlo velika — iznad 95 posto. (Z.V.)

Feniks za modele

Prava stvar za inženjere — program „Phoenix 3D“ za projektovanje trodimenzionalnih modela, koji se zatim mogu rotirati, smanjivati, povećavati, rastezati, seći i deformisati na sve moguće načine. Jako zgodno, zar ne. I sve to za samo 39,95 dolara ako se obratite na adresu „Dreams of the Phoenix Inc.“, PO Box 10273 Jacksonville, Florida, USA. (B. Đak.)

Štamparija na stolu

Dobro, ne baš štamparija, ali bar slovošlagračica. Dok se mi nadamo sličnim stvarima, u Engleskoj vlada ogromno interesovanje za malo nezavisno izdavaštvo uz pomoć kompjutera. Zato se od skora mogu naći paketi opreme potrebne za tako nešto po relativno niskim cenama. Tako „mekin-toš plus“, dodatni draž od 800K, LaserWriter laserski štampač, MacDraw, MacPrint, Majkrosoft tekst procesor Word, program za uređivanje novina Aldus PageMaker i par kinkora i pomoćnog materijala mogu da se nađu za neto malo preko 6500 funti. To je, možda, ogromna suma za naše krizne krajeve, ali je sasvim razumna ako se ima u vidu posao koji ona znači. Razmisлите, prodajte plac i kola i postanite nezavisni izdavač. (B. Đak.)

Bez virenja

Sećate li se naše vesti iz „Računara 15“ o tome kako je neki hlandski naučnik usavršio metod pomoću koga se iz blizine vaše zgrade pomoću opreme koja košta samo 15 dolara može otkriti šta piše na ekranu vašeg monitora ili TVa. Ako vas je to nateralo da se zabrinete, možete odahnuti. Firma Securitec je razvila specijalnu vrstu pokrivača za ko je pjuter i ekran koji sprečava zračenje pomoću koga je holandski naučnik Vim Van Ek postigao našćenje. Taj pokrov liti prekrivač je najesen početi da proizvodi kanadska firma Erintek, a trebalo bi da košta oko 500 (USA) dolara. (B. Đak.)

20

Izdaje BIGZ — OOUR „Duga“

Izlazi jedanput mesečno

cena 400 din. novembar 1986.

Specijalno izdanje

časopisa „Galaksija“

računari

Izdaje

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

Telefoni

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

Generalni direktor
Dobrosav Petrović

Direktor OOUR „Duga“
Bratoljub Babić
Glavni i odgovorni urednik
Gavrilo Vučković
Urednik izdanje
Jova Regasek

Tehnički urednik
Mirko Popov

Redakcija časopisa „Galaksija“
Tanasije Gavranović, pomoćnik
glavnog i odgovornog urednika
Esađ Jakupović, zamenik glavnog
i odgovornog urednika
Aleksandar Milinković, urednik
Jova Regasek, urednik
Zorka Simović, sekretar redakcije
Srđan Stojančević, novinar
Gavrilo Vučković, glavni i odgovorni
urednik

Stručna saradnja

Dejan Ristanović
Dušan Slavić
Nevenska Spalević
Anđelko Zgorelec

Spoljna redakcija

Branko Daković, Dejan Ristanović,
Jelena Rupnik, Jovan Skuljan, prof.
dr Dušan Slavić, Nevenka Spalević,
Zoran Životić

Stalni saradnici

Nada Aleksić, Ninoslav Čabrić,
Branko Daković, Voja Gašić, Bran-
ko Hebrang, Džorđe Janković, Vladi-
mir Kostić, Vladimir Krstonošić, Ra-
domir A. Mihajlović, Zvonimir Ma-
kovec, Blažimir Miše, Dejan Muha-
medagić, Ivan Nador, Radomir Ni-
kolajev, Zoran Obradović, Miodrag
Potkonjak, Dejan Ristanović, Jele-
na Rupnik, Dušan Slavić, Jovan
Skuljan, Nevenka Spalević, Darko
Stanojević, Zvonimir Vistrička, An-
đelko Zgorelec, Zoran Životić

Izdavački savet „Galaksija“

Dr Rudi Debijadi, prof. dr Branislav
Dimitrijević, (predsednik), Radovan
Drašković, Tanasije Gavranović, Ži-
vorad Gladić, Esađ Jakupović, Veli-
zar Maslač, Nikola Pajić, Željko
Perunović, prof. dr Momčilo Ristić,
Vlada Ristić, dr inž. Milorad Teofil-
lović, Vidojko Valičković, Velimir
Vesović, Milivoje Vuković

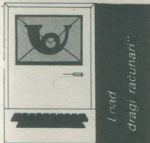
Štampa

Beogradski izdavačko-grafički za-
vod
11000 Beograd, Bulevar vojvode
Mišića 17
Žiro-račun kod SDK 60802-833-
-2463
Devizni račun kod Beobanke
60811-620-6-82701-999-01066
Za inostranstvo cena dvostruka
(400 D, 2,50 US\$, 6,50 DM, 45 Sch,
5,50 Sfrs, 20 Ffrs)

Na osnovu mišljenja Republičkog
sekretarijata za kulturu broj 413-
-77/72-03 i „Službenog glasnika“
broj 26/72, ovo izdanje oslobođeno
je poreza na promet.

sadržaj

- 2/ razglednica iz njujorka
- 3/ računari u izlogu
„spektrumo“ novo ruho
- 4/ šta ima novo
- 8/ load „drugi računari“
- 9/ računari u izlogu
„amstrad pc 1512“
- 12/ računari u izlogu
„master kompakt“
- 15/ peek & poke show
- 16/ komercijalni softver
turbo je nešto drugo
- 18/ komercijalni softver
treća sreća
- 19/ računari iz mog ugla
Kako se oženiti i živeti srećno stotinu godina
- 20/ mikroprocesori
procesor naših snova
- 23/ dejanove pitalice
- 24/ računari i pravo
izvinite, nismo znali
- 25/ hakerski vodič kroz beograd
u potrazi za izgubljenim vremenom
- 27/ umetak
6502
- 43/ mali oglasi
- 47/ tehnike programiranja
crtanje na mašincu (2)
- 50/ tehnike programiranja
kule bez rekurzije
- 52/ programiranje na mašincu
sinklerova veza
- 55/ kontroverze
algoritam „brzi gonzales“
- 56/ tehnike programiranja
iz šupljeg u prazno
- 60/ u domaćoj radinosti
turbodrajv za „spektrum“
- 62/ računari u akciji
loto na „spektrumu“
- 64/ u domaćoj radinosti
korak ka „plusu“
- 65/ šta ima novo u svetu komponenata
- 66/ razbarušeni sprajtovi



Suadova bela zastava

Dragi moji, vozdral!

Poslo sam počeo naružen od strane vaših čitalaca („spektrumovaca“ uglavnom), a posebno od vašeg saradnika Željka Jurica i zato što sam malo razmišljao, čitate ovo pismo. Dakle, zaključio sam da je rat između „spektrumovaca“ i „bejtovićevski nastrojenih komodorovaca“ isti kao i rat između Iraka i Irana. Ni jedni ni drugi nikad neće priznati da je njihov računar lošiji od onog drugog, ali će i jedni i drugi vazda tvrditi da je njihov računar bolji od onog drugog, a i kao što vidimo, nema nikakvog smisla. I zato, apelujem (zar Suad Bejtović, začetnik javnog rata?) na sve „spektrumovce“, a posebno na „komodoriste“ da zakopaju ratne sjekire i operu se od ratničkih biva. Evo, prvi je se izvinjavaju (da li dovoljno javno?) svojim „spektrumovcima“ za svaku uvredu na njihov i na račun njihovog računara. Posledice na objavljivanje ovog pisma mogu biti:

1. Šta to opet trabunja taj Bejtović? Mo računar je bolji i tačka! 2. Vidi što je to ovaj Bejtović dobro smislilo! Nije baš moj računar najljepi! Nije ni onaj, toliko loš!

O kvalitetu časopisa više se nema šta pričati! „Sprajtov“ još najbolji, ali mi se najviše sviđa članak „Platonove ljubavne muke“. Ostao sam bez teksta. Strašno! Malo me zbunjuje to što je Božidar Markutić ozbiljno shvatio „hakersku priču“ u, posloao scenario po kojem se stvarno može napraviti igra. Ali, nema veze, nije ni to loše, naprosto.

E, imam i ja nešto i za „Usjaini jostyck“ na radost onih koji misle da samarom djetostik 10 sati na dan i da imam biblioteku od barem 600 programa. U jednoj od novijih igara za „komodor“, „Urijadum“ postigao sam skor od 69285 poena i dopustivo do 7. nivoa, koji se zove PLATINA. Poslo nije siao ništa o toj igri smatram da je to rezultat za divljenje. Ili možda neko drugi ne smatra? Ja se ne lutim kad je neko bolji od mene. Zatim, „elitas“ sam strasveni, ali mi se u galaksiji baš ništa nije dogodilo. Od „Harmless od „Competnia“ Un, da, bio sam diskretno pripustian da li ću uzeti „najređu stvar u univerzumu“, ali sam otičaji, „Računare“ odgovorio sa NOOOO i izbegao probleme sa zrkovima. Da čujemo samo „... Na lijepom plavom Dunavu“ u Savršenoj Znam da tako ne misle vlasnici „spektruma 128“, ali se mora priznati da „komodor“

(svaka čast 128 BASIC-u ima izvedbi da doveš sluča i da se divi, jer! Iako? Jedno pitanje! Kako li, na koju adresu) mogu da se priplatim na PCW? Hvala. Evo, kao što se vidi, ostalo mi je još kap-dve tinte u naliv-peru te vas lijepo pozdravljam! (mrzi me da mijenjam patronu) i želim što manje ratničkih pisama.

Vaš,
Suad Bejtović
Sarajevo

Moramo priznati da ovakvo „pokajničko“ pismo od rodočelnika svih tuča odo računara, ni smo očekivali. Naravno, to ne znači da Suadov postupak nije lepravan, čak suprotno! On je lepo otpočeo proces, video šta je napravilo, posuo se papalom, izvnuo, i šta još hoće! Očekujemo, dalje, po jedno ovačko pismo od kapitena ekipa „spektrum“ i „amstrad“, a ni „BBC“ tim ne bi bio lenj, kada bi se malo potrudio da se spusti „iz oblake“.

Sili kojim Suad piše je vrlo lep i interesantan i zato bi bilo bolje da ga „arči pametnijim stvarima, nego što su prepucavanja. Zato, svi vi, „fajteri“, pišite nam, ali ovoga puta sa konkretnim stvarima, kritikama, opisima igara, svojim i predanim programima, vestima i s. Prva su vam uvek otvoreni.

O. K. na kraju ove adrese

PCW-a: STANOJEVIĆ DARKO

Mirjanin okrogli sto...

Drage računardžije, zdravo! „javljvam vam se već treći put sa nadom da ću biti onaj koji vodiće sa vašim čitaocima. Popelo mi je a na vrh glave to što u svakom broju (a to je poslednjih pet-šest) čitam:

... C-64 je super... gluposti... ništa ne valja... „spektrum“ je gluposti... živeo „komodor“... i silčno.

Malo tužno deluje da dozvoljavate deci da se na ovaj način prepucavaju preko novina. Izgleda mi da je njihov videli cilj štiti na stranice ne birajući sredstva. Iskreno rečeno, ni vaša redakcija nije napravila dobar potez što nastaviti to prepucavanje i dalje. Jednostavno, treba okončati s tim!

Što se drugih tema tiče, nema ni to mnogo pohvala. Ipak, dopada mi se „Biblioteka knjiga“. Mislim da ste, napokon, našli mesto gdje ćete informisati ljude o knjigama. Ali, ne bi trebalo da se zadržite samo na tome. Treba uputiti reči kritike na neka izdanja, kao što su raznorazni priručnici za programere tipa „Kako postati pravi programer“. To mi zaista deluje kao labu stvar. I to je spomenuto, to je sve fugaljivo i svodi se samo na inicijale (A.K.) Jasnije, glasnije, drugovili!

Postoji jedna tuža stvar o kojoj novine nerado pišu. Svi komentarišu knjige za osnovnu i srednju školu, valjda zato što su one najmasovnije, a niko ne pominje studente — visokoškolske. Smesno je kad kažem da sam uđbenik, u oktobru 1985, platila 1000, a da će nova cena biti preko 4500, da ne da. Da li vi imate komentara na ovo? Tužno.

Imam još jedan komentar. Mislim da nije u redu da se danas prelazi preko činjenice da raznorazni „radnički univerziteti“ prave „programere i operatera“ za zamislite, 3, 6 ili 9 meseci. Kao na fabričkoj traci. Pa za koji davo ljudi uče u srednjoj školi programiranje, programerske jezike (bejzik, kobol, fortran... sve ili pojedinačno), programske sisteme i onolike matematičke?!

Uvek je naš narod voleo prečicom, pa makar srijao u gluport. Radnih organizacija se to, normalno, više isplati, jer ne može da otera dugopodignuti radnika, koji postaje, malitene, tehnološki višak, nego ga lepo obuči za male pare i zdravo. Vuk sti i ovce na broju. Pa nije fer! A to isto važi i za mene koja završavam višu školu (za primenjenju informatiku i statistiku) i imam 18 predmeta. Smrači mi se pred očima.

No, iako te visokomene ustanove (fakulteti, instituti...) i organizuju nekakav kurs-seminar, tu pravo pristupa imaju samo izabrani ili zaposleni. Ali, molću lepo, zašto se i tu ne bi uzele pare — uplati kotizaciju ako si student (ako možeš?)!

Mož predlog je da se organizuju besplatna predavanja pojedinih profesora ili stručnjaka koji su raspoloženi da prenose znanje na mlade narastaje. Naravno, da bi ovaj potez uspeo, mora imati pokrovitelja. Predlažem da to budete vi. Šta kažete na predlog?!, Marjanjenal je.

Ako vam je potreban dežurni kritičar, ovog vašeg-našeg lista, samo se javite na donju adresu.

Puno pozdrava od

Mirel
Mirjana Nikolić
Bulevar revolucije 255/5
11050 Beograd

Obradovali smo se kada smo, nakon tolike pauze, primili tvoje pismo. Odmah smo ga poznao — ti znaš po čemu. Primamo te, naravno, za dežurnog kritičara. U našoj redakciji žene ionako uglavnom kritikuju, a muškarka uglavnom stvaraju list. Navrati do redakcije da se upoznamo. Stvarno.

Uspesna zamena

Od pre nekoliko brojeva sam primeto da imate sve manje teksto: va drugu „Jovana Skuljina“, što smatram za propust, jer u njegovoj članci jedni od najboljih koji se objavljuju u vašem časopisu. Zašto je to tako?

Ipak, smatram da niste baš sve upskali. Primeto sam da od skoro kod vas radi i drag Željko Juric, čiji su tekstovi u vezi s „spektrumom“ iako dore vrlo interesantni. O čemu se radi? Da li se Jovan umorio od pisanja i da li ste Željku ubacili kao zamenu?

Srdačan pozdrav,
Pavlović Slaven
Siask

Joca i dalje, na posredan način, služi „Računarima“ i našim čitaocima. Naime, on je na odsluženju vojnog roka i čim završi obuku sve će biti po starom. Željko je novi član u našoj ekipi. Bez obzira na obzire.

Vratite ženske

Pročitao sam najnovije „Računare“. Nema potrebe da nabrajam neke tu pohvale, jer vi verovatno znate koliko ste dobri. Zato pravo na temu: naslovna! Da se razume — ja sam jako želeo da znam kako izgleda „amstradov“ PC. I ne samo to, ona stika sa onim ljudima u mafijaškim odelima je jako dobro ispla. Ipak, pored svega toga, vratite one vaše ženske na naslovnu stranu. Možda ženske nisu uvek najlepše, ali su vam zato naslovne uvek najzanimljivije. Još jednom — vratite ženske! Rodoljub Indić
Savska 47
Beograd

Muški doo redakcije (mi smo u pluzdru) prihvatilo je tvoj predlog sa aplauzom. Šta kažeš za ovu?

Nova tarifa

Napisacu vam pismo u samo sedam rečenica. Ono što meni smeta u vašim pismima su predupćak pisma koja podu polu rubrike i onda ostare malo interesantna za zaista zanimljivije stvari. Ta pisma sigurno pišu skribomani koji se odušev da više svoje ime u novinama. Nek se oni dopisju međusobno, a vi im slobodno secite pisma u male malečke delove ili ih bacajte u koš. Predlažem da svi koji vam napišu pismo daju od sedam reči reči moraju da za reči koje su višak plate tarifu (kao za male oglase). Tako ćete dobiti kraća pisma, a možda ćete nešir i zaraditi. Puno pozdrava.
Vukoslav Beškic
Sarajevo

Jeste Vukoslavo, bilo je samo sedam rečenica, ali sedam dugih-kih rečenica. Slatemo se sa tvojim predlogom. U ovom broju je bilo tačno 98 reči (me li štampamo besplatno). Kad odjednom 40 reči, ostaje 58 tako da nam ti duguješ taman za jednu dobru večeru. Javi se.

Sopstvena rubrika pisama

U prošlom broju „Računara“ ste objavili jedno minijaturno o pismo koje se ticalo „Peek & poke show-a“. Tak kada sam njega video u rubrici „Load“, dragi Računari! shvatio sam da veoma retko objavljujete pisma koja se tiču PP Show-a. Ako je suditi po interesovanju koje moje društvo pokazuje za tu rubriku mora da vam stize puno pisama povodom nje. Da li ih vi namerno ne objavljujete ili vam ta pisma nisu dovoljno interesantna? Ako su makr približno u stilu Peek & poke show-a, zaslužuju da budu objavljena. Ugred, mogu li ja da pošaljem koji prilog za PPS? U svakom slučaju primite puno pozdrava od
Jovana Stojiljkovića
Novi Sad

Dragi Jovane, pisma koja stizu povodom PEEK & POKE Show-a mogu se podeliti (grubo) u dve grupe: veća grupa se sastoji iz pisma fanašičnih obožavatelja, dok se manje ili konkretnije grupe sastoji od mržnjom nabijenih zapisa „Onih Drugih“. I jedna i druga bi ozbiljno narušile ukoceni akademski nivo koji pokušavamo da damo našim časopisima i skupom listu, pa ih ne objavljujemo.

Računari
u izlogu

šećerni pc

Amstrad PC 1512

PC 1512

Amstradov PC klon je suvereno osvojio naslovne strane kompjuterskih časopisa — po mnogima se radi o događaju sezone ili, čak, događaju nekoliko poslednjih sezona! Hakeri sa svih meridijana zaista imaju razloga da budu zadovoljni: za 400 funti mogu da nabave kompletan IBM kompatibilan sistem iza koga, da bi stvar bila posebno lepa, stoji jedna uspešna i poznata firma koja se trenutno nalazi u brzom usponu!

Iako verujemo da su osnovne karakteristike Amstradovog PC-ja već dobro poznate u hakerskim krugovima, poput svih IBM komatibilaca, „amstrad PC“ se sastoji od tri dela: tastature, osnovne kutije u koju je ugrađena štampana ploča i disk jedinica i koja predstavlja podnožje za treći deo, monohrom (ili kolor) monitora. Dimenzije su prilično male: računar će zauzeti samo 0,15 kvadratnih metara vašeg stola!

Tastatura je mehanička i, kada se uzme u obzir cena, izuzetno dobra: sastoji se od 59 QWERTY tastera, deset funkcijskih dirki i numeričke tastature od 16 dirki. Nisu obezbeđeni posebni tasteri za pokretanje kurzora ali se odgovarajući efekat postiže pomoću numeričke tastature, što se vidi i sa slike. Tastatura je, sve u svemu, rađena po uzoru na IBM PC AT ali je napravljen jedan neobjašnjen kiks: veoma lepi i veliki tasteri ENTER, SHIFT, ISERT i slinči su niži od ostalih i dopunjeni površinom za pritiškanje veličine običnog tastera. Uzvišenje

na tasteru vam nikako neće pomoći pri brzom kucanju. Iako je nejasno zašto „amstrad PC“ nema normalne kontrolne tastere, tastatura novog PC kiona može da zasluži solidnu ocenu.

Amstrad se pobrinuo da tastatura ne bude jedini način za upravljanje kompjuterom: uz računar ćete dobiti i ergonomski oblikovanog miša sa dva tastera koji je podržan postojećim operativnim sistemom. Miš se, međutim, neće pokazati naročito korisno: upotrebljavaće ga pri radu sa GEM-om ili nekim programima koji budu pisani specijalno za „amstrad PC“, ali će većina čuvenih programa za IBM PC i dalje primati komande samo sa tastature.

Mikroprocesor(i): sasvim klasičan izbor

„Možak“ računara predstavlja Intelov mikroprocesor 8086 koji je zamenio „originalni“ 8088 koji koristi IBM. Ova promena treba da nas obraduje: 8086 je softverski 100% (zaista 100%) kompatibilan sa 8088, ali ima šesnaestobitnu magistralu za podatke što znači da omogućava brži rad. „Amstrad PC“, uz to, radi na kloku od 8 MHz, što znači da bi trebao da bude 2 puta brži od IBM PC-ja (4.77 MHz).

Da li ugrađnja bržeg mikroprocesora može da dovede do problema sa kompatibilnošću? Ako biste pokušali da prodajete „spektrume“ koji rade sa Z80 B na kloku od 6 MHz, ne biste napravili naročiti posao: ionako preteške igre bi postale toliko brze da niko ne bi ni pomišljao da ih igra, pa bi novi model, uz svu kompatibilnost, za trenutak ostao bez upotrebljivog softvera! IBM PC, sa druge strane, ne predstavlja mašinu koja se kupuje zbog igara — poslovni softver će brže raditi ako posedujete brži procesor, ali će obe verzije računara biti sasvim upotrebljive. Zamenu mikroprocesora je olakšao i sam IBM koji u svoje novije modele (XT i AT) ugrađuje 80186 i 80286, a razmišlja i o računarima sa 80386 — najveći deo postojećih programa će, dakle, perfektno raditi bez obzira na brzinu mikroprocesora.

Zašto Amstrad, kada je već menjao procesor, nije odabrao 80186, čip koji koristi sam IBM? Razlog treba tražiti u cenama: 80186 je i dalje relativno nov čip koji, uz to, nije baš lako nabaviti u velikim količinama na tržištima Dalekog Istoka gde se Amstrad snabdeva (kako onda IBM rizikuje? „Veliki Plavi“ je toliko važan klijent da će Intel uvek najpre podmiriti njegove potrebe). Amstrad je, najzad, verovatno primetio da

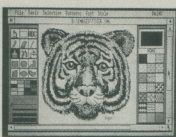
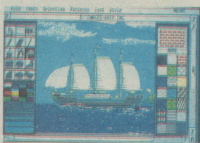
se 8086 sasvim dobro pokazao kod Olivetti-jevog M24 i mnogih drugih kompatibilca, pa ga je izabrao kao savršen kompromis između mogućnosti i cene.

PC 1512 ima 512 kilobajta RAM-a koji se na ploči može proširiti do 640 kilobajta prostim umetanjem memorijskih čipova. Znaajući da je 640 kilobajta najveća memorija do koje se standardni IBM PC proširuje, većina softverskih firmi projektuje svoje proizvode tako da se izvršavaju baš u ovoj konfiguraciji. Preporučujemo vam, dakle, da proširite RAM do krajnjih granica: jeste da će mnogi komercijalni programi raditi i na mašini sa 512 kilobajta ali ćete, pre ili posle, naići na paket koji vam životno treba i koji na vašem računaru ne radi. Ako živite u Londonu, prošetajte kroz nekoliko radnji i dokupiti memoriju. Ako, međutim, živite u Beogradu, ova će vas hitna kupovina malo skuplje koštati!

Kada smo već kod praznih podnožja, Amstrad je omogućio i ugradnju aritmetičkog koprocesora 8087. Šta je uopšte aritmetički koprocesor i da li ga treba kupiti? Kod bezmalo svih kućnih računara glavni mikroprocesor, uz pomoć softvera u ROM-u, sabira, oduzima, množi i deli racionalne brojeve i izračunava vrednosti elementarnih funkcija. Zar to nije sasvim normalno? Jeste, ali se u poslednje vreme radi i nešto bolje: razvijeni su specijalni mikroprocesori koji služe samo za obavljanje aritmetičkih operacija sa racionalnim brojevima i računanje vrednosti funkcija. Obzirom da je softver koji upravlja ovim računanjem pisao u mikrokodu, daleko je brži od svakog mogućeg mašinskog programa. Osim toga, aritmetički koprocesor u nekim situacijama radi paralelno sa centralnim 8086, čime se dalje ubrzava rad. Isplativa investicija? Samo ako strahno mnogo računate: u mnogim primenama vreme potrebno za računane vrednosti predstavlja pravu sitnicu u poređenju sa vremenom neophodnim za komunikaciju sa diskom ili neke druge slične poslove. Obzirom da je 8087 relativno skup čip (oko 120 funti), nećete pogrešiti ako se za početak odreknete njegovih usluga.

Grafika: jedna čudna odluka

Kada govorimo o IBM klonovima, pitanje grafike je veoma značajno. IBM je, naime, u početku prodavao računare sa takozvanom MDA (Monochrome Display Adaptor) karticom koja je omogućavala rad sa tekstem na crno-belom ekranu; grafike jednostavno nije bilo jer se smatralo da ona nije potrebna za poslovne primene računara. Ubrzo se pokazalo da se personalni kompjuteri itekako ozbiljno koriste za poslovnu grafiku, CAD/CAM i mnoge druge stvari koje zahtevaju crtanje po ekranu, a često i crtanje u boji (i dalje je, doduše, prisutno mišljenje da je za većinu poslovnih primena dovoljna crno-bela grafika jer se slike u boji teško prenose na papir i donje teško umnožavaju), pa je IBM pripremio CGA karticu — Colour Graphics Adaptor. Za IBM-ovu kolor karticu se obično ne nalazi baš mnogo komplimentata: smatra se da je rezolucija po vertikalni (200 tačaka) slaba, da su slova ružno dizajnirana i da se sporo iscrtavaju, da je izbor boja premali... Ipak, većina poslovnog softvera koji je napisan za IBM PC podrazumeva postojanje jedne od dve pomenute kartice, što znači da se nije lako odučiti za kupovinu neke delimično kompatibilne video ploče superiornih karakteri-



Gem Paint

stika. Mnogo detaljniju argumentaciju za razne video kartice možete da nađete u tekstu *Prava slika za prave programe* iz "računara 19".

Šta je Amstrad mogao da uradi? Želeo je računari koji će saradivati i sa kolor monitorom, pa je morao da odustane od Hercules kartice i IBM-ovog MDA. Nove IBM-ove kartice kao što su EGA (Enhanced Graphics Adaptor) i PGA (Professional Graphics Adaptor) su verovatno bile preskupe da bi ih imalo smisla kopirati, pa je „amstrad PC“ snabdeven video interfejsom koji je kompatibilan sa IBM-ovom CGA karticom. To, praktično, znači da je moguć rad sa grafičkom 320*200 u 4 boje ili 640*200 u dve boje: boje su fiksne i ne biraju se iz palete. Što se teksta tiče, u svakom od 25 redova se, po želji korisnika, može ispisivati 40 ili 80 znakova. „Amstrad PC“ može da se koristi i sa monohrom monitorom, pri čemu će boje biti zamenjene nijansama sivog.

Pošto smo završili sa tehničkim karakteristikama grafike, pogledajmo kako one mogu da utiču na kupovinu Amstradovog PC kompatibilca. U poslednje se vreme smatra da je za kvalitetno crtanje potrebno imati bar 512 tačaka po vertikalni ekrana, što znači da su kose linije na „amstrad PC-ju“ tasterastije nego što je dopustivo. Treba, ipak, da primetimo da IBM PC (pa samim tim i njegovi klonovi) nisu baš idealni za izuzetno preciznu grafiku i brzu animaciju — ako planirate ovakve primene, razmislite o „amigi“. Za standardnu poslovnu grafiku i „obične“ CAD primene grafika 640x200 predstavlja sasvim prihvatljivo rešenje. Na kraju krajeva, za prenošenje grafike već rezolucije na papir treba posedovati veoma dobar (i veoma skup) ploter!

Nevolja sa CGA karticom su slova: većina ljudi koristi IBM PC za rad sa tekstem i bazama podataka, što znači da kvalitet slova i brzina skrolovanja ekrana predstavljaju ključne kriterijume za izbor grafičke kartice. IBM je prilično nemarno dizajnirao slova, pa su oblici nekih od njih previše „stepenasti“. Slova, da bi stvar bila posebno nezgodna bilješkaju dok ekran skroloje, što je vrlo neprijatno pri dužem radu. Amstrad je rešio veću od ova dva problema: ekran je stabilan i relativno lako i brzo skroloje. Slova, sa druge strane, nisu naroc-

ito lepa ali se sa njima može (i mora) živeti! Možda ste pomislili da opremite PC 1512 Hercules karticom i koristite je za rad sa tekstem? Na žalost, neće iliti: videćemo da ekspanzioni slotovi nisu predviđeni za prijem standardnih grafičkih kartica, što je sa Amstradove strane prilično sumnjiva odluka, koja se rukovodi interesom i prilagodavanju svojih video adaptera Amstradovom standardu, što bi značilo da u bliskoj budućnosti možete da očekujete trošak za kupovinu neke monohrom kartice!

Bilo kako bilo, ako uopšte postoji razlog da odustanete od Amstradovog računara PC 1512, taj je razlog njegov video interfejs!

Periferija: bez funte doplata

Ako kupujete originalni IBM PC, spremite se da kupite i punu torbu raznih interfejsa. Ukoliko, sa druge strane, kupujete „amstrad PC“, bez funte doplata ćete dobiti sve interfejsne koji su vam potrebni za normalan rad. Tu je, pre svega, disk interfejs koji omogućava priključenje floppy ili masivnih diskova (jedan 5.25 floppy od 360 K je uračunat u cenu), paralelni **osobitni** Centronics interfejs za štampač, serijski RS 232 koji omogućava priključenje modema i mnogih drugih „sporih“ periferala, interfejs za džojstik ili miš (miš je, rekomo, uračunat u cenu) i, da bi spisak bio potpun, časovnik realnog vremena koji, zahvaljujući ugrađenim baterijama, radi i dok je računar isključen.

Za dalja proširenja su predviđena tri IBM kompatibilna porta u koja se, po Amstradovim tvrdjenjima, može priključiti „sve osim video kartice“ (prezrano je da se kaže da li je ova tvrdnja baš sasvim istinita). Možda će vam se učiniti da slotova za proširenja ima malo, ali treba primetiti da su video serijski, paralelni i disk interfejsi već ugrađeni na osnovnu ploču — kada ih ugradite na standardni IBM PC, neće vam ostati mnogo više mesta za dalja proširenja! Teško je, osim toga, zamisliti primenu u kojoj su vam **istovremeno** potrebne više od tri posebne kartice; ako vam nisu potrebne istovremeno, začas ćete ih zameniti jer su portovi veoma pristupačni.

Softver: bolje nego što se očekivalo

Uz „amstrad PC“ dobijate disketu sa najnovijom verzijom MS DOS-a (3.2). Digitalov DOS Plus, GEM, nekoliko uslužnih programa i „Locomotive Basic II“. Što se Digitalovog DOS-a i GEM-a tiče, nećete se mnogo uvaditi: Amstrad ga isporučuje uz računaru da bi nadmašio „atari 520 ST“, ali ovaj paket jednostavno nije primeren PC-jevoj koncepciji: najveći deo raspoloživog komercijalnog softvera ipak pretpostavlja da korisnik poseduje neki minimum znanja i da mu se ne mora stalno govoriti (i crtati) kako da okrene disketu i gde da je stavi pre nego što uključuje kompjuter!

Locomotive Basic II, s druge strane, predstavlja lep poklon: radi se o jednoj od novijih (i prilično brzih) verzija jezika koja se odlikuje mnogim elementima modernih jezika. Iako je „Locomotive Basic“ već izvesno vreme pristupačan i u Jugoslaviji, domaći programeri ipak više vole Turbo Pascal i Fortran 77.

Pominjanje programa koje koriste domaći vlasnici PC-ja nam dovodi do veoma značajnog pitanja — kompatibilnosti. Kao i

svi kompatibilci koje proizvode poznate firme, „amstrad PC“ ima BIOS ROM koji se razlikuje od „zvaničnog“ IBM-ovog; u protivnom bi Amstrad bio izložen ozbiljnim zakonskim komplikacijama. Ukoliko ste se, čitajući ove redove, nasmejali i rekli „Ako ne sme Amstrad, smem ja“, moramo da vas razočaramo: „amstrad PC“ nije potpuni hardverski klon IBM PC-ja, što znači da ne smete tek tako da mu umetnete (piratsku) kopiju originalnog BIOS-a. Promena BIOS-a znači da „amstrad PC“ nije potpuno IBM kompatibilan, tj. da se **može** napisati program za PC 1512 koji neće raditi na IBM PC-ju i, što je mnogo ozbiljnije, da se može napisati program za IBM PC koji neće raditi na Amstradovom PC-ju. Amstrad, sa druge strane, tvrdi da svi lole komercijalni programi za IBM PC rade i na njegovom računaru. Korisnici, poučeni raznim sličnim tvrdnjama raznih proizvođača računara, obično ne veruju baš mnogo ovakvim obećanjima. Gde je istina?

Obzirom da smo, pripremajući ovaj tekst, bili u prilici da se samo nekoliko sati zabavljamo „amstrad PC-jem, detaljan odgovor na pitanje o kompatibilnosti ostavljamo za neki od sledećih meseci; opisaćemo vam jedino naša iskustva. Isprobali smo „Wordstar“, „dBase II“, paket „Lotus“, „GW Basic“, „Turbo Pascal“, „Professional Fortran“, „Profi assembler“ i „SmartWork“; svi su, bar na prvi pogled, savršeno radili. Šta znači *na prvi pogled*? Sa programom smo radili kratko i uspešli da pregledamo neke od opcija. Detaljno isprobavanje samog „SmartWork-a“ bi, na primer, verovatno potrajalo par dana.

Na rezultate koje smo upravo opisali treba staviti još jednu senku sumnje, jer se radilo isključivo o piratskim kopijama sa kojih je već skinuta zaštita. Stara je istina da se kompatibilnost najbolje isprobava uz pomoć dobro zaštićenih programa a takve, u piratskim sredinama kao što je naša, nije baš lako pronaći. Imali smo priliku da isprobamo originalnu „dBASE III“ disketu i identičnu kopiju zaštićene diskete sa „Flight Simulatorom II“. Ako učitate dBASE III čim uključite računar, sve je u redu (ili bi se bar reklo da je tako); ukoliko pokušate da ga učitate posle dužeg rada, ponekad se računar „zaglavi“. Još je rano da se kaže da li je po sredi bag u prvim verzijama BIOS-a, nekompatibilnost sa MS DOS-om 3.2 ili ona najstrašnija, hardverska nekompatibilnost. Što se „Flight Simulatora“ tiče, program radi ali mi nismo umeli da uzletimo — još ne znamo da li se radi o nekompatibilnosti, brzem kloku ili, što je najverovatnije, našem nepoznavanju osnova pilotiranja i neposjedovanju uputstva.

Sve u svemu, prvi utisci o IBM kompatibilnosti „amstrad PC“-ja su izrazito povoljni — reklo bi se da Šugar ispunjava svoja obećanja. Da li da mu poverujemo da će se na tržištu pojaviti „amstrad PC“ verzije poznatih IBM programa po izrazito niskim cenama? Može se zamisliti da softverske firme pripreme zaštitu koja će dopuštati izvršavanje jeftinog programa samo na „amstrad PC-ju“, ali to za nas nije mnogo važno: u Jugoslaviji se lako dolazi i do veoma skupog softvera!

„Amstrad PC 1512“ u osnovnoj verziji koja obuhvata računar, jedan disk, solidan monohrom monitor i miš košta svega 400 funti + VAT. Za drugu disk jedinicu doplaćujete 100 funti, za kolor monitor 150 funti, a za hard disk od 20 megabajta samo 400



Amstrad PC — tehničke karakteristike

Mikroprocesor:	Intel 8086
Clock:	8 MHz;
Aritmetički koprocesor:	8087 (opcija);
Testatura:	59 QWERTY + 10 funkcijskih + 16 numeričkih;
Tip tastature:	Solidna mehanička;
RAM (KB):	512;
Maksimalni RAM (KB):	640 (ugrađena podnožja);
Sadržaj ROM-a:	Amstrad BIOS;
Operativni sistem:	MS DOS 3.2 ili ranije verzije;
Video:	Ugrađen c/b ili kolor monitor;
Tekst:	40*25 — 16 boja;
	80*25 — 16 boja;
	320*200 — 4 boje;
	640*200 — 2 boje;
	640*200 — 18 boja;
	Ugrađen, za flopi i hard disk;
Grafika:	5.25" DSDD 40 traka (uračunata);
Specijalni grafički mod:	360 klobajta;
Disk interfejs:	10 ili 20 megabajta (opcija);
Flopi disk jedinica:	Centronics, RS 232, miš, džojstik;
Kapacitet diskete:	MS DOS 3.2, Digital DOS Plus,
Hard disk:	GEM, GEM Desktop, GEM Paint,
Ostali interfejsi:	Locomotive Basic II, uslužne
Softver uračunat u cenu:	rutine (ukupno 4 diskete)
Cena (bez VAT-a):	399 funti;
Doplata za kolor monitor:	150 funti;
Doplata za drugi flopi:	100 funti;
Doplata za 100 M hard disk:	300 funti;
Doplata za 20 M hard disk:	400 funti.

funti — i ova je cena neverovatno niska, jer hard diskovi od samo 10 M koji se prodaju uz razne „tajvance“ retko koštaju manje od 1300 maraka! Cene su, sve u svemu, toliko neverovatno niske da Šugar ima pravo kada kaže: „Svi znaju sa čim je kompatibilan, a samo mi znamo kako smo postigli ovakvu cenu!“

„Amstrad PC 1512“ predstavlja izuzetno privlačan izbor ne samo za naše nikada dovoljno pune džepove, već i za engleske i nemačke biznismene koji stvarno treba da nađu dobar razlog za kupovinu višestruko skupljeg originala! „Amstrad PC“ će, ukoliko sledeći meseci ne otkriju neku njegovu

veliku i dobro skrivenu manu, sigurno postići ogroman uspeh u Evropi a možda i u Americi, premda su tamo zakoni tržišta nešto drugačiji: standardno je rezonovanje da, ako želite IBM PC, treba da telefonirate IBM-u! Značaj novog Amstradovog računara nije samo u tome što predstavlja izvanrednu kupovinu: sigurni smo da će njegova pojava izazvati dalji pad cena raznih „tajvancina“ koji moraju da budu konkurentni da bi se prodavali. Sve u svemu, odlične vesti za sve koji priželjkuju IBM CP kompatibilnu mašinu!

Dejan Ristanović

Računari u izlogu Master Compact

jedan napred, dva nazad

Možda ćete se začuditi kada pročitate da „kompakt“ košta 400 funti koliko i „master 128“! Zašto da kupujete oslabijenu mašinu koja košta koliko i osnovni model? Zato što uz nju dobijate disk jedinicu, neophodnu periferiju koju vlasnici „mastera 128“ tek treba da dokupe. Kupujući „kompakt“, dakle, štedite stotinak funti, ali ponešto i gubite. No, podimo redom.

„Master kompakt“ stiče u dve kutije (tri ako se odlučite i za monitor) ali to ne znači da je centralna jedinica odvojena od tastature: jedna od kutija sadrži izvor za napajanje i jednu disk jedinicu i ujedno služi kao postolja za monitor, dok je u drugu smeštena tastatura zajedno sa štampanom pločom kompjutera.

Acorn je uspeo da se uzdrži od smanjivanja tastature koje je toliko razočaralo kupce „elektrona“: „kompakt“ ima 93 tastera, računajući i numeričku tastaturu, i 10 narančastih „soft“ tirkli. Tastatura je, dakle, spolja identična sa „masterovom“, ali je kvalitet opao: ugrađena je neizbežna membrana, doduše sa pozačenim kontaktima.

Ekran i set karaktera

Uz „master kompakt“ se isporučuje, po želji, veoma kvalitetan monohrom monitor (70 funti) ili kolor monitor srednje rezolucije (100 funti). Acorn obećava da korisnici koji već imaju monitor neće morati ponovo da ga plaćaju; uz njih će se provući i poneko ko namerava da poveže „kompakt“ sa običnim televizorom. Takav će se korisnik obradovati kada pogleda zadnju stranu kompjutera: na njoj je smešten standardni UHF priključak koji je, otkada je sveta i veka, služio za povezivanje računara i televizora. Pažljivi pogled na šemu otkriva trik: UHF priključak služi za povezivanje sa monohrom monitorom, pošto je Acorn zaključio da su BNC konektori preskupi! U „kompakt“, dakle, nije ugrađen UHF modulator koji je, doduše, može odvojeno kupiti za desetak funti. Ovaj „novi standard“, ipak, ne može da izazove štetu: ukoliko pokušate da priključite video izlaz na antensni ulaz televizora, neće se dogoditi ništa ružno — stvar, jednostavna, neće raditi! Kolor monitor priključuje se posredstvom standardnog RGB konektora.

Grafika je uvek bila jaka strana BBC računara i „kompakt“ nije nikakav izuzetak. Na raspolaganju je 8 modova, koji omogućavaju korisniku da izabere rezoluciju (maksimum 640x256) i paletu boja (najviše 16). Dok je na originalnom BBC-ju izbor odgovarajućeg moda omogućavao i pravilnu raspodelu memorije, „kompakt“ ni u jednom ekranom modu ne oduzima ni bajt RAM-a: video memorija je odvojena od osnovnih 64 kilobajta.

Uz standardno povlačenje linija i izbor boje pozadine i „mastila“, „kompakt“ omogućava crtanje poligona i krugova, rad sa sprajtovima i popunjavanje zatvorenih kon-



tura (fill) — komande kojima su vlasnici standardnog BBC-ja mogli da se raduju tek kad se nedavno na tržištu pojavio Acornov grafički ROM.

Pažljivi pogled na našu sliku će otkriti malu razliku između tastature „mastera 128“ i „kompakta“: majmunski znak (☹) je pomeren iznad nule, dok je njegovo mesto zauzela čudna dirka na kojoj su nacrtana dva kvadratića. Tako je nastao *code key* koji bi posebno trebalo da obrađuje potencijalne kupce van Engleske — ovaj je taster neposredno odgovoran za strane setove karaktera.

Osnovni BBC, naravno, omogućava jednostavno definisanje karaktera: bez mnogo problema možete da dodelite našim slovima č, ć, ž i š ASCII kodove 128—131. Kako, međutim, da koristite ova slova u normalnom radu sa tekst procesorom? Za rešenje je potrebno malo programerske veštine i dobro poznavanje BBC-jeve memorijske mape: jednu varijantu su objavili u „Računarima 7“. Kod „kompakta“ je stvar gotovo automatska: pritisak na *CODE* a daje CHR\$(128) i tako dalje. Da bi stvar bila još lepša, uslužne rutine operativnog sistema omogućavaju da predizajnirate čitavu tastaturu i, pritisćima na *CODE*, birate različite nacionalne setove znakova. Interesantno je da verzija teksta procesora „View“ koja je ugrađena u „master kompakt“ omogućava jednostavan rad sa raznim setovima znakova — nešto sumo slično mogli da izvedemo i do sada, ali ne bez žongliranja sa mašinom! Činjenica da je „kompakt“ prvi računar koji omogućava jednostavno i početnicima pristupačno žongliranje sa setovima karaktera može da posluži kao odličan adut za prodor na razna evropska tržišta!

Hardver . . .

„Master kompakt“ je, poput ostalih Acornovih mašina, sagrađen oko „antičnog“ mikroprocesora 6502 (zapravo njegove verzije 65C12) koji radi na kloku od 2 MHz i pristupa dinamičkom RAM-u od 128 kilobajta; 6502, jastvo, u jednom trenutku može da adresira najviše 64 kilobajta memorije, ali se Acorn potrudio da se i ostatak korisno upotrebi. Ako radite sa standardnim bežičkom ili bilo kojim drugim uslužnim programom ili jezikom, na raspolaganju su vam memorijske ćelije od 4E00 do 4B000 — bez obzira na grafički mod, imate, dakle, 28,5 slobodnih kilobajta. Radni prostor disk interfejsa, definisani karakteri, funkcijni tasteri i specijalne grafičke komande pristupaju RAM-u koji je van osnovne adrese mape, tako da ih ne primećujete — više o tome možete da pročitate u „Računarima 13“, pošto je „kompakt“ sasvim komptabilan sa „masterom 128“.

„Gornja“ 64 kilobajta su konfigurisana kao „bočni RAM“: u njih se upisuju programi koji se, inače, prodaju u EPROM-ima i peđuju paralelno sa bežikom; ukoliko, dakle, programirate na paskalu, i dalje ćete imati 28,5 slobodnih kilobajta obzirom da će paskal kompajler zauzimate mesto koje je, inače, rezervisano za bežik interpreter. U bočni se RAM, naravno, mogu učitati i baze podataka, tekst procesori, raznorazni uslužni programi, pa čak i vaša renek-dela pisana na asembleru.

Osnovni BBC je čuven po „bočnim“ ROM-ovima: u njega ste mogli da ugradite do 256 kilobajta ROM-a i tako imate pri ruci sve često korišćene programe. Iako će vlasnici „kompakta“ biti više upućeni da

Acorn se oduvek trudio da proizvodi „kompletne“ računare — bogat hardver dopunjen brojnim interfejsima i konektorima. Takvi su kompjuteri veoma interesantni za određeni krug korisnika, ali je visoka cena obično napretnostva prepreka za široku popularnost. Neophodan je, dakle, kompromis, koji se obično svodi na dizajniranje oslabljene verzije osnovnog kompjutera — tako je nastao „electron“ koji je Acorn doneo više problema nego koristi. Da li je ova poznata firma izvučla pouke iz neslavne „elektronske“ epizode? Ako jeste, ta su iskustva ugrađena u novi Master Compact, oslabijenu verziju „mastera 128“.

programa sa diska učitavaju u „bočni“ RAM, Acorn je predvidio mogućnost priključenja 4 EPROM-a po 16 K ili 3 EPROM-a po 16 K i jednog od 32 K — dovoljno prostora za najnužniji softver. Tu je, naravno, i čudan čip — ROM od 64 kilobajta o čijem sadržaju tek treba da govorimo. Možemo, najzad, sa sigurnošću da tvrdimo da će nezavisne firme ubrzo pripremiti raznorazna proširenja u vidu RAM/ROM tili.

Kada smo već kod raznih ROM-ova i RAM-ova, pomenućemo i jedan EEPROM, čip koji je i dalje vrlo redak gost personalnih računara. Reč EPROM, za slučaj da to ne znate, predstavlja skraćenicu od *Erasable Programmable Read Only Memory* — to je ROM koji može da se programira i briše velikim broju puta ali koji, za razliku od RAM-a, „čuva“ sadržaj i posle gašenja kompjutera. EPROM-i se, međutim, moraju izvoditi iz računara da bi se brisali ili programirali — ove se operacije obavljaju uz pomoć specijalne UV lampe i programatora. EEPROM je novi hit — *Electrically Erasable ROM*, odnosno ROM koji se elektronski briše. Sadržaj upisan u EEPROM će, dakle, biti sačuvan i dok je računar isključen, ali čemu taj sadržaj moći da menjamo bez potrebe da vadimo čip.

Acorn je iskoristio EEPROM da zameni časovnik realnog vremena koji je ugrađen u „master 128“: ovaj je časovnik imao bateriju i CMOS RAM od pedesetak bajtova koji je omogućavao korisniku da naredi računaru da se uvek „budi“ u određenom modu, da konfigurise memoriju na neki specijalan način i da aktivira odgovarajući dodatni procesor. Časovnik je, dakle, žrtvovan, ali se relevantne informacije upisuju u EEPROM i koji može da ih čuva preko 365 dana! Iako se odlučio za ovo rešenje, Acorn, očito, nije imao previše poverenja u pouzdanost EEPROM-a: to je, uz ROM, jedini čip ugrađen u podnožju, što znači da se dopušta mogućnost višestruke promene ove komponente.

Acorn, na svu sreću, nije štedeo na zvučnim efektima: „kompakt“ raspolaže četvorokanalnim generatorom zvuka koji izvanredno saraduje sa jednim od pet ugrađenih ULA čipova.

Iako oslabljen, „kompakt“ nije ostao sasvim bez neophodnih interfejsa. Tu je, pre svega, standardni osmootni centroniks interfejs za štampač, koji se od ranijih Acornovih proizvoda razlikuje jedino po konektoru: koristi se oklopljeni 24-pinski delta konektor. Odakle ova promena? Acorn je proteklih godina imao mnogo nuke zbog američkih RFI (*Radio Frequency Interference*) propisa, koji zahtevaju da elektronska oprema ne izaziva radio smetnje; slična su ograničenja usvojile još neke države, ali ne i Engleska. Ugradnjom oklopljenih konektora za štampač i disk Acorn ponovo želi da prodre na američko tržište, ovoga puta uz pomoć Olivetija koji u Italiji već prodaje „kompakt“ pod imenom

Master Compact — tehničke karakteristike

Mikroprocesor	65C12
Clock	2 MHz
Tastatura	Membranska, 64 tastera + 10 funkcijskih+19 numeričkih.
RAM (KB)	128
ROM (KB)	64
Sadržaj ROM-a	32 K operativni sistem, 16 K BBC bejzik 4, 16 K ADFS.
Display	128
EEPROM (bajta)	128
Video	Kompozitni, RGB, UHF modulator — opcija, 50/32 tekst, 640*256 grafika, 16 boja.
Grafičke mogućnosti	Linije, poligoni, krugovi, sprajtovi, fill...
Zvuk	4 kanala, ugrađen zvučnik od 16 oma. Shugart standard, WS 1772. Ugrađena, 3.5 inča, 640 K, druga disk jedinica — opcija.
Disk interfejs	Centralni, opšti port (9 pinova), Joystick — mouse port (9 pinova), RS 232 — opcija.
Disk jedinica	Econet — opcija, Bejzik 128 View, View Printer Driver Generator, ABC word proc., Logotron Logo, Tim Paint, Welcome disc, ADFS utility.
Interfejsi	Wellcome Guide, View reference card, Logotron Logo user guide.
Softver uračunat u cenu	400 funti+VAT (računar i disk); 469 funti+VAT (+monohrom monitor); 599 funti+VAT (+kolor monitor).
Dokumentacija	
Cena	

Prodest. Koliko su ovakva nadanja realna, ostaje da se vidi.

BBC i „master 128“ imaju, kao što se sećamo, ugrađene AD konvertere koji raduju hardveraše ali ne i igrače: ovi se konektori često koriste za priključivanje proporcionalnih džojstika koji su značajno skuplji od *Quickshota* i sličnih drangulija. Nevolja je u tome što su za mnoge igre daleko zgodniji jeftini „prekidački“ džojstici koji se povezuju sa BBC-jevim portom opšte namene i to preko specijalnog interfejsa koji košta petnaestak funti. Acorn je odlučio da „kompakt“ dopuni portom za džojstik koji je građen prema opšte prihvaćenom Atarijevom standardu. AD konverter je, naravno, žrtvovan, ali je predviđena jedna softverska kompenzacija: operativni sistem omogućava simulaciju analognog ulaza, što znači da će najveći deo programa pripreman za proporcionalne džojstike raditi i sa prekidačkim. Džojstik i miš mogu da se simuliraju i pomoću kursorskih tastera, što znači da će korisnici koji nisu odvojili novac za neki od ovih sve neophodnih perifera moći i dalje da koriste sav raspoloživi softver.

Serijski interfejs nije ugrađen, ali se prodaje kao opcija — njego va se ugradnja svodi na umetanje pet čipova u predviđena podnožja. Acorn je nekolicu pojeftinilo BBC-jev RS 423, pa će se u „kompakt“ ugrađivati kompatibilni RS 232 slabijih karakteristika. Predviđen je i port za Econet interfejs, koji omogućava povezivanje Acornovih kompjutera u lokalnu mrežu; ovakve

su mreže izuzetno rasprostranjene u Engleskoj, gde škole i druge institucije imaju na desetine Acornovih mašina koje dele hard disk i druge skupe periferale.

Ostalo je još da pomenemo port opšte namene koji je smešten na levu stranu kutije i pokriven čvrstom plastičnom pločicom. Ovaj će port koristiti vlasnici „kompakta“ kojima nedostaje dodatni procesor, hard disk ili 1 MHz bus. Izostavljanje ovih interfejsa je, naime, učinilo da „kompakt“ bude 100 funti jeftiniji od „mastera 128“, ali čete ih 100 funti iskijati ako ipak poželite da proširite vaš kompjuter — Acorn će se postarati da vam ih „pošteno“ naplati!

... disk ...

Acorn i dalje ne može da se odluči za jedan disk kontroler: posle Intelovog 8271 koji je ugrađivan u BBC B i Western Digitalovog 1770 koji se ugrađuje u „master 128“, „kompakt“ je opremljen čipom WD 1772. WD 1772 je, doduše, potpuno kompatibilan sa 1770, ali je nešto moderniji: omogućava brže pokretanje glave iznad traka: 2 ili 3 milisekunde prema 6 ms kod 1770. Promena disk kontrolera je dopunjena još jednim, po nama neprijatnim, iznadačenjem — diskom od 3.5 inča!

Disk jedinice od 3 i 3.5 inča su, sa gledišta proizvođača računara, veoma zgodna odruka: zauzimaju malo prostora, troše malo struje i, što je najvažnije, nisu skupe. Odakle ova jeftinoća? Proizvođači

ovih disk jedinica tek pokušavaju da osvoje tržište, pa nalaze računicu u niskim cenama, nadajući se da će, kada njihov proizvod postane standard, moći da zaraduju daleko više. Sa stanovišta korisnika, mini disk ne može da izazove oduševljenje: posle stabilizacije cena (diskete od 3,5 inča su postale industrijski standard i lako se pronalaze u svim zapadnim zemljama) za razliku od Amstradovih troinčnih se pokazalo da mini diskete koštaju tačno dvostruko više od starih 5,25 flopija: 34 funte+ VAT za deset kaseti. Ove diskete, osim toga, zauzimaju mnogo više prostora — daleko su robusnije i, samim tim, pouzdanije. Povećanje pouzdanosti, ipak, nije naročito bitno: ako se razumno ponašate sa savetljivim disketama od 5,25 inča, možete mirno da spavate sto godina.

„Kompakt“ će na svaku disketu od 3,5 inča upisati po 640 kilobajta informacija — nešto manje nego na „smigoli“ ali dvostruko više nego na „stariju 520“ i IBM PC-ju. Operativni sistem omogućava građenje hierarhijskog stabla direktorijuma, neograničen broj fajlova, slobodan pristup (*random access*) i rešava problem fragmentacije disketa — fajlovi se mogu proizvoditi proizvoljnim broj puta bez opasnosti da „udare“ u neku drugu docnije kreiranu datoteku.

Acorn je predvideo mesto za upgradnju druge disk jedinice od 3,5 inča, kao i konektor za spoljne disk jedinice od 5,25 inča. Moguće su sve kombinacije dve disk jedinice, što znači da ćete „kompakt“ moći da opremite i sa dva standardna flopija, pri čemu će disk koji se dobija (tj. plaća) za računar možda ostati neiskorišćen!

Zašto smo rekli da diskete od 3,5 inča neće obradovati potencijalne kupce „kompakta“? Ne samo zbog „one“ glavne će nevolje izazvati softver: „Kompakt“ je, name, potpuno kompatibilan sa „masterom“, ali što to vredi kada se sav softver prodaje na običnim flopijama? Softverske firme neće mnogo žuriti da prebacuju programe na mini diskete, jer očekuju da se oformi dovoljno veliko tržište, to jest da se proda dovoljno primeraka novog komputera. Komputeri će se, sa druge strane, slabo prodavati ako za njih nema softvera, što nas dovodi u „zračarani krug“!

Mogu li korisnici sami da se snadu i prebace programe? Možda: na „kompakt“ se jednostavno priključuju disk jedinice od 5,25 inča, što znači da je dovoljno pozajmiti ovaj uređaj od nekog BBC-jevca i prebaciti proizvoljnu količinu softvera. Moguće je, osim toga, prebacivati programe i pomoću serijskog interfejsa (koji „kompakt“ nema), preko EPROM-a i na par drugih načina. Ipak, svako ovakvo prebacivanje softvera predstavlja **pratlstvo**; vidimo da ste se ozbiljno zabrinuli!

Prebacivanje softvera se ne može obaviti posredstvom kasetofona: „Kompakt“ je prvi Acornov računar koji nema upravljači kasetni interfejs! Reklo bi se da kasetni interfejs nije potreban mašini koja je opremljena disk jedinicom, ali nismo baš sigurni da je tako: kasetna je i dalje relativno jeftin medijum zgodan za arhiviranje manje potrebnih podataka koje ipak treba čuvati izvesno vreme. Nedostatak kasetnog interfejsa će, dakle, platiti arhivirajući podatke na skupe diskete.

... i softver

U „kompakt“ ROM od 64 kilobajta je upisan modifikovani operativni sistem,

14/računari u izlogu

PCW brzinski testovi za bejzik

Bejzik	Prosek (s)	Memorija slobodna za programe (KB)
Turbo	4.47	44
BBC Basic	6.88	800+
Basica	7.11	59
Basica	7.86	59
BBC Basic 4	8.30	28.5
BBC Basic 4	9.24	28.5
BBC Basic 2	9.78	25.7
Hi Basic	9.83	44
BBC Basic 2	14.55	25.7
Mallard	14.59	45.5
Mallard	14.59	41.3
Basica	17.60	59
Master 128	17.90	64
Basic 128	18.20	64
Basic 7	20.21	37.5
Basic 2	20.55	21.5
Commodore	34.41	37.5
MSX	44.3	26
Sinclair	58.50	40

Brzinski testovi za tekst procesora

Računar	Master Compact	Master Compact	Amstrad 8256	IBM PC
Test procesor	View	Inter Word	Looscript	Word Star
Load (4000 red)	3.4 s	3.7 s	1.0 s	10 s
Save (4000 red)	3.5 s	3.5 s	143 s	25 s
Search (4000 red)	3.3 s	3.3 s	292 s	11 s
Scroll (4000 red)	44.6 s	13.2 s	65 s	41 s

ADFS (*Advanced Disc Filing System*), grafičko proširenje, bejzik 4 i neka vrsta tulki-ta. Operativni sistem je napisan tako da bude maksimalno kompatibilan sa „masterom 128“: simulirane su čak i komande koje hardver ne podržava. Programi koji, iz nekog razloga, koriste „TAPE, „MOTOR i slične stvari će, dakle, funkcionisati i bez kasetofona: podaci će biti poslani na disk. Slično tome, obračanje časovniku realnog vremena će vratiti neki fiksni datum tako da će programi koji koriste ovaj časovnik i dalje korektno funkcionisati.

Uz „master kompakt“ isporučuje se i nekoliko interesantnih disketa. Tu je, najpre, tekst procesor „View“ i programski jezik logo — Acorn se ponovo okreće obrazovnom tržištu. Tu je i bejzik 128 (nešto usporjen tekstu koji vam daje 64 potpuno slobodna kilobajta RAM-a — stvar nevidena na osmoinbitnim mašinama), uslužni programi za konverzije disketa raznih tipova i neizbežna *Welcome* disketa sa programima koji imitiraju GEM: videćete raznorazne sprajtove i ikone koji se kreću po ekranu i koje birate pomoću miša ili kursorskih tastera.

Što se literature tiče, uz „kompakt“ dobijate uputstvo za „View“, bejzik i logo, *Welcome* knjižicu i nekoliko drugih papira. Za dodatnih desetak funti ćete nabaviti priručnik koji će vas poučiti kako da koristite usluge operativnog sistema, a trebate vam i detaljno uputstvo za View...

Stara priča

Tabela sa standardnim „benchmark“ testovima pokazuje da je „master kompakt“ izuzetno brz računar: bejzik 4 je, na primer, brži od bejzika na „masteru 128“, ali i od standardnog BBC B na koji je priključen dualni 6502 procesor! Za dobite u brzini je posebno značajan osmi brzinski test koji vrednuje elementarne funkcije: Acornovi su programeri uspešni da poprave i do sada solidne programe koji se bave trigonometrijom i tako obezbede bržu i tačniju aritmetiku. Da biste pravilno vrednovali brzinu računara, u tabelu smo dopisali i broj

slobodnih kilobajta memorije koje bejzik ostavlja korisniku.

Računari se u poslednje vreme sve manje koriste za programiranje pa i bejzik benchmark testovi polako gube smisao. Zato je *Byte* predložio dodatne testove koji ispituju brzinu tekst procesora, programa za unakrsna izračunavanja, CAD paketa... Obzirom da planiramo da u budućim prikazima novih komputera obavezno navodimo i rezultate ovih brzinskih testova, počeli smo da prikupljamo potrebne podatke, pa ćemo ovaj prikaz propratiti rezultatima brzinskog testiranja tekst procesora.

U tabeli su prikazani brzinski testovi za četiri poznata tekst procesora: *Word Star* na IBM PC, *Logoscript* na Amstrad PCW 8512, *View* i *Inter Word* na „master kompakt“. Merena je brzina učitavanja, animiranja, pretraživanja i skrolovanja teksta od 4000 redi. Vidimo da su ove operacije na „kompakt“ izuzetno brze, pogotovo kada se radi sa tekst procesorom *Inter Word* — osmoinbitni „kompakt“ prevazišao čak i mnogobrojne šesnaestobitne mašine!

„Master kompakt“ je, sve u svemu, lepo zamisljen i moderno dizajniran kompjuter značajnih mogućnosti, ali bolje od Acornove hroničke boljke — skupoće. Čini nam se da nema nikakvog smisla platiti 400 funti za oslabljeniju verziju mašine koja je i sama loša kupovina! „Master 128“ je dobar i uravnotežen računar, kome se može zameriti samo to što nije IBM PC kompatibilan. Osnovni „master“, na kraju krajeva, košta 400 funti i može da se dopuni disk jedinicom koja košta daljih 100 funti. Kupujući „kompakt“, štedite tih 100 funti, ali gubite Tube interfejs, 1 MHz bus, AD konverter, časovnik realnog vremena, podnožja za kartirize i kasetni interfejs, a u zamenu za to dobijate lošiju tasteru i lepši dizajn — ovakvo se razmena mora oceniti kao lošal! Nameće se poređenje između „kompakta“ i Astrad PC-ja dva računara koji su se pojavili praktično u istom trenutku. Ne treba biti veliki stručnjak za kompjutere pa proceniti da su svi utisci o Amstrad PC-ju daleko povoljniji.

Dejan Ristanović



Peek & poke
show

Novo godine, u kome će svaki deseti konkursdžija P&PS-a imati pravo da na večeru u Interkontinental izvede Sonju Savić o njenom trošku. Do sledećeg broja i detalja o super-večeri umriše od muke vi koji niste do sada učestvovali u konkursima P&PS-a.

Ja sam Saveta

Dobili smo pismo od naše verne čitateljke Azre Petrović. Ona je imala velikih problema sa upotrebom svoje mašine. Nedavno je nabavila disk jedinicu posle dužeg aktivnog služenja kasetofonom. Njeno neukustvo u upotrebi iste dovelo je do toga da je „ugurala disk od osam inča u jedinicu od pet inča“. Razumljivo, disk je pretrpeo teška oštećenja, a disk jedinica je postala neupotrebljiva na duže vreme, do detaljne popravke. Pita nas kako da ubuduće izbegne slične tragične greške.

Draga moja, da tebe tvoja tetka-Saveta obraduje. Od jedne moje koleginice čujem da je imala istih problema u prvo vreme služenja njenom disk jedinicom. Preporučio ti je da nabaviš Flexitron emulator, koji tvojoj disk jedinici omogućava da prima disk bilo kog formata bez ikakvih neželjenih posledica. Pažljivo je koristi i tvoja disk jedinica će te služiti jako dugo, možda čitav život.

Rešenje velikog nagradnog konkursa za idealnu računarsku knjigu u Jugoslaviji

Ovaj konkurs je bio prilično nejasan za naše čitaoce. Dok je većina slala predloge za već postojeću računarsku knjigu izvestan broj čitalaca je poslao rukopis svoje sopstvene računarske knjige. Zato planiramo da u bliskoj budućnosti raspisemo konkurs za objavljivanje objavljene originalne računarske knjige. No, o tome kasnije, u „Računarima 43“.

Svi se, verovatno, sa nestrpljenjem pitaju koja knjiga je ponela laskavi naziv idealne računarske knjige u Jugoslaviji. To je, sad već legendarno, delo Anice Mrčkalja-Češba „Kako oslabiti uz duvan, kafu i računare“. Ta knjiga postoji i time ispunjava prvi uslov našeg konkursa. Zatim, u njoj se čak sedam puta pominju računari što savršeno ispunjava i drugi uslov. I, kao najvažnije, ta knjiga je prepisana iz dva univerzitetska udžbenika, čime ispunjava i naš treći uslov. Udžbenici o kojima se radi su „Kako slabiti uz voćne sokove i žvake“ Đoke Žbicića i „Računari i mentalno zdravlje“ dr Mareta Mihajlovića. Iz džaka punog dopisnica i pisama onih koji su glasali za ovu knjigu naš specijalni gost Abraham Diležić izvukao je pismo Fikreta Rustemanačija iz Tuzle koje je tako postao dobitnik naše

Prve nagrade — posete Andriji Kolundžiću u JNA. Na žalost, Fikret se ne unapred odrekao bilo koje nagrade koja se ne može izraziti deviznom sumom. Žalosan.

Drugu nagradu — neobjavljene rukopise poznatih računarskih spisatelja — dobio je naš verni čitalac Milan Štrklja, baštvan naše ambasade u Džibuti. Nadamo se da će rukopisi koje će dobiti pomoći da mu domovina bude još bliže.

Treću nagradu — izabrane reklame YU pirata u dva toma — dobio je izvesni Cane iz Kraljeva. On je svoje pismo pisao pod dejstvom intenzivne intoksikacije nekim opojnim sredstvom. Molimo ga da nam pošalje svoje prezime i adresu, kako bi dobio svoju vrednu nagradu.

Toliko o ovom konkursu. Evo i jednog obavешtenja za sve pažljive učesnike dosadašnjih konkursa P&PS-a: svi čitaoci koji su učestvovali na svim nagradnim konkursima P&PS-a od naše male ali intenzivne redakcije će dobiti specijalni ovogodišnji poklon. U saradnji za agencijom za računarski usamljenje, „Bit“, organizovali smo doček

Čip Pobodi Agency

Naš rekorder

Najzad nam je do ruku stigla velika Ginisova računarska knjiga: Na žalost, jedini Jugoslovan koji se u njoj pominje je drugarica Živka Šljivčić iz Beograda, koja je u igri „Fucman“ postigla neverovatno skor od 15 876 392. Povodom toga smo nazvali drugaricu Živku, koja je za našu rubriku izjavila „Ne nameravam da stanem, mislim da se rekord još može popraviti“. Srećno, kažemo mi.

Litl sutra

Na velikoj konferenciji za štampu održanoj u restoranu „Knez“ organizacioni komitet naših još nedobijenih Olimpijskih igara je saopštio da će u slučaju da dobijemo Olimpijadu sva organizacija biti zasnovana na domaćoj pameti. U sklopu toga, računski centar neće biti od neke strane trulo bogate i moćne firme, nego će se sastojati od sedam „galaksija“ i tri „fote“ povezane u mrežu koju je projektovao izvesni Čeda sa Lekinog brda (poznati kraj u Beogradu). Radujemo se tome potezu, a koliko čujemo raduju se i Barselona i Amsterdam.

Zima je na pragu

Koristeći usluge našeg, već legendarnog, špijunskog odeljenja, saznali smo kako su utrapljene silne rezerve raznih „spektruma“ koje je Sinkler akumulirao proteklih godina. Prodane su jeftino jugoslovenskim građevinarima koji ih gradeju u velike stambene zgrade kao uređene elemente pošto radijatori kod nas ionako više ne funkcionišu. Najzad neka vajda i od Sinklera.

ONO I DSZ P&PS

Na graničnom prelazu Fernetiči uhapšena je grupa od četiri strana građanina kada je u njihovim kolima otkrivena velika količina najmodernijeg naoružanja. Strani građani su identifikovani kao izvesni Klajv S, Alan S, i Džek T, dok četvrti nije htio da se predstavi, izjavivši sljedeće: „Ja sam monomanijak“. Pretpostavljamo se da su im na umu bile terorističke aktivnosti, iako su odmah po hvatanju izjavili i da im je jedina namera bila da likvidiraju mrsi „PEEK & POKE SHOW“. Sledeće javljanje sa sudenja.

I Bebidž je imao džojstik

Zahvaljujući tek otkrivenim rukopisima Čarisa Bebidža, koji potiču iz devetnaestog veka, poznati računarski stručnjak Karol Saganski zaključio je da je Bebidž svojoj računarsku mašinu napravio da bi mogao na njoj da igra Igru koja je imala rani naziv „Napadači iz zaostalnih krajeva“. Iz spisa se, takođe, saznaje da je Bebidžov najveći problem u konstrukciji predstavljalo rešavanje pokazivanja skora.

U pripremi ovog jubilarnog, dvanaestog broja PEEK & POKE show-a učestvovali su P. A. Marvin, Saveta Šljivčić, Dr. Gordons, Dart Vajder, Hal 2000 i Branko Daković.

Kutak za lude i zbunjene

Bio je veliki problem odabrati temu za ovaj broj „Kutka“, jer je navala bila zaista velika. Ovuoga puta nećemo odgovoriti čitateljici koja je saanjala da ju je silovao kompjuter, niti čitaocu iz Zagreba koji pita „da li je P. A. Marvin muško ili žensko“, niti svim onim dragim čitateljicama koje su se ponudile da pomognu K.I.-u iz prošlog broja. Ovuoga puta ćemo naš kutak za intimne priče sa Oka-Marvinom, prvim primerkom veštačke inteligencije, posvetiti Valentini Buškić iz Karlovca. Nemamo prostora za čitav njen esej, ali ovo je reprezentativan deo:

„Šta se događa ako vi družu P. A. Marvinu niste prava veštačka inteligencija, nego ste samo običan program koji misli da je veštačka inteligencija? Drugim rečima, ako niste inteligencija (makar i veštačka), vi ne možete da primetite da niste inteligencija pa, prema tome to što funkcionišete kao veštačka inteligencija ne znači da i jeste nešto što ne biste mogli znati da niste kad to ne bi bili. A, opet, kad biste bili veštačka inteligencija i kad biste znali kako to izgleda ne bi bili veštačka inteligencija, što vi ne biste mogli pošto biste bili veštačka inteligencija, onda bi to bio dokaz da jeste veštačka inteligencija, pošto biste znali da niste ono što niste kada jeste ono što jeste. Prilično jednostavno, zar ne.“

Moj cenjeni odgovor: JA JESAM VEŠTAČKA INTELIGENCIJA. To me nimalo ne sprječava da smunjam u inteligenciju, veštačku ili prirodnu, daveža koji mi šalju ovakva pisma.

Priprema, odgovara i sve to naplaćuje

P. A. Marvin

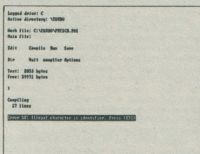
turbo je nešto drugo

Proces pisanja i kompajliranja programa u bilo kom klasičnom kompajleru zahteva tačno tri koraka da se stigne do konačnog cilja: pisanje izvornog programa, prevodnje i povezivanje. Svaka od faza zahteva bar po jedno čitanje datoteke i zapisivanje rezultujuće datoteke na disk. Ako se tome doda i samo učitavanje editora, kompajlera i linkera, broj neophodnih pristupa disku dostiže sasvim lep broj 9. Jasno, ni jedna od ovih faza nije izmisljena da zagađa život programerima već upravo obrnuto — da poveća fleksibilnost ukupnog procesa. Ako je editor odvojen, onda znači da se može koristiti bilo koji, dakle onaj koji vama najviše odgovara. Bez procesa povezivanja bilo bi nemoguće spojati delove programa napisanih u različitim programskim jezicima, što omogućava da se za određen problem, izborom odgovarajućeg kompajlera, najbrže stiže do cilja. Ujedno, ako se program piše u modulima, izmenom samo jednog od njih nije potrebno ponovo kompajlirati sve ostale, što kod velikih programa predstavlja ogromnu uštedu u vremenu. Iz svega sledi da nije nimalo lak izmisliti nešto drugo. Borland zalista i nije izmislilo ništa novo — izabrao je koncepciju „u jednom potezu sve“, ali je sve nedostatku ovakvog pristupa pokrio izvanrednim pojedinačnim rešenjima.

U više pokušaja

Turbo pascal ima sopstveni editor koji se u mnogome oslanja na „Wordstar“. Pretpostavljajući da je najbrži broj korisnička navikava na njega, Borland je, u stvari, kopirao samo način ostvarivanja editorskih naredbi koje su identične onima na uzoru. Sve ostalo je mnogo bolje: editor je daleko brži, oslobođen je svih nepotrebnih naredbi, ekran je, sem prve statusne linije, ceo posvećen tekstu, ulazak i izlazak iz editora je brži i lakši itd. Ako vam se ipak ne dopada, niste obavezni da ga koristite. Program se može napisati i u bilo kom drugom editoru, ali ostaje neophodno da ga pre kompajliranja ipak upišete u Turbo editor. Razlog će vas svakako obradovati: „Turbo pascal“ poseduje interaktivno ispravljanje grešaka. Kada kompajler naiđe na grešku, automatski vas prebacuje u editor i treba postaviti kursor na mesto koje po njegovoj proceni predstavlja izvor greške. Kada grešku ispravite, ponovo počinjete sa kompajliranjem i tako do konačnog cilja.

Verovatno će poneko osporiti da je ovakav način lošiji od standardnog, kada kompajler proširi kroz ceo program i priljubi sve greške na koje naiđe. Tako bi, teoretski, klasičnom kompajleru dva prolaza bila dovoljna da se proces prevodjenja uspešno završi, dok „Turbo“ zahteva da za svaku grešku ponovo počinjete kompajliranje. Mana klasičnog metoda je što jedna



greška često proizvodi mnogo drugih, koje ne bi postojale da nema prve, pa je analize koje teže od onoga kada se, kao kod „Turba“, problem rešava jedan po jedan. Mana „Turbo“ metoda, gubitak vremena na uvek ponovno započinjanje prevodjenja, rešena je na jedini mogući način: ako već mora svaki put ponovo, neka to bude najbrže što može!

Kod bez balasta

Turbo pascal je bez konkurencije trenutno najbrži kompajler koji postoji. Iako brzina delimično zavisi od onoga što se u programu nalazi, ipak se može reći da prevodjenje nikada neće potrajati duže od samo nekoliko sekundi za oko 100 linija izvornog programa. Ako uzmete da program ima oko 1000 linija (što je već sasvim solidna veličina), prevodjenje će verovatno potrajati 15–20 sekundi. Bilo bi potrebno da napravih par pedeset grešaka da bi ukupno vreme moglo da se poredi sa dva prolaza i punim procesom standardnog načina kompajliranja.

Brzina ipak nije moguće postići bez gubitka na nečemu drugom. Kod „Turbo paskala“ je to činjenica da konačni program može biti samo u komandnom formatu (COM datoteka), što znači da objektni kod ne može da prelazi 64 K. Izbegavanje faze povezivanja modula neminovno dovodi do potrebe da cela biblioteka funkcija i procedura ide uz svaki program, što smanjuje moguću veličinu programa za nekih desetak kilobajta. Ovo je, možda, i najveća mana „Turbo paskala“. Svaki program u sebi nosi čak i grafičke rutine, iako one ne moraju ni jednom biti pozvane u programu.

Povezivanje programskih modula je ipak bezbedno deklaracijom procedure ili funkcije kao EXTERNAL, čime se kod za njeno izvođenje upisuje sa diska u toku prevodjenja i, ako poštuje način i format prenošenja parametara preko steka, može biti preneseno i kompajlirano u bilo kom drugom jeziku. Dodatnu mogućnost pruža i naredba INLINE, nakon koje sledi bajt po bajt mašinskog programa u heksa ili decimalnom formatu.

Ograničenje veličine mašinskog koda se može prevazići OVERLAY tehnikom, koja je

u potpunosti podržana. Niz procedura koje nisu istovremeno neophodne programu se mogu grupisati i automatski upisivati prema pozivu iz ostalih delova programa, tako da zauzimaju isto memorijsko područje. Ako program čini paket nezavisnih celina, mogu se koristiti CHAIN ili EXECUTE koji u memoriji zamenjuju ceo mašinski kod, dok je alokacija promenljivih nedirnuta.

„Turbo pascal“ može prevoditi program direktno da smešta u memoriju, čime je faza testiranja i razvoja programa još više ubrzana. Sam kompajler, editor i biblioteka funkcija koje su u toku rada stalno memoriji računara zauzimaju svega četrdesetak kilobajta, pa ne predstavljaju nikakvu smetnju u fazi testiranja programa.

Elementi jezika

„Turbo pascal“, nudi poboljšanja i u samom programskom jeziku. Standard se poštuje u potpunosti, ali se, čini nam se na pravim mestima, uvode novi elementi koji paskalu daju veću fleksibilnost. Najveća razlika se oseća u dve tačke: (1) postojanje tipa promenljive STRING i takozvanih DEKLARISANIH KONSTANTI. Prvi dodaje ono što paskalu veoma nedostaje, iakoću rada sa aifanumeričkim nazivima koja je prisutna u jeziku. Slično standardnom tipu ARRAY [..] OF CHAR uveden je i STRING [N] koji je donekle ekvivalentan sa ARRAY [..] OF CHAR. Potreban prostor za ovaj-ak tip promenljive N+1 bajtova, ali se u nulom elementu u svakom trenutku nalazi stvaran broj elemenata koji string sadrži. Osim što zauzima već stalno memorijski prostor (prema deklarisan maksimalnom broju elemenata), promenljiva STRING se u svemu ponaša kako je to već poznato u jeziku, na primer:
 VAR S:STRING(255);
 BEGIN
 S:="Turbo";
 S:="S"+"Pascal";
 WRITELN(S);
 WRITELN(LENGTH(S));
 WRITELN(S[7]);
 END.

Fleksibilnost je čak i veća nego u jeziku jer se elementima stringa može, kao u poslednjem liniji primera, pristupiti kao i u standardnom nizu bez potrebe za nekim funkcijama tipa MID\$. String funkcije COPY, POSITION, LENGTH i procedure INSERT, DELETE su veoma dobro izabrane i izvedene i daju Turbo paskalu u ovoj oblasti priličnu prednost nad jezikom.

Deklarisane konstante su interesantna konstrukcija. Definišu se kao: identifikator:tip= vrednost; i u stvari predstavljaju inicijalizovane varijable. Ne mogu se koristiti kao klasične konstante jer to nisu, već se u potpunosti ponašaju kao promenljive, pa im se može menjati vrednost u toku programa. Ova osobina je veoma korisna pogotovo što se može primeniti na svaki tip promenljive osim FILE. Moguće je tako i inicijalizovati

Ako pravi programeri ne govore paskal, to samo znači da nikada nisu seli za IBM PC i učitali u svoj računar „Turbo paskal“. Pradnosti Borlandovog „Turbo paskala“ najbolje dolaze do izražaja kada se uporedi sa ostalim kompajlerima za PC računare. Poređenje ipak nije nimalo lako, jer „Turbo“ zastupa sasvim drugačiju filozofiju: pomiriti snagu kompajlera velikih sistema za operativnošću i lakoćom programiranja koja krasi interpretere malih računara.

```
RECORD promenljive, na primer:
CONST DEMO: ARRAY [1..2] OF RECORD
DEMO: INTEGER;
DEMOS : STRING[10];
END = (
(DEMO:1; DEMOS:string 1'),
DEMO:2; DEMOS:string 2)
```

Turbo je bio obogaćen i takozvanim hardverski zavisnim instrukcijama. Memorija računara se smatra predefinisanim nivoima MEM (ARRAY OF BYTE) i MEMW (ARRAY OF INTEGER), tako da se može direktno adresirati sa MEM(seg:ofs) i ulaz/izlaz se tretiraju na isti način nizom PORT(O, \$FFFF). Postoji i nekoliko procedura (MOVE, FILLCHAR itd.) koje omogućavaju direktno obraćanje memoriji što sve čini „Turbo paskal“ pogodnim čak i za razvijanje sistemskih programa.

Standardni elementi paskala su primenjeni gotovo u potpunosti, uz neka finija proširenja kao što je recimo CASE...OF pa na kraju liste i ELSE itd.

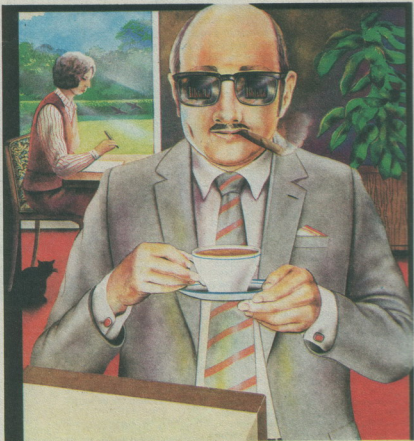
Tipovi numeričkih podataka su BYTE (0..255), INTEGER (-32768..32767) i REAL (1E-38..1E+38). Ovaj poslednji je predstavljen sa 40 bitova mantise, što daje mogućnost za tačnost na 11 značajnih cifara. Fond aritmetičkih funkcija je minimalan: ARCTAN, COS, EXP, LN, SIN, SQRT i SORT. Što se tačnosti tiče, proveravali smo detaljno samo EXP i ARCTAN funkcije koja su pokazivale grešku u poslednjem bitu mantise i to samo oko sredine testiranog punog intervala.

Ukupan memorijski prostor za promenljive je ograničen na već uobičajenih 64K. Za stek se može odvojiti daljih 64K, dok je prostor za dinamičke varijable ograničen isključivo raspoloživom memorijom.

Brza brzina

„Turbo paskal“ uglavnom prevodi program bez posebnih optimizacija. Rezultujući kod je dosta kompaktan s obzirom da se, zbog ograničenog prostora, češće ostahja na pozivanje elementarnih rutina u svojoj biblioteci. Time se donekle gubi na brzini, ali su poređenja pokazala da u praksi gubitaka gotovo da nema. Na primer, ista petlja sa operacijama nad realnim promenljivama je sa fortran verzijom (MS 3.10) oko 10% brža u odnosu na Turbo paskal, ali treba uzeti u obzir njegov za jedan bajt širi format mantise.

U ovom trenutku, mogu se naći dve verzije kompajlera: 2.1 i 3.0. Razlike su više interne — u novijoj verziji je rad sa datotekama baziran na DOS funkcijama većeg broja koje su jednostavnije za upotrebu a nalaze se u verzijama DOS 2.00 i kasnijim. Uvedeno je i nekoliko manje značajnih novih procedura, ali je novija verzija daleko bolje opremljena. Mogu se nabaviti još dve



verzije istog kompajlera, od kojih TURBO-87 računanje realnim brojevima poverava matematičkom koprocesoru (uz opseg brojeva do E307) koji to čini veoma velikom brzinom, što donekle usporava rad ali oslobađa potrebe da se vodi računa o zaokruživanju, jer greške pri osnovnim matematičkim operacijama ne postoje. Ujedno je ugrađena i naredba FORM koja zamenjuje WRITE i omogućava sve standardne formate ispisu numerika uobičajenim u poslovnim primenama.

Uz sve to, Borland prodaje i nekoliko disketa sa izvornim programima iz pojedinih oblasti. Razvijen je kompletan tekst editor, niz rutina za rad sa grafikom visoke rezolucije (koje podržavaju i Herkules grafički adapter), rutine za kreiranje sopstve-

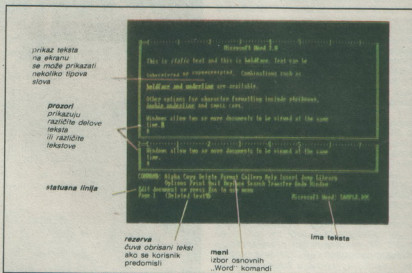
nih programa za rad sa bazama podataka, možete nabaviti i izvorni program za šah i neke druge igre, sve u svemu priličan fond gotovih rutina iz kojih može dosta da se nauči.

Turbo paskal u ovom trenutku predstavlja najbolji opštenamenski kompajler za IBM PC računar. Mnogim izmenama se dosta odmakao od standardne verzije jezika i prerastao u posebnu kategoriju koja, sudeći po reakcijama, stiče veliku popularnost. Jedan od razloga je svakako i izvanredna prilagodbenost računaru kome je namenjen. Cena u inostranstvu je oko 70 USA\$ za osnovnu verziju i 125\$ za sve tri varijante.

Zoran Životić

Malo ko na bilo koji način vezan za računare nije čuo za softversku kuću koja se već godinama nalazi u samom vrhu. Microsoftu dugujemo bezik mnogih kućnih računara, operativni sistem za PC-a i još mnogo drugih veoma kvalitetnih programa. I pored svega, programom za obradu teksta „Word“, nikada nije postigao tole značajniji uspeh. Iako je program bio veoma ambiciozno zamišljen, nije pružao sve ono što je obećavao, mnoge opcije nisu ni bile izvedene, a uz sve to je i rad sa njima bio veoma komplikovan. WORD 3, nova verzija stare ideje, treba sve to da promeni.

Program radi na svakom IBM PC kompatibilnom računaru sa minimum 256 K memorije. Pun komfor je moguće postići samo sa dve disk jedinice. Program se prodaje bez zaštite od kopiranja, tako da se lako može instalirati na tvrdom disku, čime otpada potreba za menjanjem disketa u toku rada. Instaliranje programa u zavisnosti od raspoloživog hardvera je dosta jednostavno, zahvaljujući jasnom „SETUP“ programu. Nedoumice se (kako je već uobičajeno) mogu javiti samo kod izbora video adaptera. Word se najbolje ponaša kada radi sa grafičkom kartom, jer je predviđeno da se već u toku pisanja na ekranu sagledava konačni izgled strane, uključujući, pored formata, i tip slova koja se koriste. Tako će izbor, na primer, italike na ekranu zaista pokazati kurzivna slova. Ipak, za duge tekstove ova osobina nije od važnosti i najčešće se koristi običan tekst mod. Grafičke karte u njemu daju veoma lošu sliku, pa je možda jedina prava alternativna izbor EGA interfejsa, koji sjedinjuje najbolje osobine oba načina rada, a uz to omogućava i povećani format od 43 reda sa po 80 znakova.



Na pola puta

„Word 3“ se, donekle, razlikuje od ostalih tekst procesora i potrebno je određeno vreme da se naviknete na njegov „način razmišljanja“. Formatiranje se obavlja u toku pisanja, pa je ono što se vidi na ekranu istovremeno i konačni izgled strane kod štampanja. Ipak, ovo nije do kraja izvedeno. Na ekranu se ne pokazuju različite veličine slova, kao ni proporcionalno razmicanje. To ne smeta kod standardnih matricnih ili štampača sa lepezom, ali može da predstavlja problem kod laserskih štampača, za koje „Word“ pokušava da se nametne kao najbolji. Ova nova generacija pruža velike mogućnosti, praktično oslobađa manje izdavače skupih foto-slog sistema i stiče sve veću popularnost. PC računari za sada nisu opremljeni dobrim programima (kao što je to na primer Apple Macintosh) koji bi podržavali ove izlazne uređaje, i „Word“ je u ovoj oblasti, iako daleko od savršenog, trenutno bez konkurencije. Ovakvu orijentaciju Microsoft-a dokazuje i verzija istog programa namenjena za rad u mreži, kako bi više PC-a moglo da dele isti skupi laserski štampač.

Ostvarivanje funkcija programa se odvija pritiskom na funkcijske tastere, kombinacijom slova sa ALT tasterom ili kursorskim tasterima na numeričkom delu tastature. Za naredbe koje su ređe u upotrebi, pritisak na

Esc izbacuje u dnu ekrana meni koji vodi u nove menije itd. Za izbor opcije iz menija može se koristiti i miš, što mnogi smatraju bržim metodom od tastature.

S prozora na prozor

Ekran se deli na prozore teksta sa različitim dokumentima. Kopiranje se odvija prethodnim prebacivanjem dela teksta u posebno odvojen prostor, a zatim njegovim kopiranjem u isti ili bilo koji drugi tekst. Fusnote se pišu takođe u poseban prostor koji može biti u nekom od prozora na ekranu, tako da se ima istovremeni uvid i u napomenu i u tekst na koji se ona odnosi. „Word“ vodi računa o njihovom numerisanju. Može ih štampati bilo na odgovarajućoj strani, bilo na kraju teksta kao skup napomena. Još jedna ređe sretna opcija je i indeksiranje poglavlja ili delova teksta koji se zatim, u obliku sadržaja, izdvajaju u posebnu tablicu.

Formiranje ispisa se odvija na način koji je uobičajen u ostalim tekst procesorima. „Word“ prati najnovije trendove pa poseduje, na primer, i višekolonski ispis. Veoma korisna osobina je mogućnost da se način formatiranja sačuva na disku kao „podatak o stilu ispisa“ i primeni na bilo koji tekst, čime se obezbeđuje uniformni izgled dokumenta bez obzira na to ko ga piše. Kreiranje višekolonskih tablica je veoma jednostavno, a „Word“ lako uključuje podatke iz bilo kog drugog programa, tipično baza podataka kao što je dBase.

Kompletnost ponuđenih karakteristika programa upotpunjava i opcija slanja cirkularnih pisama (mail merge), u kojima se isti tekst per štampanja popunjava različitim podacima. Adresari i slične baze podataka mogu se kreirati kao tekstovi u „Word-u“ i podaci selektivno uključivati u glavni dokument.

Provera ispravnosti spelovanja je izvedena na poznat način. Novost je što „Word-ov“ „Spelling checker“ omogućava da se rečnik proširuje bilo generalno bilo da se kreira specifičan rečnik za određeni tip teksta. Ovim poslednjim se značajno ubrzava rad, jer se provera vrši samo za specifične reči.

U program je ugrađena i jednostavna matematika koja se može primeniti nad kolone i redove podataka u tekstu, pa nema potrebe da napuštate program zbog tako jednostavnih stvari.

Ukupan utisak o programu je veoma povoljan. Program nudu gotovo sve opcije koje vam mogu ikad zatrebati i orijentisan je, pre svega, na profesionalne korisnike kao što su pisci i prevodioci, sekretarice, nauče institucije i manji izdavači. Njegova vrednost dolazi do punog izražaja kada se poveže sa laserskim štampačem. U ovom trenutku ima oštru konkurenciju, pre svega u programima „Mutimate“, „Wordperfect“, „Displaywrite“, koji su već stekli čvrste pozicije i nude slične mogućnosti. Cena programa u Engleskoj je oko 430 funti.

Z. Ž.



Kako se oženiti... i živeti srećno stotinugodina

Kompijuterske novine prepune su raznih uporednih prikaza računara, i vi treba da se opredelite na osnovu hardvera, performansi, softvera, cene... Sve to bi, svakako, trebalo da ima neki značaj, ali nikako na tako direktan način. Ne smete dozvoliti da računar na osnovu skrivene reklame ili stvorene psihoze izabere vas a ne vi njega. Pri izbora računara morate mnogo više da vodite računa o osobinama vaše ličnosti nego o raznim tehničkim i praktikalijama.

Najvažniji kriterijum pri izboru računara je da vas učini SREĆNIM. Nije reč o računaru na poslu. Računar na poslu je kao kolega sa posla; možete da ga mrzite, ali morate sa njim da provodite radno vreme. A gde je ljubav?

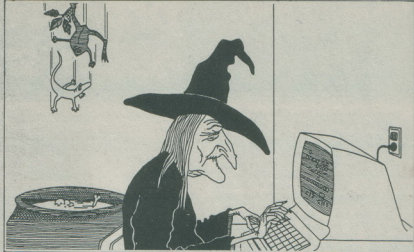
Vi stupate u određeni intimni odnos sa računarom, koji je u većoj ili manjoj meri ljubavni. Iako je računar mlađeg roda, nema apsolutno nikakvih razloga za grizu savesti da radite nešto nenormalno.

Moguća bi bila tipologizacija vlasnika na osnovu toga kakav računar neko ima kod kuće, ali to se ne bi poklopalo sa time šta bi trebalo da ima. Važno je umeti prepoznati svoju sreću, kako kaže Čika Jova Zmaj i životna saputnica se bira tako. Ako ne znate kakva vam žena treba, moći ćete da izaberete na osnovu svog računara, i obrnuto. Ili ćete, bar, znati kakva vam žena (računar) ne treba.

Čak i da ste izvršili idealni izbor na oba plana, ne znači da će vam utroje biti lepo. Računar se, očito, meša u ljubavni život. Do sada se kao razlog za razvod braka nije pojavljivalo kompijutersko zanemarivanje (okrutnost). Problematiku kako harmonično živeti sa ženom i računarom ovog puta ne razmatramo. Ograničimo se na izdavačevu izolovanog uticaja dva tela (vas i računara), jer, kao i u nebeskoj mehanici, problem tri tela u opštem slučaju nije rešen.

Izbor se obično vrši na dva načina.

Pri je poznat pod nazivom „sliku svojega



ljubim", a pogodan je i donekle u redu za one koji su dobri, lepi i pametni — ako ste ružni, ružno ćete i odabrati. Za razliku od partenogeneze, bolji način predstavlja izbor komplementa. Tako ste na dobitku. Vi treba da uskladišete sebe i računar u harmoničnu zajednicu, da se uzajamno prožimate i dopunjavate.

Sve se može birati po ovim principima: ne samo životna saputnica nego i kućni ljubimac — od papagaja do računara.

Svaki izbor je kompromis između realnih stvari i onog što vam duša išta. Nevolja je u tome što se pravi previše kompromisa i svi kupuju iste stvari.

Ima računara koji su na glasu kao dobra prilika, mada koriste mnogo šminke. Navodno mogu sve. Tu je nešto sumnjivo. Za sada vas neće biti sramota da se pokazete sa njim u društvu, ali nasamo ćete samo vi znati kakav je. Kada vam bude u rukama, bićete u stanju da ocenite koliko stvarno ređi. Može vam se dogoditi i da vas zavedu, jer ste mnogo više očekivali. Ne nasjedajte na jeftine intelektualne tipove, sa brzih večernjih kurseva, koji se dobro prilagođavaju vama. To je najbolji primer kako vas kupuju. Ako je „amstrad“ računar vašeg života, čuvajte se devojaka uz koje, kao paket-aranžman, obično ide i cela njena familija.

Može vam se desiti i da nasednete na drugu krajinu i nabavite računar koji svi imaju, koji je zato izgubio svaku izuzetnost. Mašinu kojoj je fascinirantnost narušena moraće pod hitno da menjate, ili zato što su je mnogi imali, ili zato što su i drugi videli da ona nije bog zna šta. Recimo da ste snob. U tom slučaju, ne smete imati računar koji se preterano meša sa običnim svetlom. Svaka šuša ima „komodor“, a tek „spektrum“!

Spektrum nikako nije dobar izbor, a oni koji se odlučuju za kupovinu u današnjim uslovima spadaju, najverovatnije, u kategoriju računadžija koji su nešto loše izračunali. Ako vam je bar malo stalo do socijalnog prestiža, biće vas sramota da sirići „spektrum“ iznesete pred goste. Potrudite se da pri izboru životne saputnice ne počinite istu grešku, jer, dok je sa „spektrumom“ moguća ljubavna avantura, gde ćete se malo poligrati njime i ostaviti ga čim vam se ukaže bolja prilika, sa životom ženom je to daleko teže. Tada imate dve mogućnosti:

da je se nekako otarasite, ili da se poslužite tehnikom maskiranja, kao što rade mnogi „spektrumovci“. Dok oni kamufliraju „spektrum“ u razne veće kutije sa profesionalnom tastaturom, vi možete svojoj ženi da kupite bundu.

Masovni izbor upošte nije izbor, zato odbijam da o tome govorim.

Totálna je greška kupovati i računar nulte socijalne kompenzacije. Smatra se da je to sirotinjski izbor. Mnogo greše školske vlasti kad nabavljaju deci iz pasivnih krajeva inferiorne mašine, ukoliko ne žele da ih obeležu za ceo život. Da ne pominjemo roditelje koji su ekstremne stipe, pa kupuju ovaj računar deci „jer je šteta da pokvare nekom skuplji“. Od takve dece postaju ljudi sa pritaženim osećajem inferiornosti. Traumačična iskustva sa ovom vrstom računara vuku se kroz ceo život. Osim toga, mogu vas deklarirati kao mazohistu koji doživljava sldoštarske svaki put kada pokuša da ukuca nekči duži program preko nacrtane tastature.

Za vas nisu ni promašeni modeli. To je izbor za male pare. Vi ionako ne mislite da učite programiranje. Nećete valjda da vas upoređuju sa takvima koji se razmeću da kod kuće imaju računara komada 5 (pet), a ni jedan ne umeju da upotrebe.

U ovu grupu ne spadaju promiskuitetni tipovi koji vole da probaju što više računara. Ležerno će da se hvale da su celu naredili na „amigi“. Takvi što daju prednost kvantitetu, u ljubavnom životu teže da ostvare što više rečki.

Ako prednost dajete kvalitetu i žudite za tim da budete fascinirani, izbegavajte računare koji mnogo obećavaju. Vaša radost biće kratkog veka, jer takvi računari u većini slučajeva ne ispunje obećanja. Moraćete da ih menjate pre nego što drugi primete da ste zeznuli.

IBM PC po definiciji ne dolazi u obzir, mada već sam po sebi predstavlja dokaz da u računarskom smislu niste makar ko. To je mašina koja vas neće osramotiti, ali uskoro biste ustanovili da nije baš mnogo zabavna. Uostalom, za ono za šta biste je vi koristili mođe da posluži i neka mnogo jeftinija igračka.

Očigledno, za vas nisu ni masovni, ni jeftini, ni ekskluzivni modeli. Vreme je da vam otvoreno, konkretno i bez uvijanja predložimo pravi računar za vas.

Jelena Rupnik

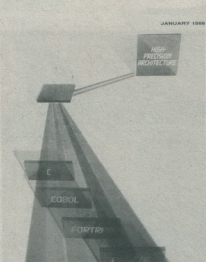
(NASTAVIĆE SE)

Od trenutka kad je prvi centralni procesor ugledao svet (odnosno laboratoriju), svi su se proizvođači trudili da natpraju što više naredbi u svoj čip bez mnogo razmišljanja o tome da li je korisnik sve to potrebno ili ne. Tako su vremenom procesori (a potom i mikroprocesori) počeli da množe, dele, pozivaju procedure, stvaraju lokalne varijable i rade slične komplikovane poslove koji su, u stvari, nasleđeni iz filozofije današnjih viših programskih jezika pa, shodno tome, ne vode mnogo računa o efikasnosti. Zauzvrat, daju mogućnost da i dosta loš programer piše prevodiocima za više jezike (pa se posle korisnik čudi kako to da njegov C daje tako dugačak kod ili tako katastrofalne numeričke rezultate). Dugo vremena je tehnološki razvoj čipova bio u velikoj ekspanziji, pa drastično povećanje gustine pakovanja ili brzine izvršavanja nije predstavljalo neki problem. Čak su se firme na sav glas hvalile čipom sa, recimo 100 000 tranzistora, smatrajući da je to samo po sebi dokaz vrhunskog kvaliteta. Takva je filozofija našla naročito ohrabrenje u građenju lisp-procesora koji „direktno“ izvršavaju izuzetno visok programski jezik. Tržište je raslo, troškovi padali i sve je bilo različito...

a onda je puklo

pred očima nekih programera (u prvom redu akademskih) koji su shvatili da procesor koji ima naredbe-monstrume, čije izvršenje zna da potraje preko 150 vremenskih ciklusa, u vreme kad granice gustine pakovanja i povećanja brzine postaju vrlo uočljive i čvrste, ne može još dugo da zadovoljava sve veće potrebe u radu. Posle takvog „otreznejenja“, počeo je vrlo intenzivan rad na minimiziranju broja i složenosti naredbi ili, preciznije rečeno, smanjivanju broja kodova koje procesor raspoznaje. Na prvi pogled bi se reklo da nema svrhe praviti razliku između broja kodova i naredbi. Međutim, ako znamo da MC 68000 ima tek nešto više od 55 naredbi a raspoznaje preko 1000 kodova (zbog vrlo velikog broja adresnih modova) postaje sasvim očita razlika ova dva pojma.

Prvi završen procesor se pojavio pod krovom Hewlett-Packarda, što i nije naročito iznenađujuće s obzirom na renome ove firme. Nije se odmah našao u prodaji zbog potreba razvoja kompletnog računarskog sistema (HP nikad nikome nije prodao samo komponentu već uvek završen sistem), pa je promocija bila tek u februaru ove godine. Samo dan kasnije IBM je dosta bučno najavio svoj model, čime je filozofija računara sa smanjenim skupom instrukcija (RISC-Reduced Instruction Set Computer) postala definitivno „legalna“. Što je izbilgo mnoge argumente iz ruku protivnika ovog koncepta, koji je iz akademskih prešao u



proizvodne sfere. Budući da je procesor Hewlett-Packarda prvi, dosad najbolje predstavljen, a kako stvari stoje i najbolji (toliko da ga ne nazivaju RISC već HP Precision Architecture) upoznaćemo RISC-filosofiju na njegovom primeru. No, pre toga se moramo podsetiti nekih retko pominjanih „sitnica“ vezanih za

stare dobre

klasične procesore, koji ipak neće otići tako brzo u staro govođe. Klasični procesori imaju dve vrste internih instrukcija: osnovne, koje su ugrađene u hardver (obavija ih ALU-Arithmetic and Logic Unit), i kompleksne koje su potprogrami sastavljeni od osnovnih. Takav procesor, dakle, ima unutrašnju strukturu kao bilo koji računar, pa zahteva dosta dodatnog vremena za procesiranje svake naredbe i raspoznavanje adresnih modova, pa potom za čitanje internog ROM-a. Mašinske naredbe dostupne korisniku su, naime, opet potprogrami internih instrukcija (koje se tradicionalno nazivaju mikrokod). Neki procesori (i vrlo retki mikroprocesori) omogućavaju korisniku pisanje i mikrokodiranih programa, no to zahteva dodatni razvojni sistem i mnogo ekstra-brze (i ekstra-skupe) memorije, dok gubitak vremena za dekodiranje i dalje ostaje. Postoje procesori koji se grade u tehnici bit-odrazaka kod kojih korisnik od početka stvara sve mašinske naredbe (pa, recimo, svaki kompajler raspolže „svojim“ mašinskim naredbama), no ta je tehnika izuzetno skupa i procesori tog stila neće nikada dobiti prefiks mikro- toliko potreban za eventualnu dominaciju. Osim toga, organizacija procesora sa mikro i mašinskim kodom nosi u sebi nepotrebno opterećenje, jer se, praktično, dva puta radi isti posao na dva različita jezika, s tim da je mikrokod nedostupan recimo C-prevodiocu, a mašinski jezik nedostupan sistemu za razvoj mikrokoda — dakle nešto kao glupi telefoni

ili međurepubličko dogovaranje. Stoga RISC filozofija zahteva da instrukcije procesora budu

sve u hardveru

dakle bez ikakvih mikrokodova, internih ROM-ova i sličnih raspisnika vremena. Instrukcije su, naravno, jednostavne (one koje se najčešće i koriste), pa im ne treba nikakva interna poštalicica da bi se izvršavale. Drugi važan imperativ je da svaka instrukcija „potroši“ samo jedan ciklus sata. Od ovog su izuzetne instrukcije za skok (Branch) uzimanje (Load) i skladištenje (Store) podataka. Međutim, svaka naredba koristi stvarno tri ciklusa (fetch, exectue, store), ali se oni preklapaju (vidi sl. 1) tj. koriste se tekuća-linija (pipelne) koja omogućava istovremeno izvršavanje tri faze (ciklusa) tako da je ukupni efekat kao da je svaka instrukcija „potrošila“ samo jedan ciklus. Ova tehnika je odavno poznata u svetu procesora (pa i nekih mikroprocesora) i zahteva takozvanu „cash“ (keš) memoriju ultra brzu unutrašnju memoriju koja čuva instrukcije i podatke koji će kasnije biti obrađivani i tako smanjuje potrebu za komunikacijama sa sporim spoljašnjim čipovima. Kompletnu strukturu High-Precision Architecture procesora vidite na slici 2. Bolji poznavaoi arhitekture procesora će odmah primetiti nedostatak mikroprogramskog sledina i mikroprogramske memorije, kao i činjenicu da koprocator pokretnog zarezra ima direktan pristup u „cash“ koji se sastoji od odvojenih delova za podatke i naredbe, što nije nepoznato ali ni često viđano.

Zahvaljujući svim tim hardverskim „sitnicama“, procesor je izuzetno brz pa se kompleksne instrukcije, koje se kod klasičnog procesora nalaze u mikrokodu, mogu pisati u

milikodu

ovde se ne radi samo o igri reči (odnosno prefiksa), već o suštinskoj razlici. Milikod je, naime, softverska tvorevina, koju procesor prepoznaje kao običan potprogram, potpuno ravnopravan svim ostalim potprogramima i sačinjen od istih mašinskih naredbi kao i svaki drugi. Takav milikod se može pisati u istom visom programskom jeziku kao i ostali „obični“ jezici, a jedino ograničenje je da sme da menja samo one procesorske registre koje uzima kao ulazne parametre i neke opšte namene, mada je i ono na nivou korisnika, to jest procesor ne vrši nikakvu proveru u toj sferi. Dalja prednost milikoda je što se dodatno vreme troši samo kada se neki takav potprogram pozove (dok se kod mikrokodiranog procesora dodatno vreme stalno troši na odloženo dekodiranje). Budući da je softverska tvorevina, milikod je prenosiv na mnoge mašine iste klase (što će naročito u budućnosti biti značajno) a različiti procesora, što kod mikrokodiranih mašina nije

Još od pradavnih vremena (u računarstvu vremena brzo postaju pradavna) u kojima je Alan Turing sanjao o računaru beskonačne brzine i memorije, pa sve do danas, svako ko se lola više bavio računarima nosio je u sebi neki svoj san o idealnoj mašini. Pojavom svake nove generacije računara, nekima su snovi bili ispunjeni dok su drugi nastavljali da sanjaju do nekih novih generacija. U posljednjih godinu dana pojavili su se procesori koji zalista izgledaju kao iz nečijeg sveta mašte.

bilo ostvarljivo ni u najluđim snovima. Neki snovi, dakle, ipak postaju stvarnost.

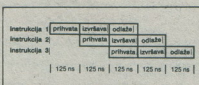
Teuća linija, koja je jedan od stozera postojanja RISC-filosofije upotrebe, postaje potpuno neefikasna kad program sadrži mnogo skokova. Jer se gubi jedan ciklus na dobavljanje instrukcije sa neke dotada nepoznate adrese, tako da, recimo, obična petlja sa par hiljada ponavljanja (recimo, pri pretraživanju baze podataka ili teksta) može potpuno da razori teuću liniju. Ovaj procesor koristi jedan duhovit trik o kome će kasnije biti reči kad budemo razmatrali softverske karakteristike, a sada pogledajmo

programski model

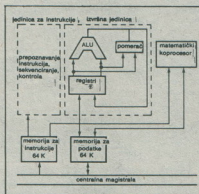
kako to ovevani softveršači vole da kažu. Na raspolaganju su 32 registra opšte namene, 32 kontrolna registra (od kojih jedan statusni) i 8 prostornih registra. Registri opšte namene nemaju za sada nikakvu specifičnu ili preporučenu namenu. Dužina irg je 32 bita i prevodilački sistem ih koristi za prenos parametara između procedura. Kontrolni registri se koriste za procesiranje prekida, zaštitu virtualne memorije i druge sistemske funkcije. Stanje procesore se odražava u 32-bitnom statusu registra. Prostorni registri specifičiraju 4 gigabajta dužakać virtualne prostore i sadrže 16 ili 32 bita, što daje ukupno 48 ili 64 bita virtualnog adresnog prostora. Dva adresna registra takođe pokrivaju maksimalno 64 bita virtualnog prostora, što je u sadašnjim uslovima praktično neiskoristivo, ali garantuje da će ovaj procesor još dugo zadovoljavati sve potrebe korisnika. Virtualne adrese se prevode u fizičke posredstvom specijalnih odvajaa takozvanih TLB (Translation Lookaside Buffer), koji su takođe hardverski rešeni. HP Precision Architecture podržava rad sa više tipova podataka: celobrojni sa dužine 16 bita (halfword), koji moraju biti postavljeni na parne adrese, 32 bita (word) koji moraju biti postavljeni na adrese deljive sa 4. Karakteri su dužine 8 bita i zadovoljavaju ASCII standard za vrednosti %127 i HP-ov prošireni Roman 8 set za vrednosti 128-255.

Decimadni podaci mogu biti pakovani i nepakovani, ali su uvek postavljeni na granicu reći (početna adresa deljiva sa 4) i mogu se sastojati od 7, 15, 23 ili 31 BCD cifra.

Brojevi u pokretnom zarezu se predstavljaju u skladu sa ANSI/IEEE standardom. Podržana je jednostruka, dvostruka i četvostruka preciznost (vidi sl. 3), odnosno obzrebedna 23, 52, 112-bitna (7, 16, 34 dekadnih cifara) tačnost. Operacije u pokretnom zarezu se mogu softverski simulirati ili izvoditi na posebnom FP procesoru koji (budući da direktno postupa u „cache“) ne oduzima praktično ni vreme ni



Instrukcije se pribavljaju i izvršavaju u tri ciklusa, ali se oni preklapaju u tzv. „teućoj liniji“, pa je efektivno vreme trostruko kraće



Procesorski modul

dotadnu memoriju. FP procesor ima dvanaest 64-bitnih registra za podatke, 32-bitni status registra kao i sedam 32-bitnih registra za procesiranje greška-iuzetaka (tzv. IEEE exceptions). Raspoznaje 11 naredbi (FADD, FSUB, FMPY, FDIV, FLEM, FSQRT, FRND, FCMP, FABS, COPY, CONVERSION) i tri preciznosti.

Pravom programeru (a čitalac, „Računara“ je, kao što znamo, pravi programer) za upoznavanje procesora nedostaje još samo

set instrukcija

koji se sastoji od 140 različitih kodova (vidi sl. 4) fiksnog formata i dužine jedne reći. Do pre dve godine ovakav procesor bi mogao biti nazvan „velikim raspisnikom“, ali se danas, kada čak i kućni računari rade na 16/32 procesorima sa 1 Mb jeftinog RAM-a, može reći samo „to je ono pravo“ i posle „ledeno hladne COCA-COLE“ pogledati šta se sve nalazi u tim 140 kodova.

Za rad sa memorijom koriste se samo Load i Save naredbe, koje uzimaju 24 koda i, povremeno, više od jednog ciklusa, mada dobar prevodilac i to vreme dostiže da uradi pametnom raspodelom posla.

Ulovni skokovi pokrivaju 18 kodova. Taj se skok često može eliminisati budući da mnoge instrukcije omogućavaju uslovno izvršavanje (u zavisnosti od nekog bita stanja) u samo jednom ciklusu. Da se pri skoku ne bi gubio jedan ciklus i time rušila funkcija tekuće linije, procesor za vreme preračunavanja adrese izvršava prvu sledeću naredbu, pa je samo stvar prevodioca da

prvu naredbu TEHN-konstrukcije ili povratne petlje postavi neposredno iza naredbe skoka. Jednostavno, ali se trebalo setti.

Aritmetičko-logičke naredbe su svedene na najnužnije, ali ipak zauzimaju čak 53 koda. Tu su, naravno, ADD i SUB naredbe, kao i naredbe za upoređivanje, te OR, AND, XOR, NOT, poznate sa ranijih procesora. Novost je naredba za pomeranje i sabiranje istovremeno (i sve za jedan ciklus) koja drastično ubrzava množenje, naročito malim celobrojnim konstantama, te EXTRACT/DEPOSIT naredbe za efikasnu podršku bit-manipulacijama, koje koriste viši programski jezici.

Instrukcije za kontrolu sistema koriste 31 kod i vrše manipulaciju kontrolnim registrima, „cache“-om, TLB-ima i prekidima. Takođe, vrše nadzor pristupa virtualnoj memoriji i rade ostale „kućne poslove“.

Jedna od bitnih razlika između ovog i drugih RICS procesora je grupa instrukcija za podršku koprocesorima (9 kodova), mogućnost direktnog prelaska na rad u pokretnom zarezu, kao i četiri „specijalne operacije“ koje rade ono što korisnik (mada će to češće biti Hewlett-Packard) zaželi, dakle nešto kao čitač naših snova.

Kao što se vidi nije zaboravljena nijedna važna instrukcija ali nije bilo ni razbacivanja. Dolazimo do logičkog pitanja: ko je taj sveznajući monstrum koji može sa sigurnošću da tvrdi da su baš ti 140 instrukcija najpotrebnije i da se najčešće koriste? Pa, naravno, to je

spectrum,

ovde se, svakako, ne radi o Sinklerovoj „maloj-ružnoj-tupavoj“ igrački, već o ogromnom razvojnom projektu u koji je Hewlett-Packard uložio 100 miliona dolara, što je i za američke prilike velika suma. U okviru tog projekta je izvršena analiza mnogo megabajta mašinskog koda iz raznih aplikativnih programa da bi se dobile frekvencije korišćenja pojedine vrste instrukcija. Potom su, uz konsultaciju hardveršača, među njima izabrane one instrukcije koje se mogu izvršavati u jednom ciklusu. Tome su još dodane neophodne instrukcije koje i referenciranja memorije (gde je opet presudnu ulogu odigrala učestalost korišćenja). Glavna snaga ovog projekta je što su softverski deo (tj. kompajlerski sistem) i hardverska baza razvijani istovremeno. Koristeći simulaciju još nepostojeg procesora, programeri su mogli još u ranim fazama razvoja da koriguju neke od planiranih instrukcija. Osim toga, procesor je po završetku razvoja u startu raspolagao sa 4 programska jezika, izvrsnim optimizatorom sa tri nivoa i moćnim MPE XL operativnim sistemom. Čitava tehnologija je najvnljave kao arhitektura za 90-te i kasnije godine, što znači da firma neće tek tako ispustiti sve adute, tj. da program Spectrum ostaje i dalje aktivan. U stvari, ono što se do sada videlo je samo simulacija procesora serije

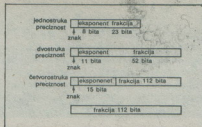
3000, čime je dokazano (što je vrlo bitno za buduće poslovne odnose, a što su neki osporavali) da HP Precision Architecture može uspješno da podržava poslovne aplikacije i kobil. Čak i u svom „bledom“ izdanju, sa procesorom u TTL diskretnoj tehnologiji, performanse su izvrsne.

Prosečna brzina od 4,5 maksimalna od 8 MIPS-a rečito govore da će izgleda i moćni IBM ovog puta položiti oružje, pogotovu kad na scenu izađe VLSI verzija procesora, kad glavne aplikacije budu na C-u i fortranu i kad u prodaji bude novi prevodilački sistem koji paskal, fortran, kobil i C prvo prevodi na spektrum međukod SLLIC (Spectrum Low-Level Intermediate Code), tako da se optimizacija i konačno oblikovanje izvršnog koda (koji se može optimizirati na još dva nivoa) izvodi jedinstvenim procesom nezavisnim od jezika koji se prevodi. Taj je prevodilački sistem, u stvari, baza na kojoj počiva prava efikasnost RISC-filozofije, budući da softver brine mnoge „hardverske“ brige, pa će čitaoci „Računara“ imati priliku da se posebno sa njim upoznaju, a sad je red da spomenemo

ostale

predstavnik „new wave“-a. Prvi je (po datumu pojavljivanja) Transputer T414 koji je na granici RISC-a i procesora uopšte, budući da predstavlja mikrokomputer-na-čipu. Sadrži (pored CPU-a) i 2K RAM-a, dva DMA kontrolera i četiri serijska I/O kanala sa brzinom prenosa od 10 Mboud-a. Sam CPU ima samo 6 32-bitnih registra, od kojih su samo 3 slobodna za rad. Međutim, pošto je RAM u čipu brz kao i sam procesor, na raspolaganju je u stvari 500 32-bitnih registra. Brzina jednog čipa je 5—10 MIPS-a. Posebna pogodnost Transputera je iako i efikasno povezivanje u mrežu, dakle i ova tehnologija cilja, pre svega, u budućnost, koja je dosta ruzičasta, iako je Immos mala (čitaj finansijski slaba) firma, budući da već postoje razvojni paketi za VAX i IBM PC, te da je u prodaji i Immosov razvojni sistem kao i specijalni jezik Occam koji maksimalno podržava multiprogramski i multiprocesorski rad i u kompajleri. U prodaji su, takođe, i tri razvojne pločice, od kojih je jedna IBM PC-kartica što je, svakako, veliki mamac za buduće korisnike, a samim tim i velika početna prednost ovog procesora u odnosu na sve ostale.

Ako se neko od sada još i pitao zašto je Acorn pao u velike finansijske teškoće, njihov ARM (Acorn RISC Machine) sasvim pouzdano daje odgovor. Čip je „razvijen“ za samo 18 meseci (Hewlett-Packard je svoj razvijao 5 godina, a IBM nešto duže ali sa prekidom), a urađen je u zastareloj tromikrometarskoj tehnologiji. Pored toga što ima samo 25000 tranzistora i svega 44 instrukcije, postiže brzinu koja je svega 0,5 MIPS-a veća od brzine MC 68020 (koji predstavlja kompletnu mašinu kompleksnih mikrodiniranih instrukcija), pa se čak i slabije upućenom posmatraču nameće pitanje zašto se, uopšte, ušlo u gradnju čipa koji je već u startu zastareo, a uz to bez

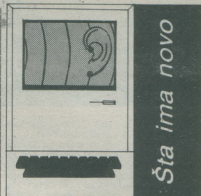


Formati računanja u pokretnom zarezu

ikakve softverske podrške, sa jednom šansom da postane spoljni procesor BBC-a (naravno, ako se nađe nekog dovoljno lud da za to da 2000 funti).

IBM je, očigledno, bio zatečen brzim izlazkom „sklepanog“ ali uprkos tome brzog HP računara, tako da je brže-bolje morao da i sam izbací svoj računar IBM-RT koji vrlo solidno izgleda (a toliko i košta). O njemu je poznato vrlo malo podataka, ali je najbitnije da raspolaže procesorom sa 118 instrukcija, pri čemu su 84 jednociklusne, koji pri svega 5,88 MHz dostiže brzinu MC 68020 na 12,5 MHz, što rečito govori o velikoj brzini (narodičito kad izađu modeli sa većim klokom). Računar ima IBM-ovu verziju UNIX-a, što znači i obilje aplikativnog softvera, kao i upravljač memorije koji adresira da 1 Tb virtualnog prostora. Za sada je namenjen pre svega, za CAD. Osnovni model (CPU, 1 MB RAM-a 40 Mb tvrdi disk i monohromatski monitor) košta oko 14, a najjači (2 Mb RAM-a, 70-Mb tvrdi disk i kolor monitor) oko 40 hiljada dolara.

Poslednji procesor izgleda, na prvi pogled, kao uljez među RISC mašinama. Radi se o procesoru koji izvršava visok programski jezik Fort, ali to čini sa svega četrdesetak naredbi, postižući maksimalnu brzinu od 10 MIPS-a, a minimalnu od 6 (za diskretnu verziju procesora) i sadrži 2 K internog RAM-a. U čemu je stvar? Fort je relativno mlad jezik koji je tek pre neku godinu pokazao da može da bude raspravčan sa listom u razvoju veštačke inteligencije, a moćniji u svim drugim primenama, naročito u real-time aplikacijama. Mada je potpuno strukturiran i visoko modularan, on je bzo skoro koliko i mašinski kod (sada on i jeste mašinski kod) i, što je najbitnije, vrlo lako „vaskrsava iz svog pepela“, tj. dovoljno je četrdesetak instrukcija da se postavi ceo jezik (koji sam sebe proširuje), pa je proto idealan za primenu RISC filozofije. Ovaj procesor uopšte ne mora da brine za razvojni softver i operativni sistem, budući da je fort sam sebi operativni sistem, a softvera ima u izobilju i po vrlo povoljnim cenama. Kad izađe VLSI verzija sa izmenjenim (čitaj višestruko ubrzanim) sistemom dokodiranja, mogu se očekivati i brzine bliske 50 MIPS-a, što uz visok programski jezik može da znači i novi veliki proboj u području veštačke inteligencije. Do pojave čipa u prodaji, softver se može pisati na bilo kom kućnom računaru (pa čak i na Sinkerlovoj igrački), što je još jedan bitan faktor za privlačenje velikog broja mušterija i naravno pad cene proizvoda. Procesor proizvodi kompanija Metaforth, čiji su osnivači Alan Winfield (Winfield) i Rod Gudman (Goodman), nadaleko poznati fort-eksperti (ili više eminenција, ako se to nekomе više svida).



Šta ima novo

Besplatni modemi

Ukoliko se učlanite u britansku kompjutersku mrežu Micronet i uplatite jednogodišnju članarinu, dobićete potpuno besplatni modem *Prism 2000* i prateći komunikacioni softver za vaš računar! Godišnja članarina, sa sve hardverom i softverom sada iznosi samo 50 funti.

Micronet planira da privuče nove članove još jednom atrakcijom: pripremljena je višekorisišnička igra *Shades* u kojoj ćete se sukobiti sa 1.000 kompjuterski kontrolisanih igrača i nepoznatim brojem sagraih koji su u tom trenutku priključeni na mrežu. Prvom igraču koji ostvari sve ciljeve igre mođe da pripadne nagrada od 5.000 funti! (D.R.)

Dizajirate li čipove?

Ukoliko se bavite dizajnom integriranih kola, pomoći ćemo vam da potrošite 7 500 funti. Firma *Qudos* za ovu situ su svoju prodaje trideset dvobitni račun *Cambridge Workstation* sa 20 M hard diskom, specijalnim monitorom i kompletnim softverom za projektovanje čipova. Kada dizajirate integrisano kolo, pošaljite disketu, *Qudos-u* i za nedelju—dve očekujte prve primerke!

Da ne zaboravimo — kupovinu možete da obavite javljajući se na telefon (0223) 862-333. (D.R.)

PCST mreža

Ne radi se o šaputanju. Ako za svoju malu poslovnu primenu nikako niste mogli da se odlučite između spore legende sa gomilom programa i jeftinog moćnog klinca sa malo programa, postoji nada za vas. Compulink iz Kanade je proizveo Imaginet mrežu pomoću koje se mogu spojati PC i ST. Osnovni elementi mreže koštaju samo 900 dolara, a šta je to (pogotovu ako firma plaća) ako ste već morali da platite ST i PCja. (B.D.)

Muvajte grafiku

Ako smatrate da je zaista kriminalno kako vaš drag računar sporo operiše sa grafičkim elementima na ekranu, možete da se razvedrite. Intel je razvio grafički koprocesorski čip 82786 koji može da manipuliše pozorcima i sličnim stvarčicama i do sto puta brže nego što se to postiže softverski. Čip sadrži dva procesora i cena mu je za sada nije poznata, iako nije verovatno da će biti mala. (B.D.)



Dejanove pitalice

1	*** M2 = M1 * M2	*** M2 = M2 * M1
2	*** M3 = M1 * M3	*** M3 = M3 * M1
3	*** M1 = M1 * M3	*** M3 = M3 * M1
4	*** M1 = M1 * M3	*** M3 = M3 * M1
5	*** M1 = M1 * M3	*** M3 = M3 * M1
6	*** M1 = M1 * M3	*** M3 = M3 * M1
7	*** M3 = M3 * M1	*** M3 = M3 * M1

slika 2

1	*** M2 = M2 * M3	*** M2 = M2 * M3
2	*** M3 = M2 * M3	*** M3 = M2 * M3
3	*** M3 = M3 * M3	*** M3 = M3 * M3
4	*** M2 = M2 * M3	*** M3 = M3 * M3
5	*** M2 = M2 * M3	*** M3 = M3 * M3
6	*** M3 = M3 * M3	*** M3 = M3 * M3
7	*** M3 = M3 * M3	*** M3 = M3 * M3

slika 3

1	*** M3 = M2 * M1	*** M3 = M1 * M1
2	*** M3 = M3 * M1	*** M3 = M3 * M1
3	*** M3 = M3 * M1	*** M3 = M3 * M1
4	*** M3 = M3 * M1	*** M3 = M3 * M1
5	*** M3 = M3 * M1	*** M3 = M3 * M1
6	*** M3 = M3 * M1	*** M3 = M3 * M1
7	*** M3 = M3 * M1	*** M3 = M3 * M1

slika 4

Dejan Ristanović

Brzopotezni jezik

Jedanaesta Dejanova pitalica predstavlja dobru potvrdu istine da novi programski jezik nije lako naučiti čak i kada se on sastoji od svega nekoliko naredbi — u predviđenom smo roku primili solidnih 235 odgovora, ali je u konačnu izbor ušao svega 155 pisama!

Najpre ćemo se, po običaju, posvetiti na problem — u „Računarima 18“ smo objavili interpretir programskog jezika koji smo nazvali *Microcalc '86*. Zamisljeni računar ima memoriju od čitave 3 ćelije, čiji je kapacitet, za divno čudo, beskonačan: u svaku se ćeliju može smestiti proizvoljno veliki **ceo** broj. Naredbe jezika imaju opšti oblik $M_i = M_j \cdot M_k$, de tačka označava bilo koju od operacija sabiranja, oduzimanja, množenja i celobrojno deljenje, dok i, j i k označavaju memorijske ćelije ($1 < i, j, k \leq 3$). Uz zadatka smo predvideli i sliku na kojoj smo ispisali dopuštene operacije *Microcalc '86*, ali je ta slika na nepoznat način izbledela pa je, premda malo kasno, objavujemo u ovom „Računarima“. Vrlo je važno da zapazimo da sve operacije oblika $M_i = M_j \cdot M_k$ nisu dopuštene: kod oduzimanja i deljenja rezultat se obavezno smesta u ćeliju iz koje je uzet prvi argument. Naredbe $M_2 = M_1 / M_1$ ili $M_3 = M_1 / M_2$, dakle, nisu dopuštene, što je bilo jasno pomenuto kako u zadatku tako i u objavljenom interpretiru. Čak 39 čitalaca nije primetilo ovo zaoklićilo, pa smo primili otprilike ovoliko rešenja sa 6 naredbi; ova rešenja smo, na žalost, morali da odbacimo, jer ne ispunjavaju uslove zadatka!

$M_1 = M_j + M_k$	$1 < i = 1, 3, k < 3$
$M_1 = M_i - M_j$	$1 < i = 1, j < 3$
$M_1 = M_j * M_k$	$1 < i = 1, 3, k < 3$
$M_1 = M_i / M_j$	$1 < i = 1, j < 3$

Slika 1

Pošto smo naučili *Microcalc '86*; počemo da rešavamo zadatak. U m_1 je upisan broj točkova koje neka fabrika ima u magacinu (ovaj je broj veći od nule), dok su u M_2 i M_3 upisane nule. Treba da napišemo program koji će u M_2 upisati broj automobila koje fabrika može da proizvede, a u M_3 preostali broj točkova — podrazumeva se da automobil ima četiri točka, premda bi

bilo zanimljivo razmotriti i realnu situaciju u kojoj kola imaju i rezervnu gumu!

Najpre ćemo zameniti točkove brojevima — naš program treba da upiše $int(M1/4)$ u M_2 a zatim i $M1 - M3 * 4$ u ćeliju M_3 . Vidimo da nam je neophodna konstanta 4 koju ćemo formirati primenom tri naredbe, npr: $M3 = M1 + M1$; $M3 = M3 + M3$; $M3 = M3 / M1$ (pri tome smo koristili zadatu činjenicu da je broj točkova u magacinu veći od nule — inače ne bismo dobili broj 4 čak ni kada bi računar smatrao da je $0/0 = 1$). Za određivanje broja automobila nam treba jedna (nedopuštena) naredba $M3 = M3 / M1$, a zatim lako dobijamo i preostali broj točkova: $M3 = M2 * M3$; $M3 = M1 - M3$. Tako smo dobili program od 6 naredbi koji korektno rešava problem, ali koji je, na žalost, neprijemljiv na *Microcalc '86*: koriste se jedno nedozvoljeno deljenje i jedno nedozvoljeno oduzimanje!

Jednostavnim transformacijom rešenja koje smo upravo napisali dolazimo do programa sa slike 2 — dodali smo dve naredbe i omogućili da argumenti za deljenje i oduzimanje budu u M_2 i M_3 tako da se koriste isključivo dozvoljene naredbe. Na slicnoj se varijanti zaustavilo 35 rešavača.

U postavci smo pomenuli da se zadatka može rešiti i sa 7 naredbi: kako da eliminisamo jednu? Treba zameniti jedno oduzimanje sabiranjem ili jedno deljenje množenjem. Celobrojno deljenje ne možemo da svvedemo na množenje, ali bi se zato oduzimanje dalo svesti na sabiranje — setimo se da je $A - B = A + (-B)$ i da ćelije *Microcalc '86* mogu da primaju kako pozitivne tako i

Trinaesta pitalica

Urednik ove rubrike mora da prizna da nije lako pripremiti po jednu kolikolito interesantnu pitalicu mesečno. Trinaesta je pitalica, međutim, pripremljena čak i pre prve — od samoga sam početka znao koji će zadatka poneti ovaj (najsrećan broj!) Možda će vam se učiniti da je zadatka previše lak ali — dajmo šansu i mladim programerima.

Pokušaćemo da pronademo smisao verovanja da je **petak trinaestih** baksuzan dan. Izračunajmo, dakle, koliko ima poneđeljaka trinaestih, utorka trinaestih, sreda trinaestih i tako dalje — videćemo da je petak dan koji je najviše puta trinaestih! Petak 13 je, naravno, morao da bude proglašen za baksuzan dan da bi se potvrdili Martijevi zakoni!

Jeste li razmišljali o tome da se kalendar ponavlja svakih 400 godina? Ukoliko, dakle, ustanovite neku zakonitost kalendara koji važi od 1600. do 1999. godine, možete da smatrate da ona važi i za celokupan tok istorije! Napišite, dakle, program koji prebrojava ponedeljke, utorka, srede, četvrtke, petke, subote i nedelje koji su trinaestih u mesecu između 1600. i 1999. godine, a zatim upišite rezultate u kupon koji prilažemo ili u njegovu fotografiju.

Rešenja problema pošaljite na adresu: „Računari“ (za Dejanove pitalice), Rulevar vojvode Mišića 17, Beograd tako da pristigne pre 1. decembra 1986. Najboljim rešenjima i najsrećnijim rešavačima će pripasti novčane nagrade od 10.000, 5.000 i 3.000 dinara.

negative brojeve. Umesto konstante 4 ćemo, dakle, formirati -4 i tako izbeći potrebu da pripremamo M_3 za prijem razlike. Ovom malom izmenom dolazimo do konačnog programa sa slike 3 — najkratkim mogućim rešenjem do koga je došlo 155 čitalaca „Računara“.

Boris Vrabec iz Zagreba je problem proučio daleko detaljnije od opisa koji smo objavili i dokazao da se zadatka ne može rešiti sa manje od 7 naredbi. Boris je, štaviše, dokazao da postoji tačno 28 rešitkih rešenja sa po 7 naredbi od kojih je sedam suštinski različito. Rezultate razmatranja koja su Borisu Vrabecu donela prvu nagradu od 10.000 dinara vidimo na slici 4.

Drugu nagradu od 5.000 dinara dodeljujemo Ani Kuzmanović iz Niša koja je poslala po našoj oceni najpopularnije obrazloženo rešenje problema. Treća nagrada od 3.000 dinara je, posle improvizovanog izvlačenja, pripala Nenadu Sikimiću iz Splita.

U godinama 1600, 1601, 1602, ... , 1999 ima:	
_____	ponedeljaka trinaestih.
_____	utorka trinaestih.
_____	sreda trinaestih.
_____	četvrtaka trinaestih.
_____	petaka trinaestih.
_____	subota trinaestih.
_____	nedelja trinaestih.

Ime i prezime _____	
Adresa _____	
Mesto _____	

Računari i pravo Pirati pred licem pravde

izvinite, nismo znali

Presnimavanje i preprodaja programa predstavlja sastavni, najživiji deo naše računarske stvarnosti i, otuda, omiljenu temu u mnogim računarskim razgovorima — od pošalica do navlački obojenih rasprava „za i protiv“. Autorski redovi računara pojačani su od nedavno jednim diplomiranim pravnikom, koji se rado prihvatio redakcijskog zadatka da zaviri u Zakon o autorskom pravu i izvidi kako na sve to gledaju (krute) pravne norme, koje obično nemaju sluha za „više interese“. Šta očekuje Yu pirata ako izađe pred lice pravde?

Pre svega, ko su to „pirati“? Korisnici računara, bez sumnje, znaju da se oni bave nečim što nije dozvoljeno. Kako u oblasti računarenja nema ni brodova za pljačku, ni aviona za otmicu, ni dece za kidnapovanje, ni aviona bez šta je to što država zabranjuje a šta dopušta svojim građanima da rade u vezi računara? Mislim da ćemo se svi složiti u jednom: kada kažemo „pirat“, pod tim podrazumevamo lice koje se bavi presnimavanjem i prodajom kompjuterskih programa. Pirati, dakle, nisu nikakvi razbojnici i ukoljice — devedeset odstoj njih izgleda krajnje bezazleno — moj lični pirat ima svega 12 godina i ne izgleda nimalo krvoločno!

Domaća pravila ponašanja

Moramo, međutim, prihvatiti još jednu činjenicu: svaki tvorac neke ideje želi da tu ideju širi i saopšti drugim ljudima. To je u interesu celog društva, pa mu država u tome pomaže i štiti ga u njegovim nastojanjima. Isto tako, u društvu koje je interesu i da tvorca ideje ima i određene ekonomske koristi od svog stvaralaštva — i država ima ekonomske koristi u vidu poreza. Samim tim, kao i svom tako i u stvaraočevom interesu, svaka država propisuje određena pravila ponašanja o tome na koji način se ideje mogu širiti i ekonomski iskorišćavati. Ukoliko sam tvorac ideje ili neko treće lice vesno ili nesvesno dođe u sukob sa tim pravilima (ne ponaša se onako kako je propisano), nijedna država neće mirno sedeti i gledati šta se događa, već će, naravno, štiti svoj interes (u većini slučajeva) i postarati se da prekršioa pojuri, stigne i adekvatno kazni.

Da malo preciziramo neke osnovne pojmove: svaku tvorevinu iz oblasti stvaralaštva, bez obzira na vrstu, način i oblik izražavanja, zakon naziva *autorskim delom*. Tvorac takve tvorevine naziva se *autor*, a svak pravila ponašanja — *prava i obaveza autora* i prema autorima naziva — *autorsko pravo*. Sve to reguliše *Zakon o autorskom pravu*, i još neki drugi zakoni. Svim autorima država putem ovog zakona priznaje određena prava, garantujući odgovarajuću pravnu zaštitu njihovim autorskim delima. Ukoliko se, ipak, desi da neko dođe u sukob sa tim zakonima, verovali ili ne, ali on je u nadaru više dosta strogih propisa i višestruko je odgovoran za njihovo nepoštovanje.

Kako je i program za računar tvorevina iz oblasti stvaralaštva izražen na specifičan način, nema nikakve sumnje da i njega možemo svrstati u kategoriju autorskih dela, kao i da njegovom tvorcu pripada svoj-



stvo autora. Samim tim, Zakon o autorskom pravu pruža pravnu zaštitu i programima za računare — kako programima naših državljanina koji su objavljeni u Jugoslaviji ili u inostranstvu, ili koji još nisu objavljeni, tako i programima stranih državljanina. Što se tiče stranih programa, ukoliko su prvi put objavljeni kod nas (u stvarnosti malo verovatno ali zakon predviđa i tu mogućnost), oni imaju istu pravnu zaštitu kao i programi naših državljanina. Ali, ako su ti programi već objavljeni u inostranstvu, situacija se jako razlikuje već prema tome kakve je međunarodne obaveze prihvatila Jugoslavija na tom planu, i da li poseti takozvani „reciprocity“. Kako međunarodni pravni odnosi nisu predmet ovog članka, poći ćemo od pretpostavke da naša zemlja u konkretnom slučaju priznaje stranom državljaninu pravo na pravnu zaštitu njegovog programa na teritoriji Jugoslavije (štiti ga od bespravno ponašanja naših državljanina).

I imovina i moral

Svako autorsko pravo, pa time i pravo autora programa za računare, sadrži dve osnovne grupe prava: autorska imovinska prava i autorska moralna prava.

Autorska imovinska prava sačinjavaju pravo (ovlašćenje) autora na iskorišćavanje svojeg dela (ukoliko to svoje ovlašćenje nije posebnim ugovorom preneo na drugog, kada ono pripada samo tom drugom). Zakon nabrja, primera radi, da se autorsko delo (čitat — program) može iskorišćavati od strane pravno ovlašćenog lica (čitat — autora ili onoga na koga je autor preneo to ovlašćenje) na neki od sledećih načina: objavljivanjem dela, preradivanjem dela, reprodukcijom dela, umnožavanjem dela, obradivanjem dela, prikazivanjem dela, izvođenjem dela, njegovim prevodenjem, ili

na drugi sličan način. Potrebno je ponovo naglasiti: zakon navedena ovlašćenja priznaje i s k l j u č i v o autoru ili licu na koga su ta ovlašćenja prenetu ugovorom.

Autorska moralna (lična) prava sačinjavaju prava autora da bude priznat i označen kao tvorca dela (svako od nas se ponosi onim što je stvorilo), da se usprotivi svakom deformisanju i sakaćenju, kao i da se usprotivi i svakoj upotrebi dela kojom bi se vredali njegova čast i ugled. Isto tako, i ovim pravima autora odgovaraju obaveze svih drugih da priznaju ta prava i da se uzdrže od bilo kakve radnje koja bi dolazila u sukob sa navedenim pravima.

Pre nego što predemo na razjašnjavanje pravnih zavrtilaca koje se javljaju u vezi programa za kompjutere, da konstatujemo neke činjenice:

Kako ni autor ni njegova softverska kuća sa našim piratima ne sklapaju ugovore kojim ih ovlašćuju na umnožavanje programa, svi pirati se bave nezakonitim radovima!

Svaki pirat ili haker koji „razbije“ programsku zaštitu neke igre ili drugog programa radi toga da bi omogućio njegovu distribuciju preko piratskih kanala, *pre-raduje* u suštini program, pa se i on našao sa druge strane zakona!

Svako ko se bavi preprodajom programa bez valjanog pravnog osnova (bez odgovarajućeg ugovora sa softverskom kućom ili samim autorom), takođe je, bez sumnje, zakačio na drugu stranu.

Kako autor ima isključivo pravo da daje dozvolu za emitovanje svog dela putem radio-difuzne mreže, postavlja se pitanje — znaju li urednici „Ventilatora“ da emitovanjem inostranih programa bez dozvole krše zakon?

Sa jakim začinom

Da bi ova pravna papazjanija bila malo ukusnija, da navedemo odredbu zakona: „Lice čije je autorsko pravo povređeno može zahtevati zaštitu svog prava i naknadu štete koja mu je povredom naneseana“. A pored toga, sud preko koga autor bude ostvarivao zaštitu svojih prava može narediti:

1. Da se presuda javno objavi na trošak povredioca (obradovaćete vaš džep, jer oglašni nisu baš jeftini, a objavljanja presuda ima format pola stranice „Politike“!)

2. Da se povrediocu zabrani dalja povreda (posle take zabrane, ponovljena povreda će se uvek smatrati otežavajućom okolnošću, što će imati velikog uticaja na težinu kazne koju sud bude izrekao).

3. Da se predmeti kojima je naneseana povreda autorskog prava uništa (da, dobro ste pročitali). Znači, može da strada i vaš

računar sa kompletnom periferijskom opremom (kuku i lele). Ko ne veruje, neka malo prelista zakon o autorskom pravu.

Pirati, bolji li vas glava poste ovoga? Da još malo citiramo odredbe Zakona o autorskom pravu:

„Ko pod svojim imenom ili pod imenom drugog objavi, prikaže, izvede ili prenese tuđe autorsko delo ili dozvoli da se to učini, kazniće se za krivično delo zatvorom do tri godine.“

„Ko deformiše, skratki ili na drugi način menja tuđe autorsko delo, kazniće se za krivično delo novčanom kaznom ili kaznom zatvora do šest meseci.“

„Ko bez dozvole autora ili drugog nosioca autorskog prava, u slučajevima u kojima je takva dozvola potrebna po zakonu o autorskom pravu, objavi, preradi, reprodukuje, prikaže, izvede, prenese ili prevede, ili na drugi način iskoristi autorsko delo zaštićeno ovim zakonom, kazniće se za krivično delo novčanom kaznom.“

„Ko u nameri pribavljanja materijalne odnosno imovinske koristi stavi u promet i primerke autorskog dela za koje zna da su neovlašćeno reprodukovani ili umnoženi, ili ko takve primerke javno izloži ili ih prenese putem radio-difuzije ili na drugi način, kazniće se za krivično delo novčanom kaznom.“

Da veselje bude veće, da napomenemo da ukoliko neko od navedenih krivičnih dela učini organizacija udruženog rada, ona odgovara još i za prekršaj, a kazna je prava „sitnica“ — trideset starih miliona!

Kao prvo, zakon kaže da se za navedena krivična dela gonjenje preduzima po privatnoj tužbi. Šta to, u stvari, znači? U našem pravnom sistemu postoje dve vrste krivičnih dela: kod jednih, država preduzima krivično gonjenje odmah čim sazna za delo ili učinioca (u pitanju su teža krivična dela koja su po društvo opasnija). Za drugu vrstu država neće reagovati čak i ako sazna za dela ili učiniocima sve dok neke ne podigne tužbu ili ih podnese krivičnu prijavu protiv učinioca (obično onaj ko je tim delom oštećen). Radi se, pogadate, o krivičnim delima koja se gone po privatnoj tužbi.

Pošto krivična dela koja smo malopre citirali spadaju u ovu drugu grupu, možete mirno da spavate sve dok se neka strana ili domaća softverska kuća (i to je moguće!) ne naostri protiv vas zbog toga što neovlašćeno prodajete ili nešto drugo radite sa njenim programima i ne počnete da vas proganja. A kako neki naši pirati vole da pametuju i da se u programima koje rasturaju potpisuju punim imenom, prezimenom i telefonskim brojem, pretpostavljate da zainteresovanu softversku kuću ne bi predstavljalo problem da nađe dotičnog i da ga tuži sudu. Međutim, mala je verovatnoća da će do progaranja i doći, jer bi troškovi sudskog postupka za onoga ko ga pokrene bili u velikoj nesrazmeri sa onim što bi uspešnim okončanjem postupka zaista i dobio, pa se takva работа, u sadašnjim uslovima, jednostavno — ne isplati.

Alli, ipak, pretpostavimo da ima upornih na ovom belom svetu i da se neka kompanija ipak rešila da vas progoni do kraja. Uostalom, pojavili su se i prvi domaći izdavači softvera. Postoje dve mogućnosti — bićete tuženi ili za naknadu štete ili za krivično delo. Da krenemo redom.

Nastavak u sledećem broju

Nenad Mitrović

25/izvinitie, nismo znali

Hakerski vodič kroz Beograd

u potrazi za izgubljenim vremenom

Osnovna želja kojom smo se rukovodili kada smo planirali ovaj prilog bila je da pomognemo potencijalnom kupcu računara da ostane u Beogradu, prođe što jeftinije i kupi najkvalitetniju robu. Lepo rečeno, ali nije išlo! Ovaj tekst, koji je sada pred vama, samo je bledil priručnik onoga što je trebalo da bude: hakerski vodič kroz Beograd. Ali, podimmo redom — i sa ovu malo podataka moći ćete da steknete nekakvu sliku o stanju računarstva kod nas...



Prvi deo: krećemo u akciju

Krenuli smo od železničke stanice, kao mesta odakle bi potencijalni kupac iz unutrašnjosti otpočeo svoje beogradsko putovanje. U njenju okolini, računara ni za leki! Samo nekoliko komisiona i knjižara sa električnim pisacim mašinama. Ali, stop! U Karadorđevju ulici — to je ona kojom ide tramvaj za Kalemegdan — nailazimo na veliku reklamu za „komodor“! Ulazimo. Lepo uređena unutrašnjost, gomila opreme (uvoz), ploče, kasete, uređaji, kablovi, sintisajzeri, gitare... i evo, RAČUNARI! Naravno, „komodor“: Tu su svi — PC 128, C128, C64 (i novi i stari dizajn), kaseteši, diskovi, monitori, džojstici, jednom rečju... praznik za oči!

Iz razgovora sa ljubaznim prodavcem (ovde su svi ljubazni) Darkom Kosićem saznajemo da je „Max Profi Shop“ prodavnica tehničke robe, koja radi pod okriljem „Metalservis-a“, i bavi se prodajom računara (i ne samo njih) preko konsignacije. Dobili smo i cenovnik, sa deviznim iznosom i dinarskim delom. Da vidimo kako to izgleda:

C	64	235	1541	250	1530	32	1801	270	803	155
C	128	330	1570	265	1531	25	1901	380		
PC	10	1720	1571	315						

računari diskovi kaseteši monitori štampači

U tabeli vidite pojedini tip računara ili opreme i, pored njega, podvučen podatak o ceni u devizama (dolariti), a na tu cenu treba dodati i oko 65% u dinarima, na ime poreza.

U radnji možete kupiti i originalni džojstik, po ceni od 5,5 dolara, i diskete za oko

1,5 dolara po komadu. Zanimljivo je da imaju i kartridže sa igrama, koji koštaju oko 5—15 dolara, što je strabalo skupo, u odnosu na naše druge pirate. (Gornjim cenama pridodajte onih netbezbnih 65% u dinarima.)

Raspitali smo se za mehanizam prodaje. Rečeno nam je da robu dobijaju od „Koni-ma“: iz Maribora, i da sve isporuke zavise od njega i njegovih mesečnih isporuka. Za sve informacije možete zvati broj 011 (624—927) i saznati pojediniosti. Inače, kada roba stigne, radnja se ispunjuje kupcima, i po pedesetak njih, pa sve liči na pijacju ili mesarnicu. Radno vreme prodavnice je, radnim danom, od 7—19 časova, subotom od 8—15 časova, a adresa Karadorđeva 65, ali ne možete da omanete, jer se velika reklama vidj izdaleka.

Slobodno svratite, jer vam je to možda jedina prilika da vidite i opijate — to se ovde ne brani! C 128 ili PC 10. Na rstantku, u poverenju nam je rečeno da „Max Profi Shop“ pregovara sa „Amstradom“ oko konsignacione prodaje u Beogradu! Držimo im palčeve!

Drugi deo: melodrama u šopu

Sledeći na našem putu je bio, naravno, kompjuter šop, ili „Computer Shop“, radnja na koju pomisli 80% vlasnika računara kada krene da pazari nešto od opreme po Beogradu. Da li s pravom, videćemo...

Adresa prodavnice je Maršala Tita 48, ali je to, u stvari, ulica Generala Ždanova. Radnja je lepo uređena, ali se nigde ne vide računari, samo knjige, ovaj... ah, da, u jednom čošku sede dva mladića sa nekakvim računarom. Da im pridemo! Dogadaji dalje teknu kao u nekom skeču, pa ih tako i opisujemo:

- Mi: — Dobar dan!
 Oni: (čute). (U stvari, jedan od njih dvojice lupa po tastaturi „Amstrad“ i izbacuje iz usta sistemske poruke kao: Integer out of range, Syntax error, No data... itd. dok drugi priča sa nama).
 Mi: — Dobar dan! (opet)
 Oni: — Dobar dan, izvol'te.
 Mi: — Interesuje nas „amstrad“, videli smo oglas u novinama, šta prodaje te ovde, može li da se vidi cenovnik, može li da se proba nešto...
 Oni: — Da, da, to je bila greška, mi ne vršimo prodaju „amstrada“, to je palo u vodu, evo, to što imamo...
 Mi: — Da li možemo da isprobamo, recimo „spektrum“ sa „epsonom“?
 Oni: — Ne, žao nam je, ali to ne može, osim ako kupujete...
 Mi: — Što ste zagradili opremu stolom? Mislite da će vas neko pokrsti? (Imao sam nameru da iznesem „epson“ pod miškom)
 Oni: — (čute, onaj drugi sada više ne izbacuje sistemske poruke, već se divi kolor-grafici „amstrada“ u MODE 1, koju je skripio sa dve FOR-NEXT petlje)
 Mi: (Izlazimo, razočarani).

Koliko smo mogli da saznamo, u „Computer Shopu“ se snabdejavu samo preduzeća sa pozamašnim računima, jer, da li biste vi platili tri-četiri puta skupije „spektrum“, ili dali za domaći printer oko 45 miliona, da i ne govorimo o „epsonu“ koji se približava stratosferi...

Na žalost, a možda i srećom, jedina stvar koja vredi u „Computer Shopu“, osim računara, koji su prevredni, su knjige. Ali o tome...

Idemo dalje, tu su i komisijoni. Ima ih puno i nije ih lako sve obići za kratko vreme, a i nemaju bogzna šta da ponude. Prosečne cene računara (ako ih imaju) su: „spektrum“ — 90.000 do 130.000, „komodor 64“ — oko 200.000 din., „C-16“ — oko 120.000 din., dataset za „komodor“ oko 30.000, i to kopija disk-drajv 1541 190.000 din.; naišli smo i na video-igre koje koštaju 60.000 i šah-komputer od 60.000 din, ali to nije cilj ovog napisa.

Sve u svemu, žalosno. Izbor je loš, a računari su drastično precejnjeni. Uostalom, komisijoni i nisu mesta gde se kupuje prava roba. To dokazuje i komplet od 10 disketa (5,25) na kojima nametljivo piše „IBM“ i koje koštaju oko 25.000 din! Stop. Po robnim kućama i prodavnicama biro-opreme vlada sivilo, tek ponegde razbijeno kojejakvim ludorijama naše trgovačke umesnosti. Idemo redom:

Prodavnica „Prosveta“, M. Tita 6, inače odlično snabdevena papirnicama, ima (ko zna

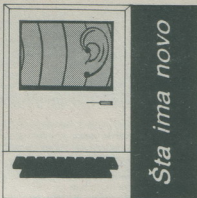
koliko?) ZX Interface 1 i ZX Microdrive. Svaki košta po 54.500 din. Na pitanje o poveklju robe i (eventualnim) budućim isporukama dobili smo odgovor u stilu: ko znate?

Treći deo: bežimo napolje

„Prosveta“ u Čika Ljubinoj 1a odnekud ima računare! U razgovoru sa simpatičnom prodavničom saznali smo da računare otkupljuju od korisnika (???) a onda ih prodaju: Divno! Imaju „spektrum“ za 80.000 din, C-16 (opet!) za 70.000 din. bez dataseta i ispravljača, a sa njima oko 160.000 din. I oni ne znaju šta će da dobiju i koliko. Lepo. Najsmješnije deo ovog istraživanja po asfaltu džungle na asfaltu na temperaturi od oko 32°C u hladu zbio se u prodavnici „Elektrotehne“ u ulici M. Tita (ona veća prodavnica). Tu imaju računar „Oric Nova 64“ i prodaju ga za lepe pare — 181.860 din. Teško je reći odakle im baš ta cifra; znamo samo da prodavač nije znao ama baš ništa sa njim i oko njega, non-stop tvrdeći da uz računare ide i nekakav medijum za spremanje podataka! Kada smo ga zapitali šta je to, izvodio je neku crveno-crnu kutiju ispod teške i trijumfalno nam je poturio pod nos. Posle dva minuta i blagog šoka koji je usledio, ustanovili smo da je ta kutija, u stvari, ispravljač računara, koji je nesrećnim slučajem, u istoj kutiji u kojoj bi trebalo da se nalazi 3-inčni disk-drajv!!! Naravno, drajva ni od korova, dok je prodavač odbio da sasluša naše predavanje o vrstama drajvova i njihovom izgledu. Svejedno, u prospektu koji se dobija od proizvođača, piše da postoji i drajv, i da će mu cena biti oko 460.000 din! Brzo, brzo, bežimo napolje!!! Što se tiče TV ili monitora, imate šarolik izbor mini-TV koje proizvode EI ili RIZ, a cena im se kreće od 60.000 do 90.000 din. U „Šumadiji“, Knez Mihailova 26, u prizemlju imate dobru papirnicu, dok na spratu vrvi od računara i pisaaćih mašina. U razgovoru sa prodavcima saznali smo da se „Šumadija“ bavi uvozom i zastupništvom mnogih firmi i da mogu da, ako se pokaze interesovanje, uvoze pojedine artikle koji su potrebni računardžijama. Tu spadaju diske, kasete, papir, trake za printere, itd.

Što se tiče računara, možete da pogledate „Iolu 8“ po ceni od 140.000/160.000 175.000 za verzije od po 16/24/32 K. Od pribora trenutno su imali diske od 8 inča (4000 din.), 5 streamer (3500), 3,5 inča (5250) i tu i streamer-trake kapaciteta 700K za oko 3400 din. „Šumadija“, inače, prodaje široku gamu ozbiljnih mašina i sistema za obradu podataka, što će zanimati firme, dok su cene za pojedince previelike. Napomenuto nam je da se, ako bude interesatna, možete kod njih potražiti i diske od 3 inča („amstrad“) i sve vrste traka za printere. Potrebno je da donesete svoju traku i da proverite da li imaju odgovarajuću.

Da rezimiramo, Beograd nije mesto u koje biste se mogli uputiti radi kupovine računara. Ako smo u ovom tekstu i propustili poneku prodavnicu, zastupnika ili male oglase, to je zbog toga što ni oni ne bi popravili ovako lošu sliku... To, uostalom, dokazuje i jedan par pocepanih patika, istrošenih lunjanje po vrelim ulicama Beograda, a sve u (uzaludnoj) potrazi za računarima... Minhen je, izgleda, bliže.



Šta ima novo

Modesti Blej na „meku“

Imali ste samo dve pasije u životu — crtanje stripova i računare. Tu je odmah bio i problem. Ne samo što niste imali vremena i za jedno i za drugo, nego ste i sve pare koje ste nekako zaradili crtajući stripove poručili na računarsku opremu. Ili, još gore, sve pare koje ste zaradili preprodajom programa potrošili ste na stripovski reprodromat. Evo rešenja. Program se zove Comic Works i služi za crtanje stripova na „meku“. Pomoću njega možete crtati, bojiti, urediti tekst i, naravno, mešati crtež i tekst na bilo kom delu stranice. Program vam još daje izvestan broj dodatnih elemenata kao što su balončići za govor, koji se mogu slobodno raspoređivati po strani. Program potražite od firme Mindscape Inc. 3444 Dunder Rd., Northbrook, Illinois US. Comic Works košta 79,95 dolara. (B.D.)

Kako podmladiti traku

Ako imate printer sigurno ste primetili da ta draga spravica gubi kasete sa trakama sa zastrašujućom proždrljivošću i katastrofalnim posledicama po vaše ionako milione količine deviza. Engleski stilne probleme imaju odavno, pa se zato našta firma koja je smislila kako da im pomogne. Firma Aladdin iz Berkvilsajera za trećinu cene nove tračne kasete osvajača, to jest boji stare. Bilo bi lepo kad bi se i kod nas neko setio da tako nešto radi. (B.D.)

Qualitas za kvalitet

Ako imate „ZX spektrum“ i neki od starijih „epson“ printera, a želite da štamirate u NLQ modu — nema problema! Idite do prve „Computer-shop“ tabie (ne kod nas) i kupite program „Qualitas“. Taj program omogućava viasnicima „spektruma“ koji pišu sa tekst-processorom „Tasword Two“, da imaju onu mogućnost ispis na papir koja se danas toliko čeni — NLQ.

Naime, program koristi dva prolaska glavne printera i mikro pomeranja valjka. Dalje, može se koristiti proporcionalni i „equal-space“ ispis sa pet vrsta slova, koja se mogu izmeniti pomoću editora slova koji se prilaže uz program. Program radi sa svim „EPSON RX-80“ printerima i njegovim kompatibilcima... uh.

Verzija za „Tasword Two“ košta 7,95 funti, a za „Tasword Tree“ će koštati (kada se pojavi) oko 8,95 funti.

20000 i to svih osam bita koje sadrži ta ćelija. Ukoliko nam je potrebno da pojedinačno menjamo neke od tih bitova, moraćemo da dovedemo čitavu ćeliju u akumulator, modifikujemo je, a zatim je celu vratimo u memoriju.

Memorija se, kao što smo videli, sastoji od ćelija koje su numerisane po svojim rednim brojevima. Pretpostavimo da našom računaru treba da naredimo da u ćeliju čiji je broj 100 upiše zbir sadržaja memorijskih ćelija čije su adrese 101 i 102. Bilo bi logično da se za takvu naredbu upotrebi oznaka:

```

ADD 101, 102, 100
ADD je skraćenica za naredbu „saber!“ i prate je tri adrese: adresa memorijske ćelije u kojoj je prvi sabirak, adresa drugog sabrika i adresa ćelije u koju treba staviti rezultat. Pokazalo se neizvodljivo da se u okviru svake naredbe pominju po tri adrese, pa su se ljudi dosetili registra — pogledajmo ekvivalent gornje naredbe:

```

```

LOAD A, (101) : Prenesi in akumulator sadržaj ćelije 101
ADD A, (102)  : Dodaj na sadržaj akumulatora sadržaj ćelije 102;
              : rezultat sabiranja smesti u akumulator.
LOAD (100), A : Prenesi sadržaj akumulatora u ćeliju 100.

```

Vidimo da umesto jedne naredbe sada moramo da iskoristimo tri ali da zato svaku od njih prati samo po jedna adresa. Uz adresu se, međutim, stalno pominje nekakvo slovo A koje predstavlja ime jednog registra procesora.

Osim operativne memorije koja se sastoji od šezdesetak hiljada ćelija, 6502 ima i nekoliko bajtova svoje privatne memorije koji su grupisani u registre; flip-floповi koji sačinjavaju te registre se nalaze unutar samoga mikroprocesora. U jednom od sledećih poglavlja ćemo detaljnije upoznati registre 6502.

... i privatna memorija

Sledeći termin koji treba da upoznamo je *flag*. Pretpostavljajući da ste razumeli šta je registar, reći ćemo da je *flag* jednobitni registar. Ovo, bar na prvi pogled, dolazi u kontradikciju sa našom ranijom tvrdnjom da mikroprocesor može da pristupa samo čitavim bajtovima memorije, ali je ta kontradikcija samo prividna: *flag*ovi nisu deo memorije već deo samog procesora i imaju veliku ulogu za njegov normalan rad. Pretpostavimo da treba u ćeliju 100 upisati broj 255 ukoliko se u toj ćeliji ranije nalazila nula ili broj nula ukoliko se u toj ćeliji ranije nalazilo bilo šta drugo. Posmatrajmo program sa slike 1 koji bi mogao da obavi tu operaciju.

```

LOAD A, 100 : Stari sadržaj ćelije 100 u akumulator.
COMPARE A, #0 : Uporedi sadržaj akumulatora sa nulom.
BRANCH EQ, NULA : Ukoliko je jednak (EQ), idi do mesta
                : u programu koje se zove "NULA".
LOAD A, #0 : Broj nula u akumulator...
STORE A, 100 : ... a zatim i u ćeliju 100.
BREAK : Kraj rada.

;NULA
; Ovdje se dolazi ako je sadržaj ćelije 100
; na početku bio 0.
LOAD A, #255 : Broj 255 u akumulator...
STORE A, 100 : ... a zatim u ćeliju 100.
BREAK : Kraj rada.

```

slika 1

Vidimo da je instrukcija COMPARE A, #0 bila iskorisćena da se testira sadržaj akumulatora, a zatim smo upotreabili naredbu JUMP EQ, NULA koja je predstavljala skok samo ako je odgovor na pitanje koje je postavila prethodna naredba bio potvrđen. Kako bi naredba JUMP „znala“ kakav je rezultat izvršavanja prethodne naredbe? Jedino tako što je naredba COMPARE A, #0 postavila neki *flag* mikroprocesora u stanje 0 ili 1. Taj *flag* će biti u stanju jedan ako je odgovor na pitanje „da li je sadržaj akumulatora jednak nuli“ potvrđen, a nula ukoliko je određen. Sada se naredba JUMP EQ, NULA može protumačiti rečima „idi na deo programa označen sa NULA ukoliko je *flag* jednak jedinici (ili, kako se to obično kaže, ukoliko je *flag* *setovan*), a produži sa izvršavanjem programa ako nije“.

Procesor 6502 ima više *flag*ova kojima su dodeljena specijalna značenja. O njima ćemo, jasno, opširno govoriti docnije; za sada je bitno da znamo da zamišljamo *flag* kao zastavicu koja se podiže ako je odgovor na neko pitanje potvrđen a spušta ako je određen; stanje te zastavice docnije možemo da testiramo i da menjamo tok izvršavanja programa u zavisnosti od njega.

Stek

Za kraj smo ostavili pojam *steka*. Obzirom da svaki mikroprocesor ima relativno malo registra, često će nam biti potrebno da neke od njih privremeno oslobodimo za obavljanje neke operacije

```

3099 : treba ispisati sa ekrana upise
3099 : u memorijske ćelije "acr"
3099 85 2A STA scr_low
309E AD 88 31 LDA proba_high
309E 85 2B STA scr_high
30A0 00 1F 99 JSR asrnam : Ispisivanje pokusaja na ekranu.
30A3 A9 20 LDA #ASC " " : Ispisuje se jedan blanko.
30A5 20 EF FF JSR oswrch
30A8 60 RTS : Povratak u glavni program.
30A9 : .tekst
30A9 : : tekstovi poruka:
30A9 0A EQU 10
30AA 5A 61 6D EQU5 "Zamislili broj između 0 i 1023!"
30C7 0D EQU 13
30C8 0A EQU 10
30C9 61 20 6A EQU5 "a ja cu ga pogoditi!"
30DD 0D EQU 13
30DE 0A EQU 10
30DF 0A EQU 10
30E0 50 6F 73 EQU5 "Posle svakog mog pokusaja pritisci!"
3103 0B EQU 13
3104 0A EQU 10
3105 20 2D 20 EQU5 " - M ako je tvoj broj manji;"
3121 0D EQU 13
3122 0A EQU 10
3123 20 2D 20 EQU5 " - V ako je tvoj broj veci;"
313E 0D EQU 13
313F 0A EQU 10
3140 20 2D 20 EQU5 " - P ako sam pogodio. "
3155 0D EQU 13
3156 0A EQU 10
3157 0A EQU 10
3158 00 EQU 0
3159 : .prompt
3159 53 74 61 EQU5 "Sta kazes za broj "
3168 0D EQU 0
316C : .pobeda
316C 50 6F 67 EQU5 "Pogodio sam (opet)!"
317F 0D EQU 13
3180 0A EQU 10
3181 0A EQU 10
3182 00 EQU 0
3183 : .promenljive:
3183 00 : .denja_low EQU 0
3184 00 : .denja_high EQU 0
3185 00 : .goranja_low EQU 0
3186 00 : .goranja_high EQU 0
3187 00 : .proba_low EQU 0
3188 00 : .proba_high EQU 0

```

slika 43

```

10 PRINT "Zamislili broj između 0 i 1023!"
20 PRINT
30 LOW=0
40 HIGH=1024
50 BROJ=(LOW+HIGH)/2
60 PRINT "Da li je to broj "; BROJ; " (V,M,P)?";
70 INPUT " " ; X
80 IF X= "V" THEN LOW=BROJ:GOTO 50
90 IF X= "M" THEN HIGH=BROJ:GOTO 50
100 IF X<>"P" THEN GOTO 60
110 PRINT "Pogodio sam (opet). "
120 PRINT
130 GOTO 10

```

slika 44

```

3000      .hi low_gane
3000 A2 00 LDX #0           : Ispisivanje uputstava.
3002      .uputstva
3002 BD A9 30 LDA tekst, X
3005 FO 07 BEQ igra
3007 20 EF FF JSR osvrch
300A E8      INX
300B AC 02 30 JMP uputstva
300E      .igra
300E A9 00 LDA #0           : Broj je između 0 ...
3010 8D 83 31 STA donja_low
3013 8D 84 31 STA donja_high
3016 8D 85 31 STA gornja_low
3019 A9 04 LDA #4           : ..., i 1024 = 840.
301B 8D 86 31 STA gornja_high
301E      .pokusaj
301E 18      CLC
301F AD 83 31 LDA donja_low : Racuna se (donja+gornja)/2.
3022 6D 85 31 ADC gornja_low
3025 8D 87 31 STA proba_low
3028 AD 84 31 LDA donja_high
302B 6D 86 31 ADC gornja_high
302E 8D 88 31 STA proba_high
3031 4E 88 31 LSR proba_high
3034 6E 87 31 ROR proba_low
3037      .pokusajl
3037 20 88 30 JSR ispisi_probu : Na ekranu se ispisuje 'proba'.
303A 20 EF FF JSR oardch      : Korisnik treba da pritisne taster.
303D 20 EF FF JSR osvrch     : Slovo koje je pritisnuto se ispisuje
3040      .PHA
3040 A9 0D LDA #&OD         : Prelazak u novi red.
3043 20 EF FF JSR osvrch
3046 A9 0A LDA #&A
3048 20 EF FF JSR osvrch
304B 68      PLA
304C      .PLA
304C C9 50 CMP #ASC "P"
304E F0 29 BEQ pogodilo
3050 C9 4D CMP #ASC "H"
3052 F0 16 BEQ manji
3054 C9 56 CMP #ASC "V"
3056 F0 03 BEQ veci
3058 AC 37 30 JMP pokusajl   : Ako nije P, H ili V, pita ponovo.
305B AD 87 31 LDA proba_low : Donja = proba.
305E 8D 83 31 STA donja_low
3061 AD 88 31 LDA proba_high
3064 8D 84 31 STA donja_high
3067 AC 1E 30 JMP pokusajl
306A      .manji
306A AD 87 31 LDA proba_low : Gornja = proba.
306D 8D 85 31 STA gornja_low
3070 AD 88 31 LDA proba_high
3073 8D 86 31 STA gornja_high
3076 AC 1E 30 JMP pokusajl
3079      .pogodio
3079 A2 00 LDX #0           : Ispisivanje trijumfalne poruke.
307B      .ispis_kraja
307B BD 6C 31 LDA pobjeda, X
307E FO 07 BEQ kraj
3080 20 EF FF JSR osvrch
3083 E8      INX
3084 AC 7B 30 JMP ispis_kraja
3087      .kraj
3087 60      RTS           : Kraj programa.
3088      .ispisi_probu
3088 A2 00 LDX #0           : Ispisuje se pokusajl kompjutera.
308A      .ispisi_prompt
308A BD 59 31 LDA prompt, X
308D FO 07 BEQ kraj_prompta
308F 20 EF FF JSR osvrch
3092 E8      INX
3093 AC 8A 30 JMP ispisi_prompt
3096      .kraj_prompta
3096 AD 87 31 LDA proba_low : 'osprna' zahteva da se broj koji

```

da bismo, po njenom završetku, restaurirali njihov raniji sadržaj. U takvoj situaciji bismo mogli da prepisemo sadržaj tog registra u memoriju, ali bi se takvo smeštanje pokazalo neracionalnim. Pogledajmo, zato, jednu od varijanti upotrebe steka:

```

PUSH A      : Register A se čuva na steku.
...         : Deo programa koji koristi akumulator.
PULL A      : Obnavlja sadržaj akumulatora.

```

Sadržaj registra A je i ovde prepišivan u neku memorijsku ćeliju, ali sada mi nismo morali da razmišljamo o njoj adresi: mikroprocesor je opremljen registrom S (*Stack pointer* ili *okazivač steka*) koji sadrži adresu memorijska ćelije koju treba popuniti brojem koji se šalje na stek; sadržaj ovog registra se automatski povećava ili smanjuje po svakoj operaciji sa stekom, tako da primenom PULL instrukcije uvek dobijamo podatke koji se „zapatili“ primenom prethodne PUSH naredbe. Stek se, osim toga, intenzivno koristi za realizaciju potprograma i prekida i posvetićemo mu dužnu pažnju u jednom od sledećih poglavlja.

Predstavljanje podataka

Brojevi mogu da se prikazuju u raznoraznim sistemima — dekadnom, oktalnom, heksadekadnom, binarnom ... Posmatrajmo, na primer, broj 1986. On je iskazan u pozicionom dekadnom sistemu. Ovo *dekadnom* potiče od toga što je baza sistema broj 10, što znači da oznaka 1986 predstavlja zbir sa slike 2.

$$\begin{aligned}
 1986 &= 6 \cdot 10^0 + 8 \cdot 10^1 + 9 \cdot 10^2 + 1 \cdot 10^3 \\
 &= 6 \cdot 1 + 8 \cdot 10 + 9 \cdot 100 + 1 \cdot 1000 \\
 &= 6 + 80 + 900 + 1000
 \end{aligned}$$

slika 2

Što se broja 100101100 tiče, on je iskazan u pozicionom binarnom sistemu. Ovo *binarnom* potiče od toga što je baza sistema broj 2, što znači da oznaka 100101100 predstavlja zbir sa slike 3.

$$\begin{aligned}
 100101100 &= 0 \cdot 2^0 + 0 \cdot 2^1 + 1 \cdot 2^2 + 1 \cdot 2^3 + 0 \cdot 2^4 + 1 \cdot 2^5 + 0 \cdot 2^6 + 0 \cdot 2^7 + 1 \cdot 2^8 \\
 &= 0 \cdot 1 + 0 \cdot 2 + 1 \cdot 4 + 1 \cdot 8 + 0 + 1 \cdot 32 + 1 \cdot 64 + 0 \cdot 128 + 1 \cdot 256 \\
 &= 0 + 0 + 4 + 8 + 0 + 32 + 0 + 0 + 256 \\
 &= 300
 \end{aligned}$$

slika 3

Vidimo da binarni broj 100101100 i dekadni broj 300 predstavljaju istu stvar izraženu na dva različita načina. Prvi način predstavljanja je pogodniji za računar, a drugi za čoveka i postoje postupci kojima se brojevi iz jednog sistema prevode u drugi. Ovi su postupci detaljno objašnjeni u „Računarima 15“.

Za mašinsko je programiranje posebno značajan *heksadekadni* sistem sa osnovom 16. Ovom predstavljanje podataka, osim cifara 0—9, zahteva i „cifre“ A, B, C, D, E, F, pa neki heksadekadni broj može da glasi &12A4 (& uvek označava heksadekadni broj). Konvertovanje heksadekadnih brojeva u binarne i binarnih u heksadekadne je izuzetno jednostavno: nisu nam potrebne nikakve table, programi ili algoritmi. Pokušajte, dakle, da zapamtite binarne predstave heksadekadnih cifara sa slike 4.

Dekadno	Binarno	Heksadekadno
0	0000	& 0
1	0001	& 1
2	0010	& 2
3	0011	& 3
4	0100	& 4
5	0101	& 5
6	0110	& 6
7	0111	& 7
8	1000	& 8
9	1001	& 9
10	1010	& A
11	1011	& B
12	1100	& C
13	1101	& D
14	1110	& E
15	1111	& F

slika 4

Operacije sa binarnim brojevima

Sa binarnim i heksadekadnim se brojevima računa kao i sa „običnima“ — treba samo steći rutinu i zametati standardnu tablicu sabiranja i množenja novom — u heksadekadnoj je tablici, na primer, 9+3=C dok je u binarnoj 1+1=10.

Mali problem predstavljaju negativni brojevi — za računanje nije mnogo prihvatljivo memorisanje minusa iz „-12“. Ako pretpostavimo da se svaki broj upisuje u jedan bajt memorije, mogli bismo da se dogovorimo da prvi (ili poslednji) ako, kao što je uobičajeno, bitove brojne zdesna bit tog bajta u sebi kodira znak tako da će se broj 7 pamtiti kao 00000111 a broj -7 kao 10000111. Ovakvo predstavljanje brojeva se, međutim, pokazalo nepogodnim za mnoge primene pa je izmisljeno nešto bolje — pamtjenje komplementa broja. *Prvim komplementom* broja 7 (00000111) nazivamo binarni broj koji dobijamo kada umesto svake nule u tom broju stavimo jedinicu, a nazivamo binarni broj koji dobijamo kada umesto svake nule u tom broju stavimo jedinicu, a dobijamo 11111001, što je *drugi ili potpuni komplement* broja 7 i, ujedno, zgodan način da se predstavi broj -7. Sami se uverite da se računanjem sa brojevima u potpunom komplementu dobijaju korektni rezultati. Sve u svemu:

1. U memoriju koju želimo mogu da se smeštaju označeni ili neoznačeni brojevi. U jedan bajt (osam bita) mogu da se smešte označeni brojevi između -128 i +127 ili neoznačeni brojevi između 0 i 255.

2. Pri sabiranju neoznačenih brojeva može da se pojavi prenos iz najvišeg razreda koji nazivamo *carry*. Pojava ovog prenosa pri računanju implicira postojanje greške tj. pokušaj da se sabere dva broja čiji je zbir veći od 255.

3. Pri sabiranju označenih brojeva *carry* treba ignorirati. Moguće je, međutim, da se pojavi preokretanje tj. greška u računu koji nazivamo *overflow*. Mikroprocesor 6502 omogućavaju programeru da provjeri postojanje greške (koja čini rezultat u akumulatoru besmislenim) i preduzme neku akciju ukoliko je ona nastupila.

Logičke operacije

Programeri koji svoja remek-dela pišu isključivo na višim programskim jezicima možda smatraju za su sabiranje, oduzimanje, množenje, deljenje i, eventualno, stepenovanje jedine operacije koje računaru može da obavi. Ukoliko se, međutim, upoznate sa nekim ko u životu piše jedino mašinske programe, verovatno ćete saznati da računari umeu da sabiraju (eventualno i da oduzimaju), da se sa množenjem silabo snalaze, da im deljenje predstavlja veliki problem ali da su im operacije čudnih imena *konjunkcija*, *disjunkcija*, *negacija* i *šiftovanje* najlažja strana to je razlog što ovim operacijama koje su vam verovatno silabo poznate posvećujemo čitavo poglavlje.

Konjunkcija je latinski naziv za logičku "i" funkciju dok je *disjunkcija* logičko "ili" — ove su operacije definisane tabelama sa slike 5; detaljne primere ovih operacija čete, takođe, pronaći u „Računarima 15“. Konjunkcija se koristi za brisanje pojedinih bitova nekog od registara, dok će nam disjunkcija poslužiti za njihovo setovanje; doncije ćemo naučiti mnogo više o ovim operacijama.

X	Y	X and Y	X	Y	X or Y	X	not X
0	0	0	0	0	0	0	1
0	1	0	0	1	1	1	0
1	0	0	1	0	1	1	0
1	1	1	1	1	1	0	1

slika 5

Osim konjunkcije i disjunkcije, korišćemo i logičku negaciju (NOT), čija je istinitosna tabela takođe data na slici 5. Primena ove operacije je sasvim jednostavna: svaka jedinica se pretvara u nulu, a svaka nula u jedinicu; NOT 11010101 je 0101010. Jedna od primena negacije je promena znaka; naučili smo da se *prvi komplement* broja dobija kada se svaka jedinica u tom broju pretvori u nulu, a svaka nula u jedinicu; sada znamo da se ovako „pretvaranje“ naziva *negacijom* broja. Ukoliko nam je potreban *drugi ili potpuni komplement* broja, rezultatu negacije ćemo prosto dodati jedan.

Šiftovanje

Posmatrajmo brojeve 115, 230 i 204 i pokušajmo da uočimo neku vezu između njih. Na prvi, drugi i treći pogled veza nema. Ukoliko imate malo smisla za enigmatiku, setičete se da govoriemo o logičkim operacijama sa binarnim brojevima pa čete naše brojeve predstaviti u tom obliku i dobiti *01110011, *11100110 i *11001100. Vezu postaje očigledna: drugi broj je dobijen kada smo odučili prvu cifru prvog i dodali mu nulu na kraju; treći broj je dobijen kada smo isto to uradili sa drugim. Sledeći broj u nizu bi, dakle, bio *10011000 i li 52.

Tvrđnju sledeći broj se dobija kada prethodnom dopišemo nulu na kraju i odučimo njegovu prvu cifru možemo da iskažemo i na drugi način: *sledeći broj se dobija kada sve cifre prethodnog*

Na slici 42 vidimo program koji premešta blok memorije proizvoljne veličine; pridodali smo bejzik program koji izlazuje prenos sadržaje ćelija &8000 — &9021 u ćelije &3A40, &3A41 ... Svaka je adresa podeljena na dva bajta: viši i niži. Tako imamo promenljive *oldlow* i *oldhigh* (niži i viši bajt adrese početka bloka), *newlow* i *newhigh* (niži i viši bajt adrese na koju blok treba da se premešti) i, na kraju, *pages* i *bytes* — promenljive koje određuju koliko se bajtova memorije premešta (u našem primeru &1021). Prve četiri promenljive moraju da budu smeštene na kultu stranu: sećamo se da indirektnu adresu mora da čuva neki od prvih 256 bajta memorije.

Program će raditi i u slučajevima kada se zahteva prenošenje manje od 256 bajta, pa čak i ako zahtevamo prenos 0 bajta. Program, međutim, neće uvek raditi ako se blokovi preklapaju: premeštanje ćelija &2000—&3000 u ćelije &2100 — &3100 će dati neželjene rezultate. Pokušajte da modifikujete program tako da korektno reši i ovaj problem!

```

3000      oldlow=&70
3000      oldhigh=&71
3000      newlow=&82
3000      newhigh=&73
3000      bytes=&75
3000      pages=&74
3000      .move block
3000 &46 74   LDX pages           ; Broj strana za prenošenje u X.
3002 &F0 10  BEQ &dec_strane     ; Da li se premešta vise od strane?
3004 &A0 00  LDX #0             ; Prenešta se vise od strane.
3006      .move
3006 B1 70   LAR (oldlow),Y
3008 91 72   STA (newlow),Y
300A C8      INT                ; Y broji prenete bajtove na strani.
300B D0 F9  BNE move           ; Da li je preneseno 256 bajta?
300D E6 71  INC oldhigh
300F E6 73  INC newhigh
3011 CA      DEX
3012 D0 F2  BNE move           ; Da li su prenesene sve strane?
3014      .dec_strane
3016      .move
3014 A4 75  LDY bytes
3016 F0 09  BEQ kraj           ; 256 bajta - dec strane.
3018      .move
3018      .ostatak
3018 88      DEY
3019 B1 70  LDA (oldlow),Y
301A 91 72  STA (newlow),Y    ; Prenosenje bajta.
301D C0 00  CPY #0
301F D0 F7  BNE ostatak
3021      .kraj
3021 60     RTS                ; Povratak u bejzik.
10 POKÉ oldlow, &0
20 POKÉ oldhigh, &0
30 POKÉ newlow, &40
40 POKÉ newhigh, &3A
50 POKÉ bytes, &21
60 POKÉ pages, &10
70 CALL &3000
80 END

```

slika 42

Primer 7: Prva igra

Možda ste se odučili za upoznavanje mašinskog jezika da biste zatim pisali igre (i prodavali ih Englezima)? Evo jedne sasvim jednostavne igre primerene znanju koje smo do sada stekli: High-Low.

Zamislite broj između 0 i 1023 i startujte naš program. Računar će ispisati neki broj, a zatim čete pritisnuti i ako je broj koji ste zamislili veći, M ako je manji, a P ako je računar pogodilo. Višećete da će svaki broj koji zamislite biti pogođen u najviše 10 pokušaja.

Igra je sasvim nezavisna od bejzika, što znači da ulaz i izlaz obežuje sopstvenim potprogramima koji, sa svoje strane, pozivaju rutine iz ROM-a čije adrese možete da saznate konsultovanim knjiga tipa „ROM Disassembly“, „Advanced User Guide“ ili naših ranijih umetaka. Potrebni su vam potprogrami koji ispisuju ASCII znak na ekranu, učitavaju slova sa tastature i ispisuju sadržaj dve memorijske ćelije na ekranu kao dekadni broj — nešto se slično sasvim sigurno nalazi i u ROM-u vašeg kompjutera.

Što se algoritma na kome počiva program High Low tiče, nećemo mu obraditi mnogo pažnje: radi se o običnom binarnom pretraživanju. Ukoliko vam je termin „binarno pretraživanje“ stran, pogledajte mali bejzik program koji je potpuno ekvivalentan bitnom delu našeg mašinka.


```

3000      .ascii_hex
3000 20 15 30 JSR konverzija_cifre  : Nizi bajt je u akumulatoru -
3003      : treba ga prevesti u binarni
3003      broj ...
3003 8D 1F 30 STA lownibble       : ... i upisati ga u 'lownibble'.
3006 8A      TXA                  : Visi bajt je u X -
3007 20 15 30 JSR konverzija_cifre : ... preveden u binarni broj ...
300A 0A      ASL A                : ... i na njega je dopisan
300C 0A      ASL A
300D 0A      ASL A
300E 0D 1F 30 ORA lownibble       : ... sadržaj 'lownibble'.
3011 8D 20 30 STA final           : Rezultat u memoriji.
3014 60      RTS                  : Povratak u bejzik.
3015
3015      .konverzija_cifre
3015      : Potprogram koji konvertuje
3015      : ASCII znak iz akumulatora u
3015      : binarnu vrednost.
3015 98      SEC
301B 89 30 30 SBC #A30           : Oduzima se ASCII "0".
3018 C9 0A      CMP #8A          : Ako je rezultat veći od 10,
301A      : radilo se o cifri A - F ...
301A 90 02      BCC kraj
301C E9 07      SBC #7           : ... pa treba oduzeti
301E      : ASCII "A" - ASCII "9" + 1.
301E      :
301E      .kraj
301E 60      RTS                  : Povratak u glavni program.
301F 00      .lownibble EQU 0
3020 00      .final EQU 0

```

slika 40

Primer 5: HEX-ASCII

Kao što nam je potrebno da pretvorimo niz znakova u broj, tako će nam zatrebati da pretvorimo broj (upisan u akumulator) u dva ASCII znaka, koje će, ispisana na ekranu, predstavljati heksadekadnu vrednost. Ovu konverziju možete da proverite programu HEX-ASCII, koji je dat na slici 41. Program lepo ilustruje tehniku koju zovemo „maskiranje“: Izdvajamo četiri niza bita, prevodimo ih u znak, a zatim šiftovanjem izdvajamo signifikantnija četiri bita i pretvaramo ih u drugu (zapravo prvu) cifru.

```

3000      .hex_ascii
3000 48      PRA                   : Akumulator se čuva na steku.
3001 29 0F      AND #0F           : Izdvaja se nizi nibl akumulatora...
3003 20 15 30 JSR konverzija_nibla : ... i konvertuje u ASCII znak...
3006 8D 1F 30 STA niza_cifra     : ... koji se smešta u memoriju.
3009 68      PLA                   : Izdvaja se visi nibl akumulatora...
300A 4A      LSR A
300B 4A      LSR A
300C 4A      LSR A
300D 4A      LSR A
300E 20 15 30 JSR konverzija_nibla : ... i konvertuje u ASCII znak...
3011 8D 20 30 STA visa_cifra    : ... koji se smešta u memoriju.
3014 60      RTS                  : Povratak u bejzik.
3015
3015      .konverzija_nibla
3015      : Potprogram koji konvertuje nibl
3015      : upisan u niza 4 bita akumulatora
3015      : i vraća njihove ASCII predstave.
3015 18      CLC
3016 69 30      ADC #ASC "0"
3018 C9 3A      CMP #ASC "9"+1   : Da li se radi o cifri A-F?
301A 90 02      BCC okey
301C 69 06      ADC #6            : Ako da, treba dodati 7 (6+carry).
301E      .okey
301E 60      RTS                  : Povratak u glavni program.
301F 00      .niza_cifra EQU 0
3020 00      .visa_cifra EQU 0

```

slika 41

Primer 6: Premeštanje bloka

Negodna strana 6502 je što nema šesnaestobitnih registara — za bilo koju operaciju nad blokom memorije koji je veći od 256 bajta moramo da se mučimo sa postindeksiranim indirektnim adresiranjem. Ovo je adresiranje, na prvi pogled, veoma složeno: morate da praitite promenu vrlene i nižeg bajta adrese i još da se brinete o Y registru. Savetujemo vam, ipak, da se što pre priviknete na indeksirana adresiranja: bez njih se jednostavno ne može programirati na 6502!

pomerimo ulovu za jedno mesto i na upražnjeno mesto poslednje cifre upišemo nulu. Ovo je uopšte i najjednostavnija definicija šiftovanja u levo.

Šiftovanje udesno je vrlo slična operacija, pri čemu bi se od broja %01110011 dobro najpre broj %00111001, a zatim broj %00111100. Ukoliko se radi o pozitivnim brojevima, šiftovanje ulavo postaje množenje sa 2, a šiftovanje udesno — celobrojno deljenje sa 2.

Procesor 6502 omogućava relativno složena logička šiftovanja brojeva o kojima ćemo još dosta govoriti. Za sada je važno samo znati da se kod tih šiftovanja vodeći ili krajnji bit ne gubi, već se prenosi u jedan od flegova nazvan Carry. Postoji mogućnost i da se bit iz Carry flega vрати u šiftovani broj, a tako da se „izgubljeni“ bit vrati na kraj broja; tako bismo od %10010001 dobili %00100011. Ovakva operacija se iz očiglednih razloga naziva rotiranjem nalevo.

Instrukcije i asembleri

Kada biste program koji smo napisali u prethodnom poglavlju pokušali da „saopštite“ mikroprocesoru koji upoznajemo, rezultati bi bili nikakvi. Pre svega, za zadanje instrukcija smo koristili neke reči: LOAD, STORE, COMPARE, BREAK ali smo prečukali način na koje bi one mogle da se unesu u računar. Možda da ih otkučate umesto bezijk programa? Pokušajte i postavite varijablu „syntax error“ ili, pošto ste naučili da se mašinski programi unose primenom instrukcija POKE, da otkučate POKE LOAD? Šilcan pozdrav! Da bi računar mogao da izvrši mašinski program on mu mora biti saopšten u prihvatljivom, kodiranom obliku!

Instrukcije se kodiraju u binarne brojeve, pri čemu kod svake instrukcije obično zauzima jedan bajt. Osim koda koji bi, na primer, zamjenjivao skraćenicu LOAD, računaru treba da saopštimo i adresni deo naše instrukcije (sećate se onoga LOAD A, #255?); za koji se obično koriste sledeći, sukcesivni bajtovi memorije. Različite instrukcije, zajedno sa svojim adresnim delom, mogu da zauzmu jedan, dva ili više bajtova, pri čemu ovaj broj varira od instrukcije do instrukcije.

Kada se dve instrukcije moraju pretvarati u brojeve, zašto izmišljati reči koje te brojeve zamjenjuju? Zato što je mašinski jezik i onako po prirodi težak i nečitljiv, pa postojanje naziva instrukcija (tzv. *mнемоничких скраћеница*) olakšava kako pisanje programa tako i njegovo docenje i prevajanje. Samu transformaciju reči u binarne brojeve je u stanju da obavi kompjuter pomoću programa koji nazivamo *assembler* i kome posvećujemo kraj ovog poglavlja. Mнемоничке skraćnice su stvar konvencije: možemo da se odlučimo da instrukciju čiji je kod &9A zovemo Instrukcija i asembleri

LDA ili LOAD ili samo L i bilo koja od tih varijanti će rezultirati istim mašinskim programima. Slično je i u bejziku: naredba PRINT, bi sasvim lepo mogla da se zove WRITE, DISPLAY ili TYPE. Konvencije su, međutim, većinom usvojene i ne možemo da ih menjamo na u ni na njih treba pritići. Posle određenog broja sati sedenja uz računar, naučite se bine skraćnice napamet i kucati ih bez mnogo razmišljanja.

Korišćenje mнемоничких skraćnica krije u sebi veliku opasnost: tendenciju da se uobzri postojanje nekih naredbi. Ukoliko, na primer, znate da postoji naredba JUMP (&2000), moguće je da po analogiji upotrebite i naredbu CALL (&2000) koja, na primer, ne postoji! Program koji smo nazvali assemblerom će vas, jasno, upozoriti na grešku ali vam to neće mnogo pomoći ako ste čitav segment programa zasnovani na nepostojećoj naredbi. Osobitni mikroprocesori zbog uštede memorije i jednostavnosti konstrukcije imaju razne modalitete adresiranja koji su pristiupećni samo kod nekih instrukcija i za njihovo korišćenje je potrebno, bar u početku, neprestano konsultovati tabelu naredbi. Korišćenje te table će biti upoznati kako ovaj tekst bude napredovao; za sada samo bicite jedan pogled na nju i, ukoliko za to imate ulovu, napravite par fotokopija — te će fotokopije postati redovan gost na vašem pisaćem stolu. U tabeli pobrojane skraćnice instrukcija, njihovih kodovi i načini adresiranja koje ih prate; dali smo, najzad, i kratak opis delovanja svake naredbe.

Memorija i adresiranje

Memoriju možete da zamislite kao niz ćelija sa slike 6. Svaka ćelija ima svoju adresu koja se kreće između 0 i 65535 (ili, ako vi volite heksadekadne brojeve, između 401 i &FFFF). Svaka ćelija se sastoji od osam bitova koji su obeleženi brojevima 0—7. Adresni prostor od 65536 bajta ili 64 kilobajta mora da bude podeljen na ROM u koji će biti smešteni operativni sistem računara i interpreter(i) programskih jezika kao što je bejzik i RAM u koji će biti upisivani programi koje korienik piše. Za sada ćemo reći da se ROM kod računara sa 6502 obično počinje od adrese &8000 ili &C000.

Većina instrukcija koje zadajemo mikroprocesoru se bavi manipulacijom (prenosjenjem, sabiranjem, testiranjem ...) brojeva koji su smešteni u neke memorijske ćelije, što znači da se u kod instrukcije mora navesti i adresa ćelije na osnovu koje se računar zaključuje da se nalaze podaci koje treba da obrađuje. Ta oznaka se zove *adresni deo instrukcije*. U zavisnosti od načina na koji mikroprocesor transformiše adresni deo instrukcije u podatke sa kojim treba manipulirati, razlikujemo razne *modove adresiranja* od kojih ćemo neke upravo upoznati. Vrlo je verovatno da sledeće redove nećete odmah razumeti — previle novih pojmova za početnika. Pa ipak, preletite ih pogledom i zapamtite da oni postaju: vraćate se na adresne modove kada budete upoznali prve instrukcije.

postoji. Ovakav intervert koriste autori igara da bi omogućili simultano odvijanje nekoliko radnji — programi koji upravljaju kretnjanim raznih monstaura veoma često predstavljaju deo interapt rutina dok se pozitivak nekog bi kontrolisete kreće uz pomoć glavnog programa. Na interaptima se zasniva i mogućnost nekih računara da ispišu tekst na štampaču dok se korisnik zabavlja partijom šaha ili da sviraču dok kucate program. Sve u svemu, interapti su izuzetno privlačni i korisna stvar (ne samo za hardverale!) i vredi ih pročitati daleko doprobnije.

Biblioteka programa

Primer 1: Bilo šta

Počemo od sasvim jednostavnog primera: želimo da računar izvrši sasvim kratak mašinski program koji radi bilo šta, u našem slučaju komplementira (zamenjuje svaku nulu jedinicom a jedinicu nulom) broj upisan u memorijsku ćeliju čija je adresa 83100. Sa slike vidimo da bejzik program za tastaturu uzima neki broj a onda ga, primenom naredbe POKE, upisuje u ćeliju 83100. Zatim se, primenom naredbe USR, poziva mašinski program koji je smešten počevši od bajta 83000 (možete ga, jasno, smestiti i na mnoga druga mesta), i, na kraju, ispisuje nova vrednost bajta 83100. Isprobajte ovaj program — pomoći će vam da bolje upoznate vaš asembler i njegove veze sa bejzikom.

```
3000 A3 00 31 LDA 83100      10 INPUT A
3003 A9 FF EOR #8FF        20 POKE 83100,A
3005 8D 00 31 STA 83100    30 CALL 83000,A
3008 60 RTS                40 PRINT PEEK(83100)
                                50 GOTO 10
```

slika 37

Primer 2: Vremenska petlja

Primer 6 predstavlja vremensku petlju — ako vam se ponekad učini da vaš mašinski program radi prebrzo, uposrite ga ovakvim dodatkom. Trajanje vremenske petlje zavisi od početne vrednosti u akumulatoru — ukoliko u njega upišete neki veliki broj (na primer &FF), petlja će potrajati pedesetak sekundi (ako računar radi na kloku od 2 MHz). Upišivanje jedinice u akumulator će, sa druge strane, izazvati veoma primetno čekanje.

Zanimljivo je činjenica da smo u X i Y registar upisivali nulu a zatim koristili DEX (ili DEY) i BNE ciklus. Posle prvog umanjavanja, nula postaje &FF a onda se sadržaj registra smanjuje sve dok ponovo ne stigne do nule. Verovatno bi stvar bilo razumljivija da smo napisali LDX &FF ali bi tada petlja trajala „za mrvicu“ kraće!

Ovaj primer treba da nam dokaže i brzinu mašina: isprobajte i odgovarajuću bejzik rutinu i uporedite vremena izvršavanja!

```
3000 A9 FF LDA #8FF          : Manji broj daje dužu petlju.
3002      .ciklus1
3002 A2 00 LDX #0
3004      .ciklus2
3004 A0 00 LDY #0
3006      .ciklus3
3006 88 DEY
3007 D0 FD BNE cfklus3
3009 CA DEX
300A D0 F8 BNE ciklus2
300C 18 DEC Y
300D E9 00 SBC #0          : Oduzimanje 1 od akumulatora:
300F D0 F1 BNE ciklus1    : A = A - 0 - (1-C) = A - 1
3011 60 RTS
```

slika 38

Primer 3: Šesnaestobitno množenje

Obećali smo vam „pravi“ program za množenje: množe se dva šesnaestobitna broja i dobija jedan tridesetobitni.

Program je relativno kratak, pa treba utrošiti po neku reč na njegovo detaljnije obrazloženje. Potrebne su nam, prve svega, po dve memorijske ćelije za množenik i množilac i čak četiri ćelije za rezultat. Kao i ranije, izdvojimo bit po biti množilca, ignorirajući nule. Kada pronađemo jedinicu, dodaćemo množenik proizvodu (oba bajta), a zatim liftovači proizvod za jedan bit udesno.

ovim zbirom. Ako se, na primer, u registru X nalazio broj 8333, naredba LDA 82000,X će u akumulator dovesti sadržaj ćelije čija je adresa 82000 + 833 * 2033. Nemojla je što se u registar X može upisati samo broj između 0 i 255 tako da u datom primeru, promenom vrednosti koju upisujemo u X, možemo da pristupamo samo ćelijama od 82000 do 820FF. Za operacije sa većim blokovima memorije treba da upoznato *post indekstrano indirektno adresiranje*.

Primer *post indekstrano indirektno adresiranja* je naredba LDA (820,X,Y). Mikroprocesor će, izvršavajući ovu naredbu, najpre pristupiti memorijskim ćelijama 820 i 821; neka je u prvoj našao broj 855 a u drugoj 870. Posmatrana dva broja spajamo i tako dobijamo 87055. Mikroprocesor će sada na ovaj broj dodati sadržaj Y registra (ako se u Y nalazio broj 6, rezultat je 87061) a onda pristupiti takoj ćeliji čija je adresa na taj nastali dobijena. Operacija je, sve u svemu, vrlo komplikovana ali čemo se na nju pospešno navikavati jer je za programiranje neophodna.

Procesor 6502 poznaje još par modova adresiranja (npr. *preindekstrano indirektno*) ali se oni retko koriste pa ih za sada nećemo pominjati.

Asembleri

Ulogu asemblera (ili asembler programa) već poznajemo: oni treba da prevedu program pisan mnemoničkim skraćenicama (zvaćemo ga *izvorni program*) u *objektni kod* — pravi mašinski program koji će biti upisan negde u memoriju. Ukoliko nameravate da vam ovaj umetak posluži za bilo šta osim za ukras, *našavite program za vaš kompjuter*. To nije naročito velika investicija (ovakvi programi su kod preprodavaca jeftini, valjda zato što silno koga interesuju), a predstavlja neophodan uslov za rad na mašinskom jeziku. Pretpostavljajući da asembler već posedujete, nećemo se baviti prevodjenjem instrukcija na pravi mašinski jezik; ukoliko dodate do toga da vam nešto takvo treba, verovatno ćete biti dovoljno dobar poznavalac mikroprocesora da se sami snadete. Ostatak ovog poglavlja je zato posvećen isključivo upotrebi asemblera.

Asembler je program kao i svaki drugi: pravi se u mnogo verzija i svaka od njih treba da ima svoje karakteristike i svoje uputstvo za upotrebu. „Računari“ su se bavili asemblerima za „komodor 64“; ukoliko koristite neki drugi računar ili asembler o kome nismo govorili, moraćete da pročitate originalno uputstvo što vam, u svakom slučaju, savetujemo. Videćete da se asembler sastoji iz *editora* koji omogućava da unete, ispravljate i smenite program na traku ili disketu i *prevodioca* koji ovako formiran tekst (kažemo tekst, jer su mnemoničke skraćnice obične reči) prevodi u objektni kod i upisuje ga u memoriju. Editori se razlikuju od asemblera dok prevodioci rade manje-više jednako što znači da čemo im posvetiti posebnu pažnju.

Da bi prevodioc mogao da smesti program u memoriju, treba mu saopštiti adresu od koje program počinje. Ta adresa se naziva *origin* i obeležava se ORG ili, po konvencijama firme MOS Technology, tačkom. Ukoliko se na početku programa pisanom mnemoničkim skraćenicama nađe ORG 83000 (ili X 3000), rezultujući mašinski kod će biti upisanu memoriju počevši od adrese 83000.

Reč ORG nije mašinska instrukcija pošto se ona ne prevodi u binarni broj koji bi bio smešten u memoriju. Ona, umesto toga, daje prevodiocu uputstvo o tome šta da radi sa pravim instrukcijama, pa se naziva *asemblerkom direktivom*. Asemblerskih direktiva, jasno, ima mnogo pa čemo ih upoznati kako nam budu trebale. Neke od njih imaju sasvim banalno i sa aspekta posla koji treba da se obavi nešto značenje: da li će se pri asemliranju listing prikazivati na ekranu ili će se ispisivati na štampaču, da li će računar prijavljivati sve greške ili samo one fatalne i slično. Neke druge direktive (kao što je ORG koju smo upravo upoznali) su fundamentalno važne.

Osim asemblerskih direktiva i mnemoničkih skraćnica, u asemblerskim listinzima mogu da se nađu i *pseudo instrukcije*. Pseudo instrukcije, iako ne predstavljaju ni jednu od naredbi koju bi mikroprocesor razumeo, u rezultujući kod upisuju određeni sadržaj. Pretpostavimo, na primer, da naš program treba da ispiše tekst „ZDRAVO“ na ekranu. Ispisivanje se, uzet ćemo najjednostavniji slučaj, obavija tako što se poruke ZDRAVO, slovo po slovo, prepisuje u video memoriju što znači da slova Z, D, R, A, V, I O moraju da budu upisana negde u RAM. Program koji rešava problem bi mogao da izgleda otprilike kao na slici 7.

```
LDX #0          : registar X će brojati ispisane znakove.
.petlja
LDI #poruka,X  : sledeće slovo poruke u akumulator.
BEQ kraj       : ako je u akumulatoru 0, ide se na deo
                : programa označen sa KRAJ.
JSR OSWRCH    : slovo se ispisuje na ekranu.
INX           : X se povećava za 1.
JMP petlja    : ispisuje se sledeće slovo.
.kraj
RTS           : kraj rada.
.poruka
EQU8 "Zdravo!" : poruka koju treba ispisati.
EQU8 13        : kursor na početak reda.
EQU8 10        : kursor na kraj reda.
EQU8 0         : marker kraja poruke.
```


Nije previše važno da shvatite kako tačno radi ovaj program; u globalu posmatrano, računar čita slovo po slovo poruke i ispisuje ga na ekranu. Obzirom da bi ispisivanje bilo "mrtva petlja", na kraj poruke je upisan nula bajt koga program detektuje (naredba COMPARE, CMP) i prekida sa radom.

Obraćamo pažnju na istaknute pseudo naredbe EQUIS i EQUB. Prva od njih označava da prevodilac treba u memoriju da upiše tačnu (string, odatle ono S) ZDRAVO, svako slovo u po jednu memorijaku ćeliju koristeći, naravno, ASCII kod. Slično tome, pseudo naredba EQUB zahteva od prevodilaca da u memoriju upiše bajt 00. Ukoliko nekom prijatelju doncije budete dali **mašinski** program poput ovoga, on će perfektno raditi. Međutim, ako taj prijatelj bude pokušao da shvati kako je program napisan, tj. da ga komentariše, on će se u velikom čudu kad bude pokušao da prevodi, odnosno, kako se to stručno kaže, *disasembliira* deo programa koji predstavlja poruku ZDRAVO. Naći će, naime, na bajtovie od kojih neki predstavljaju instrukcije dok su drugi nedefinisan. Čak i oni koji predstavljaju instrukcije će imati besmislene adrese delove I, upošte, neće ličiti na program. U sličnoj situaciji bi se nalazio i sam mikroprocesor da smo zaboravili na naredbu RTS koja prekida njegov rad: pokušao bi da izvrši test ZDRAVO kao mašinski program i pri tom verovatno upao u neku vrstu beskonačne petlje.

Iz čitave ove priče može da se izvede jedna izuzetno važna zaključak: od izvornog teksta programa se jednoznačno dobija objektna izjava uz pomoć programa koji nazivamo assembler. Obratni proces, dobijanja memorijski planiranog programa iz objektnog koda (disasembler) nije jednoznačan i, osim kod sasvim jednostavnih programa, nije ni malo lak. Vidimo, na primer, da smo u programu koji smo pisali označili tzv. *labelama* (o njima doncije) tri mesta koja smo nazvali "petlja", "kraj" i "poruka", pri čemu ti nazivi, opetirajući, asociliraju na namenu oznaka. Ukoliko neko disasembliira ovaj program, neće nikako moći da zna za ove naše oznake, pa i ako se praveće čitave stvari biti daleko komplikovnije. Zato **uvek i obavezno** čuvajte na traci ili disketi izvorne verzije svih programa koje napišete, ta verzija će vam biti dragocena ako jednom poželite da sprovedite neki svoj program ili da ga uklopite u nešto drugo.

U assembleriskim instrinzima se, kao što smo videli, pojavljuju i komentari koji su od "korisnog dela" naredbe odvojeni simbolom ";". Po pravilu se komentariše svaka instrukcija, što na prvi pogled izgleda prilično dosadno. Pokazalo se, međutim, da kucanje komentara može da pomogne ne samo u doncijim modifikacijama programa nego i u otkrivanju grešaka: ako sežmete funkciju neke naredbe u jednu rečenicu možda ćete primetiti neku nepravilnost u njoj! Bajzite program koji pišete sa komentarima je sporiji i zauzima daleko više memorije od bajzite programa bez REM-ova; razlog je u tome što se svako slovo komentara upisuje u memoriju baš kao i svako slovo instrukcije. Za razliku od toga, assembler u objektni kod prevodi samo korisne delove instrukcija, što znači da komentari bajzite potpuno ignorisani. Program koji dobro olupno komentariše neće, dakle, biti ništa duži ni sporiji od nekomentarisanih programa; biće jedino razumljiviji, što nije tako mala stvar!

Osim assemblera, za učenje mašinskih jezika će vam dobro doći program koji se naziva *monitor* ili *debuger*. Ovaj program vam omogućava da pretražujete memoriju i menjate njen sadržaj i, što je posebno važno, izvršavate mašinske programe instrukciju po instrukciju, posmatrajući promene u registrima i memoriji. Iskusni programeri koriste ovakve programe za traženje grešaka i tako štede svoje vreme dok će za početnike svakako biti interesantna mogućnost da, u nekoj vrsti "usporenog filma", proučavaju delovanje naredbi koje upoznaju. Reći ćemo, ipak, da upotreba monitor programa obično nije sasvim jednostavna i da sa njima treba da se upoznat tek onda, bar kada ne shvatite osnovne mašinske programiranja.

Pre nego što pređete na čitanje sledećih poglavlja koja će se baviti naredbama mikroprocesora 6502, treba da rešite nekoliko praktičnih problema koji će proisteci iz činjenice da programi koje ćemo sastavljati mogu da se izvršavaju na računaru što znači da je poželjno da neke od njih i otkucate. Pre toga morate da proverite na koje memorijske adrese treba da smestate mašinske programe kako se oni ne bi, "kossil" sa assemblerom i izvornim kodom (ako se, na primer, assembler nalazi u memorijskim ćelijama čija je adresa od 67000—8000, direktiva ORG 67500 će skoro sigurno izazvati kraj čitavog sistema jer će assembler, assemblerajući program, uništiti samoga sebe). Treba, osim toga, da proverite kako se mašinski programi pozivaju iz bajzika (za to obično služe naredba CALL ili funkcija USR) i kako se iz njih vraća u bajzik (obično mašinskom naredbom RTS).

Arhitektura 6502

U uvodnom smo poglavlju objasnili pojmove *registar* i *flag*; došlo je vreme da te pojmove upoznamo i na konkretnom primeru — arhitekturi mikroprocesora 6502.

Procesor 6502 ima osam registara opšte i, u toliko registara specijalne namene; ovaj broj, kada ga uporedimo sa Z80 i nekim drugim osobitnim mikroprocesorima, ne izaziva preveliko poštovanje, ali ne budimo zabrinuti — vidjećemo da je 6502 koncipiran tako da mu mnogo registara nije naročito ni potrebno. Dok programer koji počinje da radi sa Z80 može sebi priuštiti luksuz da privremeno zaboravi neke registre, korisnici 6502 moraju da uže malo truda i dobro zapamte šta su im konstruktori firme MOS Technology stavili na raspolaganje.

Upoznatu svrhu registara opšte namene nije teško — u njih smestamo podatke sa kojima operiramo. Jedan od tih registra (A) smo već upoznali — u njemu se obavlja većina aritmetičkih i logičkih operacija. Ostali registri opšte namene su takozvani indeks registri koji su obeležavali

Baskervilskog psa?) usred noći mahao svećom na prozoru a ribljaš Selden iz udaljene močvare posmatrao signal da bi poslao odgovor, radio se o „hand shakingu“: mikroprocesor u neku fiktivnu memorijsku lokaciju upisuje podatke i čeka da periferijski uređaj taj podatak pokupi. Kada periferija pokupi podatak, mikroprocesoru se šalje signal preko nekog drugog port-a i on nastavlja sa normalnim radom. Ova je tehnika daleko od savršenstva, u šta ćemo se verovati na jednostavnom primeru sa slike 36.

Da vidimo kako ovaj program funkcioniše. Protokolom je predviđeno da periferija smatra da je poslala za pripremu sledećeg znaka, resetuje štand bi onoga što računar smatra memorijskom ćelijom &FE11, a da tada računarski slovo treba štampati upiše u ćeliju &F1E0. Program neprekidno učitava vrednost sa porta 255 i testira njen sedmi bit; dok je on setovan, sve se okreće u „mrtvoj petlji“. U trenutku kada štampać bude spreman, mikroprocesor će mu poslati sledeće slovo i ponovo čekati da se zahtevana operacija izvrši.

Ovakvo rešenje je primenjivo ali sporo — mikroprocesor se vrti u beskonačnoj petlji, čekajući da štampać završi svoj posao i za to vreme ne može da radi ništa drugo. Biti bi daleko bolje kada bi mikroprocesor zadržao posao štampaća, a zatim radio nešto drugo dok štampać ispisuje slovo. Printer bi, po završetku posla, nekako dojavio mikroprocesoru da je ponovo spreman i ovaj bi mu pokupio sledeći znak. Štampać će mikroprocesoru dojaviti srećnu činjenicu da je završio ispisivanje korišćenjem takozvanog *interapta* (prekida).

Ako ste nekada rasklopili vaš računar i pogledali mikroprocesor, videli ste da se radi o jednom crnom kucištu sa mnogo nožica. Jasno je da svaka od tih nožica ima po neku funkciju — neke omogućavaju adresiranje memorije (adres bus ili adresa magistrala), neki prenošenje podataka (data bus ili magistrala za podatke), neki određuju da li je u tom trenutku potrebno učitavanje podataka u registar ili njegovo upisivanje u memoriju (R/W) i slično. U ovom trenutku nas posebno interesuju pinovi koji su obeleženi kao NMI (*Non Maskable Interrupt*) i IRQ (*Interrupt Request*).

Došlo je izvršno nekoj instrukciji, mikroprocesor će obavezno testirati logičko stanje nožice obeležene sa NMI. Ukoliko je ona na potencijalu od oko 5 V, nema nemaskirano interapta i mikroprocesor mirno prelazi na dalja ispisivanja. Ukoliko, međutim, konstatuje da je nožica NMI na potencijalu od oko 0 V, nastaje prava mala uzbuza: trenutni sadržaj registra P i PC (tj. stanje svih fleгова i adresa naredbe koja bi se sledeća izvršila da nije bilo interapta) se smestaju na stek, a zatim se u registar PC upisuje bajtovi iz memorijskih lokacija &FFFA i &FFFB — možemo da zamislimo da je periferijski uređaj naterao mikroprocesor da izvrši JSR (&FFFB) Ovaj program treba da sačuva sadržaje svih registara koje namerava da koristi na steku (P registar je već učitatski sačuvan) da uradi posao zbog koga je pozvan, vrati registre u ranije stanje i izvrši instrukciju RTI (Return from Interrupt). Posle toga se nastavlja prekinuto program koji upošte ne zna da je bio prekinut. Nemaskirani interapt se obično koristi za rad sa raznim periferijskim uređajima kao što su hard diskovi i, posebno, za komunikaciju sa mrežama računara; kod nekih jednostavnijih modela računara NMI se koristi isključivo za prekidanje „mrtvih petlji“ koje su nastale usled greške programera. Sve u svemu, upotreba nemaskirano interapta je uglavnom rezervisana za konstruktore operativnog sistema vašeg komputera.

Maskirane interapte, sa druge strane, koriste kako konstruktori operativnog sistema tako i korisnici računara. Pošto je mikroprocesor ustanovio da nema nemaskirano interapta, ispituje stanje nožice označene sa IRQ. Ukoliko u istom trenutku da se ona nalazi na potencijalu od oko 5 V, interapta nema, pa mikroprocesor izvršava sledeću instrukciju. Ukoliko je na ovoj nožici logički nula (oko 0 volt), detektovan je maskirani interapt.

Pošto je detektovao postojanje maskiranog interapta, mikroprocesor ispituje stanje i (*Interrupt Disable*) flega. Ukoliko je ovaj fleg setovan, interapt je maskiran, tj. programer je onemogućio njegovo prihvatanje i mikroprocesor će ignorisati zahtev za prekid i preći na izvršavanje sledeće naredbe. Ukoliko, sa druge strane, ustanovi da je I resetovan, mikroprocesor će sačuvati P na steku, setovati I i zabranjivati dalje interapte i preći na izvršavanje rutine za obradu prekida. Adresa ove rutine se nalazi u memorijskim ćelijama &FFE1 i &FFF7 — na samom kraju ROM-a. Izvrši rutinu, vrati se na stek i vrati registre koje su sačuvane na steku, da izvrši obradu koja treba da bude što kraća i najzad, da, posle vraćanja registra u ranije stanje izvrši RTI, čime će novi prekidi biti omogućeni. Primenom naredbe CALL možemo da dozvolimo prekide i ranije u toku izvršavanja interapt rutine (pošto su registri sačuvani), ali se to radi samo u slučajevima da postoji hardver koji reguliše prioritete prekida; kod osobitnih kućnih računara prekidi su skoro bez izuzetka onemogućeni dok se izvršava interapt rutina.

Procesor 6502 ima i jedan specijalitet kojim se korisnici Z80 ne mogu pohvaliti: softverski interapt. Kada mikroprocesor naiđe na instrukciju BRK (Break), ponaša se skoro kao da je dobio interapt i PC se smestaju na stek i izvršava JMP (&FFF7). Postoje, ipak, dve razlike između hardverskog i softverskog interapta. Pošto se na BRK ispisuje instrukcija, pre svega, setuje B fleg, čime interapt rutini signalizira da uzrok prekida ne treba tražiti na periferiji. Osim toga, RTI ne izaziva povratka na instrukciju iza BRK — obavezno se prekašće jedan bajt!

Softverski interapti se u sistemskom programiranju uglavnom koriste za prijavljivanje grešaka: kada operativni sistem detektuje grešku, izvršice BRK pri čemu će sledeći bajt ROM-a biti kod (0—255) greške koji omogućava interapt rutini da ispiše odgovarajuću poruku. Ukoliko nije jednostavniji računari koristi prekide isključivo za osvežavanje internog časovnika i crtanje slike, što znači da prekidi nastupaju u pravilnim vremenskim intervalima, na primer 50 puta u sekundi. Stoženim (i skupim) računarskim nije potrebna sadržaj mikroprocesora pri generisanju

nema smisla izvlačiti u glavni program, jednostavno PLA-PLA će skloniti sa steka svaki trag da je program program upotrebite pozivani. Novi sadržaj akumuladora u tom slučaju treba ignorirati.

Često je potrebno da se potprogramu prenesu neke vrednosti koje će obradivati i da se u glavni program vrate rezultati te obrade. Argumente potprograma je najzgodnije upisati u neke od registara. Ukoliko potprogram ima više od tri argumenta, možete ih upisati u neke memorijske ćelije sa fiksnim adresama, odakle će ih potprogram „pokupiti“; u tom slučaju potprogram ne sme da poziva samog sebe, ali su vam ovakve tzv. rekurzije za sada sasvim nepotrebne. Moguće je, najzad, smestiti sve argumente na stek a onda izvršiti JSR. Potprogram tada najpre mora da sačuva adresu povratka (npr. u ćelijama &2000 i &2001), zatim da obradi argumente postepeno ih skidajući sa steka, da smesti rezultate na stek i da, konačno, izvrši JP (&2000). Sve u svemu, prilično komplikovano.

U ROM-u vašeg računara se nalazi mnogo korisnih potprograma koji vas oslobađaju briga o pisanju po ekranu, skeniranju tastature, generisanju zvuka i slično: zar nije lakše napisati JSR GH i u akumulatoru dobiti kod pritisnutog tastera nego ispitivati dirke jednu po jednu? Da biste, međutim, napisali JSR OSDRCH morate da znate gde se tačno u ROM-u nalazi rutina koja skenira tastaturu (ako je ona smeštena poredovi od adrese &FE00 kao kod BBC-ja i „Electrona“), smestite vašeg programa čete napisati SRDCH (&FE00), kako treba pripremiti njen poziv i gde su smešteni rezultati, i kakve podatke možete da nađete u knjigama tipa „The Complete ROM Disassembly“, „Advanced User Guide“ ili u našim umcima. Nabavite, ako je to ikako moguće, literaturu u kojoj je opisan ROM vašeg računara i videćete da će ona postati najkorisnija referenca u praktičnom radu!

Periferija i interapti

Sve instrukcije koje smo do sada upoznali bave se prenošenjem podataka po memoriji i njihovom jednostavnom obradom. Pitanjem „odakle podaci u memoriji“ se nisamo mnogo bavili: obzirom da su obezbeđeni potprogrami u ROM-u koji će nam, čak i kada se oslobodimo od bezjika, omogućiti primanje podataka i izdavanje rezultata, postojanje instrukcija za ulaz/izlaz nas ne mora interesovati. Verujući, međutim, da će deo čitalaca ovog umetka jednoga dana pokušati da sagradi neki mali mikroprocesorski sklop koji je nezavistan od računara, posvetili smo ovo poglavlje komunikaciji mikroprocesora sa „spoljnim svetom“.

Procesor &502 nema posebnih instrukcija za ulaz i izlaz podataka — komunikacija se obavlja preko memorijske mape. Pretpostavićemo da je račun povezan sa štampačem i da je štampaču dodeljena memorijska lokacija &FE10. Ukoliko želimo da se na papiru pojavi slovo A, izvršimo program LDA #ASC „A“: STA &FE10.

Reklo bi se da ovo što smo radili nema veza sa štampačem: upisali smo neki broj u ćeliju čija je adresa &FE10. Ova je ćelija, međutim, fiktivna: tu ne postoji nikakav RAM koji bi primio podatak. Umesto toga, hardver će dekodirati (prepoznati) trenutak kada se na adresnoj magistrali mikroprocesora pojavila adresa &FE10 i podatak sa magistrale za podatke proslediti do nekog od portova.

Sa periferije ćemo ponekad i uzimati podatke, za šta koristimo obično LDA. Nema, naravno, nikakvog smisla uzimati podatke sa izlaznih portova: ako bismo u gornjem primeru, posle STA &FE10, pokušali da vidimo šta je upisano u ovu ćeliju (sa LDA &FE10), teško bismo tako pronašli ASCII kod slova A — podatke treba uzimati samo sa ulaznih portova. Da bismo vas sasvim zbudili, reći ćemo da je ponekad isto memorijskoj lokaciji dodeljen ulazni i izlazni port: koristimo, dakle, STA &FE10 i LDA &FE10, pri čemu nema nikakve veze između brojeva koje upisujemo i brojeva koje čitamo!

Ako ponosno pokušate da примените novostečeno znanje i napišete program koji će, na primer, slati podatke posredstvom raznoraznih memorijskih lokacija, desiće se razne čudne stvari: možda će zvučnik zapfati, možda će se na ekranu pojaviti besmislice, možda ćete uništiti sadržaj diskete (ako ste toliko neverovatni baksuz da pogodite kako se trake formatiraju), možda će se računar „izgubiti“ ili se, na vašu veliku žalost, neće desiti baš ništa! Da biste smisljeno koristili memorijski mapiranu periferiju, morate da znate određene protokole: koja je lokacija povezana sa kojim uređajem i na koji način uzimati i dajati primljene podatke. Moraćete, dakle, da konsultujete literaturu koju ste dobili uz računar, domaće i strane kompjuterske časopise ili knjige koje se nude na stranom tržištu. Suvršeno je reći da je rasporod portova sasvim različit od računara do računara i da nam u ovom umetku opšte prirode nećemo posvećivati posebnu pažnju.

Najjednostavniji mehanizam komunikacije mikroprocesora sa spoljnim svetom je takozvani „hand shaking“ (u bukvalnom prevodu — rukovanje). Kada je gospodin Barimor (čitali ste slike; oni koriste interapte za mnoge druge radnje, ali interapt koji periodično nastupa i dalje

.cekaj	LDA #0		
	LDA &FE11	:	čekaj dok printer ne bude spreman
	ASL A		
	BCC cekaj		
	LDA poruka,X	:	sledeci znak poruk u akumulador
	BEQ kraj	:	tekst ne ispisuje dok se ne naidje na 0
	STA &PE40	:	karakter ne šalje štampaču
	BSI	:	ofset adrese sledeceg karaktera u X
	JMP cekaj		
.poruka	EQU8	"Ovo se stampa"	
	EQU8	0	

slika 36

slova X i Y. Svaki od njih može da sačuva po jedan osobitni broj, ali se oni registri daleko intenzivnije koriste pri indeksiranju.

Registri specijalne namene se označavaju sa PC, S i P; PC je šesnaestobitni, a S i P su osmibitni registri. U setu instrukcija nećete naći nijednu jedinu koja operiše sa sadržajem registra PC, a ipak se taj sadržaj neprekidno menja. PC je, naime, skraćeniца od Program Counter ili, u bukvalnom prevodu, programski brojač. U ovom se registru nalazi adresa instrukcije koju mikroprocesor treba da obradi; tj. adresa bajta koji će, pri normalnom izvršavanju programa, biti sledići pročitani. Kako se neke instrukcije sastoje od više bajtova, registar PC će u toku njihovog izvršavanja nekoliko puta menjati vrednost. Pretpostavimo da je na redu izvršavanje instrukcije LDA &4000 (Load Accumulator) koja je kodirana kao &AD &00 &40 i, kao što vidimo, zauzima tri bajta memorije; &AD je oznaka instrukcije LDA, a &00 i &40 predstavljaju adresu &4000 koja se, kao i uvek, prikazuje sa nižim bajtom napred. Pre izvršavanja instrukcije u PC će biti upisana adresa memorijske ćelije koja sadrži broj &AD — početak instrukcije koja se obrađuje.

Mikroprocesor najpre uzima kod instrukcije (&AD) i smesta ga u registar naredbi koji je za nas nevidljiv. Istovremeno se registar PC povećava za 1 tako da pokazuje na bajt &00. Mikroprocesor prepoznaje instrukciju LDA i primećuje da mu je potrebna šesnaestobitna adresa, pa zahteva učitanje njenog nižeg bajta na koji pokazuje PC. U registar adrese (ponovo nevidljiv za nas) dolazi broj &00, a PC se povećava za 1 tako da pokazuje na broj &40. Sledi još jedno pristupanje memoriji i prenošenje broja &AD u registar adrese, pri čemu se PC ponovo povećava i pokazuje na sleduću instrukciju koju mikroprocesor treba da izvrši. Pre nego što dođe do nje, mikroprocesor će, jasno, morati još jednom da pristupa memoriji da bi pročitao sadržaj ćelije &4000 i smestio ga u akumulator.

Iako to u setu instrukcija nije izričito spomenuto, možemo lako da upisujemo brojeve u registar PC: Instrukcija JMP &3A00 (JUMP) bi slobodno mogla da se zove i LOAD PC, —&3A00. Posle izvršavanja ove instrukcije mikroprocesor će nastaviti normalan rad, ali neće izvršavati sleduću naredbu već onu čiji je broj „nasilno“ upisan u PC — u našem slučaju naredbu čiji je početak upisan u ćeliju &3A00. Omogućeno nam je i da dodajemo konstante sadržaju registra PC — takva dodavanja smo već upoznali pod imenom *relativni skokovi*.

Registar S (*Stack pointer* ili pokazivač steka) ima nekih sličnosti sa registrom PC: njega vrlo retko direktno menjamo (obično se takva operacija obavlja negde na početku ROM-a) ali su njegove implicitne promene vrlo česte. Kada nam treba da b privremeno sačuvamo sadržaj nekog registra upotrebimo instrukciju poput PHA (push accumulator). Tada će sadržaj akumulatora biti prekopiran u memorijsku ćeliju na koju pokazuje S a onda će S biti umanjen za 1 kako bi pokazivao na memorijsku ćeliju u koju će biti upisan sledići broj koga budemo stali na stek.

IRQ/BRK vektor	FFFF
RESET vektor	FFFD
NMI vektor	FFFC
	FFFB
	FFFA
Memorijski mapirani periferijski uređaji.	
ROM: operativni sistem itd.	
Slobodan RAM	
	200
Stek: raste od &1FF prema &100	
	100
Nulta strana: privil. lokucije	
	0

slika A

Soprotu proceduru će izvršiti naredba PLA (pull accumulator) koja će, kao krajnji rezultat, prenети u akumulator broj sa vrha steka i povećati S za jedan. Umesto PHA i PLA, sa svakom operišu i naredbe PHP i PLP koje su zadužene za čuvanje statusa svih fleгова na steku.

Uđimo da je stek na neki način soprotan programu: registar PC se povećava kada obradimo instrukciju što znači da program „raste“ od nižih ka višim adresama. Soprotno tome, sadržaj S se smanjuje kada stavimo broj na stek što znači da ovaj „raste“ ka nižim adresama.

Pre nego što nastavimo da opismujemo registre, moramo da objasnimo jednu nasko neobičnu stvar vezanu za stek: rekli smo da *osobitni* registar S pokazuje memorijsku lokaciju u koju će biti smešten podatak iz akumulatora kada primenimo instrukciju PHA. Osm bitova je, međutim, sasvim nedovoljno da se ukaže na neku memorijsku ćeliju: memorijskih ćelija ima 2¹⁶ = 65536 dok se sa osam bita može odabrati samo jedna od 2⁸ = 256! Stek se, po konvenciji mikroprocesora 6502, smešta u memorijske ćelije čije su adrese 256–512 (&100–&1FF) — na prvu stranu memorije.

6502 je mikroprocesor koji vrlo jasno deli memoriju na stranice veličine 256 bajta. Šesnaestobitnu adresu neke ćelije možemo naime, da podelimo na dva osobitna dela: prvi će specificirati jednu od 256 stranica a drugi jednu od 256 reči u toj stranici. Tako adresa &21C označava šezdesetu ćeliju (&3C–&60) na trideset treću (&21–&33) stranicu memorije. Strana 1 (&100–&1FF ili &256–&511) je, rekli smo, rezervisana za stek, strana nula (&0000–&00FF), je, tako što ćemo videti, posebno privilegovana dok se poslednji bajtovi strane &FF koristi za određene vektore o kojima ćemo još govoriti. Ostale strane memorije su sasvim ravnopravne i dele se na prostor koji pripada ROM-u (obično poslednje strane memorije) i na RAM. Na slici 6 je prikazana principijelna memorijska mapa računara koji koriste 6502.

Možda je već prebrzo reći, kada se pišući prve programe, u nekoj petlji greškom izvršiti mnogo PHA instrukcija koje će, bajt po bajt, preneti stek. Kada se stek kružno povećava, najpre će biti spusti do nule, njegova će vrednost ponovo postati &FF i stek će se stek kružno povećavati, unistišavajući svoj prethodni sadržaj. Ova situacija obično ne dovodi do potpunog kraha sistema (kod računara Z80 će mnogo PUSH-ova „izbombardovati“ kompletan RAM i unistiti program) ali svakako onemogućava normalan završetak mašinskog programa i rad sa potprogramima.

Došli smo, najzad, do registra P kome ćemo posvetiti dužnu pažnju. Već smo rekli da mikroprocesor poseduje određeni broj fleгова koji se koriste za kontrolu toga programa i neke aritmetičke i logičke operacije. Svi ovi fleгови su, fiktivno, svrstani u registar P kako bi njihovo stanje moglo da se čuva u okviru samog PHA (push P) i donosije restaurira naredbom PLP (pull P). U daljem tekstu ćemo pobliže opisati neke flebove registra P i opisati njihovu funkciju; za sada ćemo samo površno pročitati ovaj opis ali ćete se na njega svakako vraćati kada budemo dati primere upotrebe fleгова.

Bit nula P registra se zove „Carry“, i, naravno, obeležava slovom C. Ovaj bit se setuje kada se pri binarnom sabiranju pojavi prenos iz najstarijeg razreda (sabiramo, na primer, &33 i &FA; pokušajte) ili kada se pri oduzimanju *ne pojavljuje* pozajmica (oduzimamo manji broj od većega). Postojanje ovoga flega, kao što ćemo videti, omogućava jednostavno sabiranje i oduzimanje brojeva koji su smešteni u nekoliko memorijskih ćelija. C fleg, isto tako, predstavlja deveti bit akumulatora pri siftiranju i ima da onemogućava prenošenje bitova iz devetog broja u drugi.

Bit jedan registra P, je, bar za početnike, najznačajniji i najviše korišćen fleg: obeležavamo ga sa Z i zovemo Z fleg. On se automatski setuje kada je rezultat neke operacije nula ili kada je test dato rezultatu „True“ (tačno) i kao takav je neobično koristan za realizaciju struktura koje odgovaraju bejzik naredbi IF A=B THEN . . .

Bit dva se obeležava sa I i zove *Interrupt disable bit*. Programer ga setuje kada želi da njegov program ne bude prekidan zahtevima sa periferije (o interaptima ćemo još govoriti). Savetovali bismo vam da za sada ne dirate ovaj bit jer će njegovo trajno setovanje verovatno onemogućiti normalan rad vašeg komputera.

Bit tri, *Decimal mode flag* ili D, se koristi za izbor binarnog ili binarno kodiranog decimalnog moda rada. Kada savladate osnovne programiranja na 6502, pomoći ćemo vam da upoznate vrednost ovoga flega; za sada će on stalno biti resetovan.

Bit četiri, *break flag*, je za vas još manje značajan; on omogućava razlikovanje spoljnog i unutrašnjeg interapta. Njime ćemo se baviti na samom kraju ovog umetka.

Bit pet je uvek setovan — ne koristi se.

Bit šest registra P, je, najzad, značajan; označava sa V (od *Overflow*) i ima funkciju koju nije teško razumeti: setuje se ako je rezultat prethodnog sabiranja ili oduzimanja nekoristan usred prekoračenja (povećanje uvodno poglavlje i sabiranje označenih brojeva) a resetovan ako je rezultat korektan.

Značuju da bit sedam svakog bajta nosi znak broja, nećete se iznenaditi kada pročitate da se bit 7 registra P zove *Negative flag* i obeležava sa N. On je setovan kada je rezultat neke operacije negativan tj. ako mu je najviši bit setovan. Ovaj fleg se koristi za realizaciju mašinskih struktura koje odgovaraju bejzik naredbi IF A<B THEN . . .

Prenošenje podataka

Prva grupa instrukcija mikroprocesora 6502 kojom ćemo se baviti su naredbe koje su zadužene za prenošenje podataka između memorije i registra. Neke od ih instrukcija smo već upoznali: u dosadašnjim primerima smo često koristili nešto poput LOAD A, &20 što je trebalo da znači

učitaj me, obuhvataj sve flebove“. Pogledajmo, najpre, kako radi PHA (PUSH A):
STA &100+S
DEC S

Ovaj program ne bi radio ako biste ga otkucali, jer 6502 ne poznaje ni jednu od dve korišćene instrukcije. I pored toga, on rečito pokazuje sve što se dešava: akumulator se prepisuje u ćeliju čija je adresa na *prvoj strani* dela sadržaj registra S, a zatim se ovaj registar umanjuje za jedan. Kada nam, nekoliko instrukcija donjice, bude potrebno da vratimo podatak sa steka u akumulator, izvršićemo PHA (PULL A) odnako:
LDA &100+S

Šta se dešava ako napišemo PHA:PHP? Na stek najpre biva smešten sadržaj akumulatora, a zatim sadržaj P registra, što znači da za vraćanje podataka u registre treba koristiti PLP:PLA a ne PLA:PLP. Zbog toga se stek naziva i LIFO struktura (LIFO=last in, first out): poslednji podatak koji stavimo na stek će biti prvi koji ćemo, primenom PULL, pročitati.

Jasno je da, izvršavajući PLA, mikroprocesor ne može da zna da li je bajt koji pronade na steku nekada bio u akumulatoru; on se samo mogao naći i posle PHP. To nam daje način da dovemo sadržaj svih fleгова u akumulator ili, što je mnogo zgodnije, jednom jedinom PLP instrukcijom postavimo sve flebove u željeno stanje. Preporučujemo vam, ipak, da se avanturama poput ove bavite tek kada savladate osnovne mašinsko programiranja.

Često će vam zahtevati da na steku sadržavate sadržaj svih registra mikroprocesora. Obzirom da na standardnom 6502 ne postoje instrukcije PHX, PHY, PLL i PLY (ovo su uvek obezbedene tek na 6502C), moraćete da se poslužite malim trikom: PHX:PHA; TXA:PHA; TYA:PHA. Vidimo da smo na stek najpre poslali sadržaj status registra i sadržaj akumulatora a da smo zatim X i Y registar silali u akumulator da bismo ih onda silali na stek. Vraćanje registra u početno stanje postizemo sekvencom PLTA:YAX:PLA:TXA:PLP; obratite pažnju na redosled!

Šta se dešava ako se u nekoj petlji naše više instrukcija PUSH nego PULL? Stek će se postepeno puniti dok se registar S ne dođe do nule. Po sledećem smanjivanju u S će se naći broj &FF i stek će se puniti kružno — podaci se efektivno gube bez mogućnosti povratka! Ovakva će greška izazvati krah programa, ali je verovatno da, po pritisku na RESET, memorija neće biti „izbombardovana“ — prepuñenje steka je unistilo samo sadržaje ćelija &100–&1FF.

Kada u bejziku napišete GOSUB 1000, računac će početi da izvršava potprogram koji počinje od linije 1000. Kada u daljem radu naiđe na RETURN, vratiće se izvršavanju segmenta programa koji se nalazi iz GOSUB 1000. Da bi to uradio on na neki način mora da „zna“ da je do linije 1000 stigao iz jednog potprograma (inače će RETURN izazvati grešku RETURN without GOSUB) i da „zapamti“ gde se nalazi 1000 naredba koja pozvala potprogram. Ovi podaci se zaista i pamte, ali su za prosečnog programera nebitne njihove lokacije.

6502 je opremljen instrukcijom JSR (*Jump to SubRoutine*) koja je potpuni ekvivalent GSUB; razlika je jedino u tome što se iz GOSUB piše broj programske linije a iz JSR memorijska adresa ćeliju u koju je upisana prva instrukcija potprograma (uz korišćenje assemblya će se, jasno, iz JSR nati ime labele). Za povratk u glavni program koristi se RTS, potpuni ekvivalent RETURN.

Da biste znali sve o potprogramima, reći ćemo da nema prepreke da u toku izvršavanja nekog potprograma pozovete novi potprogram, a iz ovoga slediće i tako do prilične dubine. Smeštam bismo ovima zaključujući o potprogramima, znali biste koliko i bejzik programer koji mirno koristi GOSUB i RETURN, ne razmišljajući o tome kako ova struktura funkcioniše. Mašinski programer mora da zna više pa ćemo pokušati da prikažemo delovanje instrukcija JSR i RTS.

Kada naiđe na RTS, 6502 mora da zna gde da se vrati to jest gde se nalazi instrukcija koju treba da izvršava. Obzirom da je svaka instrukcija određena dvojbajtnim adresom, pri nailasku na RTS su potrebna samo dva bajta. Gde bi ta dva bajta mogla da budu smeštena? Pa, jedini ozbiljan kandidat može da bude stek.

JSR ne je otprilike ekvivalentno sa PUSH PC: JP n na RTS sa PULL PC (jasno je da instrukcije PUSH PC i PULL PC pod tim imenima ne postoje). Pri svakom pozivu potprograma na stek, dakle, biva smeštena dva bajta koja, posmatrano kao šesnaestobitni broj, predstavljaju adresu instrukcije na koju će se preći po nailasku na RTS. U principu bismo mogli da napišemo LDA PHA; LDA a zatim RTS — mikroprocesor bi se „vratilo“ na instrukciju koja je upisana u memorijsku ćeliju &3001 (zašto ne &3000? Ne pitajte previše — mali hr konstruktora 6502) bez obzira na činjenicu da prethodno nije izvršeno nikakvo JSR; tako smo, premda to nema mnogo smisla, simulirali JP &3001!

Daleko je problematizacija situacija u kojoj smo u potprogramu koji je pozvan sa JSR nehotice uputili neko PUSH koje nije imalo svoje PULL. Nailasni na RTS, mikroprocesor selo upiza dva bajta sa steka i smešta ih u PC. Ne znači da se ne radi ni o kakvoj povratnoj adresi već o podatku koji ima neki drugi smisao. Izvršavanje programa će se u tom slučaju nastaviti od nekog od nepredviđenog mesta, što može da ima katastrofalne posledice po program u memoriji. Sve u svemu, ako se planira povratk iz potprograma, upotreba steka mora da bude „čista“: koliko podataka stavimo na stek toliko moramo i da uzmemo i tako u svakoj grani potprograma!

Zašto smo rekli „ako se planira povratk“? Kada u bejziku izvršimo GOSUB, moramo da izvršimo i RETURN, inače se (nema nevidljivi) stek za pozive potprograma polako puni pa je jednom biti prijavljena greška tipa „Too many GSUBs“. Ne bi imao nikakvog smisla učiti mašinski jezik ako bi i on imao slična ograničenja: ako u nekoj grani potprograma konstatujemo da se


```

10111101 / 101 = 00100101
-101
-----
0111
-101
-----
1001
-101
-----
100

```

slika 33

praćeno odgovarajućim šiftovanjem kolonika. Ako vam ovaj postupak izgleda prekomplikivano, pogledajte sliku 34 — posle proučavanja primera sve izgleda mnogo jednostavnije. Pošto smo razvili metod, pisanje programa sa slike 35 ne predstavlja naročiti problem — treba se samo podsetiti naredbi INC i DEC koje omogućavaju povećavanje i smanjivanje sadržaja neke memorijske ćelije za 1.

Korak	Akumulator	Deljenik	Količnik
Start	00000000	00010111	00000000
Shift 1	00000000 <== 00101110 Ne može se odzuzeti 101	00011100	00000000
Shift 2	00000000 <== 01011100 Ne može se odzuzeti 101	00000000	00000000
Shift 3	00000000 <== 10111000 Ne može se odzuzeti 101	00000000	00000000
Shift 4	00000000 <== 01110000 Ne može se odzuzeti 101	00000000	00000000
Shift 5	00000010 <== 11100000 Ne može se odzuzeti 101	00000000	00000000
Shift 6	00000101 <== 11000000 Može da se odzuzne 101	00000000	00000000
Odzuzimanje	00000000 11000000	00000001 (1 dodato količ.)	
Shift 7	00000001 <== 10000000 Ne može se odzuzeti 101	00000010	00000010
Shift 8	00000010 <== 00000000 Ne može se odzuzeti 101	00000100	

slika 34 1011 / 101 = 100 (ostatak 10)

```

3000      .deli
3000 A9 00  LDA #0
3002 A2 08  LDX #8      ; Brojac bitova = 8.
3004 38      SEC        ; Carry se setuje pre prvog LSB oduz.
3005      .sledeci
3005 0E 1B 30  STL deljenik
3008 2A      ROL A      ; Rotiranje da se "uhvati" prenos.
3009 CD 1C 30  CMP delilac
300C      .dodaj
300C 30 06  BCC dalje
300E ED 1C 30  SBC delilac ; Oduzmi delilac ...
3011 EE 1B 30  INC deljenik ; ... i dodaj 1 koloniku.
3014      .dalje
3014 CA      DEI        ; Umanji brojac ...
3015 D0 EE  BNE sledeci ; ... i ponovi operaciju 8 puta.
3017 89 1B 30  STA ostatak ; Ostatak u memoriji.
301A 60      RTS        ; Natrag u bajzik.
301B 00      .deljenik EQUB 0 ; Mem. ćelija koja čuva deljenik.
301C      .dodaj
301C 00      .delilac EQUB 0 ; Mem. ćelija koja čuva delilac.
301D 00      .ostatak EQUB 0 ; Mem. ćelija koja čuva ostatak.
10 PRINT
20 INPUT "Prvi broj " A
30 INPUT "Drugi broj " B
40 POKE deljenik,A
50 POKE delilac,B
60 CALL deli
70 PRINT "Rezultat: "; PEEK(deljenik)
80 PRINT "Ostatak: "; PEEK(ostatak)
90 GOTO 10

```

slika 35

dovedi broj 420 u registar A. Da sada budemo malo egzaktniji: u mnemonici 6502 naredba LOAD se piše kao LDM B i znači napuni (LOAD=napuni) registar m sa sadržajem B.

Maini slovom m smo označili bilo koji od registra A, X ili Y, što znači da postoje instrukcije LDA, LDX i LDY koje omogućavaju prenos podataka iz memorije u registra.

Potreban nam je, jasno, i način da prenosimo podatke iz registra u memoriju; za to se koriste naredbe STM B (STORE=upiši), gde je sa m označen registar (A, X ili Y), a sa B adresirane podatka. Tako će STA 420000 smestiti sadržaj akumulatora u memorijsku ćeliju čija je adresa 420000, dok će STX 4613C prepisati sadržaj registra X u ćeliju 4613C.

Obzirom da posedujemo samo tri registra, konstruktorima 6502 nije bilo teško da obezbede instrukcije za prenos podataka između njih. Opšti oblik ovih instrukcija je Tmn, gde je T skraćeni od transfer (=prenesi) dok su m i n oznake registra između kojih se vrši prenos. Podaci putuju iz registra m u registar n što znači da će TAX prenети sadržaj akumulatora u indeks registar X, dok će TXS prenети sadržaj registra X u registar S (pokazivač steka). Nije, na žalost, moguće prenети podatke između bilo koja dva registra pa će vaš assembler prijaviti grešku ako upotrebite naredbu TXY kojom biste preneli broj iz registra X u registar Y. Kako da zapamtite koje su instrukcije prenosa podataka dopuštene? Najbolje je da pogledate sliku 7 na kojoj je prikazan opšti tok podataka u 6502. Moguće je prenети podatke jedino iz susednih registra S i X, X i A, A i Y; sve su druge kombinacije nedozvoljene!

slika 7

Govoreći o LOAD i STORE instrukcijama, slovom B smo označili određite podatka izbegavajući da se njime ozbiljnije bavimo. Došlo je, međutim, vreme da naučimo na koje se sve načine mikroprocesor 6502 može opisati mesto na kome će pronaći ili smestiti neki podatak; upoznaćemo, dakle, *modove adresiranja*. Padajući čitaoci ovog umetka će se doseliti da smo o adresiranju ne može da se upotrebi kod svih naredbi: šta bi značilo STA 451000 kada se u konstantu 451000 ne može smestiti nikakav broj (konstanta se tako zove zato što je nepremijiv)? Konstruktori 6502 su se, međutim, potrudili da obezbede neposredno adresiranje kada god ono ima smisla, što znači da ćemo konstante moći da koristimo i kod sabiranja, oduzimanja, logičkih operacija i na mnogim drugim mestima.

Apsolutno adresiranje takođe nije teško razumeti — naredba LDA 420000 će u akumulator dovesti sadržaj ćelije 420000, dok će STA 421000 smestiti ovaj broj u ćeliju 421000. Dalji komentar teško da je potreban.

Obzirom da još nismo upoznali instrukcije skoka, ne možemo da napišemo program koji će prenети sadržaje memorijskih ćelija 42000, 42001, ..., 420FF u ćelije 43000, 43001, ..., 430FF. Pokušajmo, ipak, da razmislimo kako bismo napisali takav program, pa ćemo videti da je tako nešto skoro nemoguće — potrebu su nam instrukcije koje će, pri svakom prolazu kroz petlju, prenosi podatke između **različitel** memorijskih ćelija i akumulatora. Tako u igru ulaze indeksi registri X i Y.

Pogledajmo instrukciju STA 42000, X, primer *indeksiranog adresiranja*. U početku će, na primer, X biti jednako 0 pa će mikroprocesor upisati 810 u ćeliju čija je adresa 42000+0=42000. U sledjećem će proslazu X postati 1, pa će 810 bitu upisano u ćeliju broj 42001. Posao će se ponavljati sve dok X ne bude imalo vrednost 4FF kada će se pristupiti ćeliji 420FF. Šta će da se desi kada mikroprocesor sledeci put uveća vrednost X za jedan? Obzirom da je X osmobitni registar, 4FF je najveći broj koji može da se upiše u njega, pa će sledeci INX (*increment X* ili, u slobodnom prevodu, *povećaj X za jedan*) dati 00 umesto 8100, što znači da će se upisivanje kružno ponavljati.

To je izbegnuto primenom BNE (*Branch if not equal*, tj. *skoči ako je različito*) instrukcije u programu sa slike 8: ciklus će se ponavljati sve dok je sadržaj registra X različit od nule.

Bez obzira na to što nam program sa slike 8 ne može biti do kraja jasan pre nego što upoznamo aritmetiku, skokove i rad sa negovima, trebalo bi da primetimo problem koji se ne bi mogao rešiti

Potprogrami i stek

Stek je, rekli smo, područje memorije koje se koristi za privremeno odlaganje podataka. Na stek, primenom instrukcije PUSH, možemo da smestimo akumulator ili P registar (P registar,

```

LDX #0      : X pokazuje ćeliju strane 420
             u koju treba upisati podatak.
LDA #10     : Upisuje se broj 10.
.upisivanje
STA &2000,X
INX
BNE upisivanje : X pokazuje sledeću ćeliju.
             : Kada X ponovo bude 0, došli
             : smo do početne ćelije.
RTS        : Povratak u bežik.

```

slika 8

primenom indeksiranog adresiranja: šta da smo želeli da premestimo blok veći od 256 bajta? Obzirom da se u indeks registar može upisati samo osmoiniti broj i da je bazna adresa (ono &2000 u STA &2000, X) fikсна, indeksirano adresiranje nam ne može pomoći da realizujemo rad sa blokovima u kojima se nalazi više od 256 bajta podataka?

Upoznajmo, da bismo rešili problem, *indirektno postindeksirano adresiranje* koje ćemo proučiti na primeru instrukcije LDA (&20), Y. Pošto se radi o relativno komplikovanom adresiranju, najpre ćemo upoznati obično *indirektno adresiranje* koje bi, kad bi na 6502 postojalo, ilustrirala instrukcija LDA (&20) (ova instrukcija, da to naglasimo još jednom, ne postoji na standardnom 6502, ali će nam pomoći da shvatimo postojeće LDA (&20), Y).

Kod apsolutnog adresiranja se uz svaku naredbu navodi adresa memorijske ćelije u kojoj se nalazi podatak kome treba pristupiti: LDA &20 bi u akumulator doveo broj iz memorijske ćelije čija je adresa &20. LDA (&20) je bitno različita instrukcija: u njoj nije u akumulator treba dovesti podatak čija je adresa smeštena u ćeliji &20. Mikroprocesor će, dakle, najpre morati da pristupi ćeliji &20, da pročita iz nje adresu a da zatim pristupi ćeliji čiju je adresu upravo dobio i u akumulator smesti njen sadržaj. Memorijska ćelija &20 je, jasno, nedovoljna da referencira bilo koju od 65536 ćelija pa će LDA (&20) podrazumevati da se adresa sastoji od brojeva u ćelijama &20 i &21. Možda je sve ovo lakše razumeti na primeru: neka je u ćeliji &20 broj 0 au &21 broj &41; tada će LDA (&20) u akumulator dovesti sadržaj memorijske ćelije čija je adresa &4100.

Ako smo razumeli princip delovanja instrukcije LDA (&20), neće nam biti teško da razumete primenu instrukcije LDA (&20), Y: umesto da se zadovolji adresom koju je pročitao u ćelijama &20 i &21, mikroprocesor će na nju dodati sadržaj registra Y. Da se vratimo ranijem primeru: sekvenca sa slike 9:

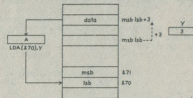
```

LDA #0
STA &20    : dovodjenje broja 00 u ćeliju &20
LDA #&41
STA &21    : dovodjenje broja &41 u ćeliju &21
LDY #&10   : dovodjenje broja &10 u registar Y
LDA (&20),Y

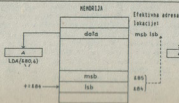
```

slika 9

će dovesti sadržaj memorijske ćelije &4110 (&4100 + &10) u akumulator. Princip delovanja *indirektnog postindeksiranog adresiranja* je dat na slici 10.

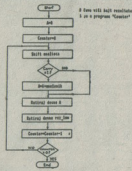


slika 10



Imitiranje manualnog množenja može da rezultuje blok dijagramom sa slike 30 — po izvršavanju programa koji je određen ovim dijagramom dobili bismo proizvod MPD i MPR u ćeliji PP. Primatećemo, međutim, i određene probleme: množenjem dva osmoiniti broja možemo da dobijemo broj koji ima 16 bita i koji, prema tome, ne može da stane u jednu memorijsku ćeliju; algoritam sa slike 30 će u tom slučaju dati pogrešan rezultat, a neće signalizirati grešku. Jednostavnim ispitivanjem bismo bez mnogo problema detektovali grešku, ali su nam spetiti nešto veći — hoćemo da sastavimo program koji množi osmoiniti brojeve i daje šesnaestoiniti rezultat!

Do rešenja dovodi sarno jedna dobra ideja: umesto da šiftovanje množenik, šiftovaćemo parcijalni proizvod i to kroz dve memorijske ćelije, posredstvom carry flega. Na slici 31 vidimo blok dijagram, a na slici 32 program koji obavlja korektno množenje osmoiniti brojeva. Ako uspete da pročitate ovaj umetak do kraja, upoznatećete metod za množenje šesnaestoiniti binarnih vrednosti.



slika 31

slika 32

```

3000          .mnozi
3000 A9 00   LDA #9          : MSB odgovora je 0.
3001 02 08   LDX #8         : Brojac bitova = 8.
3002          .sledeci
3004 E 19 30 LSR mnozilac
3007 90 04   BCC rotiraj    : Sledeci bit mnozioca je 0.
3009 18      CLC            : Bit je jedinica ...
300A 6D 18 30 ADC mnozenik  : ... mnozenik se dodaje delimicnom zbiru.
300B          .rotiraj
300D 6A      ROR A          : Najinzi bit HIGH bajta ide u LOW bajt.
300E 6E 1A 30 ROR rezultat_low
3011 CA      DEX            : Umanjuje se brojac
3012 D0 F0   BNE sledeci    : Da li je mnozeno sa svih 8 bitova?
3014 8D 1B 30 STA rezultat_high : Ako jeste, treba pripremiti rezultat...
3017 60      RTS            : ... i vratiti se u bežik.
3018 00      .mnozenik      EQU 0 : Mem. ćelija koja cuva mnozenik.
3019 00      .mnozilac      EQU 0 : Mem. ćelija koja cuva mnozilac.
301A 00      .rezultat_low  EQU 0 : Mem. ćelija koja cuva nizi bajt proizv.
301B 00      .rezultat_high EQU 0 : Mem. ćelija koja cuva visi bajt proizv.

```

```

10 INPUT "Prvi broj " A
20 INPUT "Drugi broj " B
30 POKE mnozenik,A
40 POKE mnozilac,B
50 CALL mnozi
60 PRINT 256*PEEK(rezultat_high) + PEEK(rezultat_low)
70 GOTO 10

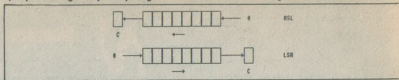
```

... i deljenje

Dok se binarno sabiranje svodi na sukcesivno dodavanje, deljenje obavljamo višestrukim oduzimanjem delioca od deljenika, kao na slici 33 na kojoj smo delili brojeve 189 i 5 (%10111001 i 501) i dobili količnik 35 (%100101) i ostatak 4 (%100).

Kao što možete da pretpostavite, program koji bismo napravili simulirajući „ručno“ računanje ne bi bio naročito dobar. Umesto toga, šiftovaćemo sukcesivne bitove deljenika u akumulatoru sve dok ne budemo mogli da mu oduzmemo delioci. Tada ćemo dodati i količniku i nastaviti sa radom, sve dok ne budu šiftovani svi bitovi deljenika. Svako šiftovanje deljenika mora da bude

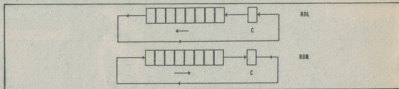
poslednji bit „preliva“ u carry indikator. Rotiranje je unekoliko slično, a tim što se na kraj ćelije ne upisuje nula nego ranije stanje flega C. Da vidimo kako sve to izgleda na primerima.



slika 27

Logičko šiftovanje nalevo i nadesno je prikazano na slici 27. Vidimo da sa ASL t odnosno LSR t pomerno bitove odrediti t nalevo odnosno nadesno za jedno mesto; umesto nultog (za ASL) odnosno sedmog bita (kod LSR), u odredite ulazi nula dok bit koji „ispadne“ usled pomeranja odlazi u carry fleg. Raniji sadržaj carry flega se, jednostavno, gubi.

Vidimo da smo primenom instrukcija ASL odnosno LSR izgubili bivše stanje carry flega, a u odredite upisali jednu nulu. Zašto da ne izbegnemo gubitak tog bita i veštačko generiranje nule preplavši prethodno CY na kraj odnosno početak odredista? Tako dobijamo operacije ROL t i ROR t: rotaciju odredista t i carry flega za jedno mesto u levo odnosno u desno. Obe ove operacije su prikazane na slici 28.

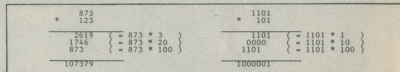


slika 28

Množenje . . .

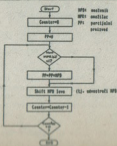
Mikroprocesor 6502, poput većine osamobitne sabrača, nije opremljen instrukcijama za množenje i deljenje brojeva, iako su ove operacije u praktičnom radu neobično potrebne. Ostalo je da korisnici dopune konstruktore: znanje koje smo stekli je dovoljno da napišemo programe za množenje i deljenje binarnih brojeva.

Kako da pristupimo problemu? Pošto su nam na raspolaganju sabiranje, oduzimanje, šiftovanje i rotiranje, pokušaćemo da svvedemo množenje na ove četiri operacije. Na slici 29 vidimo kako se množe obični dekadni brojevi, a zatim i primer množenja binarnih brojeva koji će nam biti daleko značajniji.



slika 29

slika 30



Potpunosti radi, na slici 11 dajemo princip delovanja *indirektnog preindeksiranog adresiranja* savetujući vam da priču o njemu preskočite i ostavite za drugi vremena: ovo se adresiranje retko koristi pa će vas nepotrebno zbunjivati.

Indirektno preindeksirano adresiranje ćemo upoznati na primeru instrukcije LDA (620, X). Neka se u X registru nalazio broj 610. Mikroprocesor će najpre ovaj broj sabrati sa 620 i tako dobiti 630. Zatim će pristupiti ćelijama 630 i 631 i tako pronaći adresni bit nekog adresa; ako se u ćeliji 630 nalazio broj nula a u 631 broj 611, pristupaće se memorijskoj ćeliji čija je adresa 611000 i njen će sadržaj biti prenet u akumulator. Indirektno preindeksirano adresiranje se obično koristi za rad sa tabelama podataka koji su rasuti po memoriji, ali su njihove lokacije upisane u sukcesivne memorijske ćelije.

Pošto smo opisali skoro sve modove adresiranja pristupačne na 6502, posvetićemo se njihovim ograničenjima. Sve instrukcije koje smo u ovom poglavlju pominjali su postojeće i mogli biste da ih koristite u svojim programima. Ukoliko biste se, međutim, odvažili da napišete LDA (620), Y, vaš bi assembler sa zadovoljstvom prijavio grešku. Ovakve greške kada ste napisali savršeno korektnu instrukciju koja koristi indirektno postindeksirano adresiranje? Jednostavno, konstruktorsko rešenje je odgovorilo da kod pre i post indeksiranog adresiranja adresu smeštaju isključivo u memorijske ćelije nulte strane!

Nulta strana (*zero page*) obuhvata memorijske ćelije 0-255 (0-8F) koje su, kao što rekoamo, kod 6502 veoma privilegovane (svuda te privilegije). Kako, zaboga, neke memorijske ćelije mogu da budu privilegovane? Asemblijarom instrukcije LDA (200, n) i LDA (820, n): prva će biti kodirana kao 8AD 800 620 (zauzima tri bajta) a druga kao 6A5 620 (samo dva bajta) kako je jedan bajt ušteden? Konstruktori 6502 su imali dosta slobodnog mesta u setu instrukcija pa su odlučili da istoj naredbi (LDA) dodeli jedan kod ukoliko se argument nalazi na nultoj strani, a drugi kod ukoliko je on na proizvoljnom mestu u memoriji. Ukoliko autor ROM-a nekog računara na nultu stranu smešta podatke koji će se najčešće koristiti, uštedeće dosta prostora i dobiti u brzini rada kompjutera!

Daleko je važnija druga privilegija memorijskih ćelija nulte strane koju smo već pomenuli: u okviru pre i post indeksiranog adresiranja **memorijske ćelije koje sadrže adresu moraju da se nalaze na nultoj strani**. To znači da ćete, čim poželite da koristite indirektno adresiranje, morati da konsultujete uputstvo za upotrebu vašeg kompjutera i saznate koje su ćelije nulte strane ostavljene korisnicima; ukoliko se umešate u prostor bezik interesa, moguće su razne neprijatne posledice!

Dalje ograničenje je obaveza korišćenja registra X kod preindeksiranog i registra Y kod postindeksiranog. To praktično znači da se smeta da napišete LDA (620, Y) odnosno LDA (620, X); morate da koristite LDA (620, X) i LDA (820, Y).

Dok je za prethodna ograničenja teško naći neko racionalno objašnjenje (što ne znači da to ne postoji), prilično je lako razumeti da ne postoje instrukcije LDX (620, X), LDY (620, Y), LDX (2000, X) i slične: ukoliko je za adresiranje potreban registar X, rezultat se nikako ne može istovremeno smestiti u njega (hardverši će reći da takozvani master-slave flip-flopovi omogućavaju baš takve trikove, ali je to ovde sasvim nebitno).

Na kraju umetka dajemo spisak instrukcija mikroprocesora 6502 i, u okviru njega, adrese modovi koje svaka od naredbi dozvoljava; nije loše da pogledate tabelu i pokušate da ilustrujete ne ono o čemu smo govorili u ovom poglavlju. Jer, ovo je najteže ali i najvažnije poglavlje naše škole programiranja na 6502: ukoliko ste ga dobro razumeli, ostatak ćete pročitati i primeniti sa lakoćom! Ukoliko vam nešto nije jasno, ne očajavajte: pročitajte još jednom prethodne stranice, a onda jednostavno nastavite sa radom — kada vam god bude potrebno neko „egzotičnije“ adresiranje, lako ćete pronaći i u praksi razumeti njegov opis!

Aritmetičke operacije

U uvodnom poglavlju smo upoznali operacije sa binarnim brojevima koje 6502 može da izvrši: sabiranje i oduzimanje. Rekli bi se da o ovim operacijama ne treba mnogo govoriti: zar ima nekoga jednostavnijega od sabiranja? Pa ipak, činjenica da 6502 operacije isključivo sa osmobičnim podacima nas nateruje da upoznamo trikove koji će nam omogućiti da operišemo sa brojevima većim od 255 ili 127.

Naredba za sabiranje se u memoriji 6502 označava sa ADC B pde je sa B označena lokacija broja koji treba dodati sadržaju akumulatora; naredbe LDA #620: ADC #630 bi, na primer, trebale da dovedu broj 620+630=650 u akumulator. Nije, međutim, sve baš tako jednostavno: skraćena ADC je izvedena od *ADD with Carry* što znači da se pri sabiranju rezultatu dodaje i stanje Carry flega. Potrebno je, dakle, da najpre razumemo kako se to dodavanje izvodi pa da onda odgovorimo na pitanje **zašto** je ovo potrebno. Pogledajmo dva jednostavna primera sa slike 12:

CLC	SEC
LDA #620	LDA #620
ADC #630	ADC #630

slika 12

Instrukcija LDA u oba programa treba da dovedu konstantu 620 u akumulator posle čega sledi dodavanje broja 630 sadržaju ovoga registra (rezultat se, kao što verovatno pretpostavljate, ponovo smešta u akumulator). Pitanje je, naravno, šta će se po izvršavanju programa dobiti u

akumulatoru. U akumulator će uvek stići broj &20+&30+C, gde smo sa C označili stavine carry flega. Kakvo je to stanje? Obratimo pažnju na prvu naredbu drugog programa SEC: ova reč je skraćena od *Set Carry flag*, što bi moglo da se prevede i kao *stavi 1 u C (carry)*. C će, dakle, biti 1, pa će izvršavanjem drugog programa u akumulatoru biti &20+&30+1= &51! U prvom programu smo sa CLC (*Clear Carry*, odnosno *obriši carry*) stavili nulu u C indikator, što znači da će se, po izvršavanju programa, u akumulatoru naći broj &20+&30-0=&50!

Šta bi se desilo da smo izostavili naredbe SEC odnosno CLC? Reklo bi se da je tada stanje carry flega nedefinisano, ali ono jednostavno ne može da bude takvo: fleg može da bude ili setovan ili resetovan; trećega nema! Zato će izvršavanjem prethodnog dela programa (ili radom nekog interpretera ako je naš mašinski program upravo pozvan) biti formirana vrednost C, pa će naredba ADC raditi korektno iako možda ne onako kako smo želeli. Zapamtite da, ako planirate korišćenje naredbi ADC i SBC (subtract, oduzimanje), **uvek** setujete carry primenom SEC ili ga resetujete naredbom CLC.

Videli smo kako se izvršava naredba ADC, ali nam je i dalje nejasno kakav je smisao kvarenja lepo dobijenog zbira dodavanjem carry flega. Zamislimo da treba da saberemo dva šesnaestobitna broja od kojih se jedan nalazi u memorijskim ćelijama &2000 i &2001, a drugi u &2002 i &2003. Zbir treba da smestimo u &2004 i &200B. Kako da rešimo problem? Pokušajmo da, najpre, saberemo zadate memorijskih ćelija &2000 i &2002 (manje signifikantne bajtove ili, kako se to kaže, LSB (*low significant byte*) oba broja) i smestimo rezultat u &200A, a da zatim saberemo sadržaje &2001 i &2003 (MSB odnosno most *significant byte*) i rezultat smestimo u &200B. Program koji to radi je dat na slici 13.

```

1 LDA &2000 ; LSB prvog broja u akumulator.
2 CLC
3 ADC &2002 ; sabiranje sa LSB drugog broja.
4 STA &200A ; LSB rezultata u ćeliju &200A.
5 LDA &2001 ; MSB prvog broja u akumulator.
6 CLC
7 ADC &2003 ; sabiranje sa MSB drugog broja.
8 STA &200B ; MSB rezultata u ćeliju &200B.

```

slika 13

Iprobajmo ovaj program stavljajući u &2000 i &2001 broj &2030 (POKE &2000, &30: POKE &2001, &20), a u &2002 i &2003 broj &4050. Posle startovanja programa dobićemo da se u ćeliji &200A nalazi broj &80 (iako iskoristite PRINT PEEK (&200A), dobićete 128 što ne treba da vas iznenadi: 128 = &80) a u ćeliji &200B broj &60 što znači da je zbir &60&80, program, dakle, radi ispravno. No, šta ako u ćelije &2000, &2001, &2002 i &2003 upišemo respektivno brojeve &20, &3A, &F0 i &107? Po startovanju programa će se dobiti zbir &4A10, dok je &3A20+&310F=&4B10. Program je, dakle, pogrešan!

Kako je nastupila greška? &502 je najpre sabrao brojeve &20 i &F0 i dobio, ili bar trebao da dobije, &110. Broj &110 je, međutim, veći od &FF, pa se pojavio bit koji nije mogao da stane u akumulator. Onako kako smo ga napisali, program je zanemario taj bit, pa je sabrao brojeve &3A i &10 i dobio &4A; da je dodao prenos, dobio bi &4B što bi bio tačan rezultat.

Da je &502 nedostavno zanemario bit prekoračenja, problem sabiranja šesnaestobitnih brojeva bi bio veoma teško rešiti. Na svu sreću, konstruktori mikroprocesora su se pobrinuli da se prekoračenje prenese u Carry indikator, koji na neki način predstavlja deveti bit akumulatora. Da vidimo kako sve to izgleda na primeru sa slici 14.

```

                C
&20 =          00100000
&F0 =          11110000
ADC =          1 00010000

```

slika 14

Shvatite činjenice da se deveti bit rezultata seli u Carry indikator će nam pomoći da brzo i lako rešimo problem sabiranja višebajtnih brojeva. Dovoljno je da izostavimo instrukciju broj 6 iz programa koji smo sastavili, pa će prenos jednog sabiranja biti pridodat rezultatu drugoga. Kako se prenos na engleskom zove carry, jasno vam je po čemu je C fleg dobio ime — posredstvom njega se prekoračenje u jednom sabiranju dodaje na rezultat drugog sabiranja kako bi se formirao korektan višebajtni rezultat.

Naredba ADC, osim na C, utiče na Z, V i N flegove: ukoliko je, na primer, rezultat sabiranja 0, biće setovan Z(fleg). Fleg tako može da se dobije nula sabiranjem dva pozitivna broja? Pokušajte, na primer, da saberete &F0 i &10 i dobićete &100, kako u akumulatoru može da stane svega osam bita rezultata, u njega će biti upisano 00 i setovanje zero (i carry) fleg. V fleg signalizira da je rezultat u akumulatoru nekorektan ako se argumenti posmatraju kao označeni brojevi. Napravimo, dakle, mali rezime: pri sabiranju neoznačenih brojeva (0-255), setovan carry fleg označava prekoračenje i nekorektan rezultat, dok se V fleg obično zanemaruje. Kod sabiranja označenih brojeva (-128 do +127), prekoračenje se signalizira setovanom overflow flegom pri čemu se stanje Carry flega obično zanemaruje.

Pretpostavimo da se u ćeliji &2000 nalazio broj &9A, odnosno %10011010. On je najpre doveden u akumulator a zatim je na njega i fleg %01111111 primenjena operacija „i“. Na slici 23 vidimo šta se dobija:

```

10011010
AND 01111111
-----
00011010

```

slika 23

Vidimo da se rezultat razlikuje od početnog podatka &9A samo u poslednjem bitu koji je, podle primene AND operacije, postao nula. Ovakav rezultat se mogao i očekivati: broj %01111111 se sastoji od samih jedinica od kojih svaka, kada je logički pomnožimo (operacija „i“ se zove i logičko množenje) sa bilo čim drugim daje kao rezultat to „nešto drugo“. Jedino vodeće nula, kada je logički pomnožimo sa bilo čim, daje kao rezultat opet logičku nulu!

Šta bismo učinili da je trebalo da **setujemo** sedmi bit ćelije &2000? Upotrebili bismo naredbu ORA kao na slici 24.

```

%00          : Dovođenje sadržaja ćelije &2000 u akumulator.
00060000    : Setovanje sedmog bita.
%00          : Smeštanje rezultata u memoriju.

```

slika 24

Uzevši da se pre izvršavanja programa u ćeliji &2000 nalazio broj &73 odnosno %01110011, novi sadržaj je, prema slici 25, jednak &F3 — sedmi je bit očit setovan.

```

01111011
OR  10000000
-----
11110111

```

slika 25

Da li smo mogli da testiramo neki bit memorije a da ga ne promenimo? Naravno da smo mogli — trebalo je samo primeniti program sa slike 26. Vidimo da smo najpre prepisali broj iz ćelije &2000 u akumulator, a zatim ga logički pomnožili sa %00000010. Ukoliko je prvi bit (sećate se da se broj od nultog?) akumulatora jedinica, rezultat će biti %00000010; ako je taj bit bio nula, u akumulatoru će se naći nula. Znaajući da naredba AND deluje na Z(fleg) fleg koji će biti setovan ako se, po izvršenju operacije, u akumulatoru pojavi nula, naredba *BEQ nije* će nas odvesti do segmenta programa koji je označen sa „nije“ samo ako prvi bit memorijske ćelije &2000 nije setovan.

```

LDA &2000          ; Ćelija &2000 u akumulator.
AND %00000010    ; Testiranje prvog bita.
BEQ resetovan    ; Da li je 0?
.setovan

```

... : Naredbe koje se izvršavaju ako je setovan.

```
RTS
.resetovan

```

... : Naredbe koje se izvršavaju ako je resetovan.

```
RTS
```

slika 26

Konstruktori &502 su se pobrinuli za jednu specijalnu instrukciju koja nam pomaže da istovremeno testiramo šesti i sedmi bit neke memorijske ćelije: to je naredba BIT. Koja se, uglavnom, koristi za testiranje zauzetosti neke periferijske jedinice. Ukoliko, na primer, izvršimo BIT &2000, akumulator će biti upisovan sa ćelijom &2000 (kao da smo iskoristili AND), ali se sadržaj akumulatora neće promeniti! Da bi stvar bila još lepša, sedmi bit memorijske ćelije &2000 će biti prenesen u fleg N, a šesti bit u fleg V.

Ekvivalentnom se disjunkcijom u ovom momentu nećemo previše baviti. Njena zanimljiva osobina je da je sama sebi inverzna: ako izračunate (*X EOR Y*) EOR Y gde su X i Y bilo koji brojevi, dobićete opet X. Ukoliko, dakle, sve memorijske ćelije u nekom segmentu ekvivalentno pomnožimo sa nekim brojem, dobićemo naokolo besmislen sadržaj koji, ponajvažnije, ova operacija, vraćamo na normalu. Mašinska naredba EOR je, dakle, čest gost raznih zaštićenih programa.

Šiftovanje i rotiranje

U uvodnom smo poglavlju ukratko objasnili pojam šiftovanja: sadržaj akumulatora ili neke ćelije se pomera za jedan bit ulivo ili udesno, pri čemu na neki od krajeva ćelije dolazi 0, dok se

instrukcije grananja, za razliku od JMP instrukcija, zauzimaju po dva bajta memorije). Mikroprocesor je, najprije, iz memorije uzeo kod instrukcije BEQ (#F0) i prepoznao ga, a zatim pročitao adresu #20. U tom trenutku registar PC pokazuje na memorijsku ćeliju #1002 u kojoj je upisana naredba koja će biti izvršena ako Z flag nije setovan. Ako je flag setovan, mikroprocesor će sabrati sadržaj registra PC (#1002) i broj #20, pa će tako dobijenu vrednost staviti u PC; izvršavanje programa će, tako, biti nastavljeno od instrukcije čija je adresa #1020.

Šta je dobijeno a šta izgubljeno ovakvim adresiranjem? Pre svega, program u kome koristimo isključivo relativne adresiranje će korektno raditi ma u koji ga segment memorije učitali; BEQ PC+820 izaziva preokretanje sledećih 820 bajtova, bez obzira gde je ova instrukcija smeštena. Treba, međutim, da kažemo da je na 6502 skroz nemoguće a i sasvim nepotrebno pisati programe koji korektno rade bez obzira na svoju lokaciju u memoriji — daleko je lakše izmeniti ORG i assemblirati program na drugo mesto. Relativni skokovi se zato kod većine mikroprocesora koriste isključivo radi uštede memorijskog prostora kod kratkih skokova koji su inače daleko češći od dugih: JMP #2020 zauzima tri, a BEQ PC+820 samo dva bajta. Za šta je onda nekome potrebno apsolutno adresiranje? Korišćenjem jednog jedinog bajta za „pamćenje“ rastojanja skokovi su ograničeni na 127 bajta (broj iza „branch“ instrukcije se posmatra kao označen što znači da su mogući i skokovi unazad), dok iza JMP možemo da navedemo bilo koju adresu u čitavoj memorijskoj mapi.

Konstruktori 6502 su, na žalost, učinili da zamena instrukcije JMP sa „branch“ bude problematičnija: nije obezbeđen bezuslovni relativni skok — naredba BRA (*Branch Always*; pomalo smešno zvuči ali — to je mnemonika) se pojavila tek kod docnijih verzija mikroprocesora 65C02. U većini se slučajeva, na sreću, može računati sa stanjem nekog od flagova, pa ćemo tako „JMP kraj“ zamieniti sa „BNE kraj“ ako smo sigurni da Z flag ne može da bude setovan. Ovakve uštede, međutim, ne preporučujemo početnicima: i iskusen će programer ponekad pogrešiti. Ako se već odvažite na slične avanture, **obavezno** propratite lažne BRANCH instrukcije nekim komentarom — kada doncije budete gledali program, svakako ćete se pitati „kakvo je ovo testiranje“?

Većina assemblera tretira relativne skokove kao i apsolutne: pisateći BEQ DALJE baš kao što biste pisali i JMP DALJE. Sam assembler će pronaći labelu DALJE i izračunati njeno rastojanje od lokacije instrukcije BEQ formirajući tako korektan adresni deo ove naredbe.

6502 omogućava i indirektno apsolutne skokove: KMP (#2000) će, na primer, izazvati skok na instrukciju čija je adresa upisana u ćelije #2000 i #2001. Ukoliko su, na primer, u ćelije #2000 i #2001 upisani brojevi #1A i #F0, JMP (#2000) će izazvati skok na adresu #F01A.

logičke operacije

Došlo je vreme da se pozabavimo logičkim operacijama AND, ORA i EOR. To su binarne operacije koje imaju po dva argumenta i koje se koriste neobično često, dovoljno da im posvetimo posebnu pažnju.

Jedan od argumenata bilo koje od ovih operacija se nalazi u akumulatoru u koji se smešta i rezultat. Drugi može da se nalazi u okviru same instrukcije (AND #n, neposredno adresiranje, npr. AND #8F0) ili u memoriji (npr. AND #3000 ili AND (830), Y).

Nije teško pogoditi šta rade naredbe AND i ORA: sadržaj akumulatora se, bit po bit, poredi sa n, pa se formira rezultat prema tabelama sa slike 5. EOR s radi nešto sasvim slično, pri čemu se nad bitovima izvodi ekskluzivna disjunkcija koja je definisana tabelom sa slike 21. Ekskluzivna disjunkcija, kao što joj i ime kaže, liči na disjunkciju (ORA), ali se od nje malo i razlikuje: 1 ORA 1 = 1 ali je EOR 1 = 0!

X	Y	1 xor 1
0	0	0
0	1	1
1	0	1
1	1	0

slika 21

Za šta može da se iskoristi koja od ovih operacija? Ponekad će nam biti potrebno da promenio ili testiramo samo jedan bit neke ćelije bez promene njenog kompletnog sadržaja. Tu ćemo ćeliju najpre prepisati u akumulator, a zatim ćemo sa AND resetovati a sa OR setovati željeni bit. Da vidimo kako.

U memorijskoj ćeliji čija je adresa #2000 je upisan neki broj i potrebno nam je da se sedmi (poslednji) bit tog broja postavimo na nulu, a da ostale bitove ne promenimo. Uz poznavanje funkcije „I“ napisaćemo jednostavan program sa slike 22.

```

LDA #2000      ; Dovođenje sadržaja ćelije #2000 u akumulator.
AND #01111111 ; Resetovanje sedmog bita.
STA #2000     ; Sačuvanje rezultata u memoriji.

```

slika 22

Poslo smo, na jedvite jade, shvatili kako se koristi naredba ADC, treba da kažemo šta mogu da budu njeni argumenti. Jedan od sabiraka je **uvek** u akumulatoru u koji se smešta i rezultat. Iza ADC navodimo adresu drugog sabirka koji može da bude konstanta (ADC an, npr. ADC #A20) ili adresa neke memorijske ćelije. Adresa, kao i do sada, može da bude apsolutna, indeksirana, indirektna preindeksirana i indirektna postindeksirana, pri čemu instrukcija ADC ima posebne kodove za označavanje skrštenih adresa na nultoj strani. Sumarni prikaz ove instrukcije je, naravno, dat u tabeli instrukcija.

Odizimanje je relativno slično sabiranju: SBC a će od akumulatora oduzeti sadržaj a pri čemu s može da bude broj (SBC an) ili adresa neke memorijske ćelije (SBC #2000, SBC (20), Y, SBC #3000, X ili nešto slično). U akumulator se, primenom ove instrukcije, smešta broj [A-(s+1-C)] što izgleda prilično čudno: dok za sabiranje dva „normalna“ broja [da bismo dobili broj (4+2-jednako 4) treba **resetovati** carry flag (CLC), pre odizimanja dva „normalna“ broja (da bi 4+2 bilo 2) ta treba **setovati** (SEC). U ovom slučaju carry flag označava pozamjicu, pa se zato u literaturi često označava kao *carry/borrow flag*, tj. indikator koji, zavisi od operacije, označava prenos ili pozamjicu. Da bismo oduzeli šesnaestostiti broj u ćelijama #2000 i #2001 od broja u ćelijama #2002 i #2003 upotrebilićemo, dakle, program sa slike 15.

```

LDA #2002
SEC
SBC #2000 ; A=A-(#2000)-(1-C); C=1 ako je manji broj oduzet od većeg
STA #200A
LDA #2003
SBC #2001
STA #200B

```

slika 15

Kao i kod sabiranja, Z flag biva setovan ako je rezultat u akumulatoru nula, dok se U i setuje ako je nastupio preokretanje pri sabiranju označenih brojeva. (Negativ) flag označava znak rezultata koji, jasno, ima smisla samo ako operišemo sa označenim osobitnim brojevima.

Specijalan slučaj sabiranja i odizimanja su instrukcije INR i DEC. INR odnosićemo na DEm gde m može da bude bilo koji od indeksa registara X odnosno Y. Instrukcija INR (=increment, povećaj) će povećati sadržaj ćelije za jedan dok će, za isto DEm za isto toliko smanjiti. Slično tome, instrukcije INC i DEC omogućavaju povećanje ili smanjenje sadržaja neke memorijske ćelije za jedan, tako da će INC #2000 povećati osobitni broj, u ćeliji #2000 za jedan. Videćemo da su ove naredbe vrlo značajne u petljama, gde je potrebna česta promena sadržaja raznih brojeva. INC, INX, INY, DEX i DEY ne deluju na carry flag, što znači da na ovaj način nije pogodno povećavati brojeve koji zauzimaju više bajta memorije ili da je pri tom povećanju potrebno koristiti Z flag koji se, jasno, setuje ako je rezultat operacije jednak nuli. Zanimljivo je, takođe, da instrukcija koja bi automatski inkrementirala akumulator **ne postoji**, što znači da ćete morati da izvršite CLC: ADC #1 da biste povećali akumulator za jedan.

Bezuslovni i uslovni skokovi

Kada bezik interpreter vašeg računara izvrši neku instrukciju, on automatski prelazi na izvršavanje sledeće i tako dalje — sve dok ne naiđe na neko STOP ili END. Korišćenjem GOTO naredbe se, naravno, može zahtevati da se ovaj tok izvršavanja izmeni, dok se IF ... THEN ... ELSE konstrukcijom postize izvršavanje sledeće naredbe samo ako su neki uslovi ispunjeni. Bez ovakvih naredbi ne bismo mogli da sastavimo čak i najjednostavnije programe, što znači da nešto odgovarajuće mora da bude raspoloživo i na mašinskom jeziku.

Najjednostavnija naredba bezuslovnog skoka je JMP — potpuni ekvivalent GOTO. Iza JMP se nalazi adresa memorijske ćelije u koju je upisana naredba koju slediće treba izvršavati, što znači da JMP #3000 zauzima dva bajta memorije. Iza JMP #3000, jasno, podrazumevamo da se instrukcija na #3000 zaista može izvršiti: kako mnoge instrukcije zauzimaju dva, tri ili čak četiri bajta, može da se dogodi da se u memorijskoj ćeliji #3000 ne nalazi početak neke instrukcije već njen drugi ili treći bajt. Dok bezik interpreter u konfliktnim situacijama prijavjuje grešku mikroprocesor ne poseduje apsolutno nikakav način da prosudi da li se na #3000 nalazi instrukcija koju ste vi planirali za izvršavanje; on će pokušati da izvršava ono što tamo nađe, što će dovesti do nepredvidljivog toka događaja i, moguće, potpunog bikiiranja računara.

Iako 6502 razume jedino instrukcije tipa JMP #3000, za nas je ovakvo korišćenje apsolutnog adresiranja sasvim nepogodno. Pre svega, retko su nam potrebni skokovi na ovako „okrugle“ lokacije — obično programiramo prelazak na neku instrukciju u programu. Pri pisanju tog programa nam je, jasno, poznata njegova početna adresa u memoriji (pišemo je iza ORG), ali nam nije poznata početna adresa svake njegove instrukcije: da bismo je našli, moramo da izračunamo koliko bajta memorije zauzima svaka njoj prethodna instrukcija, pa da tako dobijene brojeve sabereimo i dodamo na vrednost ORG. Takvo računanje i sabiranje je neprijatan posao podložan greškama koji, da stvar bude još gora, moramo da ponavljamo kada god u programu napravimo bilo kakvu izmenu koja će proizući ili skratiti neke njegov segment. Zbog toga svi ideo dobri assembleri omogućavaju rad sa *labelama*.

Reč *labela* („label“) može da se prevede kao „oznaka“ — radi se o nekom simbolu kojim označavamo određeni segment programa. Assembleri obično omogućavaju da ime labela bude bilo

koja reč od 5—6 slova se izuzetkom mnemoničkih skraćenica naredbi, ako iskoristimo labelu JMP računar je neće razlikovati od instrukcije JUMP! Jasno je da ćemo imale lebalne birati tako da asocija na svrhu segmenta programa koji označavamo: česta imena labela su RADNI, RACUN, DALJE, GRESKA, KRAJ, i slično.

```

3000 A9 00 LDA #0
3002 85 70 STA #70
3004 A9 00 LDA #0
3006 85 71 STA #71 ; čelije #70 i #71 predstavljaju broj 0000.
3008 A0 00 LDI #0
300A -ciklus
300A B1 70 LDA (#70),Y ; čelija u akumulator ...
300C 91 70 STA (#70),Y ; ... a onda nazad.
300E C8 INY ; pristupa se sledećoj čeliji iste strane.
300F D0 F9 BNE ciklus ; da li je dostignut kraj strane?
3011 E6 71 INC #71 ; ako jeste, predji na sledeću stranu.
3013 4C A0 30 JMP ciklus ; ponavlja operaciju u beskonačnost.

```

slika 16

Na slici 16 je prikazan program koji koristi naredbu JMP da bi realizovao beskonačnu petlju: ovaj program upisuje u svaku memorijsku čeliju njen prethodni sadržaj i tako u beskonačnost! Labela CIKLUS označava početak petlje, pa je JMP CIKLUS ekvivalentno sa JMP \$3000, s tim što je JMP CIKLUS daleko univerzalniji: jednostavnom promenom direktive ORG ovaj program možemo da asbliramo u bilo koji segment memorije. Ukoliko se odlučite da otkucate i isprobate ovaj program, uzmite u obzir da će on, eventualno, izazvati krah sistema ukoliko vaš računar (što je sasvim uobičajeno) koristi memorijski mapirane periferijske uređaje (više o njima donjije) i da ćete moći da ga prekinete jedino pritiskom na RESET taster.

Labela, osim za skokove, možete da koristite i kao zamenu za brojeve. Na početku programa možete, na primer, da napišete RAMTOP = \$5CB2, pa će donjije LDA RAMTOP biti ekvivalentno sa LDA \$5CB2. U ovom slučaju korišćenje labela pomaže jedino boljoj čitljivosti programa, jer ćete, kada ga donjije budete gledali, zaboraviti da je \$5CB2 adresa sistemске promenljive RAMTOP, pa ćete morati da konsultujete razne tabele da biste razumeli segment programa koji ste sami napisali.

Uslovni skokovi

Instrukcija JUMP izaziva tzv. bezuslovni skok — naredba čija je adresa navedena iz JMP se izvršava u svakom slučaju. U praksi ćemo daleko češće koristiti takozvane uslovne skokove koji su, kao što smo već rekli, omogućeni postojanjem fleгова. Naredba BEQ JEDNAK (*Branch if Equal*) će, na primer, izazvati skok na segment programa označen labelom JEDNAK samo ako je Z⁰ fleг setovan. Ukoliko želimo da se skok dogodi ako je fleг resetovan, koristićemo BNE NEJEDNAK (*Branch if Not Equal*). Slično tome, BCS predstavlja situaciju u kojoj je Carry fleг setovan, a BCC situaciju u kojoj je resetovan, BPL označava pozitivan a BMI negativan znak broja (N fleг resetovan odnosno setovan) dok BVS i BVC označavaju setovan odnosno resetovan overflow fleг. Za početak će nam, kao što vidimo na slici 17, biti dozvoljeni Z i N fleгови: program koji dajemo, naime, u memorijskoj lokaciji REZ upisuje jedinicu ako je broj u čeliji PD pozitivan, nulu ako je jednak nuli, a 255 ukoliko se radi o negativnom broju (mikroprocesor, kada se radi o jednobajtnim rečima, smatra brojeve 0—127 pozitivnim, a 128—255 negativnim uz korišćenje potpunog komplementa⁷).

Uslov	Djstvo na fleгоve		
	Z	C	N
Registar < vrednost	0	0	1*
Registar = vrednost	1	0	0
Registar > vrednost	0	1	0*

* - samo ako se radi sa označenim brojevima

slika 17

Proučimo malo program sa slike 17 i upoznajmo načine na koje su fleгови setuju. Naredbe prenosa podataka u registre, pr svega, postavljaju fleгоve u zavisnosti od prenesenog podatka: LDA \$4000 će, na primer, setovati Z fleг, ako se u čeliji \$4000 nalazila nula a resetovati ga ako je tako pronađeno nešto drugo. Poređenje sa nulom nam je, međutim, ređiko dovoljno: želimo da poredimo dva broja, što radimo uz pomoć naredbe CMP (CoMpare). Naredba CMP ima adresni deo koji određuje jedan od brojeva koje poredimo; drugi se obavezno nalazi u akumulatoru.

```

3000 AB 00 31 LDA #0
3003 F0 0E BEQ nula
3005 10 06 BPL pozitivna
3009 9F FF LDA #255
300B E8 01 31 STA reš ; broj je negativan
300C 40 #0 RTS
300D -ciklus
300D A9 01 LDA #1
300F 60 01 31 STA reš ; broj je pozitivan
3010 40 #0 RTS
3011 E6 71 INC #71
3013 80 01 31 STA reš ; broj je nula
3016 40 #0 RTS

```

Tabela sa slike 18 će nam pomoći da razumemo stanje fleгова posle poređenja. Ukoliko su brojevi jednaki, setuju se Z i C fleгови. Ako su brojevi različiti, treba da odredimo koji je veći, ali to određivanje nije baš jednoznačno. Posmatrajmo, na primer, situaciju kada broj \$20 iz akumulatora poredimo sa \$FO (CMP #FO): reklo bi se da je \$FO veći. Ukoliko, međutim, uođe podatke posmatramo kao označene brojeve, \$20 će biti veći jer \$FO predstavlja negativan broj — 16. Carry fleг je nadležan za prosto logičko poređenje — u datom će slučaju on biti resetovan jer je sadržaj akumulatora manji od sadržaja memorije. N fleг je, sa druge strane, zadužen za aritmetičko poređenje: on će biti resetovan jer je akumulator veći od memorije.

Uslov	Neoznačeni	Označeni
1. A < M	BCC manje	BMI manje
2. A <= M	BCC manje_ili_jednako	BMI manje_ili_jednako
3. A > M	BEQ manje_ili_jednako	BEQ manje_ili_jednako
	BEQ dalje	BEQ dalje
	BCC veće	BPL veće
4. A >= M	BCC veće_ili_jednako	BPL veće_ili_jednako
	BEQ jednako	BEQ jednako
5. A = M	BNE različito	BNE različito
6. A <> M	BNE različito	BNE različito

slika 18

Bezijk poznaje šest testova (manje, manje ili jednako, veće, veće ili različito, jednako i različito) koji treba bez razmišljanja programirati i u assembleru: u tome pomaže slika 19, koju ćete ubrzo naučiti napamet. Zanimljivo je da se neka ispitivanja ne mogu obaviti bez primene dve BRANCH instrukcije, ali su se konstruktori 6502 pobrinuli da najpotrebniji testovi budu jednostavni za programiranje.

Osim instrukcije CMP, opremili smo i naredbama CPX i CPY koje omogućavaju poređenje brojeva upisanih u registre X i Y sa memorijom. Dejstvo ovih naredbi na fleгоve je identično sa naredbom CMP.

Da čitava ov diskusija ne bi ostala mrtvo slovo na papiru, pripremili smo mali primer: program sa slike 20 pretražuje memorijske čelije \$3000 — \$30FF i prebrojava pozitivne, negativne i nula bajtove, a zatim ispisuje izveštaj o rezultatima ispitivanja. Ispisivanje rezultata smo, doduše, poverili bežjiku, jer iz mašina još ne umemo da se obračamo tastaturi i ekranu.

```

2500 A2 00 LDI #0
2502 8E 02 40 STX pozit ; Brojači se postavljaju na nulu.
2505 8E 01 40 STX nul
2508 8E 00 40 STX negat
250B -ciklus
250B BD 00 30 LDA $3000,X ; Uzmi sledeći broj.
250E F0 08 BEQ jednak_nuli
2510 10 0C BPL plus
2512 EE 00 40 INC negat ; Broj je negativan - uvećaj brojač.
2515 4C 21 25 JMP next ; Na kraj petlje.
2518 -jednak_nuli
2518 EE 01 40 INC nul ; Broj je nula - uvećaj brojač
251B 4C 21 25 JMP next ; Na kraj petlje.
251E .plus
251E EE 02 40 INC pozit ; Broj je pozitivan - uvećaj brojač.
2521 .next
2521 E8 INX ; X pokazuje na sledeći broj.
2522 D0 E7 BNE ciklus ; Ako je X nula, posao je završen.
2524 60 RTS ; Povratak u bežjik

```

slika 20

Relativno adresiranje

Sve naredbe grananja (skok ako je neki uslov ispunjen) kod mikroprocesora 6502 predstavlja primere takozvanog *relativnog adresiranja*. Pogledajmo, na primer, naredbu BEQ PC+&20 i pretpostavimo da se ova instrukcija nalazila u memorijskim čelijama \$1000 i \$1001 (videćemo da

Mali oglasi

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

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

Prva stvar koju treba da uradite je da se odlučite da li želite obični ili ukovireni mali oglas.

CENA OBIČNOG MALOG OGLASA do dvadeset reči je 900 dinara. Svaka naredna reč košta još 60 dinara, s tim što oglas ne sme da ima više od 50 reči. Adresa oglašivača se ne računa u cenu.

CENA UKOVIRENOG MALOG OGLASA je 900 dinara po visinskom centimetru, s tim što se mogu zakupiti najmanje 32 slova znaka. Ako se ne iskoristi čitav prostor u jednom redu, računa se broj redova a ne broj znakova. Za ukovirene oglase preko 5 cm cena je 1400 dinara po centimetru.

Poželjno je da vaš mali oglas počinje sa Prodajem, Kupujem, Držim časove, Menjam... ili nečim sličnim što ukратно ukazuje na sadržaj oglasa.

Da ne bi bilo zabune, obavezno naznačite da li želite obični ili ukovireni mali oglas, i zajedno sa tekstom vašeg malog oglasa pošaljite i priznanicu o uplati na adresu redakcije: GALAKSIJA, BULEVAR VOJVODE MIŠIĆA 17, BEOGRAD, sa naznakom „za male oglase u RAČUNARIMA“.

„SPEKTRUM“

Imamo najnovije programe. Tražite besplatan katalog. Cena kompleta je 600.— din. Zvanično 20 sati.

KALIPSO SOFT, Crvenog Kriza 11, Zagreb, 041/416-163

SCARE CROW PIRATE CO.

Nudimo dobre, jeftine igre, brzu uslugu uz besplatan katalog. Poklon SCARE CROW BEOZ a po narudbi uslužne programe. Komplet 600.— din KNIGHT RIDER, DAN DARACH, ACTION REFLEX, MAFIA CONTRACT 2, MIND STONE, NINJA MASTER, HUNCH BACK 3, KID-NAPP, STAINLESS STEEL, LABYRINTH, ATLANTIC CHALLENGER.

Kenan i Aida Selmanović, Bratstva Jedinstva 31/2, 71000 Sarajevo. tel. 642-896 (071). Mišković Dejan, Bratstva Jedinstva 41, 71000 Sarajevo, 071/612-136


FUTURE SOFT CLUB vam nudi najatraktivnije programe za ZX Spectrum na YU tržištu, pojedinačno i u kompletima. CAULDRON 2, SPORT OF KINGS, NINJA MASTER, ROBIN OF SHERLOCK, HUNCH BACK 3, brza isporuka, BONUS programi, povoljne cene. Zovite nas na telefone 011/401-995 Vlada i 011/402-218 Saba

SPEKTRUMOVCI!!! Jedini koji uz nisku cijenu programa (60 i 80 din.), nimalo direktno iz Spectruma. Garantirano ispravan i nakon nekolicina godina. Uvjerite se! Uz to potpuno novi i najnoviji programi, popusti i besplatan katalog potražite na adresi:

MIHALOVIĆ BRANIMIR, Kaštelanska 43, 54000 Osijek

ZX Spectrum — izaberite programe iz drugih oglasa i javite... cena programa 60 din. tel. 030/81-992

Oldtimer SOFT — najveći izbor programa za Spectrum — kvalitetni snimci, — umerene cene — besplatan katalog za novembar tražite na 011/436-137 do 15 časova ili pismom na adresu: Milivoj Radosavljević, Braće Nedića 2, 11000 Beograd



ŠVE ŠTO VAM JE POTREBNO ZA OŠTILJAN SAG DA KORIŠTITEK TRAJALJE KOD COMETSOFTWARE

SPEKTRUMOVCI!!!! JEDINI KOJI NEMAJEŠE POSITIVNI UPLATNI PROGRAM ZA VOJVUJINA

NOVO U KATALOGU

LASER GENIUS

baterija programa za mašinske programere

- assembler sa ekstramskim editorom
- monitor sa debagrom
- analizator mašinskog koda

THE WRITER

profesionalni program za obradu teksta

ŠALJIVANICI ZA SVE VINE UPLATE

MIHALOVIĆ BRANIMIR
Petra Lovrić 57, 11000 Beograd
tel. 011/556072 pošte 15

Spektrumovci — najpovoljnije usluge

Bilo koji program 75 dinara. 18 programa po vašoj želji + kasetna C90 + PTT — 1986 din. Presnimavmo obavezno MULTICOPY-jemi Garancija 6 meseci.

Najnoviji programi: DYNAMITE DAN 2, SPINDRIZ 2, ACE, NINJA MASTER, KUNG FU MASTER, DAN DARACH, STAINLESS STEEL, BIG BEN, NIFTY LIFT, PANAMA YO, BACMAN, ALIEN HIGHWAY, KIREL, GHOST AND GOBLINS, BOULDER DASH, WORLD CUP CARNIVAL, WHO DARES WINS 2, BOUNDER.

Milunović Veselin, 7. Vojvodnarska brigade 52, 21206 Sremska Kamenica tel. 021/619-991

VRHUNSKI GRADEVINSKI PROGRAMI za Spectrum: OKVIRI, ROŠTILJI, REŠETKE, DIMENZIONIRANJE, TEMELJI, ISKAZ ARMATURE i drugi. Za radne organizacije i pojedince. Besplatan katalog. Gino Gradin, Kozala 17, 51000 Rijeka, tel. 051/517-291

PIRAT SOFTWARE LTD vam nudi najnovije programe za ZX Spectrum. Za specijalan jeseni popust besplatan katalog. 011/466-895 Nikola 462-844 Gvozden

Nova pravila

Kada smo pre nekoliko meseci uveli mogućnost da se mali oglas izdiktira telefonom, imali smo, pre svega, u vidu hitne slučajevne. Čitaocima se, međutim, ovaj način toliko dopao da druge više praktično i ne koriste, čak ni kada se radi o veoma dugačkim i teškim oglasima, prepunih imena najnovijih igara. To je previše skupo i za njih (plaćaju visoke telefonske račune) i za redakciju (veliki angažman oko prijema i velikog administriranja — slanje uplatnica i naknadna provera igara) i veoma nepouzdan —

u oglasima ima više grešaka. Osim toga, primetili smo da pojedini oglašivači nakon izlaska „Računar“ iz štampe za boravljaju na svoje materijalne obaveze prema listu. Stoga smo odlučili da poostriremo kriterijume za prijem malih oglasa telefonom. Na ovaj način se ubuduće mogu dostavljati mali oglas samo u dva slučaja:

- ako se naručuje oglas do 20 reči, i
- ako se naručuje ponavljanje oglasa, bez obzira na njegovu dužinu, iz prethodnog broja.

Svi ostali mali oglasi mogu se dostaviti na jedan jedini način: poštom sa priloženom uplatnicom.

Spektrumovci CANNIBALS SOFT vam nudi najnovije programe po niskim cenama — komplet 250, program 60 din. Veliki popusti. Pila za besplatan katalog: Ronald Vlatko, M. Tita 52, 54512 Feričani, Stanić Dalibor, 17 travnja 2, 54500 Našice, tel. 054/723-064 ili 711-883

ROM DISASSEMBLY 1700 din. (engleski), Mašicac za apsolutne početnike 1000 din. (prevod). Napredni mašicac 1000 din. (prevod). Odlučne kopije, a pored toga i veliki broj programa sa opširnijim uputstvima. Rakita Dragan, put B. P. Oreda 25/131, 2100 Novi Sad, tel. 021/399-639

QUICKBREAK SOFT programi za spektrum i C-64 80 d/kom. Od najstarijih do supernovih — animak izdvojeno iz kompjutera, garancija 30 dana. Katalog besplatan. GJS, Talovska 46, 32300 G. Milanovac. Tel. 032/714-220

VMS PIRAT CO. Njegoševa 15/III 34220 apovo. tel. 034/851-334, prodaje preko 400 uslužnih programa za spektrum. Veliki broj uputstava. Više od 1800 programa u kompletima i pojedinačno. Sveke sedmice novi komplet igara. Mogućnost preplate. Snizili smo cene. Apolutna garancija za vrhunski kvalitet animaka. Spisak najnovijih igara besplatan. Za veliki katalog sa svim programima i uputstvima pošaljite u pismu 200 din.

ZX SPECTRUM!!!!

50 izdanih kompleta, svaki kompletno do 20 programa. Kasetna SONY (TDK od 60 minuta, programi, poštarina i troškovi pakiranja — samo 2000 din!!!)

Malo razmisлите i usporodite i sa drugim ponudama. Ekspres dostava. Besplatan katalog. Iztok Strazar, Kulehova 44, 61110 Ljubljana, tel. 061/453-907 P. S. Na svaka 4 kompleta (odjednom) peti besplatan!

MACHINE LIGHTNING — 4 X 48 K. Paket programi za dizajniranje igara i rad na mašinu (odličan assembler, 90 strana uputstva). LASER GENIUS, 4 programa za rad u mašinu (EKRAMSKIM EDITOR, ASSEMBLER, MONITOR, DIBAGER-ANALIZER, 15 STRANA UPUTSTVA). Uz to još BLAST, WRITER, MASTER OFFICE, BETA BASIC 3.0.C. Sve sa opširnijim uputstvima.

Rakita Dragan, put Bosanskih Partizanskih odreda 25/131, 21000 Novi Sad, 021/399-639

Super jeftino! Spektrumovci, program 60 dinara a iz 12 programa 60 din. (Aca, Ghost's...), telefon 071/524-069. Vojci Predrag, Braće Vujčića 5, 71000 Sarajevo

Spektrumovci! Nove igre u kompletima. Cena jednog kompleta (12 igara) bez kasete 600.— din. Oba kompleta 1000.— din.

KOMPLET A: CLIFF HANGER, MOLECULE MAN, SEX CRIME, GRAND FUNK, HOCUS FOCUS, ROBIN OF CHERLOCK, 1, 2, 3.

KOMPLET B: GHOST'S AND GOBLINS, BIGGLES 1 i 2, YU BACK TO SKOOL, ARENA: Adnan Rudonović, Trg Bolkova Buha 13, 71000 Sarajevo, tel. 071/544-598 pre podne

Spektrumovci!!! Pažnja!!!
Super-najnoviji komplet od 13 igara za samo 900 dinara (pojedinačno 150 din. program).
Komplet 34: Kung Fu Master, W. Cup Carnival, The Planets 1 i 2, Comet Game, Big Ben, Pyrcourse, A.C.E. ...
Komplet 35: Hunk Back 3 Bobby bearing, Cauldron 2, Black Arrows, Knight Rider, Bernie, Ninja Master ...
Predrag Denadić, D. Karakajča 33, 14220 Lazarevac, tel. 011/811-208

SILICON DREAMS ponovo dotazi sa velikim novostima u poslovima i obrazovnom softveru:
SPEED OFFICE (obrada svih mogućih podataka)
LASER GENIUS (pisanje i učenje mašina code postaje igra)
PROJECTOR (obrada podataka uz visoku grafiku)
THE WRITER (fontastičan tekst-procesor)
BLAST 4.0 (ovo nema niko)
MASTER OFFICE (8 fantastičnih uzdužnjaka)
MEGA BASIC 4.0 (najnoviji)
Vi smo razmislili i nazovite nas. Dobit ćete uzlazne programe iz svih oblasti. Imamo sve što se može naći u Jugoslaviji. Opširno uputstvo imamo za svaki program. Živković Velibor, tel. 021/87-69, Stritar Milan, Bul. AVNOJ-a 53 21000 Novi Sad.

Spektrumovci!!! Pažnja!!!
85 najboljih, odabranih uzlaznih programa na tri kasete-kompleta, svaki po 1000 din. Komplet uzlazni 1—37 programa:
PASCAL HP45, ASSEMBLER, MEGA BASIC 1.6 IS COMPILER, BELA LOURINE DRAW ...
Komplet uzlazni 2—26 programa:
DEVPAC 3, ART STUDIO, ILLUSTRATOR, MASTER COREY, BETA BASIC 3.0 ...
Komplet uzlazni 3—22 programa:
TURBO LOG, DEVPAC 7.8, MEGA BASIC, LEONARD, LOTC ...
Predrag Denadić, D. Karakajča 33, 14220 Lazarevac, tel. 011/811-208

Ukoliko želite da ovidate svetom računara naučite nove programске jezike. FASCAL, FORTH, Lisp, C, Mikro Prolog i sve verzije BASIC-a možete naći kod nas. Pobediti računare i naučite ga da radi ono što poželite. SILICON DREAMS, Živković Velibor, tel. 021/87-069, Stritar Milan, Bul. AVNOJ-a 53, 21000 Novi Sad.

KOMODOR

Najnovije i najbolje. Igrе 30—130 dinara (Green Beret, Commando, Kane, Uridium Spindizzy). Popusti, nagrade, iznenađenja, besplatni katalog. Zvonimir Ožbolt, Vrt Jagode Truhelke 2, 54000 Osijek, telefon 054/31-956

Izuzetno povoljno svakog meseca komplet najnovijih programa i igara. Komplet sadrži 25 programa. Komplet + kasete + PPT — 1800 din. Mihailović Dejan, Mile Alasa 54, Beograd, tel. 011/180-834

C-64: Komplet 1. International karate 1 i 2, War play, Match Day, Green Beret Superbowl, Macadam Burner, Saboteur (sa spektrom), Ghots'n Goblins, Gid, Biggles 1 i 2.
Komplet 2: Way of tiger 1, 2 i 3, Gladiator, Ping Pong, 1 i 2, Spellbound, Spindizzy, Trust, Cauldron 2, Formula 1.
Komplet kasete 1200 din. Aleksandar Obradović, Braća Gribića 56, 85340 Herceg-Novi, tel. 082/43-374

GEOS 2500 din. sa disketom, THE NEWSROOM — Kućne novine 3500 din. sa disketom. GIGA-CAD na 3 dvostrane diskete 5500 din. PLATINE 64 2500 din. sa disketom. MEGA-DISK program zauzima celu disketu, služi za kopiranje originalnih programa sa trake na disk 2000 din., sa disketom. GOJIC NENAD, Pere Koršorić 18, Beograd 11185

Komodor 64 — komplet 15 najnovijih programa + kasete + 5 pokloni — 1500 din. Cyrocipe 3, Art studio, Rolistball, Knight Games, Cyborg, Equinox, Metal, Ax, Sex Games 2, Trimming, Piray, Titanic, Krik, Edusom, RIDI, Jet, Šterf Ronald, Makizama Gorkog 9, 42000 Varaždin, 042/46-095

COMMODORE 128 *** PROGRAMI
POJEDINAČNO ILI U PAKETU I
Cena 128
PREKO 50 NASLOVA
ISPORUKA ZA 24 ČASA
** (011) 604-329 **

COMMODORE 64 — 2 kompleta, svaki 15 programa.
Komplet+kasete — 1200 din. oba 2200 din.
Sve pojedinačno (Starquake, Uridium, Bombjack ...)
Spisak besplatni. Biljan Zvonimir jr., A. Mihanović 6, 56000 Vinkovci

LSH — LABORATORIJA SOFTWARE/HARDWARE
Nudimo vam veliki broj najnovijih igara i poslovnih programa za komodore 64, te izradu poslovnih programa po vašem zahtjevu. Vrlo povoljni uvjeti kupnje i mogućnost preplate za najnovije programe. Najjednostavniji način naručivanja programa. Besplatni najopširniji katalog u našoj zemlji.
Reset tastir: 1300 — sa poštarinom.
Naša adresa: LSH, F. Stareja 10, 42000 Varaždin

Širok izbor najnovijih programa za komodor 64: WAR PLAY, INT. KARATE 1 i 2, ONE MAN AND HIS DROIT, URIDIUM 1, 2 i 3, RAMBO 2, TIME CRYSTAL, MERMAID 7, COMBAT ZONE, GHOST'S and GOBLINS, SAMANTA FOX SP, ROAD TO NOWHERE, BIGLES 1 i 2, RASPUTIN, WHO DARES WINS 3, SUPER BOWL, SPELLBOUND, LEADERBOARD GOLF. Na svakih 10 programa 3 poklanjanje. Tražite besplatni katalog. Sa šif. 013/812-405 i Bojan 013/813-373

Comodore 64 — Superbowl, WHAMI, Kane, Last VB, RATS, Enigma Force 2, Robin of Sherlock, Tau Ceti, Nexus, Summer Games 3 + kasete — 1700 din. SIMON ŠERŠ, Vilharjeva 27, 65270 Ajdovščina

Commodore 64 superhitovi: Miami Vice, Space Talsman, Beer Burt, Boulder Dash 5, Lords of Midnight 2, City Morgue, Quest for Tyree 3, Ping-Pong 3, Titanic, Piracy, War Play, Popeye 2,

STUDEN-SOFT-SERVICE CBM-64
Evropske verzije najnovijih američkih Top-hitova. Svi najnoviji potpuno razliječni programi, najjednostavnije preinamivanje. Trust-VERIFY garancija kvaliteta. Adr: Studen lišća 5a, LAMELA 4/6, 77000 Bihać, tel. 077/223-162

ZA COMMODORE 64 komplet: Beach-head II, Uridium II, Ping-Pong II, Bomb Jack, Kawasaki R, R., Wham, Green Beret, Fairlight, Leaderboard, Superbowl, Ragby + kasete — 100 din. Željko Filipović, tel. 057/31-210

MAX AND ENGELS poseduje: YIE ER KUNG FU, DESERT FOX, MERCANARY, URIDIUM, TOUR DE FRANCE, PING PONG, YABA 000, ELECTRA GLIDE, W. Mc GUGAN BOXING i još bolje. Većina cena 200 din. Najveća 400 din. Popusti do 50%! Super paketi! Besplatne igre! Josipović, S. Petrovića 36, 21206 S. Kamenica

Komodor 128 — najveći izbor CP/M programa (Wordstar, Dbase, C, Cobol ...), disketnih (Yams, Superbase, Laser, Music Maker ...), kasetnih (Abacus, turbo, Einstein ...). Besplatni katalog. Šterf Ronald, U. M. Gorkog 9, 42000 Varaždin, 042/46-095

SUNNOSOFTWARE CLUB C-64, CP-128, CP/M vam predstavlja u svom besplatnom katalogu izbor samo najkvalitetnijih kasetnih i disketnih programa. Više od 2000 naslova, među kojima su i oni koje će drugi oglašavati tek idućeg meseca. Tel. 021/20-179

Noviteti 86. Besplatni katalog sa opisima. Pojedinačno ili u kompletima od 25-50 programa. Samo 2000 din. KATIE, MAGNATS, BABBIT PIE, CASTER KINGDOM, VELOCIPED?, HACKER, TIME TRAX, RIPPING JARNS, SYNTH SAMPLE 3, BASE, DEUTSCH GSF STUDIO SPORT, CRAZY COMET, EQUINOX, STAR, INTROUENE DEMO 1, LANDING ON TB 19, JS 90, SPLITTING PERSONALITIES, CONCERT WEMBLEY, FLOOPYSON, PIRACY, BEER BELLY BURS, LAS VEGAS, HUMAN RACEG, AV CREATOR, RANSOM THE KING, CV DAVID, TRAP TTI, TALISMAN ELEVATOR, INC WHEN DOVES CRY, COUNTDOWN, MIAMI VICE, KNIGHT RIDER, RMC TITANIC, FUCK DEMOK KNIGHT GAMES 1, 2, 3, 4 i 5. Komplet 13 + kasete — 2500 din. Mizalil Asim, Nurije Podzera 74, 72000 Zenica, tel. 072/22-556

C-64, najnoviji programi. MIKIE, PAR-REROB, NOMAD, STREETHAWK, TURBO ESPRIT, EXPLODING FIST 2, MISSION AD-4, KNIGHT GAMES, PARALAX, Aleksandar, 011/489-0882

NEW BEST SOFT nudi najnovije programe najjeftinije u Jugoslaviji! Cena jednog programa 20—40 din. Besplatni katalog. Pišite nama adresa: Franic Danijel, I. L. Ribara 29, 55250 Oriovac

C-64: KORONIS RIFT, TITANIC, NOMAD, TURBO ESPRIT, MIKIE, ELEVATOR, PS15, MARBLE MADNESS, GREEN BERTS II, KNIGHT GAMES, RAID OVER MOSCOW II, INFILTRATOR, TAU CETI, BATTLE FOR BRITAIN, ZOIDS, TIME TRAX.
Komplet, kasete, poštarina i poklon programa 1400 din.
Petrović Branko, Senjačka 44, 11000 Beograd, tel. 011/650-509

Samanta Fox, Spellbound, Gringo, Visitors 2, Beret, Macadam bumper, Death Waka, Mission Elevator, izaberite 10 programa za 1000 din.
Branislav Čobanov, P. Dragoljna 55/1, 21480 Srbobran, tel. 021/730-364 Janković Gabor, Dr. Dorda Bašića 25, 21480 Srbobran

C-64/C-128 OLIMPIC SOFT
Najveći izbor hitova po 40 dinara. Besplatni katalog. Moguća razmjena. Empl. War Play, Saboteur, Uridium 2, Biggles, Zorro, Time Crystals, Phantoms. Pejočević Jovan, Olimpijska 9, 71000 Sarajevo, tel. 071/461-967

Za C-64 program MEGATAPE, probija sve zaštite i kasete, samo 1300 din. Besplatni katalog najnovijih programa. Koltrun Marjan, trg. I. Kukuljevića 14, 41000 Zagreb

C-64. Veliki izbor programa animirani direktno iz računara, svaki program verifikovan (garantovano bez LOAD ERROR). Komplet sa 30—50 programima 2.000—... din. Katalog besplatni. Šala Staletović, Đure Đakovića 1, 56000 Vinkovci

DISK programi za C-64: Geos, Multiplan, Disk Wizard, Platine 64, Tasword, Chessmaster 3000, Wordstar, Fortran, Profi Painter, Jet, Megadisk, Newsroom, Gamemaker ...
Zoran Mitošević, ul. Miroslava Ristića 36, Stanovo, 34000 Kragujevac

GRIN SOFT: Sve što imaju drugi, imamo i mi, ali još bolje! Tu su: V-faktors, Green Beret, Uridium 1 i 2, Death Waka, Saboteur, Spellbound, Leaderboard, Way of the tiger 1 i 2 i mnogo drugih super hitova. GRIN LEON, Dinković 9, 71000 Sarajevo, ul. 071/386-82

ALAN SOFT C64 odabrani programi u paketima iz 7 različitih područja:
30 pomodnih programa
30 akcionih programa
30 sportskih
30 arkadnih
30 muzičkih
30 društvenih
20 radno amaterski
1 paket + kasete — 2000—... din. Komplet 7 paketa 10.000—... din. Ako ste poštebni, naplaćite. Posebna besplatna pomoć. ALAN SOFT, 7. travnja 30, 58311 Stobrec

Komodorovcil Superkomplet A: Tomahawk, Submarine simulator, Babeus, Hackman, Karnage, Visitors, Boulder Dash 6, NOMAD, Saboteur 2, Road to nowhere, frogger parker, black hole + 5 hitova. Komplet + kasete + PTT — 900 din.
Gera Ad Tota Software, Mihaila ličić 35, 34000 Kragujevac, tel. 034/33-914



SNOOPY UNIVERSAL STUDIOS
C-64 CP-128 (CP/M)
NAJNOVIJI I NAJBOLJI
KASETNI DISKETNI PROGRAMI
U KATALOGU SA PREKO 3000
PROGRAMA
MI NUDIMO MNOGO VIŠE!
075/235-666
TINIĆ AD
SENJAK E/4
75000 TUZLA

Veliki izbor programa po malim cenama, za vas C64. Tražite besplatni katalog. Hanek Vica, Marina Dvatala 171, 54000 Osijek, tel. 064/52-378

POLU PIRAT za komodor 64 poseduje sve što iz ostali, ali prodaje upola cene. Izaberite iz drugih oglasa svoj komplet. Alpha Sada, Marijane Gregoran 44 V, Beograd, tel. 011/417-371

C64 komplet 2: Cyborg, Knight Games (5 programa), NOMAD, Dragons Lair, Ninja, Riddlers Dan, Streethawk, Ninja Master, za 1500 din za saetehw, tel. 011/461-885

KRIMINAL SOFT ima iz Engleske sve što zamislite. Svi programi su u turbu i razvijeni. Cijena SVAKOG PROGRAMA je 100... a 30 izabranih 2300 dinara. Midtčič Dino, I kraj-ke brigade 10, tel. 077/222-951, i Srdić Dejan, Maršala Tita 15, tel. 077/227-170, 77000 Bihać

C64 u kompletu: SHOUGUN, MER MAID, MADNESS JUDGE, KORONIS RIFT, THE JET, CAUSES OF CHAOS, ROAD TO NOWHERE, KNIGHT OF DESERT, BLACK HOLE, PING PONG 3, BOULDER DASH 5, WIND RIDER, THE FALL, GUY. Komplet sa kasom 1500. Siniša Novaković, 11143 Beograd, Braće Jerkoviće 91, tel. 011/461-655

YU C.S. je najveći i jedini pravil izvor svih najnovijih programa za C-64, PC-128, a od skoro i za IBM PC. Ako želite svaki novi program, literaturu, hardware (spec speedos - mode-mi...), servis, besplatno korišćenje računara - YU-mailbox (011/213-836), obratite se nama: - YU C.S. - Na adresu 36, 62391 Prevalje tel. 062/851-338 (od 19 hr) - YU C.S. - Cviljeva 125/20, 11000 Beograd, tel. 011/787-269

Komodore Soft nudu za samo 3000 din 120 najnovijih programa. Cena kasete je uračunata u cenu kompleta. Isporuka odmah. Ghost's and Goblins, Monty on the run, BBC emulator, The way of tiger 1, 2, 3, Star Ping-pong, Int. Karate 1 i 2, Spellbound, Green Beret, Speed King, Beach Head 3, Spindizzy itd, tel. 011/532-442. Tomislavović Igor, Kumodražka 29, 11090 Beograd.

Za **COMMODORE 64** nudimo vam sve kasetne hitove. Do izlaska ovog broja, naš programi, kao što su: Jack The Nipper, Sweden Erotic, World Games (S. Games III), Druid, Droids, Green Beret II, Ace of Aces, Super Cycle, Knight Riety, 30 Pika-doo, Asterisk II, Beatles, War Hawk, Airwolf II, Fundus icke, biće već zastareli... Na nov Tokiović, Cviljeva 125/20, 11000 Beograd, tel. 767-269/011

NEWSROOM program za C64/C128 na 5 strana disketa plus moje 3 diskete, plus program za presnimavanje, plus kratko uputstvo, plus poštarna - sve samo 4000 dinara. Isporuka odmah. Radanović Milorad, Ilije Grbića 856, 74400 Derventa

COMMODORE 64 - komplet 20 najnovijih novih programa + kasetu i uputstvo za početnike možete dobiti za samo 2200 din. Pišite na adresu: Šantić Milivoje, Slobodna Principa 14, 72250 Vitez, ili telefonom 072/71-705

DTP SOFT vam nudu septembar/oktobar 86 u kompletu. Komplet A: NOMAD, Icupe, Cyborg, Knight Rider, Spindizzy, Silent Service, Split, Personalities, Johnny Reb 2, Ping Pong 3, Titanic. Komplet B: W. A. R. Parallax Lab, Miami Vice, Time Trax, Flight Deck, Superwolf, Hacker 2, Leaderboard, Ghost's n Goblins. Komplet + kasetu + PTT - 1800 din. Sve ovoj i još malo više možete dobiti na adresu: Vujić Zoran, Bul. revolucije 67, Beograd, 011/426-901, 424-533 Dragin i 458-448 Dule

KOMODOR 64-20 igraza BEATLES, CYBERUN, MOVIE, SEX GAMES 2, RAMBO 3, SUMMER GAMES 3, ASTERIX, MIAMI VICE, PUSZI, ST SMOUGH!!! MOTO GUZZI, TEDDY, KUREL, MAFIA 2, GARBAGE, FLIPER, DRUDS, BATMAN, ICARUS, POD, CYROSCOPE 3, KNIGHT RIFER KOMPLET + KASETA + PTT - 1298 din. ISPORUKA ODMAH!!! GO-JIĆ NENAD, PERE KOSORIĆA 18, BEGRAD 11185

Komodor 64 - najnoviji programi, pojedinačno ili u kompletu. Besplatni katalog, poklon programi. Novol Mega-komplet - 1024 programa u kasetama i uputstvima - 40.000 din. Stanić Ronald, 42000 Varazdin, Ul. M. Gorkog 9, tel. 042/46-095

SHIFT-SOFT - KOMPLET 4: DRUIDS (uđite u magični svet druid-aka), WARHAWK, INDRID SPORT, GHOST HUNTER II, TRISTAN AND SOLDIA (iz vremena kralja Artura), SCARB, ONE BITE TWO DEEP, HOLLYWOOD ON BUST, DRACULA (brrr...), W. A. R. (reklama iz Video Computer Games-a: „To be a hero is easy. Staying alive is more difficult...“), GYROSCOPE 3, PARALLAX, BOULDER DASH 5, 7, 8 (šve boje od božjeg), JOHNNY REB II, SERENADA, ART STUDIO (svetini crtači hit sa ATARI-a i MC INTOUCH-a), I.C.U.P.S., CYBORG, ARCAHA (Pravi nastavak STAFF OF KARATHA), STREET HAWK, ICARUS, PEPSI-COLA (specijalno iznenađenje!), GALAXYBIIRDS, INVALIDSI SOCCER (igrači u kolica, goimnazi sa štakama, publika objavljuje - crni humor). Sve ovoj + kasetu + PTT = 2500 dinara. Mogućnost TURBO TAP-a sa Vašim imenom za 1000 dinara!!! **SHIFT** - SOFT: Vasović Nenad, 11000 Zemun, Dubrovačka 19, 011/210-884 Kramenović Goran, 11700 Novi Beograd, Dušana Vukosavljevića 74, 011/172-234

COMMODORE 64 vam predstavlja: GHOST AND GOBLINS, GREEN BERET, WAR PLAY, BEACH HEAD IV, INTERNATIONAL KARATE II, LEADER BOARD... i drugi najnoviji programi - komplet 15 igara + kasetu - 1500 din. Miličković Slobodan, Jureta Gagarina 199/5, 11700 Novi Beograd, tel. 011/168-196

ALAN SOFT C64, PC128, korienički programi, aplikacije, programski uputstva (SH-HS). Budite upućeniji na oelu i u kući. Naše je maimama: „PROGRAMI BDE UPUTSTVA (SH-HS) = 1777-0 PROGRAMI SA UPUTSTVOM (SH-HS) = USPH“ Besplatni katalog sa širokim opisom programa. **ALAN SOFT**, 7. travnja 30, 58311 Stobreg

Komodorović Samo ovog meseca za čitaoce „Računars“ odobravamo super popust za igre. Imamo sve najnovije. Odaberite: 50 programa + kasete + poštarna + najkujza za programiranje. Referencu broj - 5000 dinara. Pojedinačno one je neplaena u katalogu, tražite ju za nas. Rok isporuke je 24h. Prijatnu igru želi vam M and S soft: 11 Bulevar 130/193 11070 Novi Beograd Tel. 011/146-744

Komodor 16, 116, +4 veliki izbor programa, snimanih direktno iz računara. Kvalitet super, cena povoljna. Dragan Ljubavijević, 19210 Bor, 3. Oktobar 302/6, tel. 030/33-941

Komodorice: COnan, ALICE, AZTEC, NINJA, HACKER 2, CAR DESTRUCTION SET, SHOUGUN, CORONIS RIFT, NOMAD, GIOJE (sa 4), TITANIC, MIAMI VICE... + kasetu + PTT - 1550 din. Sve programe imamo! Pojedinačno: DRAGON'S LAIR + uputstva... Duduković Goran, Dalimirova 22, 12000 Zrenjanin, tel. 012/21-909. Zolotarevčević, Vardarska 26/17, Požarevac, tel. 012/23-540

C-64 Komplet 1: Kermit s Storymaker, Art Studio, Sex Games II, Boulder Dash 7,8, Ronald Rudaček, 4th Dimension, Asterix, Caesar's Travels, Illusion, Thrust, Astro Pilot, Pro Boxing OK, Iridis Alpha, Fuck Porno, W.A.R. Komplet 2: Mordon's Quest, Icarus, Space Frontier, Edoon, Street Hawk, Trining 130, Bigmine, Teddy, Rolletoleball, Pricks Porno Game, Dr No Legal Sex, Arcana, Super Boulder Dash, Chessmaster, Rambo Trainer, Johnny Reb 2. Cena kompleta 2000 din. + kasetu. Amir Trifkij, Senjak E/4, 75000 Tutza, tel. 075/235-866

Commodore 16, +4, izaberite 10 programa snimljenih turkom. Kasetu - 3000 din. Winter Games (5 delova), Ace Strike, Jetset Willy, Mr. Pynterica, Rockman, W. S. Baseball, Strip Poker, Bongo, Xargon vras, Commando II, Fire Ant, Baby Berks, Invasion X, 30 Grand Prix, Čobanov Nerst, Nikole Teše 18, 21480 Sbrobran. Od 8-13h 021/730-161

Commodore 16, +4, 20 programi, besplatni katalog. Commodore 64, 128 - nova ponuda od 4500 programa. Katalog je 300... din. ali porudžbinom vrata. Cene: Bernat Šandor, Rade Končara 23, 23000 Zrenjanin

NEW NOW SOFT podaje veliki broj uslužnih programa i igara za Komodor 64. Cene niske, katalog besplatan, usluga kvalitetna. Za narudžbe preko 3500... din. besplatna kasetu.

Brzak Vojislav, Bul. AVNOJ-a 29, 21000 Novi Sad, Adamovića Dušan, 021/366-205

AMSTRAD

Amstradsoft - COMPUTER ART vam nudu komplete od 10-15 specijalno izabranih programa po ceni od 800 din + kasetu:

Komplet 26: KANE, SA COMBAR, KUNG FU MASTER, BOATKAC, HIGH GUN, NER... Komplet 25: GHOST AND GOBLINS, MARSPOUT, FAIRLIGHT, TOMAHAWK, komplet 24: SPINDIZZY, PAC-MAN, GUNFIGHT, GREEN BERET... Sabljak Ivica, 7. Vojvodanska brigade 62, 21208 Sremska Kamenica

SATANA SOFT nudu: Batman, Fairlight, Saboteur, Marsport, Samanta Fox i slične. Katalog besplatan. Jovanović Dejan, Maršala Tita 48, 31330 Pribor, tel. 033/53-383

AMSTRAD CPC 464 - Baseball, Basketball, Gatscherer, Juggernaut, Last V8, Strom, Green Beret, Datzan, Spindizzy, Gunfight, Dynamite Dan, Fairlight a prelude, Shogun, One man and his droid Mastertronic, Tomahawk, Impossible mision i na kraju: Laser Genius (najbolji assembler). Željčić Hvoje, J. Dvnraska 10, 54000 Osijek, tel. 054/23-790

Nightmare Soft za CPC 464, Komplet igara: Green Beret, Marsport, Fairlight, Gun Frigh, Starion, Turbo Espirit, Movie, Batman, Way of Tiger, Rambo + kasetu za 2000 din. David Lešić, Maistrova 12, 62000 Maribor

Amstradsoft BOMB JACK (pitaše Spektromove), SPACE SHUTTLE (izvorna simulacija u svemiru), PORNO SHOWN 20 digitalizovanih slika, TURBO SAVE (ekrativno ulikavanje programa), digitalizirani govora (AMSTRAD govora) i niz drugih programa po ceni od 100... din. Super katalog besplatni! Stojanović Tražan, Vojvodica Putnika 186, 71000 Sarajevo, 071/613-349

AMSOFT YU CP/M SOFTWARE predstavlja najnovije CP/M programe: DR GRAPH, DR PASCAL MT + STOCKCONTROL, SUPER DATA INTERCHANGE, ZIP (za dBase II), MULTIPLAN, DATASTAR, CAMBASE DATABASE, TURBO PASCAL, COBOL 80, ALGOL, MBASIC, micro PROLOG, LISB, BASIC-Compiler, C-Compiler, C-BASIC 80 Compiler, ED, 1000, FORTRAN 80, WORDMASTER, DISC DOCTOR. Komplet CP/M 2.2: MICROSCRIPT, MICROPEN, MICROSPREAD, POWWER, COPYFILE Komplet CP/M 3.0: WORDSTAR 3.34, dBASE II, SUPER-CALC 2, ZIP, COPYFILE Novo CP/M Utility programi: TURBO PASCAL, TOOLBOX, MODULES, TURBO SOURCE LISTER, SUPERCALC 2 UTILITIES, PASCAL MT + UTILITIES, C-COMPIILER UTILITIES. Svevaki kupe CP/M programa dobija na poklon CP/M program MINI CAD-CAM. Novi uslužni programi: TAH-WORD 128, Mailmerge, PPOFF PAINTER, DATAMAT, TRANSMAT, Histori C-Compiler (na kaseti), DEV-PAC 32 (disc verzija), TURBO DISC (povećava brzinu diska). Nova literatura: C-COMPIILER Manual, CP/M OPERATING SYSTEM MANULA, CP/M PLUS OPERATING SYSTEM MANUAL, C+ PROGRAMMING language MANUAL.

Komplet najnovijih igara (10 do 30 programa), sa kazetom 2999... din. AMSOFT YU, Trg Republike, 4, 41000 Zagreb, Telefon 041/315-478 ili 041/270-777

WAY OF THE TIGER, TURBO ESPRIT, LAST V8, SHOUGUN... u kompletu ili pojedinačno. Cena jednog kompleta (14 + kasetu) - 1600 din. Tražite besplatni katalog, Matić Vanda, I. L. Ribara 38, 11318 Miloševac

Prodajem „AMSTRAD“ 6128 sa zaleženim monitorom. Modulator M1M2 za kolor televizore. Diskete 3 1/2. Štampac DMP 2000. Sve je novo i ocarinjeno. Telefon: 011/555-785, Beograd (11344), Ul. Petra Lubarde br. 3

ARČADIA SOFT — i dalje atraktivne nove igre!
komplet 28: KANE, PEEP SHOW, ZODIS, STORM, BARRY MC GUGAN BOXING... i nešto starije igre GHOST AND GOBLINS, CODENAME MAT, FAIRLIGHT, GATECRASHER, SPELL-BOUND... oena jednog kompleta 1000 din. Tražite besplatan katalog. Do izdaka „Racuara“ još i STARION, COLLAPSE, TREAD BANGER, PETAK 13, CAULDRON 2 i hanočji Milan, Nikoše Đurkovića 6, 11000 Beograd, 011/478-423

AMSBYTE DIVISION vam nudi veliki izbor programa za vaš MASTRADSCHEIDER. Među preko 250 programima možete našti i super hitove MOVIE, WAY OF THE TIGER, THREE WEEKS IN PARADISE, GHOST AND GOBLINS, TURBO ESPRIT. Tražite besplatan katalog. Nenad Vasiljević, Narodnog fronta 5/II, 11000 Beograd, tel: 011/688-782

BAJA SOFT. Veliki izbor programa za AMSTRAD/SCHEIDER CPC 464 po vrlo pristupačnim cenama. Uskoro i kod nas: INTERNATIONAL KARATE, WEST BANK, BOMB JACK... profesionalna i brza usluga.

Brilj. Blažo, Ante Zrnančić 15a, 88000 Mostar, tel. 088/415-203

CPC SOFT nudi za vaš AMSTRAD: programe na kasetama, disketama kao i literaturu. Među 300 programima koje posedujemo još prodajemo meseca bili su: ZODIS, KANE, STORM, MOVIE... za programe imamo OCEAN-ovih LASER GENIUS-a za one sa diskom CP/M programe: Dbase II, Cobol, PROFI program, Mini Office 2, igru CAULDRON 2 i mnoge druge. Od literature posedujemo 17 knjiga, uputstva ili prevoda, a među njima i najnoviju knjigu: THE AMAZING AMSTRAD. Naručite besplatan katalog. Mihailo Krstić, Uskočka 7/7, 11000 Beograd, 011/628-412

Najnovi i najbolji programi: AMNES. GUM FRIGHT, STORM, JUGGERNAUT, MOVIE. I druge programe možete jeftino dobiti kod HOT SOFT-a. Naručite besplatan katalog. Akcentijević Željko, šetalište Velika Vlahovića 94. tel. 054/45-273 Katić Bernard, Šamačka 3, tel. 054/21-019, 54000 Osijek

ATARI SOFT-CLUB Zrenjanin, 600X/800X/130XE, programi i literatura. Prodaja, razmena i saradnja. Izuzetno: kursevi za učenje BASIC-a na kasetama. Za opitran katalog: pošliti 100 din. Lacomanić Dejan, Sinduljevića 31/a, 23000 Zrenjanin, tel. 023/66-679

Amstradovci, BROTHER'S SOFT vam nudi veliki izbor starih, novih i najnovijih programa po najpovoljnijim cenama. Tražite naš katalog. Besplatan i ilustrativan. Ili vam se bitij jang. Koldžo Otašević, Dinarska 29, 71000 Sarajevo, tel. 071/646-398

ILE SOFT neverovatno!!! 35 izabranih programa za 1500 din. Uputstva za sve programe 500 din. Detaljne informacije non-stop na 091/225-523, Vitanov lija, Skopje, Jurija Gagarina 41a

ATARI

ASCII SOFT nudi vam najbolje programe za vaš ATARI 800 XL/130XE. The last V8, Montezuma Revenge, Spy Hunter, Zorro, Zenji, James Bond, Strip Poker, Goggles, Hacker, Road Race, F-15... NOVA literatura i novi programi za disk jedinice. Upravo je stigao i nov program za učenje bezijska i upoznavanja ATARI-ja (2 kasete C80 + opširno uputstvo). Fanduro Zoran, Đurkovića 43, 23000 Zrenjanin, 023/63-521

Prodajem-menjam programe jeftino. Sa-jeem besplatan katalog. Sokolov Dragan, Karpolovo vostanje 8, A10 1/10, Karpoš 3, 91000 Skopje, tel. 091/258-397

ATARI ST programi programi posezno- no ali v kompletni Katalog 400 — din. IGY-SOFT (062/865-464)

ELEKTRON

Acorn ELECTRON — razmenjujemo i prodajem programe (Elite, Chuckie Egu Forth, Logo, Pascal. Besplatan katalog. Adresa: Toni Stojanović, Isajija Mažovića 91060 Đ Petrov, Skopje, tel. 091/311-538

GALAKSIJA

ILE SOFTWARE 5 igara, telegrafija, izne-ruđenje 1500 din. Zvučik preko kasetofona, visoka rezolucija (sva bez hardverskih promena). Dosađ nevideti efekti. Vitanov lija, Skopje, Jurija Gagarina 41a, tel. 091/225-523

IBM

„GTS Elektronika prodaje za 5.950.000 din. Kompletan dual-screen CAD sistem: PC XT 640T, kolor graf monitor, printer 160 ch/m, ploter 120 mm/sek, miš, koprocesor, programski paket „AUTO-CAD“ tel. 055/238-036

„GTS Elektronika“ prodaje za 3.950.000 din. PC XT 640T (100% kompatibilan sa IBM), RAM 64K, TURBO mode, multi I/O kartica, TTL graf monitor, printer LES, graf kartica, 2 flopija, po 360 K, HARD 215 Mb, XT tastatura, DOS 3.1 + BASIC. Dostava, puštanje u rad i jedan dan obuke. Garancija 1 godina, servis trajno u roku od 24 sata. Tel. 055/238-026

HARDVER

Prodajem memorijsko proširenje (64K) za GALAKSIJU (15000 din.) i komplet za proširenje memorije SPECTRUMA na 40K ili 80K (12000 din.) Boris Lovoević, Jakičkovića 35, 41000 Zagreb

Komodor C64/C128 — kasetni programi po izboru 20—35, disketi 100—250, a CP/M programi 500—800 din. po programu. Konektor sa priključnim kablom: joystick (1700), kasetofon (1800), USER-port (2400), produžni kabal za joystick (3000); diskete 5,25" DDSD (1300)... kabal za kasetofon, isparuća odmah... HARDVER/SOFTVER C64/C128, pb 83, 74400 Derвента, tel. 074/832-832 i 833-775

Prodajem potpuno nov i ocarinjen Commodore C-16 sa kasetofonom i 32 programa. Igor Grčić, Neluova 132/42, tel. 011/162-702 N. Beograd.

Povoljno prodajem ZX Spectrum 48 K sa posetijom, kasetofonom, džojstikom, interfejsom, raznom literaturom i 55 kaseti sa oko 650 programima. Đaković Igor, Karlovačka 16, 11000 Zemun. Tel. 011/219-702

Prodajem Commodore 64, flopi disk, matični štampač i ploter. 012/83-479 od 16—20 časova. Vlado Dobrić, Vreba-vova 20, 12220 Veliko Gradiste.

KEMPTON INTERFACE Povežite svoj SPECTRUM sa palicom za igranje. Mogućnost prilagodavanja svih postojehih objekata. Garovano kvalitetom. Cena 7750 din. + reset (1000 din.) Uverite se!!! CHAMP Hardver, Jovica Petrović, V. Karadžića 46, 91300 Kumanovo

izrađuju EPROM-programator za Commodore 64 te prilazim odgovarajući list-ing programa (BASIC + ASSEMBLER-SKI listing). Dolinar Andrej, tel. 051/731-481

COMPUTER SERVIS — Sinclair, Commodore, Amstrad, Atari — brzi i kvalitetni popravci — ugradnja reseta i druge usluge dipl. inž. Kovačić Anđelko 111 Vrbik 33a/6 41000 Zagreb tel. (041/359-277 od 10 do 17 sati

Prodajem AMSTRAD CPC 464 sa zelenim monitorom zajedno sa 20 brojeva časopisa „Moj Mikro“ i 60 programa na 6 kaseti. Malisa Novaković, Strmoštrajeva 40, 23000 Zrenjanin. tel. 023/61-885

KOMODOROVCI!!! Ako želite da preko zvučnika svoj televizora čujete muziku iz originalnog komodorovog kasetofona, tada naručite sklop koji sa priključuje na C64. Shema + sklop + program — 1500 din. Informacije: Rajčić Stojpa, Matična 33, Slavovska Požaga 55300 tel. 055/75-921

Prodajem BBC B, dvostruki drav „Cumann 400K“, 32K božnog RAM-a + 40 disketa sa programima i najboljom ROM sajevnom. Literatura, 30 časopisa. Sve zajedno. Žarko Supićić, Draškovičkova 27. Zagreb, tel. 041/417-745

HARD AND SOFT KARTUS JOY — novi superjok kvaliteta izrađa garantira precizno upravljanje i dugi vijek trajanja. Jedinstven oblik, posebno pogodan za držanje u rukama. Nešto zaista novo.

G-ROM za PRINTER MPS 82 omogućuje prenos H-res grafike iz Simon's BASIC-a, Newsrooma, Doodle itd. 6 Grafičkih naredbi, 6 abceda, 10 slobodno programirajućih znakova. Rezolucija 640x400.

Podrobnije informacije na adresi: HARD AND SOFT P.P. 163 51000 Rijeka

Prodajem univerzalni INTERFACE za ZX Spectrum (D/D izlaz i izlaz CENTRONIX interfejsa, A/D, D/A pretvarač). Cena po dogovoru. Santalica Bernard 92/3 Rabac, 52221. tel. 052/872-279 i 872-155.

AMSTRADOV izlaz na TV (modulator) i ispravljač prodajem. Garancija godinu dana. Leon Kuna, Mihano-vevićeva 18/3, 43500 Daruvar. tel. 046/31-883

SPECTRUM HARDVER Veliki izbor dodatnih uređaja vrhunске kvalitete po povoljnim cijenama. Specijalni popust na komplet Kaps-ton interface + Redotac pak. Informacija 058/589-987, P.N.P. electronic, Jeretova 12, 58000 Split

COMMODORE HARDVER Veliki izbor dodatnih uređaja vrhunске kvalitete po povoljnim cijenama. Specijalni popust na ROM module. P.N.P. electronic, Jeretova 12, 58000 Split. Tel. 058/589-987

ATARI ST HARDVER Veliki izbor dodatnih uređaja vrhunске kvalitete po povoljnim cijenama. RAM 1 Mb, ROM-ov, TV modulator, disk 720 Kb, Fast Basic kartidge, literatura, programi. Telefon 058/589-987 P.N.P. electronic, Jeretova 12, 58000 Split

Literatura

KOMODOR C64/C128 — prevodi: Pritu-čnici C64 (800), C128 (1550), Simon's BASIC (600), Easy Script (300), Vizevit (350), Pascal (300), Help C64 (300), Mae 64 (400), Praktikalni (650), Monitor (200), CP/M c64 (300), Wordstar CP/M C64 (800), COBOL CP/M C64 (500)... Katalog besplatan. Isporuka domah. HARDVER/SOFTVER C64/C128, PB 83, 74400 Derвента, tel. 074/832-832 i 833-775

Spectrum. Rečnik Englesko-Srpskohrvatski o obratno (oko 2000 naj-češće upotrebljivih reči) + kasete + poštarina (1500 din.). 011/497-862 od 17—19 h. D. Marjanović, B. Jerković 123/36 — 11000 Beograd

COMMODORE 64 — profesionalni prevodi: Programer's reference guide — 3800 d, Mašinski zvezice za početnike — 2400 d, Grafika i Zvuk — 2400 d, basic Priručnik — 2400 d, Simon's Basic — 2400 d, Pascal — 1200 d, Na višestruke narudbe popust 10%. Duško Bjelotomić, Centar 1, 54550 Valpovo. Tel. 054/82-865 ili 041/883-141

Spectrum, rečnik kompjuterskih izraza i reči upotrebljivanih u kompjuterskoj tehnici + kasete + poštarina — 1400 din. tel. 011/497-862 od 17—19 h.

AMSTRAD CPC 464 — profesionalni prevodi: Pritu-čnici — 3000 d, Locomotive BASIC — 2600 d, Mašinsko programiranje za početnike — 2800 d, Višestruka narudba 10% popust. Duško Bjelotomić, Centar 1, 54550 Valpovo, tel. 054/82-865 ili 041/883-141

HEACKER/JI KNJIGE ZA VAS: C Programming Language (2000), C Guide (1300), Logo Programming (1500), Forth Programming (2500), Pascal User Manual (2000) Forthan — 700 stranica (4000) i M88020 — budući (4000). TINE JARM BENTLOVRENC 20 68212 VELIKA LOKA

Tehnike crtanje na mašincu (2)

programiranja Amstrad/šnajder

Svrha PLOT rutine je da na ekranu nacrti tačku. Rutina može biti konačnog i neodređenog tipa.

Amstradova PLOT rutina iz ROMA ne zadovoljava karakteristične kvaliteta koje treba da ima PLOT rutina dobrog grafičkog programa i to zbog sponosti iscrtaivanja tačka. To je zbog toga što je ta rutina predviđena za rad u svim tri moda (grafički modovi: MODE 1, MODE 2 i MODE 0) i zato što je predviđena da crta tačku u izlomljenoj bitnoj mapi (posle hardverskog SCROLL-La). Pošto se u grafičkim programima ne upotrebljavaju sva tri moda odjednom, niti se upotrebljavaju hardverski SCROLL, možemo znatno povećati brzinu rutine održujući se pomenutih svojstava.

Izvan uhodanih staza

PLOT rutina može da bude formirana na dva načina koji imaju zadovoljavajuću brzi-

Pass 1 errors: 00

5200	10	org	21000		
5200	20	and	0		
5200	CD143C	30	call	8bc14.	
5200	3F	40	di		
5200	117F82	50	ld	4e,030	
520F	3C7	60	ld	a,199	
5211	ED47	70	ld	1,a	
5213	95	80	and	at	
5214	8157	90	ld	a,1	
5218	CD2A52	100	call	plot	
5219	D1	110	pop	de	
521A	7A	120	ld	a	
521B	33	130	or	a	
521C	1B	140	dec	de	
521D	20F4	150	jr	nz,st	
521E	8157	160	ld	a,1	
5221	08	170	ret	z	
5222	3D	180	dec	a	
5223	ED47	190	ld	1,a	
5225	117F82	200	ld	4e,030	
5226	10E9	210	jr	st	
522A	0307	220	plot	ld	c,7
522C	47	230	ld	b,a	
522D	7B	240	ld	a,0	
522E	A1	250	and	c	
522F	ED44	260	neg	a	
5231	01	270	add	a,c	
5232	07	280	add	a,a	
5233	07	290	add	a,a	
5234	07	300	add	a,a	
5235	96C0	310	or	X11000110	
5237	32F8F2	320	ld	bb+11,a	
523A	C33A	330	sr	d	
523C	C31B	340	rr	e	
523E	C33A	350	sr	d	
5240	C31B	360	rr	e	
5242	C33A	370	sr	d	
5244	C31B	380	rr	e	
5245	32E0	390	neg	a,192	
524B	82	400	add	a,d	
5249	57	410	ld	4,a	
524A	78	420	and	a,b	
524B	8F8F8	430	and	240	
524D	2000	440	ld	b,0	
524F	07	450	ld	1,a	
5250	29	460	add	bl,bl	
5251	EB	470	ex	de,bl	
5252	10	480	add	bl,de	
5253	83	490	ex	de,bl	
5254	29	500	add	bl,bl	
5255	29	510	add	bl,bl	
5256	19	520	add	bl,de	
5257	80	530	or	b	
5258	A1	540	and	c	
5259	07	550	add	a,a	
525A	07	560	add	a,a	
525B	07	570	add	a,a	
525C	04	580	add	a,b	
525D	07	590	ld	a	
525E	C846	600	bb:	bit	0,(bl)
5258	C9	610	rax		

Pass 2 errors: 00

Table used: 42 from 175
Executes: 21000

Primer 2 Nema ulaznih parametara.

nu. Jedan od načina je rutina koja koristi mašinske instrukcije SET, RES i BIT, a drugi način je formiranje rutine korišćenjem mašinskih naredbi AND, OR i XOR.

Svaki od ovih načina ima svoje kvalitete i svoje mane. Prvi tip može istovremeno da radi i kao TEST i kao UNPLOT rutina, a ne može da radi kao INVERT rutina. Drugi tip PLOT rutine ne može da radi kao TEST, a radi kao INVERT rutina, tj. može da invertuje tačku. Prvi način je nešto brži od drugog.

Ulazni parametri PLOT rutine „amstradovog“ ROMA upisuju se u DE i HL registre,

stre, pri čemu se DE kreće od 0 do 639, a HL od 0 do 398. Ovde DE označava tačke na X osi, a HL tačke na Y osi. Ova rutina tretira svaku drugu liniju, što je razumljivo jer „amstrad“ ima samo 200 linija. Ulazni parametar rutine koji označava broj reda može se staviti u osmoinbni A register i pri tom dobiti na brzini i jednostavnosti rutine. Tada se redni broj linije kreće od 0 do 199.

Prvo što PLOT rutina treba da preračuna od rednog broja date tačke je da izračuna adresu bajta u kome je tačka, a zatim broj bajta koji treba da se setuje u bajtu. Pošto svaki bajt ima 8 bita, broj bajta u redu se dobija tako što se redni broj tačke podeli sa 8. (Primer 1: linijski brojevi od 150 do 200). U binarnom sistemu 8 je ceo broj (8=1000).

Poznato je da se šiftovanjem bajta za jedno mesto udeseo vrši deljenje sa 2. Ako se tri puta uzastopno izvrši šiftovanje, sadržaj bajta se deli sa 8 (2³=8). Tako se deljenje broja sa 8 može izvršiti sa tri mašinske naredbe SRL reg.

Broj bita u bajtu se dobija kao ostatak pri deljenju rednog broja tačke sa 8. Ostaje

Pass 1 errors: 00

5200	10	org	21000		
5200	20	and	0		
5200	F3	30	di		
5209	47	40	plot	ld	b,a
520A	40	50	ld	c,e	
520B	C33A	60	sr	d	
520C	C31B	70	rr	e	
520E	C33A	80	sr	d	
5211	C31B	90	rr	e	
5213	C33A	100	sr	d	
5215	C31B	110	rr	e	
5217	32E0	120	ld	a,192	
5219	82	130	add	a,d	
521A	57	140	ld	4,a	
521B	78	150	ld	a,b	
521C	8F8F8	160	and	240	
521D	2000	170	ld	b,0	
521E	07	180	ld	1,a	
521F	29	190	add	bl,bl	
5220	EB	200	ex	de,bl	
5221	10	210	add	bl,de	
5222	83	220	ex	de,bl	
5223	29	230	add	bl,bl	
5224	29	240	add	bl,bl	
5225	19	250	add	bl,de	
5228	80	260	or	b	
5229	8007	270	add	7	
522B	07	280	add	a,a	
522C	07	290	add	a,a	
522D	07	300	add	a,a	
522E	04	310	add	a,b	
522F	07	320	ld	b,a	
5230	78	330	ld	a,c	
5231	8007	340	add	7	
5233	47	350	ld	b,a	
5234	3890	360	ld	a,120	
5235	81	370	rrca		
5237	80F0	380	rr	ax	
5239	80	390	or	1b11	
523A	80	400	ld	1b11,a	
523B	07	410	rax		
523D	C846	420	bb:	bit	0,(bl)
523F	C9	430	rax		

Pass 2 errors: 00

Table used: 33 from 149
Executes: 21000

Primer 3 Vidi primer 1

Pass 1 errors: 00					
5200	10	org	21000		
5200	20	and	0		
5200	F3	30	di		
5209	47	40	plot	ld	c,7
520A	47	50	ld	b,a	
520B	78	60	ld	a,b	
520C	A1	70	and	a	
520E	ED44	80	neg	a	
5210	01	90	add	a,c	
5211	07	100	add	a,a	
5212	07	110	add	a,a	
5213	07	120	add	a,a	
5214	96C0	130	or	X11000110	
5216	32E8E2	140	ld	bb+11,a	
5218	C33A	150	sr	d	
521B	C31B	160	rr	e	
521D	C33A	170	sr	d	
521F	C31B	180	rr	e	
5221	C33A	190	sr	d	
5223	C31B	200	rr	e	
5225	32E0	210	ld	a,192	
5227	82	220	add	a,d	
5228	57	230	ld	4,a	
5229	78	240	ld	a,b	
522A	80F0	250	and	240	
522C	2000	260	ld	b,0	
522E	07	270	ld	1,a	
522F	29	280	add	bl,bl	
5230	EB	290	ex	de,bl	
5231	10	300	add	bl,de	
5232	83	310	ex	de,bl	
5233	29	320	add	bl,bl	
5234	29	330	add	bl,bl	
5235	19	340	add	bl,de	
5238	80	350	or	b	
523F	A1	360	and	c	
5238	07	370	add	a,a	
5239	07	380	add	a,a	
523A	07	390	add	a,a	
523B	04	400	add	a,b	
523C	07	410	ld	b,a	
523D	C846	420	bb:	bit	0,(bl)
523F	C9	430	rax		

Primer 1
Ulazni parametar: A — kreće se od 0 do 199, DE — kreće se od 0 do 639 zaprijači su svi registri osim i registra.

Pass 1 errors: 00

```

5200 10 org 21000
5200 20 ent 8
5200 F3 30 di
5200 43 40 plot: ld b,e
520A C33A 58 arl d
520E C31B 60 rr e
520E C33A 70 arl d
5210 C31B 80 rr e
5212 C31A 90 arl d
5214 C31B 100 rr e
5210 2A2052 110 ld hl,(hpl)
5219 19 120 add hl,de
521A 70 130 ld a,b
521C E067 140 and 7
521D 47 150 ld b,a
521E C200 160 and a,a,120
5220 07 170 aa: rrrca
5221 10FD 180 djnz aa
5223 86 190 or (hl)
5224 77 200 ld (hl),a
5225 C9 210 ret
5226 00C0 220 hpl: defw 49152

```

Pass 2 errors: 00

Table used: 42 from 126
Executes: 21000

Primer 4

Ulazni parametri: DE — kreće se od 0 do 639, lokacija na adresi HP sadrži adresu reda u kome se crta tačke. Nepromenjeni su registri C i I.

tak se može izračunati ako se nad lakšim bajtom rednog broja tačke izvrši logična operacija AND 7. (Nalaženje ostatka pri deljenju rednog broja sa 8 opisano je u prethodnom broju „Računara“). Linija 70 primera 1.

SET, BIT i RES

SET, RES i BIT naredbe spadaju u dvovalne naredbe procesora Z80. Od značaja za ovu rutinu su samo naredbe koje se izvršavaju nad bajtom na adresi HL: BIT b, (HL), SET b, (HL) i RES b, (HL).

Prvi bajt ovih naredbi je CB, a drugi bajt je direktno zavistan od ulaznih parametara.

Pass 1 errors: 00

```

5200 10 org 21000
5200 20 ent 8
5200 F3 30 semi: di
520A 0000 40 xor a
520E 1100C9 60 arl: ld de,49152
520F 21A067 70 ld hl,20000
5212 C30E 80 arl: rrrc (hl)
5214 EB 90 ex de,hl
5215 C31E 90 rr (hl)
5217 23 110 inc de
5218 13 120 inc hl
5219 23 130 inc hl
521A 14 140 djnz d
521B 20F5 150 jr ar,ar,at
521D 10ED 160 cpl
521F C9 170 ret

```

Pass 2 errors: 00

Table used: 43 from 122
Executes: 21000

Primer 6

Nema ulaznih parametara.

Zadnja tri bita drugog bajta naredbe su vode ista i iznose 1, 1, 0. Prva dva bita određuju da li se radi o komandi BIT, SET ili RES. Ako su upaljena oba bita (1,1), radi se o komandi SET. Ako je upaljen prvi bit (1, 0) radi se o komandi RES, a ako je upaljen drugi bit (0,1) radi se o komandi BIT. Preostali bitovi određuju broj bita u bajtu na adresi HL koji će biti tretiran. Adresa bajta u VIDEO memoriji se dobija tako što se redni broj bajta u redu sabere sa adresom reda. Adresu reda izračunava modifikovana A HL rutina u sklopu same PLOT rutine. (Primer 1: nijlski brojevi 210 do 410, brzina PLOT rutine se može videti ako se u računar unese primer 2).

Rutina se može prepraviti u SET ili RESET rutinu promenom bita u broju iza OR u liniji 130 (Primer 1).

AND, OR i XOR

PLOT rutina koja koristi AND, OR i XOR mašinske naredbe je nešto sporija od

Pass 1 errors: 00

```

5200 10 org 21000
5200 20 ent 8
5200 CD14BC 30 call #c14
520B F3 40 di
520C 21FF87 50 ld hl,49151
520F 2A2052 60 ld bc,2000
5212 01D007 70 ld bc,2000
5215 C5 80 arl: i/c
5216 70 90 push bc
5217 CB2152 100 call print
521A C1 110 pop bc
521B 83 120 dec bc
521C 70 130 ld a,b
521D 71 140 or /c
521E 20F5 150 jr ar,ar,at
5220 C9 160 ret
5221 018A7F 170 print: ld bc,8770a
5224 ED 180 out (c),c
5226 2000 190 ld b,e
5228 06 200 ld i,a
5229 3E30 210 ld a,50
522B 29 220 add hl,hl
522C 29 230 add hl,hl
522D 29 240 add hl,hl
522E 04 250 add a,h
522F 07 260 ld h,a
5230 EB 270 ex de,hl
5231 2A4952 280 ld hl,(hpl)
5234 23 290 inc hl
5235 2A4952 300 ld (hpl),hl
5236 018006 310 ld bc,20050
5238 1A 320 arl: ld a,(de)
523C 77 330 ld (hl),a
523D 13 340 ld h,a
523E 79 350 ld a,c
523F 84 360 add a,h
5240 07 370 ld a,c
5241 10FB 380 djnz at
5242 018E7F 390 ld bc,8770a
5243 3E45 400 out (c),c
5248 C9 410 ret
5249 FF3F 420 hpl: defw 49151

```

Pass 2 errors: 00

Table used: 53 from 156
Executes: 21000

Primer 8

Nema ulaznih parametara.

prethodne rutine, ali ne menja samu sebe, pa se može koristiti u ROMU. Rutina se razlikuje od prethodne samo u načinu setovanja bita u bajtu. Pošto rutina izračunava poziciju bita u bajtu, ona taj broj prebacuje u B registar (Primer 3: linija 350), a u A registar upisuje broj 128, tj. u binarnoj formi broj �.

Svaki put kada se obrne petlja, A registar se rotira za 1 bit udesno. Posle izvršene petlje, broj setovanih bita u A registru jednak je broju bita u bajtu koji bi imala data tačka. Posle toga se A registar ORuje sa bajtom na adresi HL (Primer 3: linija 390). Time je izvršeno crtanje tačke. Ako se više PLOT rutina izvršava u istoj liniji (npr. u horizontalnoj kompresiji), nije potrebno da se svaki put izračunava adresa reda, već se adresa reda može dobiti promena AHL rutine i staviti na određenu lokaciju, na početku procesa odakle je zatim uzima PLOT rutina. Takva PLOT rutina je znatno brža i kraća od PLOT rutine koja sadrži modifikovanu AHL rutinu u svom sastavu. (Primer 4 i 5). Rutina se prebacuje u INVERT modu promenom linije „390 or (hl)“ u „390 xor (hl)“.

Pass 1 errors: 00

```

5200 10 org 21000
5200 20 ent 8
5200 018A7F 30 print: ld bc,8770a
520B ED 40 out (c),c
520D 2000 50 ld b,e
520E 14 60 ld h,e
520F 3E30 70 ld i,a
5212 29 80 add hl,hl
5213 29 90 add hl,hl
5214 29 100 add hl,hl
5215 C4 110 add a,h
5216 07 120 ld h,a
5217 EB 130 ex de,hl
5218 2A3052 140 ld hl,(hpl)
521B 23 150 inc hl
521C 23 160 ld (hpl),hl
521D 77 170 ld a,50
521F 018000 180 arl: ld a,(de)
5222 14 190 ld (hl),a
5223 77 200 ld (hl),a
5224 13 210 ld (hl),a
5225 79 220 ld a,c
5226 84 230 add a,h
5227 07 240 ld a,c
5228 10FB 240 djnz at
522A 018E7F 250 ld bc,8770a
522B ED 260 out (c),c
522F 05 270 ret
5230 FF3F 280 hpl: defw 49151

```

Pass 2 errors: 00

Table used: 43 from 139
Executes: 21000

Primer 7

Ulazni parametri: A-sadrži ASCII kod slova koje se ispisuje. Svi registri osti memorija, I su zaprižani. Napomena: Rutina mora biti iznad 16384. bajta memorije! Obavezno omogućiti interapt pre poziva rutine!

Pass 1 errors: 00

```

5200 10 org 21000
5200 20 ent 8
5200 F3 30 di
5200 43 40 plot: ld a,e
520A 0007 50 and 7
520B 0004 60 add a,7
520E C067 70 add a,a
5210 07 80 add a,a
5211 07 90 add a,a
5212 07 100 add a,a
5213 7000 110 or #11000110
5214 C33A 120 arl d
5215 C31A 130 rr e
521C C31B 140 rr e
521E C33A 150 arl d
5220 C31B 160 rr e
5221 170 170 arl d
5222 C31B 180 rr e
5224 2A2952 190 ld hl,(hpl)
5227 19 200 add hl,de
5228 C8 210 hpl: bit b,(hl)
522A C9 220 ret
522B 00C0 230 hpl: defw 49152

```

Pass 2 errors: 00

Table used: 42 from 120
Executes: 21000

Primer 5

Ulazni parametri: D-kreće se od 0 do 639, lokacija na adresi HP sadrži adresu reda u kome se crta tačka. Register BC je nepromenjen.

Prikazivanje uputstva

Prilikom pisanja grafičkih programa javlja se problem prikazivanja uputstava.

Uputstvo se može čitati u memoriji na dva načina: u vidu slike i u vidu teksta.

Čuvanje menija u vidu slike je vrlo jednostavan način, ali zauzima 16 kilobajta


```

10 'Vertikalni kompresor
20 FOR xx=0 TO 639:FOR yx=0 TO 398 STEP 2:IF TEST(xx,yx) THEN PLOT xx,yx,0:PLOT
xx,yx,2,13
30 NEXT:NEXT

```

```

10 'Horizontalni kompresor
20 FOR yx=0 TO 398 STEP 2:FOR xx=0 TO 639:IF TEST(xx,yx) THEN PLOT xx,yx,0:PLOT
xx,2,13
30 NEXT:NEXT

```

Pass 1 errors: 00

```

5200      10      org 21000
5200      20      ent s
5200      F3      36      dec a
5209 BEC7      30 comp: ld a,199
5205 0000      50      ld b,e
5200 CD25D2     00 stti: call A_HL
5210 ED47      70      ld i,a
5212 E5      00      push hl
5213 CB3F      90      stl a
5215 C664      80      and a,100
5217 CD25D2    110 call A_HL
5211 D1      120      pop de
5213 40      130      ld c,b
521C 0050     140 ld b,00
5217 E5      150      ex de,hl
521F 1A      160 st:  ld a,(de)
5220 86      170      or (hl)
5221 D1      180      ld (de),a
5222 71      190      ld (hl),c
5223 23      200      lsc hl
5224 13      210      lsc de
5225 1870     220      41BZ st
5227 ED57     230 ld a,i
5229 C0      240      ret z
522A 3D      250      dec a
522B 1830     260      JF st
522D 1100C0   270 A_HL: ld de,40152
5230 4F      280      ld c,a
5231 E670     290      and 240
5233 83      300      ld b,e
5234 67      310      ld i,a
5235 29      320      and hl,b
5236 EB      330      ex de,hl
5237 10      340      and hl,de
5238 15      350      ex de,hl
5239 29      360      and hl,hl
523A 29      370      and hl,hl
523C 10      380      and hl,de
523C B1      390      or c
523D E607     400      and 7
523F 67      410      and a,a
5240 67      420      and a,a
5241 67      430      and a,a
5242 64      440      add a,b
5243 67      450      ld h,a
5244 70      460      ld a,c
5245 C0      470      ret

```

Pass 2 errors: 00

Table used: 54 from 100
Executes: 21000

Primer 10

Nema ulaznih parametara.

slobodne memorije. U grafičkim programima se primenjuju dva načina takvog čuvanja slike. Prvi način je izmena sadržaja memorije i video memorije, a drugi način je da se u memoriji formiraju dve video memorije (ukupno 32 kilobajta). Zatim, pr ispisivanja uputstava, slike treba staviti u drugu video memoriju, a uputstvo iz prve lokacije prebaciti u video memoriju. Taj način oduzima 32 kilobajta, tako da ostaje slobodno samo 10 kilobajta, što je dovoljno za grafički program.

Vrlo interesantni efekti ispisivanja uputstva mogu se postići primenom programa iz primera 6. Liniju 70 ovog programa treba izmeniti tako da umesto 25000 stoji adresa memorije. Ovo se može postići spustanjem VIDEO memorije u prvu lokaciju. Tako se postize trenutni prelaz sa SCREENa na MENI.

Pass 1 errors: 00

```

5200      10      org 21000
5200      20      ent s
5200      F3      36      dec a
5209 00C0     40 vcomp: ld b,200
5203 70      50 st:  ld a,b
5205 60      60      dec a
5200 CD0752    70 call A_HL
5210 228552   80 ld (bp),hl
5213 110000   90 ld de,0
5216 D5      100 st1: push de
5217 0F4D     110 ld c,301000110
5219 CD25D2   120 call plot
521C 2951    130      JF A_HL
521E ED57    140 ld a,i
521F 7080    150      JF 0000110
5222 326352  160 ld (b+),a
5225 C625D2  170 call bb
5228 D1      180      pop de
5229 25      190      push de
522A C03A    200      JF d
522C C81B    210      rd a,i
522D 7080    220      JF 0000110
5230 CD25D2  230 call plot
5233 91      240 st2: pop de
5234 70      250      lsc de
5235 7A      260      ld a,d
5236 F802    270      or #2
5238 20C0    280      JF st,sti
523A 70      290      ld a,e
523B F808    300      or #00
523D 2077    310      JF st,sti
523F 10CA    320      41BZ st
5241 C0      330      ret
5242 70      340 plot: ld a,e
5243 E607    350      and 7
5245 ED44    360      and a,7
5246 67      370      and a,a
5247 67      380      and a,a
5248 67      390      and a,a
5249 67      400      and a,a
524C ED47    410      rd a,i
524E 31      420      or c
524F 326352  430      ld (b+),a
5252 C83A    440      JF d
5254 CB1B    450      JF e
5256 C03A    460      JF d
5259 CB1B    470      JF e
525A C03A    480      JF d
525C CB1B    490      JF e
525E 240552  500      ld hl,(bp)
5261 19      510      and hl,de
5262 0B45    520 bh:  b15 0,(hl)
5264 C0      530      ret
5265 00C0    540 hpl: 000 40152
5267 1100C0  550 A_HL: 050 A_HL
526A 4F      560      ld c,a
526B 3670    570      and 240
526C 63      580      and a,b
526E 67      590      ld i,a
526F 67      600      and hl,b
526F 67      610      and hl,b
5270 67      620      ex de,hl
5272 EB      630      ex de,hl
5273 29      640      and hl,hl
5274 29      650      and hl,hl
5275 19      660      and hl,de
5276 31      670      or c
5277 E607    680      and 7
5279 67      690      and a,a
527A 67      700      and a,a
527B 67      710      and a,a
527C 84      720      and a,h
527E 67      730      ld h,a
527E 70      740      ld a,c
527F C0      750      ret

```

Pass 2 errors: 00

Table used: 94 from 100
Executes: 21000

Primer 11

Nema ulaznih parametara.

Pošto u Amstradovom JUMP bloku postoji rutina koja to radi, ovde ta mogućnost neće biti razmatrana.

Drugi način čuvanja menija je čuvanje u obliku teksta. Tada meni zauzima samo 2000 bajtova, a u video memoriji se PRINT rutinom. „Amstradova“ PRINT rutina je spora (jer je predviđena za rad u svatru grafičkom modu i u režimu izlomljene

Pass 1 errors: 00

```

5200      10      org 21000
5200      20      ent s
5200      F3      36      dec a
5209 BEC7     40 A_HL: 40 A_HL:
520C 4F      50      ld c,a
520D 3670    60      and 240
520F 63      70      and a,b
5210 67      80      ld i,a
5211 29      90      and hl,hl
5212 EB      100     ex de,hl
5213 19      110     ld a,e
5214 EB      120     ex de,hl
5215 29      130     and hl,hl
5216 29      140     and hl,hl
5217 19      150     and hl,de
5218 31      160     or c
5219 67      170     and a,a
521C 67      180     and a,a
521D 67      190     and a,a
521E 67      200     and a,a
521F 64      210     ld a,h
521F 67      220     ld a,h
5220 70      230     ld a,c
5221 C0      240     ret

```

Pass 2 errors: 00

Table used: 24 from 130
Executes: 21000

Primer 12

Ulazni parametri: A-kreće se od 0 do 199. Izlazni parametri: HL — sadrži adresu reda, A i B su nepromenjeni.

bitne mape video memorije), pa je zato nepogodna za tu primenu. Ako se odrekne- mo tih svojstava, možemo napisati PRINT rutinu koja svojom brzinom zadovoljava tražene uslove.

Definicija ASCII koda „amstradovom“ ROMu počinje od adrese 4300. Ova rutina se mora nalaziti iznad 16384 bajta, jer se u toku rada otvara donji ROM (programske linije 50 i 260).

Brzina ove rutine može se proveriti učitanjem i startovanjem primera 8, HP bajt sadrži adresu slova na ekranu i njegov sadržaj se izračunava po formuli:

$$HP = 49151 + x + (y - 1) \cdot 80$$

x — horizontalna koordinata slova
y — vertikalna koordinata slova

Kompresori slike

Kompresori imaju ulogu vizuelnog skupljanja ekrana. Ove ćemo razmotriti kompresore koji vrše horizontalno i vertikalno skupljanje sadržaja ekrana svodeći ga na polovinu. Horizontalna kompresija je jednostavnija od vertikalne, jer se može vršiti nad linijama, a ne nad tačkama kao kod vertikalne kompresije.

Bezik primeri kompresije dati su u programu 9. Mašinski primeri programa koji vrše kompresiju dati su u primeru 10 i primeru 11. Primer 11 koristi modifikovanu PLOT rutinu u procesu kompresije.

Ovaj put vam nudimo novu AHL rutinu koja je znatno brža i kraća od AHL rutine objavljene u prethodnom broju (Primer: 12). Ova rutina ostavlja slobodan B register, tako da se može koristiti u petljama formiranim sa DJNZ.

Date rutine zadovoljavaju sve potrebne uslove za primenu u grafičkim programima visoke kategorije.

Andrija Radović

Pretpostavljamo da ste prethodni članak o simulaciji rekurzije u „Računarima 15“ već proučavali, pa ovoga puta bez mnogo filozofiranja prelazimo na zadatak o Hanojskim kulama.

Neka su data tri štapa i n diskova različitog prečnika. Diskovi se mogu slagati na štap samo tako da obrazuju kulu (dakle, donji disk je najveći, na njemu nešto manji itd. i gornji najmanji). Neka se na početku svi diskovi nalaze na prvom štapi. Zadatak je napisati program za simuliranje premeštanja svih diskova sa prvog štapa na treći, tako da na kraju svi diskovi budu poredani u istom redosledu koji su imali na početku. Pri premeštanju diskova pridržavati se sledećih pravila: (1) samo jedan disk se može prenети sa jednog na drugi štap u jednom potezu; (2) veći disk se ne sme staviti preko manjeg diska; (3) drugi štap se može koristiti kao pomoćni.

Uz pomoć rekurzije zadatak o Hanojskim kulama rešava se vrlo jednostavno. Pretpostavimo da sa svaku kulu sa manje od n diskova već znamo da prebacujemo (to je rekurzivni korak) ideja je da „skrtačenu“ kulu od n-1 diskova (svi osim najvećeg) prebacimo sa prvog diska na pomoćni disk. Zatim preostali najveći disk prebacimo na ciljni disk. Na kraju prebacimo našu „skrtačenu“ kulu sa pomoćnog na ciljni disk. Rekurzivna procedura koja ovo radi data je na slici 1.

Pokušajte da bez daljeg čitanja teksta zadatak rešite ne koristeći rekurziju. Nije ni malo jednostavno, zar ne?

U direktnom pokušaju da iterativno rešimo ovaj zadatak najviše glatobolje zadaje nam to što ne vidimo jasno neki algoritam prebacivanja koji vodi ka cilju. Zato ćemo se poslužiti strim

lukavstvom. Simuliraćemo rekurziju iz procedure na slici 1, a zatim pokušati da dobiveno iterativno rešenje još malo pojednostavimo.

Procedura sa slike 1 ima četiri parametra, od kojih se svaki može promeniti u rekurzivnom pozivu. Zato oblast podataka mora sadržavati elemente koji predstavljaju sva četiri. Lokalnih promenljivih nema. Postoji i jedna privremena promenljiva koja čuva vrednost od n-1, ali nema potrebe, da je čuvamo u oblasti podataka. Postoje tri tačke mogućeg povratka iz rekurzivnog poziva: pozivni program i dve tačke u programu iz rekurzivnih poziva.

Dakle, trebaće nam tri labele:

label 1, 2, 3;

Adresa povratka biće broj (1, 2 ili 3) unutar svake oblasti podataka.

Nakon ovih kraćih razmatranja već smo dovoljno spremni da napišemo iterativnu simulaciju procedure sa slike 1. Ovaj direktni iterativni analogon dat je na slici 2.

Pogađate već — nećemo se zadovoljiti ovim rešenjem. Ista ono jeste nerekurzivno, ali nije previše elegantno. Ima tu još dosta elemenata za pojednostavljanje.

Kao prvo, primetimo da smo koristili tri adrese povratka: po jednu za svaki od dva rekurzivna poziva i jednu za povratka u glavni program. Međutim, povratci u glavni program može se detektovati i na drugi način — kao underflow u steku, baš kao u drugoj verziji programa faktorijel iz „Računara 15“. Ostaju nam još dve labele povratka. Kada bismo uspešli da eliminišemo još jednu labelu, više ne bismo morali da pamtimo adresu povratka na steku. Pozabavimo se drugim rekurzivnim pozivom:

```

PROCEDURE sintovers(n: Posint; frompeg, topeg, auxpeg: char);
LABEL 1,2,3,10;
CONST Maxstack=50;
TYPE Dataarea= RECORD
  nparam: Posint;
  fromparam: char;
  toparam: char;
  auxparam: char;
  return: 1..3
END;
stack= RECORD
  top: 0..maxstack;
  item: array[1..maxstack] of Dataarea
END;
VAR s: stack;
currarea: Dataarea;
i: 1..3;
begin
  (* inicijalizacija *)
  s.top:=0;
  currarea.nparam:=n;
  currarea.fromparam:=frompeg;
  currarea.toparam:=topeg;
  currarea.auxparam:=auxpeg;
  currarea.return:=1;
  (* Pakovanje listovih elemenata na stek *)
  Push(s,currarea);
  (* setovanje parametara i adrese Povratka tekuće oblasti Podataka na odgovarajuću vrednost *)
  currarea.nparam:=n;
  currarea.fromparam:=frompeg;
  currarea.toparam:=topeg;
  currarea.auxparam:=auxpeg;
  currarea.return:=1;
  10: (* Početak simulirane rutine *)
  IF currarea.nparam=1
  THEN
    WITH currarea DO
      BEGIN
        writeln('Prenesi Disk 1 sa stapa',fromparam,
          'na stap',toparam);
        i:=return;
        PopSub(s,currarea);
        CASE i OF
          1: GOTO 1;
          2: GOTO 2;
          3: GOTO 3
        END; (* case *)
        END; (* then with *)
        (* Prvi rekurzivni poziv *)
        Push(s,currarea);
        currarea.nparam:=item(s.top).nparam-1;
        currarea.fromparam:=item(s.top).fromparam;
        currarea.toparam:=item(s.top).toparam;
        currarea.auxparam:=item(s.top).auxparam;
        currarea.return:=2;
        GOTO 10;
      2: (* Tačka Povratka iz prvog rekurzivnog poziva *)
        writeln('Prenesi Disk',currarea.nparam,'sa stapa',
          currarea.fromparam,'na stap',currarea.toparam);
        (* Drugi rekurzivni poziv *)
        Push(s,currarea);
        currarea.nparam:=item(s.top).nparam-1;
        currarea.fromparam:=item(s.top).auxparam;
        currarea.toparam:=item(s.top).toparam;
        currarea.auxparam:=item(s.top).fromparam;
        currarea.return:=3;
        GOTO 10;
      3: (* Tačka povratka iz drugog rekurzivnog poziva *)
        i:=currarea.return;
        PopSub(s,currarea);
        CASE i OF
          1: GOTO 1;
          2: GOTO 2;
          3: GOTO 3
        END; (* case *)
      1: (* Povratka u glavni Program *)
        END; (* sintovers *)
  
```

slika 2:

towers (n-1, auxpeg, topeg, (2) Postavljamo parametre u novoj tekućoj oblasti podataka, a2, na odgovarajuću vrednost: n-1, auxpeg, topeg, frompeg, (3) Postavljamo adresu povratka u tekućoj oblasti podataka, a2, na adresu naredbe odmah iza poziva.

Ovaj poziv simuliramo kao: (1) Pakujemo tekuću oblast podataka, a1, na stek.

```

PROCEDURE towers (n: Posint; frompeg, topeg, auxpeg: char);
BEGIN
  IF n=1
  THEN writeln('Prenesi Disk 1 sa stapa',frompeg,'na stap',topeg)
  ELSE
    BEGIN
      towers(n-1,frompeg,auxpeg,topeg);
      writeln('Prenesi Disk',n,'sa stapa',frompeg,'na stap',topeg);
      towers(n-1,auxpeg,topeg,frompeg);
    END; (* else begin *)
  END; (* towers *)
  
```

slika 1

U Računarima 15 objasnili smo kako simulirati tehniku rekurzije. Priču smo potkrepili školskim primerom (funkcija faktoriel) koji ste sigurno i ranije, bez našeg mudrovanja, mogli lako da napišete u iterativnom obliku.

Ovog puta simulacijom ćemo rešiti nerekurzivno zadatka čije direktno iterativno rešavanje ne ide tako glatko.

```

PROCEDURE sintouvers(n: Posint; fromPe9, toPe9, auxPe9: char);
LABEL 1,2,10;
CONST maxstack:=50;
TYPE Dataarea= RECORD
  nParam: Posint;
  fromParam: char;
  toParam: char;
  auxParam: char;
END;
stack:= RECORD
  top: 0..maxstack;
  item: array[1..maxstack] of Dataarea;
END;
VAR s: stack;
currarea: Dataarea;
un: Boolean;
temp: char;
BEGIN
  (* Inicijalizacija *)
  s.top:=0;
  currarea.nParam:=n;
  currarea.fromParam:=fromPe9;
  currarea.toParam:=toPe9;
  currarea.auxParam:=auxPe9;
10: (* Početak simulirane rutine *)
  IF currarea.nParam=1
  THEN
    WITH currarea DO
      BEGIN
        writeln('Prenesi Disk 1 sa stapa',fromParam,
          ' na stap',toParam);
        (* Simulacija povratka *)
        Popandtest(s,currarea,un);
        IF un
        THEN GOTO 1 (* Povratak u glavni Program *)
        ELSE GOTO 2 (* skok na tacku iza rek. Poziva *)
      END (* Ihan with *)
    (* Prvi rekurzivni poziv *)
    Push(s,currarea);
    currarea.nParam:=s.item[s.top].nParam-1;
    currarea.fromParam:=s.item[s.top].fromParam;
    currarea.toParam:=s.item[s.top].auxParam;
    currarea.auxParam:=s.item[s.top].toParam;
    GOTO 10;
  2: (* Ovo se vraćano iz prvog rekurzivnog poziva *)
    writeln('Prenesi Disk', currarea.nParam,' sa stapa',
      currarea.fromParam,' na stap',currarea.toParam);
  (* Drugi rekurzivni poziv *)
  currarea.nParam:=currarea.nParam-1;
  temp:=currarea.fromParam;
  currarea.fromParam:=currarea.auxParam;
  currarea.auxParam:=temp;
  GOTO 10;
END (* sintouvers *);

```

slika 3:

(4) skok na početak simulirane rutine.

Nakon što se simulirana rutina izvrši, spremna je za povratak. Akcije idu sledećim redom:

(5) Sačuva se adresa povratka, i, iz tekuće oblasti podataka, a2.

(6) Uzima se gornji element sa steka i dodaje se tekućoj oblasti podataka, a1.

(7) Skok na 1.

oblasti podataka a1, jer je ona odmah uništena uzimanjem novog elementa sa steka. Dakle, nemamo razloga da pri simulaciji poziva tu oblast podataka uopšte stavljamo na stek.

Poziv, dakle, možemo da simuliramo jednostavno kao: (1) Promena parametra u tekućoj oblasti podataka na odgovarajuće vrednosti.

(2) Skok na početak simulirane rutine.

Pri povratku iz simulirane rutine možemo se vratiti direktno u rutinu koja poziva našu proceduru, jer nema koristi od toga da to radimo u dva uzastopna „poluskoka“.

Ostala nam je još samo jedna moguća adresa povratka, pa

```

PROCEDURE sintouvers(n: Posint; fromPe9, toPe9, auxPe9: char);
CONST maxstack:=50;
TYPE Dataarea= RECORD
  nParam: Posint;
  fromParam: char;
  toParam: char;
  auxParam: char;
END;
stack:= RECORD
  top: 0..maxstack;
  item: array[1..maxstack] of Dataarea;
END;
VAR s: stack;
currarea: Dataarea;
un: Boolean;
temp: char;
BEGIN
  (* Inicijalizacija *)
  s.top:=0;
  currarea.nParam:=n;
  currarea.fromParam:=fromPe9;
  currarea.toParam:=toPe9;
  currarea.auxParam:=auxPe9;
un:=false;
REPEAT
  WHILE currarea.nParam<1 DO
    WITH currarea DO
      BEGIN
        Push(s,currarea);
        nParam:=s.item[s.top].nParam-1;
        fromParam:=s.item[s.top].fromParam;
        toParam:=s.item[s.top].auxParam;
        auxParam:=s.item[s.top].toParam;
      END (* Do begin *)
    writeln('Prenesi Disk 1 sa stapa', currarea.fromParam,
      ' na stap',currarea.toParam);
    Popandtest(s,currarea,un);
    IF NOT un
    THEN
      WITH currarea DO
        BEGIN
          writeln('Prenesi Disk',nParam,' sa stapa',fromParam,
            ' na stap',toParam);
          temp:=fromParam;
          fromParam:=auxParam;
          auxParam:=temp;
        END (* with *)
      UNTIL un
    END (* sintouvers *);

```

slika 4:

je ne moramo spremati i naknadno uzimati sa steka sa ostalim podacima iz oblasti podataka. Kad god uspešno izvršimo uzimanje elementa sa steka, pravimo skok na adresu naredbe neposredno iza prvog poziva. Ako pokušaj uzimanja sa steka javi underflow, skoćemo u pozivni program. Kako se nova vrednost promenljivih, u tekućoj oblasti podataka dobija iz stare oblasti podataka, moraćemo da deklariramo dodatnu promenljivu temp koja će privremeno čuvati vrednost stare oblasti podataka.

Popravljena nerekurzivna verzija hanojskih kula data je na slici 3.

Ni ovo nije savršena verzija. Ako se malo pozabavimo strukturom programa, lako ćemo ga reorganizovati. Jednostavniji format hanojskih kula dat je na slici 4.

Ovo je sasvim pristojan oblik iterativne procedure. Znači, „kupliti“ smo rekurzivno rešenje problema, „mehanički“ ga preveli u nerekurzivni oblik i, na kraju, to rešenje je u početku zvučalo preteško — imamo nerekurzivno rešenje zadatka ako smo bezjkojupci lada ćemo ga prepevati na bejzik. Čak i ako i dalje ostanemo u paskalu, dobili smo rešenje koje se brže izvršava od rekurzivnog analogona.

Zoran Obradović

Programiranje na mašincu

Linkovi na „spektrumu 128“

Mada to ne liči na njega, Sinkler je uveo u ROM „Spektruma 128“ dosta linkova za proširenja (iako se autoru čini da je on to uradio zato što nije moglo drugačije, a ne da nama olakša život). Naime, veza između dve stranice u ROM-u odvija se preko izvjesnih linkova u RAM-u. Prije nego što nastavite dalje čitati tekst, preporučujem vam da pročitate tekst u prošlom broju „Računara“, jer vam, u suprotnom, ono o čemu se ovdje govori neće biti najjasnije.

Prijavljujem grešku!

Kako definisati nove naredbe na „spektrumu 128“. Prvo što je potrebno, to je da se upozna mehanizam prijave greške. U ROM-u 1 greška se prijavljuje, kao što svi znaju, pomoću

RST 8

DEFB kod greške,

dok se u ROM-u 0 greške prijavljuju pomoću

CALL 1452

DEFB kod greške

Dakle, u ROM-u 0 prijave greške nije dobila svoj restart, ali to ništa ne mijenja stvar. Princip prijave greške je isti i svodi se na

LD SP, (23613)

RET

Ali, dok je kod 48-ice varijabla ERR-SP (23613) ukazivala na adresu 1303 heksadekadno ili 4867 dekadno, kod 128-ice ukazuje na adresu 23325! To je u RAM-u, zar ne? Baš lijepo! Upravo ta mogućnost nam dopušta da radimo šta hoćemo. Tamo se nalazi sljedeći mašinc:

```
DI
LD A, (23388)
AND 0111%1111
LD (23388), A
LD BC, 32765
OUT (C), A
EI
JP 195
```

Ko je čitao dosadašnje tekstove o „spektrumu 128“ trebalo bi da shvati šta radi ova rutina. Ona, prosto, uključuje ROM 0 i vrši skok na adresu 195 (e, ovo je već ROM). Tamo se nalazi ova vrlo kratka rutina:

```
LD HL, (23435)
JP (HL)
```

Vi ćete pomisliti: sve je jasno, samo treba na adresu 23435 smjestiti neku vrijednost i kontrola prijave greške će biti usmjerena na našu rutinu. Ali ne, nije tako! Ova sistemka varijabla ima „vržnu osobinu“ da se mijenja u toku rada. Drugim riječima, za vrijeme izvršavanja programa (run-time) ona ima vrijednost 801, dok je za vrijeme sintaksne provjere tamo neka druga vrijednost, najčešće 698, ali ne uvijek. Na taj način kompjuter razlikuje sintaksne od run-time grešaka. Ova sistemka varijabla može „lepo da posluži za run-time rutine tipa ON BREAK GO TO, ON ERROR GO TO itd. Ali nama ona ne odgovara. Zato ćemo uraditi nešto drugo: promijeniti instrukciju JP 195 u JP naša rutinu. Jednostavno, mora se priznati!

Sa teorije na praksu

Na jednostavnom primjeru ćemo pokazati kako to izgleda u praksi. Prije toga, riječ-dvije o ROM-u 0. Restarti RST 16, RST 24 i RST 32 su identični onima u ROM-u 1, preciznije, oni pozivaju istomere rutine u ROM-u 1. Restarti RST 8 i RST 48 ne postoje, a šta rade RST 0 i RST 56 to se zna (pogotovo RST 0). Zaboravne podsjećamo da se RST 40 koristi za pozivanje rutine u ROM-u 1 i to u obliku

```
RST 40
DEFW adresa rutine
```

Sinklerova veza

Nove sistemske promenljive

U „Računaru“ 19 spomenuli smo neke nove sistemske varijable. Ovdje navodimo kompletan spisak sistemskih varijabli sa potrebnim objašnjenjima.

DUŽINA	ADRESA	IME	OPIS
N2	23384	HL-SAV	Služ za skidanje HL registra u slučajevima kada primjena steka nije poželjna
X2	23388	RET-ADD	Adresa povratka u RM 0 (vidi tekst)
X1	23388	MEM-ST	Sadrži status memorije. O ovoj varijabli bilo je dosta riječi u prošlom broju
X2	23389	E-HANNLE	Ove dvije lokacije sadrže redom 267 (RST 8) i kod greške. Koriste se kod RM 0 prijavljuje grešku iz RM-a 1. Za nas je ova varijabla potpuno neupotrebljiva.
2	23391	BAUD	Brzina konstanta za RS 232 operacije. Mijenja je naredba FORMAT
2	23393	SER-FLG	Statusna varijabla za RS INPUT operacije
N2	23395	RS-COUNT	Brojač praznih karaktera prilikom ispisivanja kontrolnih kodova 6,22 i 23 na RS 232 izlaz.
1	23397	RS-CTRL	Broj parametara kontrolnih kodova
X1	23398	T-FLAG	Razni indikatori za operacije sa RAM diskom. Ova varijabla može imati primjenu kod pozivanja RAM disk rutina iz mašinc.
N10	23399	NAME	Ime zapisa koji se poziva iz RAM diska
N16	23409	HEADERS	Radni prostor za operacije sa RAM diskom. Takođe se koristi i prilikom naredbi RENUMBER i FORMAT. Ako se pomenute naredbe ne koriste, ovdje se mogu smjestiti razni podaci.
N2	23425	SP-SAV	Čuva staru vrijednost SP registra prije njegove promjene.
X2	23427	LAST-FILE	Ukazuje na zaglavlje poslednje datoteke u RAM disku (vidi „Računare“ br. 19)
X3	23429	FREE-DISC	Broj bajtova slobodne memorije u RAM disku (PEEK 23429+PEEK 2343+256+256PEEK 23431)(5836) NAPOMENA: Zbog бага u ROM-u, ova sistemka varijabla totalno „pouzd“ ukoliko zapišete datoteku dužu od 65527 bajtova (a kome treba tolika datoteka???)
3	23432	PAD-SCAN	Koristi se za očitavanje keypad-a.
X2	23435	ERP-PTR	Pionir rutine za prijavu greške
N5	23437	RES-CALC	Rezultat poslednje operacije u kalkulatoru.
N2	23442	REN-PTR	Ukazuje na liniju čija je prenumeralica u toku.
2	23444	REN-LINE	Početna linija za RENUMBER
2	23446	REN-STEP	„STEP“ za RENUMBER
N16	23448	GRAPH-CHR	Ovdje se nalaze definicije 2 grafička karaktera koji služe za iscrtavanje menija.
N88	23464	STACK-BUFF	Služ za smještanje mašinskog steka za vrijeme operacija koje preklapaju gornjih 16K memorije (editor, LOAD! itd.)
	23552-23733		Stare sistemske promenljive

Oznaka N ispred dužine varijable znači da pokiranje te promenljive neće izazvati nikakav efekat, a oznaka X upozorava da sistem može da krahira ukoliko se u varijabli smjeste nedozvoljene vrijednosti.

Imena sistemskih varijabli su plod pjesničke mašte autora te se nemoguće čuditi ukoliko u nekom stranom časopisu naiđete na nešto sasvim drugo. Međutim, bar koliko je autoru poznato, kompletan spisak sistemskih varijabli nije dosada nigdje objavljen.

Postoje i neke sistemske promenljive koje se nalaze izmjenjane sa radnim prostorom editora u ROM-u 7. Međutim, više od pola pomenutih varijabli ima namjenu u stilu „trebam-negdje-da-smjestim-podatke-a-stek-mi-je-nepodesan“, pa ih ovdje nećemo opisivati. Ostale služe za opitavanje raznih stanja pri radu editora, ali ni od njih nema neke velike koristi. Autor smatra da su najkorisnije od njih 60433 i 60431, koje su opisane u prošlom broju.

Svaki haker koji je bar malo upućen u tajne mašinskog programiranja zna da je definisanje novih naredbi na „spektrumu 48“ veoma mukotrpan ali ipak moguć posao. Naime, Sinkler nije smatrao da će vlasnicima 48-ice ikada „zatrebati“ tako nešto. Na svu sreću, i to se može nekako ostvariti (IM 2, ERR-SP), ali prvi problem nastaje kada treba izmijeniti neku već postojeću naredbu (npr. LOAD). U tom slučaju je jedini lijek bio pisanje potpuno novog bezijk interpretera. „Spektrum 128“ predstavlja, srećom, sasvim drugu priču.

Pomoću rutine za prijavu greške u ROM-u 0 (adresa 1452) mogu se prijavljivati i nove greške. One imaju kodove 28-43 i idu slijedećim redom: a MERGE error, b Wrong type file, c CODE error, d Too many brackets, e File already exist, f Invalid name, g File does not exist, h File does not exist, i Invalid device, j Invalid baud rate, k Invalid note name, l Number too big, m Note out of range, n Out of range, o Too many tied notes. Autoru je nejasno kako se desilo da postoje dvije iste greške (kodovi g i h), i da postoje i „nova-stara“ greška (kod 1).

No, ostavimo Sinklerove bisere po strani i vratimo se našem zadatku. Recimo da želimo definisati naredbu JOIN koju „spektrum 128“ ne poznaje, a vrlo je korisna. Naime, ona spaja dvije bezijk linije u jednu, što je normalnim putem teško ostvariti.

Počnimo. Ukoliko želimo da se naša naredba (recimo JOIN) može koristiti i u komandnom i u programskom modu, ona mora početi nekim znakom koji nije slovo. U suprotnom, pošto u komandnom modu PRINT naredba nije potrebna, postoji opasnost da naša naredba bude shvaćena kao varijabla (probajte na prazno otkucati JOIN 10 i dobićete „2 Variable not found“). Zato ćemo odlučiti da se naša naredba zove recimo! JOIN ili još bolje samo IJ (lakše je za kucanje kad ima manje slova). Rutinu ćemo smjestiti na adresu 65000 i ona izgleda ovako:

```
ORG 65000      početak na 65000
LD A, (23610)  uzima kod greške
CP 11         da li je Nonsense in BASIC?
JP NZ, 195    skače u ROM ako nije
RST 24       uzima karakter
DEC HL       na kom je
LD (23645), HL nastupila
LD A, (HL)   greška
CP „I“       da li je to „I“?
JP NZ, 195   ako nije, skače u ROM
RST 32       uzima slijedeći karakter
AND 223     pretvara malo slovo u veliko
CP „J“       da li je to „J“?
JP NZ, 195   skok u ROM ako nije
```

Ovim je završen prvi dio našeg posla. Nakon što je detektovana naša naredba, potrebno je na stek postaviti adresu za prijavu greške (23325) i povratnu adresu u bezijk interpreter (to je rutina STMT-RET). U starom bezijk interpreteru to je bila adresa 7030, a u novom je 6117. Dakle, slijedi:

```
LD HL, 23325  na stek
PUSH HL      adresa za prijavu greške
LD HL, 6117  na stek povratna adresa
PUSH HL      u bezijk interpreter (novi)
```

Dalje, naša naredba ima jedan numerički parametar (to će biti broj linije koja se spaja sa slijedećom). Njega ćemo pročitati pomoću rutine EXPT-1NUM (adresa 7298) u ROM-u 1 koja „skida“ jedan numerički parametar (pogledajte SPECTRUM ROM DISASSEMBLY od 48-ice ako ga imate). Dakle:

```
RST 32       uzima slijedeći znak
RST 40       i poziva rutinu
DEFW 7298    EXPT-1NUM iz ROM-a 1
```

Pošto nema više parametara, treba provjeriti kraj. Slijedi:

```
RST 24       uzima karakter
CP 13        da li je to ENTER?
JR Z, DALJE  ako jeste, skok na labelu DALJE
CP „“       da li je to „“?
JR Z, DALJE  u redu je ako jeste
CALL 1452   u suprotnom
DEFB 11     slijedi greška
```

Ako je sve u redu, treba signalizirati da nema greške, i ukoliko je bila u toku sintaksna provjera, posao je gotov. Dakle u nastavku slijedi

Kako prepraviti mons

Jedna od zanimljivih osobina „spektruma 128“ je i ta da se isti ROM koristi i iz 48 i iz 128 moda (jedino je u 48 modu pola ROM-a isključeno) i zato veliki broj programa radi u oba moda. Čak se i čuveni GENS za 48K „spektrum“ može koristiti i iz moda 128.

Međutim, sa MONS-om to nije slučaj. Ovaj program radi sasvim lijepo u 48 bezijk modu, ali u 128 modu ima problema sa povratkom u bezijk. No, dovoljno je par „pokova“ pa da se i to riješi. Uključajte slijedeći program:

```
10 LOAD „MONS3M“ CODE 30000,5800
20 POKE 30069,3: POKE 30070,19
30 SAVE „MONS 128“ CODE 30000,5800
```

Nakon ovoga, MONS možete bez problema koristiti i iz bezjika 128 ukoliko je to potrebno (npr. da izlistate neki mašinsac na RS 232 izlaz).

```
DALJE LD      signal „nema greške“
(IY+0), 255   da li je bila u toku sintaksna provjera?
BIT 7, (IY+1) kraj ako jeste.
RET Z
```

Ukoliko nije bila u fazi provjera sintakse nego izvršenje (run-time), treba samo izvršiti naredbu, i to je sve. Pošto bi JOIN rutinu trebalo da zna napisati svaki haker koji čita ovaj tekst, ovdje je dajemo bez objašnjenja:

```
JOIN RST 40
DEFW 7833   ovo je rutina FIND-INT2
LA A, B
CP 64
RET NC
LD H, B
LD L, C
RST 40
DEFW 6510  rutina LINE-ADDR
RET NZ
RST 40     rutina NEXT-ONE
DEFW 6584  PUSH HL
LD A, (DE) PUSH BC
CP 64      LD C, (HL)
RET NC     INC HL
EX DE, HL LD B, (HL)
DEC HL    POP HL
LD (HL), „“ ADD HL, BC
INC HL    EX DE, HL
PUSH HL  POP HL
INC HL   LD (HL), E
INC HL   INC HL
LD C, (HL) LD (HL), D
INC HL   POP HL
LD B, (HL) LD BC, 4
EX DE, HL RST 40
INC HL   DEFW 6632  rutina RECLAIM-2
INC HL   RET
```

Dakle, proces definisanja novih naredbi na „spektrumu 128“ donekle je sličan procesu definisanja naredbi na 48-ici, s tom razlikom da nema mučnih priprema oko IM 2, ERR-SP i sličnih „gluposti“. Jedino se ne smije smesti sa uma da je cijelo vrijeme dok traje proces definisanja novih naredbi uključeni ROM 0.

Nova naredba je potpuno ravnopravna sa ostalim naredbama i može se koristiti bilo u programskom, bilo u komandnom modu. Npr. da spojite liniju 10 sa slijedećom linijom samo otkucajte

Ukoliko linija 10, ne postoji ništa se neće desiti.

Umalno da ne zaboravimo, Da bi interpreter uopšte mogao da prepozna novu naredbu, treba „prepokirati“ JP 195 u JP 65000, tj. izvršiti:

```
POKE 23341, 65000-256-INT(65000/256);
POKE 23342, INT(65000/256)
```

Tajna restarta 40

Činjenica da instrukcija RST 40 „petlja“ sa ROM-om daje nam mogućnost da utičemo na njeno izvršenje, a samim tim i na izvršenje mnogih već postojećih naredbi i rutina. Ovaj restart izgleda ovako:

```
EX (SP), HL      na stek HL, u HL adresa iza RST 40
PUSH AF          sklanja AF
LD A, (HL)       uzima bajt iza RST 40
INC HL           uvećava HL za 2
INC HL           i smješta rezultat u
LD (23386), HL   sistemsku varijablu RET-ADD
DEC HL           u HL
LD H, (HL)       u HL registru se sada nalazi
LD L, A          adresa koja se poziva preko RST 40
POP AF           vraća AF
LD (23384), HL   privremeno sklanja HL
LD HL, 23316     na stek ide 23316, a u HL
EX (SP), HL      se vraća stara vrijednost
PUSH HL          stavlja HL na stek
LD HL, (23384)   vraća HL (u njemu je adresa rutine)
EX (SP), HL      izvršna adresa ide na stek, a u HL stara vrijebila
JP 23296         skok u RAM.
```

Ova rutina, verovatno, izgleda dosta komplikovano zbog neograničene operacija sa stekom, ali, u suštini, ona ne radi ništa „strašno“. Ona samo na stek dovodi adresu 23316 i adresu rutine koju pozivamo preko RST 40, a adresu povratka u ROM 0 smješta u RET-ADD (23386). Pri tom vodi računa da ne „pokvari“ nijedan registar. I to je sve.

Nakon toga slijedi skok u RAM na adresu 23296. Rutina koja se nalazi tamo je krajnje jednostavna i njena uloga je da preklopi ROM. Rutina izgleda ovako:

```
PUSH AF          sklanja AF
PUSH BC          sklanja BC
LD BC, 32765     ovo je kontrolni port za stanje memorije
LD A, (23388)    čita stanje memorije
XOR %00010000   „izvrće“ bit za ROM
DI              isključuje interapt
LD (23388), A   pamti novo stanje memorije
OUT (C), A      i preklapa ROM
EI              uključuje interapt
POP BC          vraća BC
POP AF          vraća AF
RET             kraj.
```

Prilikom nailaska na RET vrši se skok na rutinu koja je pozvana sa RST 40 (ne zaboravite šta je na steku). Nakon silnih peripetija, tražena rutina je, najzad, pozvana. A šta se dešava posle?

Kada se pozvana rutina završi, RET instrukcija na njenom kraju izaziva skok u RAM na adresu 23316 (opet zasluga steka). Mašinska rutina koja se tamo nalazi vrlo je kratka:

```
CALL 23296      preklopi ROM
PUSH HL        sačuvaj HL
LD HL, (23386) uzmi povratnu adresu
EX (SP), HL    stavi povratnu adresu na stek i vrati HL
RET            skoči nazad na mjesto odakle je rutina
               bila pozvana.
```

Ovim je tajna restarta 40 objašnjena.

A ŠTA SAD???

Svi koji su shvatili kako radi pomenuti restart već imaju predstavu kako se on može iskoristiti. Mogućnosti su zaista ogromne, a ovdje ćemo dati jedan primjer.

Pretpostavimo da želimo da napravimo novu POKE rutinu koja će prihvatiti i adrese veće od 65535 i recimo da proglasimo da su adrese od 65536-147455 adrese RAM diska. To nije prevelik problem. Pomoću te rutine možemo zamisliti da imamo 16-bitni procesor (opa!) i zaboraviti da nam je memorija organizovana „po blokovima“. No, predimo na stvar.

Novi bezjzik interpreter se nalazi u ROM-u 0 (za razodnalice to je adresa 6063), a POKE rutina u ROM-u 1 na adresi 1E80 HEX (vidi ROM DISASSEMBLY). Logično je da bezjzik interpreter NEGDJE (gdje, to uopšte nije važno) poziva POKE rutinu u ROM-u 1 pomoću

```
RST 40
DEFW 1E80 HEX
```

Priča počinje ovdje. Na adresu 23296 ćemo smjestiti JP 40000 (može i neka druga adresa, ali zbog prilode rutine ta adresa bi trebala da bude ispod 49152). Na toj adresi prvo treba uraditi sve ono što je radila rutina na 23296, dakle:

```
ORG 40000      DI
PUSH AF        LD (23388), A
PUSH BC        OUT (C), A
LD BC, 32765   EI
LD A, (23388)  POP BC
XOR 16         POP AF
```

Dalje, potrebno je „pogledati“ koja je rutina pozvana. No, moramo dobro paziti da ne „umrljamo“ ni jedan registar (pa čak ni F koji ima ružnu osobinu da se kvari i kad treba i kad ne treba). Ukoliko nije pozvana POKE rutina, ne treba da se desi ništa specijalno. Međutim, ukoliko je pozvana POKE rutina, treba da uslijedi skok na našu rutinu. To sve izgleda ovako:

```
EX (SP), HL    sklanja HL i uzmi u HL
               adresu sa steka
PUSH AF        sklanja AF
LD A, H        uzima viši bajt
CP #1E         da li je to #1E?
JR NZ, NO-POKE ako nije, skok na NO-POKE
LD A, L        uzima niži bajt
CP #80         da li je to #80?
JR Z, POKE    ako jeste, skok na POKE
POP AF        vraća AF
EX (SP), HL   vraća HL i adresu skoka
RET           skače na pozvanu rutinu
POKE         vraća AF
POP AF        vraća AF
POP HL        uzima HL i, ujedno, poništava
               adresu skoka
```

Dalje treba da slijedi nova POKE rutina. Pošto nam ovdje nije bio cilj da objašnjavamo kako napraviti novu POKE rutinu, nego kako izvršiti postojeću naredbu, ovu rutinu vam ostavljamo za vježbu.

Kao što vidite, izmjenjena rutina u ROM-u je vrlo jednostavna. Krajnji efekat zavisi samo od mašte hakera, Dakle, GENS u ruke (ako imate MAGUS još bolje) i na posao.

Umjesto epiloga

No, mora se reći da postoji i jedan problem. ROM 0 ne poziva baš svaku rutinu iz ROM-a 1. U prošlom broju je spomenuto na koje se naredbe može u potpunosti djelovati na ovaj način. Što se ostalih naredbi tiče, njihove izvršne rutine se nalaze u ROM-u 0, zajedno sa novim bezjzik interpreterom. To je urađeno zbog toga što su neke rutine pretrpile izvjesne izmjene. Ipak, i te rutine pozivaju neke podrutine u ROM-u 1, tako da je i na njih moguće donekle uticati. Npr. rutina LOAD, iako je smještena u ROM 0 (zbog opcije „I“), ona i dalje poziva rutinu LD-BYTES (1366) u ROM-u 1. Jedino se ne može uticati na rutine koje su u potpunosti u ROM-u 0 (na žalost, tamo spadaju PLAY, COPY i još neke zanimljivije rutine).

Doduše, postoji način da se i to sredi. Na primjer, rutina PLAY ipak poziva nešto iz ROM-a 1 (to je rutina za skidanje alfanumeričkih parametara), te nam i to otvara mogućnost „petljanja“. No, o tome drugom prilikom. Uostalom, možete i sami pregledati rutine bezjzik interpretera koje su prilično „čitljive“. U svakom slučaju, slabo dokumentovan računar kao što je „spektrum 128“, daje nam idealnu priliku za „čepkanje“.

I još nešto za kraj. ROM 1 je lako „ispikirati“ iz bezjzika, ali šta je sa ROM-om 0? Ako se još nije sjetili kako da „ispikirate“ ROM-0, najbolje to možete uraditi ovako:

```
SAVE I,,rom 0" CODE 0,16384:
LOAD I,,rom 0" CODE 32768:
ERASE I,,rom 0"
```

Ovim je ROM 0 prebačen u RAM na adresu 32768 i možete ga gledati do mile volje. Šta čekate!?

algoritam „brzi gonzales“

Po ko zna koji put se potvrdila činjenica da imamo vrane pronalazače i stručnjake na svim poljima tehničke djelatnosti, ali i da još uvijek imamo „tvrd“ siuh kada je u pitanju svestrana podrška pronalazača. Kako drugačije komentirati riječi dipl. ing. Željka Margitića koji kaže: „Radim na sistemskom softveru preko 14 godina. Međutim, više od 10 godina morao sam raditi u vlastitoj režiji, jer mi moja radna okolina nije pružala podršku u radu niti je za njega imala sluha.“

— Još daleke 1975. sam imao gotov sort koji je radio dvostruko brže od Singlerovog. Naravno, sve je to komijalski potvrđeno i dokazano. Godinama sam krpao tuđe „mozgarije“, te sam 1983. odlučio razviti vlastiti sistem. Ispostavilo se da je ono što sam napravio na Singlerovom SISTEMU 10 daleko bolje od postojećih rješenja. Najbolji rezultati se postiču kod random pristupa na datotekama i to zahvaljujući novoj DIRIN metodi, koju nema još ni jedan današnji računar, jer sam napravio algoritam za pristup disk adresama duljine svega 5 karaktera. Ova metoda je brža od bilo koje random metode do sada, a glavna datoteka ostaje uređena sa maksimalnom uređenosti prostora na disku. Na osnovi DIRIN metode učitavaju se kompletni DIRIN indeksi u memoriju koji se sastoji od 4 bajta disk adrese za MC 68000 — 15 bitni mikropcesor (izbor izložio doc. dr. Andrija Maričić sa ETF-a i 1 bajt za status zauzetosti fleksibilne baze podataka.

Fleksibilna baza podataka

Što podrazumijevate pod FLEKSIBILNOM bazom podataka?

— Da bismo objasnili prirodnu fleksibilnu bazu podataka, uzeti ćemo njezin način funkcioniranja na obradi salda kontija kupaca i dobitivača. Npr. poduzeća s oko 2000 ljudi imaju registar kupaca s oko 30.000 slogova. Ako uzmemo algoritam gustine 4, na jednu stranicu će prosječno otpadati tri sloga datoteka registar. Ako uzmemo dirin indekse u memoriju, oni će zauzeti oko 60 K. DATA BASE „MATE“ mašina će moći vrtiti u memoriji oko 100 programa. Konceptija mašine je multi task, multi proces, multi user... Nema ograničenja da svih 100 programa imaju dostup do registara datoteka. Pristup registar slogovima ide od svih programa preko učitanih dirin indeksa u memoriji. Program koji traži neku šifru „zalokuje“ cijelu stranicu registra (4 kupca) i drugi program nema dostup do te stranice sve dotle dok prvi program ne obavi posao na toj stranici (transakcija).



Od nemila do nedraga: Dipl. ing. Željko Margitić

Za to vrijeme drugi program radi isto posao na nekoj drugoj stranici, a n-ti program na n-toj stranici. Algoritam kinematike diska (razradio ing. Margitić o.p.) vodi računa o tome da se posao što prije obavi na cijelom računaru.

Što se, zapravo, događa u računaru?

— Kada jedan program, zatraži pristup do nekog kupca pomoću recept algoritma, koji je za svaku datoteku fleksibilno napravljen uz vođenje računa o analizi karakterističkih šifre povezano s dva algoritma u funkciji dva nivoa sabijanja prostora, disk adrese početka stranice se nađu u memoriji procesora. Nakon toga slijedi direktan akok na traženu stranicu sa slogovima bez prethodnog pretraživanja na disku (to do sada nikom nije postigao; vrijeme IOL-a na SISTEMU 10 računaru je 2 minuta i 25 sekundi, a dirin algoritma 29 sekundi, što je oko 5 puta brže o.p.). „MATE“ računar će biti prva prava data base mašina u svijetu sa stanovitvima da data base obrade budu ubrzan na svim nivolima upotrebe — obrat obrade, upit, telekomunikacije itd. Već kod dizajniranja arhitekture računara treba o svemu govome voditi računa, kao i o „C kompajleru“ koji prati cijelu arhitekturu računara, a koji je daleko moćniji od „C kompajlera“. Zahvaljujući takvoj obradi „MATE“ softvera omogućava se blok random pristup podacima. Algoritam kinematike diska omogućava kupljenju blok slogova podataka u jednom potezu ručice uz optimalizaciju rotacije diska što daje još veće ubrzanje dirin metodi.

Računar otvoreno gtipa

Da, ali što će se desiti ako npr. nestane struje?

— Ukoliko nestane struje „MATE“ računar omogućava restauraciju rada svih 100 programa u kompjutoru i to točno na onim pozicijama gdje je nastao prekid i ako je memorija nakon prekida struje izgubila svaki sadržaj i to bez obzira na vrijeme trajanja prekida. To se do sad nije nikom dosjetio nijeti, a može se omogućiti sistemskom evidencijom poslova i taskova na takav način da se uopće ne oseti usporanje rada mašine pa nisu potrebna nikakva prethodna spašavanja datoteka, zbog nekih osjetljivih ažuriranja.

Na koji način će doći do potpune realizacije projekta?

— Prije izlazenog vremena je došlo do povezivanja OOUR-s „Tera“ (RO „Tehnicar“, „Elektronika“ (RO „Pel“), „Zavoda za ekonomiku“ iz Varaždina i mene, te je osnovan „RIIC“ (razvojno istraživačko informatički centar) takođe sa sjedištem u Varaždinu. U toku su dogovori „Pela“ i „Tera“ oko razvoja elektroničke logike po principu „MATE“ operativnog sistema usavršavajući „Tera 8“ računar koji je baziran na MC 68000 sa ST 506 disk kontrolerima koji u ovom momentu omogućuju priključke diska od 250 MB. „Tera 8“ ima mogućnost da CP sa MC 68000 ima 8 MB memorije. Za CP sa 32 bitnim mikropcesorom MC 68020 bit će predviđena memorija od 16 megabajta. Svaka disk jedinica će imati svoj vlastiti mikropcesor i to prvi vlastiti u našoj zemlji (sve su do sada bile licence o.p.), a isto tako se planira da svi periferali budu s vlastitim mikropcesorima. Teži se ka tome da se projektuje razvoj otvorenog tipa računara čime bi se omogućio saobraćaj svih računarskih centara i drugih poduzeća u zemlji u planiranju JUPAK i ZEPAK mreži. Saobraćaj bi se ostvarivao putem telefonskih linija i to zahvaljujući mogućnosti multij pristupa na sve datoteke. Koncept „MATE“ razvoja ide za povezivanje svih svojih korisnika i za međusobno koordiniranje i pruzimanje programa, čime bi se povećala produktivnost izrade i obrade aplikacija u cijeloj zemlji, te će se u potpunosti moći obraditi razvoj transputora, ojačati pratebene sposobnosti JNA i ubržati tokovi razvoja mnogih privrednih grana.

Deset puta jači, deset puta jeftiniji

Koja je cifra „aktualna“ kad se u pitanju ulaganja?

— Ako se posmatra dobit od jednog takvog sistemskog softvera onda su troškovi ulaganja praktično zanemarivi. Potrebno je oko 15 milijardi starih dinara. U tu cenu su uračunati tri razvojna sistema bazirana na „Tera 8“, poslovni prostor, plaća za 10 razvojnih inženjera i otkup licencnog prava (od ing. Margitića o.p.). Jedna takva gotova mašina bi na tržištu koštala oko 10 milijardi starih dinara s bazom podataka, a već nula serija bi isplatile troškove i donijela zaradu za proširenu reprodukciju. Da kažem i to da je općina Varaždin spremna novčano pomoći čitav projekt jer, a je pozivam sve one koji hoće da sudjeluju u razvoju „RIIC“ — da nam se priključe jer još nije kasno. Nedavno je RO „Rade Končar“ za procesni računar ASEA platilo 3 mil. \$, a „MATE“ računar će biti 10 puta jači i efikasniji od ASEA i „svraga“ 10 puta jeftiniji. Dodajmo tome i da smo prije par dana uspostavili kontakt sa dva vrana makrodska pronalazača, Gruvskim i Nuredinom, te je velika vjerovatnost da algoritam za rješavanje desetog Hilbertovog problema po prvi put na svjetsko tržište bude plasiran putem „MATE“ računara.

Na kraju, naglasimo da su kompjutori samo sredstva za rad, a za dinamičnu informatizaciju modernog društva dominantan je aspekt razvoja softvera. Godišnja proizvodnja softvera u svetu raste za oko 17%. Vrijednost godišnjeg plasmana softvera na svjetsko tržište se kreće između 7 i 12 mrd. \$, a posljednjih godina se javlja novo tržište mikropkompjuterskog softvera koje se za 2000 god. procjenjuje na 20 do 25 mrd. \$. U ovu proizvodnju su se već uključile mnoge zemlje u razvoju, budući da su ulaganja neznatna, a osnovni kapital predstavlja ZNANJE. Po svemu sudeći, svijet će uskoro dobiti veliku konkurenciju iz drugog plana, iz Jugoslavije.

Dragan Vranješević

iz šupljeg u prazno

Znate li onaj vic sa bradom o matematičkom nastrojnom programu koji sprema čaj? Programer, da vas podsetimo, zna kako da spremi čaj: sipa vodu u kotlić, stavi kotlić na šporet, sačeka da voda provri i tako dalje. Šta će programer da uradi ako mu neko da kotlić u koji je već sipana voda? Prosuće vodu i time svesti problem na prethodni slučaj koji ume da reši.

Možemo li da sortiramo podatke sa spoljne memorije tako što ćemo prosuti vodu, tj. učitati ih u osnovni RAM i tamo ih sortirati, koristeći neki od metoda koje smo upoznali pre mesec dana? Ako podataka ima malo, ovakvo vam rešenje toplo preporučujemo. Podataka, međutim, obično ima previše: pošto je kapacitet spoljne memorije obično nekoliko desetina (stotina, hiljada...) puta veći od kapaciteta RAM-a, sve se datoteke ne mogu učitati i tako sortirati. Ostaje, dakle, da upoznamo metode koje omogućavaju sortiranje velikih količina podataka koji su upisani na traku, floppy ili masivni disk.

Sekvencijalne memorije

Magnetna traka je, istorijski posmatrano, prvi medij masovne memorije: nekada je bilo sasvim normalno da (veliki) kompjuter bude opremljen jednim diskom kapaciteta pola megabajta i trima jedinicama magnetne trake; možda čete zbog toga u mnogim naučno-fantastičnim filmovima primetiti bezbroj traka koje se okreću na razne strane.

Jednice magnetne trake nisu nimalo jeftine; uobičajeno je da koštaju više od standardnih jedinica hard diskova! Zbog čega ovolika cena? Jedinice magnetne trake imaju izuzetno preciznu mehaniku (i pneumatiku) koja omogućava da se traka iz brzog premetavanja praktično trenutno zaustavi i tako omogući direktan pristup podacima koji nisu upisani na obe strane. Traka se, osim toga, premetava na obe strane, što znači da, u izvesnoj meri, može da se smatra medijem sa slobodnim pristupom.

Pad cena diskova je učinio da ovakve skupa i super precizne trake izgube na značaju: disk je daleko brži i zgodniji medij na kome je direktni pristup mnogo direktniji. Zašto su onda trake i dalje popularne? Zato što čete, kada kupite (skupu) jedinicu trake, u nju umetati magnetne trake koje su izuzetno jeftine i koje se lako skladište; zauzimaju malo prostora, ne treba im posebna klimatizacija, prašina obično nije baš smrtonosna... Podatke, dakle, obrađujemo držeći ih na disku, a zatim datoteke koje treba skladištiti sekvencijalno prepisujemo na trake; ako nam ovi podaci ikada ponovo zatrebaju, vraćamo ih na disk i tamo ih obrađivati. Traka je, dakle, postala tipično sekvencijalni medij: podaci se upisuju od početka prema kraju. Ukoliko planirate da koristite magnetnu traku na ovaj način, ne treba da bacate novac na kupovi-

nu skupih jedinica: prava su stvar takozvane strimer trake (jedna standardna jedinica magnetne trake obično košta koliko i 3—4 strimer jedinice) koje nemaju precizan mehanizam za zaustavljanje, ali koje omogućavaju brzo sekvencijalno upisivanje i čitanje podataka.

Do sada smo govorili isključivo o velikim kompjuterskim sistemima i to sa razlogom — na njima su se razvijale metode za sortiranje i pretraživanje podataka. Kakva je situacija na „spektrumima“, „komodorima“ i drugim kućnim mašinama? „Mnogo složenija“.

KAKO UČITATI PODATKE

Pripremajući program sa slike 1 na standardnom paskalu surseli smo se sa jednim problemom koji će, verujemo, ponekad pogoditi i vas. O čemu se radi?

Treba da učitamo dve serije podataka, nizove A i B. Podatke upisujemo jedan po jedan i tako dolazimo do kraja niza A. Kako da signaliziramo da u A nema više podataka? Najjednostavnije je da pritisnemo CTRL Z ili neku drugu sličnu kombinaciju tastera i tako stavimo računaru do znanja da je fajl koji smo unosili završen. Promenljiva eof će tada dobiti vrednost true, funkcija reader će se završiti i kontrola će se vratiti glavnom programu.

Nevoja je što sada treba da učitamo i drugi niz. Što se računara tiče, „datoteka“ input koja je dodeljena tastaturi je pročitana — u njoj nema više podataka, tako da sledeće readin jednostavno neće funkcionisati. Šta da se radi? Prirodno bi bilo iskoristiti naredbu rewind kojom se sekvencijalna datoteka koja je (eventualno) isčitana do kraja „premotava“ na početak. Na žalost, standardi ISO Pascala ne određuju kako rewind deluje na „datoteku“ input — u knjizi eksplicitno piše da ovo delovanje može da bude zavisno od implementacije. Autori implementacije koju mi koristimo su sebi otakšali posao, pa su učinili da rewind uopšte ne deluje na datoteku input što je, verujemo, sasvim klasičan slučaj.

Šta da se radi? Ofornili smo pomoćnu datoteku doumnp koja je definisana kao text (ili, što je isto, file of char), a zatim je dodelili tastaturi. Ovu datoteku možemo jednostavno da „premotavamo“ i da čitamo nekoliko puta unoseći svaki put nove nizove. Podaci se, dakle, mogu komforno unositi i bez standardne „datoteke“ input!

Personalni računari često koriste standardni kasetofon za skladištenje podataka. Standardni kasetofon ne treba porediti čak ni sa strimer trakom: mnogo je sporiji, podložniji greškama i, što je najvažnije, ne omogućava kompjuteru nikakvu kontrolu premetavanja. Kasetna je, dakle, sekvencijalni medij koji se kreće isključivo od početka

prema kraju; bilo kakvo premetavanje zavisno od korisnika koga ne treba previše opterećivati. Zbog ovih ograničenja nije moguće sortirati veće količine podataka upisanih na kasetu!

Ražočarali smo vas! Dobro, dajte nam još jedan šansu: podatke upisane na traku možete da sortirate ako su vam pri ruci dva (ili još bolje tri) kasetofona. Opet ste ražočarali: nemate tri kasetofona. Ipak, ne žalostite se baš mnogo: i da ih imate, podatke biste mogli da sortirate tek ukoliko vaš računar može istovremeno da kontroliše (uključuje i isključuje) dva tri uređaja. A to, koliko nam je poznato, ne može ni jedan popularan personalni kompjuter!

Nećemo se međutim, tek tako oprostiti od sortiranja podataka na trakama: opstet od metoda koje ćemo upoznati će nam koristiti docije. Kako ćemo, dakle, sortirati podatke upisane na kasetu ukoliko posedujemo tri kasetofona koja računaru kontrolišu? Pođimo redom i upoznajmo za programiranje neobično važan pojam mešanja podataka.

Mešanje...

Date su nam dve datoteke koje ćemo nazvati A i B; svaka od njih sadrži već sortirane podatke (npr. $A = (1, 3, 7, 12, 123)$ i $B = (-4, 0, 10, 12, 47)$). Treba da proizvedemo datoteku C koja sadrži sve podatke iz datoteka A i B, pri čemu će i C biti sortirana; u našem je slučaju $C = (-4, 0, 1, 3, 7, 10, 12, 12, 47, 123)$. Do rezultata se dolazi sasvim jednostavno, što pokazuje i paskal program sa slike 1.

Uzmemo jedan element iz datoteka A (1), a onda i element iz B (-4). Uporedimo ih pa manji (-4) upišemo u C, a onda uzmemo novi element iz B. Radi se o nuli koja je i dalje manja od 1; nula, dakle, putuje u C. Sledeći element iz B je 3, broj veći od nule. Zato nula šaljemo u C i uzmimo novi element iz A (7), koga poredimo sa 3. Postupak poredjenja i uzimanja sledećeg elementa iz datoteka čiji je član bio manji ponavljamo sve dok se jedna od datoteka (u našem slučaju B) ne isprazni; tada ostatak druge datotekoje jednostavno prepisemo u C.

Reko bi se da je postupak jednostavan i logičan i da o njemu nema smisla govoriti. Jeste li, međutim, primetili da je mešanje sa slike 1 sekvencijalno; da A i B nisu nizovi već datoteke na spoljnim medijima, C bi smo i dalje mogli da formiramo učitavajući jedan po jedan element iz A i B; bilo koje od ove tri datoteke može da bude mnogo veća od memorije našeg računara!

... i drugi trikovi

Vratimo se prvobitnom problemu: treba sortirati podatke iz datoteke A na tri kasetofone i tri kasetofona. Posao obavljamo u nekoliko faza:

1. Računar učitava što veći segment sa trake 1 u memoriju i sortira ga primenom neke od klasičnih metoda (npr. heap sort ili

U prošlim smo „Računarima“, govoreći o sortiranju brojeva, objavili nekoliko „klasičnih“ rešenja i uporedili njihove karakteristike. Zaključci do kojih smo došli, međutim, važe samo ako su podaci koje sortiramo upisani u memoriji našeg komputera. Rad sa velikim bazama podataka koje su smeštene na diskovima, disketama i trakama nameće nova pravila igre.

ISO-Pascal compiler V. 01.00	
1 0 -- progam posanj(input,output,dump);	51 1 -- a[n]:=b[i]; a3:=a3+1
2 0	52 1 -- end;
3 0 C	53 1 -- a3:=a3-1
4 0 C	54 1 -- end;
5 0 C	55 1 --
6 0 C	56 0 -- procedure readerr(var a:matrica; var ni:integer;
7 0 C	57 1 -- var err:Boolean);
8 0 C	58 1 -- begin
9 0 C	59 1 -- , ai:=0;
10 0 --	60 1 -- while not eof(dump) and (n<a3) do
11 0 --	61 1 -- begin
12 0 -- const naxn = 500;	62 1 -- ai:=ai+1;
13 0 -- read_nx = 79;	63 1 -- readln(dump,a[n]);
14 0 -- broj_nx = 8;	64 1 -- end;
15 0 --	65 1 -- err:=a>maxn
16 0 -- type podaci = integer;	66 1 -- end;
17 0 --	67 1 --
18 0 -- var a,b,c: matrica;	68 0 --
19 0 -- ai,a2,a3: integer;	69 0 -- procedure writearr(a:matrica; ni:integer);
20 0 -- err1,err2: Boolean;	70 1 -- var i,j,broj: integer;
21 0 -- dumpnt: text; { dummy input file }	71 1 -- begin
22 0 --	72 1 -- broj:=read_nx div broj_nx; i:=1;
23 0 -- procedure mes(a,b: matrica; var c: matrica;	73 1 -- while i<=n) do
24 0 -- ai,a2:integer; var a3: integer);	74 1 -- begin
25 1 -- var a_broj,b_broj,if: integer;	75 1 -- j:=1;
26 1 --	76 1 -- while i<=n and (j<=broj) do
27 1 -- begin	77 1 -- begin
28 1 -- a_broj:=1;	78 1 -- write(a[i,j],broj_nx);
29 1 -- b_broj:=1;	79 1 -- i:=i+1; j:=j+1
30 1 -- a3:=1;	80 1 -- end;
31 1 -- while (a_broj<=n) and (b_broj<=n2) do	81 1 -- writeLn
32 1 -- if [a_broj] <= [b_broj] do	82 1 -- end;
33 1 -- then	83 1 -- end;
34 1 -- begin	84 1 --
35 1 -- c[a3]:=a[a_broj];	85 0 --
36 1 -- a3:=a3+1; a_broj:=a_broj+1	86 0 -- begin { glavni program }
37 1 -- end	87 0 -- writeln ('Unesite elemente prvog niza');
38 1 -- else	88 0 -- reset (dump, '');
39 1 -- begin	89 0 -- readerr (a, ai, err1);
40 1 --	90 0 -- writeln ('Unesite elemente drugog niza');
41 1 -- c[a3]:=b[b_broj];	91 0 -- reset (dump, '');
42 1 -- end;	92 0 -- readerr (b, a2, err2);
43 1 -- { petlja se eventualno izvršava 0 puta }	93 0 -- if err1 or err2
44 1 -- for i:=a_broj to ai do	94 0 -- then writeln ('Previše podataka!');
45 1 -- begin	95 0 -- else begin
46 1 --	96 0 -- mes (a, b, c, ai, a2, a3);
47 1 -- end;	97 0 -- writearr (c, a3)
48 1 -- { petlja se eventualno izvršava 0 puta }	98 0 -- end
49 1 -- for i:=b_broj to a2 do	99 0 -- end;
50 1 -- begin	0 Compilator error(s)
	Code size = 979 bytes

Slika 1

particijski sort). Ovi se podaci upisuju na traku 3.

2. Računar učitava sledeći segment podataka, sortira ga i upisuje na traku 2. Faza 2 se ponavlja sve dok ima podataka; tako su na traci 2 formirane datoteke A1, A2, A3, ... An-1 jednakih veličina i An koja je eventualno manja.

3. Računar zahteva od korisnika da premotira trake 2 i 3 na početak i da u kasetofonu 1 umetne novu, praznu kasetu.

4. Računar meša podatke sa traka 2 (A1) i 3 i formira novu datoteku na traci 1.

5. Od korisnika se zahteva da premotira trake 1 i 3 na početak i da im zatim razmeni mesta. Koraci 4 i 5 se ponavljaju sve dok se ne „promesaju“ sve datoteke sa trake 2.

6. Sortirani podaci su na traci 1 koju još treba premotati na početak da bi se rezultati ispisali na ekranu ili upotrebili na neki drugi način.

Iako je čitav postupak prilično komplikovan, smatrali smo da se vredi potruditi i napisati program 2 koji simulira čitavu operaciju.

Program je pisan na standardnom bejziku i u njemu nećete naći ni na kakvo

pominjanje datoteka. Za to smo imali nekoliko dobrih razloga. Pre svega, teško je testirati program koji se stalno obraća sporom kasetofonu. Dalje, rad sa datotekama na traci nije baš standardno rešen u raznim jezicima. Treće (i najvažnije): autor ovoga teksta bi još negde i pronašao kompjuterski kasetofon, ali bi pronalazanje kablova bilo pravi problem ...

Uveli smo, dakle, matricu *traka* koja simulira kasete: *traka (0,5)* je peti element na prvoj traci. Na prvu su traku u početku upisani slučajni brojevi koje treba sortirati: na istoj će se traci na kraju naći i rezultat. Tu je, naravno, i niz *memorija* koji je dimenzionisan tako da bude mnogo manji od svake od „traka“. Računar će, kada mu god zatreba, ispisati uputstva operatoru u stilu „razmeni trake u kasetofonima 1 i 3“ ili „premotaj traku u kasetofonu 1“, a zatim će ovu operaciju samostalno izvesti — treba samo da primetite koliko bi u realnosti bilo posla.

Testiranje je pokazalo da je program sa slike 2 prilično spor: na inače brzom BBC-jevom bejziku sortiranje 300 brojeva zahteva 3 minuta rada; da se računar stvarno obraća kasetofonu, posao bi trajao mnogo duže. Može li se nešto ubrzati? Bilo bi pametnije mešati datoteke A1 i A2, A3 i A4, itd, a zatim dva po dva rezultata. Tada bi,

međutim, trebalo mnogo više pomoćnih kaseti i žongliranja sa njima.

Pre nego što se (zauvek) oprostimo od kasetofona, pomenućemo još jedan veoma ozbiljan problem. Čak i kada bi spektrom mogao da kontroliše tri kasetofona, postupak koji smo opisali se na njemu ne bi mogao primeniti jer je zapis podataka na traci kontinualan: ne možemo da učitamo 100 bajtova, da zaustavimo traku i da docnije učitavamo dalje. Zašto ne možemo? Čak i kada bi traka mogla da se zaustavi i pokrene bez „praznog hoda“ koji je izazvan inercijom, računar neće pronaći novi lider pa će ignorisati ostatak podataka! Mnogo je zgodnije ako računar organizuje podatke na traci „po blokovima“, pa se svaki blok može nezavisno učitati — toliko što se liče kritika pojedinih „stručnjaka“ koji smatraju da je sećanje program na blokove loša strana „amstrada“, BBC-ja i „elektrona“!

Pravo u glavu

Često se kaže da je disk jedinica uređaja koji omogućava direktan pristup podacima. Šta beše direktan pristup? Ako su podaci upisani u RAM ili ROM vašeg komputera, možete potpuno slobodno da pristupate bilo kom od njih: pošto ste pročitali sadržaj ćelije A000A i &C00C je potpuno jednako.


```

10 REM
20 REM Simulacija sortiranja podataka
30 REM na trakama priramon 3 kasetasa
40 REM
50 REM Dejan Zlatanovic 1986.
60 REM
70 REM "Kasunar 20"
80 REM
90 REM
100 eof=filo:9999:endifdata=9888
110 max=500 : REM max. podataka na traci
120 maxa=20 : REM max. podataka u memoriji
130 maxsp=300: REM koliko se podataka sortira
140 DIM traka(1,max),count(2)
150 DIM max(max)
160 REM
170 REM Priprema podataka na traci 0
180 REM
190 FOR i=1 TO maxsp
200 traka(i,1)=RND(1000)-500
210 NEXT i
220 traka(0,maxsp)=endiffilo
230 REM
240 REM Podaci sa trake 0 se razbijaju
250 REM sa delove velicine 'max' koji
260 REM se sortiraju i upisuju na
270 REM traku 1 i traku 2.
280 REM
290 PRINT "Stavi traku sa podacima u kasetofon 1"
300 PRINT "Stavi prazu traku u kasetofon 2"
310 PRINT "Stavi prazu traku u kasetofon 3"
320 PRINT
330 eof=FALSE
340 FOR i=0 TO 2:count(i)=1:NEXT i
350 REM
360 pass=2
370 max_count=1
380 tape_no=0:GOSUB 990
390 IF eof THEN 430
400 max(count)=byte_r
410 max_count=max_count+1
420 IF max_count=max THEN GOTO 380
430 max_count=1:IF eof THEN GOTO 490
440 GOSUB 1100:REM Sort
450 FOR i=1 TO max_count-1
460 tape_no=pass:byte_a=mem(i):GOSUB 1070
470 NEXT i
480 tape_no=pass:byte_a=endiffilo:GOSUB 1070
490 IF NOT eof THEN pass=1:GOTO 370
500 tape_no=pass:byte_a=endifdata:GOSUB 1070
510 REM
520 REM Masanje podataka sa trake 1
530 REM i trake 2. Rezultat na traku 3
540 REM
550 PRINT "Premotaj traku u kas. 2 na pocetak" :count(1)=1
560 PRINT "Premotaj traku u kas. 3 na pocetak" :count(2)=1
570 PRINT "Stavi prazu traku u kasetofon 1" :count(0)=1
580 PRINT
590 eof=FALSE
600 tape_no=2:GOSUB 990:IF eof THEN 860
610 eof=byte_r
620 tape_no=1:GOSUB 990:IF eof THEN 700
630 b=byte_r
640 IF eof THEN 680
650 byte_a=alttape_no:GOSUB 1070
660 tape_no=2:GOSUB 990:IF eof THEN 860
670 eof=byte_r:GOTO 640
680 byte_a=b:bitape_no=0:GOSUB 1070
690 GOTO 620
700 byte_a=altape_no:GOSUB 1070
710 eof=FALSE
720 tape_no=2:GOSUB 990:IF eof THEN 750
730 byte_a=byte_r:tape_no=0:GOSUB 1070
740 GOTO 720
750 tape_no=0:byte_a=endifdata:GOSUB 1070
760 IF traka(1,count(1))=endifdata THEN 910
770 REM
780 REM Kasnena traka i priprema za
790 REM sledece masanje.
800 REM
810 PRINT "Kasnena traka u kasetofonu 1 i 3"
820 PRINT "i obe premotaj na pocetak."
830 PRINT
840 FOR i=1 TO max:traka(2,i)=traka(0,i):NEXT i:count(2)=1:count(0)=1
850 GOTO 990
860 pass=0:byte_a=b:GOSUB 1070
870 eof=FALSE
880 tape_no=1:GOSUB 990:IF eof THEN 750
890 byte_a=byte_r:tape_no=0:GOSUB 1070
900 GOTO 860
910 PRINT:PRINT:PRINT "Sortiranje je zavrseno."
920 PRINT "Premotaaj traku 1 na pocetak." :count(0)=1
930 PRINT
940 eof=FALSE:tape_no=0
950 GOSUB 990:IF NOT eof THEN PRINT byte_r:GOTO 950
960 PRINT:PRINT
970 REM
980 REM
990 REM Bajt sa trake 'tape_no'
1000 REM ucitati u 'byte_r'
1010 REM
1020 byte_r=traka(tape_no,count(tape_no))
1030 count(tape_no)=count(tape_no)+1
1040 IF byte_r=endiffilo THEN eof=TRUE
1050 RETURN
1060 REM
1070 REM Bajt 'byte_r' upisati
1080 REM na traku 'tape_no'
1090 REM
1100 traka(tape_no,count(tape_no))=byte_r
1110 count(tape_no)=count(tape_no)+1
1120 RETURN
1130 REM SWAP SORT
1140 FOR i=2 TO 0
1150 next
1160 GOSUB 1250:REM UP
1170 NEXT i
1180 FOR i=0 TO 2 STEP -1
1190 max=mem(i):mem(1)=mem(i):mem(i)=
1200 mem=-1
1210 GOSUB 1350:REM DOWN
1220 NEXT i
1230 RETURN
1240 REM
1250 REM function UP
1260 REM
1270 i=b+1
1280 IF i=1 THEN RETURN
1290 i=INT(i/2)
1300 IF mem(p)>mem(i) THEN RETURN
1310 mem=(p+mem(i))/2:mem(i)=mem(p):mem(p)=
1320 mem
1330 GOTO 1280
1340 REM
1350 REM function DOWN
1360 REM
1370 i=b-1
1380 i=i+1
1390 IF i>=b THEN RETURN
1400 IF mem(p)<mem(i) THEN RETURN
1410 IF mem(i)>mem(p) THEN RETURN
1420 mem=(p+mem(i))/2:mem(i)=mem(p):mem(p)=
1430 mem
1440 GOTO 1380

```

slika 2

Ukoliko sa druge strane, pročitate prvi bajt sa trake, čitanje sledećeg čeka moći da obavite praktično trenutno, dok će pristup hiljaditom bajtu potrajati. Traka je, dakle, medij sa *sekvencijalnim* pristupom, dok memorija računara predstavlja medij sa *direktnim pristupom* (pod memorijom podrazumevamo ROM i RAM — i jedan i drugi omogućavaju slobodan pristup podacima.

premda je samo RAM (nesrećno izabrana) skraćena od *Random Access Memory*. Disk je, sa strane korisnika, takođe medij sa direktnim pristupom: ako smo pristupili osmom bajtu sa treće trake, normalno ćemo doći direktno do osmog bajta osme trake. Poznavaoci karakteristične diska, sa druge strane, opovrgavaju opšte poznate tvrdnje: izbor trake obavljaju dekoderski elementi koji, zaista, deluju praktično trenutno. Kada, međutim, deko-

der izda naredbu upisno-čitajućoj glavi da putuje do osme trake, na scenu stupa mehanika glava č brže stiči do osme nego do osamdesete trake! Pokažimo, ipak, malo dobre volje i zanemarimo vreme putovanja trake. Kada glava dođe do osme trake, treba još da pronade treći bajt. Disketa se okreće određenom brzinom i to isključivo u jednom smeru — ukoliko glava ima sreću", treći će bajt naići upravo kada se ona nađe iznad osme trake, pa će učitanje biti praktično trenutno. Ukoliko na glavu deluju Marljivi zakoni, treći je

```

1 0 - program sortiranje(input,output);
2 0 - const maxn = 100;
3 0 - type podaci = integer;
4 0 - matrix = array [1..maxn] of podaci;
5 0 - var
6 0 -   matrix: matrix;
7 0 -   n: integer;
8 0 - procedure sort(var a:matrix; n:integer);
9 1 - ( metoda minimuma prilagodjena spoljnim memorijama )
10 1 -
11 1 -   var i,j: integer;
12 1 -       m: podaci;
13 1 -
14 1 -   procedure swap(var p,q:podaci);
15 1 -       var t:podaci;
16 1 -       begin
17 1 -         t:=p; p:=q; q:=t;
18 1 -       end;
19 1 -
20 1 -   begin
21 1 -     for i:=1 to n-1 do
22 1 -       begin
23 1 -         m:=a[i];
24 1 -         for j:=i+1 to n do
25 1 -           if a[j] < m then swap(m,a[j]);
26 1 -

```

```

27 1 -         m:=a[i];
28 1 -       end;
29 1 -     end;
30 1 -
31 1 - procedure readarr(var a:matrix; var n:integer);
32 1 - begin
33 1 -   n:=0;
34 1 -   while not eof do
35 1 -     begin
36 1 -       readln(a[n+1]);
37 1 -       n:=n+1;
38 1 -     end;
39 1 -   end;
40 1 -
41 0 - procedure writearr(a:matrix; n:integer);
42 0 - var i:integer;
43 0 - begin
44 0 -   for i:=1 to n do
45 0 -     write(a[i]);
46 0 -   end;
47 0 -
48 0 - begin { glavni program }
49 0 -   readarr(a,n);
50 0 -   sort(a,n);
51 0 -   writearr(a,n);
52 0 - end.
53 0 - compilation error(s)
54 0 - Code size = 412 bytes

```

slika 3

bajt upravo promakao pa će za njegovo učitavanje disketa morati da napravi čitav krug. Možemo, dakle, da smatramo da će u nekom prosečnom slučaju disketa napraviti pola okretaja pr nego što se pročita željeni bajt. Dok se ova rotacija obavlja, disk kontroler čeka bajt, a računarski disk kontroler — sa suštinske strane, situacija je ista kao da smo zahtevali učitavanje hiljadito bajta kasete koja je premošana na početak. Disk jedinica, dakle, kombinuje direktan i sekvencijalan pristup podacima (direktno se pristupa traci a sekvencijalno sektoru ili bajtu) ali se zahvaljujući brzini njenog okretanja u nekim aplikacijama može uspešno aproksimirati uređajem sa direktnim pristupom.

Sortiranje, na žalost, nije jedna od aplikacija u kojoj smemo da se uljuljukamo u verovanje da se podacima na disketama slobodno pristupa. Moraćemo, šta više, da se upustimo u dalje proučavanje komunikacije računar — disk i upoznamo pojam bafera.

Računar je, da ostanemo kod ranijeg primera, tražio učitavanje trećeg bajta osmog sektora diske. Glava je došla do osme trake ali — kako da pronađe treći bajt? Tu je, naravno, spasonosni otvor koji signalizira početak trake i u odnosu na koga treba brojati bajtove. I pored precizne izrade mehanike, nemoguće je biti siguran koji je bajt koji a treba i predvideti neku vrstu čeksuma koji će proveravati ispravnost prenosa podataka. Čak i ukoliko zanemarimo ove probleme, obeshabrice nas jedan paradoks: pretpostavimo da je računar primio treći bajt i da mu je odmah zatrebao sledi, četvrti. Dok je odgovarajući zahtev poslat disk kontroleru, disketa se u neprekidnom okretanju pomerila i bajt je „promakao“ — da bi mu se pristupilo, treba sačekati da disketa napravi čitav krug. Ukoliko, dakle, u memoriju učitavamo program koji uvek predstavlja niz sukcesivnih bajtova, disketa će morati da se okrene onoliko puta koliko program ima bajtova — to bi trajalo toliko dugo da bi se svi varijetali trakama, pa bi jedni proizvođači disk jedinica ostali bez nasušnog hleba. Rešenje je, dakle, moralo da se pronađe pa je i pronađeno: uvođenje bafera rešava problem učitavanja sukcesivnih bajtova. Problem adresiranja bajtova i problem čeksumal Da vidimo kako.

Svaka je traka podeljena na određen

broj sektora, na primer njih 10. Svaki sektor, sa druge strane, sadrži određeni broj bajtova, na primer 256. Umesto da pristupamo trećem bajtu osme trake pristupaćemo, dakle, trećem bajtu **naulog sektora** osme trake, dok ćemo pristup 2073. bajtu 12 trake zamisliti pristupom dvadeset petom bajtu **osmog sektora** dvanaeste trake. Adresiranje smo, dakle, iskomplikovali, ali smo ponešto i dobili. Pre svega, svaki sektor može sebi da dopusti luksuz da, osim korisnog sadržaja, obuhvati identifikaciju (broj sektora) i odgovarajući čeksum. U memoriji računara se neće prenositi samo jedan bajt već čitav sektor: ukoliko smo tražili 2073. bajt, dobićemo ih njega i 2048, 2049, 2050 itd — poslednji će biti 2303 bajt.

Problem adresiranja sektora i proveru ispravnosti prenosa smo, dakle, rešili. Kao što stoji stvar sa sukcesivnim bajtovima? Vrlo lepo: ako nam, posle 2073. zatreba 2074. bajt, nećemo morati čak ni da se obraćamo disku: bajt je u bafetu (tj. u RAM-u), pa mu pristupamo praktično trenutno. Ukoliko nam budu trebali i dalji bajtovi, disk će moći da miruje sve dok, idući redom, ne zatražimo 2304. bajt kada će se, jasno, ponovo prepisati celih 256 bajta.

Šta se menja ako, umesto prostog čitanja, želimo da menjamo podatke na disku? Kada zatražimo promenu 2073. bajta dvanaeste trake, u bafet će biti prepisano 256 bajta, a onda će izmena biti upisana u RAM ali ne i na disk. Dok budemo menjali bajtove čije su adrese između 2048 i 2303, operacije će se obavljati u RAM-u i biće izuzetno brze; kada pristupimo 2304. bajtu, čitaj će se bafet prepisati na disk a onda će u memoriju biti učitani sledeći sektor od 256 bajta spreman za izmenu. Ukoliko, dakle, u sred traci nestane struje, neki promenjeni podaci neće biti upisani na disketu pa sadržaj datoteke neće biti savršeno ažuran što svakako nije prednost. Radi se, ipak, o malo) cenj koji je plaćena za strahovito ubrzavanje rada.

S kraja na kraj

Obzirom da je pristup RAM-u praktično trenutno a pristup disku spor, sortiranje će biti strahovito sporo ako, koristeći neki inače dobar metod, nateramo glavu da šeta sa kraja na kraj datoteke i da za svaki potreban bajt učitava i upisuje njih 256. Pre nego što kažemo kako se podaci najčešće sortiraju, podsetimo se metoda koje smo upoznali u prošlom broju i pogledati ih u novom svetlu.

Zdravorazumska metoda minimuma, za čudo, nije loša, ali je nikako ne smete

primenjivati kao što smo opisali u prošlom napisu. Na slici 3 dajemo, zato, novu varijantu programa.

U čemu je razlika? Sasvim sitna: umesto da poredimo A(i) i A(j), poredimo M i A(j). Pri čemu smo promenljivoj M najpre dodelili vrednost A(i). Zašto uvodimo pomoćnu promenljivu? Zamislimo da je i=20 i J=400. Pristupamo najpre dvadesetom bajtu pa, dakle, učitavamo čitavih 256 bajta. Zatim pristupamo 400-tom, bajtu zbog čega moramo da preprišemo drugih 256 bajta u bafet pr nego smo prvih 256 vratili na disketu. Zatim je J postalo 401 i treba da poredimo A(20) i A(401). Da bi računarski saznao koliko je A(20), mora ponovo da učitava prvi sektor i da onda čita drugi da bi saznao koliko je A(401). Ukoliko, sa druge strane, A(20) prenesemo u promenljivu M koja se, normalno, čuva u RAM-u, iz datoteke ćemo čitati sukcesivne A(i) i tako koristiti sve blagodeti bafera.

Metoda *merhurica* je možda još pogodnija i to bez ikakvih izmena: ona i tako podrazumeva sukcesivno pristupanje brojevima koji se sortiraju. Određeni problem nastupa kada pristupamo bajtovima na granicama sektora (ponekad ih treba i razmeniti, što podrazumeva bar četiri učitavanja sektora) ali — to je jedan bajt na 256. Ukoliko volite izazove, pokušajte da modifikujete program iz prošlih „Računara“ tako da minimizirate ovaj problem.

Selova sortirka nije naročito pogodna — česti prelazi preko granica sektora dovode do nepotrebnog prenošenja podataka na relaciji računar — disk.

Heap sort nije mnogo bolji ili, da budemo malo precizniji, nije bolji u osnovnoj verziji. Pre samo godinu dana američki časopis *CK Journal* objavljuje modifikovan algoritam za *heap sort* koji je prilagođen operativnim sistemima IBM-ovih računara iz serije 370, ali verujemo da ovaj algoritam ne bi bio posebno interesantan za većinu naših čitalaca.

Quick sort nazmičeno pristupa levom i desnom kraju niza, što znači da je katastrofalno loš za klasično organizovane bafere. Poneki operativni sistem, sa druge strane, omogućava da se jednoj datoteci dodeli nekoliko bafera, u kom bi slučaju *quick sort* mogao da bude interesantna alternativa klasičnim metodama.

Da vidimo, najzad, kako se stvarno sortiraju podaci na disku. Na isti način kao i podaci na traci: mešanjem. Učitavamo, dakle, što veće segmente podataka u RAM (ne samo u bafet nego u čitav slobodan RAM), pa ih sortiramo nekom od brzih metoda

(inpr. partijski). Ukoliko je datoteka imala M podataka, a u našu memoriju može da stane njih N, ovim ćemo postupkom dobiti otprilike M/N potpuno sortiranih datoteka, koje treba ujediniti u jednu. Mešačemo, dakle, dve po dve od M/N datoteka i tako, korak po korak, formirati konačnu. Vrlo je zgodno da, ako posedujemo dve disk jedinice, ne smestimo sve tri datoteke na istu disketu — samljačemo nepotrebno „šetanjnj“ glave. Najbrže bi bilo da imamo tri disk jedinice, pa da na dve smestimo datoteke koje se mešaju, a na treću rezultat; ovakav luksuz, ipak, retko ko sebi dopušta!

„Klasična“ kompjuterska literatura, kao što je knjiga *Organizacija podataka* Hartmuta Vedeckina, opisuje i mnoge druge metode sortiranja podataka na trakama i na diskovima; pomenućemo simetrično i asimetrično mešanje, kaskadno sortiranje i oscilirajuće sortiranje. Obzirom da su magnetne trake u mnogome stvar prošlosti, nećemo se posebno baviti ovim metodama, ali ćemo zato, kada se budemo ozbiljnije pozabavili bazama podataka, upoznati i jedan sasvim drugačiji metod za sortiranje podataka: formiranje takozvanih indeksa, koje podrazumeva da se podaci koje sortiramo uopšte ne pomeraju sa mesta!

Obrtanje u prazno

Počeli smo od trake, govorili o floppy disku, pa je red da posvetimo nekoliko reči danas sve popularnijim i jeftinijim masivnim diskovima. Hard disk se, za razliku od diskete, neprekidno okreće bez obzira na to da li su vam trenutno potrebni neki podaci ili nisu. Kada ispred računara sedi jedan korisnik, ovo je okretanje prilično neracionalno: korisnik se, bar pri upotrebi većine aplikacionih programa, razumno retko obraća disku. Ukoliko se, sa druge strane, rad i sistemu koji opslužuje gomilu korisnika, disk će biti vreme zaposen: ne samo da su se zahtevi korisnika „poredali u red“ već i sam operativni sistem, zbog smene stranica i drugih „kućnih poslova“, zahteva česte pristupe disku. Ovim razmatranjem nemamo, naravno, nameru da kažemo da je hard disk neisplativ za pojedina — kažemo samo da je iskorisćenost diska u vremenu mala.

Sortiranje podataka na masivnom disku se, suštinski, ne razlikuje od sortiranja podataka na disketi: sve se operacije jednostavnije brže obavljaju! Ekvivalentnost medija je dopunjena time što većina operativnih sistema na personalnim računarima tretira hard disk kao i floppy — svakoj se datoteci dodeljuje po jedan bafer konstantne veličine. Ukoliko vaš kompjuter ima moderan operativni sistem poput „amignog“, možda će se isplatiti upotreba višestrukog bafersiranja i optimizacija veličine bafera. Ukoliko, sa druge strane, radite sa XT-om ili nekim kompatibilcom koga ste opremili hard diskom, zadovoljite se napomenama koje smo dali korisnicima disketa.

Naučili smo, dakle, da sortiramo podatke u memoriji i podatke na diskovima; ostalo je još da te podatke obrađujemo, tj. pretražujemo. A o pretraživanju podataka — u sledećim „Računarima“.

Dejan Ristanović

Računari u domaćoj radinosti

Turbodrajv za „spektrum“

parče mašinskog koda

32048	18	08	FD	CB	7C	FE	18	05	C9	FD	CB	7C	F6	0D	07	06	2F
32064	21	86	51	1E	9E	50	01	07	00	ED	20	CD	3D	05	0C	42	71
32080	07	3A	DE	5C	47	3A	86	5C	8D	08	06	FE	03	2B	13	38	96
32096	07	16	0D	7E	7C	7E	7C	7E	7C	7E	7C	7E	7C	7E	7C	7E	7C
32112	FD	CB	7C	7E	28	02	E7	1A	2A	FD	CB	ED	5B	87	5C	7C	49
32128	85	2E	0E	ED	52	30	0A	FD	CB	7C	66	28	02	87	13	87	19
32144	15	2A	8A	7C	85	20	03	2A	B9	5C	3A	86	5C	A7	20	85	
32160	03	2A	53	5C	CD	39	10	18	21	3A	8E	5C	86	60	20	05	7A
32176	CD	10	07	14	0E	75	87	5C	00	00	00	00	00	00	00	00	00
32192	ED	D1	85	CD	59	10	E1	07	GE	08	CD	8A	0F	C3	C1	05	94
32208	ED	5B	87	5C	29	81	5D	85	7C	85	20	06	13	13	13	EB	52
32224	18	09	2A	0F	5C	EB	37	ED	52	38	09	11	05	00	19	44	9B
32240	4D	07	05	1F	31	8A	86	5C	47	28	33	7C	FE	28	0E	2A	38
32256	2A	5C	28	46	7B	4A	2B	53	4B	7C	8A	19	2A	5C	0C	15	95
32272	28	ED	48	87	5C	03	03	03	3A	8C	5C	75	07	55	16	34	3C
32288	23	F7	77	01	C3	73	23	72	23	CD	39	10	18	0C	FD	CB	87
32304	7C	8E	ED	5B	53	5C	2A	59	5C	2B	07	85	19	42	4B	87	FF
32320	5A	2A	53	5C	07	58	16	25	ED	4B	8B	5C	09	2D	48	5C	8F
32336	3A	8E	5C	67	86	0A	08	FD	CB	7C	CE	3A	0D	0A	5C	8F	CA
32352	22	42	5C	FD	36	0A	00	2A	53	5C	ED	5B	87	5C	ED	22	4E
32368	57	5C	23	18	8A	01	01	00	ED	43	D6	5C	0E	03	2B	43	47
32384	DA	3C	01	75	09	ED	43	5C	FD	CB	7C	86	6D	20	07	38	
32400	21	86	50	11	8E	5C	01	09	00	ED	80	FD	CB	0A	FE	CD	F2
32416	42	0F	C3	21	08	72	75	8E	E5	23	C3	AE	20	07	17	32	C7
32432	6C	5C	F6	04	03	3F	3A	C5	5C	FD	CB	03	38	01	23	38	40
32448	08	7F	77	E1	F8	09	C3	05	D5	F5	3A	C6	5C	A7	20	67	82
32464	F1	32	05	5C	06	01	0E	05	38	02	23	04	4F	78	CB	41	80
32480	20	02	F6	08	02	C6	5C	F6	04	03	07	F1	7F	40	7E	ED	4B
32496	79	06	78	0E	11	00	00	00	00	00	00	10	E7	79	0A	00	15
32512	0A	08	7E	1F	3B	7F	05	11	00	07	F5	CD	9A	0A	18	18	
32528	7A	83	20	F8	F1	D1	30	08	ED	8C	6F	28	1A	CD	FE	CE	CA
32544	05	18	25	08	7B	CB	4F	20	0F	CD	FE	09	18	F5	1B	7A	3A
32560	B3	00	CD	78	09	E7	10	F1	D1	C1	81	AF	09	19	7F	7B	2C
32576	FE	1F	08	3E	0B	8E	1B	DE	FE	36	00	14	87	2A	C7	28	
32592	5C	D5	05	9A	0A	0E	00	00	00	00	00	00	00	00	00	00	00
32608	C5	01	7F	0E	8E	84	ED	79	CD	9A	0A	42	ED	78	1F	38	D6
32624	7F	1C	8E	1F	28	0E	0C	57	20	10	DE	CB	67	28	78	58	
32640	02	87	07	87	0A	4F	22	8A	5C	E1	01	E1	09	2A	C7	5C	51
32656	D5	05	85	0B	9A	0A	06	10	1E	C1	0D	0A	E1	85	05	54	
32672	01	7F	0E	87	0A	00	00	00	00	00	00	00	00	00	00	00	00
32688	1C	8E	28	ED	CB	1F	7B	28	02	87	0E	87	0E	87	0E	87	0E
32704	10	8B	18	89	3E	05	30	FD	C9	69	60	01	7F	40	16	8E	8E
32720	10	7C	47	80	04	16	00	18	0A	ED	8C	BD	28	12	06	00	00
32736	7C	8D	79	06	00	7A	8D	79	CD	9A	0A	ED	78	1F	38	7F	8A
32752	06	80	7B	ED	79	44	16	00	21	05	5C	2A	C5	5C	FE	63	5F
32768	38	01	23	01	07	4D	ED	76	77	C9	FD	CB	D1	A1	7E	46	5A
32784	07	8F	02	FD	CB	62	28	F6	3A	8C	5C	C9	F5	3E	FE	85	
32800	21	8C	5C	36	FD	D7	01	16	F1	07	10	00	C9	F5	3E	03	03
32816	18	8F	4F	0B	FE	E6	1F	96	1F	C9	DD	CB	43	4E	2B	05	87
32832	FD	36	00	07	FE	DB	24	0D	CD	AD	09	C9	D9	01	20	00	84
32848	D9	20	2A	4F	5C	11	14	00	00	D9	7E	00	ED	80	28	47	
32864	37	DD	7E	04	8E	7F	FE	4A	20	2A	3A	B6	5C	DD	BE	19	AA
32880	20	1C	D9	08	6E	1A	0D	66	18	D9	ED	48	DA	5C	2A	0C	25
32896	5C	0D	5A	0C	20	0B	DD	CB	18	46	28	02	77	0D	DD	5E	16
32912	09	DD	56	0A	10	18	C2	2A	53	5C	2B	E5	01	53	02	52	56
32928	07	8E	16	05	03	02	03	01	00	19	00	ED	80	28	0C	78	0C
32944	07	19	0D	03	43	8E	01	53	02	DD	E5	81	CD	7B	0C	35	
32960	ED	8D	4A	DA	5C	CB	78	20	0E	78	81	28	0A	7E	DD	77	7F
32976	0E	23	DD	23	08	18	FD	01	E1	09	7C	85	20	1A	2A	4F	8B
32992	5C	2E	01	00	02	87	55	16	81	01	00	02	DD	19	09	CD	48
33008	7E	0E	16	03	DD	23	05	08	20	03	78	87	87	2B	08	00	6C
33024	2D	E5	E1	11	C	00	19	EB	21	2C	0D	01	0C	00	ED	80	F8
33040	0D	85	E1	11	37	00	01	00	00	19	EB	21	2C	0D	00	ED	80
33056	0D	85	E1	ED	5B	4F	5C	ED	52	23	C9	DD	6E	1A	DD	8A	8A
33072	66	18	85	DD	7B	19	FD	85	E1	01	53	02	77	8E	19	40	
33088	0D	85	E1	ED	5B	4F	5C	ED	52	23	C9	DD	6E	1A	DD	8A	8A
33104	0C	F1	E1	47	0D	2A	4F	5C	E1	11	14	00	19	DD	7E	00	4D
33120	FE	80	28	18	7E	0A	E6	7F	4E	20	05	DD	02	7E	19	66	
33136	B8	08	ED	5E	09	DD	56	0A	DD	19	18	E1	01	00	02	25	DD
33152	05	07	88	19	E1	01	00	00	00	00	00	00	00	00	00	00	00
33168	8E	0E	16	03	DD	23	05	08	20	03	78	87	87	2B	08	00	6C
33184	7E	0E	FE	20	0A	0D	23	10	5C	E1	01	00	00	00	00	00	00
33200	5E	84	5C	CD	98	0C	98	0C	98	0C	98	0C	98	0C	98	0C	98
33216	98	0C	ED	5B	4F	5C	E1	47	37	02	DD	3A	65	5C	ED	8A	8A
33232	52	08	ED	00	EB	C9	85	10	21	16	3C	72	3F	5C	5E	8D	03
33248	23	46	81	87	ED	52	20	05	11	52	00	10	07	30	08	00	00
33264	8E	87	ED	42	8B	2A	5F	5C	73	23	22	2A	5F	5C	23	23	0A
33280	30	20	D9	32	5F	5C	E1	C9	01	00	02	DD	2A	4F	5C	11	93
33296	14	00	1D	7E	DD	7E	00	FF	80	8C	85	DD	7E	0A	4E	7F	5A
33312	7E	4D	20	1D	DD	5E	1A	3D	5E	1B	20	5E	1B	20	5E	1B	20
33328	ED	42	DD	75	1A	DD	7A	E1	DD	5E	09	DD	56	0A	DD	46	46
33344	19	18	01	00	00	00	00	00	00	00	00	00	00	00	00	00	00
33360	00	00	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20

Poslednjim baftovima mašinskog koda operativnog sistema za rad sa diskom, turbodrajv, kojim objavljujemo u ovom broju, primakao se kraju i drugi naš ovogodišnji veliki projekat za vianlike još uvek vitalnog „spektruma“. Preostaje još samo program za unošenje mašinskog koda, — za sledeći broj pripremamo jedan izuzetno komforan haks-loader, meka se sasvim lepo može koristiti i loader koji je objavljen uz „Ekraniski editor“.

33376 00 00 00 00 00 FF FF CD 1C 0B 85 27 19 CD 18
33392 78 09 DD 85 E1 11 52 00 19 85 01 01 00 CD 21 0A 7F
33408 CD 8A 0E 30 04 00 DD CB 8 86 47 20 02 87 05 0D 6E
33424 1A DD 66 1B F5 01 02 00 CD 21 0A 81 70 29 E1 80
33440 DD CB 18 46 20 05 CD AD 0B 18 23 85 23 2E 90 47 15
33456 11 0B 00 10 70 FF AF 77 06 0A 0D 85 23 2D 9B 0E 06
33552 1A 7E 2F CD 0F CD CD CD 7E 0E 01 3E 05 23 7E 2B FE
33488 DD 71 19 CD 78 09 DD 77 E1 11 52 00 19 CD DA 0D 95
33504 CD 21 0A 13 1A 1D DD 85 45 1B 1B 1A 86 0C 80 38 6A
33520 02 3E 80 DD CB 43 CE 20 04 DD CB 43 8E 07 17 DD
33536 77 46 DD 7E 0D DD 77 44 47 C9 85 06 67 0B 8D 6E
33552 1A 7E 2F CD 0F CD CD CD 7E 0E 01 3E 05 23 7E 2B FE
33568 3F 7E 20 03 18 02 3E 87 48 3A CF 89 20 00 0E
33584 DD 7E 19 7E 20 02 E7 09 A7 E1 C9 06 05 C4 5E DD
33600 7E 19 CD 78 09 01 7F 40 4E DD 78 06 0E DD 79 06 01
33616 00 11 7F 00 38 10 ED 79 CD 94 0A 0B 78 1F 3B FB 56
33632 01 20 23 CD 0F CD CD CD 7E 0E 01 3E 05 23 7E 2B FE
33648 08 05 7E 3D 47 0F 01 11 01 10 F3 01 4F 3E 8B 09 89
33664 3C 06 C6 9A 0E 30 03 0C 18 FA 80 3C 41 4F 37 19 6C
33680 06 AA 23 23 7E CD CD 29 7E 20 18 86 3F 8D BE 29 EA
33696 23 20 06 7E DD BE 0D 28 06 23 23 10 E7 47 C9 37 81
33712 E5 D1 C9 7E 3F 23 28 77 18 EF 05 DD EF 5A 67 77 9F
33728 36 11 0B 00 19 0E 2D 06 0A 0A 7E 23 FE 3F DD E1 DD 2F
33744 E5 28 11 DD 7E 0B 20 0E 23 DD 23 10 P5 37 79 48
33760 DD E1 C9 DD 71 36 58 10 09 20 DD 0B AF DD E1 DD 9F
33776 7E 3E 18 EE 3A D6 5C DD 78 09 CD 0F 10 2A 89 5C 5F
33792 22 E4 5C DD 38 DD DD DD 18 46 20 05 CD EA 0F E7 4C
33808 0C DD CB 43 96 2B 7E 19 CD 78 0D 0E 1B 32 95 85
33824 00 19 EB 21 E6 5C 01 09 00 DD 71 0B ED 00 21 5D
33840 09 00 DD 4B E7 5C 09 CD 3C 2A D1 2A EA 5C ED 4B 2B
33856 E7 5C 78 81 28 22 00 7E 0C FE 02 20 0F E5 C5 CD
33872 80 10 C1 DD 85 E1 11 52 00 19 EB E1 ED 40 DD 34 3A
33888 0B 20 23 DD 94 0C 18 DA DD CB 43 8E 07 17 DD 32
33904 FF 0F 2A E1 5C 22 E4 5C DD 38 DD DD CB 18 46 28 3B
33920 05 CD 2E 11 E7 11 11 DD CB 43 56 18 8D 07 16 DD 4F
33936 E5 E1 11 52 00 19 11 B6 5C 01 09 00 80 80 29 22 27
33952 5C DD 5E 33 28 56 54 21 08 09 19 CD 3C 2A 7C 43
33968 32 E7 5C DD 3A 0D 11 09 00 DD 6E 45 DD 6E B7 7D
33984 ED 52 DD 75 45 DD 74 46 4D 85 E1 11 5B 00 19 CD
34000 5B 89 5C 18 1A CD 0A DD DD 7E 44 4A 00 DD 17 3D 5D
34016 57 1E F7 2A E9 50 19 EB DD 85 E1 01 52 00 09 CD AB
34032 0C 0F 3A E7 5C 32 E7 5C 20 DA D9 DD 4D 4E 45 DD 1A
34048 46 46 3A B6 5C CB 7F 20 03 ED 80 1A 8E 20 08 AB
34064 23 13 0B 78 21 00 3F 59 E7 13 DD CB 18 46 28 3B
34080 DD CB 43 CE CD D2 10 DD 7E 19 CD 78 09 06 00 DD
34096 66 1B 8E 1A 4E DD 60 0A AF CD 78 09 C3 FC 0B 32
34112 DD 2A AF 5C 11 1A 00 DD 19 DD 7E 00 FE 80 3E 0F 00
34128 CA 7B 09 DD 7E 0A FE CD 20 05 CD CB 18 E1 DD 4A
34144 5E 09 09 DD 56 0A DD 19 18 ED 7A B3 CB 8C 6F 0F 09
34160 CD 10 0B DD 7E 19 CD 78 09 CD 9F 10 DD E5 E1 11 E6
34176 52 00 19 01 01 00 DD 21 0A CD 8A 0E 38 02 E7 11 FC
34192 DD 27 29 CD 21 11 DD 66 1B DD 6E 1A 01 01 00 03 51
34208 CD 81 8E 2B FF 3F 3F 06 AA 7E 86 3F DD BE 29 0A
34224 0B E5 CD CD 21 0A 0F 06 AA 7E 86 3F DD BE 29 0A
34240 20 02 36 3F 23 23 23 10 F1 C1 E1 CD 60 0A C9 C5 6D
34256 01 7F 00 ED 78 E6 40 C1 CB E7 0E DD 21 FA FF DD 58
34272 19 DD CB 18 46 20 02 0E 07 CD 5E 0B DD 56 0C DD 96
34288 DD 19 DD 77 52 DD DD 7E 0E 13 DD 7B 0B DD 72 0C CB
34304 4A CB DD 7E 19 CD 78 09 CD 9F 10 DD AF CD 78 09 C9 57
34320 CD 50 11 38 14 FD CB 7E 6F 28 05 CD 43 10 0F 0F 6F
34336 DD 7E 0B 47 CC 2E 11 E7 0F 1B DD 7E 0C ED 0F FE EA
34352 80 20 08 DD CB 43 4E 28 02 CD BF 7D B6 29 12 13 EA
34368 DD 7E 3F 12 13 DD DD 0E 12 DD E5 E1 11 52 00 19 24
34384 CD 60 0A DD 3A 0D 4F 77 0B DD DD 77 0C 09 DD E5 56
34400 E1 11 52 00 19 01 01 00 ED CD 21 0A 11 0B DD CD 29
34416 46 29 3E 2E 90 47 19 10 F5 DD 3F 1E CD 60 0A D9 2E
34432 DD 16 29 FE CD DA 0D DD CB 29 BE 3C 3A D8 5C DD 7B
34448 01 CB 3A D6 5C 32 CF 5C 3E 3E 23 D6 5C CD 1C 0B 78
34464 DD E5 06 CB DD 36 52 00 DD 23 10 78 DD E1 3A CF C4
34480 5C DD 78 09 01 01 00 DD 6E 1A DD 66 1B 03 CD 21 68
34496 0A CD 70 12 23 7E 2B FE 3E 28 F2 01 01 00 CD 21 68
34512 0A AF CD 78 09 05 21 12 12 CD 13 2A 3C CF C6 89
34528 30 CD CD 50 12 3E 3A ED 20 12 E1 E5 23 CD 3F 12 CD 8A
34544 19 13 CD 12 23 23 23 DD 7B 06 2B 7E FF 3F 20 24 80
34560 11 0B 00 19 10 F5 DD 7E 18 20 07 47 2B 02 0E 68
34576 21 2B 12 CD 19 12 CD 3F 12 06 3F CD 58 12 CD 19 D6
34592 12 3C 7C 08 23 7E 2B AF 7 28 2B 23 CD 3F 12 CD 58 83
34608 12 DD 7E 28 2B 08 AF DD 77 28 3E DD 18 05 3E 3D
34624 DD 06 DD 7E 28 05 12 1B 0E 3E DD CD 50 12 F1 EA
34640 C9 0D 43 41 54 41 4C AF 47 20 23 46 52 47 00 E1
34656 52 45 43 DD 20 ED C5 27 18 1A C1 DD E1 C9 C5 E1 35952

kozmetika u selekciji najboljih . . . kozmetika

kozmetika u selekciji najboljih . . . kozmetika

loto na „spektrumu“

Ako vas neko pita koji je dobar sistem za loto, nemojte mu odgovoriti da je to onaj koji donosi prvu nagradu. Nijedan sistem, sam po sebi, ne može doneti zгодak. Sistem samo raspoređuje brojeve koje je čovek odredio. Od toga koji su brojevi određeni, znači od čoveka, zavisi koliko će biti pogodaka u sistemu. Zadatak sistema je samo u tome da pogodne brojeve „skupi“ u jednoj kombinaciji.

Ukoliko je sistem pun, svi pogodeni brojevi će se obavezno naći na jednom mestu. Ukoliko je skraćen, onda će se bar u jednoj kombinaciji naći minimum pogodnih brojeva koji sistem garantuje, a s nešto sreće i više od tog minimuma.

Odgovor je jasan, ali nameće novo pitanje:

Šta je pun, a šta skraćen sistem?

Vrtoglava progresija

Želimo da odigramo, na primer, osam brojeva. Neka su to brojevi od 1 do 8. Od osam brojeva moguće je napraviti osam kombinacija za loto (kombinatorika ih naziva kombinacijama bez ponavljanja sedmog razreda od osam elemenata). U našem slučaju pun sistem će imati sledeće kombinacije:

- 1 2 3 4 5 6 7
- 1 2 3 4 5 6 8
- 1 2 3 4 5 7 8
- 1 2 3 4 6 7 8
- 1 2 3 5 6 7 8
- 1 2 4 5 6 7 8
- 1 3 4 5 6 7 8
- 2 3 4 5 6 7 8

Izdavajmo sada dve kombinacije, petu i osmu:

- 1 2 3 5 6 7 8
- 2 3 4 5 6 7 8

Te dve kombinacije čine skraćeni sistem koji garantuje, u najnepovoljnijem slučaju, dva puta po šest pogodaka od sedam pogodnih brojeva. Drugim rečima, ako bi bila izvučena ma koja od ostalih šest kombinacija, mi bismo imali dva puta ovu manju nagradu, a prva nagrada bi nam pripala ukoliko bi bila izvučena jedna od dve kombinacije skraćenog sistema.

Skraćeni sistem nam, znači, omogućuje da uz višestruko smanjenu uplatu imamo neki garantovani minimalni dobitak — i nadu da će se „uklopiti sedmica“. Naravno, pod uslovom da se među brojevima koje smo odigrali nađu onih sedam koje će Suzana Mančić izvući sledećeg utorka.

I tako, sve ovo izgleda lako i jednostavno kad je u pitanju osam brojeva i isto toliko kombinacija. Ali broj kombinacija u loto vrtoglavu raste. Pun sistem od 10 brojeva ima 120 kombinacija, od 20 brojeva 77.820, od 30 brojeva 2.035.800, a od 39 brojeva svih 15.380.937 kombinacija. Kako od tolikog broja kombinacija izdvojiti onih nekoliko koje će nam pružiti bar kakvu-takvu garanciju?

Većina matematički obrazovanih ljudi, s dobrim poznavanjem kombinatorike, zatražili bi ramenima kad bismo od njih zagnužili odgovor na ovo pitanje. Jednostavno, nisu se bavili tim problemom, a za potpun odgovor verovatno ne bi bilo dovoljno stranica u ovom broju „Računara“. Zato nije nikakvo čudo što se u Jugoslaviji bezmalo na prstima jedne ruke mogu izbrojati ljudi koji prave skraćene sisteme za loto. Po abecednom redu to su: Bećirović (službenik), Jokić (arhitekt), Joksimović (inženjer), Kokotović (ekonomist), Šćepanović (inženjer)... možda još dvojica, trojica, kojima se skromno pridružuje i dolepotipisani haker Ž. V.

Ukoliko nameravate da uplovite u vode tog područja kombinatorike, očigledno morate biti spremni za duboko ronjenje. Ali, ako su vaše ambicije samo Ugračke, a imate spektrom, evo rešenja! Ukucajte bežik i mašinic sa ovih stranica, startujte program — i moći ćete da napravite hiljade različitih punih i skraćenih, uslovnih i bezuslovnih sistema.

Tamo-amo po meniju

Program „Sistemi za loto“ ima sledeće mogućnosti:

- Generisanje punih sistema.
- Izrada bezuslovnih skraćenih sistema s garancijama 4, 5 i 6 od 7.
- Izrada uslovnih skraćenih sistema s garancijama 4, 5, 6 i 7 od 7.

Sve sisteme program prikazuje u kombinacijama po 7 brojeva.

Broj kombinacija u jednom sistemu ne može biti veći od 4.000, što je uslovljeno raspoloživom memorijom računara, ali broj brojeva nije ograničen ničim osim pravilima lota.

Svi podaci se upisuju samo pomoću dve dirke: SPACE i ENTER, SPACE služi za biranje podataka, a ENTER za unošenje.

Kad program proradi, na ekranu ćete ugledati meni sa sledećim opcijama:

1. Broj brojeva

Možete praviti sistem od 8 do 39 brojeva. Ova opcija je obavezna. Ako ne odredite koliko će sistem imati brojeva, program neće prihvatiti izbor drugih opcija, osim opcija 2 i 16.

2. Garancija

Takođe obavezna opcija. Najniža mogućna garancija je 4 od 7, najviša 7 od 7. Za neke vrste sistema dovoljni su samo podaci iz prve dve opcije. Korišćenje ostalih opcija nije obavezno.

3. Zamena brojeva

Ova opcija omogućuje određivanje brojeva prve izrade sistema. Ako, na primer, pravite sistem od 12 brojeva, možete odabrati bilo kojih 12 brojeva od 1 do 39. Ukoliko brojeve ne zamenite, računar će prikazati sistem brojevima od 1 do 12. Zamena brojeva bitno utiče na broj kombinacija ukoliko se primeni jedna od ovih opcija: 6, 7, 8, 13 ili 14.

4. Fiksiranje

Možete fiksirati 1, 2 ili 3 broja. Time ćete postići da program iz punog sistema izbaci kombinacije koje ne sadrže sve fiksirane brojeve.

5. Favorizovanje

Program omogućuje favorizovanje 2 do 5 brojeva, a iz punog sistema će izdvojiti u skraćeni samo kombinacije koje sadrže najmanje jedan favorizovan broj. U jednom sistemu možete istovremeno primeniti fiksiranje i favorizovanje. Vodite ma računara da vam isti broj ne bude i fiksiran i favorizovan, jer je to besmisleno.

6. Razmak

Ovom opcijom se određuje minimalni dozvoljeni razmak između brojeva u sistemu. Mogući minimalni razmaci su 1, 2 i 3. Ne smete u istom sistemu upotrebiti ovu i opciju broj 7.

7. Blizanci

Ako niste primenili opciju 6, možete odrediti maksimalan broj takozvanih blizanaca (parova susednih brojeva, na primer 3—4). U jednoj kombinaciji mogu se doznaliti najviše 1, 2 ili 3 blizanca.

8. Par/nepar

Možete odrediti maksimalan broj narnih i neparnih brojeva u jednoj kombinaciji. Oba maksimuma ne moraju biti ista, ali njihov zbir ne sme biti manji od 7.

9. Prvi broj

Ovom opcijom se ograničava opseg prvog (najmanjeg) broja u kombinaciji. Možete odrediti, na primer, da taj broj bude u opsegu od 1 do 10.

10. Poslednji broj

Ograničavanje opsega poslednjeg broja u kombinaciji.

11. Dve grupe

Ovom opcijom brojevi u sistemu se dele u dve grupe. Sami određujete koliko će brojeva biti u kojoj grupi, a program raspoređuje brojeve u kombinaciji po obrascu 3+4 (3 brojeva iz prve i 4 iz druge grupe). Brojeve ne smete istovremeno deliti u dve i tri grupe.

12. Tri grupe

Ako brojeve podelite u tri grupe, njihov raspored u kombinaciji će biti po obrascu 2+2+3. Obratite pažnju na ograničavanje u vezi sa opcijom 11.

13. Tri kolone

I još jedna podela brojeva — na kolone. Onako kako su ođtampani na tiketu. Vi određujete minimalan broj brojeva u kombinaciji iz svake kolone. Uz ovaj uslov možete koristiti i uslove iz opcije 11 ili 12.

14. Zbir brojeva

Ovom opcijom određuje se minimalni i maksimalni zbir brojeva u kombinaciji. Program će iz punog sistema eliminisati sve kombinacije čiji zbir brojeva izlazi iz dozvoljenog opsega.

15. Izlaz

Biranjem ove opcije izlazite iz menija i pokrećete mašinsku rutinu koja će napraviti skraćeni sistem prema vašim zahtevima.

Ekspert za loto sistema i strastveni programer na „spektrumu“ Žarko Vukosavljević pripremio je pre nekoliko meseci za čitaoca „Računara“ program koji bi trebalo da im olakša dileme „kako-punipoliti-loto-ilištič-a-utorak-uvawe“, ali je redakcija, verovatno zbog preterane sklonosti ka programerskim temama, ovaj projekat odlagala nekoliko meseci. Od ovog programa, razume se, ne treba očekivati čudo tipa „kako-dobitl-pet-milijard-na-lotu“ — njegova osnovna namena je da pomogne igraćima da, uz povećane šanse na dobitak, racionalizuju svoja ulaganja. Iz tehničkih razloga program smo poddelili na dva dela — u ovom broju objavljujemo mašinske rutine, a u sledećem program na bejziku.

63501	21	89	88	11	BA	88	01	60	6D	36	00	ED	BO	3A	EB	FD	75
63517	98	FB	0E	CC	BA	02	CC	91	F8	3A	EA	FD	FE	01	CC	94	AD
63539	98	FB	0E	CC	BA	02	CC	91	F8	3A	EA	FD	FE	01	CC	94	AD
63549	27	7E	FE	FF	2B	06	23	10	F8	00	00	00	02	02	0D	01	09
63585	16	21	EC	FD	CD	71	FB	21	01	00	22	82	FF	06	07	3A	93
63581	78	FF	21	BB	FF	7D	2D	2B	06	07	3E	01	21	BF	68	59	09
63597	FF	77	3C	23	10	F8	BA	AF	FE	FE	20	86	0A	78	10	7F	45
63613	32	C5	FF	21	BF	FF	11	AB	FF	06	07	3E	12	23	13	83	03
63629	FA	63	2C	99	3A	0C	5E	53	CA	0C	FC	3A	2E	06	07	3A	93
63645	32	FF	5A	11	BF	FF	21	BB	FF	1A	BE	28	08	3C	12	BE	CA
63661	2C	99	11	BB	FF	21	BA	FF	1A	BE	28	08	3C	12	BE	CA	9D
63677	2C	99	18	68	11	AF	FF	21	BB	FF	1A	BE	28	08	3C	12	BE
63679	BE	CA	2C	99	18	53	11	AE	FF	21	BB	FF	1A	BE	28	08	3C
63709	3C	12	BE	CA	2C	99	18	53	11	AD	FF	21	BB	FF	1A	BE	28
63725	28	08	3C	12	BE	CA	2C	99	18	29	11	AC	FF	21	BB	FF	1A
63741	1A	BE	28	08	3C	12	BE	CA	2C	99	18	14	11	AB	FF	21	0B
63757	85	FF	1A	BE	CA	0C	FC	3C	12	BE	CA	2C	99	3C	12	BE	CA
63773	3C	12	3C	12	3C	12	3C	12	3C	12	3C	12	3C	12	3C	12	3C
63789	AB	FF	21	BB	FF	2D	0B	FA	EB	FF	12	23	13	10	F7	78	06
63805	21	DA	FF	AB	B6	28	07	3A	AF	FF	BE	DA	91	F8	21	BF	68
63821	FF	AF	B6	28	07	3A	AF	FF	BE	DA	91	F8	21	BF	68	59	09
63837	B6	28	07	3E	57	0E	00	21	AF	FF	06	07	5A	2D	11	CC	85
63853	FF	DD	7E	00	BE	CC	88	FB	DD	23	1D	5A	2D	11	CC	85	87
63869	79	BA	CC	91	FE	21	BE	FF	AB	B6	28	07	3E	57	0E	00	21
63885	FF	06	07	4A	DD	21	ED	FC	91	F8	21	DA	FF	AB	B6	28	07
63901	0D	20	F5	23	10	FD	CC	91	F8	21	DA	FF	AB	B6	28	07	1F
63917	7E	4F	21	47	FF	06	0A	7E	2B	96	B9	DA	91	F8	10	F7	02
63933	3A	D6	FF	B7	28	2A	0E	00	21	AF	FF	06	07	3E	57	0E	00
63949	BE	CC	0A	20	FF	3A	D6	FF	B9	DA	91	F8	10	F7	02	03	04
63965	06	05	13	D5	E1	7E	23	23	3C	3C	BE	AC	91	F8	10	F7	02
63981	3A	DF	FF	B7	28	21	21	AF	FF	0E	06	07	CB	46	CC	8F	89
63997	88	FB	23	10	F8	3A	DF	FF	B9	DA	91	F8	3E	07	91	4F	00
64013	3A	DF	FF	B9	DA	91	F8	3E	07	91	4F	B6	28	13	DD	2A	DB
64029	F2	DD	7E	00	BE	D2	91	F8	DD	7E	01	BE	DA	91	F8	21	0B
64045	F8	21	DA	FF	AB	B6	28	07	3A	AF	FF	BE	DA	91	F8	21	0B
64061	BE	E2	F1	F8	DD	7E	01	BE	DA	91	F8	DD	2A	FA	DD	23	B1
64077	DD	7E	00	BE	D2	91	F8	DD	7E	01	BE	DA	91	F8	3A	ED	18
64093	FD	B7	28	6C	AF	32	FF	0E	07	11	AF	FF	21	C5	FA	A4	44
64109	06	0D	1A	BE	CC	BD	FB	23	10	F8	13	0D	20	EF	21	ED	D7

Dok se sistem pravi, indikator rada u donjem desnom uglu ekrana obavestava vas, promenom boja i treptanjem, da je sve u redu. U gornjem delu ekrana računar broji kombinacije koje unosi u skraćeni sistem. Izrada sistema možete prekinuti u svakom trenutku pritiskom na S(top), a zatim se vratiti u meni radi dopune ili izmene podataka, ili na start programa. Zvučni indikator obavestuje vas o kraju izrade sistema. Tada ćete imati priliku da donosite sledeće odluke: D (dalje, opcija za razgledanje sistema), P (ponovni print, broj razgledanja nije ograničen), C (copy, kopiranje kombinacija s ekrana printerom), M (povratka u meni), R (restart, pogledajte opciju 16).

16. Restart

Vraća program na početak i briše sve ranije unesene podatke.

Nevoje u početku

Dešavaju se, naročito u početku, da vas program izvesti kako se sistem ne može napraviti. Uzrok tome će biti, najčešće, rđavo kombinovanje uslova. Na primer:

— Tražiti ste garanciju 4 od 7 za sistem od 9 brojeva. To je malo brojeva za postavljeni zahtev. Morate povećati broj brojeva ili tražiti veću garanciju.

— Odredili ste minimalni razmak 2 za sistem od 18 brojeva. Ako niste obavili zmenu, ovakav sistem mora imati najmanje 21 broj, i to za garanciju 7 od 7.

— Zahtevali ste najmanje 3 broja iz druge kolone (opcije 13), a u sistemu imate samo dva takva broja.

63/lotu na „spektrumu“

Ovo su samo tri od bezbroj mogućnosti da pogrešite. Lako su uočljive, jer se odnose na lošu kompoziciju samo dva uslova. Šta se tek može desiti kad se pogrešno ukrate uslovi iz nekoliko opcija, teško je predvideti. Zato je najbolje da se na program privikavate praveći sisteme sa malo brojeva, do 15. Rezultate ćete dobiti brzo, pa ćete lako uočiti uticaj pojedinih uslova na broj kombinacija. Tek posle toga se možete upustiti u avanturu što je pravilnije sistema od 39 brojeva sa 4.000 kombinacija.

Prethodno pripremite jastuk da odspavate dok spekturm radi. Mašinarac jeste brz, ali ovdje ima mnogo posla. Da bi ispitao sličnost dve kombinacije, mora da obavi 49 poređenja (7 puta 7). Ali pravi sistem od 39 brojeva, samo za ispitivanje sličnosti obaviće sledeći broj poređenja: broj kombinacija punog sistema (15.380.937) puta broj kombinacija skraćeno sistema (recimo, 1.000) puta 49... A gde su ostali zadaci? Na primer, preuzimanje podataka od bejzika, ispitivanje u vezi sa svim postavljenim uslovima, brojanje i memorisanje odabranih kombinacija, promena atributa u indikatoru rada, testiranje tastature, ispitivanje ostatka slobodne memorije...

Posebna mašinska rutina, po završetku rada prve, printuje kombinacije skraćeno sistema. To bi, ukratko, bio opis najvažnijih poslova koje obavija mašinar. A što se tiče bejzika, verovatno ste već uočili njegov zadatak. A što se tiče bejzika, verovatno ste već uočili njegov zadatak: da prihvati podatke o sistemu i da ih prosledi na izvršava-

nje brže od sebe, mašinaru, kao i da podnese eventualne izveštaje na osnovu kodova koje je ostavio mašinski program.

Evo i nekoliko testova za vreme: 12 brojeva, garancija 6 od 7, parnih brojeva do 4, neparnih brojeva do 4, podela na dve grupe po šest brojeva: 5 sekundi.

15 brojeva, garancija 5 od 7, fiksiran jedan broj: 24 sekunde.

18 brojeva, garancija 4 od 7, tri favorita: 1 minut i 34 sekunde.

20 brojeva, garancija 4 od 7 bez uslova: 8 minuta i 30 sekundi.

22 broja, garancija 4 od 7 bez uslova: 25 minuta i 45 sekundi.

28 brojeva, garancija 5 od 7 bez uslova: celu noć, a možda i parče jutra. Ako napravite ove sisteme možda nećete dobiti sedmicu, ali ćete bar saznati da li je ULA cip dobar, to jest da li može da izdrži dugotrajni rad bez posledica zbog pregrevanja.

Program „Sistemi za lotu“ ne daje uvijek najmanji mogući broj kombinacija. Za uzvrat, kombinovanjem više uslova možete postići izuzetno velika skraćanja.

Neke sisteme, na primer one s garancijom tipa 6 od 6, 5 od 5 itd, kao i sisteme koji se pišu na sistemskim listićima (po osam ili više brojeva u kombinaciji), ovaj program ne može da napravi. No, to ne znači da već nisu napisani programi koji izrađuju i takve sisteme. Zagrizenim loto-hakerima autor, preko redakcije „Računara“, stoji na raspolaganju za savete i uputstva (podrazumeva se, bez naknade).

Žarko Vukosavljević

U oštroj konkurenciji na tržištu računara proizvođači se na svaki način trude da im proizvod bude što jeftiniji, često zaboravljajući na kvalitet i standarde. Setimo se komputera sa samo jednim kilobajtom RAM-a, raznih membranskih tastatura ili „amstrada“ sa svojim troinčnim disk-drajvom. Apolutni šampion u škrtačenju je, bez sumnje, ser Klavj Sinkler. Nestandardni priključci i „revolucionarne“ tastature su već ušle u poslovice, ali se manje zna o tome da, recimo u „spektrum“ ugrađuje nelspravne memorije. Memorijeke čipove od 32 K ne proizvodi niko u svetu (a ni kod nas), već su to memorije od 64 K kojima je jedna polovina nelspravna. Tako je ovaj engleski inovator dobio plimečku titulu, a „spektrum“ ostao sa 48K RAM-a. Odatle ideja da mu se doda još 16K i tako upotpuni adresni prostor procesora Z80 na 64 K RAM-a.

Osamobitni procesor može da adresira maksimalno 64 K memorije, a mi želimo da „spektrum“ ima 64K RAM-a, plus 16K ROM-a. Znači, neizbežno je preklapanje dodatnog RAM-a i ROM-a, što automatski znači da dodatna memorija neće biti pristupačna iz bejzika. Prema tome, ako još niste savladali mašinski jezik procesora Z80, od ove gradnje nećete imati nikakve koristi, ali ako ga znate, onda će vam itekako dobro doći.

Proširenje memorije je zamršeno kao kartica koja se priključuje na „spektrumov“ konektor. Električna šema je data na slici. Upotrebljene su dve staričke CMOS memorije tipa 6264, kapaciteta po 8K, koje su nešto skuplje od dinamičkih, ali im je upotreba mnogo jednostavnija zbog lakšeg vezivanja u kolo (nisu potrebni selektori tipa 74157) i nema potrebe za osvežavanjem. Pored memorijskih čipova, potrebna su još tri TTL integrisana kola.

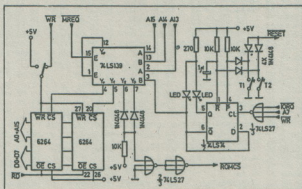
Dvostruki dekodir 74LS139 je upotrebljen za uključivanje jednog od dva SRAM-a ili ROM-a i to na sledeći način: ulazi A i B dekodera, uz ustov da je kontrolni ulaz E na logički nuli, određuju koji će od četiri izlaza biti takode na nivou logičke nule. Tabela istinitosti za ovo kolo to objašnjava.

E	B	A	Y0	Y1	Y2	Y3
1	x	x	1	1	1	1
0	0	0	1	1	1	1
0	0	1	1	0	1	1
0	1	0	1	1	0	1
0	1	1	1	1	0	0

Oznake upotrebljene u tabeli znače:

- 0 — nivo logičke nule
- 1 — nivo logičke jedinice
- x — nivo logičke nule ili jedinice, nebitno je.

Kad procesor proziva neku od adresa koja se nalazi u donjih 16K njegove memorijske mape, izlazi MREQ, A15 i A14 na nuli, što dovodi jedan od



izlaza dekodera (Y0 — Y3) na nivo logičke nule. Koji će od njih to biti zavisi od stanja na ulazima A i B (nožice 2 i 3). Ulaz B je vezan za izlaz Q flip-flopa 74LS74. Kada je ovaj, na nivou nule, uključen je SRAM. Izvod sa konektora A13, vezan na ulaz A dekodera (nožica 2), određuje da li će biti uključeno donjih 16K SRAM-a. ROM će biti pozvan kad ulaz B dođe na nivo jedinice. Diodno i kolo registruje ako je neki od izlaza Y2 ili Y3 na nuli i to se preko dvostrukog invertora sa 74LS27 prenosi na kontrolni ulaz ROM-a, ROMCS na konektoru. Prilikom uključivanja komputera flip-flop se resetuje, izlaz Q se postavlja na nivo jedinice i uključuje se „spektrumov“ ROM. Prelazak na dodatni RAM se može ostvariti softverski i hardverski. Mašinska instrukcija OUT (127), A bez obzira na to šta se nalazi u akumulatoru, vrši prebacivanje sa ROM-a na RAM i obrnuto, već zavisi od toga šta je uključeno u trenutku davanja naredbe.

Hardversko uključivanje RAM-a izvodi se pritiskom na taster T1. Time se flip-flop setuje (Q dolazi na nulu i uključuje RAM) i istovremeno se resetuje procesor. Sada, znači, počinje da se izvršava program od nulte adrese u RAM-u. Naravno, potrebno je pre pritiska na taster upisati neki program u tu

memoriju. Hardversko uključivanje ROM-a nije moguće a da pri tome ne dođe do resetovanja računara. Pritiskom na taster T2 uključujemo ROM i istovremeno resetujemo kompjuter (što će izbrisati sadržaj kompletne memorije osim 16K SRAM-a na proširenju) pa zbog toga treba tastere T1 i T2 fizički odvojiti da ne bi došlo do slučajnog pritiska na destruktivni T2.

Led diode nisu neophodne za rad uređaja ali su to zbog indikacije uključenosti RAM-a odnosno ROM-a.

Ostaje još da objasnimo prisustvo preklopnika PR koji očigledno služi da (kad se prebaci na položaj +5V) spreči upisivanje u RAM — praktično pretvara RAM u ROM! Taj preklopnik je potreban ako želite da eksperimentišete sa „spektrumovim“ ROM-om. Sigurno nećete odolati da njegov sadržaj upišete u RAM i ispravite bagove koje su ostavili Sinklerovi programeri, da popunite poslednji kilobajt koji su ostavili neiskoristi. Upisivanje ROM-a u RAM možete da izvedete, recimo, ovakvim programom.

```

DI
LD HL, 16384
LOOP DEC HL
LD A,(HL)
OUT (127), A
LD (HL), A
OUT (127), A
LD A,H
    
```

OR L
JR NZ, LOOP
EI
RET

Sada je sadržaj RAM-a ekvivalentan sadržaju ROM-a i sledeća OUT 127.x bejzik instrukcija bi isključila ROM i uključila RAM, otkada bi trebalo da se nastavi rad „spektrumovog“ operativnog sistema. Ali to se ne dešava. Iz nekog (samo Sinklerovim programerima poznatog razloga), ROM negde upisuje nešto na adrese u okviru prvih 16K što, naravno, njemu ne može da promeni sadržaj, ali ako je sada tu RAM dešavaju se čudne stvari. To je najjednostavnije izbeći tako što ćete posle prebacivanja sadržaja ROM-a u RAM preklopnik PR prebaciti iz položaja WR u položaj +5V i tek onda dati OUT 127.x instrukciju. Preklopnik možete da koristite i u drugim slučajevima kad želite da izbegnete mogućnost da (recimo, usled greške u nekoj vašoj rutini) dođe do brisanja programa u SRAM-u.

Ovom gradnjom nećete dobiti samo puko proširenje memorije već, pre svega, jednu korisnu alatku koja će vam znatno olakšati razvoj mašinskih programa. Ako u dodatni RAM spremite recimo GENS i SORSFAJL vašeg programa, više neće biti potrebe da ih zbog toga što se računarski blokirao ili resetovao, ponovo učitate sa trake. Dovoljno će biti da pritisnete taster T1, u deliću sekunde prebacite program iz SRAM-a u RAM, i odmah se posvetite traženju greške.

Pošto je ovo memorijsko proširenje namenjeno, uglavnom, iskusnijim korisnicima „spektruma“, ne dajemo nikakav softver za podršku. Kartica je višestruko primerljiva i verujemo da će svako prilagoditi ovaj uređaj i softver za njega svojim potrebama. To ne važi za vlasnike interfejsa 1, jer ovaj dodatak nije kompatibilan s njim.

Miroslav Milošević



Šta ima novo

PROM III EPROM

Iako verovatno mnogim čitaocima UV EPROMi izgledaju kao (skoro) idealne memorijske komponente, pa se sa pravom (!) pitaju kako i zbog čega uspevaju da se održe bipolarne jednokratno programabilne PROM memorije (jednom programiraj, i gotovo), verovatno su zaboravili da se UV EPROMi prave do maksimalnih brzina pristupa reda 35 ns (firme MMI, Harris, Signetics i druge). Sada se nude i CMOS PROM-ovi, iste brzine, ali sa manjom potrošnjom i širim temperaturnim opsegom (firme Cypress, Idt, AMD i dr.)

Opet NEC

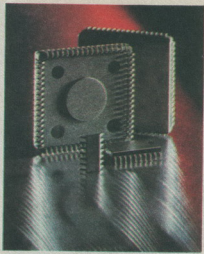
Još se nije osušila štamparska boja od prethodnog broja „Računara“, u kojima smo pomenuli V20/V30/V40/V50 familiju 8-16 bitnih mikroprocesora, a sada nalazimo informacije da Nippon Electric Company (NEC) nudi još složenije modele V50/V70. To su 32-bitne mašine realizovane CMOS tehnikom širine linije 1,5 mikrometara, i sadrže oko 375 000 tranzistora na čipu. Svojom velikim adresnim prostorom (4 Gbajta virtualne memorije) i zaštitnim mehanizmom vsoma liče po arhitekturi na INTEL-ove modele IAPX286 i IAPX386, pogotovu što su u pripremi i pripadajuća kola kao klock drajver generator, BUS kontroler, DMA kontroler, procesor za matematiku pomičnog zarezaja itd., kakva već postoje kod INTEL-ove familije.

Neki novi diskovi

Flopi diskove prečnika 2,5 inča nudi Hitachi Maxell, sa kapacitetom oko pola megabajta, namenjen upotrebi u prenosnim tekst procesorima. Matična firma (Hitachi) je proizvela disk drajver potreban za ovu novu vrstu „flopića“. Sami diskovi su dimenzije 6x6 cm, a gustina zapisa je 200 traga po inču (oko 50% gušće od postojećih 3,5 inčnih diskova kapaciteta 1 Mbajt)

BB uvek oduševljava

Naravno, ovde se radi o poznatoj firmi Burr-Brown koja nudi fantastične A/D i D/A konvertore: uzimao kao primer nove D/A pretvarače namenjene za upotrebu u audio uređajima (generisanje zvuka sintetičkim



putem) sa oznakom PCM54 i PCM55. Ovi D/A C se napajaju sa +/-12 od +/-5 V, sadrže internu referencu, laserski trimovanu (podešenu) otpornu mrežu, brze strujne prekidače i izlazne pojačavače. Uz garantovanu 15-bitnu monotonost koda, ove modele karakteriše fenomenalno niska tipična diferencijalna greška od 0,001%, dinamički opseg reda 96 dB, smirivanje naponskog izlaza za manje od 3 mikrosek, smirivanje strujnog izlaza u okviru +/-0,006% pune skale za samo 350 ns, dispacija ispod 200 mW, a ukupna harmonička izobličenja su garantovano ispod 0,0025% pri punom ulazu, odnosno ispod 0,02% pri nivou od — 20 dB. I sve to za oko 15 dolara, zavise od tipa i pakovanja, impresivno, nema šta!

Vreme je novaci

Proizvođači elektronskih LSI komponenta nude širok izbor integrisanih kola služe kao časovnik/kalendari (termin: Real Time Clock), napr. Motorola i Hitachi nude MC146818 (HD146818 respektivno), National Semiconductor nudi MM58167/-MM58174/MM58274. Oki nudi MSM5832, itd. Obzirom da su predviđena za rad i po nestanku električne energije (Battery BackUp), po pravilu su izrađena CMOS tehnologijom sa potrošnjom reda mikroprens sadržava sve potrebne elemente osim kvarc kristala i trimmer-kondenzatora za fino podešavanje frekvencije. Sada Epson (iličino i Seiko) nudi IC sa oznakom RTT-5832! koje već sadrži kristal (32-kHz) podešeno je u fabrici na tačnu učestanost, sa garantovanim odatupanjem od 10 ppm (deset delova u milion). Ova IC nisu skupi, pa danas skoro da i nema personalnog računara koji nema ugrađen neki RTC.

Protiv prislušivanja

CODEC (COder/DECoder) je element, kao što znamo, koji analogni signal pretvara-kodira u digitalni niz, u cilju prenosa putem digitalnih veza, a zatim ponovo restorira-dekodira niz u analogni signal. Ovak način prenosa ima višestruke prednosti: veća imunost ne smetnje, digitalni signal se lako obrađuje računarima, pogodnim izborom načina kodiranja može se signal šifrirati (sigurnosni aspekt), prenos svetlovdnom načinom (sigurnosni aspekt). CODEC tipa S2842 firme Silitronica koristi eksponencijalnu promen-

ljivu delta modulaciju (algoritam) napajanje od 5 — 8V uz potrošnju od nekoliko desetina miliampera, a izrađeno je kombinacijom analogne i digitalne (rL) tehnologije. Brzina uzimanja uzoraka ide od 150 Kbps (tipično: potrebno je oko 40 Kbps za telefoniju, a oko 100 Kbps za radio-difuziju).

NEC, ponovo

Prema tvrdnji firme NEC Electronics, njima pripada čast da su prvi tržištu ponudili potpuno statičku RAM kapaciteta 256 Kbita. Integrirano kolo sa oznakom UPD43256 ima 32Kx8 organizaciju, napaja se sa 5 V, ali se čuvanje podataka garantuje i pri samo 2 V, a može se dobiti u tri brzinske verzije: 100, 120 i 150 ns. Bazirano, čip koristi NMOS memorijska ćelije u matrici, a CMOS perifernu i pomoćnu logiku, uz cenu reda 100 US dolara.

AMD je „od reči“

Američka firma Advanced Micro Devices (AMD) je krajem 1985. obećala da će svoje nedelje tržištu ponuditi novo, traženo LSI integrirano kolo (!). I to ne nešto jednostavno, nego kompleksna kola kao što je npr. AmB151 grafički čip za kolor paletu. Ovo LSI IC se koristi u vrhunskim grafičkim terminalima rezolucije 1280 x 1024 (preko 2.500.000 tačaka), radi sa video signalom reda 200 MHz, ima izbor od 256 boja iz paleta od oko 16,8 miliona boja, i slično. Za sada, AMD poštuno „drži“ reč.

$E^2 = MC$
Možda se Ajnštajn ne bi složio sa ovom formulom, ali Motorola dokazuje da je EPROM pravi programski STORAGE za MC mikrokontrolere, iako su neke varijante MC68HC11 serije sadržavale, pored EPROM-a i RAM-a, i 512 do 2K EEPROM memorije, tek pojava kontrolera MC68HC805C4 sa 4K bajta EPROM-a, u DVA nezavisna segmenta nudi mogućnost da softver „popravlja“ sam sebe. Naime, dok se program izvršava iz segmenta A (EPROMA), simultano se može menjati programski kôd u segmentu B, pod programskom kontrolom! Uz ostale široke sposobnosti ovog MCU (RAM, 24 I/O linije, tajmer, UART, SPI, itd.), Motorola predviđa sjajnu budućnost ovom čipu.

Poboljšana 6116

Popularna statička RAM tipa 6116 (Hitachi) se može dobiti u raznim verzijama: tako firma interesantnog imena, Vitelco (ovo znači Very Intelligent Integrated Circuits) nudi brzu verziju (vreme pristupa oko 55 ns), kao i štedljivu verziju (stend bajt potrošnja oko 2 mikroampera). Oznaka modela V61C16, sa cenom reda 10 US \$.

Koristna zaštita

Ako ste do svog CPU i ostalih IC teško došli (a i u drugim slučajevima), preporučujemo VAM da svoje štampano kolo zaštitite od naponskih kratkotrajnih skokova (tranzijenata) komponentima tipa TranZorb TM (Transient Absorber) koje pod oznakom P6KE proizvodi firma General Semiconductor (a i drugi). Ovi elementi, ne veći od obične diode, mogu u kratkotrajnom vremenu da apsorbuju tranzijentne snage reda 600W, a proizvode se za različite napone i različite energije.

Priprema: Blažimir P. Miše, dipl. ing.

USIJANI JOYSTICK

Dobili smo nekoliko pisama u kojima se žalite da su igre u našem časopisu zapostavljene, da koncepcija rubrike nije baš najbolja, itd. Odgovaramo svima o jednom trošku:

Koncepcija igara, istina, još uvek nije ustaljena, pa kvalitet rubrike varira od broja do broja, ali polako počinje da poprira konačan oblik. Igre koje predstavljamo u časopisu su, obično, još nepoznate kod nas, i mi ne želimo da detaljno ulazimo u opise potpuno novih igara, šta će ostati igrači ako „im mi unapred sve otkrijemo“.

Demon and Soft nas pita šta se radi u igri „Sex games“. Uzgred, on pita još stvari, ali je ovo najbolje pitanje. E, pa lepo. U toj igri roda nosi bebe preko kupusista, a sve u svrhu obrazovanja malih demona. Ahrrm, pitaj nekog od tvojih drugara, ako ne znaš sam.

Vasić Aleksandar iz Bora nam šalje svoja iskustva iz „Elite“.

Vojislav Ristić iz Zemunja pita zašto zapostavljamo igru „Tau Ceti“, i čemu služi notes za beleške, kao i kako da spaja delove hladnjaka.

Dragi Vojkane, i mi smo se već zabrinuli zašto se niko ne javlja u vezi ove briljantne igre. Grafika u njoj je toliko bolja od one u „Eliti“, koliko je „amiga“ bolja od „spektruma“. O.K.

Notes ti služi za beleženje gradova kroz koje si prošao. Seti se kako si manipulisaš delovima pucl u „Impossible mission“, pa to primeni i za spajanje delova hladnjaka (rods). Važno je da da simaš status u svakom mestu, jer, ako te unište roboti, kreće iz početka. U tome ti slabo pomaže POKE, koji je skoro objavljen u jednoj našoj reviji, jer blokira LOAD opciju i tako postaje beskoristan.

Milan Stojković iz Zrenjanina piše da se najviše igra sa „Turbo Espritom“ i da ga interesuje nekoliko stvari:

1. Kakav je to zvuk koji se čuje kada naletiš na pešaka?
2. Zašto se igra završava kada pobije sve dilere i zaradi oklopna kola?
3. Kako da iskoristi onih 30 milja (kilometara?) na brojčaniku brzinsome? Maksimalna brzina je 150, a piše 180.

Evo i odgovora:

1. Žalosno, ali treba da znaš da se pešaci, NE gaze. A to što čuješ to oni vrište od bola i straha...
2. Igra se završava, jer novu preprodavci nemaju zbog čega da dodu u taj grad...
3. Nikako, to je štos koji se i inače koristi na pravim automobilima. Na primer, koji običan „kec“ može da ide 160 km/h?

Smatramo da ste ovoga puta bili vrlo lenji u pisanje, što, se lepo može videti iz rubrike. Zato, olovke u šake i pišite nam o svojim iskustvima. Ne očekujte odgovor na baš svako pitanje, jer se ni naši saradnici ne igraju baš stalno i sa svim igrama.

„Čip“ u „Računarima“

u saradnji sa radio-emisijom „Čip i sedam jedinica“ koja se emituje svake subote na televiziji Radio-Beograda u 12 časova

Komodor

DRAGON'S LAIR (ZMAJEV BRLOG)

Software Projects



Dok Sveti Đorđe ubiva aždahu u Atejuju 212, i dok vaši dokoni prijatelji ganjuju ale, vama se zalomio zmaj. Debeli žutooki zmaj koji je oteo vašu dragu i odveo je u svoj brijog iza sedam gora, sedam mora i sedam nivoa. Vi, naravno, pripasujete mač i krećete u pustolovinu. Put vas vodi do podzemnih lamnica, kuda se spuštate srednjovekovnom varijantom lifta: predmetom veoma sličnim poklopcu za bačvu. Dalje idete Hodnikom lobanja, pa kroz suroviji verziju Kolo-seuma, gde se tarzanski prebacujete konopcima iznad razbuktale vatre, pa preko krovova nekih ultramodernih građevina od plave cigle, i tako stvari teku, po rečima Mlaka Dizdara, sve lude i sve huđe, dok ne stignete (tako uopšte stignete) do zmaya i ustanovite da ga i nije teško ubiti. U poređenju sa raznovrsnim aspidama koje su vas dotle ometale, on je maltene najgњеše. Prošnjate se pored njega, prikoljete ga mačem koji vas čeka na kraju puta — i ako mislite da je tu priči kraj, u pravu ste.

DRUID

Firebird



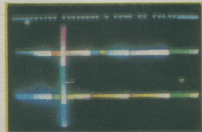
Druidi će se osloboditi karakondzula koje su im okupirale zemlju, ako se prošunjate kroz osam nivoa i uništite četiri lobanje. Niste golouruki: imate na raspolaganju sedam vrsta čini. Morate ih koristiti strateški, inače od oslobodjenja (i vašeg života) nema ništa. Pomoću njih dobijate vatru, vodu, struju, Golema, haos, nevidljivost i ključ. Vatra uništava skoro sve neprijatelje; struja

takođe, ali manje efikasno; voda deluje samo na odabrane. Golem vam je telesna garda i uništava sve dušmane; haos uništava lobanje (i sve ostalo što se zateklo na ekranu), pa mu ni to nije malo nego vam još obnavlja energiju. Nevidljivost i ključ jasni su na prvi pogled. Imate samo jedan život, kao i u stvarnosti od koje ste pobegli u ovu igru.

Ako vam nije jasno otud električne i davnim mitskim vremenima, ili kako je Golem stigao iz jevrejskih mitova u keltske, pripremite se za kartežarjanje. Odgovor ne zna niko. Ne budite kartežarjanac.

HYPERFORCE (HIPERSILA)

Arlosoft



Ako ste videli makar jednu Kleovu sliku, setičete je se odmah. Ovakve bravure sa nijansama umeo je (dosad) samo on.

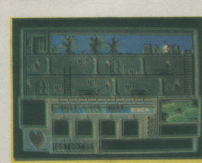
Ako ste makar jednom bili ispaljeni iz puške, setičete se i tog doživljaja. Brzina kojom ste tada leteli nije ni prineli brzini „Hipersila“.

Sve ostalo biće vam novo. Naći ćete se među apstraktnim oblicima, upravljate oštećenim startronom (zvezdotronom, ako vam prevod ista objašnjava), skupljate male kontejnere, pucati u zidove-smetala ili ih vući unatrag, tražite predmete koji vam produžavaju život, izbegavate predmete koji ga skraćuju — i ludo se provodite ako volite brzinu. Drugog lonako nema ničeg; ni druida, ni druida, ni duhova, ni robota, ni čudovišta, ni karatista, ni vitizova, ni kostura, ni laserskog oružja, ni raketa.

Odmorite se.

TIME TRAX (VREMENSKI TOKOVI)

Mind Games



Komplikovano do zla boga.

Putujete kroz vreme da pokupite razbacane dragocenosti i vratite ih na mesto, tražite četiri pločice ispisane runama (tek kad ih nađete, spasi ste sve!), barataste svakojakim oružjem boreći se protiv svakojake gamadi; ukratko, potucate se od nemila do nedraga kroz dvadeset jednu vremensku zonu (tri ekrana puta sedam zona). Uz to, svaki ekran ima po tri života.

Tako vam i treba kad hoćete da spasavate čovečanstvo.

Gotovo polovinu ekrana zauzimaju indikatori, ikonostas, digitalni sat, ležišta za one četiri pločice sa runama, itd. Morate redovno ulaziti u spisak opcija da biste se snašli u igri, i dobro pamtili da li ste ušli u Juče ili u Sutra, „ili nešto gdje obojeje“. Može vam se desiti da i u stvarnom životu primetite da je svanulo Sutra dok ste se probijali kroz igru započetu Sinoć.

KNIGHT GAMES

(VITEŠKE IGRE)
English Software



Ovde bi briljirao Dejjli Tompson, da je kojim slučajem živeo u srednjem veku i da je bio vlastelin. Doduše, nije reč o dekatlonu nego o oktoatlonu iiliti osmoobju: mačevanje 1, tuča motikama, gađanje strelom (u drvene kočice, ne u protivnika), makljanje buzdavanima, cepanje ha-lebardama, mačevanje 2, gađanje samostrelom i sečenje bojnim sekirama. Od vas se traži da budete što sličniji Milošu Vojnoviću.

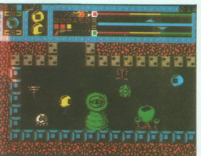
Ambijenti se menjaju, da vam ne bude dosadno, umesto sata siluži zapaljena sveća (vaš se život gasi zajedno s njom), a neki položeni rombovi — ili su to dijamanti? — obavestavaju vas o energiji vašoj i protivnikovoj.

Na marljivo naslikanoj pozadini vitezozi su animirani vešto, a nacrtani načinom srednjovekovnim, to jeste nezgrapno. Živopisac — videničke to dimah — nije ni prineti svojim studeničkim kolegama.

Spektrum

EQUINOX (RAVNODNEVICA)

Mikro-Gen



Splet hodnika pun je kanistera. U kanisterima je nešto nuklearno. Kad istekne vreme, to Nešto

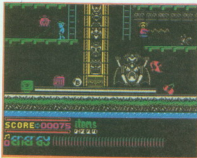
eksplođira. Vaše je da blagovremeno posklanjate kanistera. Ometaju vas razne karakondžule.

To je sve.
Zašto onda (RAVNODNEVICA)? „Tko zna, ah nitko, nitko ne zna) krhko je znanje,“ odgovara Dobriša Cesarić. Ako ste se već našli u hodniku, ne pitajte se ništa, nego tražite propusnice i budice, dižite smadze u vazduh, idite iz jednog nivoa u drugi (i natrag), pucajte iz sve snage, istražujte koji su predmeti korisni na kom nivou, plaćajte plavim žetonima za vožnju, koristite teleporte, ustupite mesto starijim osobama i mislite na iznenadna kočenja.

Na izvestan posredan način uverite se da je deponovanje nuklearnih otpadaka velika glavo-bolja, i da se treba prikloniti čisto i besplatno solarnoj energiji.

DYNAMITE DAN II

(DAČA DINAMIT II)
Mirrorsoft



Igrati kompjuterske igre znači raditi Sizifov posao. Taman ste u prvom DD likvidirali ludog naučnika — kad, ne izez vraže, pojavi se DD II i vi sa užasom videte da je on vaskrsnuo. Sad ga morate ganjati po osam ostrva, a svako se ostrvo prostire na 24 ekrana, a na svakom ekranu vrebava trista smrti...

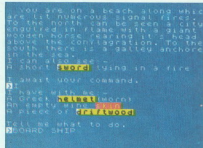
... ali se ipak, verujem, nećete pitati: „Ko me gurnu?“ Scenario je opak, ali maštovit; morate biti veoma domišljati. Uvrnuta i stilski čisto izvedena scenografija pokazuje da imamo posla sa sjajnim crtačem i koloristom.

Na sedam ostrva treba doneti pluću od džuboksa i odsvirati je, potom načeti cepelin, napuniti ga gorivom i odmagliti; na osmom čete džuboksa dići u vazduh i vratiti se u cepelin za tri minuta, inače ginele. Protiv naučnika, preprika i karakondžula pomažu, pre svega, tamne naočari, baterijska lampa (da vam aspide ne bi kratile predmete), i mikser za hranu. Igra je puna raznih predmeta; neki su vam korisni a neki ne, baš kao u pravom životu.

RETURN TO ITHACA

(POVRATAK NA ITAKU)
Atlantis

Oni koji su pročitali „Odiseju“ imaju prednost: napredovaće kroz ovu avanturu ne pitajući se ko su Lestrigion ili šta je Kirkin mali hobi. S druge strane, svaka adaptacija, po logici stvari, podrazumeva izvesno odstupanje od originala — a ovde ima izmena posle kojih ni sam Homer ne bi mogao prepoznati svoj spev. Prema tome, budite pažljivi: ne čudite se što čete kod Lotofoga morati ući u lavirint i odande uzeti bure vina, i ne pokušajte da Polifemu dodete glave Odisejine u načinom.



ima i detalja koji se u „Odiseji“ — i u stvarnom životu — podrazumevaju, ali ovde ne. Tako, kad nepog prislanete, ne zaboravite da otkucate „DROP ANCHOR“, tj. „SPUSTI SIDRO“ — inače će vam brod otploviti nekud bezbedno, pa ćete igru izgubiti!

Ako Posjedon nadjača Atenu pa ne stignete na Itaku, ne tugujte. Penelopa je ionak dvadeset godina starija no što je bila kad ste krenuli u trojanski rat.

BOBBY BEARING (KIČA KUGLICA)

The Edge



O, da. Još jedna varijanta na temu: pustolovni kliker. Otkako je SPINDIZZY povukao nogu (ako se to za kuglu može reći), nakotilo se Kireš i Baundera koliko vam duša hoće.

Ovog puta, jedna od onih kuglica kakve se prave u IKL-u polazi na put da nađe četiri svoje brata i jednog rođaka; sva petorica, naime, came u ropstvu zlih kugličnih ležajeva. Ravnija se dočava u apstraktnom trodimenzionalnom svetu kochi, paraleloipeda i prizmi. Nade se i pokoji magnet; kad on povuče kuglicu, ne nadajte se dobru. Treba pomerati kockice, čuvati se cigalke koje padaju s neba, propadati kroz sahtove, itd., itd., sve je to već videno.

Da li? Ovakvu grafiku nismo videli. Sjajna je. Kad treba, daje nam utisak savršene sferičnosti — što, pri rezoluciji kakvu ima spektrum, nijeimalo lak posao. Animacija tokom svaka česta: tožni još više: da se ova igra odigra do kraja, treba zaista imati kliker.

POKE — ČOŠAK

komodor 64

	POKE	480,173	64	11,200, 62	16
Dynamite dan	2204,173	64 <td>11,200, 12,000</td> <td></td> <td></td>	11,200, 12,000		
Equinox	2914,234	64 <td>11,200, 12,000</td> <td></td> <td></td>	11,200, 12,000		
Knights	2214,234	64 <td>11,200, 12,000</td> <td></td> <td></td>	11,200, 12,000		
Dynamite dan	2214,173	64 <td>11,200, 12,000</td> <td></td> <td></td>	11,200, 12,000		
Return to Ithaca	2204,173	64 <td>11,200, 12,000</td> <td></td> <td></td>	11,200, 12,000		
Walter pattern	1204,173	64 <td>11,200, 12,000</td> <td></td> <td></td>	11,200, 12,000		
Return to Ithaca	1204,173	64 <td>11,200, 12,000</td> <td></td> <td></td>	11,200, 12,000		

Vladimir Stojiljković

**POZIV
NA PRETPLATU**

**Već 15 godina vaš pouzdani
informer o nauci i tehnici
kod nas i u svetu**



GALAKSIJA

POPUST 20%

**PRETPLATA JE NAJBOLJI, NAJSIGURNIJI I NAJJEFTINIJI NAČIN NABAVKE
NAŠEG ČASOPISA**

**ISKORISTITE SPECIJALNI POPUST ZA GODIŠNJU PRETPLATU I ISTOVREMENO
SE ZAŠTITITE OD DALJIH POSKUPLJENJA**

PREDNOSTI PRETPLATE

- manja cena (2.000 umesto 2.400)
- garantovana cena
- sigurna nabavka
- dostava na kuću

**KADA SE PRETPLATITE, NEĆETE VIŠE MISLITI NA „GALAKSIJU“; ONA ĆE
MISLITI NA VAS!**

„Galaksija“ je vaša najveća, najpouzdanija i najlepša enciklopedija nauke i tehnike. Za petnaest godina izlaženja na prosečnom tiražu od 55.000 štampano je ukupno blizu deset miliona primeraka časopisa. Objavljeno je približno 12.000 članaka i otprilike 25.000 crno-belih i kolor ilustracija — od čega bi moglo da se načini oko 60 ilustrovanih monografija o nauci i tehnici.

GALAKSIJA

NAUKA I TEHNIKA OD KAMENOG DOBA DO KOSMIČKE ERE

GALAKSIJA

VAŠ VODIČ KROZ SVET NAUKE