

mr LIDIJA POPOVIĆ MOMIR POPOVIĆ

commodore ***i/o***

1985.

recenzenti:

STANKO STOILJKOVIĆ
glavni i odgovorni urednik "SVETA KOMPJUTERA"

MILAN CABARKAPA
profesor programiranja OVRD"Veljko Vlahović"

obrada na tekst procesoru:

MILKA LOVRIC i M. POPOVIC

grafički urednik:

STEVAN PAKOVIC

izdavač:

RO za projektovanje i inženjering
BEOGRAD BIRO, Mladenovac
Trg Ive Lole Ribara 32/a

štampa:

GRD GRAFOSREM, Sid
Branka Erića 9

Drugo izdanje
tiraž: 3.000 primeraka

Rešenjem br. 413-1379/85-06 od 28.08.1985 Republičkog komiteta
za kulturu, a na osnovu člana 36. stav 1. tačke 7. i 35. Zakona
o oporezivanju proizvoda i usluga u prometu, knjiga je oslobo-
djena poreza na promet.

U V O D

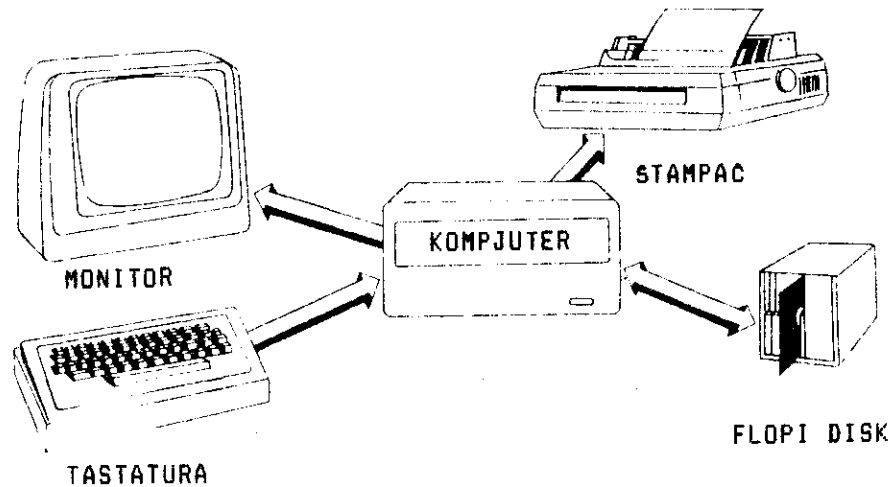
Rad autora ove knjige preko časopisa SVETA KOMPJUTERA, pokazao je u kontaktima sa sirom čitalačkom publikom, da postoji veliki interes za knjigom koja bi pokrivala napredniji nivo programiranja, vezan za rad periferijskih uređaja. Praksa je pokazala, da mnogi korisnici posle prvog zanosa vezanog za rad sa kompjuterom, ne znaju kako da ga iskoriste za obavljanje nekog korisnog zadatka. S druge strane, mnoge radne organizacije su se odlučile za uvođenje kompjutera u njihovo poslovanje putem mikro kompjutera, kao prve faze. Većina literature na našem jeziku, vezana za kompjutere, uglavnom se bavi nekim opštim upoznavanjem sa kompjuterima i raznim osnovnim kursevima BASIC-a. Informacije vezane za rad periferijskih uređaja, kao osnovnog preduslova bilo kakvog ozbiljnijeg rada, da ne kažemo, profesionalnog, skoro da je nemoguće naći (ne računajući povremene članke po časopisima).

U želji, da svojim skromnim doprinosom pokušamo da popunimo prazninu iz ove oblasti, napisali smo ovu knjigu. Nismo želeli da se bavimo nekim fiktivnim periferijskim uređajima u cilju neke opštosti i univerzalnosti (iz kojih se ne može baš previše naučiti za konkretan sistem), pa smo odabrali da pišemo o periferijskim uređajima vezanim za COMMODORE kompjutere. Iskustvo koje imamo u radu, je pokazalo da su COMMODORE kompjuteri (pre svih se to odnosi na C-64), zastupljeni u velikoj meri i kao kućni kompjuteri, i kao mali poslovni sistemi. Iako smo svakim danom sve više bombardovani sve boljim i sve jeftinijim kompjuterima, činjenica je da će se C-64 još dugo zadržati kao zvezda na našem kompjuteskom podneblju. A s druge strane, iskustva koja imaju naši severni susedi Madjari u primeni C-64, kao poslovnog sistema (bilo u administraciji ili u nekom istraživačkom institutu), još više nas je učvrstilo u uverenju da će ova knjiga naći svoj put do čitaoca.

Sama knjiga je primer kako kompjuter i I/O uređaji mogu da se koriste u razne svrhe. Umesto da se knjiga slaže na klasičan način ili fotoslogom, kompletna priprema štampe je uradjena uz pomoć TEKST PROCESORA, (rezultate možete proceniti sami). Dovoljno je ilustrativan i podatak, da se cela knjiga koju držite u ruci čuva na jednoj disketi od 320KB, za sve eventualne kasnije izmene i dopune.

Uz knjigu možete dobiti kasetu sa svim većim i značajnijim programima, da biste mogli maksimalno vremena posvetiti aktivnom radu umesto viščasovnog unošenja velikih programa. Uputstvo za korišćenje programa nalazi se na kraju knjige.

PERIFERIJSKI UREDJAJI



SI. A - 1 TIPIČAN KOMPJUTERSKI SISTEM

Na gornjoj slici je prikazan jedan tipičan kompjuterski sistem, opremljen osnovnim perifernim uređajima:

- tastatura
- monitor
- flopi disk
- stampac
- kasetofon

Periferni uređaji se u kompjuterskom žargonu često nazivaju I/O uređaji (INPUT/OUTPUT = I/O), da bise odmah naglasila njihova osnovna funkcija. Jer, INPUT na engleskom znači ulaz, a OUTPUT izlaz sa stanovišta protoka informacija. Otuda i naslov knjige COMMODORE I/O. Drugim rečima, pod perifernim uređajima podrazumevamo sve uređaje koji omogućavaju komunikaciju čoveka sa kompjuterom. Bez ovih uređaja, kompjuter bi bio sasvim beskorisna mašina. Šta više, danas se glavni napori konstruktora raznih modela kompjutera (bilo sa hardverske bilo sa softverske strane), usmeravaju na obezbeđenje maksimalnih mogućnosti za priključenje što više perifernih uređaja. Može se slobodno reći da uspešnost nekog modela kompjutera, možemo ocenjivati preko podrške perifernih uređaja.

Bitna osobina svakog I/O uređjaja je način komunikacije sa kompjuterom koji može biti:

- jednosmeran
- dvosmeran

Jednosmerna komunikacija odražava mogućnost slanja ili primanja signala od kompjutera. Pa prema tome, kao tipične predstavnike jednosmernih I/O uređjaja možemo izdvojiti:

- štampač koji samo prima signale od kompjutera i pretvara ih u štampani izveštaj
- tastatura koja samo šalje signale kompjuteru

Na sl. A - 1, gde je prikazan tipičan kompjuterski sistem, tastatura je namerno izdvojena kao poseban periferni uređjaj (što je slučaj kod većih i boljih sistema), jer ljudi često tastaturu identifikuju sa kompjuterom, obzirom da je kod većine kućnih kompjutera tastatura smeštena u isto kućište sa centralnom jedinicom (CPU), što je slučaj i sa C-64, VIC-20, +4 i još nekim COMMODORE modelima. Rad sa tastaturom je u principu vrlo jednostavan, i krajnje sličan radu sa tastaturom pisane mašine. Osnovna razlika je u tome, što tastatura za kompjuter ima niz kontrolnih tipki vezanih za upravljanje radom samog kompjutera. U svakom slučaju najvažnija tipka na tastaturi je tipka RETURN, koja ostavlja direktnu vezu sa kompjuterom.

Dvosmerna komunikacija je osobina vezana za periferni uređjaj koji su u stanju da primaju ali i da šalju podatke kompjuteru. Ova osobina je vezana uglavnom za I/O uređjaje predviđene za čuvanje programa i podataka, kao što su:

- kasetofon
- flopi disk
- hard disk i sl.

Monitor ili klasičan televizor (zavisno od toga šta upotrebljavate) je u principu jednosmeran I/O uređjaj, obzirom da samo prima informacije od kompjutera, koje zatim reprodukuje na svom ekranu, radi vizuelne kontrole onoga što radimo sa kompjuterom. Međutim, već se pojavljuju monitori osetljivi na dodir, pa će i ovaj tip I/O uređjaja polako preći u dvosmerne.

Spisak I/O uređjaja za COMMODORE se ni izbliza ne iscrpljuje ovim što smo nabrojali. Zato pomenimo još neke.

MODEMI - I/O uređaj pred kojim je budućnost, jer omogućava povezivanje kompjutera telefonskom mrežom. U visoko razvijenim zemljama postoji citav niz kompjuterskih sistema, povezanih u mreže (COMPUNET, PRESTEL i sl.), u koje se možete uključiti preko modema i sa vašim COMMODORE kompjuterima. Na taj način postajete korisnik ogromnih datoteka podataka, koje pružaju informacije o najrazlicitijim stvarima. Problem je samo u telefonskom računu za inostranstvo, plaćanju potrebne pretplate, kao i naknade za specijalne usluge, a sve u devizama. Zato mislimo, da bi stvaranje jedne takve mreže na jugoslovenskom nivou, bilo korak od 7 milja u oblasti kompjuterizacije, što bi nas značajno približilo razvijenim zemljama.

PLOTERI, GRAFICKE TABLE, SVETLOSNE OLOVKE, DIGITAJZERI su I/O uređaji koji omogućavaju uvidjenje mikrokompjutera u oblast inženjerstva, projektovanja i stvaranja mikro CAD sistema (Computer Aided Design - projektovanje pomoću kompjutera).

A/D ili D/A KONVERTORI analogno-digitalni ili digitalno-analogni, konvertori uvode mikrokompjutere u procesnu industriju, upravljanje raznim tehnološkim procesima, kao i njihovu primenu u naučno-istraživačkim projektima. Iskustva naših suseda, Mađara u ovoj oblasti primene C-64 kompjutera, su više nego ohrabrujuća.

SINTISAJZERI u sprezi sa mikrokompjuterima otvaraju novu dimenziju u stvaranju i reprodukcovanju muzike.

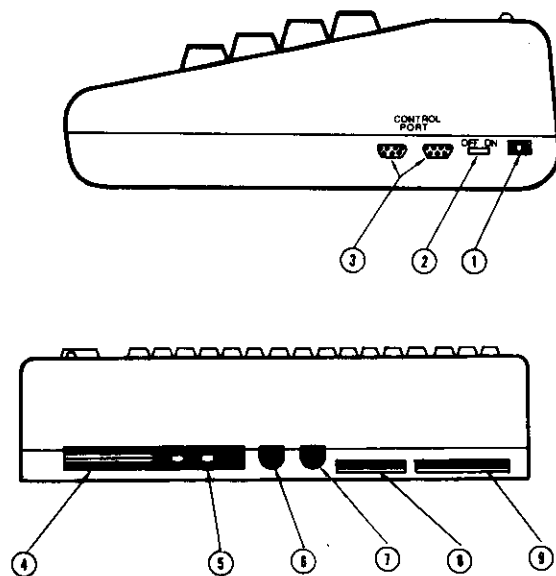
PALICE ZA IGRU, "MISEVI" i još sijaset drugih I/O uređaja samo dopunjuju listu I/O uređaja, koji se mogu vezati na vaš COMMODORE kompjuter.

Ova knjiga se zadržava na prikazu osnovnih I/O uređaja, koji se najčešće sreću u radu sa COMMODORE kompjuterima a to su:

- KASETOFON
- FLOPI DISK
- STAMPAC

Ovakva konfiguracija omogućava da C-64 može raditi, čak i u poslovnim sistemima, za razne nivoe upotrebe, uzimajući u obzir željenu brzinu pristupa podacima kao i njihovu količinu.

POVEZIVANJE PERIFERIJSKIH UREDJAJA SA C-64



1. NAPAJENJE STRUJOM
2. PREKIDAC ZA UKLJUCENJE
ON - uključen
OFF - isključen
3. CONTROL PORT (A i B)
- palice za igru:
JOYSTICK, PADDLE
- svetlosna olovaka:
LIGHT PEN
4. EXPANSION PORT
- memorijski ROM moduli:
CARTRIDGE
- CP/M modul
- IEEE 488 paralelni interfejs
- LINK modul za povezivanje više C-64 u mrežu
5. RF IZLAZ
- televizor
6. AUDI-OVIDEO IZLAZ
- monitor
- HI-FI uređaj
7. SERIAL I/O PORT (IEEE)
- CBM štampači
- C-1541 floppy disk
- CENTRONICS interfejs
8. CASSETTE PORT
- CBM kasetofon
9. USER PORT
- RS-232 interfejs
- CENTRONICS interfejs
- modemi
- A/D i D/A konvertori

Sl. A - 2 Mesta za spajanje I/O uređaja sa C-64

PRILIKOM POVEZIVANJA SVI I/O UREĐAJI I KOMPJUTER MORAJU BITI ISKLJUČENI. NEPOŠTOVANJE OVIH PRAVILA MOŽE DOVESTI DO OZBILJNIH OŠTEĆENJA, ILI C-64 ILI NEKOG OD I/O UREĐAJA.

Sa sl. A - 2 i opisa uz nju možete videti deo spiska I/O uređaja koji se mogu priključiti na C-64. Uz spisak I/O uređaja smo dodali i memorijske ROM module, koji su pre periferni dodaci za C-64 nego I/O uređaji, ali je dobro znati gde ih priključiti. Pored toga, naslovi su dati na engleskom, da biste mogli da pratite oglase i članke iz strane štampa, koji se pozivaju na određeni PORT. PORT možete prevesti kao vrata, ulaz, mesto priključka, a mi ćemo usvojiti ovaj odomaćeni kompjuterski izraz.

Opise i funkciju pinova na pojedinim PORTOVIMA nismo davali, jer ova knjiga nema za cilj davanje saveta po pitanju hardverskih prepravki i modifikacija. U svim boljim knjigama na engleskom, a pre svega u "PROGRAMERS REFERENCE GUIDE" imate potreban detaljan opis. A na našem jeziku informacije po tom pitanju možete povremeno naći u raznim kompjuterskim časopisima, a pogotovu u časopisima: "MOJ MIKRO", "SVET KOMPJUTERA" i manjim delom u "RACUNARIMA".

Napajanje strujom C-64 obavlja preko svog ispravljača koji se DIN priključkom vezuje na njega, preko ulaza 1. Obavezno vadite utikač ispravljača iz struje čim prestajete sa radom na više časova.

Ono što nas u ovom momentu interesuje je povezivanje C-64 sa:

- televizorom ili monitorom
- kasetofonom
- diskom
- štampačem

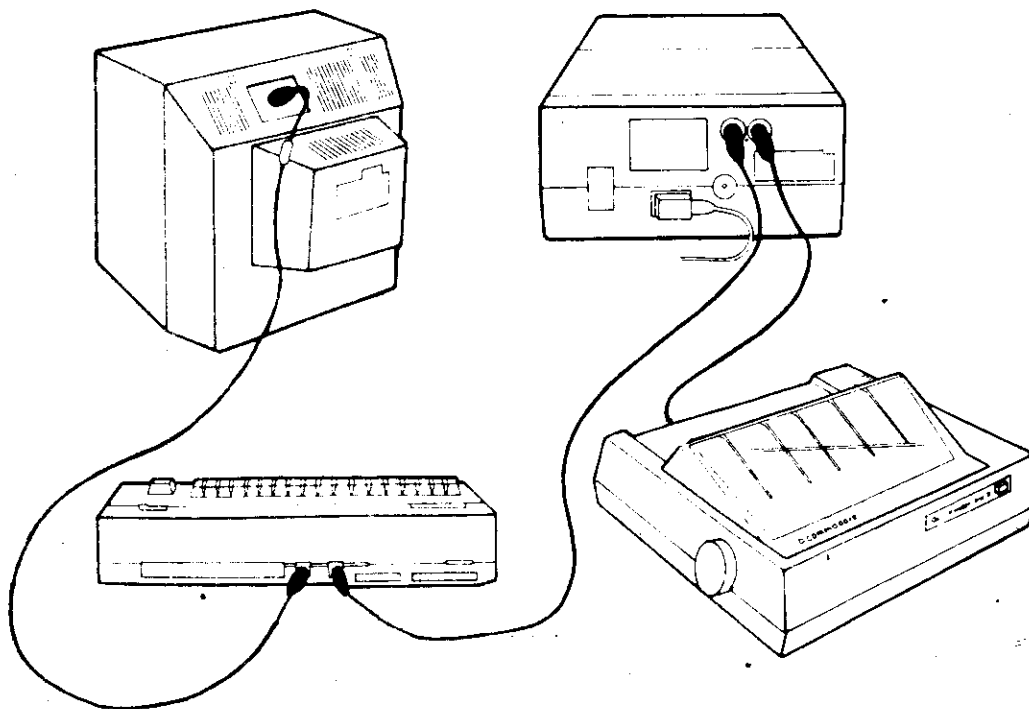
TELEVIZOR se povezuje sa C-64 na krajnje jednostavan način. Uz C-64 dobijate koaksijalni kabal, gde odgovarajući kraj uključite u C-64, a drugi kraj u TV. Uključite TV i C-64. Odaberite slobodan broj programa, koji ćete koristiti samo za C-64, prebacite birac kanala na UHF (preko njega gledate II program) i okrecite birac dok se na ekranu ne pojavi željena slika.

MONITOR se priključuje na C-64 pomoću odgovarajućeg DIN priključka na AUDIO/VIDEO IZLAZ (broj 6 na slici).

KASETOFON CBM C2N je kasetofon specijalno pravljen za COMMODORE kompjutere. Priključite ga na CASSET PORT (broj 8 na slici). Utikač kasetofona ima zarez koji omogućava priključenje samo na odgovarajući način. Zato, nemojte silom gurati utikač već proverite da li je zarez na pravoj strani. Na utikaču se nalazi i jedna metalna žičica, koja je predviđena za priključenje na uzemljenje, kod profesionalnih modela. Ako kasetofon koristite sa C-64 ili VIC-20, odrezite ovu žičicu, koja može izazvati oštećenje, ako dohvati do nekog metalnog predmeta, za vreme dok je kompjuter uključen.

CBM STAMPACI I FLOPI DISK 1541

Štampači radjeni specijalno za C-64 kao što su SEIKOSHA GP 100 VC, MPS-801, MPS-802, C-1525, C-1526, kao i disk C-1541 vezuju se na C-64 preko SERIAL I/O PORTA (broj 7 na slici), pomoću DIN kabla koji se dobija uz ove I/O uređaje. Na ovaj PORT možete priključiti do 5 disk jedinica i 1 štampač (otuda mu i naziv serijski PORT).



Sl. A - 3 Povezivanje CBM - stampaca i flopi diska C-1541

Obzirom da disk jedinica na svojoj zadnjoj strani ima dve DIN utičnice, u jednu vezujete kabal do kompjutera, a u drugu kabal do sledećeg diska ili stampaca sto mozete videti sa sl. A - 3. Ako imate samo stampac onda ga DIN kablom direktno vezujete na SERIAL PORT.

Problem u izvesnoj meri nastaje ako zelite da radite sa stampacem ili flopi diskom koji nije predvidjen za direktan rad sa C-64. Rešenje je u dodacima koji se nazivaju INTERFEJSI (INTERFACE). Interfejsi su uređjaji, koji omogućavaju povezavanje kompjutera sa I/O uređjajima. Interfejsi mogu biti:

- paralelni
- serijski

Kao što im samo ime kaže, paralelni interfejsi šalju 1 bajt preko 8 zica, gde svaka zica služi za prenos jednog bita. Serijski interfejsi za prenos podataka koriste jednu zicu, tako da se 1 bajt prenosi kroz zicu bit po bit. Ovaj pojednostavljen prikaz dovoljno govori u prilog tome, da se paralelni prenos podataka obavlja znatno brže nego serijski. To je jedan od osnovnih razloga (pored još nekoliko), zašto je COMMODORE-ov disk C-1541 sporiji nego što bi trebalo da bude. Kod štampača skoro da i nema razlike u brzini, u zavisnosti od upotrebljenog interfejsa, jer štampači rade pri takvim brzinama da je skoro nebitno kakav interfejs koriste.

Trenutno najpoznatiji interfejsi su:

- RS-232 serijski
- CENTRONICS paralelni
- IEEE 488 paralelni

Nazalost, C-64 ima ugrađenu samo jednu verziju IEEE 488 serijskog interfejsa (SERIAL PORT). Na PORTOVIMA koje smo dali, postoje potrebni pinovi za povezivanje na ove interfejse, ali priključci nisu standardni, pa je potrebno naknadno kupiti ove interfejse, ali specijalno namenjene za C-64.

O RS-232 i CENTRONICS interfejsu ćemo detaljnije govoriti u odeljku za štampače, obzirom da su oni za njih i predviđeni.

IEEE 488 paralelni interfejs se priključuje na EXPANSION PORT koji ima potrebne magistrale za adrese i podatke, pa je preko njega moguće ostvariti vezu C-64 sa znatno boljim i bržim (a samim tim i znatno skupljim) flopi diskovima kao što su 2031, 8250, 4040, 8050 i 9090. To se takođe odnosi i na neke štampače koji zahtevaju ovaj interfejs. Osnovni problem kod ovog interfejsa je što je on veoma skup, kao i potreban kabal. Ovaj interfejs je prvobitno razvijen za povezivanje kompjutera sa laboratorijskom opremom, pa otuda i njegova cena. Uzevši u obzir njegovu cenu i cenu potrebnog kabla, kao i cenu I/O uređjaja sa kojima povezuje C-64, mislimo da je za naše uslove preskupo rešenje i nedostupno većini korisnika.

K A S E T O F O N

Kasetofon je, svakako, periferni uređaj koji se najviše koristi za čuvanje programa i podataka. Bez obzira na sve nedostatke koji se javljaju u radu sa kasetofonom, on je periferni uređaj koji može korisno poslužiti u nizu primena. Cena, mogućnost nabavke i još niz drugih činilaca uzrokuje, da je najčešća početna konfiguracija KOMPJUTER + KASETOFON.

COMMODORE je napravio malo neuobičajen postupak kada su u pitanju kasetofoni. Predvideo je direktnu mogućnost upotrebe, samo kasetofona koga prodaje uz dodatnu cenu. Međutim, za uzvrat nismo dobili ono što se iz takvog opredeljenja može očekivati, a to je pre svega brzi prenos podataka. COMMODORE to pravda ustupkom radi sigurnosti u prenosu podataka, što sigurno ne možemo uzeti zdravo za gotovo, ako pogledamo rešenja drugih proizvođača. Jedno je ipak sigurno, da smo na ovaj način oslobođeni problema oko rada sa raznim kasetofonima (vlasnici SPECTRUM-a znaju to mnogo bolje), ali smo ostali bez stotinjak maraka u džepu. I pored svega toga, mnogi zaljubljenici elektronike se vrlo često interesuju da li postoji način da se COMMODORE kompjuteri vežu za običan kasetofon. Naravno, da je i to moguće, pa za one koji žele da se upuste u takav poduhvat, uz malo znanja elektronike i lemljenja, preporučujemo članak MATJAZA KLJUNA "Priklučenje kompjutera COMMODORE-64 na običan kasetofon ("MOJ MIKRO", januar 1985.). U članku je dato uputstvo sa šemom i potrebnim delovima za gradnju interfejsa, potrebnog za povezivanje C-64 i običnog kasetofona. Pa, ko voli, nek' izvoli.

UPUTSTVO ZA RAD

Ono što svakog novopečenog vlasnika interesuje je, kako spojiti uređaj sa kompjuterom i kako otpočeti sa radom. Što se tiče samog spajanja pogledajte poglavlje POVEZIVANJE PERIFERIJSKIH UREĐAJA SA KOMPJUTEROM. Još jednom napominjemo, za vašu sigurnost, da sva povezivanja kompjutera i perifernih uređaja (pa i kasetofona), treba izvoditi kada su svi uređaji isključeni iz struje. Svako eksperimentisanje na ovom planu, može dovesti i do ozbiljnih oštećenja.

Što se tiče rada sa kasetofonom i načina njegovog korišćenja, može se reći po onoj narodnoj: "Koliko para, toliko muzike". To u osnovi znači, da je rad sa kasetofonom relativno jednostavan, ali da zato od njega ne možemo očekivati čuda. Međutim, moramo napomenuti, da se u poslednje vreme sve više čuje, kako je

kasetofon spravica koja služi samo u početku rada sa kompjuterom, a da bilo kakav ozbiljniji rad isključuje njegovu primenu. Tu se svakako pravi greška, jer kasetofon može poslužiti i u profesionalne svrhe:

- stvaranje kopija važnih programa (tzv. BACK UP programa), koje možemo izgubiti nepažljivim rukovanjem sa diskom ili disketama
- arhiviranje odredjenih podataka i sl.

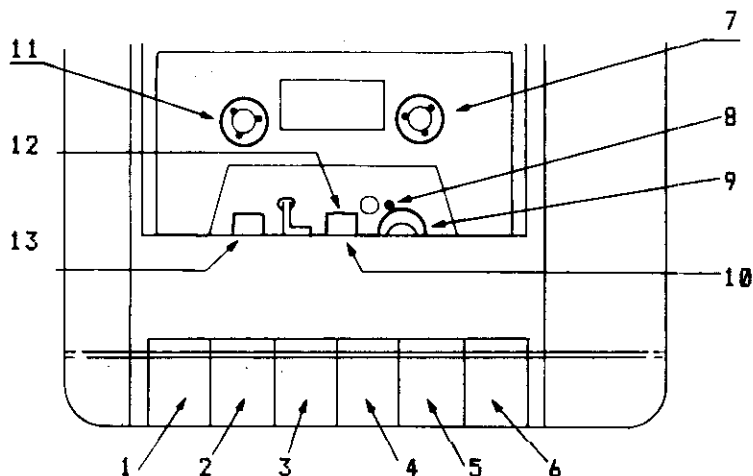
Jednom rečju, kasetofon nam može poslužiti da rasteretimo diskete, od podataka koji nam neće trebati duži vremenski period, ili da pravimo kopije važnijih programa, koji se na mnogo raznih načina mogu oštetiti na disketi, koju upotrebljavamo. Pravljenje kopija svih važnih programa i podataka sa kojima radimo, je jedno od osnovnih pravila profesionalnog rada, koje nije suvisno poštovati ni u kućnom radu. Samo jedan mali pogrešan korak dovoljan je, da čak i višemesečni trud bude upropašćen u jednom trenutku i bude zauvek izgubljen.

Aktiviranje kasetofona

COMMODORE kasetofon je pod direktnom kontrolom samog kompjutera, pa cim uključite kompjuter, aktivirali ste i kasetofon koji je spreman za dalji rad. COMMODORE preporučuje da prilikom uključivanja kompjutera istovremeno držite pritisnutu tipku C=, kako bi time aktivirali i kasetofon. Praksa je, međjutim, pokazala da taj postupak nije neophodan, ali vas ništa ne košta da ga poštujuete.

Provera ispravnosti

Svako, ko je bar jednom rukovao sa običnim kasetofonom za snimanje i reprodukciju muzike, neće imati nikakvih problema pri proveri ispravnosti COMMODORE-ovog kasetofona C2N. Kao što vidite sa sl.B-1, kasetofon C2N ima sve osnovne komande kao i svaki drugi audio kasetofon, pa se nećemo posebno zadržavati na njegovom opisu. Proveru ispravnosti rada kasetofona (sa stanovišta samog uređjaja), možete sprovesti na uobičajen način. Bitno je napomenuti, da proveru ispravnosti snimanja možete uraditi, samo ako je kasete u kasetofonu. Ako pokušate da pritisnete tipke PLAY i RECORD, bez kasete u kasetofonu, osetićete otpor koji vas upozorava da taj postupak možete učiniti samo kada je kasete u kasetofonu. Zato preporučujemo, da izbegavate ovaj postupak bez kasete.



- | | |
|----------------------------------|-------------------------------|
| 1 - RECORD (Snimanje) | 7 - Pehvatni točkic |
| 2 - PLAY (Učitavanje) | 8 - Pogonska osovinica |
| 3 - REWIND (Premotavanje unazad) | 9 - Frikcioni točkic |
| 4 - F.FWD (Premotavanje unapred) | 10 - Glava za sniman./učitav. |
| 5 - STOP (Zaustavljanje rada) | 11 - Pogonski točkic |
| 6 - EJECT (Izbacivanje kasete) | 12 - Prostor za snim./učit. |
| | 13 - Glava za brisanje |

SI B-1 K A S E T O F O N

UCITAVANJE PROGRAMA

Kao i svaki drugi uređaj za čuvanje podataka, kasetofon može da služi za:

- 1 - arhiviranje programa
- 2 - arhiviranje datoteka podataka

U početnim koracima, kasetofon služi uglavnom za prvu svrhu, dok tek kasnije korisnik počinje da radi i sa datotekama podataka. O osnovnim principima rada sa datotekama podataka na kasetofonu, govorićemo u kasnijim poglavljima. Sad ćemo se zadržati na učitavanju i snimanju programa.

Rad sa kasetofonom u ovom smislu, svodi se na korišćenje tri osnovne komande:

LOAD SAVE VERIFY

LOAD komanda je verovatno jedna od prvih komandi sa kojima se sreće korisnik kompjutera. Obzirom, da se uz kompjuter nabavi i neka igra ili gotov program, potrebno je učitati taj program sa kasete u kompjuter, da biste videli šta on radi. Kompletna sintaksa komande LOAD je:

LOAD "ime programa", uredjaj, sekundarna adresa

ime programa - parametar kojim se definiše naziv programa koji želimo da učitamo. Ako ubacimo ovaj parametar, kompjuter će preotavati sve dok ne nađe program sa traženim imenom, koga će zatim učitati u kompjuter

uredjaj - parametar kojim definišemo uredjaj sa kojim kompjuter treba da komunicira. Ako se izostavi ovaj parametar, kompjuter pretpostavlja da treba da radi sa kasetofonom. Za kasetofon je predviđen broj 1 a za disk broj 8

sekundarna adresa - ako se ovaj parametar ne definiše, C-64 će učitati program i smestiti ga počev od 2049 adrese u memoriju (tako se učitavaju svi BASIC programi). Ako postavite sekundarnu adresu 1, C-64 će učitati program u iste memorijske lokacije odakle je program i snimljen. To se pre svega odnosi na mašinske programe, koji nisu striktno vezani za određene memorijske lokacije, kao što je to slučaj sa BASIC programima. Ako ne znate tip programa koji učitavate a sumnjate da je mašinski, slobodno definišite sekundarnu adresu

PRIMERI:

LOAD

Učitava se prvi program na koji se nađje na kaseti, posle pritiska na tipku PLAY na kasetofonu.

LOAD "TEST"

Posle pritiska na tipku **PLAY**, kompjuter premotava traku sve dok ne nadje program sa nazivom **TEST**, koga tada ucita.

LOAD "MASINAC",1,1

Kompjuter ucitava program **MASINAC** i smesta ga u iste memorijske lokacije, odakle je program snimljen na kasetu. Potrebno je, radi onih koji prvi put rade sa kompjuterom, opisati postupak ucitavanja programa. Posto otkucate naredbu:

LOAD "PROGRAM1"

i premotate traku kasetofona na odgovarajuće mesto (sluzeci se brojačem okretaja), pritisnite tipku **RETURN**. Nakon toga, kompjuter ce na ekranu javiti poruku:

PRESS PLAY ON TAPE

sto znaci, da treba pritisnuti tipku **PLAY** na kasetofonu. Kada to uradite kompjuter vam na ekranu daje poruku:

**OK
SEARCHING FOR PROGRAM1**

sto znaci da trazi program sa imenom **PROGRAM1**. Kada na traci naidje na zeljeni program, na ekranu daje poruku:

FOUND PROGRAM1

time vas obaveštava da je nasao program sa imenom **PROGRAM1**. Kompjuter ceka odredjeno vreme a zatim nastavlja proceduru ucitavanja. Vi ga mozete ubrzati pritiskom na tipku **C=**, ili **CTRL** ili **RAZMAKNICU**. Nakon toga kompjuter ce vam dati poruku:

LOADING

sto znaci da je počeo ucitavanje programa. Kada završi ucitavanje javlja poruku:

READY

Time je završeno ucitavanje programa i sada mozete startovati ucitani program, editovati ga i menjati po sopstvenoj zelji.

Na kraju treba pomenuti još jednu mogućnost učitavanja programa sa kasetofona. Istovremenim pritiskom tipke RUN/STOP i C=, kompjuter vam javlja:

PRESS PLAY ON TAPE

Kada pritisnete PLAY učitace se prvi program sa kasete, na koji kompjuter naidje i izvršice se njegovo automatsko startovanje.

Radi vaše informacije napominjemo, da za vreme učitavanja programa nemate nikakvu sliku na ekranu niti zvuk (poznata karakteristika SPECTRUM-a). Ako baš volite poznati piskavi zvuk pri učitavanju kompjuterskih programa, DENIS PAP je dao jednostavnu semu za dodavanje malog zvučnika na vaš kasetofon u reviji "MOJ MIKRO", jul 1985., str. 30. Stručnjaci za lemljenje i prepravke mogu opet da stupe u akciju.

Povezivanje programa sa prenosom parametara

Sve ovo što smo prethodno rekli, uglavnom je dosta poznata stvar. Medjutim, ima nekoliko tajni koje su vezane uz LOAD naredbu i koje mnogi od korisnika ne znaju ili ne umeju da koriste. Iako osnovni BASIC, C-64 kompjutera ne može ni izbliza da se podiči, nekom od osnovnih mogućnosti strukturiranog programiranja (što je iz SIMON'S BASIC-a veoma lako), to ne znači da ne možete raditi sa više raznih programa i prenositi parametre iz jednog u drugi.

Kada koristite LOAD naredbu u direktnom modu, kompjuter prethodno obriše postojeći program u memoriji i sve promenljive koje su deklarisanе programom. Medjutim, ako želite da sa vrednostima, prethodno izračunatih promenljivih, udjete u potpuno nov program (što je već na izvestan način strukturano programiranje), tada je potrebno da koristite LOAD naredbu u programskom modu, npr. u memoriji kompjutera imate sledeći program:

```
100 FOR X = 1 TO 10
110 Y(X) = X * X
120 NEXT
130 LOAD "KOREN"
```

a na kasetofonu imate program "KOREN":

```
100 FOR X = 1 TO 10
110 Z(X) = SQR(Y(X))
120 NEXT
```


Kada budete startovali program koji se nalazi u memoriji on će izračunati kvadrate brojeva od 1 do 10 i rezultate smestiti u niz Y(X), tada će program tražiti od vas da pritisnete PLAY na kasetofonu (PRESS PLAY ON TAPE), i nakon toga će učitati program "KOREN" koji će se automatski startovati i izračunati kvadratne korene svakog člana prethodno izračunatog niza i rezultate smestiti u niz Z(X). Da ste program "KOREN" učitavali u direktnom modu, time biste poništili prethodno izračunate vrednosti niza Y(X) (na vrednost 0), pa bi program izračunavao stalno kvadratni koren od 0. Na ovaj način možete velike programe koji koriste znatnu količinu memorije za razne nizove i promenljive, podeliti u manje logične celine, koje bi povezivali LOAD naredbom, kao zadnjom programskom linijom (ovakav princip rada dosta je poznat i primenjuje se i kod IBM PC kompjutera). Morate paziti da u programima, koje učitavate, ne dimenzionisete prethodnim programom već dimenzionisane nizove.

Osnovna prednost ovakvog načina rada je, da možete programirati manje celine (koje je lakše testirati) i da zatim prenosite parametre iz jednog u drugi program. Medjutim, mnogi će možda odustati od ovakvog načina rada ako ih bude obeshrabrila brzina (sporost) kasetofona. Ipak, uzmite u obzir da će te, kad tad, preći na disk i tad će te ovaj princip rada moći mnogo komotnije da upotrebljavate, imajući na umu njegove velike prednosti. Zato se pripremajte radeci sa kasetofonom za prelazak na disk (cena diska - u markama - pada iz dana u dan). Postupak je potpuno identican, samo je u naredbi LOAD potrebno definisati broj uređaja 8.

Učitavanje mašinskih programa

Učitati i startovati BASIC program nije neka naročita mudrost. Medjutim, šta ako ste dobili neki mašinski program ili neku korisnu rutinu? Znaete da je učitete sa:

```
LOAD "RUTINA",1,1
```

ali kako startovati ovaj program, ako ne znate adresu sa koje je snimljen. Zato nije na odmet, da se upoznate sa par sistemskih promenljivih vezanih za rad sa kasetofonom, koje vam mogu koristiti na više načina.

Od 828 - 1019 memorijske lokacije u C-64 i VIC-20 nalazi se tzv. KASETNI BAFER, koji kompjuter koristi za čuvanje važnih informacija prilikom učitavanja programa. Ovih 192 bajta mogu se inače slobodno koristiti u razne svrhe, kada ne koristite

kasetofon (za smestanje neke manje mašinske rutina i sl.). Medjutim, kada kompjuter učitava/snima program ili datoteku podataka on koristi KASETNI BAFER na sledeci način:

LOKACIJA	KORISCENJE
828.....	Tip datoteke: 3 - program, 2 - podaci
829.....	Startna adresa-bajt niže značajnosti (LOW BYTE-LB) za LOAD/SAVE
830.....	Startna adresa-bajt više značajnosti (HIGH BYTE-HB) za LOAD/SAVE
831.....	Krajnja adresa-bajt niže značajnosti za LOAD/SAVE
832.....	Krajnja adresa-bajt više značajnosti za LOAD/SAVE
833 - 1019.....	Ime programa ili datoteke (na ime datoteke kompjuter dodaje blanko znake da popunu bafer do 1019 adrese)

Tab.B-1 - Opis KASETNOG BAFERA

Sad se postavlja pitanje, kako iskoristiti ovo što ste saznali. Mogućnosti su razne, a za početak da vidimo kako iskoristiti informacije iz kasetnog bafera za učitavanje mašinskih programa.

Kao što verovatno znate, mašinski program se ne može startovati RUN komandom, kao što je to slučaj sa BASIC programom, već komandom oblika:

SYS broj memorijske lokacije

Drugim rečima, moramo znati startnu adresu, gde je smesten mašinski program, da bismo ga mogli startovati. Medjutim, komandom LOAD "RUTINA",1,1, kompjuter je učitao mašinski program sa nazivom RUTINA, smestio ga u memoriju na lokaciju odakle ga je i snimio i dao poruku na ekranu:









READY

Sada smo u čudu, znamo da je program negde u memoriji, a ne možemo da ga startujemo. RUN komanda ne pomaže jer je program mašinski a ne znamo startnu adresu, gde je kompjuter smestio program. Spas je u informacijama iz **KASETNOG BAFERA**.

Za momenat ćemo naparaviti izlet u binarne brojeve. Pošto je C-64 osmobična mašina, on u svakom od svojih 65536 bajtova (64 * 1024 = 65536), jer jedan kilobajt ima 1024 bajtova), može da smesta brojeve između 0 i 255.









Binarni brojevi, bajtovi i pointeri

Zamislite bajt kao osam sijalica. Ona, koja je upaljena označava broj 1, a koja nije broj 0, sasvim logično, zar ne?

vrednost	0	1	0	1	0	0	0	0	
sijalica									LB
red.br.	7	6	5	4	3	2	1	0	

$$0 \cdot 2^7 + 1 \cdot 2^6 + 0 \cdot 2^5 + 1 \cdot 2^4 + 0 \cdot 2^3 + 0 \cdot 2^2 + 0 \cdot 2^1 + 0 \cdot 2^0 = 80$$

$$0 + 64 + 0 + 16 + 0 + 0 + 0 + 0 = 80$$

vrednost	1	1	0	0	0	0	1	1	
sijalica									HB
red.br.	7	6	5	4	3	2	1	0	

$$1 \cdot 2^7 + 1 \cdot 2^6 + 0 \cdot 2^5 + 0 \cdot 2^4 + 0 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 = 195$$

$$128 + 64 + 0 + 0 + 0 + 0 + 2 + 1 = 195$$

$$LB + HB \cdot 256 = 50000$$

Sl.B-2 - Sematski prikaz POINTERA

Ako malo pažljivije pogledate prethodnu sliku, verovatno, će vam sada biti jasnije zašto je 11000011 u binarnom sistemu, jednako 195 u dekadnom sistemu, kao i da je 01010000 u binarnom sistemu, jednako 80 u dekadnom sistemu. Ako bi sve sijalice bile ugašene, vrednost bajta bi bila jednaka 0. Ako bi sve bile upaljene, po prethodnoj analogiji, vrednost bajta bi bila jednaka 255. Jedno je sasvim sigurno, da kompjuter ne može smestati u svoje bajtove, brojeve veće od 255.

Svima onima, koji su se upustili malo u sferu mašinskog programiranja, ovo je prva lekcija koju moraju naučiti. Raznim matematičkim metodama i principima elektronike kompjuter može, koristeći brojeve od 0 do 255, da množi, deli, stepenuje, radi sa alfabetskim simbolima, crta i radi još mnogo drugih stvari. Da bi se sve ovo objasnilo, bila bi potrebna nova knjiga a pitanje je da li bi se jednom knjigom i moglo sve da objasni.

Medjutim, nemojte se obeshrabriti na samom početku. Covek može programirati i raditi sa kompjuterom, a da ni iz bliza ne mora znati sve te principe. Zašto bi, uostalom i postojali viši programski jezici kao što su BASIC, PASCAL, COBOL, FORTRAN i sl. Ipak, nije loše poznavati neke osnove rada sa binarnim brojevima, kao i principe smeštanja podataka raznih oblika, u bajtove. Ovaj kratki izlet u binarne brojeve, vam je bio potreban da biste naučili, kako kompjuter pamti adrese od 0 do 65535 (upravo toliko ih i ima vaš C-64), a da pri tome u jedan bajt može stati najveći broj od 255. Konstruktori računara su to rešili vrlo jednostavno i elegantno.

Uzmimo, na primer, da je startna adresa naše mašinske rutine koju smo učitali, na adresi 50000. Da imamo sesnaestobitni kompjuter, tada bi to moglo da stane u jedan bajt od 16 bita:

$$\begin{array}{r} 1100001101010000 \\ \text{HB} \quad \text{LB} \quad (2) \end{array} = 50000 \quad (10)$$

Posto je vaš kompjuter (nažalost) osmobicna mašina, konstruktori su prvih osam bitova (s leva u desno) sesnaestobitnog bajta proglasili za bajt više značajnosti (HB), a zadnjih osam bitova za bajt niže značajnosti (LB), i potreban bajt od 16 bita smestili u dva bajta od 8 bita. Da biste, sada, tačno odredili adresu na koju ukazuju ovi bajtovi, potrebno je uraditi sledeće:

- pomnožiti vrednost koja se nalazi u bajtu više značajnosti sa 256 (zbog toga je i značajniji od svog suseda)
- na dobijenu vrednost treba dodati vrednost koju sadrži bajt niže značajnosti

To jednostavno može da se predstavi formulom:

$$\text{HB} * 256 + \text{LB} = \text{potrebna adresa}$$

U primeru sa sijalicama namerno je odabran slučaj, koji odgovara našem primeru, gde je gornji niz sijalica bajt niže značajnosti, a donji red bajt više značajnosti. Da biste izračunali memorijsku

RAD SA KASETOFONOM

lokaciju, na koju ukazuju ova dva bajta, potrebno je uraditi sledeće:

$$195 * 256 + 80 = 50000 \quad \text{znači} \quad \text{HB} = 195 \quad \text{LB} = 80$$

Potrebno je, da vam napomenemo još jednu stvar, C-64 uvek smešta u memoriju prvo LB bajt, pa onda HB bajt. Na izgled malo nelogično, ali takva je konvencija. U literaturi se ova dva bajta nazivaju raznim imenima, npr. **POINTERI**, **VEKTORI**, **LINKOVI** i sl. U osnovi njihov zadatak je da ukažu na određenu adresu. Obzirom da će te se u sledećim poglavljima o datotekama, često sretati sa ovim bajtovima, mi smo odlučili da usvojimo izraz **POINTER**. Ako neko više voli naše izraze, najbliži prevod bi bio **UKAZIVAC**. Ali, pošto kod nas nema neke usvojene konvencije o prevodjenju kompjuterskih izraza, mislimo da je bolje neke izraze koristiti u originalu, nego ih silom i rogobatno prevoditi.

Posle ovog kratkog slaloma kroz binarne brojeve, bajtove i pointer-e, vratimo se našem učitanom mašinskom programu sa nazivom **RUTINA**. Nadamo se, pogledom na tab. B - 1, možete i sami da uočite, kako biste rešili problem, tj. otkrili na koju startnu adresu je kompjuter učitao pomenuti mašinski program. Ako ne možete sami, pomoci će vam sledeća linija, otkucana u direktnom modu:

```
SYS(PEEK(830)*256+PEEK(829))
```

i mašinski program će se startovati. Kao što vidite, pointeri na adresi 829 i 830, ukazuju na koju startnu adresu je kompjuter smestio mašinski program. Ako je naša pretpostavka tačna, da je startna adresa na 50000, tada je:

- u bajtu 829 smestena vrednost 80
- u bajtu 830 smestena vrednost 195

jer je $195 * 256 + 80 = 50000$.

Pošto se BASIC programi uvek smeštaju od startne adrese 2049, proverite vrednost bajtova 829 i 830 kada radite sa BASIC programom, pa će te se uveriti sadrže vrednosti 1 i 8 respektivno, jer je: $8 * 256 + 1 = 2049$

Za momenat vas podsećamo da BASIC naredbom **PEEK(adresa)**, uvek možete ispitati sadržaj bilo koje lokacije u memoriji vašeg računara.

Kao što ste videli iz prethodnih par stranica, informacije iz kasetnog bafera su nam pomogle da odgonetnemo startnu adresu masinskog programa, koji smo učitali naredbom LOAD "RUTINA",1,1 a koja nam je prethodno bila nepoznata.

Spasavanje izgubljenih programa

Prilikom svakog snimanja programa na kasetu, C-64 pravi dve kopije vasesg programa. To je jedan od razloga duzeg snimanja ili učitavanja programa. Druga kopija programa služi da se prilikom učitavanja programa, proverí njegova ispravnost. Ovo naizgled, može biti nepotrebno gubljenje vremena, ali će vas u odredjenom momentu i te kako obradovati ova osobina. Iako je snimanje programa na kasetu dosta sigurno, ipak vam se može desiti da prilikom učitavanja nekog programa dobijete poruku na ekranu:

? LOAD ERROR

To znači, da je prilikom učitavanja programa došlo do neke greške. Prvi korak, koji se najcesce primenjuje, je ponovno učitavanje programa. Medjutim, ako ponovo dobijete istu poruku, nastaju problemi. To se pre svega odnosi na neke važne programe, na koje ste utrosili dosta truda i do kojih vam je veoma stalo. Tada počinje da vam se mota kroz glavu misao, gde je greška. Uzroka može biti više, a neki od njih su:

- posle duze upotrebe glava za upisivanje/citanje programa se uprljala ili namagnetisala
- doslo je do nekih fizičkih ostecenja trake usled prašine ili se traka malo istegla i sl.

Prvi uzrok je lako otkloniti. Kupite kasetu za čiscenje i demagnetizaciju glave kasetofona i resili ste problem. Ako je u pitanju drugi uzrok, problem i dalje ostaje, ali se u odredjenim situacijama može rešiti.

Kada snimate vas program na kasetu, kompjuter to radi u tri zasebne celine:

- 1 - snima na kasetu tzv. heder (header - zaglavlje) u koje upisuje tip programa, startnu adresu odakle je uzeo program i krajnju adresu do koje se program prostirao kao i naziv programa
- 2 - snima na kasetu celokupni program
- 3 - ponovo snima na kasetu ceo program kao kopiju, koju će koristiti za proveru prilikom učitavanja

Kada učitavate program, heder programa (kao što već pretpostavljate) ide u kasetni bafer. Prva kopija programa se učitava u memoriju, a pomoću druge kopije, kompjuter utvrđuje da li su obe kopije identične. Ako je sve u redu kompjuter javlja poruku READY, a ako nije LOAD ERROR.

Ukoliko ste se ranije nervirali zbog sporosti kasetofona, sada znate, da je jedan od glavnih razloga za to duplo snimanje programa. Sada vam ova osobina može spasiti program. Informacije koje je kompjuter smestio u kasetni bafer će vam ponovo značajno pomoći.

Posle smo učitali BASIC program, kompjuter unapred zna startnu adresu. Medjutim, BASIC interpreter mora znati gde je kraj programa, jer odatle pa na dalje u memoriji on smešta prvo promenljive pa onda nizove. Pogledajmo te pointerne:

VARTAB	45	-	početak BASIC promenljivih, LB
	46	-	početak BASIC promenljivih, HB
ARYTAB	47	-	početak BASIC nizova, LB
	48	-	početak BASIC nizova, HB
STREND	49	-	kraj BASIC nizova, LB
	50	-	kraj BASIC nizova, HB

51. B-3 BASIC pointeri

VARTAB, ARYTAB, STREND su BASIC pointeri koji su namešteni da ukazuju na kraj BASIC programa pre nego što ga startujemo. Kada je kompjuter učitao program i pronašao grešku, on namerno ne postavlja ove pointerne na kraj BASIC programa jer "smatra" da je program pogrešan. Medjutim ako u direktnom modu otkucate:

```
FOR X=45 TO 49 STEP 2: POKE X,PEEK(831): POKE X+1,PEEK(832): NEXT X
```

resetovali ste pointerne BASIC programa na ispravne vrednosti, koristeći informacije iz kasetnog bafera. Sada možete izlistati program, proveriti da li je ispravan i ponovo ga snimiti na ispravnu kasetu.

Potrebno je napomenuti nekoliko stvari:

- Kada dobijete poruku LOAD ERROR, nemojte pokušavati da listate program pomoću naredbe LIST, jer posle toga nećete moći da primenite prethodni postupak. Izlistajte program tek posle primene prethodno navedenog postupka
- Ako se desilo da je greška bila u prvoj kopiji programa, verovatno će te morati naći greške u programu i ispraviti ih pre nego program ponovo snimate na ispravnu kasetu
- Ako je greška na kaseti bila u drugoj kopiji programa, nakon primenjenog postupka program će raditi bez ispravki
- Ako je greška bila u hederu, onda je ceo problem mnogo teži ali ga je moguće rešiti. Treba da znate, kako se u memoriji kompjutera pakuje BASIC program. Posto ova oblast ne spada u domen knjige, preporučujemo vam članak iz "SVETA KOMPJUTERA" ("Mala tajna - velika pomoć", februar 1985.), gde ćete naći upitstvo za rešenje ovog problema.

Kao što ste iz prethodnog videli, moguće je spasiti program, kada ste mislili da je već izgubljen. Iz prakse nam je poznato da mnogi vlasnici C-64 i VIC-20, koriste tzv. TURBOTAPE programe, koji znatno ubrzavaju snimanje i učitavanje programa. Ponekad se brzina rada kasetofona, kada se koristi ovaj program, izjednačava sa brzinom rada diska, što znatno povećava mogućnost rada sa kasetofonom. Međutim, morate znati da se tada programi ne snimaju u dve kopije i da je mogućnost greške znatno veća, jer se u tom slučaju na centimetar dužine trake smešta znatno više informacija, čime se uvećava mogućnost greške usled fizičkih oštećenja na traci.

Zato preporučujemo, da vaše važne programe snimate jedanput u normalnom modu, a da radnu verziju koju slobodno koristite snimate u TURBOTAPE modu.

Na kraju ovog odeljka napominjemo, da je napred opisani metod moguće koristiti i za brze učitavanje programa snimljenih u normalnom modu. Ako imate neki dugacak program, koji je potrebno učitavati više minuta, ceo postupak možete prepoloviti. Kada učitavate takav program, pratite brojač okretaja na kasetofonu, i čim predje par okretaja više od polovine, prekinite učitavanje pritiskom na RUN/STOP tipku.

Time ste sprečili kompjuter da proverava ispravnost učitanoog programa sa snimljenom kopijom. U direktnom modu otkucajte napred navedeni program, koji resetuje pointere i posle toga možete normalno startovati program sa RUN.

Radi svake eventualnosti, za one koji nisu u to sigurni, napominjemo da direktni mod predstavlja izdavanje naredbi i komandi kompjuteru koje on odmah izvršava. To znači da treba da otkucate naredbu ili više naredbi odvojenih sa : i da pritisnete tipku RETURN. Kompjuter će odmah izvršiti vaše naredbe. Kada ispred vasih naredbi dodate broj programske linije, kompjuter to pamti kao program i spreman je da ih izvršava tek posle komande RUN. Zato se takav način rada zove programski mod.

SNIMANJE PROGRAMA

Dosad smo uglavnom govorili o učitavanju programa sa kasete i varijacijama na tu temu. Sada ćemo govoriti o suprotnoj aktivnosti SNIMANJU PROGRAMA na kasetu.

Snimanje programa pisanih u BASIC-u se obavlja na jednostavan način komandom:

SAVE "ime programa", broj uređjaja, sekundarna adresa

IME PROGRAMA - naziv programa pod kojim želite da ga snimate
BROJ UREĐJAJA - 1 za kasetofon, 8 za disk, ako ga izostavite kompjuter pretpostavlja da želite da radite sa kasetofonom
SEKUNDARNA ADRESA

- 1 - kada kasnije učitavate ovaj program, kompjuter ga smešta na iste memorijske lokacije odakle ga je snimio (za mašinske programe)
- 2 - ako postavite sekundarnu adresu na 2, kompjuter će snimiti program na kasetofon sa postavljanjem EOF markera na kraj programa (END OF TAPE MARKER - marker koji simulira kraj kasete). Ako kasnije pri pretraživanju kasete za određenim programom kompjuter naiđe na EOF marker, on prekida dalje pretraživanje
- 3 - sekundarna adresa koja kombinuje mogućnosti obe prethodne sekundarne adrese

Primeri za primenu komande SAVE:

SAVE

Snima program na kasetu bez imena.

SAVE "PROGRAM"

Snima program na kasetu sa imenom PROGRAM.

SAVE "PROGRAM",1,1

Snima program na kasetu sa imenom PROGRAM i pamti startnu i krajnju adresu odakle je uzeo program zbog kasnijeg učitavanja.

SAVE "PROGRAM",1,2

Snima program na kasetu sa imenom PROGRAM i postavlja na kraju programa EOF marker.

SAVE "PROGRAM",1,3

Kombinuje prethodna dva primera.

Naredbu SAVE mozete koristiti u okviru programa, npr.:

```
100 SAVE "PROGRAM"
```

Kada kompjuter izvršava program i naidje na naredbu SAVE, prekida se izvršavanje programa dok se ne obavi snimanje. Posle snimanja programa, kompjuter nastavlja izvršenje prve sledeće naredbe.

Sama procedura snimanja programa je krajnje jednostavna i identična je klasičnom snimanju na kasetofonu. Kada izdate naredbu SAVE kompjuteru, on vam na ekranu javi poruku:

```
PRESS PLAY & RECORD ON TAPE
```

i čeka dok se istovremeno ne pritisnu tipke PLAY i RECORD na kasetofonu. Za vreme snimanja ekran se briše, a crvena lampica na kasetofonu vas upozorava da se vrši snimanje programa. Posle snimanja kompjuter vas obavestava:

```
OK  
READY
```

sto znači da je program snimljen.

Lokaciju programa na kaseti pamтите pomoću brojača okretaja na kasetofonu. U tom smislu je potrebno dati par preporuka:

- za snimanje programa je najbolje da koristite kasete sa mogućnošću snimanja 15 - 30 minuta. Ako koristite kasete od 60 ili 90 minuta trajanja, trošicete dosta vremena u premotavanju, dok nadjete program
- kada upotrebljavate kasetu prvi put, premotajte kasetu na početak, dovedite brojač okretaja na 0 i zatim premotajte

kasetu do 10-og obrtaja, odakle ćete snimiti prvi program. Time izbegavate početni deo trake na koji se ne može snimati. Ovaj postupak je naročiti značajan ako snimanje vršite sa nekim programom za brzo učitavanje i snimanje programa (FASTMODUL, TURBOTAPE i sl.)

- naredne programe uvek snimajte od celih obrtaja, npr. 50, 100, 110, 200 i sl., a ne da snimate po principu program iza programa. Celi brojevi obrtaja odakle počinju programi, čine vašu evidenciju mnogo ažurnijom, a praznine između programa su korisne zbog eventualnih gresaka prilikom snimanja programa.

Snimanje mašinskih programa

U prethodnom delu teksta rekli smo da naredbom SAVE"ime programa",1,1 snimamo mašinski program iz memorije kompjutera pamteći početnu (različitu od 2049) i krajnju adresu programa, koji snimamo. Ova tvrdnja je tačna samo pod pretpostavkom, da ste taj isti program PRVO prethodno učitali sa kasetofona, pa želite sada da napravite kopiju programa.

Na žalost, ako želite da snimate neki vaš mašinski program, ili jednostavno da snimate proizvoljni deo RAM memorije na kasetofon, naredba SAVE neće mnogo pomoći. Vrlo je čudno da COMMODORE nije obezbedio ovu neophodnu komandu.

Rešenja za ovaj nedostatak ima više. Za one koji rade u mašinskom jeziku, tj. koriste neki od postojećih ASEMBLERA (programi pomoću kojih pišete mašinske programe upotrebljavajući mnemonicke skraćenice, kao što su LDA - Load Accumulator - napuni akumulator i sl.) ovo ne predstavlja nikakav problem. Vecina boljih asemblera ima mogućnost snimanja bilo kog dela RAM memorije. Zato, ako programirate u mašinskom jeziku i koristite neki od asemblera moći ćete bez problema da snimate mašinske programe. U "SVETU KOMPJUTERA" br. 5/85 imate listing jednog asemblera, pa ako imate strpljenja, obzirom na dužinu ovog programa, možete ga ubaciti u vaš kompjuter i snimati za dalju upotrebu.

Medjutim, treba dosta vremena da se udje u tajne mašinskog programiranja, pa ćemo vam dati jedan program pisan u BASIC-u, koji vam omogućava da snimate na kasetu ili disketu, bilo koji deo RAM memorije, bez obzira da li on sadži neki mašinski program ili određene podatke.

C-64 u sebi sadrži mašinske rutine kao deo operativnog sistema namenjene za upravljanje memorijom, i ono što nas posebno interesuje, upravljanje INPUT/OUTPUT aktivnostima. Te mašinske rutine su pisane u obliku potprograma, pa ih je bez ikakvog problema moguće koristiti iz BASIC-a. Skup tih mašinskih rutina COMMODORE je nazvao KERNAL i u svim boljim knjigama (pogledajte "Programers Reference Guide") možete naći opis aktivnosti svake rutine i način upotrebe. Osnovni princip rada sa ovim rutinama sastoji se iz tri koraka:

- 1 - postavljanje osnovnih parametara
- 2 - poziv rutine
- 3 - ustanovljenje mogućih grešaka

Da bismo rešili naš problem upotebićemo sledeće rutine:

Naziv rutine	Opis rada	Startna adresa
SAVE	snima željeni deo memorije na odredjeni uredjaj	65496
SETLFS	postavlja broj uredjaja i sekundarnu adresu	65466
SETNAM	dodeljuje ime programu ili delu memorije koji želite da snimate	65469

Tab. B - 2 KERNAL rutine potrebne za snimanje RAM memorije

Kao što primećujete iz opisa rada ovih rutina, verovatno možete odmah pretpostaviti da BASIC naredbe LOAD i SAVE u svom radu takodje koriste ove rutine. Zato nas još više čudi zasto standardni BASIC nema u sebi SAVE naredbu za snimanje bilo kog dela memorije. Medjutim, neka to ostane kao kritika na dušu konstruktorima COMMODORE-a a mi pogledajmo BASIC program koji obavlja željenu funkciju:

```

20 INPUT "{CLR}UNESITE NAZIV.....";IME$
30 INPUT "{CUR DN}KASETOFON/DISK (1/0)?..";KD
40 INPUT "{CUR DN}STARTNA ADRESA.....";SA
50 INPUT "{CUR DN}KRAJNJA ADRESA.....";KA
55 D=LEN(IME$);IF D > 16 THEN D=16
60 FOR I=1 TO D
70 N=ASC(MID$(IME$,I,1))
80 PDKE 49151+I,N
90 NEXT I
100 SHB=INT(SA/256);SLB=SA-SHB*256
110 KHB=INT(KA/256);KLB=KA-KHB*256
115 REM **INICIJALIZACIJA SETLFS RUTINE **
120 PDKE 780,1;POKE 781,KD;POKE 782,255;SYS 65466
125 REM ** INICIJALIZACIJA SETNAM RUTINE **
130 POKE 780,D;POKE 781,0;POKE 782,192;SYS 65469
140 POKE 251,SLB;POKE 252,SHB;POKE 780,251
150 POKE 781,KLB;POKE 782,KHB;
160 SYS 65496

```

Listing B - 1 Program za snimanje RAM memorije
* SNIMANJE

Da biste proverili ovaj program, pokušajte da pomocu programa snimate na kasetofon sam program. Za startnu adresu uzmite 2049, a za krajnju adresu prethodno otkucajte:

```
PRINT PEEK(45) + PEEK(46) * 256
```

i broj koji dobijete ce vam dati krajnju adresu smestanja gornjeg programa. Posle snimanja programa, obrišite program u memoriji sa NEW i učitajte snimljeni program i videćete da je program ponovo u memoriji.

Objašnjenje programa

10 - 50

Unošenje osnovnih parametara.

55 - 90

Program smesta ASCII kodove odabranog imena na odgovarajuće mesto (lokacija 833 i nadalje) u kasetni bafer (vidi tabelu OPIS KASETOG BAFERA).

100 - 110

Izračunavaju se bajtovi više i niže značajnosti za startnu i krajnju adresu (SLB, SHB, KLB, KHB) na već poznat način.

115 - 120

Inicijalizacija SETLFS rutine. U akumulator (registar mikroprocesora 6502) se smesta broj uredjaja, a u X registar se

smesta 1 kao sekundarna adresa i zatim se naredbom SYS 65466 poziva sama rutina SETLFS.

125 - 130

Inicijalizacija SETNAM rutine. Prethodno smo u linijama 60 - 90 smestili ASCII kodove izabranog imena, a sada u akumulator moramo smestiti broj slova u imenu (broj slova smesten u promenljivoj D). Zatim u X i Y registar smestamo bajtove niže i više značajnosti startne adrese, gde smo smestili ime, što možete proveriti jer je: $65 + 3 * 256 = 833$. Posle ovih pripremnih radnji možemo pozvati rutinu SETNAM naredbom SYS 65469.

135 - 160

Posle inicijalizacije SETLFS i SETNAM rutine možemo inicijalizovati rutinu SAVE, zbog koje smo sve ovo i radili. Potrebno je u dve prazne lokacije na 0 strani (izabrali smo lokacije 251 i 252 jer su one ostavljene za slobodno korišćenje) smestiti bafer niže i više značajnosti startne adrese (SLB i SHB). Zatim je u akumulator smeštena adresa lokacije, u kojoj smo smestili bajt niže značajnosti (SLB), a to je lokacija broj 251. Zatim u X i Y registar treba smestiti bajtove niže i više značajnosti krajnje adrese snimanja (KLB i KHB) respektivno. Sada konačno možemo pozvati SAVE rutinu koja će snimiti željeni deo RAM memorije na kasetofon ili disk.

Namerno smo detaljno prošli kroz PROTOKDL pozivanja željenih rutina KERNAL-a, jer i sve ostale rutine u KERNAL-u imaju sličan protokol. Na ovaj način možete koristiti moćno oružje masinskog jezika, iz BASIC-a a da pri tome ne morate znati baš mnogo o masinskom jeziku. Drugim rečima, zašto ne koristiti nešto što vam je na dohvat ruke i može znatno ubrzati rad vašeg BASIC programa. S druge strane možete dobiti mnoge nove osobine na vašem C-64, koje nisu dostupne kroz naredbe. Na taj način smo uostalom i napravili BASIC program koji može da snimi bilo koji deo RAM memorije.

Napomena

Akumulator, X i Y registri su registri širine 1-og bajta u mikroprocesoru 6502, koji je "mozak" vašeg kompjutera. Stanja i vrednosti ovih registara možete pratiti u sledećim memorijskim lokacijama:

NAZIV REGISTRA	LOKACIJA
Akumulator	780
X	781
Y	782

VERIFIKACIJA SNIMLJENIH PROGRAMA

Ako želite da proverite da li ste ispravno snimili program, upotrebite sledeću komandu:

VERIFY "ime programa", broj uređaja

Ako na primer, otkucate komandu VERIFY "PRIMER", kompjuter će javiti poruku:

PRESS PLAY ON TAPE

Kada pritisnete tipku PLAY kompjuter će uporediti program na traci sa programom, koji se trenutno nalazi u memoriji. Ako je sve ispravno javiće vam:

VERIFYING
OK

što znači da je program ispravno snimljen. Ako javi grešku:

? VERIFYING ERROR

znaci da program nije ispravno snimljen i da treba ponoviti postupak snimanja. Ako dodje ponovo do greške onda očistite glavu kasetofona za snimanje/učitavanje specijalnom kasetom. Ako i posle toga bude problema, uzmite novu kasetu. Nakon ovih postupaka bi trebalo sve da bude u redu. Medjutim, nove greške u snimanju značice da morate potražiti pomoć stručne osobe.

PROGRAMI ZA BRZO UCITAVANJE I SNIMANJE

Postoji par programa koji mogu ubrzati snimanje i učitavanje programa sa kasetofona čak i brže nego što se to radi sa diskom. To ubrzanje se odnosi samo na rad sa programima a ne i podacima. Kod nas su najpoznatiji FASTMODUL i TURBOTAPE. Posle učitavanja i startovanja ovih programa možete ostale BASIC programe snimati i učitavati sledećim komandama:

- L "ime programa" - učitava program snimljen ubrzanim postupkom
- S "ime programa" - snima program ubrzanim postupkom
- V "ime programa" - proverava da li je program ispravno snimljen

Kao što smo ranije rekli, kada programe snimate ubrzanim postupkom, program se snima samo u jednoj kopiji.

BRZO TRAZENJE PROGRAMA NA KASETI

Za kraj poglavlja o kasetofonu, ostavili smo jedan vrlo koristan program koji je dao Mike Roberts u časopisu "YOUR COMMODORE", i koji vas može osloboditi problema o vođenju evidencije programa na kasetofonu. Da, dobro ste pročitali, ovaj program vam omogućava da bez pamćenja broja okretaja, dodjete do programa koji želite da učitate, ili da premotate traku do slobodnog mesta za snimanje. Program na izvestan način "glumi" direktoriju kasete, na sličan način kao što, imate direktoriju na svakoj disketi. Nećemo detaljno objašnjavati program, samo vam napominjemo da se bazira na inteligentnom korišćenju ulazno/izlaznog registra smeštenog na memorijskoj adresi 1, koji pored ostalog ima funkciju upravljanja radom kasetofona.

BIT	FUNKCIJA
0	LDRAM SIGNAL (0 = BASIC interpreter isključen)
1	HIRAM SIGNAL (0 = KERNAL isključen)
2	CHAREN SIGNAL (0 = uključen ROM karaktera)
*3	Linija podataka za kasetofon
*4	Kontrola tipki kasetofona (0=neka tipka pritisnuta 1=sve tipke isključene)
*5	Kontrola motora kasetofona(1=isključen 0=uključen)
6-7	Nema funkciju

Tab. B - 3 INPUT/OUTPUT REGISTAR

Kada uključite C-64 i otkucate PRINT PEEK(1) dobićete vrednost 55 jer je:

BIT	0	1	2	3	* 4	* 5	6	7
Bin. vred.	1	1	1	0	1	1	0	0
Dek. vred.	1+	2+	4+	0+	16+	32+	0+	0= 55

Manipulacijom vrednosti 4. i 5. bita koji kontrolišu tipke i motor kasetofona, možete ostvariti potrebne elemente za rad sa kasetofonom na način kako sledi, a što možete videti analizom datog programa.

U programu se koristi i precizni interni časovnik u C-64 preko promenljive TI(od TIME - VREME), pomoću koje možemo očitavati vreme i meriti željenu dužinu premotavanja trake. Program ćemo u daljem tekstu zvati IMENIK.

Princip rada sa IMENIKOM je sledeći:

RAD SA KASETOFONOM

- 1 - IMENIK treba uvek snimiti na pocetak kasete
- 2 - kada zelite da ucitate neki program (ili da ga snimate), premotajte kasetu na pocetak i naredbom LOAD ucitajte IMENIK. Kada se IMENIK ucita u C-64 NEMOJTE PRITISKATI TIPKU STOP na kasetofonu. Ona mora ostati u donjem poloazaju
- 3 - startujete IMENIK sa RUN, i on ce vam dati listu programa na kaseti, a zatim ce vas pitati: UBACITE ZELJENI BRDJ PROGRAMA, kao i to da li zelite snimanje ili ucitavanje
- 4 - sada dolazite do najvaznijeg dela u postupku koji morate obavljati potpuno korektno. IMENIK ce vam javiti poruku na ekranu:

PRITISNI F/FWD TIPKU I <RETURN>

kompjuter zatim, ceka pomocu GET petlje u programskoj liniji 130 sledecu aktivnost. Potrebno je pritisnuti tipku F/FWD na kasetofonu dok drugim prstom drzite pritisnutu tipku PLAY. Nakon toga obe tipke moraju ostati ukljucene. Sada, drzeci pritisnutu tipku F/FWD pritisnite tipku STDP da biste oslobodili tipku PLAY. Na kraju ovog manevra ostala je pritisnuta samo tipka F/FWD, za brzo premotavanje kasete. Medjutim, kasetofon ne pocinje da okrece kasetu unapred sve dok ne pritisnete tipku <RETURN> na kompjuteru. Tog momenta, kasetofon pocinje sa premotavanjem kasete do zeljenog mesta za snimanje ili ucitavanje. Na taj nacin, ne morate voditi racuna o brojacu okretaja, jer C-64 efikasno i tacno proračunava potrebno vreme za premotavanje trake (pogledajte programske linije 150 - 160). Kada se premotavanje trake završi, nastavite sa radom na uobicajen nacin.

Ako vam se za momenat postupak ucinio komplikovanim, vec posle par proba sve tece krajnje rutinski, a vi ste u stanju da nadjete program na kaseti mnogo brze nego bilo kojom drugom metodom. Pored toga imate uvek azuran spisak programa na samoj kaseti.

Postoje odredjena ogranicenja u radu sa IMENIKDM:

- 1 - koristite ovu metodu uvek na novoj kaseti, jer ako se na kaseti nalaze odredjeni programi, oni nisu rasporedjeni u intervalima koji su programirani u IMENIKU, pa je logicno da ih IMENIK ne moze naci
- 2 - IMENIK je u verziji koju dajemo, predvidjen za rad sa 10 programa ne vecih od 8K, za kasetu C-60 min. (svaki sektor od 8K zauzima 6 minuta). Ako zelite da radite sa duplo manjim programima od kojih ni jedan nije veci od 4K izmenite naredbu 160 tako sto cete mnozitelj 10 zameniti sa 5. Ako zelite duplo vece programe od 16K zamenite 10 sa 20
- 3 - broj programa snimljenih na kaseti je smesten u DATA naredbi

u programskoj liniji 230. Broj sledećin DATA naredbi mora odgovarati ovom broju. U nazivu programa pod apostrofima ostavite dovoljan broj blanko znakova da ubacujete imena programa koje snimate na kasetu. To je zbog toga da IMENIK uvek ostane iste dužine i pored upisivanja imena novih programa, jer je nakon toga potrebno IMENIK ponovo snimiti na početak kasete, da bi lista programa bila azurna.

Posle kratkog vremena na prilagodjavanje na ovu metodu rada sa kasetofonom, videćete da rad sa njim postaje mnogo lakši nego inače.

```

1 REM * KATALOG ZA KASETOFON *
10 PRINT "{CLR}";READ X
20 DIM C$(X)
30 FOR I=1 TO X
40 READ C$(I);PRINT C$(I)
50 NEXT
60 PRINT
70 INPUT "{CUR DN}{CUR DN}UCITAVANJE ILI SNIMANJE (U/S)? ";R$;
80 INPUT "{CUR DN}{CUR DN} KOJI PROGRAM ZELITE ";WP
90 IF WP=1 THEN 190
110 POKE 1,39
120 PRINT "{CUR DN}{CUR DN}PRITISNI F/FWD "
125 PRINT "{CUR DN}{CUR DN}ZATIM PRITISNI <RETURN>"
130 GET A1$;IF A1$="" THEN 130
140 POKE 1,7
150 T=TI
160 IF TI<T+(10*60*(WP-1)) THEN 160
170 POKE 1,39
180 PRINT:
190 IF R$="U" THEN PRINT "{CUR DN}{CUR DN}          PRITISNI TIPKU (RVS O
N) STOP (RVS OFF)"
195 PRINT "{CUR DN}{CUR DN} UCITAJ PROGRAM KAO STO OBICNO CINIS"
200 PRINT :IF R$="U" THEN 230
210 IF R$="S" THEN PRINT "{CUR DN}{CUR DN}PRITISNI STOP"
220 PRINT "{CUR DN}{CUR DN} SPREMAN DA SACUVA NOVI PROGRAM ODNOSNO DA
TOTEKU"
230 DATA 10
240 DATA "PROGRAM 1           1"
250 DATA "PROGRAM 2           2"
260 DATA "PROGRAM 3           3"
270 DATA "PROGRAM 4           4"
280 DATA "PROGRAM 5           5"
290 DATA "PROGRAM 6           6"
300 DATA "PROGRAM 7           7"
310 DATA "PROGRAM 8           8"
320 DATA "PROGRAM 9           9"
330 DATA "PROGRAM 10          10"
340 DATA "
350 END

```

Listing 8 - 2 Programa za brzo pretraživanje po kaseti
* IMENIK

D A T O T E K E

Kreiranje svakog ozbiljnijeg programa sastoji se iz dva dela: algoritamskog rešavanja problema i organizacije podataka, koji se u programu koriste. U većini jednostavnih programa, glavni problem je napraviti algoritam i BASIC program po tom algoritmu. Kod složenijih programa, potrebno je proširiti to znanje, poznavanjem organizacije podataka i rada sa podacima.

ORGANIZACIJA PODATAKA

Podaci se organizuju u složene strukture podataka, koje zovemo DATOTEKE. Svaka datoteka predstavlja niz slogova logično povezanih u celinu. SLOG, kao osnovni element datoteke, može predstavljati jedan elementaran podatak, ili grupu elementarnih podataka logično povezanih. Primer za datoteku, čiji je slog jedan elementaran podatak, bila bi datoteka telefonskih brojeva, datoteka imena učenika u nekoj školi, imena proizvoda u nekoj radnji i sl. Izgled datoteke gde je slog jedan elementaran podatak, npr. za telefonske brojeve je:

```

1. slog          ...          5. slog
*****
* 123078 * 212233 * 667703 * 150949 * 332513 *
*****

```

Ukoliko napravimo kombinaciju više elementarnih polja na sledeći način:

```

*****
*IME I PREZIME*DATUM RODJENJA*MESTO RODJENJA*POL*
*****

```

i formiramo niz slogova koji će sadržati ova četiri podatka za svakog pojedinca, dobićemo datoteku ličnih podataka izvesne grupe ljudi (npr. stanovnika jedne opštine). Izgled datoteke, kod koje je slog definisan na ovaj način je:

```

*****
*PETAR PETROVIC*13.01.1956*BEOGRAD* 1 * 1. slog
*-----*
*ZORAN MARIC *10.09.1943*ZAGREB * 1 * 2. slog
*-----*
*VESNA ILIC *12.12.1961*BEOGRAD* 2 * 3. slog
*-----*
* ... * .... * ... * . * 4. slog
*-----*
* ... * .... * ... * . * ....

```

Postoji više načina koji omogućavaju rad sa velikim brojem Rodataka, koji zavise od toga kako su podaci organizovani u datoteke. Podaci mogu biti organizovani na tri osnovna načina, koje zovemo TIPOVI DATOTEKA:

1. Sekvencijalni (SEQUENTIAL)
2. Slučajni (RANDOM)
3. Relativni (RELATIVE)

O svakom tipu organizacije govoriceo detaljno nešto kasnije. Tako organizovane podatke cuvamo na trakama (kasetama) ili disketama, i kada nam zatrebaju u nekom programu, ponovo ih uzimamo sa kasete ili disketa. Znači, ukoliko želimo da radimo sa podacima, moramo naučiti dve stvari:

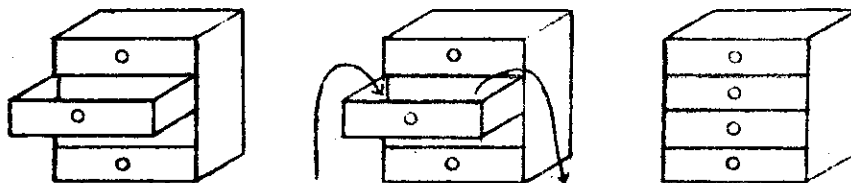
- kako da organizujemo podatke
- kako da uzimamo ili smestamo podatke na kasete ili diskete kad god je to potrebno.

SEKVENCIJALNE DATOTEKE

Sekvencijalne datoteke predstavljaju niz podataka, kao što su npr. telefonski brojevi, adrese, imena i sl. One se mogu cuvati na kaseti ili disketi. Podaci se cuvaju u istom poretku u kome su i uneti. Prvi podatak upisan u datoteku bice prvi podatak koji ce biti procitan, kada se datoteka ucitava.

SEKVENCIJALNE DATOTEKE NA KASETI

Sekvencijalne datoteke mozemo uporediti sa ladicama u ormanu. Prvo ih otvorimo, stavimo nešto unutra ili uzmemo nešto iz njih, a zatim ih zatvorimo.



```
OPEN 1,1,1,"TEST DAT"   PRINT#1,WS(X)   INPUT#1,WS(X)   CLOSE1
ili
OPEN 1,1,0,"TEST DAT"
```

Sl. C - 1 Rad sa sekvencijalnim datotekama

Naredbe koje nam omogućavaju da izvodimo te postupke sa datotekama su:

OPEN - Otvara datoteke i ima opšti oblik

OPEN A, B, C, "ime datoteke" gde je:

- A - broj pridružen datoteci, koji može biti od 1 - 127
- B - broj perifernog uređaja gde se nalazi datoteka (1-kasetofon, 3-ekran, 4-printer, itd.)
- C - sekundarna adresa, koja određuje da li će podaci biti upisivani (1), ili učitavani (0). Ona može biti i jednaka 2, što znači upisivanje datoteke i markera za kraj trake (EOT - END OF TAPE marker) posle upisanog zadnjeg podatka.

Na primer **OPEN 1,1,1,"TEST"** znači da je otvorena datoteka pod imenom TEST, sa pridruženim brojem 1, koja se nalazi na kaseti i koja će biti upisivana na kasetu.

PRINT# - Upisuje podatke na kasetu i ima opšti oblik

PRINT# A,D gde je:

- A - broj datoteke koji joj je pridružen u OPEN naredbi, kojom je određeno da se datoteka upisuje
- D - BASIC promenljiva iz koje će podaci biti upisivani.

Na primer **PRINT# 1, W\$** upisaće sadržaj string promenljive u datoteku na kaseti, prethodno otvorenu u OPEN naredbi.

INPUT# - Učitava podatke iz datoteke na kaseti u program. Njen opšti oblik je

INPUT# A,D gde je:

- A - broj datoteke koji joj je pridružen u OPEN naredbi, kojom je određeno da se datoteka učitava
- D - BASIC promenljiva u koju će podaci sa kasete biti preneti.

Na primer ako otvorimo datoteku naredbom **OPEN 1,1,0,"TEST"**, naredba **INPUT# 1, W\$** pročitace sadržaj datoteke 1, sa kasete i smestiti ga u promenljivu W\$.

GET# - Alternativna naredba INPUT# naredbi, s tom razlikom što naredba GET# učitava karakter po karakter elementarnog podatka. Znači, naredba GET# radi isto što i naredba GET, samo što karaktere uzima sa kasete, umesto sa tastature.

CLOSE - Zatvara datoteke i ima oblik

CLOSE A gde je:

A - broj datoteke otvorene u OPEN naredbi.

Buduci da je najlakse uciti na primeru, daceo tri primera jednostavnih programa, koji kreiraju datoteke na kaseti, ucitavaju podatke nazad iz datoteka, i pokazuju njihov sadrzaj na ekranu. Oni vam mogu poslužiti da sami kreirate datoteke za svoje programe, kada savladate tehniku rada sa datotekama. Prva dva programa kreiraju datoteke na kaseti, dok treci program ucitava datoteke i stampa sadrzaj datoteke na ekranu.

I primer

Kada startujete ovaj program, on vas prvo "pita" za broj elemenata u datoteci, zatim dimenzioniše niz u kome ce držati te elemente, zatim vam "traži" da unesete ime datoteke. Ovim programom možete kreirati više datoteka, jer po završetku kreiranja jedne datoteke program vas ponovo "pita" za novo ime datoteke, sve dok ne unesete +1, koje služi kao indikator za kraj rada programa. Posle imena datoteke, program vas upućuje kako da unosite podatke, popunjava niz, upisuje ga na kasetu i pokazuje sadržaj datoteke na ekranu. Kada unosite podatke, primeticete, da je broj podataka koji možete uneti za jedan veci od broja koji ste zadali. Taj poslednji podatak služi kao indikator kraja datoteke, odnosno END-OF-FILE (EOF) marker. U ovom primeru to je broj -1. Medjutim, nije obavezno uneti EOF marker. Ako kreirate više datoteka i upisujete ih jednu za drugom, preporucljivo je da koristite EOF marker.

II primer

Ovaj primer radi isto sto i prethodni, na malo drukćiji način, koristeći DATA linije. DATA linije moraju biti ispravno napisane, da bi se izbegle moguće greske. Prvi element u DATA liniji cita se kao ime datoteke, a ostali do EOF markera "-1" kao elementi datoteke. Kao što se vidi iz DATA linije, ovim programom se kreiraju dve datoteke sa imenima W21 i W22. To znaci, da se ovim programom može kreirati više datoteka odjednom. Kraj podataka, koji se unose DATA linijama oznacen je sa "+1", posle koga se završava rad programa.

III primer

Ovaj program vam omogućava da vidite šta je u datotekama, odnosno, da ucitavate datoteke. Pri tome obratite pažnju da

DATDTEKE

premotate kasetu na mesto gde se nalazi datoteka, koja je prethodno kreirana, a koju zelite da ucitate.

1. PRIMER

```

10 INPUT " BRDJ ELEMENATA U DATDTECI ";N
20 DIM W$(N+1)
30 INPUT " IME DATDTEKE ";IME$:IF IME$="+1" THEN END
40 PRINT " UNESITE PDDATKE-PDSLE SVAKDG <RETURN> "
50 OPEN 1,1,1, IME$
60 FOR X=0 TO N:INPUT W$(X):PRINT#1,W$(X):IF W$(X)<>"-1"
  THEN NEXT
70 CLOSE 1:FOR X=1 TO N:PRINT W$(X):NEXT
80 PRINT " PRITISNITE BILD KDJU TIPKU "
90 GET A$:IF A$="" THEN 90
100 GOTO 10
110 END

```

2. PRIMER

```

10 DATA W21,DRVO,DEO,PRICA,STO,GRAD,SAT
20 DATA PRVI,DRUGI,TRECI,CETVRTI,PETI,SESTI,-1
30 DATA W22,LET,HDD,PUT,SNEG,OTVOR
40 DATA LOSE,DOBRO,SREDNJE,MALI,VELIKI,SAN
50 DATA PONEDELJAK,UTDRAK,SREDA,CETVRTAK,PETAK,-1,+1
60 READ IME$:R$=CHR$(13)
70 IF IME$="+1" THEN GOTO 120
80 OPEN 1,1,1,IME$
90 READ NIZ$:PRINT# 1,NIZ$;R$
100 IF NIZ$<>"-1" THEN 90
110 CLOSE 1 : GOTO 60
120 CLOSE 1 : END

```

3. PRIMER

```

10 INPUT " IME DATOTEKE";IME$
20 IF IME$="+1" THEN END
30 OPEN 1,1,0, IME$ : N=0 : DIMW$(16)
40 INPUT# 1,W$(N) : N=N+1
50 IF W$(N-1)="-1" THEN 70
60 GOTO 40
70 CLOSE 1:FOR X=0 TO (N-1):PRINT W$(X):NEXT
80 PRINT " PRITISNITE BILO KOJU TIPKU "
90 GET A$:IF A$="" THEN 90
100 RUN 10 : END

```

SMESTANJE PODATAKA

Ako želite u potpunosti da ovladate radom sa datotekama, morate znati kako se podaci smestaju na kasete i diskete. Način smestanja razlicit je za string i numericke podatke.

SMESTANJE STRING PODATAKA

Koristite naredbu PRINT# za smestanje podataka na kasetu ili disketu, u obliku sekvencijalne datoteke, i interesuje vas, kako se podaci smestaju. Odgovor na ovo pitanje necete naci baš u svakoj knjizi. Medjutim u nekim knjigama, npr. "Programer's Reference Guide", zatim Uputstvo za rad sa diskom itd., mozete pronaci "odgovor" na to pitanje. Najkrace receno, mozete pročitati sledeće:

Naredba PRINT# omogućava upis podataka u datoteku. Kako je naredba PRINT# analogna naredbi PRINT, s tom razlikom što podatke smesta na kasetu ili disketu, umesto na ekran, podsetimo se naredbe PRINT. Ukoliko koristite naredbu PRINT u obliku PRINT A\$,B\$,C\$ sadržaji promenljivih odštampe se na ekranu tako, da za svaku promenljivu bude rezervisano deset pozicija. Na primer ako je A\$="PRVI", B\$="DRUGI", C\$="TRECI", na ekranu ce se se posle primene gornje naredbe pojaviti:

```
PRVI      DRUGI      TRECI
```

Takodje, ako koristite naredbu PRINT# za upis sekvencijalne datoteke (bilo na kasetu, bilo na disk), u obliku

```
PRINT#1, A$, B$, C$
```

(prethodno ste otvorili datoteku sa identifikacionim brojem 1), podaci ce biti smešteni u datoteku u obliku

```
PRVI      DRUGI      TRECI      CR EOF
1234567890123456789012345678901 2
          10          20          30
          $1. C - 2
```

CR=CHR\$(13) ASCII kod tipke RETURN
 EOF-END OF FILE MARKER -indikator za kraj datoteke

U gornjoj naredbi nisu korišćeni separatori za razdvajanje podataka pri unošenju, tako da se sadržaj promenljivih A\$, B\$,

C\$, može učitati samo kao sadržaj jedne promenljive, naredbom INPUT#. Ilustrujmo to primerom za rad sa datotekama na kaseti. Program koji sledi upisuje sadržaje promenljivih A\$, B\$, C\$ u datoteku sa identifikacionin brojem 1, koja se nalazi na kaseti:

```
10 A$="PRVI"
20 B$="DRUGI"
30 C$="TRECI"
40 OPEN 1,1,1, "DAT1"
50 PRINT#1, A$, B$, C$
60 CLOSE 1
```

Zatim učitajte sadržaj datoteke, sledecim programom:

```
10 OPEN 1,1,0, "DAT1"
20 INPUT#1, A$
30 PRINT A$
40 CLOSE 1
```

Da je tačno ono što piše u literaturi, na ekranu biste dobili:

```
PRVI      DRUGI      TRECI
123456790123456789012345
      10          20
```

Umesto toga na ekranu dobijate:

```
PRVI      DRUGI      TRECI
1234567890123456789012345678901234567890
      10          20          30          40
```

Na ovaj način imate na ekranu odštampan izgled podataka onako, kako su oni smešteni u datoteci. Odmah uočavate da podaci nisu dopunjeni do deset blanko znakima (blanko znak-naziv za prazninu na ekranu, ASCII kod 32) već da se posle svakog podatka smešta deset blanko znakova. To znači da se podaci u sekvencijalne datoteke ne smeštaju kao na sl. C - 2, već u obliku:

```
SADRZAJ  PRVI      DRUGI      TRECI CR EOF
KARAKTER 1234567890123456789012345678901234 5 6 7890
          10          20          30          40
```

Sl. C -3

Da biste na još jedan način proverili da je ovo drugo tvrdjenje tačno, napravite program koji učitava sadržaj datoteke naredbom

GET#, uzimajući karakter po karakter, i štampa na ekranu ASCII kodove svih karaktera koji su uzeti iz datoteke:

```
10 OPEN 1,1,0, "DAT1"
20 FOR I=1 TO 36
30 GET#1, X$
40 PRINT ASC(X$); " ";
50 NEXT I
60 CLOSE 1
```

Kada startujete ovaj program dobićete na ekranu sledeće:

```
P R V I                                D R U G I
80 82 86 73 32 32 32 32 32 32 32 32 32 32 68 82 85 71 73
                                T R E C I CR EOF
32 32 32 32 32 32 32 32 32 32 84 82 69 67 73 13 199
```

Budući da GET# naredba uzima karakter po karakter iz datoteke, više nema nikakve sumnje da se podaci naredbom

```
PRINT#1, A$, B$, C$
```

ne smeštaju onako kako to stoji u pomenutim knjigama, nego u obliku koji je dat na sl. C - 3. Znači, ako za smestanje podataka koristite ovu naredbu, posle svakog podatka biće dodano deset blanko znakova. Da bi eliminisali ove nepotrebne blanko znake, podatke možete smestati koristeći naredbu

```
PRINT#1, A$;B$;C$
```

Ako je sadržaj promenljivih A\$="PRVI", B\$="DRUGI", C\$="TRECI", izgled datoteke 1 posle primene ove naredbe biće:

```
P R V I D R U G I T R E C I CR EOF
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
```

Sl. C - 4

Ukoliko biste pokušali da naredbom INPUT#1, A\$,B\$,C\$ učitate podatke nazad, videli biste da posle te naredbe A\$ sadrži sva tri prethodno uneta podatka, zajedno sa blanko znacima u slučaju kada su promenljive odvojene zarezima (,), u PRINT# naredbi, odnosno bez blanko znakova kada su promenljive odvojene sa (;) u PRINT# naredbi. To se desilo zbog toga što između podataka u datoteci nema nikakvih separatora, koji odvajaju jedan podatak od drugog, tako da se INPUT# naredbom u jednu promenljivu učitava sve do

CR=CHR\$(13), jer CR ima funkciju da odvaja podatke. Znači da bi podaci bili ucitani pravilno potrebno ih je pravilno odvojiti separatorima. CR odnosno RETURN kod automatski se postavlja iza poslednjeg podatka u PRINT# naredbi, medjutim to nije jedini način za postavljanje separatora. Sada cemo izloziti tri osnovna načina za postavljanje separatora izmedju podataka.

POSTAVLJANJE SEPARATORA IZMEDJU PODATAKA

I Način

Sastoji se u tome da svaku promenljivu unosite posebnom PRINT# naredbom, npr. :

```
PRINT#1, A$
PRINT#1, B$
PRINT#1, C$
```

Na taj način automatski ce biti postavljen kod RETURN - CHR\$(13), iza svakog podatka, jer iza svake naredbe pritiskamo tipku RETURN.

II Način

Sastoji se u tome da se iza svakog podatka direktno ubaci kod RETURN - CHR\$(13), naredbom:

```
PRINT#1, A$CHR$(13)B$CHR$(13)C$
```

Kada se primene ova dva načina za odvajanje podataka, izgled datoteke je:

```
PRVI CRDRUGI CRTRECI CR EOF
1234 5 67890 1 23456 7 8 90123456
      10                20
```

Sl. C - 5

III Način

Sastoji se u tome da se podaci odvoje zarezom (,). Najčešće se nekoj promenljivoj R\$ dodeli vrednost R\$=",", a zatim se koristi naredba:

```
PRINT#1, A$R$B$R$C$
```

Obratite pažnju na to, da ne stavljate nikakve znake interpunkcije između promenljivih. Posle primene ove naredbe datoteka izgleda:

```
PRVI,DRUGI,TRECI CR
1234567890123456 7 8901
      10          20
```

Sl C - 6

Prva dva načina omogućavaju vam da podatke učitavate naredbom INPUT#, a treći naredbom GET#.

SMESTANJE NUMERICKIH PODATAKA

O smestanju numerickih podataka u literaturi mozete pronaci sledece:

- Numericki podaci smestaju u sekvencijalne datoteke na kaseti ili disketi u obliku:
- jedan blanko znak, ako je broj pozitivan ili znak (-), ako je broj negativan
 - broj
 - znak za kursor udesno (->)

Ako primenite naredbu PRINT#1,1;-3;5;7 izgled datoteke 1 bio bi:

```
SADRZ. 1 ->- 3 -> 5 -> 7 -> CR EOF
KARAK.1 2 3 4 5 6 7 8 9 10 11 12 13 14
      Sl. C - 7
```

Ovaj format za smestanje numerickih podataka obezbedjuje dovoljno informacija za INPUT# naredbu, tako da ih pravilno ucita, kao odvojene brojeve, ne koristeći nikakve druge separatore. Pokušajte to da uradite i uvericete se da to nije tacno. Prvo, sl. C - 7 ce vas navesti da pomislite da je znak (->) separator, i da predstavlja CHR\$(29)-pomeranje kursora udesno. Medjutim ako napravite primere slicne predhodnim, videćete kako se zaista smestaju numericki podaci. Program koji sledi, upisuje sadržaj numerickih promenljivih A, B, C, D, u sekvencijalnu datoteku na kaseti:

```
10 A=1:B=-3:C=5:D=7
20 OPEN 1,1,1, "DAT2"
30 PRINT#1, A;B;C;D
40 CLOSE 1
```

Zatim učitajte podatke programom:

```
10 OPEN 1,1,0, "DAT2"
20 INPUT#1, A,B,C,D
30 PRINT A,B,C,D
40 CLOSE 1
```

kako vam to savetuju u priručnicima i videćete da je to nemoguće. Umesto toga učitajte sadržaj datoteke GET# naredbom:

```
10 OPEN 1,1,0, "DAT2"
20 FOR I=1 TO 14
30 GET#1, X#
40 PRINT ASC(X#); " ";
50 NEXT I
60 CLOSE 1
```

Kada startujete ovaj program dobićete na ekranu:

```
1      - 3      5      7      CR EOF
32 49 32 45 51 32 32 53 32 32 55 32 13 199
```

Znači ako primenite naredbu PRINT#1, 1;-3;5;7 izgled datoteke neće biti kao na sl. C - 7, već:

```
SADRZAJ 1 - 3 5 7 CR EOF
KARAKTE.1 2 3 4 5 6 7 8 9 10 11 12 13 14
```

Sl. C - 8

Iz toga možemo da zaključimo da se numerički podaci smeštaju u obliku:

- jedan blanko znak, ako je broj pozitivan znak minus (-) , ako je broj negativan
- broj
- jedan blanko znak

UCITAVANJE PODATKA INPUT# NAREDBOM

String podaci koji su smešteni uz korišćenje separatora CHR\$(13), (1. i 2. NACIN), učitavaju se naredbom INPUT#. Pravilno učitavanje omogućava separator CHR\$(13). Numerički podaci koji su razdvojeni separatorom CHR\$(13), takodje se pravilno učitavaju naredbom INPUT#.

UCITAVANJE PODATAKA GET# NAREDBOM

String podaci koji su smešteni koristeći separator ",", (3. NACIN), učitavaju se naredbom GET#, koja uzima karakter po karakter podatka. Pri tom se, u programu koji učitava datoteku, vrši ispitivanje da li je sadržaj promenljive, koja se učitava jednak ",",. Kada je to tačno, to je znak da je učitana jedan ceo podatak, a zatim se nastavlja učitavanje sledećeg podatka. Numerički podaci, takodje, se mogu učitavati GET# naredbom, s tim što je indikator kraja podataka blanko znak (ASCII kod 32).

Primeri koji su dati u prethodnom poglavlju napravljeni su za smeštanje string podataka, koristeći prvi način za odvajanje podataka - po jedna PRINT# naredba za svaki podatak. Na taj način posle svakog podatka automatski je postavljen CHR\$(13), i učitavanje je izvršeno pravilno. Primeri koji slede ilustruju primenu 2. i 3. NACINA za razdvajanje podataka, smeštanje numeričkih podataka, mešovityh podataka u okviru jednog sloga i učitavanje podataka naredbom GET# i INPUT#.

4. PRIMER

```
10 IME$="ANA"
20 PREZ$="LAMZA"
30 POL$="ZENSKI"
40 OPEN 1,1,1,"DAT1"
50 PRINT#1,IME$,PREZ$,POL$:CLOSE 1
60 PRINT IME$,PREZ$,POL$
70 END
```

4.PRIMER puni datoteku DAT1 string podacima ANA, LAMZA, ZENSKI. String promenljive su odvojene zarezima u naredbi PRINT#1, znači podaci će biti smešteni tako, da će se postavljati 10 blanko znakova iza svakog podatka. Program zatim daje na ekranu sadržaj tih promenljivih u formatu za ekran.

5. PRIMER

```
10 IME$="ANA"
20 PREZ$="LAMZA"
30 POL$="ZENSKI"
40 OPEN 1,1,1,"DAT2"
50 PRINT#1,IME$;PREZ$;POL$:CLOSE 1
60 PRINT IME$;PREZ$;POL$
```

5.PRIMER puni datoteku DAT2 istim podacima kao u prethodnom

primeru, samo su string promenljive odvojene sa ";" u PRINT# naredbi. Stoga se sadržaj promenljivih smešta u datoteku kao na sl. C - 4. Primer zatim daje na ekranu sadržaj tih promenljivih u istom formatu.

6. PRIMER

```
10 INPUT "IME DATOTEKE ";NAZIV$
20 OPEN 1,1,1,NAZIV$
30 INPUT#1,IME$:CLOSE 1
40 PRINT IME$
50 END
```

6.PRIMEROM se učitava datoteka DAT1 ili DAT2 na taj način što vas program pita da unesete ime datoteke koju hoćete da ućitavate. To ućitavanje je nepravilno jer će se u IME\$ smestiti sadržaj sve tri promenljive.

Ako ste uneli ime datoteke DAT1 posle izvršenja ovog programa na ekranu će se odštampati IME\$ u obliku:

```
ANA          LAMZA          ZENSKI
```

Ako ste uneli ime datoteke DAT2 posle izvršenja ovog programa, na ekranu će se odštampati IME\$ u obliku:

```
ANALAMZAZENSKI
```

Prethodni primeri su vam praktićno pokazali gresku, koja nastaje pri smestanju podataka kada se ne koriste separatori.

Da biste videli kako se koristi separator CHR\$(13) u 4. PRIMERU zamenite liniju broj 50 sa :

```
50PRINT#1,IME$CHR$(13)PREZ$CHR$(13)POL$
```

Startujte ovako izmenjen program i zatim ućitajte datoteku pomocu sledećeg programa:

7. PRIMER

```
10 INPUT "IME DATOTEKE";NAZIV$
20 OPEN 1,1,1,NAZIV$
30 INPUT#1,IME$,PREZ$,POL$:CLOSE 1
40 PRINT IME$,PREZ$,POL$
50 END
```

Posle izvršenja ovog programa IME\$ sadrži ANA, PREZ\$ sadrži

LAMZA, POL\$ sadrži ZENSKI, što znači da je učitavanje izvršeno pravilno. Sledeće što možete da uradite je da liniju 50 izmenite tako da separator bude "," i da se učitavanje vrši GET# naredbom. Dodajte programu liniju :

```
45 Z$=","
```

a liniju 50 izmenite u oblik:

```
50 PRINT#1,IME$Z$PREZ$Z$POL$
```

Posle izvršenja ovog programa podaci su smešteni u datoteku kao na sl. C - 6. Učitavanje izvršite B.PRIMEROM.

B. PRIMER

```
10 INPUT "IME DATOTEKE";NAZIV$
20 OPEN 1,1,0,NAZIV$
30 GOSUB 500: IME$=B$
40 GOSUB 500: PREZ$=B$
50 GOSUB 500: POL$=B$
60 PRINT IME$:PRINT PREZ$:PRINT POL$
70 CLOSE 1 : END
500 B$=""
510 GET#1,A$
520 IF A$="," THEN RETURN
530 A$=A$+CHR$(0)
540 B$=B$+A$
550 GOTO 510
```

9.PRIMER ilustruje rad sa numerickim podacima, odnosno smestanje numerickih podataka u datoteku NUM.

9. PRIMER

```
10 A=-3.248
20 B=3456
30 C=33.23
40 OPEN 1,1,1,NUM
50 PRINT#1,ACHR$(13)BCHR$(13)C
60 PRINT A,B,C
70 CLOSE 1 : END
```

Odnosno učitavanje numerickih podataka naredbom INPUT# (10. PRIMER):

10. PRIMER

```

10 OPEN 1,1,0,NUM
20 INPUT#1,A,B,C:CLOSE 1
30 PRINT A:PRINT B:PRINT C:
40 END

```

Na kraju da vidimo kako se smestaju i ucitavaju mesoviti podaci, string i numericki. Recimo da hocemo da formiramo datoteku sa slogom oblika:

```

      15          10          5      3      7
PREZIME MESTO-RODJ. STAROST PDL PLATA
      A$          B$          C      D$   E

```

Podatke unositi u sledecem obliku :

```

-PREZIME:      do 12 karaktera
-MEST-RODJ.:  do  8 karaktera
-STAROST:      do  3 broja
-PDL:1 za zenski, 0 za muski
-PLATA:        do  7 karaktera

```

Program za formiranje omogucava unosenje podataka u datoteku cije ime birate, a zatim stampa slogove datoteke na ekranu.

FORMIRANJE SEKVENC. DATOTEKE - KASETA

```

10 PRINT"FORMIRANJE DATOTEKE NA TRACI"
20 PRINT"SLOG JE OBLIKA:"
30 PRINT
40 PRINT"*PREZIME*MES-RODJ.*STAROST*
POL*PLATA*":PRINT
50 INPUT"BROJ SLOGOVA U DATOTECI";N
60 DIM A$(N),B$(N),C(N),D$(N),E(N)
70 INPUT"IME DATOTEKE";NAZIV$
80 OPEN 1,1,1,NAZIV$
85 R$=CHR$(13)
90 FOR X=1 TO N
100 INPUT"PREZIME";A$(X)
110 INPUT"MESTO RODJENJA";B$(X)
120 INPUT"STAROST";C(X)
130 INPUT"PDL";D$(X)
140 INPUT"PLATA";E(X)
150 PRINT#1,A$(X)R$B$(X)R$C(X)R$D$(X)
R$E(X)
160 CLOSE 1
170 FOR X=1 TO N

```

```

180 PRINT A$(X)TAB(15)B$(X)TAB(25)
C(X)TAB(30)D$(X)TAB(33)E(X)
190 NEXT X
200 PRINT:PRINT"PRITISNI NEKU TIPKU"
210 GET G$:IF G$="" THEN 210
220 END

```

Ako želite da učitate datoteku da bi menjali slogove ili vršili neku obradu možete koristiti sledeći program:

UCITAVANJE SEQ. DATOTEKA - KASETA

```

10 INPUT"IME DATOTEKE";NAZIV$
20 INPUT"BROJ SLOGOVA";N
30 DIM A$(N),B$(N),C(N),D$(N),E(N)
35 OPEN 1,1,0, NAZIV$
40 FOR X=1 TO N
50 INPUT#1,A$(X),B$(X),C(X),D$(X),E(X)
60 NEXT X
70 CLOSE 1
80 FOR I=1 TO N
90 PRINT A$(I)TAB(15)B$(I)TAB(25)C(I)
TAB(30)D$(I)TAB(35)E(I)
100 NEXT I
110 END

```

Prethodna dva primera mogu vam poslužiti da kreirate svoje sopstvene datoteke, a zatim ih učitate. Ako dopunite poslednji program, posle učitanja možete menjati sadržaj slogova datoteke, a zatim, tako izmenjenu datoteku pamtiti u obliku nove datoteke. Taj postupak se zove AZURIRANJE DATOTEKE. O tome ćemo više govoriti u nekom od sledećih poglavlja.

R A D S A D I S K O M

Najveći broj kupaca kućnih kompjutera, na početku kupuje samo centralnu jedinicu i kasetofon. Tako da se prva iskustva vezana za pamćenje, smestanje i rad sa prethodno zapamćenim programima i podacima, stiču upravo u radu sa kasetofonom. Sa nepredovanjem u programiranju i proširivanjem oblasti u kojima želite da koristite vaš kompjuter, kasetofon sve više i više postaje ograničavajući faktor. Njegova najveća mana je njegova sporost. To vam neće smetati, dok su vaši programi mali. Međutim, što više budete znali bićete u mogućnosti da pravite složenije programe, čija će dužina biti sve veća. Tako da će vam sporost u radu kasetofona sve više smetati. To će biti prvi znak da vam je potrebna periferijska jedinica, koja vrši istu funkciju kao kasetofon, ali sa boljim karakteristikama. Jednom rečju, potreban vam je DISK.

Prednost diska nad kasetofonom nije samo u brzini, jer postoje brojni programi koji nekoliko puta ubrzavaju smestanje ili učitavanje sa kasetofona. Najveća prednost leži u brzini u kojoj disk nalazi informacije, koje smo ranije smestili, ili mesto za informacije koje tek želimo da zapamtimo. Razlika između diska i kasetofona može da se uporedi sa razlikom između kasetofona i gramofona. Kada slušate muziku sa kasete i želite da nadjete neku određenu muzičku numeru, jedini način je premotavanja unapred ili unazad do željenog mesta. Kod gramofona ste u mogućnosti da tražite željenu numeru, pomerajući iglu u bilo kom pravcu. Disk ima mogućnost da sam vrši to pomeranje i locira željeno mesto na disketi.

Na početku treba objasniti dva osnovna pojma vezana za rad sa diskom. To su:

- DISK JEDINICA
- DISKETA

DISK JEDINICA je tehnički uređaj analogan kasetofonu.

DISKETA je fizički medijum na kome se smestaju i pamt podaci i programi, analogna je kaseti.

DISK JEDINICA

Disk jedinica 1541 je tehnički uređaj koji ima sledeće karakteristike:

- sopstvenu ROM memoriju od 16K, koja služi za disk kontroler i operativni sistem diska (DOS). Zbog toga se ova disk jedinica naziva "inteligentna disk jedinica"
- 2K RAM memorije, koja služi korisniku u radu sa diskom
- postojanje sopstvene memorije omogućava rad sa diskom bez trošenja memorije centralne jedinice
- sadrži dvosmerni serijski bas, koji je sličan paralelnom IEEE-488 interfejsu, koga COMMODOR-e koristi na svojim profesionalnim modelima. Osnovna razlika je u tome što se za prenos podataka koristi jedna žica (otuda i naziv serijski bas) umesto 8 koliko koristi IEEE-488. Time je brzina prenosa podataka znatno smanjena, ali je izbegnuta upotreba vrlo skupog paralelnog IEEE-488 interfejsa
- na pozadini disk jedinice nalaze se dva priključka za serijski bas, čime je omogućeno povezivanje u liniju do pet diskova i jednog štampača. O povezivanju centralne jedinice i diska pogledajte u poglavlju koje specijalno govori o tome
- disk 1541 je kompatibilan sa COMMODOR-e diskovima 4040 i 2031.

Osnovna tehnička specifikacija diska 1541 data je sledećom tabelom:

T E H N I C K A S P E C I F I K A C I J A

KAPACITETI

1. Ukupni kapacitet.....174848 bajtova po disketi
2. Kapacitet u zavisnosti od tipa datoteka:
 - sekvencijalne.....168656 bajtova po disketi
 - relativne.....167132 bajtova po disketi
(maksimalan broj slogova relativne datoteke je 65535 slogova)
3. Maksimalan broj programa na disketi (ili datoteka).....144 po disketi
4. Broj traka.....35 po disketi
5. Broj blokova.....683 po disketi
6. Broj blokova slobodnih za korisnika.....664 po disketi
7. Broj blokova po traci.....17 - 21

PROCESORI

1. 6502.....mikro procesor identican kao kod centralne jedinice

2. 6522 (2 kom.).....I/O mikroprocesor

BAFERI

1. 4 kom. po 256 bajtova.....2 KB RAM

NAPAJANJE

1. Sopstveno: - napon.....220 V
 - frekvenca.....50/60 Hz
 - snaga.....25 W
2. Topljivi osigurač dimenzionisan na.....750 mA i 250 V

DISKETE

1. Velicina.....5 1/4 inča
2. Gustina.....jednostruka (single density)
3. Tip diskete.....jednostrana (single sided)

Tab.D - 1 - Tehnička specifikacija

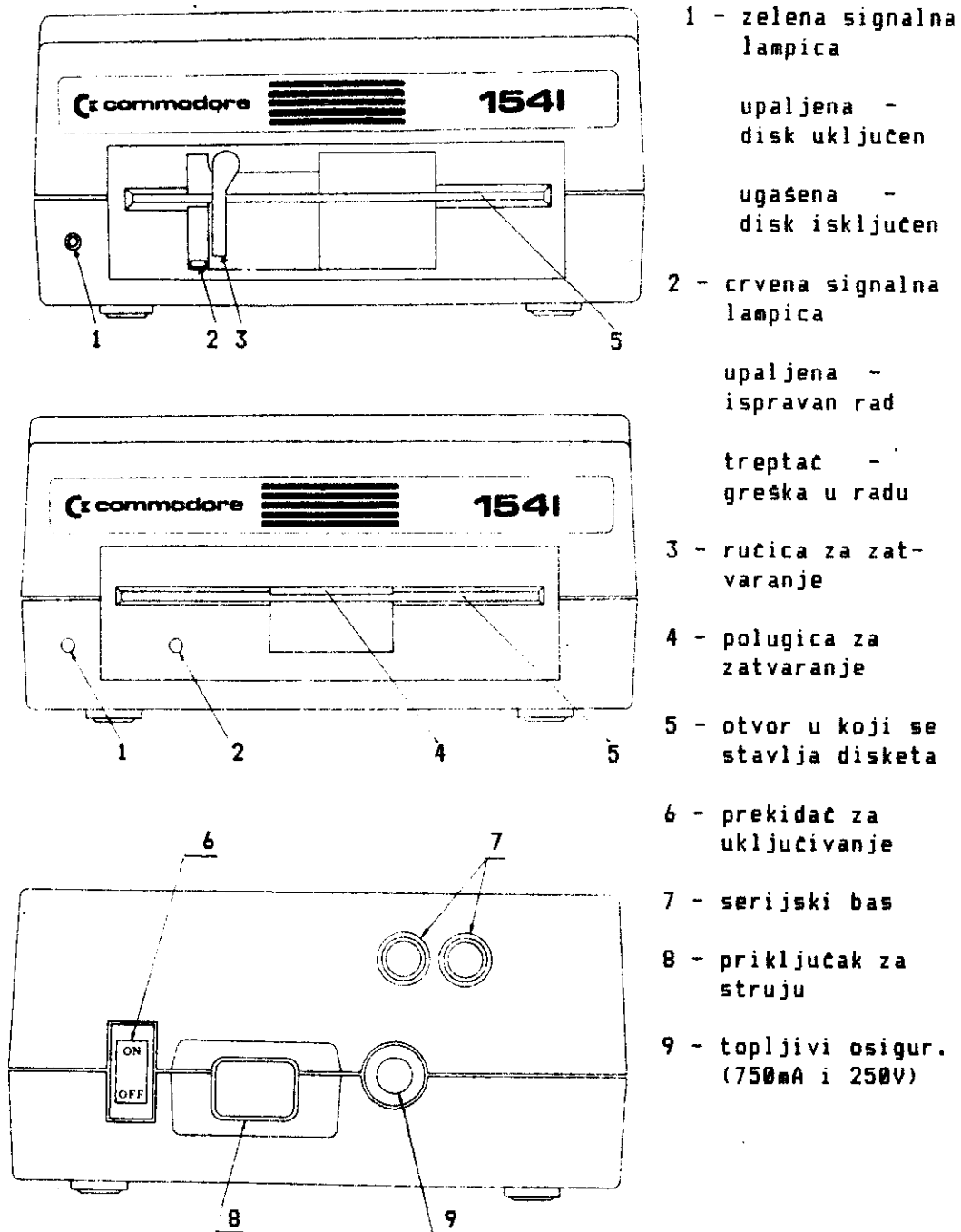
Korisnicima koji se prvi put susreću sa diskom mogu ovi podaci biti teški za razumevanje, pogotovo ako nisu upoznati sa osnovnim pojmovima kao što su ROM, RAM, bajt, bas i sl. Pošto bi upoznavanje sa svim ovim pojmovima previše proširilo sadržaj ove knjige, možemo dati nekoliko uporedjenja koja će vam pomoći da lakše shvatite šta znači raditi sa diskom.

Kad kažemo da na disketu staje 168656 bajtova u sekvencijalne datoteke, to znači da bajt možete porediti sa jednim karakterom ili još bolje slovom u vašem pismu, knjizi ili referatu. Drugim rečima na vašu disketu će stati oko 70 stranica kucanog teksta (jedna stranica - 2400 slova). Morate uzeti u obzir i podatak da na vašoj disketi može biti maksimalno 144 različita programa ili datoteke. Ali, ako imate velike programe jasno je, da će na disketu stati mnogo manje programa. Jednostavnom računicom možete doći do podatka da bi na disketu stalo 144 različita programa ili datoteke, potrebno je da oni u proseku ne budu veći od 1 KB.

Takodje treba poštovati preporuku, da diskete sa relativnim i slučajnim datotekama treba čuvati na posebnim disketama, odvojenim od programa. O razlozima za ovo biće više reči, kada budemo govorili o komandama diska.

Na sl.D - 1 dat je izgled prednje strane dva osnovna tipa disk jedinice 1541. Zelena signalna lampica pokazuje da li je disk

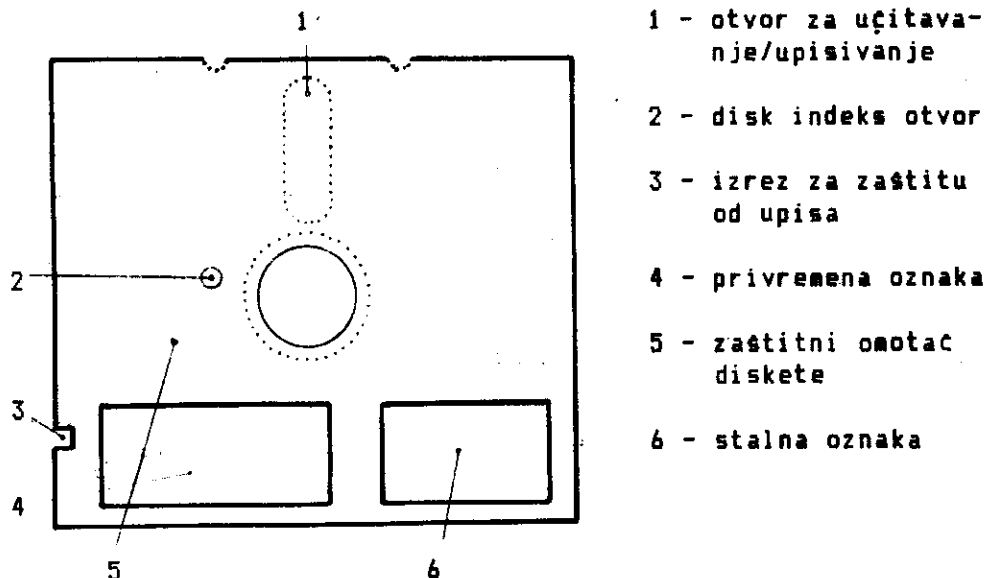
RAD SA DISKDM



Sl. D - 1 Izgled prednje i zadnje strane diska

jedinica uključena ili ne. Crvena signalna lampica pokazuje kada disk radi. Ukoliko gori neprekidno, znači da disk radi ispravno. Ukoliko se naizmenično pali i gasi, znači da disk ne radi ispravno, i da treba ponoviti traženu operaciju. Broj 5, predstavlja otvor u koji se stavlja disketa. Kod prvog tipa diska posle stavljanja diskete potrebno je polugicu br. 4, pritisnuti na dole. Kod drugog tipa diska, potrebno je ručicu br. 3, postaviti u položaj u kome se nalazi na sl. D - 1.

DISKETA



Sl. D - 2 Izgled diskete

Disketa je tanka plastična ploča, kružnog oblika, koja je prevučena namagnetisanim slojem. Kako je njena površina veoma osetljiva, ona se nalazi u zaštitnom omotaču. Da bi bilo moguće učitavati/upisivati, na zaštitnom omotaču izrezan je otvor koji služi za to. Sa tog otvora, otkrivena površina diskete vidi se oko centralnog otvora diskete (ta površina diskete se ne koristi), i kao disk indeks otvor, koji služi za sinhronizaciju brzine rotacije diskete.

Označavanje diskete

Diskete se obično prodaju u kompletu od 5, 10 ili više komada, zajedno sa nalepnicama za zaštitu od upisa i nalepnicama za označavanje. Na zaštitnom omotaču diskete nalaze se dve vrste oznaka:

- stalna oznaka, koju stavlja proizvođač diskete
- privremena oznaka koju stavlja korisnik diskete

Prva, je nazvana stalnom jer se obično ne uklanja, dok je druga privremena, pošto se može zameniti kad god je to potrebno. Da biste produžili vek trajanja diskete, kada vršite označavanje diskete, pridržavajte se sledećih pravila:

- izbegavajte da pišete na privremenoj oznaci, koja je već stavljena na disketu. Ako već morate, koristite flomaster
- pre nego što nalepите novu oznaku, odlepите staru. Dok to radite neka disketa bude u svom papirnom omotaču.

Zaštita od upisa

Disketa koja je zaštićena od upisa može da služi samo za učitavanje programa/datoteke. Ovo obezbeđenje sprečava da se nenamerno unište važne informacije, koje su ranije smeštene na disketu. Zaštita od upisa izvodi se pomoću nalepnica predviđenih za tu svrhu. Odlepите jednu nalepnicu, uvijte je i zalepите oko otvora za zaštitu od upisa, na rubu diskete. Ako želite da odstranite zaštitu, da biste dodali ili izmenili neke podatke ili programe, jednostavno odstranite pomenutu nalepnicu. Preporučujemo vam, da zaštitite sve diskete, na kojima se nalaze važni programi i datoteke.

FORMATIZACIJA NOVE DISKETE

Iz iskustva nam je poznato, da mnogi korisnici kada prvi put kupe prazne diskete, ne znaju da je njih prethodno potrebno formatizovati pre dalje upotrebe. To je potrebno, jer te iste diskete mogu da se koriste u raznim disk jedinicom, gde svaka

vrši formatizovanje diskete na svoj način.

Zato izdvajamo postupak za formatizovanje diskete, koji ćemo dati na jednom primeru:

```
10 OPEN 15,8,15
20 PRINT#15,"NEW:naziv diskete,id"
30 CLOSE 15
```

Prethodni program vrši formatizaciju nove diskete i potrebno ga je uvek izvršiti kada upotrebljavate prvi put novu disketu. Pri tom sami birate naziv diskete, sa maksimalno 16 karaktera i identifikaciju id od 2 karaktera. Postupak formatizacije traje nekoliko minuta. Na početku će te čuti neobičan zvuk koji može da vas iznenadi. Međutim, proces formatizacije se nastavlja (čujete okretanje diska), za to vreme vi možete obavljati neku drugu funkciju (listanje programa i sl.), mada se to obično ne radi. Kada je formatizacija završena možete dalje raditi sa disketom. O formatizaciji će još biti reči kada budemo govorili o disk komandama.

KOMANDE ZA RAD SA PROGRAMIMA

Učitavanje sa diskete

Ukoliko posedujete disketu koja je prethodno popunjena nekim programima (demonstracione diskete, diskete sa igrama ili gotovim programima i sl.), prvo što vas interesuje je, kako da učitavate te programe. Budući da ste već naučili kako da učitavate programe sa kasete, neće vam biti teško da naučite kako da to uradite sa diskete. Naredba LOAD, sa kojom ste se detaljno upoznali kod kasetofona:

```
LOAD "ime programa", uredjaj, sekundarna adresa
```

vrši istu funkciju i kad radite sa diskom. Jedina razlika je u tome što je broj uredjaja za kasetofon 1 i nije obavezan, dok je za disk broj uredjaja 8 i obavezan je.

NAPOMENA

Na centralnu jedinicu C-64 može se vezati maksimalno pet diskova. U praksi je najčešći slučaj da su za centralnu jedinicu vezane najviše dve disk jedinice. U tom slučaju jedan disk se identifikuje brojem 8, a drugi brojem 9. Ako je za centralnu jedinicu vezan samo jedan disk tada je uvek broj za disk 8. Taj broj može da se promeni (npr. u broj 9), koristeći softversko ili hardver-

sko rešenje (pogledajte u priručniku za rad sa diskom). U daljem izlaganju pretpostavićemo da radite sa jednom disk jedinicom čiji je broj 8.

Ako na disketi imate BASIC program pod imenom IGRA, učitavanje se vrši sledećom komandom:

```
LOAD "IGRA", B
```

Ako na disketi imate mašinski program pod imenom IGRA1 učitavanje se vrši komandom:

```
LOAD "IGRA1", B, 1
```

Sve što je rečeno o učitavanju mašinskih programa sa kasete važi i za učitavanje mašinskih programa sa diskete.

Ukoliko komandu LOAD koristite u programskom modu, možete koristiti sadržaj prethodno definisanih promenljivih, za predstavljanje imena programa ili broja uređaja. Na primer, možete koristiti komandu LOAD u obliku:

```
LOAD A$,B
```

ukoliko ste prethodno u programu postavili ime programa u promenljivu A\$, a broj uređaja u promenljivu B.

Direktorija diskete

Kada smo govorili o prednostima diska, rekli smo da je njegova najveća prednost u brzini kojom nalazi programe ili datoteke. Sem toga, kod kasetofona ste vi morali da vodite računa, o tome gde vam se nalaze programi, i da to mesto pamtite pomoću brojača okretaja. Disk vas oslobadja tog mučnog posla. Program koji se nalazi u njegovoj ROM memoriji, a zove se operativni sistem diska (DDS), između ostalog, zadužen je i za to. Ovaj program vodi računa o tome gde se smeštaju programi i datoteke, na koju stazu, u koje blokove i sve te informacije organizuje i čuva u posebno izdvojenim blokovima diskete. Ti blokovi se nazivaju MAPA SLOBODNIH (ISKORIŠĆENIH) BLOKOVA (originalni naziv je BLOCK AVAILABILITY MAP ili skraćeno BAM) i DIREKTORIJA DISKA (DIRECTORY).

Jedno od kasnijih poglavlja, izdvojeno je za detaljno objašnjenje BAM-a i DIREKTORIJE, i namenjeno je onima koji se za to posebno interesuju. Za one koji ne žele preterano da se zamaraju ovim novim pojmovima (jer im za korisnički rad to nije ni potrebno),

sada ćemo dati samo osnovne informacije o BAM-u i DIREKTORIJI.

BAM je lista u kojoj se čuvaju informacije o iskoriscenju svih 683 blokova na disketi, i smešten je otprilike negde na sredini diskete. Kad god se neki program upiše na disketu pomoću SAVE komande, ili se zatvori neka datoteka pomoću CLOSE naredbe, lista BAM-a se azurira, listom blokova koji su iskorišćeni za smeštanje ovog programa ili datoteke.

DIREKTORIJA je lista svih programa ili datoteka koje su smeštene na jednoj disketi. Ona je smeštena na disketi desno od BAM-a. Kako se na jednu disketu može smestiti maksimalno 144 programa ili datoteke, to je maksimalan broj elemenata ove liste, takodje, 144. Kad god se neki program ili datoteka smesti na disketu, i DIREKTORIJA se takodje azurira.

Ukoliko želite da vidite na ekranu kako izgleda DIREKTORIJA diskete sa kojom radite, to će vam omogućiti komanda LOAD sledećeg oblika:

```
LOAD "$", 8
```

Posle izvršenja ove naredbe DIREKTORIJA vaše diskete nalazi se u memoriji C-64. Da biste je videli na ekranu otkucajte komandu LIST, i na ekranu će se pojaviti DIREKTORIJA. Na sledećoj slici dat je izgled direktorije jedne diskete:

```
0 "PLM-SOFT          " 01 2H
5  "PROGRAM 1"      PRG
5  "PROGRAM 2"      PRG
4  "PROGRAM 3"      PRG
4  "PROGRAM 4"      PRG
646 BLOCKS FREE.
```

Sl.D - 3 - Izgled DIREKTORIJE diskete

Prvi red ovog spiska, koji je odštampan riverzno, sadrži ime diskete koje joj se dodeljuje prilikom formatizacije, i identifikacioni kod diskete. U sledećim redovima dat je spisak programa na disketi. Uz svaki program na početku linije stoji broj koji predstavlja broj blokova potrebnih za smeštanje programa. Iza imena se nalzi reč koja označava da li se radi o programu (PRG) ili datoteci (SEQ, REL, USR). Poslednji red sadrži broj slobodnih blokova. Ukoliko imate štampač direktoriju možete izlistati i na štampaču ako otkucate:

```
OPEN 4,4: CMD 4: LIST
```

Posle stampanja otkucajte:
PRINT#4: CLOSE 4

Princip dzokera

Dva znaka iz seta karaktera * i ? koriste se na specijalan način pri učitavanju programa sa diskete. Pre nego što objasnimo kako se koriste, želimo da vas podsetimo na nešto što vam je sigurno poznato, a gde se primenjuje isti princip. Kada igrate karte, u mnogim igrama, postoji jedna karta koja može da zameni bilo koju kartu. Ta karta se zove DZOKER. Na isti način * može da zameni nekoliko karaktera (reč), a ? može da zameni jedan karakter u imenu programa koji želite da učitate. Sada ćemo dati nekoliko primera u kojima se koriste ovi znaci.

LOAD "PRI*", 8

Ova komanda naci će prvi program koji počinje slovima PRI, i napuniće taj program. Ako se na disketi nalaze programi PRIMENA1, PRIMENA2 i PRIMENA3, posle izvršenja ove komande u memoriji kompjutera je program PRIMER1. Znači, kada želite da učitate program ne morate unositi celo ime programa, već možete dati jedno ili više slova kojima počinje ime programa, a ostala zameniti znakom *.

LOAD "*", 8

Ova komanda omogućava vam da učitate poslednji program koji ste prethodno učitali. Ako posle uključivanja sistema, prvi put primenjujete LOAD naredbu, i ona je gornjeg oblika biće učitani prvi program u DIREKTORIJI.

LOAD "O?AJ", 8

Ova komanda omogućava da učitate bilo koji program sa diskete, koji ima ime takvo da je drugi karakter bilo koji karakter a ostali deo reči je slova O i AJ. Npr. zaboravili ste da li se vaš program zove OVAJ ili ONAJ, ako primenite naredbu gornjeg oblika rešićete problem. Izvršiće se učitavanje programa bilo da mu je ime OVAJ ili ONAJ.

Princip dzokera može da se koristi i kod učitavanja DIREKTORIJE. Ukoliko želite da proverite da li se određeni program ili grupa programa nalazi u DIREKTORIJI, ne morate učitali celu DIREKTORIJU.

LOAD "?:PRIMER", 8

Posle primene ove komande, ako date komandu LIST na ekranu ce vam biti izlistan u DIREKTORIJI samo program PRIMER ukoliko on postoji na disketi.

LOAD "\$:PRI*", 8

Posle primene ove komande, ako date komandu LIST na ekranu ce vam biti izlistani u DIREKTORIJI svi programi koji pocinju sa PRI, npr. PRIMER1, PRIMER2, PRIMER3, PRIMENA itd.

Upisivanje na disketu

Upisivanje programa na disketu, vrši se kao i upisivanje na kasetu komandom:

SAVE "ime programa", uredjaj, sekundarna adresa

Razlika je u tome sto je broj uredjaja obavezan i taj broj je 8. Posle izvršenja ove komande, program ce biti prenet iz memorije kompjutera na disketu.

SAVE komanda u radu sa diskom ima jednu izvanrednu mogućnost, koja se sastoji u tome da se nova verzija programa može zapamtiti pod istim imenom kao i stara verzija koja već postoji na disketi. Oblik komande koji to omogućava je:

SAVE "@:ime programa", uredjaj, sekundarna adresa

Na primer, ako na disketi imate program sa imenom PRIMENA, koji ste posle učitavanja izmenili i sada želite da na disketi zapamtite novu verziju programa, a da pri tom ne zadržite staru verziju i da ime nove verzije takodje bude PRIMENA, to možete uraditi komandom oblika:

SAVE "@:PRIMENA", 8

Ako posle izvršenog upisivanja izlistate DIREKTORIJU, videćete da postoji samo jedan program sa imenom PRIMENA.

Provera upisivanja

Provera tačnosti upisa programa na disketu, vrši se komandom, koja je analogna odgovarajućoj komandi za kasetofon. Njen oblik je:

VERIFY "ime programa", uredjaj

Kao i kod kasetofona, funkcija ove komande je da uporedi sadržaj snimljen na disketi sa sadržajem koji je u memoriji kompjutera, upoređujući te sadržaje bajt po bajt. Broj uređaja je 8.

KOMANDE DISKA

Komande diska omogućavaju komunikaciju između diska i centralne jedinice. Ta komunikacija se obavlja kroz KOMANDNI KANAL, za koji je rezervisan broj 15.

Komunikaciju omogućavaju naredbe OPEN, PRINT#, INPUT# i CLOSE, sa kojima ste se prvi put upoznali u poglavlju SEKVENCIJALNE DATEO-TEKE. Naredbom OPEN vrši se otvaranje komandnog kanala. Kada se naredba PRINT# koristi sa komandnim kanalom, na taj način centralna jedinica šalje komande disku. Kada se naredbe INPUT# koristi sa komandnim kanalom, na taj način centralna jedinica čita informacije o grešci u radu sa diskom, koje su vam često potrebne.

Format naredbe OPEN u ovom slučaju je:

OPEN A, B, C

- A - broj između 1 i 255 (preporučljivo je da ne bude veći od 127), koji se pridružuje kanalu
- B - broj uređaja (broj 8, pod pretpostavkom da radite sa jednim diskom)
- C - broj kanala, za komandni kanal je rezervisan broj 15

Uobičajeno je da se broj pridružen kanalu izabere tako da bude jednak broju kanala. Tako da je oblik naredbe OPEN za komandni kanal:

OPEN 15, 8, 15

Mada može da se koristi i jedan od sledećih oblika:

OPEN 1, 8, 15
 OPEN 9, 8, 15
 OPEN 2, 8, 15

Kada radite sa datotekama neophodno je da posle završenog rada, zatvorite sve otvorene datoteke. To se isto odnosi i na komandni kanal. Ukoliko u programu otvorite nekoliko datoteka i komandni kanal vodite računa da, prvo zatvorite datoteke, a da komandni kanal zatvarate poslednji. Komandni kanal se zatvara komandom oblika:

CLOSE 15

Slanje disk komande

Postupak slanja disk komande odvija se u tri koraka:

- 1 - otvaranje komandnog kanala 15 (OPEN)
- 2 - saopštavanje komande disku (PRINT#)
- 3 - zatvaranje komandnog kanala (CLOSE)

Naredbe koje omogućavaju izvršavanje ovog postupka mogu da se saopšte disku u oba moda, direktnom i programskom.

Komanda NEW

PRVA FUNKCIJA ove komande je formatizacija nove, prethodno neupotrebljavane diskete. Na samom početku ovog poglavlja, izdvojili smo postupak za formatizaciju. Sada ćemo ga detaljnije objasniti.

Format komande: PRINT#15, "NEW:naziv diskete, id"
ili skraćeno: PRINT#15, "N:naziv diskete, id"

NAZIV DISKETE - ime sa maksimalno 16 karaktera koje se dodeljuje disketi (izbor imena zavisi od korisnika i logično je da se bira tako, da se po njemu prepoznaje sadržaj diskete)

ID - identifikacioni kod diskete, maksimalne dužine 2 karaktera. Ovaj kod je veoma značajan jer se pojavljuje ne samo u direktoriji diskete već se smesta uz svaki blok na disketi

DRUGA FUNKCIJA ove naredbe je brisanje direktorije diskete i izvršava se komandom sledećeg oblika:

PRIN#15, "NEW:naziv diskete"

Primer za korišćenje ove komande za formatizaciju diskete dat je u pomenutom postupku formatizacije.

Komanda COPY

Ova komanda omogućava kopiranje bilo kog programa sa diskete na istu disketu. Ona ne služi za kopiranje sa jedne diskete jednog diska na drugu disketu drugog diska.

Format komande: PRINT#15, "COPY:novi program = stari program"
 ili skraceno: PRINT#15, "C:novi program = stari program"

Ukoliko imate nekoliko programa na disketi, koje zelite da iskopirate tako da posle kopiranje predstavljaju jedan program, to ce vam omoguciti komanda COPY sledeceg oblika:

PRINT#15, "COPY:novi program=stari program1, stari program2,..."

1. PRIMER

Ako na disketi imate program PROMENA i zelite da iskopirate taj program na istu disketu pod imenom VARIJACIJA, postupak je sledeci:

- otvorite komandni kanal, OPEN 15,8,15
- saopstite komandu za kopiranje disku
PRINT#15, "C:VARIJACIJA=PROMENA"
- zatvorite komandni kanal, CLOSE 15

Ove tri naredbe mozete napisati u obliku programa:

```
10 OPEN 15,8,15
20 PRINT#15, "C:VARIJACIJA=PROMENA"
30 CLOSE 15
```

ili u obliku jedne linije u direktnom modu:

```
OPEN 15,8,15: PRINT#15, "C:VARIJACIJA=PROMENA": CLOSE 15
```

2. PRIMER

Ako na disketi imate programe DE01, DE02, DE03 i zelite da ih iskopirate u jedan program sa imenom CELINA, to mozete uraditi programom:

```
10 OPEN 15,8,15
20 PRINT#15, "C:CELINA=DE01,DE02,DE03"
30 CLOSE 15
```

ili jednom linijom u direktnom modu koja se formira kao u gornjem primeru.

Ovo kopiranje je veoma korisno ako hocete da spojite vise programa u jedan program, koji zelite da koristite kao celinu.

Komanda RENAME

Ova komanda omogućava promenu imena programa koji se nalaze na disketi.

Format komande: PRINT#15, "RENAME:novo ime = staro ime"
ili skraćeno: PRINT#15, "R:novo ime = staro ime"

PRIMER

Ako na disketi imate program koji se zove IMENIK a zelite da zamenite to ime imenom ADRESAR, poslužite se sledećim programom:

```
10 OPEN 15,8,15
20 PRINT#15, "R:ADRESAR = IMENIK"
30 CLOSE 15
```

ili sve tri programske linije saopštite kompjuteru u okviru jedne linije u direktnom modu.

Proces promene imena traje veoma kratko, jer je jedina promena koju treba izvršiti promena imena programa u direktoriji diskete.

Komanda SCRATCH

Ova komanda omogućava brisanje jednog ili više programa na jednoj disketi.

Format komande: PRINT#15,"SCRATCH:ime programa1,ime programa2..."
ili skraćeno: PRINT#15,"S:ime programa1,ime programa2..."

U poglavlju koje se odnosi na detaljno objašnjenje BAM-a i DIREKTORIJE, biće objašnjeno šta se dešava kada se komanda SCRATCH saopšti disk jedinici.

1. PRIMER

Ako zelite da obrišete programe DE01, DE02 i DE03 sa diskete, poslužite se sledećim programom:

```
10 OPEN 15,8,15
20 PRINT#15, "S:DE01,DE02,DE03"
30 CLOSE 15
```

ili sve smestite u jednu liniju u direktnom modu.

2. PRIMER

Princip dzokera važi i za ovu komandu, tako da možete izbrisati

sve programe sa diskete koji počinju nekim odredjenim slovom. Brisanje programa iz prethodnog primera može da se izvrši i na sledeći način:

```
10 OPEN 15,8,15
20 PRINT#15, "S:D*"
30 CLOSE 15
```

ili u obliku jedne linije u direktnom modu.

Komanda INITIALIZE

Postoje neke greške u radu sa diskom, koje mogu da vam onemoguće dalji rad. Ova komanda vraća disk u stanje u kome je bio kada je prvi put uključen u toku rada, znači, vrši njegovu ponovnu inicijalizaciju.

Format komande: PRINT#15, "INITIALIZE"
ili skraćeno: PRINT#15, "I"

I za primenu ove komande, potrebno je prethodno otvoriti komandni kanal 15, a zatim posle izvršenja ove komande ga zatvoriti. Ako želite da izvršite ponovnu inicijalizaciju vašeg diska otkucajte:

```
OPEN 15,8,15: PRINT#15, "I": CLOSE 15
```

i pritisnite tipku RETURN.

Komanda VALIDATE

Ako disketu koristite duže vreme, postupak upisivanja programa i datoteka i brisanja programa i datoteka ponovićete nebrojeno mnogo puta. Kao posledica toga pojaviće vam se neiskorišćeni prazni prostori na disketi, međusobno odvojeni. Ti prostori su suviše mali (veličine dva-tri bloka), da bi se u njih mogao smestiti neki novi program. Tako vam se može desiti, da kad izlistate direktoriju, vidite da imate 80 praznih blokova, međutim, kada hoćete da upišete program, npr. dužine 30 blokova, počete naizmenično da se pali i gasi crvena kontrolna lampica. Znači, tih 80 blokova rasuto je po celoj disketi, u odvojene male prostore. Jasno, da vam se ne isplati da tako gubite prostor na disketi. Stoga, morate izvršiti reorganizaciju vaše diskete, tako da se ti mali prostori ujedine u jedinstven prostor, koji je moguće koristiti. Reorganizacija diskete vrši se pomoću komande VALIDATE.

Format komande: PRINT#15, "VALIDATE"

ili skraćeno: PRINT#15, "V"

PRIMER

Reorganizacija diskete vrši se programom:

```
10 OPEN 15,8,15
20 PRINT#15, "V"
30 CLOSE 15
```

ili linijom u direktnom modu:

```
OPEN 15,8,15; PRINT#15, "V"; CLOSE 15
```

Šta sto vrši reorganizaciju diskete, ova komanda ima još jednu značajnu ulogu. Kada radite sa datotekama prvo ih morate otvoriti (OPEN), i posle završene obrade obavezno ih morate zatvoriti (CLOSE). Ukoliko pri upisu podataka u datoteku, ne zatvorite datoteku na kraju rada, datoteka će biti upisana na disketu, njeno ime će se nalaziti u direktoriji diskete, međutim, više nećete moći ništa sa njom da radite (npr. da je ucitate). Disk jedinica će vam "skrenuti pažnju" na to, da sa tom datotekom nešto nije u redu, time što će neposredno ispred tipa datoteke u direktoriji diskete, staviti *. Npr. ako sekvencijalna datoteka PODACI nije bila pravilno zatvorena u programu, kada izlistate direktoriju videćete:

```
0 "PLM-SOFT"          " 01 2A
5 "PROGRAM 1"        PRG
5 "PROGRAM 2"        PRG
4 "PROGRAM 3"        PRG
5 "PROGRAM 4"        PRG
20 "PODACI"          *SEQ
623 BLOCKS FREE.
```

Datoteku PODACI nećete više moći da koristite, a 20 blokova koje zauzela ova datoteka takodje, nećete više moći da koristite. Ukoliko pokušate da izbrišete ovu datoteku SCRATCH komandom, videćete da je to nemoguće. Pomislićete da su ti blokovi zauvek izgubljeni. Rešenje ipak postoji, a to je opet primena VALIDATE komande. Примените opet navedeni program. Čućete, za kratko, okretanje u disk jedinici, a zatim će disk biti miran nekoliko sekundi. Taj postupak će se ponoviti nekoliko puta. Ako vam je disketa skoro potpuno popunjena, taj proces će trajati nekoliko minuta. Strpljivo sačekajte, jer će posle završene reorganiza-

cije, svi prazni blokovi biti opet dostupni za rad.

VAZNO UPOZORENJE

Ako vam se na disketi nalaze slučajne (RANDOM) ili relativne (RELATIVE) datoteke, a primenite VALIDATE komandu svi blokovi ovih datoteka bice takodje reorganizovani, a ove datoteke unistene. Stoga, slučajne i relativne datoteke držite na posebne disketama i nikada ne primenjujite VALIDATE komandu na te diskete.

Provera greske u radu sa diskom

Dok radite sa diskom, vrlo često mogu da nastupe greske. Informacija o gresci ucitava se kroz kanal 15, koji se tada naziva kanal greske. Ucitavanje greske izvršite pomocu sledeceg programa:

```
10 OPEN 15,0,15 : INPUT#15, G, PG$, S, B
20 PRINT "I/O GRESKA : PRINT "BROJ GRESKE....."; G
30 PRINT "TEKST GRESKE...."; PG : PRINT "STAZA....."; S
40 PRINT "BLOK....."; B : CLOSE 15
```

Prvi podatak koji se ucitava kroz kanal greske u promenljivu G, je broj greske. Drugi, koji se ucitava u promenljivu PG\$, je kratka poruka u cemu je greska. Treći je staza, a četvrti blok gde je napravljena greska.

DDS SUPPORT PROGRAM

Na demonstracionoj disketi koju dobijate uz kompjuter nalazi se program pod imenom DDS SUPPORT PROGRAM ili WEDGE, koji omogućava da na jednostavniji način koristite disk komande. Kada uključite sistem, ucitajte sa demonstracione diskete program DOS BOOT koji omogućava da se ucita program DDS 5.1 (DDS WEDGE). Direktno ucitavanje programa DOS 5.1 pomocu naredbe LOAD nije moguće.

Posle startovanja ovog programa, rad sa diskom je mnogi lakši, jer su disk komande pojednostavljene. Tabela D - 2 daje uporedni pregled disk komandi iz DDS 5.1 i standardnih komandi.

Kao što se iz pregleda vidi, disk komande iz DDS 5.1 su jednostavnije i lakše za upotrebu. Sem toga, prednost ovog programa je i u tome što ima komande koje omogućavaju izlistavanje direktorije na ekranu, bez ucitavanja direktorije. Znači, ako imate u memoriji kompjutera neki program, koji još

niste sacuvali na disketi, a zelite da vidite direktoriju diskete (da bi npr. videli, da li ima prostora za smeštanje programa),

PREGLED DISK KOMANDI

DOS 5.1	STANDARDNI DOS	FUNKCIJA
/ime programa	LOAD"ime programa",8	ucitava program
fime programa	-	ucitava program i startuje ga
%ime programa	LOAD"ime programa",8,1	ucitava masin. prog.
>\$	-	izlistava direktoriju diskete bez ucitavanja u memoriju
←ime programa	SAVE"ime programa",8	upisuje program
>I	PRINT#15,"I"	inicijalizac. diska
>S:ime programa	PRINT#15,"S:ime progr."	brisanje programa
>\$:ime programa	-	izlistava samo 1 program u direktoriji
>#9	-	menja broj uredjaja 8 u 9
>	-	ucitava grešku

Tab. D - 2 - Pregled disk komandi

komanda >\$ ce vam to omoguciti. Ili, ako zelite da taj program sacuvate pod imenom PRIMENA, a niste sigurni da li na disketi vec postoji program sa tim imenom, to mozete proveriti komandom >\$:PRIMENA. Ako postoji na disketi, taj ce program biti izlistan u direktoriji. Sledeca prednost, je postojanje komande koja ucitava i automatski startuje program (fime programa), i komanda koja menja broj uredjaja 8 u 9. Komanda > omogucava proveru greške na mnogo jednostavniji nacin, nego kad se radi iz standardnog DOS-a. U standardnom DOS-u morate da napravite program koji vam omogucava obavljanje ove funkcije, dok je u DOS-u 5.1 moguće ucitavanje greške kroz kanal 15 u direktnom modu.

SEKVENCIJALNE DATOTEKE NA DISKETI

SEKVENCIJALNE DATOTEKE NA DISKETI

Sekvencijalne datoteke na disketi organizuju se na isti način kao i na traci. Smještanje podataka u sekvencijalne datoteke na disketi, analogno je smještanju podataka u sekvencijalne datoteke na kaseti. Isto važi i za naredbe. Jedina razlika je u obliku naredbe DPEN.

Oblik naredbe DPEN je:

OPEN A,B,C,"ime datoteke, tip, pravac" gde je:

- A - broj pridružen datoteci (isto kao kod datoteke na kaseti)
- B - broj koji označava perifernu jedinicu na kojoj se nalazi datoteka. Za disk jedinicu taj broj je 8
- C - kanal za manipulisanje podacima, broj od 2-14. Uobičajeno je da se koristi isti broj za datoteku i kanal.

IME DATOTEKE - ime koje se dodjeljuje datoteci

TIP - tip datoteke, koji je za sekvencijalnu datoteku označen sa SEQ ili S

PRAVAC - označava da li se upisuje u datoteku ili se učitava iz datoteke. Ako se upisuje tada je PRAVAC = WRITE ili PRAVAC = W. Ako se učitava, tada je PRAVAC = READ ili PRAVAC = R.

Program, koji sledi, služi za formiranje sekvencijalnih datoteka na disketi.

Program za formiranje

```

10 PRINT "(CLR)(CUR DN) FORMIRANJE DATOTEKE NA DISKETI"
20 PRINT "(CUR DN) SLOB JE OBLIKA"
30 PRINT
40 PRINT "PREZIME*MEŠ-RODJE*STAROST*POL*PLATA*"
45 PRINT
50 INPUT "BROJ SLOBOVA U DATOTECI";N
60 DIM A$(N),B$(N),C(N),D$(N),E(N)
70 INPUT "IME DATOTEKE";NAZIV$
80 OPEN S,8,S,NAZIV$+"$,S,W"
85 R$=CHR$(13)
90 FOR X=1 TO N
100 INPUT "PREZIME";A$(X)
110 INPUT "MESTO RODJENJA";B$(X)
120 INPUT "STAROST";C(X)
130 INPUT "POL";D$(X)
140 INPUT "PLATA";E(X)
150 PRINT#5,A$(X)R$B$(X)R$C(X)R$D$(X)R$E(X)
155 NEXT X
160 CLOSE 5

```

SEKVENCIJALNE DATOTEKE NA DISKETI

```
170 FOR X=1 TO N
180 PRINT A$(X)TAB(15)B$(X)TAB(25)C(X);
185 PRINT TAB(30)D$(X)TAB(33)E(X)
190 NEXT X
200 PRINT;PRINT" PRITISNI NEKU TIPKU"
210 GET B$: IF B#="" THEN 210
220 END
```

Ako sada zelite da ucitate slogove prethodno formirane datoteke poslužite se sledecim programom.

Program za ucitavanje

```
10 INPUT"IME DATOTEKE";NAZIV$
20 INPUT"BROJ SLOBOVA";N
30 DIM A$(N),B$(N),C(N),D$(N),E(N)
40 OPEN "5,6,5,NAZIV$ + ".B,R"
45 FOR X=1 TO N
50 INPUT#5, A$(X),B$(X),C(X),D$(X),E(X)
60 NEXT X
70 CLOSE#5
80 FOR I=1 TO N
90 PRINT A$(I)TAB(15)B$(I)TAB(25)C(I);
95 PRINT TAB(30)D$(I)TAB(35)E(I)
100 NEXT I
110 END
```

S L U C A J N E D A T D T E K E

U pocetku ce vam biti sasvim dovoljno da znate da radite sa sekvencijalnim datotekama. Medjutim, vrlo brzo cete i sami uvideti da sekvencijalne datoteke nisu najpogodnije za mnoge obrade podataka. Na primer, obradjujete datoteku PODACI, koja ima slog oblika:

```
*****
* IME I PREZIME * OC-1 * DC-2 * OC-3 * OC-4 *
*****
```

DC-1, OC-2, OC-3, OC-4 su ocene učenika iz cetiri predmeta. Ukoliko zelite da izmenite sadržaj nekog sloga datoteke PODACI, npr. slog:

```
*****
* ZDRAN BABIC * 4 * 3 * 4 * 2 *
*****
```

tako da polje OC-1 sadrzi 5, a polje OC-2 sadrzi 3 postupak se izvodi u sledecim fazama:

- učitati celu datoteku PODACI
- izmeniti sadržaje polja OC-1 i OC-2 u navedenom slogu
- upisati ponovo celu datoteku na kasetu ili disketu

Očigledno je, da se samo zbog jedne izmene obavlja nepotrebno učitavanje cele datoteke, a zatim ponovno upisivanje cele datoteke. Ako datoteka ima 500 slogova jasno je koliko ce dugo taj postupak da traje. Znači, svaka obrada koja zahteva izmenu sadržaja slogova datoteke, nepogodna je za sekvencijalne datoteke.

Ili na primer, ako hocete da učitete poslednji slog da bi ste videli njegov sadržaj, morate da učitete sve slogove, koji prethode tom slogu, sto je takodje, nepotrebno gubljenje vremena.

Ove nedostatke sekvencijalnih datoteka ispravljaju slucajne i relativne datoteke. Njih je isključivo moguće organizovati na disketi, jer se slogovima ovih datoteka pristupa na slucajan način (moguć je pristup bilo kom slogu). Slučajan način pristupa podacima moguć je samo ako se podaci organizuju na disketi. Da bi bolje razumeli objašnjenja, koja se odnose na slucajne datoteke podsetite se sadržaja poglavlja koja se odnose na izgled diskete, smestanje podataka na disketu i komande koje se šalju disku.

RAD SA SLUCAJNIM DATOTEKAMA

Rad sa slučajnim datotekama omogućavaju komande DOS-a. To su komande za učitavanje ili upisivanje podataka direktno u bilo koji blok, bilo koje staze na disketi, i komande za davanje informacija o iskorišćenim i slobodnim blokovima na disketi.

Slučajne datoteke se kreiraju tako, da se direktno adresiraju blokovi na disketi u koje se upisuju podaci ili iz kojih se učitavaju podaci. Pri tom se koriste baferi diska, kojih ima 8 (2K RAM memorije diska podeljeno je na 8 delova od po 256 bajtova, i te delove zovemo baferi). 4 od tih 8 bafera koriste BAM, disk kontroler, I/O komandni kanal i sistemske promenljive. Preostala 4 se mogu koristiti za rad sa slučajnim datotekama. Znači, maksimalan broj bafera koji može biti otvoren u jednom trenutku je 4.

Rad sa slučajnim datotekama može se podeliti na tri dela:

- 1 - kreiranje datoteka, odnosno, upisivanje datoteka na disketu
- 2 - učitavanje datoteka sa diskete
- 3 - izmena sadržaja slogova datoteke

Postupak kreiranja slučajne datoteke sastoji se iz sledećih sedam koraka:

1. korak - otvaranje kanala između C64 i bafera u disk jedinici
2. korak - kopiranje sloga podataka u taj bafer počev od prvog karaktera sloga
3. korak - nalaženje sledećeg slobodnog bloka na disketi
4. korak - saopštavanje DOS-u da želite da upisujete sadržaj u taj blok
5. korak - prenos svih podataka iz bafera u taj blok
6. korak - formiranje indeksnog niza koji povezuje blok i sifru sloga, koja predstavlja polje po kome će se pretraživati datoteka. U primeru datoteke PODACI to može biti PREZIME
7. korak - pamćenje indeksnog niza u obliku sekvencijalne datoteke.

Postupak učitavanja kreirane slučajne datoteke sastoji se iz sledećih pet koraka:

1. korak - učitavanje indeksnog niza u BASIC niz definisan u programu koji vrši učitavanje datoteke
2. korak - otvaranje kanala izmedju bafera diska i C64 centralne jedinice
3. korak - traženje indeksa koji odgovara sifri sloga, koji treba učitati. Uz taj indeks su pridružene informacije o stazi i bloku, gde se nalazi traženi slog
4. korak - čitanje celog bloka, koji je odredjen prethodnim korakom, iz diskete u bafer
5. korak - prenošenje sadržaja bafera u BASIC promenljivu predvidjenu za to, programom koji vrši učitavanje datoteke.

Postupak izmene sadržaja slogova slučajne datoteke sastoji se iz sledeca cetiri koraka:

1. korak - čitanje prvog bloka, čiji se sadržaj želi izmeniti, u bafer diska, na nacina kako je to uradjeno u prva cetiri koraka za učitavanje datoteke
2. korak - pozicioniranje na deo bloka u baferu, koji treba izmeniti, upisivanjem novog sadržaja
3. korak - kopiranje novog sadržaja iz BASIC promenljive u bafer diska, i to samo u deo bloka koji je specificiran za izmenu
4. korak - upisivanje sadržaja bafera nazad u blok diskete na isto mesto sa koga je i učitano....

Sada ćemo se upoznati sa komandama i naredbama, koje omogućavaju izvršavanje prethodnih postupaka.

NAREDBE I KOMANDE

Otvaranje slučajne datoteke

Opšti oblik ove naredbe je

OPEN A,B,C,"#D" gde je:

- A - broj datoteke (0 - 127)
- B - broj periferiske jedinice (0 za disk)
- C - broja kanala za prenos podataka (2 - 14)
- D - broj bafera, koji može biti izostavljen. DOS tada automatski bira jedan bafer za rad.

Zatvaranje datoteke, upisivanje i učitavanje vrši se pomoću nare-

SLUCAJNE DATOTEKE

dbi CLOSE, PRINT#, INPUT#, GET#, koje imaju isti oblik kao i za sekvencijalne datoteke.

Disk komande

BLOCK-READ

Format komande: PRINT#15, "BLOCK-READ:"K;D;S;B
ili skraceno: PRINT#15, "B-R:"K;D;S;B

- K - kanal za citanje slucajne datoteke
- D - broj koji označava disk - uvek 0
- S - broj staze
- B - broj bloka

Ova komanda čita sadržaj jednog lociranog bloka (brojem staze i brojem bloka) i smešta ga u bafer diska. Koristi se na sledeći način:

- 1 - otvoriti komandni kanal (OPEN 15,0,15)
- 2 - otvoriti kanal za prenos podataka slucajne datoteke (npr. OPEN 3,0,3,"#")
- 3 - definisati lokaciju bloka i učitati sadržaj bloka (PRINT#15,"B-R:"C;D;S;B)
- 4 - posle izvodenja ove komande, moguće je komandom GET# ili INPUT# učitati podatke iz bafera
- 5 - proveriti sadržaj status registra ST, da bi znali da li je kraj podataka
- 6 - zatvoriti sve otvorene kanale, odnosno datoteke

Primer koji sledi ilustruje učitavanje jednog bloka sa diskete. To je blok 4 sa staze 15.

```
10 OPEN 15,0,15
20 OPEN 3,0,3,"#"
30 PRINT#15,"B-R:"5;0;15;4
40 S$=""
50 GET#5,A$
60 FOR I=4 TO 143 STEP 4
70 IF ST=0 THEN S$=S$+A$
80 NEXT I
90 PRINT "KRAJ UCITAVANJA"
100 CLOSE 5
110 CLOSE 15
```

Objašnjenje programa:

- 10 - otvaranje komandnog kanala
- 20 - otvaranje slučajne datoteke
- 30 - učitavanje 4-tog bloka sa 15-te staze u bafer diska
- 40 - popunjavanje BASIC promenljive S\$, koja treba da primi sadržaj bloka, praznim nizom
- 50 - definisanje brojača petlje, koja služi za učitavanje sadržaja bloka (256 karaktera)
- 60 - uzimanje jednog karaktera učitanoog bloka i smestanje tog karaktera u BASIC promenljivu A\$
- 70 - ispitivanje sadržaja status registra ST. Sve dok je on jednak 0, traje učitavanje. Kada postane različit od 0 to je indikator za kraj učitavanja. Dodavanje sledeceg učitanoog karaktera na prethodni sadržaj promenljive S\$
- 80 - kraj petlje
- 90 - stampanje poruke na ekranu
- 100 - zatvaranje slučajne datoteke
- 110 - zatvaranje komandnog kanala

BLOCK-ALLOCATE

Format komande: PRINT#15,"BLOCK-ALLOCATE:"D;S;B
 ili skraćeno: PRINT#15,"B-A:"D;S;B

- D - broj koji označava disk (0)
- S - broj staze
- B - broj bloka

Ova komanda proverava da li je blok definisan brojem staze i brojem bloka, slobodan ili već iskorišćen. Ukoliko je blok slobodan, ova komanda menja BAM i time alocira blok. Ukoliko je blok već iskorišćen, BAM se ne menja, a kroz kanal greške (15), disk saopštava koji je sledeći slobodan blok i staza. Ukoliko ne postoji ni jedan slobodan blok, disk saopštava broj staze 0 i broj bloka 0. Ukoliko je blok koji ste želeli da alocirate slobodan, kroz kanal greške disk saopštava poruku "OK". Komanda se koristi na sledeći način:

- 1 - otvoriti komandni kanal (OPEN 15,8,15)
- 2 - definisati broj bloka i broj staze i proveriti da li je taj blok slobodan (PRINT#15,"B-A:"0;S;B)
- 3 - proveriti kanal greške (INPUT#15,B;PG\$;S;B)
 0 - kod greške
 PG\$ - poruka o grešci

S - broj staze

B - broj bloka

Ukoliko je blok slobodan, tada je $G=0$, $PG\$\text{"OK"}$, $S=0$, $B=0$, blok se locira na stazu S i blok B iz koraka 2.

Ukoliko je blok već iskorišćen, tada je $G=65$, $PG\$\text{"NOBLOCK"}$, $S=\text{broj sledeće slobodne staze}$, $B=\text{broj sledećeg slobodnog bloka}$. Blok se alokira na stazu i blok dat ovim korakom.

4 - zatvoriti kanal (CLOSE 15)

BLOCK-WRITE

Format komande: PRINT#15,"BLOCK-WRITE:"K;D;S;B

ili skraćeno: PRINT#15,"B-W:"K;D;S;B

K - kanal za upisivanje slučajne datoteke

D - broj koji označava disk (0)

S - broj staze

B - broj bloka

Ova komanda upisuje sadržaj jednog bloka slučajne datoteke u bafer diska, a zatim se sadržaj bafera upisuje pomoću naredbe PRINT# u prethodno alokirani blok. Komanda se koristi na sledeći način:

- 1 - uraditi alociranje bloka komandom BLOCK-ALLOCATE (to je preporučljivo, da biste izbegli upisivanje preko već potpunih blokova, i time uništavanje sadržaja već postojećih programa, datoteka ili čak i BAM-a)
- 2 - ako je $PG\$\text{"OK"}$ ili ako se dobije poruka o sledećem slobodnom bloku, postupak se nastavlja
- 3 - otvoriti slučajnu datoteku za upis blokova (OPEN 5,8,5,"#")
- 4 - upisati podatke u bafer naredbom PRINT# (PRINT#5,A)
- 5 - podaci su sada u baferu. Upis u alokirani blok na disketi vrši se komandom PRINT#15,"B-W:"5;0;S;B
- 6 - zatvoriti slučajne datoteke i komandni kanal (CLOSE 5, CLOSE 15)

Upis blokova može da se vrši na dva načina, bez alociranja blokova i sa alociranjem blokova. Ukoliko se vrši upis bez alociranja bloka, morate znati koji su blokovi slobodni, a koji zauzeti. Ako upisujete na potpuno praznu disketu, slobodni su svi blokovi sem blokova na 18-toj stazi koji služe za BAM i direktoriju diskete. U tom slučaju možete upisivati, bez straha od greške, na sve ostale staze. Međutim, svaki sledeći upis može

da izazove uništavanje već popunjenih blokova, zbog toga je jedini pravilan način, da se upisivanje sadržaja blokova vrši tek posle izvršenog alociranja.

Primer koji sledi ilustruje alociranje i upis podataka u prvi slobodan blok.

```

10 OPEN 15,0,15
20 OPEN 5,0,5,"#"
30 PRINT#5,"PODACI"
40 S=1;B=1
50 PRINT#15,"B-A:"0;B;B
60 INPUT#15,0,PS,C,D
70 IF 0=65 THEN S=C;B=D;GOTO 50
80 PRINT#15,"B-W:"5;0;S;B
90 CLOSE 5; CLOSE 15

```

Objašnjenje programa:

- 10 - otvaranje komandnog kanala (kanala greske)
- 20 - otvaranje slučajne datoteke
- 30 - upisivanje string promenjive u bafer
- 40 - postavljanje broja staze i broja bloka na 1
- 50 - alociranje bloka 1 na stazi 1
- 60 - učitavanje greske kroz kanal greske
- 70 - ako je kod greske G#65 program propada na sledecu programsku liniju. Ako je kod greske G=65, S=C(sledeca slobodna staza) a B=D(sledeci slobodan blok) program se vraća u liniju 50 i vrši alociranje bloka D na stazi C. Ponovo se učitava kanal greske i ukoliko ne postoji greska prelazi se na sledecu programsku liniju
- 80 - upisivanje sadržaja bafera u prethodno alocirani blok
- 90 - zatvaranje komandnog kanala i slučajne datoteke

NAPOMENA

U priručniku za rad sa diskom na strani 31 dat je program za upisivanje bloka sa prethodnim alociranjem. U tom programu napravljena je greska u liniji 80. Ukoliko startujete ovaj program moći ćete da upišete sadržaj bafera samo u blok 1 na stazi 1, ukoliko je on slobodan. Da biste mogli da upišete sadržaj bafera u bilo koji slobodan blok, poslužite se prethodnim programom.

BLDCK-FREE

Format komande: PRINT#15,"BLOCK-FREE:"D;S;B
ili skraćeno: PRINT#15,"B-F:"D;S;B

- D - broj koji označava disk (0)
- S - broj staze
- B - broj bloka

Dva komanda je suprotna komandi BLOCK-ALLOCATE, po tome što oslobadja blokove, koje više ne želite da koristite. Ona je slična SCRATCH komandi, po tome što u stvari ne briše podatke na disketi, već samo oslobadja prostor na disketi za upis novih podataka. Ona samo menja mapu iskorišćenih blokova (BAM). Koristi se na sledeći način:

- 1 - otvoriti komandni kanal (OPEN 15,8,15)
- 2 - definisati broj bloka i staze koji treba osloboditi (npr. PRINT#15,"B-F:"0;1;14 oslobadja 14-ti blok na stazi 1)

BUFFER-POINTER

Bafer pointer čuva informaciju o poziciji na stazi, na kojoj je upisan poslednji deo podataka (kada se vrši upisivanje iz bafera u određeni blok). Isto tako, on čuva informaciju o poziciji na stazi, sa koje se učitava sledeći deo podataka (kada se vrši učitavanje sadržaja bafera u promenljive BASIC-a). Menjajući sadržaja bafer pointera, moguće je slučajno pristupati pojedinačnim bajtovima unutar jednog bloka. Na ovaj način moguće je podeliti svaki blok na slogove.

Na primer, slogovi datoteke PODACI sadrže po 50 karaktera. Stoga je moguće podeliti svaki blok slučajne datoteke, na 5 slogova. Znaajući broj staze, broj bloka i broj sloga moguće je pristupati svakom pojedinačnom slogu.

Format komande: PRINT#15,"BUFFER-POINTER:"K;L
ili skraćeno: PRINT#15,"B-P:"K;L

- K - kanal za prenos podataka
- L - lokacija unutar bloka

Na primer, komanda PRINT#15,"B-P:"3;50, postavlja bafer pointer na 50 karakter u okviru bloka.

PRIMERI ZA RAD SA SLUCAJNIM DATOTEKAMA

Jedan od najvažnijih problema vezanih za slučajne datoteke, je taj što ne postoji način da se automatski pamti koji su blokovi

na disketi, iskorišćeni za smestanje određene datoteke. Najčešće primenjivani način, koji rešava ovaj problem je formiranje sekvencijalne datoteke, koja se pridružuje slučajnoj datoteci. Slogovi sekvencijalne datoteke su oblika:

```
*****
* BROJ STAZE * BROJ BLOKA *
*****
```

To znači, da za rad sa slučajnom datotekom, treba da budu otvorena tri kanala:

- komandni kanal
- kanal za slučajnu datoteku
- kanal za sekvencijalnu datoteku

Sem toga, to znači da se dva bafera popunjavaju istovremeno.

Primer koji sledi omogućava upisivanje 10 blokova slučajne datoteke.

```
10 OPEN 15,8,15
20 OPEN 5,8,5, "*"
30 OPEN 6,8,6, "0:SIFRE,S,W"
40 FOR I=1 TO 10
50 INPUT{(CLR){CUR DN) UNEBI SADRZAJ BLOKA";A$
60 PRINT#5,A$CHR$(13)I
70 S=1;B=1
80 PRINT#15,"B-A: "0;S;B
90 INPUT#15,8,P8$,C,D
100 IF S=65 THEN S=C;B=D;80TO 80
110 PRINT#15,"B-W: "5;0;S;B
120 PRINT#6,SCHR$(13)B
130 NEXT I
140 CLOSE 6: CLOSE 5: CLOSE 15
```

Objašnjenje programa:

- 10 - otvaranje komandnog kanala
- 20 - otvaranje slučajne datoteke 5, i dodeljivanje jednog bafera toj datoteci
- 30 - otvaranje sekvencijalne datoteke SIFRE
- 40 - postavljanje brojača I, petlje koja omogućava upis 10 blokova
- 50 - unošenje sadržaja bloka
- 60 - prenos A\$ i I u bafer
- 70 - postavljanje broja staze i broja bloka na 1

- 80 - 100 - alociranje bloka na prvu slobodnu poziciju
- 110 - upis sadržaja bafera u alocirani blok
- 120 - upis broja staze i broja bloka u sekvencijalnu datoteku 6
- 130 - povećanje brojača I za 1 i povratak u liniju 40
- 140 - zatvaranje sva tri kanala

Primer koji sledi omogućava učitavanje 10 blokova slučajne datoteke.

```

10 OPEN 15,8,15
20 OPEN 5,8,5, "#"
30 OPEN 6,8,6, "SIFRE,S,R"
40 FOR I=1 TO 10
50 INPUT#6,S,B
60 PRINT#15,"B-R:";S;B
70 INPUT#5,A$,X
80 PRINT A$;" "X
90 PRINT#15,"B-F:";B
100 NEXT I
110 CLOSE 6: CLOSE 5
120 PRINT#15,"S:SIFRE"
130 CLOSE 15

```

Objasnjenje programa:

- 10 - otvaranje komandnog kanala
- 20 - otvaranje slučajne datoteke 5 i dodela bafera
- 30 - otvaranje sekvencijalne datoteke 6, SIFRE
- 40 - postavljanja brojača I, petlje koja omogućava učitavanje 10 blokova slučajne datoteke
- 50 - učitavanje broja staze i broja bloka, prvog upisanog
- 60 - prenos sadržaja bloka sa diskete u bafer diska
- 70 - učitavanje sadržaja bafera u BASIC promenljive A\$ i X
- 80 - štampanje sadržaja A\$ i X na ekranu
- 90 - oslobadjanje bloka iz koga je vršeno učitavanje
- 100 - povećavanje brojača I za 1 i povratak u liniju 40
- 110 - zatvaranje datoteka 5 i 6
- 120 - brisanje datoteke SIFRE
- 130 - zatvaranje komandnog kanala

Prethodna dva primera su samo ilustrativni primeri za upisivanje odnosno učitavanje 10 blokova slučajne datoteke. Zbog toga se u drugom primeru oslobadaju blokovi alocirani u prvom primeru i briše datoteka SIFRE. Kada budete pravili svoje slučajne datoteke razumljivo je, da necete vršiti oslobadjanje prethodno alociranih blokova i brisanje sekvencijalne datoteke. Sem toga, ovi primeri pokazuju da je moguć slučajan pristup samo blokovima u celini. To znači da ovakav način rada ima smisla ako nam je potreban slučajan pristup grupama podataka dužine 1-og bloka. To je čest slučaj

u radu sa mašinskim programima. Medjutim, ako je datoteka organizovana u slogove koji su znatno kraći od 256 karaktera, npr. datoteka PODACI, da bi pristupali slogovima na slucajan način, morali bismo upisivati po jedan slog u jedan blok, cime bi gubili mnogo prostora na disketi. Npr. slog datoteke PODACI dugačak je 50 karaktera, znači, gubili bismo po 206 bajtova u svakom bloku. Da bismo prevazišli ovaj problem, koristimo komadu BUFFER-POINTER za pozicioniranje u okviru jednog bloka u baferu.

Primer koji sledi ilustruje upisivanje 2 bloka slucajne datoteke, gde je svaki blok podeljen na 5 slogova dužine 50 karaktera.

```

10 OPEN 15,8,15
20 OPEN 5,8,5, "#"
30 OPEN 6,8,6, "SIFRE,S,W"
40 FOR I=1 TO 2
50 FOR J=1 TO 5
60 INPUT"(CLR)(CUR DN) UNESITE SLOG";A$
70 PRINT#15,"B-P: "5,(J-1)*50+1
80 PRINT#5,A$
90 NEXT J
100 S=1:B=1
110 PRINT#15,"B-A: "0,S;B
120 INPUT#15,8,PB$,C,D
130 IF 8=65 THEN S=C:B=D:GOTO 110
140 PRINT#15,"B-W: "5,0;S;B
150 PRINT#6,SCHR$(13)B
160 NEXT I
170 CLOSE 6: CLOSE 5: CLOSE 15

```

Objasnjenje programa:

- 10 - 30 - otvaranje komandnog kanala i datoteka
- 40 - 50 - postavljanja brojača petlje
- 60 - unosenje sloga datoteke u BASIC promenjivu A\$
- 70 - pozicioniranje bafera na poziciju 1, 51, 101, 201 u zavisnosti od brojača J. Na ovaj način slogovi će biti upisivani počev od tih pozicija, u okviru jednog bloka. Pri učitavanju bice moguć pristup svakom od ovih slogova
- 80 - prenos sadržaja promenljive A\$ u bafer počev od definisanih pozicija
- 90 - kraj petlje za J
- 100 - 130 - alociranje bloka
- 140 - upis sadržaja bafera u blok
- 150 - kraj petlje za I
- 170 - zatvaranje kanala i datoteka

Primer koji sledi ilustruje učitavanje 2 bloka slucajne datoteke, pri čemu svaki blok sadrži 5 slogova dužine 50.

```

10 OPEN 15,8,15
20 OPEN 5,8,5, "*"
30 OPEN 6,8,6, "SIFRE,8,R"
40 FOR I=1 TO 2
50 INPUT#6,S,B
60 PRINT#15,"B-R:"5;0;B;B
70 FOR J=1 TO 5
80 PRINT#15,"B-P:"5;(J-1)*50+1
90 INPUT#5,A$
100 PRINT I;" "A$;" "J
110 NEXT J
120 PRINT#15,"B-F:"0;8;B
130 NEXT I
140 CLOSE 6: CLOSE 5
150 PRINT#15,"9:SIFRE"
160 CLOSE 15

```

Objasnjenje programa:

- 10 - 30 - otvaranje datoteka i kanala
- 40 - postavljanje brojača I
- 50 - učitavanje broja staze i broja bloka iz datoteke 6, SIFRE
- 60 - prenos sadržaja definisanog bloka u bafer
- 70 - postavljanje brojača J
- 80 - pozicioniranje bafer pointera za učitavanje počev od pozicija 1, 51, 101, 151, 201
- 90 - uzimanje prvih 50 karaktera, drugih 50 karaktera,... (u zavisnosti od brojača J), i smeštanje tog sadržaja u BASIC promenljivu A\$
- 100 - štampanje brojača I i sadržaja promenljive A\$ na ekranu
- 110 - kraj petlje za J
- 120 - oslobadjanje bloka
- 130 - kraj petlje za I
- 140 - zatvaranje datoteka 5 i 6
- 150 - brisanje datoteke 6
- 160 - zatvaranje komandnog kanala

NAPOMENE

U radu sa slučajnim datotekama posebno obratite pažnju na sledeće elemente:

- 1 - podatke smestati na posebne diskete odvojene od programa
- 2 - ne primenjivati komandu VALIDATE na diskete sa slučajnim datotekama. Ta komanda bi realocirala, sve blokove datoteka.
- 3 - preporučljivo je da dužina sloga ne prelazi 80 karaktera jer naredba INPUT#, može u jednom obraćanju disku da uzme više 80 karaktera

BAM I DIREKTORIJA

Onima koji žele da upoznaju u dušu svoju disk jedinicu, namenili smo ovo poglavlje. Detalno ćemo objasniti BAM, DIREKTORIJU i sve ono što omogućava rad disk jedinice. Tekst koji sledi lakše će te razumeti ako se služite tabelama 6 - 2, 6 - 3, 6 - 4, 6 - 5 i 6 - 6.

FORMAT DISKETE

U prethodnom poglavlju date su osnovne informacije o disketi. Svo vaše znanje svodi se na činjenicu da je:

"Disketa tanka plastična ploča, kružnog oblika, koja je prevučena namagnetisanim slojem ..."

Sada ćemo razotkriti tajne diskete.

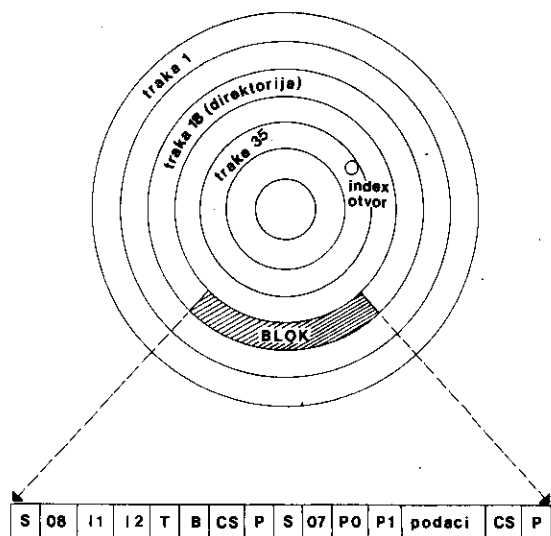
Namagnetisani sloj kojim je prevučena površina diskete, je nanesen na disketu tako, da je na celoj disketi istog sastava i gustine.

FORMATIZACIJA DISKETE je proces kojim disk jedinica deli celu površinu diskete na STAZE, koje će kasnije biti moguće lako pronaći, kada budemo smeštali podatke i programe, ili ih učitavali. Zatim, te staze deli na površine, jednake veličine, koje zovemo BLOKOVI. Disketa je podeljena na 35 staza, sa brojem blokova koji se kreće od 17 do 21. Staze su označene rednim brojevima 1 - 35 počev od spoljašnjeg ruba diskete. Najmanji broj blokova nalazi se na stazama, koje su bliže centru diskete, a broj blokova se povećava sa udaljenošću staze od centra diskete. Raspored blokova po stazama dat je sledećom tabelom.

BROJ STAZE	RASPON	UKUPNO BLOKOVA
1 - 17	0 - 20	21
18 - 24	0 - 18	19
25 - 30	0 - 17	18
31 - 35	0 - 16	17

Tab. 6 - 1 Raspored blokova po stazama

Ovaj jednostavni proces je udružen sa nešto složenijim procesom, koji će omogućiti disk jedinici, da identifikuje mesto svakog bloka na disketi, i da pomeri glavu za učitavanje/upisivanje na pravo mesto. Na sledećoj slici dat je izgled formatizovane diskete.



legenda:

S = SYNC

I1 = ID1

I2 = ID2

T = broj trake

B = broj bloka

CS = provera sume

P = preznina

P0 } pointeri za povezivanje narednog bloka
P1 }

podaci = 254 bajta

Sl. 6 - 1 Izgled formatizovane diskete

Svaki blok se sastoji od zone od 256 bajtova, koja služi za smeštanje podataka. Sem toga, u blok se upisuju i druge informacije:

- identifikacioni kod diskete (I1, I2)
- broj staze na kojoj se blok nalazi (T)
- redni broj bloka na stazi (B)
- neki standardni podaci koje disk jedinica kasnije koristi, kao proveru pravilnog odvijanja rada diska (to su tehnička rešenja koja nas ne interesuju i na ciju promenu uglavnom ne možemo ili ne znamo da utičemo; S, 08, 07, C5)

Jedna od 35 staza diskete nije namenjena za smeštanje podataka. Ta staza je staza sa rednim brojem 18, i na njoj su smešteni BAM i DIREKTORIJA. Kada se disk prvi put formatizuje, samo prva dva bloka ove staze (0-ti i 1-vi), koriste se u ovu svrhu. Ostali blokovi biće iskorisceni kada se budu upisivali programi/datoteke na disketu.

BLOK NULA

Prvi deo bloka nula od 0-tog do 143-eg bloka, namenjen je za BAM. Osnovna svrha BAM-a je da čuva informaciju, za svaki blok na disketi, da li je taj blok slobodan za smeštanje podataka, ili je već iskoriscen. Dok citate sledeci tekst gledajte tabelu G - 2.

BAJTOVI 0 i 1 sadrže informaciju o tome gde se nalazi prvi blok direktorije. Znači, ovi bajtovi su POITERI prvog bloka direktorije.

BAJT 2 sadrži informaciju koja je nebitna za disk jedinicu 1541.

BAJT 3 je nula bajt, koji je ostavljen za neku kasniju dopunu DOS-a.

BAJTOVI 4-143 predstavljaju BAM.

BAJTOVI 144-255 predstavljaju HEDER diskete.

Struktura i izgled BAM-a

BAM se sastoji od 140 bajtova. Oni su podeljeni na 35 grupa od po 4 bajta. Svaka grupa sadrži informaciju o slobodnim i iskoriscenim blokovima jedne od 35 staza na disketi i to sledećim redom:

BAM I DIREKTDRIJA

bajtovi 4, 5, 6, 7	pridruženi su	stazi 1
bajtovi 8, 9, 10, 11	pridruženi su	stazi 2
bajtovi 12, 13, 14, 15	pridruženi su	stazi 3
bajtovi 136, 137, 138, 139	pridruženi su	stazi 34
bajtovi 140, 141, 142, 143	pridruženi su	stazi 35

Prvi bajt u toj četvorci sadrži broj blokova koji su slobodni na stazi, kojoj je ova četvorka pridružena. Sledeća tri bajta sadrže informaciju o stanju blokova 0 - 7, 8 - 16 i 16 - 23, odgovarajuće staze. BITDVI ovih bajtova imaju vrednost 1 ako je blok na koji se odnose, SLOBODAN, a vrednost 0 ako je ZAUZET.

Recimo, da na traci 2 ima pet slobodnih blokova i da su slobodni blokovi 5, 9, 15, 17 i 20. Četvorka bajtova ima izgled kao na slici 2.

1. BAJT				2. BAJT				3. BAJT				4. BAJT																			
7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0

0	0	0	0	0	1	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0	0	1

2	0						7	-						0	15	-							8	23	-			16			

2	+	2	=	5																											

Sl. 6 - 2 Izgled četvorke bajtova BAM-a

Mozete uočiti da BAM može da čuva informaciju o 24 bloka na svakoj stazi, a znate da se broj blokova po stazi kreće od 17-21. Bitovi u BAM-u koji se odnose na ove nepostojeće blokove, postavljaju se u procesu formatizacije, na vrednost 0, kao da su ti blokovi već iskorišćeni. Ukoliko je ceo blok zauzet, sva četiri bajta su nula bajtovi.

Heder diskete

BAJTOVI 144 - 161 sadrže ime diskete, koje se definiše pri formatizaciji diskete. Ukoliko je ime kraće od 16 karaktera, dopunjeno je do 16 siftovanim blanko znacima.

BAJTOVI 162 - 163 sadrže identifikacioni kod diskete koji se, takodje, definiše pri formatizaciji. Taj kod se nalazi i u svakom bloku na disketi kao identifikacija pripadnosti bloka disketi.

S T A Z A 1 0			B L O K 0		
I Z O L E D B A M - a					
B A J T		N U M E R I C K I S A D R Z A J		S A D R Z A J	
0-1		1B,01		STAZA I BLOK GDE SE NALAZI PRVI BLOK DIREKTORIJE	
2		65		ASCII - KOD SLOVA A KOJI UKAZUJE NA 4040 FORMAT	
3		0		NULA BAJT OSTAVLJEN ZA BUDUCE POTREBE DOS-a	
4-143		-		MAPA SLOBODNIH I ISKORISCENIH BLOKO- VA NA STAZAMA DISKETE OD 1 DO 35 0-ISKORISCEN 1-SLOBODAN BLOK	
H E D E R D I S K E T E					
144-161				IME DISKETE DOPUNJENO SIFTOVANIM BLANKO ZNACIMA	
162-163				IDENTIFIKACIONI KOD DISKETE	
164		160		SIFTOVANI BLANKO ZNAK	
165-166		50,65		ASCII-KODOVI ZA 2A, OZNAKU DDS VERZIJE I FORMATA	
167-170		160		SIFTOVANI BLANKO ZNAK	
171-255		0		NULA BAJTOVI KOJI SE NE KORISTE	

Tab. 6 - 2 Izgled bloka 0 na stazi 10

SIFTOVANI BLANKO ZNAK

Ako vam nije jasno sta je siftovani blanko znak podsetite se malo na naredbe CHR\$ i ASC. Razmaknica, kojom se postavljaju blanko znaci ima ASCII kod 32. Ako drzite pritisnutu tipku SHIFT i razmaknicu, na ekranu ce te opet dobiti blanko znak, medjutim ASCII kod ove kombinacije je 160 i to je siftovani blanko znak.

BAJT 164 sadrži siftovani blanko znak.

BAJTOVI 165 - 166 sadrže ASCII kodove znakova 2A koji označavaju DOS verziju diska.

BAJTOVI 167 - 170 sadrže siftovane blanko znake.

BAJTOVI 171 - 255 su nula bajtovi koji se ne koriste.

Informacije iz HEDERA su značajne jer na osnovu njih znamo gde se čuvaju ime diskete, identifikacioni kod i DOS verzija. Ukoliko želite da promenite ime diskete, a da pri tom ne vršite kopiranje na drugu formatizovanu disketu sa novim imenom, možete to da učinite tako što će te direktno upisati novo ime u deo bloka nula koji sadrži ime diskete (komande koje to omogućavaju su BLOCK - WRITE i BUFFER - POINTER).

BLOK JEDAN

Izgled direktorije

Blok jedan je prvi blok direktorije diskete (pratite tab.6 - 3).

BAJTOVI 0 - 1 sadrže pointere za sledeći blok direktorije.

Preostalih 254 bajta podeljeno je na 8 delova od po 30 bajtova, odvojenih međusobno sa po 2 bajta koji se ne koriste. Svaki od tih delova je CLAN DIREKTORIJE i sadrži sve neophodne informacije o jednom programu/datoteci, koji je upisan na disketi. Iz izgleda prvog bloka direktorije, jasno je, da ukoliko postoji više od 8 programa/datoteka, direktoriji mora biti potreban još jedan blok, koji ima isti izgled kao i prvi blok. Jedina razlika je u tome, što bajtovi 0 i 1 sadrže pointere, koji ukazuju na stazu i blok gde se nalazi sledeći (treći) blok direktorije. Budući da je maksimalan broj programa/datoteka, koje se mogu upisati na disketu, 144 to znači, da je za direktoriju potrebno maksimalno 18 blokova diskete. Poslednji blok diskete ima isti izgled kao i svi ostali blokovi. Kako bajtovi 0 i 1 nemaju nikakav smisao u ovom bloku, oni su postavljeni na vrednost 0 i 255, znači ukazuju na 255-i blok 0-te staze, koji ne postoji na disketi.

S T A Z A 1 8		B L O K 1	
I Z G L E D D I R E K T O R I J E			
B A J T		S A D R Z A J	
0 - 1		STAZA I BLOK SLEDECEG BLOKA DIREKTORIJE	
2 - 31		1. CLAN DIREKTORIJE	
34 - 63		2. CLAN DIREKTORIJE	
66 - 95		3. CLAN DIREKTORIJE	
98 -127		4. CLAN DIREKTORIJE	
130-159		5. CLAN DIREKTORIJE	
162-191		6. CLAN DIREKTORIJE	
194-223		7. CLAN DIREKTORIJE	
226-255		8. CLAN DIREKTORIJE	

Tab. 8 - 3. Izgled bloka 1 na stazi 18

Izgled jednog člana direktorije

Član direktorije sadrži sve neophodne informacije o programu ili datoteci, koja je upisana na disketi (pratite tabelu 4). Budući, da to može biti program, sekvencijalna datoteka, relativna datoteka ili datoteka mašinskog programa, prva informacija koja se nalazi u bajtu 0, sadrži vrednost koja se dobija kada se na tip člana i 128 primeni logička operacija DR. U binarnom sistemu logička operacija OR interpretira se kao sabiranje binarnih brojeva. Na primer, ako se operacija OR primeni na brojeve 128 i 1 binarna interpretacija je:

$$\begin{array}{r}
 10000000 \\
 + \\
 00000001 \\
 \hline
 10000001
 \end{array}$$

 IZGLED JEDNOG CLANA DIREKTORIJE

B A J T	S A D R Z A J
0	128 + TIP CLANA - UKAZUJE NA PRAVILNO ZATVDREN ULAZ U DIREKTORIJU TIPOVI CLANA 0 - DEL (IZBRISAN CLAN) 1 - SEQ (SEKVENCIJALNA DATOTEKA) 2 - PRG (PROGRAM) 3 - USR (DATOTEKA MASIN. PROGRAMA) 4 - REL (RELATIVNA DATOTEKA)
1 - 2	STAZA I BLDK PRVOG BLOKA PODATAKA
3 - 18	IME PROGRAMA/DATOTEKE DDPUNJENO SIFTOVANIM BLANKO ZNACIMA
19 - 20	SAMO ZA RELATIVNE DATOTEKE: STAZA I BLOK PRVOG BLDKA SEKTOR SLOGOVA
21	SAMO ZA RELATIVNE DATOTEKE: BROJ KARAKTERA U SLOGU
22 - 25	NEISKORISCENI BAJTOVI
26 - 27	STAZA I BLOK PROGRAMA/DATOTEKE KOJI ZAMENJUJU POSTOJECI PROGRAM/DATOTEKU KADA SE KORISTI OPEN @ I SAVE @
28 - 29	BROJ BLOKOVA U PROGRAMU/DATOTECI: LB I HB

Tab. 6 - 4 Izgled jednog člana direktorije

TIP CLANA 0 (DEL) je značajan za pravilno izvršavanje komande SCRATCH koja briše programe/datoteke sa diskete. Kako je ovaj tip jednak 0 kada se primeni operacija OR dobija se:

$$\begin{array}{rcl}
 128 & \text{interpretirano} & 10000000 \\
 \text{OR} & \text{binarno} & + \\
 0 & & 00000000 \\
 \hline
 & & 10000000 = 128 \\
 & & (2) \quad (10)
 \end{array}$$

TIP CLANA 1 (SEQ) označava da je član direktorije sekvencijalna datoteka. Kada se ta datoteka pravilno zatvori u programu, sadržaj 0-tog bajta člana direktorije je 129. To se dobija iz:

128	interpretirano	10000000	
OR	binarno	+	
1		00000001	

		10000001	= 129
		(2)	(10)

TIP CLANA 2 (PRG) označava da je član direktorije program. Kada je program pravilno upisan SAVE komandom 0-ti bajt člana direktorije sadrži 130. To se dobija iz:

128	interpretirano	10000000	
OR	binarno	+	
2		00000010	

		10000010	= 130
		(2)	(10)

TIP CLANA 3 (USR) označava da je član direktorije datoteka mašinskog programa. Kada je ona pravilno zatvorena 0-ti bajt člana direktorije sadrži 131. To se dobija iz:

128	interpretirano	10000000	
OR	binarno	+	
3		00000011	

		10000011	= 131
		(2)	(10)

TIP CLANA 4 (REL) označava da je član direktorije relativna datoteka. Kada je ona pravilno zatvorena 0-ti bajt člana direktorije sadrži 132. To se dobija iz:

128	interpretirano	10000000	
OR	binarno	+	
4		00000100	

		10000100	= 132
		(2)	(10)

BAJTOVI 1 - 2 sadrže informaciju o tome gde se nalazi prvi blok podataka.

BAJTOVI 3 - 18 sadrže ime programa/datoteke dopunjeno siftovanim blanko znacima ako je kraće od 16 karaktera.

BAJTOVI 19 - 20 su značajni samo za relativne datoteke, sadrže informaciju gde se nalazi prvi blok sektor slogova (detaljno objašnjenje u poglavlju RELATIVNE DATOTEKE).

BAJT 21 takodje je značajan samo za relativne datoteke, sadrži informaciju o dužini sloga relativne datoteke.

BAJTOVI 26 - 27 su značajni ako se koriste naredbe OPEN @ ili SAVE @ koje omogućavaju zamenu stare datoteke/programa, novom datotekom/programom koji zamenjuje staru verziju.

BAJTOVI 28 - 29 sadrže informaciju o dužini programa/datoteke izraženu u broju potrebnih blokova za njihovo smeštanje.

Izgled bloka podataka sekvencijalne datoteke

BAJTOVI 0 - 1 sadrže pointere sledećeg bloka podataka.

BAJTOVI 2 - 255 sadrže podatke sekvencijalne datoteke medjusobno odvojene separatorima. Separator je CHR\$(13).

BLOK PODATAKA SEKVENCIJALNE DATOTEKE	
B A J T	S A D R Ž A J
0 - 1	STAZA I BLOK GDE SE NALAZI SLEDEĆI BLDK PODATAKA SEKVENCIJALNE DATOTEKE
2 -255	254 BAJTA PODATAKA ODVOJENIH SEPARATOROM CHR\$(13)

Tab. 8 - 5 Izgled bloka u kome su smešteni podaci sekvencijalne datoteke

Izgled bloka programa

BAJTOVI 0 - 1 su pointeri sledeceg bloka podataka.

BAJTOVI 2 - 255 sadrže programske informacije. Prva dva bajta, od ovih 254, u prvom bloku podataka sadrže STARTNU ADRESU PROGRAMA. Ako direktno učitamo sadržaj ova dva bajta možemo da otkrijemo startnu adresu programa (slično smo radili kada smo učitavali sadržaj kasetnog bafera). Znanje ove činjenice, omogućava nam da startujemo masinske programe čije startne adrese nisu poznate, ili da menjamo startne adrese programa.

B L O K P R O G R A M A	
B A J T	S A D R Ž A J
0 - 1	STAZA I BLOK GDE SE NALAZI SLEDECI BLOK PROGRAMA
2 -255	254 BAJTA PROGRAMA. PROGRAM SE SMESTA U CBM MEMORIJSKOM FORMATU (SA SKRACENIM KLJUCNIM RECIMA) KRAJ PROGRAMA OZNAČAVAJU TRI NULA BAJTA

Tab. 6 - 6 Izgled bloka u kome je saesthen deo programa

PUT KROZ 1B-TU STAZU

Sada ćemo dati program koji vam omogućava da vidite sadržaj blokova 1B-te staze bilo koje diskete, kao i da otkrijete i menjate, ako želite, startnu adresu svih programa sa diskete.

Program vam na početku daje pregled mogućih aktivnosti u obliku MENIJA:

- 1 - CITANJE HEDERA DISKETE
- 2 - PUT KROZ BAM
- 3 - STARTNA ADRESA

Učitajte program u memoriju vašeg kompjutera i startujte ga. U disk jedinicu ubacite disketu sa kojom želite da radite i izaberite jednu od aktivnosti.

Citanje hedera diskete

Ovaj deo programa omogućava citanje hedera diskete (tab.6 - 2) i stampanje na ekranu sledecih podataka:

- ime diskete
- identifikacioni kod
- DOS verzija

Linije 255 - 380 omogućavaju izvršavanje ove aktivnosti. Sada ćemo ukratko objasniti kako se dolazi do hedera diskete.

Heder se nalazi u bloku 0 18-te staze, pa je potrebno učitati direktno taj blok. To učitavanje se vrši linijom 270, pomoću BLOCK-READ komande (B-R). Posle izvršenja ove komande ceo blok 0 se nalazi u baferu diska kome je pridružena datoteka i kanal 9 preko koga će se učitavati sadržaj bafera. Sledeći korak je pozicioniranje u okviru bafera, čime se definiše pozicija od koje se učitava sadržaj bafera. Pozicioniranje se vrši linijom 280, pomoću komande BUFFER-POINTER (B-P). Pozicionira se na 144-ti bajt, jer počev od 144-tog bajta u bloku 0 nalazi se heder diskete. Petlja koja se nalazi od 290-te do 320-te linije učitava prvih 18 bajtova hedera, koji sadrže ime datoteke. Linija 330 omogućava učitavanje bajtova 162, 163, 164, 165, 166 u hederu koji sadrže respektivno, identifikacioni kod (162 - 163), siftovani blanko znak (164), ASCII kodove za 2A verziju DOS-a (165 - 166).

Put kroz BAM

Ovaj deo programa omogućava vam, da prodjete kroz ceo BAM diskete, i na taj način utvrdite koji su blokovi i na kojim stazama slobodni, a koji zauzeti. Na ekranu će te prvo dobiti:

```
STAZA BROJ .....1
SADRŽAJ BAJTA 4 JE .....255
IMA SLOBODNIH BLOKOVA .....21
BLOK BROJ 0 JE SLOBODAN
UKUPNO SLOBODNIH BLOKOVA      21
```

Ovaj tekst stajace stalno na ekranu. Menjace se broj staze od 1 do 35. Za svaku stazu videcete na ekranu koliko ima slobodnih blokova i koji su blokovi slobodni a koji nisu. Poruka SADRŽAJ BAJTA odnosi se na bajt u BAM-u koji ukazuje na slobodne i

zauzete blokove, odgovarajuće staze. Poslednja poruka UKUPNO SLOBODNIH BLOKOVA , daje na kraju ukupan broj slobodnih blokova na disketi. Taj broj se mora poklopiti sa brojem koji dobijate, kada izlistate direktoriju diskete.

Linije 19 - 251 omogućavaju izvršavanje ove aktivnosti. Ako želite da ubrzate prolazak kroz BAM, u liniji 230 obrišite sledeće:

```
FOR W=1 TO 500 : NEXT
```

a zatim startujte program. Objasnimo ukratko kako se prolazi kroz BAM.

BAM se nalazi u bloku 0 18-te staze, tako da prvo treba učitati ceo taj blok u bafer diska. To se vrši linijom 40, pomoću komande BLOCK-READ (B-R). Zatim se, vrši pozicioniranje na 0-ti bajt bafera, linijom 45. Učitava se sadržaj prva tri bajta u bloku 0 (0, 1, 2). Sam BAM se nalazi od 4-tog do 143-ćeg bajta. Petlja koja se nalazi od 60-te do 250-te linije, omogućava:

- učitavanje svih bajtova i to u serijama od po 4 bajta
- proveru drugog, trećeg i četvrtog bajta, bit po bit. Ukoliko bit sadrži 0 to znači da je blok zauzet, a ukoliko sadrži 1 znači da je slobodan. Prvi bajt sadrži broj slobodnih blokova na stazi

Petlja se izvršava 35 puta, što odgovara broju staza na disketi.

Startna adresa

Ovaj deo programa otkriva vam startnu adresu bilo kog programa sa diskete i daje mogućnost promene te startne adrese.

Linije 385 - 730 omogućavaju izvršavanje ove aktivnosti. Objasnimo ukratko na koji način se nalazi startna adresa.

Na početku programa unosite ime programa, čiju startnu adresu tražite, u promenljivu FL\$. Podaci o programu nalaze se u članu direktorije diskete, koji je formiran za taj program. Stoga, prvo treba pronaći taj član. Potrebno je prvo učitati prvi blok direktorije diskete u bafer. To se radi u liniji 460. Zatim se vrši pozicioniranje na prvi bajt bafera, i učitava taj prvi bajt. Sadržaj tog bajta je pointer za drugi blok direktorije, linija 470.

Petlja od 490-te do 550-te linije omogućava učitavanje po 32 bajta prvog bloka, odnosno učitavanje člana direktorije:

- prvo se učitava ime programa (3 - 18 bajt u članu), linije 500, 510
- taj sadržaj se upoređuje sa imenom unetim u FL\$, linija 515
- ukoliko se ti sadržaji poklope, učitava se prvi i drugi bajt člana, pointer za prvi blok programa. Ukoliko se ne poklope, učitava se sledeći član direktorije, sve dok se ne pronadje član koji je potreban
- ako program sa zadatim imenom ne postoji na disketi, linija 570 formira odgovarajuću poruku

Kada je član direktorije za zadati program pronadjen, i iz njega pročitana pointer za prvi blok programa, moguće je učitati taj blok, linija 580. Kada se taj blok nalazi u baferu, pozicionira se bafer pointer na drugu i treću poziciju, linije 590 i 600. Ovo je veoma bitno, jer se u drugom i trećem bajtu prvog bloka programa, nalazi startna adresa programa, i to njen LB i HB. Preracunavanje startne adrese u dekadnu vrednost vrši se u liniji 610.

Na sličan način se menja i startna adresa. Vi zadajete startnu adresu u dekadnom sistemu, linija 680 preračunava tu vrednost u LB i HB. Zatim se u drugi i treći bajt prvog bloka programa, upisuju te nove vrednosti. Time je promenjena startna adresa programa, odnosno, program relociran.

KOMANDE SAVE, LOAD I SCRATCH

Sada možemo u potpunosti da sagledamo, šta se dešava kada damo jednu od ovih komandi disku.

Ako želimo da sačuvamo na disketi, program pod imenom PRIMER, dajemo disku komandu:

SAVE "PRIMER",B

To će izazvati sledeće aktivnosti:

- 1 - centralna jedinica daje instrukciju disk jedinici da otvori programoteku pod tim imenom
- 2 - disk jedinica proverava direktoriju diskete da vidi da li

- već postoji program pod tim imenom
- 3 - kada utvrdi da ne postoji program sa takvim imenom, disk jedinica upisuje ime programa PRIMER u direktoriju diskete, sa početnim blokom i stazom, 0 i 255 koji u stvari ne postoji (formira član direktorija za program PRIMER)
 - 4 - koristeći BAM, koji se stalno nalazi u memoriji disk jedinice, ona traži stazu, najbližu IB-toj stazi, bilo u pravcu centra diskete, bilo u pravcu ruba diskete, koja ima slobodan blok (BLOK 1). Zatim označava u BAM-u da je taj blok iskorišćen
 - 5 - kada je BLOK 1 pronadjen, disk jedinica upisuje njegovu lokaciju (staza, blok) u bajtove 1 i 2 člana direktorije formiranog u koraku 3. Zatim, prihvata 254 bajta programa iz memorije C-64 i smesta ih u bafer
 - 6 - traži sledeći slobodan blok, najbliži bloku pronadjenom u koraku 4 (BLOK 2)
 - 7 - pointeri BLOKA 2 (staza, blok) upisuju se u prva dva bajta bafera formiranog u koraku 5
 - 8 - ceo sadržaj bafera se upisuje u BLOK 1, pronadjen u koraku 4
 - 9 - ovaj postupak se ponavlja počev od 5-og koraka, sve dok C-64 ne da komandu disk jedinici, da zatvori programoteku (kada je ceo sadržaj programa prebačen iz C-64 na disketu)
 - 10 - za poslednji pun bafer podataka programa, prva dva bajta se postavljaju na vrednost 0 i 255, da ukazuju na nepostojeći blok
 - 11 - poslednji postupak je upis broja blokova potrebnih za smestanje programa, u odgovarajuće bajtove člana direktorije

Kada disku damo komandu:

```
LOAD "PRIMER", 0
```

to će izazvati odvijanje sledećih aktivnosti:

- 1 - disk jedinica proverava da li u direktoriji postoji ime programa jednako unetom imenu
- 2 - kada utvrdi da takav program postoji, na osnovu bajtova 1 i 2 člana direktorije ovog programa, pronalazi lokaciju prvog bloka programa
- 3 - prva dva bajta ovog bloka ukazuju na sledeći blok programa. Ostalih 254 bajta prenose se sa diskete u centralnu jedinicu (bajtovi programskih informacija)
- 4 - pronalazi se sledeći blok programa i postupak ponavlja sve dok se ne dodje do poslednjeg bloka programa (on se prepoznaje po tome, što njegova prva dva bajta ukazuju na

nepostojeci blok, 255-ti blok 0-te staze)

Na kraju da objasnimo šta se dešava kada se disku da komanda za brisanje programa ili datoteke, npr.:

OPEN 15,B,15 : PRINT#15, "S:PRIMER" : CLOSE 15

Ona će usloviti izvršavanje sledećih aktivnosti:

- 1 - disk jedinica proverava da li postoji program/datoteka sa takvim imenom, u direktoriji diskete, na taj način što upoređuje uneto ime sa sadržajem bajtova 3 - 18, u svim članovima direktorija
- 2 - kada utvrdi da takav program postoji u članu direktorije, koji se odnosi na taj program, postavlja vrednost 128 u 0-ti bajt. To odgovara tipu DEL, za brisanje programa/datoteke, čime se ovaj član direktorije oslobadja za novi program/datoteku
- 3 - na osnovu sadržaja bajtova 1 i 2 u istom članu, pronalazi lokaciju (staza, blok) prvog bloka programa, zatim drugog (iz prvog bloka programa) itd. do poslednjeg
- 4 - kada je informacija o lokaciji svih blokova potrebnih za ovaj program potpuna, disk jedinica menja BAM. Sve blokove koji su korišćeni za smeštanje ovog programa, u odgovarajućim bitovima BAM-a oznaci sa 1 što znači, da su slobodni za novi upis

Bitno je da uočite, da SCRATCH komanda, ne vrši pravo brisanje sadržaja blokova koje zauzima program, koji želimo da obrišemo, već samo oslobadja te blokove za upis novog sadržaja preko starog.

```

1 C$=CHR$(0):PRINT "{RED}":POKE 53280,15
2 POKE 53281,15
5 PRINT "{CLR}{RED}{CUR DN}{CUR DN} {RVS ON} M E N I {RVS OFF}"
6 PRINT "{CUR DN}{CUR DN}CITANJE HEDERA DISKETE....1"
7 PRINT "{CUR DN}{CUR DN}PUT KRZD B A M .....2"
8 PRINT "{CUR DN}{CUR DN}STARTNA ADRESA PROGRAMA...3"
9 PRINT "{CUR DN}{CUR DN}K R A J RADA.....4"
10 PRINT "{CUR DN}{CUR DN}{CUR DN}{RVS ON} IZABERITE BROJ {RVS OFF}";:
INPUT N
11 IF N<1 OR N>4 THEN 5
12 ON N GOSUB 255,19,400,15
13 GOTO 5
15 END
19 REM ** PUT KRZD SREDISTE BAMA ***
20 PRINT "{RED}{CLR}{CUR DN}{CUR DN} {RVS ON} PUT KROZ SREDISTE
BAMA-A {RVS OFF}"
25 GOSUB 5000
30 OPEN 1,8,15:OPEN 9,8,9,"#":GOSUB 4000
40 PRINT#1,"B-R"9;0;18;0 : GOSUB 4000
45 PRINT#1,"B-P"9;0: GOSUB 4000
50 GET#9,S2$,8L$,A$:BR=1
60 FOR I=4 TO 143 STEP 4
65 IF BR=18 THEN 250
70 PRINT#1,"B-P"9;I
80 GET#9,B1$,B$(1),B$(2),B$(3)
81 IF B1$="" THEN B1$=C$
82 IF B$(1)="" THEN B$(1)=C$
83 IF B$(2)="" THEN B$(2)=C$
84 IF B$(3)="" THEN B$(3)=C$
90 PRINT "{BLK}{CLR}{CUR DN}STAZA BROJ.....{RED}";BR
110 PRINT "{CUR DN}{CUR DN}{BLK}IMA SLDOBNIH BLOKOVA....{RED}";
115 PRINT ASC(B1$):X=X+ASC(B1$)
120 IF BR<=17 THEN T=20:
130 IF 17<BR AND BR<=24 THEN T=18:
140 IF 24<BR AND BR<=30 THEN T=17
150 IF 30<BR AND BR<=35 THEN T=16
155 V=0:Q=0
160 FOR Z=1 TO 3:PRINT
165 PRINT "{BLK}SADRZAJ BAJT-A{RED}";I+Z;
166 PRINTTAB(19)"{BLK}JE..... {RED}";"{CUR L}{CUR L}{CUR L}{CUR L}"
ASC(B$(Z))
170 FOR E=0 TO 7
180 BIT=ASC(B$(Z))AND2^E
190 IF V>T THEN 240
200 IF BIT<>0 THEN SEK$="SLOBODAN";Y=Y+1;Q=Q+1;GOTO220
210 SEK$="ZAUZET ";
220 PRINT "{CUR DN}{BLK}BLK BROJ..{RED}";V;"{BLK} JE {RED}";TAB(20)SE
K$
226 PRINT:PRINT:PRINT
228 PRINT "{BLK}UKUPNO SLOBODNIH BLOKOVA...{RED}";Y
230 V=V+1:PRINT "{CUR UP}{CUR UP}{CUR UP}{CUR UP}{CUR UP}{CUR UP}{CUR
UP}";FORW=1TO500:NEXT
240 NEXT E:PRINT "{CUR UP}{CUR UP}{CUR UP}";NEXT Z
241 IF Q<>ASC(B1$) THEN Y=Y+(ASC(B1$)-Q)
250 BR=BR+1:NEXT I:GOSUB 5000
251 CLOSE 1:CLOSE9
255 REM ** CITANJE HEDERA DIR{RVS OFF}EKTDRIFE **
260 OPEN 1,8,15:OPEN 9,8,9,"#"
270 PRINT#1,"B-R"9;0;18;0
280 PRINT#1,"B-P"9,144:GOSUB 4000
290 FOR I=1 TO 18
300 GET#9,A$:IF A$=""

```

* BAMIDIREKTORIJA

```

300 GET#9,A$:IF A$="" THEN A$=C$
310 IME$=IME$+A$
320 NEXT I
330 GET#9,I1$,I2$,SP$,A1$,A2$
340 PRINT"{RED}{CLR}{RVS ON} C I T A N J E   H E D E R A {RVS OFF}"
350 PRINT"{RED}{CUR DN}{CUR DN}{CUR DN}NAZIV DISKETE.....{BLK}";IME$
360 PRINT"{RED}{CUR DN}{CUR DN}IDENTIFIKACIONI KOD.....{BLK}";I1$+I2$
370 PRINT"{RED}{CUR DN}{CUR DN}DOS VERZIJA.....{BLK}";A1$+A2$
380 GOSUB 5000:CLOSE9:CLOSE1:RETURN
385 REM ** PROMENA STARTNE ADRESE **
400 PRINT "{CLR}{RVS ON} PROMENA STARTNE ADRESE PROGRAMA {RVS OFF}"
410 CR$=CHR$(13):C0$=CHR$(0):C1$=CHR$(1)
420 INPUT{CUR DN} IME PROGRAMA ";FL$
430 D=LEN(FL$):IF D>16 THEN FL$=LEFT(FL$,16):D=16
440 PRINT{CUR DN} TRAZIM PRDGRAM "FL$
450 DPEN1,8,15:DPEN9,8,9,"*";GOSUB4000:SZ=1:OK%=0
460 PRINT#1,"B-R"9;0;18;SZ;GOSUB4000
470 PRINT#1,"B-P"9;1;GET#9,NS$:IFNS$="" THENNS$=C0$
480 PRINT#1,"U1"9;0;18;SZ;GOSUB4000
490 FORI=2TO226STEP32:PRINT#1,"B-P"9;I
495 GET#9,TY$:IFTY$(<)>CHR$(I30)GOTO550
500 PRINT#1,"B-P"9;I+19:PRINT#9,CR$;
510 PRINT#1,"B-P"9;I+3:INPUT#9,NM$
515 IFLEFT$(NM$,D)<>FL$ THEN GOTO550
520 PRINT#1,"B-P"9;I+1:GET#9,LT$:IFLT$="" THENLT$=C0$
530 PRINT#1,"B-P"9;I+2:GET#9,LS$:IFLS$="" THENLS$=C0$
540 OK%=1:I=226
550 NEXTI:IFOK% THEN580
560 SZ=ASC(NS$):IF0<SZANDSZ<20GOTO460
570 PRINT"{CLR}{CUR DN}PROGRAM {BLK}" FL$"{RED} NE POSTOJI NA DISKETI
"
575 CLOSE9:CLOSE1:GOSUB 5000:RETURN
580 TX=ASC(LT$):SZ=ASC(LS$):PRINT#1,"U1"9;0;TX;SZ
585 GOSUB4000
590 PRINT#1,"B-P"9;2:GET#9,LO$:IFLO$="" THENLO$=C0$
600 PRINT#1,"B-P"9;3:GET#9,HI$:IFHI$="" THENHI$=C0$
610 LO%=ASC(LO$):HI%=ASC(HI$):AD=LO%+256*HI%
620 PRINT{CUR DN} TEKUCA STARTNA ADRESA JE ...."AD
630 PRINT{CUR DN} HOCETE PROMENU ({RVS ON}D{RVS OFF})-(RVS ON)N{RVS O
FF}) ?
640 GETG$:IFG$(<)>"D"ANDG$(<)>"N"GOTO640
650 IFG$="N"GOTO710
660 INPUT{CUR DN} NOVA STARTNA ADRESA {CUR L}{CUR L} ";AD$
670 IFASC(AD$)=160THENPRINT{CUR UP}{CUR UP}{CUR UP}";80TO660
680 AD=VAL(AD$):HI%=AD/256:LO%=AD-256*HI%
690 PRINT#1,"B-P"9;2:PRINT#9,CHR$(LO%)CHR$(HI%)
700 PRINT#1,"U2"9;0;TX;SZ;GOSUB4000:GOTO580
710 PRINT{CLR}{CUR DN}PROGRAM {BLK}"FL$"{RED} JE RELOCIRAN"
720 PRINT{CUR DN}NA STARTNU ADRESU {BLK}";AD;"{RED}";GOSUB5000
730 CLOSE 9:CLOSE 1:RETURN
4000 INPUT#1,G,PG$,S,B:IF G=0 THEN RETURN
4010 PRINT{CLR}{CUR DN}{CUR DN}{CUR DN}{RVS ON} I/O G R E S K A {RVS OFF}"
4020 PRINT{CUR DN}{CUR DN}BROJ GRESKE.....";8
4030 PRINT{CUR DN}{CUR DN}NAZIV GRESKE.....";PG$
4040 PRINT{CUR DN}{CUR DN}STAZA.....";S
4050 PRINT{CUR DN}{CUR DN}BLDK.....";B
4060 CLOSE9:CLOSE1:END
5000 PRINT{RED}{CUR DN}{CUR DN}{CUR DN}{CUR DN}{CUR DN}{CUR DN}{CUR
DN}
{RVS ON} <RETURN> {RVS OFF} ZA DALJE"
5010 GET Y$:IF Y$="" THEN 5010
5020 RETURN

```

RELATIVNE DATOTEKE

Relativne datoteke predstavljaju drugi tip datoteka, kod kojih je moguć slučajni pristup podacima. Slučajne datoteke, sa kojima ste se upoznali u prethodnom poglavlju, najčešće se koriste u masinskom programiranju. Iako je moguće, koristeći slučajne datoteke, raditi sa većim brojem datoteka (kreiranje, pretraživanje, ažuriranje datoteka), ti postupci su veoma komplikovani. Pogotovo je komplikovano, uzimati pojedinačna polja iz sloga i posebno ih obradljivati, jer za to stalno morate koristiti funkcije LEFT\$, MID\$, RIGHT\$, VAL, da biste došli do sadržaja jednog odredjenog polja u slogu. Upravo zbog ovih komplikacija razradjen je drugi tip datoteka sa slučajnim pristupom, koji je isključivo namenjen za rad sa datotekama podataka i omogućava jednostavno kreiranje, pretraživanje i ažuriranje.

Relativne datoteke podržavaju osnovni teorijski princip kreiranja baza podataka, a to je struktuiranje podataka počev od polja, preko sloga, datoteke do povezanih grupa datoteka - baza podataka. Drugim recima, relativna datoteka se kreira tako da se opsuje struktura sloga i pojedinačna polja u slogu.

RAD SA RELATIVNIM DATOTEKAMA

Rad sa relativnim datotekama omogućavaju odredjene komande DOS-a. Za svaku relativnu datoteku DOS rezervise i kreira blokove, koji se nazivaju SEKTORI SLOGOVA, a predstavljaju listu pointera za lokaciju početka slogova datoteke na disketi. Ta lokacija je definisana brojem bloka i brojem staze. Svaki sektor slogova može da sadrži informacije o lokaciji početka 120 slogova datoteke. Maksimalan broj sektor slogova koji se pridružuje jednoj datoteci je 6. To znači da je maksimalan broj slogova relativne datoteke 720, pri tom slog može da ima maksimalno 254 karaktera. Drugo ograničenje za maksimalan broj slogova u datoteci, je maksimalan broj slobodnih blokova na disketi (683 za klasične diskete), tako da datoteka može da ima onoliko slogova koliko joj to dozvoljava slobodan prostor na disketi.

Sektore slogova možete smatrati direktorijom vaše relativne datoteke, jer oni čuvaju informacije o tome gde se nalaze svi slogovi datoteke i omogućavaju slučajni pristup tim slogovima. Rad sa relativnim datotekama oslobadja vas zamornog posla, koji ste imali u radu sa slučajnim datotekama, a to je kreiranje posebne sekvencijalne datoteke, u kojoj se nalaze informacije o lokaciji blokova i slogova slučajne datoteke. Sektori slogova, upravo, imaju funkciju te sekvencijalne datoteke, s tom razlikom što njih

RELATIVNE DATOTEKE

automatski kreira DOS. Da biste našli jedan odredjeni slog slucajne datoteke, morali ste programski obezbediti pretraživanje sekvencijalne datoteke. Relativna datoteka vas oslobadja i tog posla. Dovoljno je definisati koji se slog traži, a zatim ce automatski biti pretraženi sektori slogova, dobijene informacije o broju bloka i staze gde se slog nalazi i uzet sadržaj sloga sa diskete.

B L O K S E K T O R S L O G O V A	
B A J T	S A D R Ž A J
0,1	STAZA I BLOK GDE SE NALAZI SLEDECI BLOK SEKTORA SLOGOVA
2	BROJ SEKTORA SLOGOVA (BROJ OD 0-5)
3	DUZINA SLOGA
4,5	STAZA I BLOK GDE SE NALAZI PRVI SEKTOR SLOGOVA (BROJ 0)
6,7	STAZA I BLOK GDE SE NALAZI DRUGI SEKTOR SLOGOVA (BROJ 1)
8,9	STAZA I BLOK GDE SE NALAZI TRECI SEKTOR SLOGOVA (BROJ 2)
10,11	STAZA I BLOK GDE SE NALAZI CETVRTI SEKTOR SLOGOVA (BROJ 3)
12,13	STAZA I BLOK GDE SE NALAZI PETI SEKTOR SLOGOVA (BROJ 4)
14,15	STAZA I BLOK GDE SE NALAZI SESTI SEKTOR SLOGOVA (BROJ 5)
16-255	STAZA I BLOK GDE SE NALAZI 120 BLOKOVA PODATAKA

Tab. H - 1 Izgled bloka u koji je smesten jedan sektor slogova

 B L O K P O D A T A K A

B A J T

S A D R Z A J

0,1

STAZA I BLOK GDE SE NALAZI SLEDECI BLOK PODATAKA

2-255

254 BAJTOVA U KOJIMA SE NALAZE PODACI
 PRAZAN SLDG SADRZI FF (SVE BINARNE JEDINICE) U
 PRVOM BAJTU, DOK SU SVI OSTALI BAJTOVI DO KRAJA
 POPUNJENI SA 00 (BINARNA 0)

Tab. H - 2 Izgled bloka u kome su smešteni podaci

PRAVILA ZA RAD

1. Relativna datoteka se sastoji od slogova, koji se sastoje od polja fiksne duzine.
2. Duzina polja je broj karaktera, koji se može smeštitu u to polje, uvećan za jedan. Broj karaktera se uvećava za jedan, jer kad unosite sadržaj polja, posle svakog unošenja pritiskate tipku RETURN, i time se iza karaktera koji ste uneli, automatski postavlja CHR\$(13).
3. Ukoliko je podatak koji unosite kraći od unapred definisane duzine polja, ostatak polja će se automatski popuniti prazninama (CHR\$(32)). Na primer, ako u slogu definisete polje IME duzine 8 + 1 (CHR\$(13)), a unosite ime Petar izgled tog polja posle popunjavanja je sledeci:

 P E T A R

 1 2 3 4 5 6 7 8 CHR\$(13)

4. Podatak koji se unosi ne sme biti duži od duzine polja umanjene za jedan. Na primer, ako je ime koje unosite ALEK-SANDAR, biće prekoračna duzina od 8 karaktera polja IME. Da bi se izbegla ova greska, posle svakog unošenja podataka, treba izvršiti otsecanje suvišnih karaktera koristeći naredbu LEFT\$. Za prethodni primer ta naredba bi imala oblik:

$$IME\$ = LEFT\$(IME\$,8)$$

5. Preporučljivo je posle svakog polja ostaviti jedan blanko znak koji razdvaja dva susedna polja.
6. U toku rada sa relativnom datotekom moraju biti otvorena dva kanala, jedan za prenos podataka i drugi za prenos komandi odnosno gresaka izmedju diska i centralne jedinice (kanal 15). Drugi kanal može imati broj od 2 - 14, medjutim kako su kanali 3 i 4 rezervisani za druge svrhe (ekran i štampač) preporučljivo je izabrati broj od 5 - 14. Iz ovog sledi da ne mogu biti istovremeno otvorene dve relativne datoteke, jer je svakoj potreban kanal 15, a postoji samo jedan kanal 15.
7. Kada radite sa relativnim datotekama nemojte koristiti SCRATCH komandu, jer ce ona izbrisati sve postojeće relativne datoteke.

KREIRANJE DATOTEKE

Relativna datoteka se kreira specijalnim oblikom naredbe OPEN:

OPEN A,B,C,"IME,L"+CHR\$(DUZINA SLOGA)

A - broj datoteke

B - broj periferijske jedinice

C - broj kanala

IME - ime datoteke

Na primer, naredbom **OPEN 5,8,5,"PODACI,L"+CHR\$(50)** bice kreirana relativna datoteka sa slogom duzine 50 karaktera. Duzina sloga se racuna po formuli:

DUZINA SLOGA = SUMA DUZINA POLJA + 2 * (BROJ POLJA)

jer se iza svakog polja nalazi po jedan znak za CHR\$(13) i po jedan blanko znak. Broj koji predstavlja duzinu sloga, ne sme precu 255.

Posle kreiranja datoteke vrši se upis sloga u kreiranu datoteku, pri tom treba imati na umu da se slog sastoji iz više polja, koja se pojedinačno upisuju u datoteku. Da bi taj postupak bio moguć, pre upisa se vrši pozicioniranje odgovarajucih pointera, koji ukazuju na pocetak odredjenog polja. Informacije neophodne za izvršavanja ove funkcije prenose se kroz komandni kanal 15. Naredba koja omogućava pozicioniranje pointera ima opšti oblik:

```
PRINT#K,"P"CHR$(C)CHR$(L)CHR$(H)CHR$(M)
```

K - broj komandnog kanala (15)

C - broj kanala za relativu datoteku

M - pozicija u slogu, od koje treba početi upis podataka u datoteku

L,H - definiše redni broj sloga u datoteci

Redni broj sloga u datoteci definiše se sadržajem 2 bajta. Jedan bajt može da sadrži 256 različitih vrednosti, a relativna datoteka može da ima preko 700 slogova. Stoga je neophodno koristiti 2 bajta da bi se mogli definisati brojevi slogova veći od 256. Taj način predstavljanja brojeva, sigurno vam je već poznat, jer je uobičajen u radu sa mikro kompjuterima. Redni broj sloga (dekadni broj), izražava se na sledeći način:

$$RBS = H * 256 + L$$

RBS - redni broj sloga (dekadni broj)

H - sadržaj bajta više značajnosti (HIGH BYTE)

L - sadržaj bajta niže značajnosti (LOW BYTE)

Na primer, ako je redni broj sloga manji od 255, recimo 20 tada:

$$20 = 0 * 256 + 20 \quad \text{znači} \quad H = 0 \quad L = 20$$

i naredba za pozicioniranje ima oblik:

```
PRINT#15,"P"CHR$(5)CHR$(20)CHR$(0)CHR$(1)
```

Ovom naredbom se pozicionira pointer za upis 20-tog sloga datoteke 5, počev od prvog karaktera u slogu.

Ukoliko je redni broj sloga veći od 255, recimo 520, H i L se računaju na sledeći način:

$$H = \text{INT}(520/256)$$

Oznaka INT znači ceo deo broja u zagradi. To znači, kada se izvrši deljenje 520 sa 256, u H se smešta broj koji označava koliko se puta 256 sadrži u 520. U ovom slučaju H = 2. L se dobija na sledeći način:

$$L = 520 - 2 * 256 = 520 - 512 = 8$$

Naredba za pozicioniranje ima oblik:

```
PRINT#15,"P"CHR$(5)CHR$(8)CHR$(2)CHR$(1)
```

Ovom naredbom se pozicionira pointer za upis 520-tog sloga, relativne datoteke 5 počev od prvog karaktera u tom slogu.

NAPOMENA

Kada prvi put otvorite datoteku i time je kreirate, da bi ste izvršili upis slogova, dobićete kroz kanal greške sledeću grešku:

50 RECORD NOT PRESENT

Poruka o grešci se pojavljuje, zbog toga što otvarate novu datoteku u kojoj još ne postoji ni jedan slog, već vi nameravate da upisujete slogove. Stoga, tehnički gledano greška ne postoji jer zaista ne postoji ni jedan slog. Da biste sprečili da vam ova greška prekida izvršavanje programa, potrebo je da se u programu obezbedi deo koji proverava sadržaj kanala greške. Ukoliko je broj greške jednak 50, zanemariti tu grešku, i nastaviti izvršavanje programa. Ukoliko postoji neka druga greška, sa brojem razlicitim od 50, stampati poruku o grešci i prekinuti rad programa. Uobicajeno je, da se svi ovi postupci grupišu u jedan potprogram.

Primer za kreiranje relativne datoteke

Kreirajmo relativnu datoteku koja ima slog oblika:

```
*****
* IME * PREZIME * ULICA I BROJ * GRAD * BROJ TELEFONA *
*****
   10      20      20      15      12
```

Duzina pojedinačnih polja napisana je ispod imena polja. Duzina sloga računa se po formuli:

$$DS = 10 + 20 + 20 + 15 + 12 + 2 * 5 = 87$$

Program koji sledi vrši kreiranje pomenute datoteke:

```
10 OPEN 15,0,15
20 OPEN 5,0,5,"PODACI,L,"+CHR$(87)
30 00SUB500
35 INPUT{(CLR){CUR DN}{CUR DN) UNESITE BROJ SLOBOVA U DATOTECI ";N
40 RBS=1;00SUB 600
50 PRINT{(CLR){CUR DN){CUR DN) UNOSENJE SADRZAJA POLJA"
55 BR=1
60 INPUT{(CUR DN){CUR DN) IME";IME$
70 IME$=LEFT$(IME$,10)
80 INPUT{(CUR DN){CUR DN) PREZIME";PREZ$
90 PREZ$=LEFT$(PREZ$,20)
100 INPUT{(CUR DN){CUR DN) ULICA I BROJ";UL$
110 UL$=LEFT$(UL$,20)
120 INPUT{(CUR DN){CUR DN) GRAD";GRAD$
```

```

130 BRAD$=LEFT$(BRAD$,15)
140 INPUT"(CUR DN)(CUR DN) BROJ TELEFONA";TEL$
150 TEL$=LEFT$(TEL$,12)
160 GOSUB 700: GOSUB 500
180 PRINT#5,IME$: GOSUB 500
190 BR=BR+12
200 GOSUB 700: GOSUB 500
210 PRINT#5,PREZ$: GOSUB 500
220 BR=BR+22
230 GOSUB 700: GOSUB 500
240 PRINT#5,UL$: GOSUB 500
250 BR=BR+22
260 GOSUB 700: GOSUB 500
270 PRINT#5,GRAD$: GOSUB 500
280 BR=BR+17
290 GOSUB 700: GOSUB 500
300 PRINT#5,TEL$: GOSUB 500
305 IF RBS=N THEN GOTO 340
310 RBS=RBS+1
320 GOSUB 600
330 GOTO 50
340 CLOSE 5: CLOSE 15
350 PRINT"(CLR)(CUR DN)(CUR DN)   KRAJ RADA PROGRAMA  "
360 END
500 INPUT#15,G,PG$,S,B
510 IF G<20 THEN RETURN
520 IF G<>50 THEN GOTO 540
530 RETURN
540 PRINT G,PG$,S,B:STOP:RETURN
600 L=RBS
610 IF L>255 THEN H=INT(L/256)
620 L=L-256*H
630 RETURN
700 PRINT#15,"P"CHR$(5)CHR$(L)CHR$(H)CHR$(BR)
710 RETURN

```

Listing H - 1 Kreiranje relativne datoteke
* KREIRANJE

Objasnjenje programa:

- 10 - otvaranje komandnog kanala
- 20 - otvaranje relativne datoteke 6, kojim se definiše dužina sloga 87 karaktera, ime datoteke je PODACI
- 30 - poziva potprogram za proveru greške u radu sa diskom
- 35 - unosenje broja slogova u datoteci, taj broj služi kao kriterijum za kraj rada programa
- 40 - RBS je oznaka za promenljivu koja sadrži redni broj sloga, a BR za promenljivu koja označava od koje pozicije u slogu počinje upis sadržaja polja. Početna vrednost za RBS i BR je 1, što znači da se vrši upis prvog sloga, počev od prve pozicije u slogu. Zatim se poziva potprogram koji pretvara

- redni broj sloga RBS u oblik $L + 256 * h$ i izracunava H i L
- 50 - stampa na ekranu poruku "UNOŠENJE SADRŽAJA POLJA"
 - 60 - unošenje sadržaja polja IME, u promenljivu IME\$
 - 70 - kontroliše dužinu polja IME\$, ako je sadržaj promenljive IME\$ duži od 10, vrši se otsecanje suvišnih karaktera
 - 80 - unošenje sadržaja polja PREZIME u promenljivu PREZ\$
 - 90 - kontroliše dužinu polja PREZ\$
 - 100 - unošenje sadržaja polja ULICA I BROJ u promenljivu UL\$
 - 110 - kontroliše dužinu polja UL\$
 - 120 - unošenje sadržaja polja GRAD u promenljivu GRAD\$
 - 130 - kontroliše dužinu polja GRAD\$
 - 140 - unošenje sadržaja polja BRDJI TELEFONA u promenljivu TEL\$
 - 150 - kontroliše dužinu polja TEL\$
 - 160 - poziv potprograma za pozicioniranje pointera, koji omogućava upis polja IME\$ u prvom slogu (RBS = 1, H = 0, L = 1) počev od prvog karaktera u slogu (BR = 1);
poziv potprograma za proveru greške u radu sa diskom
 - 180 - prenos sadržaja polja IME\$, kroz kanal 5 u bafer
 - 190 - BR se povećava za 12, tako da je pozicija za upis sadržaja sledećeg polja PRES\$, 23-ći karakter u slogu
 - 200 - 300 - ove programske linije vrše postupke analogne postupcima u linijama 160 - 190, i omogućavaju pozicioniranje i prenos ostalih polja u bafer. Kada se bafer napuni sadržaj bafera prenosi se na disketu u relativnu datoteku PODACI. Taj postupak se izvodi automatski bez naše kontrole
 - 305 - predstavlja kriterijum za kraj programa. Kako se ukupan broj slogova unosi na početku programa, u promenljivu N, a redni broj sloga RBS se povećava za 1 kada se završi upis jednog sloga, program se završava kada je RBS = N. U liniji 340 dato je šta treba uraditi kada je kraj upisa
 - 310 - posle završenog upisa jednog sloga, redni broj sloga se povećava za 1
 - 320 - poziva potprogram za izracunavanje L i H za novi broj sloga
 - 330 - povratak na liniju kojom počinje deo programa za unošenje sadržaja sloga i prenos u bafer
 - 340 - kada je završen upis svih slogova završava se rad programa, zatvaraju kanali otvoreni na početku programa
 - 350 - stampa na ekran poruku "KRAJ RADA PROGRAMA"
 - 360 - kraj programa

Ovim je opisan postupak unošenja sadržaja slogova i upis u relativnu datoteku. Slede potprogrami:

500 - 540

Potprogram koji služi za ispitivanje kanala greške. Linija 500 omogućava učitavanje, kroz kanal greške, sledećih informacija:

- G - broj greške
- PG\$ - poruka o grešci
- S,B - staza i blok gde se nalazi greška

Ukoliko je $G < 20$ na postoji nikakva greška i vrši se povratak u program (linija 510).

Ukoliko je G različito od 50 postoji grška i potrebno je odstampa-
pati poruku o grešci i zaustaviti rad programa (linije 520 i 540)

Ukoliko je $G = 50$ tu grešku treba ignorisati, jer je to greška
RECORD NOT PRESENT, koja zapravo nije greška, i vratiti se u pro-
gram.

600 - 630

Potprogram koji pretvara redni broj sloga RBS u oblik $L+H*256$, i
izračunava H i L .

700 - 710

Potprogram koji pozicionira poitere u baferu za upis sadržaja
pojedinih polja, od tačno definisane pozicije, koju sadrži BR.

DODAVANJE NOVIH SLOGOVA DATOTECI

Ako ste naučili da vršite kreiranje i upis slogova relativne
datoteke, neće biti nikakvih problema da naučite kako da dodate
slogove već postojećoj datoteci. Objasnimo to na primeru.

U primeru za kreiranje datoteke, kreirali smo datoteku PODACI. Na
početku programa unosi se broj slogova datoteke, koje ćemo
upisivati, kao broj N . Pretpostavimo da je taj broj u prethodnom
primeru bio 30. Sada želimo da dodamo još 20 novih slogova. Redni
broj poslednjeg upisanog sloga je 30. Da bi dodali sledeći slog,
redni broj poslednjeg sloga treba povećati za 1. Znači, na
početku programa RBS mora biti jednako 31. Postupak za upis sloga
je potpuno isti kao u prethodnom primeru. Kada je upisan 31-vi
slog, postupak se ponavlja za 32-gi, 33-ci itd. sve do 50-og
sloga. Broj N , koji se unosi na početku programa u ovom slučaju
je 50. Kada se RBS izjednači sa N , odnosno 50, izvršeno je
dodavanje slogova datoteke.

NAPOMENA

Ako vršite upis slogova u već postojeću datoteku, vodite računa
da redni broj sloga bude uvek veći od rednog broja poslednjeg
upisanog sloga. Ako to nije slučaj, novi sadržaj sloga izbrisace
stari sadržaj već postojećeg sloga.

Program za dodavanje novih slogova

```

10 OPEN 15,8,15
20 OPEN 5,8,5,"PODACI"
30 GOSUB 500
35 INPUT"(CLR){CUR DN}{CUR DN} UNESITE BROJ SLOGOVA U DATOTECI ";N
40 RBS=30:GOSUB 600
50 PRINT"(CLR){CUR DN}{CUR DN} UNOSENJE SADRZAJA POLJA"
55 BR=1
60 INPUT"(CUR DN){CUR DN} IME";IME$
70 IME$=LEFT$(IME$,10)
80 INPUT"(CUR DN){CUR DN} PREZIME";PREZ$
90 PREZ$=LEFT$(PREZ$,20)
100 INPUT"(CUR DN){CUR DN} ULICA I BROJ";UL$
110 UL$=LEFT$(UL$,20)
120 INPUT"(CUR DN){CUR DN} GRAD";GRAD$
130 GRAD$=LEFT$(GRAD$,15)
140 INPUT"(CUR DN){CUR DN} BROJ TELEFONA";TEL$
150 TEL$=LEFT$(TEL$,12)
160 GOSUB 700: GOSUB 500
180 PRINT#5,IME$: GOSUB 500
190 BR=BR+12
200 GOSUB 700: GOSUB 500
210 PRINT#5,PREZ$: GOSUB 500
220 BR=BR+22
230 GOSUB 700: GOSUB 500
240 PRINT#5,UL$: GOSUB 500
250 BR=BR+22
260 GOSUB 700: GOSUB 500
270 PRINT#5,GRAD$: GOSUB 500
280 BR=BR+17
290 GOSUB 700: GOSUB 500
300 PRINT#5,TEL$: GOSUB 500
305 IF RBS=N THEN GOTO 340
310 RBS=RBS+1
320 GOSUB 600
330 GOTO 50
340 CLOSE 5: CLOSE 15
350 PRINT"(CLR){CUR DN}{CUR DN} KRAJ RADA PROGRAMA "
360 END
500 INPUT#15,G,PG$,S,B
510 IF G<20 THEN RETURN
520 IF G<>50 THEN GOTO 540
530 RETURN
540 PRINT G,PG$,S,B:STOP:RETURN
600 L=RBS
610 IF L>255 THEN H=INT(L/256)
620 L=L-256*H
630 RETURN
700 PRINT#15,"P"CHR$(5)CHR$(L)CHR$(H)CHR$(BR)
710 RETURN

```

UCITAVANJE DATOTEKE

Ako ste kreirali relativnu datoteku, i ona već postoji na disku, pitanje je kako sada učitati slogove te datoteke. Postupak učitavanja sličan je postupku upisivanja, tako da će vam biti lako da ga savladate. Otvaranje već postojeće relativne datoteke vrši se naredbom OPEN oblika:

```
OPEN 5,8,5,"PODACI"
```

(datoteka PODACI kreirana je u jednom od prethodnih primera), i disk već ima informaciju da je to relativna datoteka.

Program za učitavanje relativne datoteke

```
10 OPEN 15,8,15
20 OPEN 5,8,5,"PODACI"
30 PRINT "{CLR}{CUR DN}{CUR DN} UBACITE REDNI BRDJ SLOBA"
40 PRINT "{CUR DN} KOJI ZELITE DA UCITATE"
45 INPUT "{CUR DN} ILI NULU ZA KRAJ PROGRAMA";RBS
50 GOSUB 500
55 IF RBS=0 THEN 80 TO 200
60 BR=1
70 GOSUB 600:GOSUB 700
90 INPUT#5,IME$:GOSUB 700
100 PRINT "{CLR}{CUR DN} IME: ";TAB(20)IME$
110 BR=BR+12
120 GOSUB 600:GOSUB 700
130 INPUT#5,PREZ$:GOSUB 700
140 PRINT "{CUR DN} PREZIME: ";TAB(20)PREZ$
150 BR=BR+22
160 GOSUB 600:GOSUB 700
170 INPUT#5,UL$:GOSUB 700
180 PRINT "{CUR DN} ULICA I BRDJ: ";TAB(20)UL$
190 BR=BR+22
200 GOSUB 600:GOSUB 700
210 INPUT#5,BRAD$:GOSUB 700
220 PRINT "{CUR DN} GRAD: ";TAB(20)BRAD$
230 BR=BR+17
240 GOSUB 600:GOSUB 700
250 INPUT#5,TEL$:GOSUB 700
260 PRINT "{CUR DN} BROJ TELEFONA: ";TAB(20)TEL$
265 BET A$:IF A$="" THEN 265
270 GOTO 30
280 CLOSE 5:CLOSE 15
290 PRINT "{CLR}{CUR DN}{CUR DN} KRAJ RADA PRORAMA"
300 END
500 L=RBS
510 IF L>255 THEN H=INT(L/256)
520 L=L-256*H
530 RETURN
600 PRINT#15,"P"CHR$(5)CHR$(L)CHR$(H)CHR$(BR)
610 RETURN
700 INPUT#15,B,PB$,S,B
```



```

710 IF B<20 THEN RETURN
720 IF B<>50 THEN 740
730 PRINT"(CLR)(CUR DN) NETACAN REDNI BROJ SLOBA":RUN
740 PRINT B,PG$,S,B
750 STOP:RETURN

```

Listing H - 3 Učitavanje relativne datoteke
* UCITAVANJE

Objašnjenje programa:

- 10 - 20 - otvaranje komandnog kanala 15 i kanala za relativnu datoteku
 - 30 - 45 - unošenje rednog broja sloga koji želite da učitajte ili 0 za kraj rada
 - 50 - poziv potprograma koji pretvara RBS u oblik L+H*256
 - 55 - kriterijum za kraj rada je unošenje rednog broja sloga koji je jednak 0, RBS = 0
 - 60 - postavljanje promenljive koja sadrži poziciju od koje se vrši učitavanje u okviru sloga, na vrednost 1. Znači, učitava se sadržaj sloga počev od prve pozicije
 - 70 - poziv potprograma za postavljanje pointera, koji omogućava učitavanje ga čiji je redni broj unet na početku programa
 - 80 - poziv potprograma koji proverava postojanje greške u radu sa diskom
 - 90 - učitavanje sadržaja prvog polja IME\$ iz relativne datoteke. Poziv potprograma koji proverava postojanje greške u radu sa diskom
 - 100 - štampanje na ekranu poruke "IME JE sadržaj promenljive IME\$"
 - 100 - 260 - učitavanje sadržaja svih ostalih polja i sloga, koji učitavamo, pri tom se izvršavaju postupci analogni postupcima opisanim u linijama 60 - 100. Jedina razlika je u tome što se u promenljivu BR postavljaju vrednosti 13(1 + 12), 35(13 + 22), 57(35 + 22), 64(57 + 17). Posle izvršenog učitavanja na ekranu se štampaju odgovarajuće poruke i sadržaj učitanih polja
 - 270 - povratak na liniju 30 i unošenje rednog broja sledećeg sloga koji treba učitati
 - 280 - zatvaranje otvorenih kanala
 - 290 - poruka o kraju rada programa
 - 300 - kraj programa
- Zatim slede potprogrami.

500 - 530

Potprogram koji pretvara redni broj sloga RBS u oblik L+H*256.

600 - 610

Potprogram koji pozicionira pointere bafera diska i omogućava učitavanje pojedinog polja u okviru sloga.

700 - 750

Potprogram za proveru greške u radu sa diskom.

Ako je $G < 20$, greška ne postoji i vrši se povratak u program (linija 710).

Ako je G različito od 50, tada treba stampati broj greške, poruku o grešci, broj staze i bloka gde je greška, i zaustaviti izvršenje programa (linije 720, 740, 750).

Ako je $G = 50$, to je greška RECORD NOT PRESENT, znači ne postoji slog u datoteci sa tim rednim brojem. Zato se daje poruka na ekranu "NEMOGUC REDNI BROJ SLOGA" i ponovo startuje program (linija 730).

IZMENA SADRZAJA DATOTEKE - AZURIRANJE

AZURIRANJE je izmena sadržaja slogova već postojeće datoteke. Na primer, ako posmatramo datoteku PODACI, koja sadrži slogove sa imenima, prezimenima, adresom i brojem telefona neke grupe lica, tada ako se nekom licu promeni broj telefona, tu promenu je potrebno izvršiti u slogu tog lica. Kod sekvencijalne datoteke nije bilo moguće učitati i promeniti sadržaj sloga, a da se pri tom ne učitaju svi ostali slogovi. Kod relativne datoteke to je moguće. Taj postupak ilustrovan je sledećim primerom.

Program za ažuriranje slogova datoteke

Ovim programom se menjaju brojevi telefona u postojećim slogovima datoteke PODACI.

```

10 OPEN 15,8,15
20 OPEN 5,8,5,"PODACI"
30 PRINT "{CLR}{CUR DN}{CUR DN} UBACITE REDNI BROJ SLOGA"
40 PRINT "{CUR DN} ZA AZURIRANJE ILI NULU ZA"
45 INPUT "{CUR DN}          KRAJ PROGRAMA";RBS
50 GOSUB 500
60 IF RBS=0 THEN 160
70 PRINT "{CUR DN}{CUR DN}UNESITE POZICIJU OD KOJE JE "
80 INPUT "{CUR DN} UPISAN BROJ TELEFONA";BR
90 GOSUB 600;GOSUB 700
100 INPUT#5,TEL#:GOSUB 700
110 PRINT "{CLR}{CUR DN} STARI BROJ TELEFONE JE ";TEL#
120 INPUT "{CUR DN} UNESITE NOVI BROJ TELEFONA";TEL#
130 GOSUB 600;GOSUB 700
140 PRINT#5,TEL#:GOSUB 700
150 GOTO 30
160 CLOSE 5;CLOSE 15
170 PRINT "{CLR}{CUR DN}{CUR DN} ZAVRSENO AZURIRANJE"
180 END

```

```

500 L=RBS
510 IF L > 255 THEN H=INT(L/256)
520 L=L-256*H
530 RETURN
600 PRINT#15,"P"CHR$(5)CHR$(L)CHR$(H)CHR$(BR)
610 RETURN
700 INPUT#15,B,P8$,S,B
710 IF B<20 THEN RETURN
720 IF B<>50 THEN 740
730 RUN
740 PRINT B,P8$,S,B
750 STOP:RETURN

```

Listing H - 4 Azuriranje relativne datoteke
* AZURIRANJE

Objašnjenje programa:

- 10 - 20 - otvaranje komandnog kanala 15 i relativne datoteke 5
- 30 - 45 - unošenje rednog broja sloga RBS, koji treba azurirati ili 0 za kraj rada
- 50 - poziv potprograma za $RBS = L + H * 256$
- 60 - kriterijum za završetak rada je unošenje $RBS = 0$
- 70 - 80 - unošenje pozicije u okviru sloga, od koje počinje polje BROJ TELEFONA, koje treba azurirati (64-ta pozicija)
- 90 - poziv potprograma za pozicioniranje pointera bafera na 64-tu poziciju. Poziv potprograma za proveru greške u radu sa diskom
- 100 - učitavanje sadržaja polja TEL\$ iz relativne datoteke
- 110 - štampanje na ekranu poruke "STARI BROJ TELEFONA JE sadržaj polja TEL\$"
- 120 - unošenje novog broja telefona u polje TEL\$
- 130 - isti postupci kao u liniji 90
- 140 - upis novog sadržaja polja TEL\$ u relativnu datoteku
- 150 - povratak na liniju 30 i početak azuriranja sledećeg sloga
- 160 - zatvaranje komandnog kanala i relativne datoteke
- 170 - štampanje na ekranu poruke "ZAVRŠENO AZURIRANJE"
- 180 - kraj programa

Potprogrami koji slede detaljno su objašnjeni u prethodnom primeru, za učitavanje slogova datoteke.

Sa malim izmenama, ovaj program može da posluži i za azuriranje ostalih polja u slogu.

B A Z A P O D A T A K A

U prethodnim poglavljima upoznali ste se sa svim tipovima datoteka, i načinom rada sa tim tipovima. Ako ste to savladali, sada vas čeka najvažnije, rad sa bazama podataka. Program koji dajemo, na kraju poglavlja sadrži osnovni koncept rada sa bazama podataka. Kada ga savladate omogućice vam da uspešno koristite C-64 i u toj oblasti. Medjutim, napominjemo, da ovaj program treba da vas nauči da radite sa bazama podataka, što znači da on ne daje sve moguće varijante, koje zavise od specifičnosti oblasti primene, odnosno, ne sadrži u sebi rešenje baš svih problema vezanih za ovu oblast.

Osnovni koncept rada sa bazama podataka sastoji se iz sledećih aktivnosti:

- 1 - formiranje datoteka
- 2 - učitavanje datoteka
- 3 - pretraživanje datoteka
- 4 - azuriranje datoteka

Ovim programom biće detaljno objašnjen način na koji se izvršavaju ove aktivnosti. Sem toga program će pratiti jedan konkretan primer rada sa bazama podataka.

STRUKTURA I SADRŽAJ PRGRAMA

Program je napravljen tako da daje osnovne menije, koji vode korisnika programa kroz ceo program i primenu. Na početku ćemo dati osnovnu strukturu i sadržaj programa.

Glavni program (1 - 260)

U ovom delu programa vrši se:

- osnovna inicijalizacija, koja se sastoji u izboru boje ekrana, boje slova i formiranju početnog ekrana programa
- formiranje OSNOVNOG MENIJA iz koga se bira željena aktivnost
- realizacija željene aktivnosti pozivom odgovarajućeg potprograma

Slede potprogrami.

Kreiranje datoteke (270 - 660)

Postupak kreiranja datoteke sastoji se iz unošenja osnovnih parametara potrebnih za kreiranje datoteke. Ti osnovni parametri su

vezani za osnovni princip rada, koji ćemo objasniti kroz primer.

Upis u datoteku (670 - 970)

Potprogram daje MENI ZA UPIS U DATOTEKU. Kada je datoteka kreirana moguće je vršiti upis podataka u nju. Sem toga, ako postoji već ranije kreirana datoteka, moguće je odredjenim postupcima vršiti dodavanje podataka toj datoteci. Znači, ovaj meni daje dve mogućnosti:

- upis u novu datoteku (prethodno kreiranu)
- upis u postojeću datoteku (ranije kreiranu i popunjenu nekim podacima)

Izbor jedne od ove dve aktivnosti odvodi vas u potprogram koji realizuje tu aktivnost.

Upis u novu datoteku (980 - 1290)

Potprogram omogućava upis podataka u novu, prethodno kreiranu datoteku.

Upis u postojeću datoteku (1300 - 1600)

Potprogram omogućava upis podataka u ranije kreiranu datoteku u kojoj već postoje neki podaci.

Učitavanje datoteke (1690 - 2050)

Potprogram daje MENI ZA UCITAVANJE DATOTEKE. Ako postoje već formirane i popunjene datoteke, neophodno je predvideti i aktivnost koja omogućava učitavanje sadržaja datoteke. To učitavanje može da se vrši na dva načina:

- učitavanje svih slogova datoteke
- učitavanje izabranog sloga (kriterijum za izbor mora biti prethodno definisan)

Izbor jedne od ove dve aktivnosti odvodi vas u potprogram, koji realizuje tu aktivnost.

Učitavanje svih slogova (2060 - 2310)

Potprogram omogućava učitavanje svih slogova datoteke i štampanje na ekranu sadržaja tih slogova.

Učitavanje izabranog sloga (2320 - 2710)

Potprogram omogućava učitavanje izabranog sloga. Kriterijum za izbor sloga je sadržaj jednog polja sloga, i prethodno je definisan u delu za kreiranje datoteke. Učitani slog stampa se na ekranu.

Ažuriranje datoteke (3500 - 4030)

Potprogram omogućava ažuriranje izabranog sloga datoteke. Postupak se sastoji u izboru jednog odredjenog sloga, po prethodno utvrdjenom kriterijumu, učitavanju izabranog sloga i izmeni sadržaja polja u slogu, nekim novim sadržajem. Tako izmenjeni slog ponovo se upisuje u datoteku.

Provera greške (2720 - 2830)

Potprogram preko kanala 15 učitava grešku u radu sa diskom. Komanda

INPUT#15,G,P6\$,S,B

učitava broj greške G, poruku o gresci P6\$, broj staze i bloka gde se nalazi greška S i B. Za rad sa bazom podataka značajne su greške kod kojih je G = 50, G = 62 i G = 75. U zavisnosti od toga šta predstavljaju te greške, u samom programu predviđene su odredjene aktivnosti. Ukoliko se pojavi neka druga greška, ovaj potprogram omogućava stampanje na ekranu poruke o gresci, i zaustavljanje rada programa.

Pretvaranje RBS u oblik L + H * 256 (2840 - 2910)

Potprogram koji izračunava L i H za redni broj sloga RBS.

Kontrola dužine polja (2920 - 2970)

Potprogram koji sprečava da se u odredjeno polje unese sadržaj duži od predviđene dužine u postupku kreiranja datoteke.

Ažuriranje sifara - 1 (2980 - 3130)

Potprogram vrši ažuriranje DATOTEKE SIFARA, kada se vrši upis u novu ili postojeću DATOTEKU PODATAKA.

Pauza sa GET naredbom (3140 - 3200)

Potprogram koji omogućava prihvatanje određenog karaktera sa tastature. Sve dok ne unesete odgovarajući karakter, program ne radi ništa već čeka na taj unos.

Instrukcije (3210 - 3360)

Potprogram daje osnovne informacije o programu, i kratke instrukcije za rad.

Imena datoteka (3400 - 3495)

Potprogram stampa na ekranu imena datoteka sa kojima trenutno radite. Zatim vam daje mogućnost da promenite datoteke, izaberete nove datoteke sa kojima želite da radite, i unesete imena novih datoteka.

Azuriranje sifara - 2 (5000 - 5130)

Potprogram vrši azuriranje DATOTEKE SIFARA, kada se vrši postupak azuriranja DATOTEKE PODATAKA.

PRINCIP TRI DATOTEKE

Osnovni cilj ovog programa je kreiranje DATOTEKE PODATAKA, tako da bude moguće pretraživanje, azuriranje i učitavanje datoteke. Da bi to bilo moguće potrebno je DATOTEKI PODATAKA pridružiti još dve datoteke, DATOTEKU POLJA i DATOTEKU SIFARA. Sada ćemo objasniti strukturu ove tri datoteke.

DATOTEKA PODATAKA je datoteka u koju se smeštaju podaci, koje želimo da zadržimo, i sa njima kasnije radimo. Ova datoteka je po organizaciji relativna i pridružen joj je kanal 5. Struktura sloga je slobodna, formira je korisnik programa. Izgled sloga je:

```
*****
* POLJE-1 * POLJE-2 * POLJE-3 * POLJE-4 * ... *
*****
```

S1.1 - 1 Slog relativne DATOTEKE PODATAKA

Korisnik programa određuje broj polja u slogu, imena polja i dužinu polja. Pri tom važe sledeća ograničenja:

Pauza sa GEI naredbom (3140 - 3200)

Potprogram koji omogućava prihvatanje određenog karaktera sa tastature. Sve dok ne unesete odgovarajući karakter, program ne radi ništa već čeka na taj unos.

Instrukcije (3210 - 3360)

Potprogram daje osnovne informacije o programu, i kratke instrukcije za rad.

Imena datoteka (3400 - 3495)

Potprogram stampa na ekranu imena datoteka sa kojima trenutno radite. Zatim vam daje mogućnost da promenite datoteke, izaberete nove datoteke sa kojima želite da radite, i unesete imena novih datoteka.

Azuriranje sifara - 2 (5000 - 5130)

Potprogram vrši azuriranje DATOTEKE SIFARA, kada se vrši postupak azuriranja DATOTEKE PODATAKA.

PRINCIP TRI DATOTEKE

Osnovni cilj ovog programa je kreiranje DATOTEKE PODATAKA, tako da bude moguće pretraživanje, azuriranje i učitavanje datoteke. Da bi to bilo moguće potrebno je DATOTEKI PODATAKA pridružiti još dve datoteke, DATOTEKU POLJA i DATOTEKU SIFARA. Sada ćemo objasniti strukturu ove tri datoteke.

DATOTEKA PODATAKA je datoteka u koju se smestaju podaci, koje želimo da zapamtimo, i sa njima kasnije radimo: Ova datoteka je po organizaciji relativna i pridružen joj je kanal 5. Struktura sloga je slobodna, formira je korisnik programa. Izgled sloga je:

```
*****
* POLJE-1 * POLJE-2 * POLJE-3 * POLJE-4 * ... *
*****
```

Sl.I - 1 Slog relativne DATOTEKE PODATAKA

Korisnik programa određuje broj polja u slogu, imena polja i dužinu polja. Pri tom važe sledeća ograničenja:

- ukupna dužina sloga ne sme biti veća od 255 karaktera. U programu je predviđena rutina koja sprečava prekoracenje ove dužine
- imena polja mogu imati maksimalno 8 karaktera
- ukupan broj slogova ove datoteke je maksimalno 256

DATOTEKA POLJA je sekvencijalna datoteka, kojoj je pridružen kanala 6. U ovu datoteku smestaju se informacije o DATOTECI PODATAKA, koje su neophodne u kasnijim obradama. To su sledeće informacije:

- dužina sloga (sadržaj promenljive DS)
- broj polja u slogu (sadržaj promenljive BP)
- imena polja (sadržaj niza IME*(J))
- dužina polja (sadržaj niza DP(J))

Kako je ova datoteka sekvencijalna bitan je redosled smestanja podataka. Gore navedeni podaci smestaju se u redosledu koji je dat sledećom slikom:

```
*****
* DUZINA * BROJ * IME * DUZINA * IME * DUZINA * ... *
* SLOGA * PDLJA * POLJA-1 * POLJA-1* PDLJA-2 * POLJA-2* ... *
*****
```

Sl. 1 - 2 Izgled DATOTEKE POLJA

Par podataka, IME POLJA i DUZINA POLJA ponavlja se onoliko puta koliko ima polja u slogu relativne datoteke (sadržaj promenljive BP).

DATOTEKA SIFARA je relativa datoteka, kojoj je pridružen kanal 7. Sadržaj ove datoteke omogućava izvršavanje sledećih aktivnosti:

- dodavanje slogova već postojećoj datoteci
- učitavanje izabranog sloga
- ažuriranje izabranog sloga

Prvi slog ove datoteke sadrži samo jedan podatak, ukupan broj slogova u DATOTECI PDDATAKA, (sadržaj promenljive URBS). Ostali slogovi ove datoteke sadrže ključ po kome se pretražuje DATOTEKA PODATAKA. U DATOTECI PODATAKA izabere se jedno polje, čiji sadržaj predstavlja kriterijum za izbor sloga pri učitavanju, ili ažuriranju. To polje se naziva KLJUC PRETRAZIVANJA, i njegov sadržaj se pamti u DATOTECI SIFARA. Svaki slog DATOTEKE PODATAKA mora imati svoj KLJUC PRETRAZIVANJA po kome će kasnije biti

pronadjen, kada bude bilo potrebno da se ucita ili azurira. U ovom programu postoji samo jedan kljuc pretrazivanja za svaki slog, i to je sadrzaj prvog polja u slogu DATOTEKE PODATAKA. Medjutim, moguće je proširivanjem ovog programa uvesti više kljuceva pretrazivanja za jedan slog DATOTEKE PODATAKA. Na sledecoj slici dat je izgled DATOTEKE SIFARA:

```
*****
* UKUPAN BROJ * KLJUC * KLJUC * KLJUC * KLJUC * .... *
* SLOGDVA * 1.SLOGA * 2.SLOGA * 3.SLOGA * 4.SLOGA * .... *
*****
```

Sl. 1 - 3 Izgled DATOTEKE SIFARA

PRIMENA PROGRAMA

Nacin na koji se ovaj program koristi najlakše je pratiti preko primera. Zadatak je da se formira jedna konkretna datoteka, nad kojom će biti moguće izvršavati sve programom predviđene aktivnosti. Kreirajmo datoteku podataka ADRESAR koja ima slog oblika:

```
*****
* PREZIME * IME * TELEFON * ADRESA * GRAD *
*****
```

Duzina pojedinačnih polja neka bude:

- 20 karaktera za PREZIME
- 10 karaktera za IME
- 12 karaktera za TELEFON
- 20 karaktera za ADRESU (ulica i broj)
- 15 karaktera za GRAD

Kljuc po kome ce se kasnije pretrazivati datoteka je prvo polje u slogu, PREZIME.

Datoteci polja dodelimo ime ADP, a datoteci sifara ime ADS.

Inicijalizacija

Startujte program i na ekranu ce te dobiti:

PLM-SOFT

C-1985

RAD SA BAZAMA PODATAKA
DA LI SU VAM POTREBNE INSTRUKCIJE (D/N)

pronadjen, kada bude bilo potrebno da se ucita ili azurira. U ovom programu postoji samo jedan kljuc pretrazivanja za svaki slog, i to je sadrzaj prvog polja u slogu DATOTEKE PODATAKA. Medjutim, moguće je prosirivanjem ovog programa uvesti više kljuceva pretrazivanja za jedan slog DATOTEKE PODATAKA. Na sledecoj slici dat je izgled DATOTEKE SIFARA:

```
*****
* UKUPAN BROJ * KLJUC * KLJUC * KLJUC * KLJUC * ..... *
* SLOGOVA * 1.SLOGA * 2.SLOGA * 3.SLOGA * 4.SLOGA * ..... *
*****
```

Sl. 1 - 3 Izgled DATOTEKE SIFARA

PRIMENA PROGRAMA

Način na koji se ovaj program koristi najlakše je pratiti preko primera. Zadatak je da se formira jedna konkretna datoteka, nad kojom će biti moguće izvršavati sve programom predviđene aktivnosti. Kreirajmo datoteku podataka ADRESAR koja ima slog oblika:

```
*****
* PREZIME * IME * TELEFON * ADRESA * GRAD *
*****
```

Duzina pojedinačnih polja neka bude:

- 20 karaktera za PREZIME
- 10 karaktera za IME
- 12 karaktera za TELEFON
- 20 karaktera za ADRESU (ulica i broj)
- 15 karaktera za GRAD

Kljuc po kome ce se kasnije pretrazivati datoteka je prvo polje u slogu, PREZIME.

Datoteci polja dodelimo ime ADP, a datoteci sifara ime ADS.

Inicijalizacija

Startujte program i na ekranu ce te dobiti:

```
PLM-SOFT          C-19B5
```

```
      RAD SA BAZAMA PODATAKA
      DA LI SU VAM POTREBNE INSTRUKCIJE (D/N)
```

Ako su vam potrebne instrukcije odgovorite na pitanje sa D i na ekranu ce te dobiti kratke instrukcije:

INSTRUKCIJE

OVIM PROGRAMOM MOGUCE JE RADITI SA VISE RELATIVNIH DATOTEKA. RAD SE SASTOJI IZ SLEDECIH AKTIVNOSTI:

1. FORMIRANJE NOVIH DATOTEKA
2. UCITAVANJE DATOTEKA
3. DODAVANJE NOVIH SLOGOVA
4. AZURIRANJE DATOTEKE

AKTIVNOST SE BIRA IZ OSNOVNOG MENIJA PRITISNI NEKU TIPKU ZA DALJE

Ako odgovorite sa N, na ekranu ce te dobiti OSNOVNI MENI, iz koga birate zeljenu aktivnost:

OSNOVNI MENI

KREIRANJE NOVE DATOTEKE...1
 UPISIVANJE U DATOTEKU.....2
 UCITAVANJE DATOTEKE.....3
 AZURIRANJE DATOTEKE.....4
 KRAJ RADA.....5

IZABERITE BROJ AKTIVNOSTI

Kreiranje datoteke

Buduci da kreirate novu datoteku iz osnovnog menija izaberite broj 1. Na ekranu ce te dobiti:

KREIRANJE NOVE DATOTEKE

OSNOVNI PARAMETRI

<RETURN> ZA DALJE

a zatim, ekran koji od vas traži da unesete osnovne parametre potrebne za kreiranje datoteka. Prvi od tih parametara je BROJ POLJA koji u ovom primeru iznosi 5. Taj broj se smesta u promenljivu BP:

UNESITE BROJ POLJA ? 5

Zatim se pojavljuje ekran koji traži od vas da unesete IMENA POLJA i DUŽINU POLJA. U ovom primeru potrebno je uneti:

- PREZIME i broj 20
- IME i broj 10
- TELEFON i broj 12
- ADRESA i broj 20
- GRAD i broj 15

Izgled ekrana:

IME POLJA DO 8 KARAKTERA

```

IME POLJA.....1 ? PREZIME
DUZINA POLJA.....1 ? 20
IME POLJA.....2 ? IME
DUZINA POLJA.....2 ? 10
IME POLJA.....3 ? TELEFON
DUZINA POLJA.....3 ? 12
IME POLJA.....4 ? ADRESA
DUZINA POLJA.....4 ? 20
IME POLJA.....5 ? GRAD
DUZINA POLJA.....5 ? 15

```

Sledeće što treba da unesete su imena datoteka, ADRESAR, ADP i ADS. Izgled ekrana:

UNESITE NAZIVE DATOTEKA

```

DATOTEKA PDDATAKA.....? ADRESAR
DATOTEKA PDLJA.....? ADP
DATOTEKA SIFARA.....? ADS

```

Posle ste uneli sve potrebne parametre kompjuter vrši pripremu za kreiranje datoteke, koja traje izvesno vreme, i obavestava vas o tome:

```

SADA SE VR&I PRIPREMA ZA
KREIRANJE DATOTEKE

```

SACEKAJTE MALO

Posle izvršenja ove aktivnosti, vrši se povratak u osnovni meni.

Upisivanje u novu datoteku

Kada ste kreirali datoteku, sledeća aktivnost je upisivanje podataka u datoteku. Znači, treba da izaberete broj 2 iz osnovnog menija i na ekranu dobijete:

NAZIVI DATOTEKA U UPOTREBI

DATOTEKA PODATAKA...ADRESAR
 DATOTEKA POLJA.....ADP
 DATOTEKA SIFARA.....ADS

NOVE DATOTEKE ...DA/NE...? NE

Na ovaj način uvek znate koje su datoteke u upotrebi. Ako želite da radite sa datotekama čiji je spisak dat na prethodnom ekranu, odgovorite NE na postavljeno pitanje. Ako želite da promenite datoteke sa kojima radite, na postavljeno pitanje odgovorite sa DA. Zatim, unesite imena datoteka sa kojima želite da radite (te datoteke moraju već postojati). Zatim, na ekranu dobijate MENI ZA UPIS U DATOTEKU:

MENI ZA UPIS U DATOTEKU

UPIS U NOVU DATOTEKU.....1
 UPIS U POSTOJEĆU DATOTEKU....2

IZABERITE BROJ

Budući da je datoteka ADRESAR nova datoteka izaberite broj 1 iz ovog menija. Sledeći ekran je ekran za unosenje podataka. Unesite sledećih 5 slogova:

BABIC	VESNA	041-345-278	URALSKA 10	ZAGREB
PETROVIC	MIROSLAV	011-663-288	SAVSKA 5	BEOGRAD
JOVIC	SANJA	013-435-341	TRIGLAVSKA 30	PANCEVO
ANDJIC	ZORAN	088-211-852	HUMSKA 38	MOSTAR
ANTIC	MIRJANA	011-123-545	II BULEVAR 53	NOVI BEOGRAD

Izgled ekrana za unosenje prvog sloga je:

UNESITE Y ZA KRAJ PODATAKA

```

REDNI BROJ SLOGA   1
PREZIME           ? BABIC
IME               ? VESNA
TELEFON           ? 041-345-278
ADRESA            ? URALSKA 10
GRAD              ? ZAGREB
  
```

Kada ste uneli poslednji slog, unesite Y kao sadržaj prvog polja, i to će biti indikator za kraj unosenja. Na ovaj način upisali ste 5 slogova u datoteku podataka, ADRESAR. U datoteku polja, ADP upisana je dužina sloga (90 karaktera), broj polja (5), imena polja (PREZIME, IME, TELEFON, ADRESA i GRAD), i dužina odgovarajućih polja (20, 10, 12, 20, 15). U datoteku sifara ADS upisan je ukupan broj slogova (5), a zatim ključevi pretraživanja za svaki slog. Kako je za ključ pretraživanja određeno polje PREZIME, znači da su u datoteku sifara ASD kao ključevi upisani sledeći sadržaji:

- BABIC za 1-vi slog
- PETROVIC za 2-gi slog
- JOVIC za 3-ci slog
- ANDJIC za 4-ti slog
- ANTIC za 5-ti slog

Posle završenog upisivanja, vrši se povratak u osnovni meni.

Učitavanje svih slogova

Ako sada želite da vidite sadržaj, prethodno upisanih slogova, izaberite u osnovnom meniju broj 3. Kao i kod upisivanja, dobićete prvo ekran sa imenima datoteka koje su trenutno u upotrebi, a zatim ekran sa MENIJEM ZA UCITAVANJE:

MENI ZA UCITAVANJE DATOTEKE

```

UCITAVANJE SVIH SLOGOVA.....1
UCITAVANJE IZABRANOG SLOGA.....2
  
```

IZABERITE BROJ

Izaberite broj 1, za učitavanje svih slogova. Na ekranu ćete dobiti sadržaj 1-vog sloga:

SADRZAJ SLOGA 1

PREZIME BABIC
 IME VESNA
 TELEFON 041-345-278
 ADRESA URALSKA 10
 GRAD ZAGREB

<RETURN> ZA DALJE ILI *K* ZA KRAJ

Pritiskom na tipku RETURN dobićete sledeći ekran sa sadržajem 2-og sloga, itd. sve do poslednjeg sloga. Posle ekrana sa sadržajem poslednjeg sloga vrši se povratak u osnovni meni. Ako u toku učitavanja želite, posle nekog sloga da prekinete učitavanje, pritisnite slovo K.

Učitavanje izabranog sloga

Ako želite da učitate samo jedan odredjeni slog, u meniju za učitavanje izaberite broj 2. Recimo, da želite da učitate slog sa prezimenom ANDJIC, kada se pojavi sledeći ekran treba da unesete kao ključ pretraživanja, reč ANDJIC i kao broj karaktera ključa, broj 6. Izgled ekrana:

UCITAVANJE IZABRANOG SLOGA PO KLJUCU

KLJUC PRETRAZIVANJA...? ANDJIC
 BROJ KARAKTERA KLJUČA..? 6

Zatim, na ekranu dobijate sadržaj učitanoog sloga:

SADRZAJ SLOGA 4
 PREZIME ANDJIC
 IME ZORAN
 TELEFON 088-211-852
 ADRESA HUMSKA 38
 GRAD MOSTAR

<RETURN> ZA DALJE ILI *K* ZA KRAJ

Da u datoteci ADRESAR postoji još jedan slog sa istim ključem, ANDJIC, posle pritiska RETURN tipke, pojavio bi se sadržaj tog sloga. Budući da možete da birate broj karaktera ključa, možete npr. da učitate sve slogove sa ključem koji počinje slovom A ili slovima AN, i sl. Probajte da unesete kao ključ reč AN, i kao broj karaktera ključa broj 2. Na ekranu će te prvo dobiti sadržaj sloga čiji je ključ ANDJIC, a zatim posle pritiska RETURN tipke,

sadržaj sloga čiji je ključ ANTIC. Ako hocete da prekinete učitavanje izabranog sloga, pritisnite slovo K.

Ažuriranje datoteke podataka

Ako želite da promenite sadržaj nekog od upisanih slogova, u osnovnom meniju izaberite broj 4. Prvo će te dobiti na ekranu spisak imena datoteka koje su trenutno u upotrebi, a zatim ekran koji od vas traži da unesete ključ sloga koji treba ažurirati. Recimo, da želite da izmenite slog čiji je ključ JOVIC, unesite kao ključ pretraživanja tu rec, a kao broj karaktera ključa broj 5. Izgled ekrana:

AZURIRANJE IZABRANOG SLOGA PO KLJUCU

KLJUC PRETRAZIVANJA...? JOVIC
BROJ KARAKTERA KLJUCA..? 5

Ukoliko nema sloga sa takvim ključem dobićete ekran izgleda:

NEMA SLOGA SA TAKVOM SIFROM
PROBAJTE PONOVO

i posle toga se vrši povratak u prethodni ekran, na unosenje novog ključa. Ako je ključ ispravan na ekranu se pojavljuje sadržaj odgovarajućeg sloga:

SADRZAJ SLOGA 3

PREZIME	JOVIC
IME	SANJA
TELEFON	013-435-341
ADRESA	TRBLAVSKA 30
GRAD	PANCEVO

<RETURN> ZA DALJE ILI *K* ZA KRAJA

Iza imena prvog polja nalazi se kursor. Ukoliko želite da izmenite sadržaj prvog polja, otkucajte novi sadržaj preko starog i pritisnite tipku RETURN. Npr. promenite ime SANJA u SASA i broj telefona 013-431-341 u 013-422-222. Ako ne želite da menjate sadržaj ostalih polja pritisnite tipku RETURN kada se kursor nalazi iza imena odredjenog polja. Npr. ostavite ostala polja neizmenjena. Posle prolaska kroz sva polja sloga, vrši se ponovni upis izmenjenog sloga u datoteku ADRESAR. Ako ne želite više da ažurirate, sa K se vraćate u osnovni meni. Ako želite da proverite, da li je sadržaj sloga zaista izmenjen, učitajte slog sa

ključem JOVIĆ i na ekranu će te dobiti sadržaj ažuriranog sloga:

SADRŽAJ SLOGA 3

PREZIME JOVIC
 IME SASA
 TELEFON 013-422-222
 ADRESA TRIGLAVSKA 30
 GRAD PANCEVO

<RETURN> ZA DALJE ILI *K* ZA KRAJ

Upisivanje slogova u postojeću datoteku

Ako sada želite da dodate još nekoliko slogova u postojeću datoteku ADRESAR, u osnovnom meniju izaberite broj 2, a u meniju za upis, također, broj 2. Posle spiska imena datoteka koje su trenutno u upotrebi, dobićete ekran za upis 6-og sloga (poslednji upisani slog je imao redni broj 5). Unesite sledećih pet slogova:

BREBRIC	VERICA	041-321-458	KUZMIJAKOVA 25	ZAGREB
DJURDJEVIC	BOBAN	000-000-000	TAKOVSKA 10	KRAGUJEVAC
DAMJANIC	GORICA	011-331-548	20.OKTOBAR 58	BEOGRAD
JEVTIC	MILICA	021-378-956	GAJEVA 75	NOVI SAD
PERIC	VLADA	041-438-591	J. MIHICA 3	ZAGREB

izgled ekrana za unošenje 6-og sloga:

UNESITE Y ZA KRAJ PODATAKA

REDNI BROJ SLOGA 6
 PREZIME ? BREBRIC
 IME ? VERICA
 TELEFON ? 041-321-458
 ADRESA ? KUZMIJAKOVA 25
 GRAD ? ZAGREB

Kada unesete poslednji 10-ti slog, završite upisivanje, unošenjem slova Y. Na ovaj način upisali ste još 5 slogova u datoteku

ADRESAR, znači, u toj datoteci se sada nalazi 10 upisanih slogova. Datoteka sifara ADS je ažurirana, posle ovog upisivanja. Sadržaj prvog sloga ove datoteke (ukupan broj slogova), nije više 5 već 10. Sem toga, dodato je još 5 slogova, koji sadrže ključeve novih upisanih slogova. Ključevi pretraživanja su:

- BREBRIC za 6-ti slog
- DJURDJEVIC za 7-mi slog
- DAMJANIC za 8-mi slog
- JEVTIC za 9-ti slog
- PERIC za 10-ti slog

Posle završenog upisivanja vrši se povratak u osnovni meni.

Kraj rada

Kako ste prošli kroz sve aktivnosti programa, ako ne želite više da radite izaberite broj 5 iz osnovnog menija, za kraj rada programa.

Koristeći ovaj program možete formirati više različitih relativnih datoteka, koje možete koristiti u razne svrhe. Uvek morate imati na umu, da uz relativnu DATOTEKU PODATAKA uvek moraju ići još dve datoteke, koje omogućavaju izvršavanje svih aktivnosti, DATOTEKA POLJA i DATOTEKA SIFARA. Npr. ako kompjuter koristite samo kod kuće, možete formirati DATOTEKU PODATAKA sa spisakom svih knjiga iz vaše biblioteke, svih ploča iz vaše muzičke kolekcije ili eksponata iz vaše zbirke. Ako kompjuter koristite u nekoj poslovnoj obradi, možete formirati datoteku podataka u kojoj će se voditi razne evidencije potrebne za vaš posao i sl. Ako imate više formiranih relativnih datoteka podataka, ovaj program vam omogućava da radite sa svim tim datotekama jer vam uvek daje mogućnost da menjate datoteke sa kojima radite.

OBJASNJENJE PROGRAMA

Prethodno ste se upoznali sa načinom rada programa i njegovim mogućnostima. Međutim, da biste videli na koji način je omogućeno izvršavanje svih aktivnosti programa, daćemo detaljno objašnjenje programa. Ono vam može biti korisno, ukoliko želite da menjate program ili ga proširujete. Sem toga, može da vam posluži kao osnova da sami napravite svoj sopstveni program za rad sa bazama podataka, u skladu sa vašim zahtevima i potrebama. U tom smislu slobodno ga koristite.

Inicijalizacija

1

Dimenzionisanje nizova koji se koriste u programu.

40

Izbor boje ekrana i boje slova. Izabrana je crna boja za ekran a zelena za slova kao kod profesionalnog monitora. .

42 - 50

Formiranje početnog ekrana programa.

80 - 150

Formiranje ekrana osnovnog menija iz koga se bira zeljena aktivnost.

160 - 260

Izbor zeljene aktivnosti uslovljen je izborom broja od 1 do 5. Prihvatanje tog broja sa tastature vrši potprogram, koji se nalazi počev od linije 3170. U zavisnosti od sadržaja promenljive $B = VAL(B\$)$ izvršava se jedna od 5 mogućih aktivnosti, pozivom odgovarajućeg potprograma.

Ako je $B = 1$ vrši se poziv potprograma, koji počinje linijom 300 a služi za kreiranje nove datoteke.

Ako je $B = 2$ vrši se poziv potprograma, koji počinje linijom 850 a služi za upis u datoteku.

Ako je $B = 3$ vrši se poziv potprograma, koji počinje linijom 1720 a služi za učitavanje datoteke.

Ako je $B = 4$ vrši se poziv potprograma, koji počinje linijom 3500 a služi za ažuriranje datoteke.

Ako je $B = 5$ prelazi se na liniju 240, odnosno na kraj rada programa.

Kreiranje datoteke

310 - 330

Formiranje početnog ekrana.

360 - 460

Unosenje osnovnih parametara. Prvo se unosi broj polja u promenljivu BP, i postavlja dužina sloga DS na vrednost 2. Petlja koja počinje linijom 380 omogućava unosenje imena polja i dužina odgovarajućih polja. Broj prolazaka kroz petlju uslovljen je sadržajem promenljive BP. Sva uneta imena polja smestaju se u niz IME\$(J), a dužina odgovarajućih polja u niz DP(J). Izračunava se i dužina sloga tako što se u svakom prolasku kroz petlju, sadržaj promenljive DS povećava za dužinu polja DP(J) i 2 karaktera (jedan za CHR\$(13) i jedan za blanko znak). Na taj način se dobija tačna dužina sloga, koja odgovara formuli datoj u prethodnom poglavlju, RELATIVNE DATOTEKE. Ukoliko je DS manje od 255 sve je u redu, i nastavlja se sa unosenjem imena datoteka. Ako je DS veće ili jednako 255 na ekranu se stampa poruka "PREDUGACAK SLOG - UNESITE KRACA POLJA" i program obezbedjuje povratak na ekran za unosenje osnovnih parametara.

470 - 492

Unosenje imena DATOTEKE PDDATAKA u promenljivu IDAT\$, imena DATOTEKE POLJA u promenljivu POLJA\$, imena DATOTEKE SIFARA u promenljivu SDAT\$. Maksimalan broj karaktera za ova imena je 16. Linije 490, 491 i 492 vrše odsecanje suvišnih karaktera, ukoliko unesete ime duže od 16 karaktera.

511 - 512

Kreira se relativna DATOTEKA SIFARA (kanal 7), sa dužinom sloga jednakom dužini prvog polja DATOTEKE PDDATAKA, uvećanom za 2. Razlog za to je što slogovi ove datoteke sadrže ključeve pretraživanja DATOTEKE PDDATAKA, a odlučeno je da taj ključ bude sadržaj prvog polja.

520 - 590

Otvora se sekvencijalna DATOTEKA POLJA (kanal 6), za upisivanje a zatim se vrši upis sadržaja promenljivih DS (dužina sloga), BP (broj polja) i naizmenični upis imena i dužine svih polja u slogu, IME\$(J) i DP(J).

600 - 650

Otvora se kanal 15 neophodan za rad sa relativnom datotekom. Kreira se relativna DATOTEKA PDDATAKA (kanal 5), sa dužinom sloga koja je prethodno izračunata i smeštena u promenljivu DS. Zatim

se vrši upis sadržaja "KRAJ" u 256-ti slog ove datoteke. Taj upis omogućavaju naredbe:

```
PRINT#15,"P"CHR$(5)CHR$(0)CHR$(1)
PRINT#5,"KRAJ"
```

Kako je $L = 0$ a $H = 1$ to je $RBS = 0 + 1 * 256$, odnosno redni broj sloga je 256. Da bi se izvršio upis u 256-ti slog datoteke, prethodno treba rezervirati mesto na disketi za 255 prethodnih slogova. Na ovaj način se obezbeđuje, da cela DATOTEKA PODATAKA bude na jednoj disketi, što je neophodno. Ukoliko nema dovoljno mesta za ovu datoteku na disketi, crvena signalna lampica na disku naizmenično će se paliti i gasiti, i na taj način vas upozoriti da uzmete drugu disketu. Znači, kada kreirate DATOTEKU PODATAKA predvidite maksimalan broj slogova datoteke, izračunajte L i H za taj slog i naredbama gornjeg oblika rezervišite mesto na disketi za sve slogove. U ovom programu taj maksimalan broj je 256, s tim što se u 256-ti slog ne upisuje uobičajeni sadržaj već "KRAJ". Ako želite da promenite taj maksimalan broj slogova, izmenite L i H u liniji 620. Kad god se vrši upis u relativnu datoteku potrebno je proveriti da li postoji greška u radu sa diskom, i zato se poziva potprogram za proveru greške koji počinje linijom 2750. Zatim se vrši zatvaranje kanala 15 i kanala 5 za relativnu datoteku.

Upisivanje u datoteku

850 - 950

Formiranje ekrana sa menijem za upis u datoteku. Izbor aktivnosti vrši se izborom broja 1 ili 2, koristeći potprogram za prihvatanje karaktera sa tastature, GDSUB 3170.

Ako je B = 1 izvršava se GOSUB 700 a zatim GOSUB 1010.

Ako je B = 2 izvršava se GOSUB 700 a zatim GOSUB 1330.

Po povratku iz potprograma zatvara se kanal 15 i kanal 5 relativne datoteke.

700 - 790

Potprogram daje poruku na ekranu "UPIS U DATOTEKU", a zatim spisak imena datoteka koje su trenutno u upotrebi. Taj spisak se formira u potprogramu koji počinje linijom 3400. Pitanje daje mogućnost promene datoteka koje su u upotrebi. Spisak, pitanje i eventualno unošenje imena novih datoteka, daje poziv potprograma GOSUB 3400.

Otvara se kanal 15, kanal 6 sekvencijalne DATOTEKE POLJA, kanal 5 relativne DATOTEKE PODATAKA.

Vrši se provera kanala 15 zbog eventualne greške u radu sa diskom. Ako je $G = 62$ ili $G = 70$, to znači da ne postoji relativna datoteka u koju želimo da upisujemo, a čije je ime smešteno u promenljivu IDAT\$. Stoga je u ovom slučaju predviđeno štampanje poruke na ekranu "NEMA DATOTEKE SA TAKVIM IMENOM", zatvaranje otvorenih kanala i povratak u prethodni ekran koji daje mogućnost promene imena datoteke. Ako je G različito od 62 ili 70, znači da nema greške u radu sa diskom i nastavlja se rad programa.

1010 - 1290

Ovaj potprogram vrši upisivanje slogova u novu, znači praznu datoteku, koja je prethodno kreirana. Prvo se iz sekvencijalne DATOTEKE PDLJA (6) učitava sadržaj promenljive DS, zatim promenljive BP, i naizmenično članova nizova IME\$(J) i DP(J). To je potrebno zbog toga, što su ti podaci neophodni za formiranje ekrana za unošenje podataka. Posle zatvaranja datoteke 6, redni broj sloga RBS postavlja se na početnu vrednost 1, jer se upisuje prvi slog. GOSUB 2870 pretvara RBS u oblik $L + H * 256$ i izračunava L i H. Formira se ekran za unošenje podataka. Posle unošenja sadržaja svakog polja, kontroliše se dužina unetog sadržaja i eventualno otsecaju suvišni karakteri, sa GOSUB 2950. Linija 1180 obezbeđuje da unošenje slova Y bude znak za kraj unošenja podataka. Pozicionira se pointer bafera za upis sadržaja odgovarajućeg polja sloga relativne datoteke i vrši upis. Kada se izvrši upis svih polja sloga, poziva se potprogram za ažuriranje DATOTEKE \$IFARA sa GOSUB 3010. Ažuriranje datoteke \$ifara sastoji se u upisu trenutnog rednog broja sloga RBS, u prvi slog relativne DATOTEKE \$IFARA a zatim se u slog sa rednim brojem $RBS + 1$, znači u drugi jer je početna vrednost za $RBS = 1$, upisuje sadržaj prvog polja sloga DATOTEKE PODATAKA, koji predstavlja ključ pretraživanja potreban za kasnije pronalazjenje sloga. Postupak upisivanja se nastavlja povećanjem rednog broja RBS za 1, znači upisom drugog sloga, ažuriranjem datoteke \$ifara za $RBS = 2$, itd. sve do poslednjeg sloga koji se upisuje.

1330 - 1680

Ovaj potprogram vrši upis novih slogova u već postojeću datoteku, u koju su ranije upisani neki slogovi. Pravilan upis u postojeću datoteku, treba da obezbedi da upis novih slogova ne poremeti sadržaj ranije upisanih slogova. Uslov za to je, da redni broj sloga, koji se upisuje u postojeću datoteku, bude za 1 veći od rednog broja poslednjeg, ranije upisanog, sloga. Informacija o

ukupnom broju slogova, po završenom jednom upisivanju u DATOTEKU PODATAKA čuva se u relativnoj DATOTECI SIFARA, kao sadržaj prvog sloga. Npr. ako se vrši upis 5 slogova u novu datoteku, kako je to prethodno opisano, kada se upisuje prvi slog RBS = 1, sadržaj prvog sloga DATOTEKE SIFARA je 1, za RBS = 2 taj sadržaj je 2 itd. do RBS = 5 sadržaj je 5. Kada se završi upis sadržaj prvog sloga DATOTEKE SIFARA je 5, znati, redni broj poslednjeg upisanog sloga sačuvan je u prvom slogu DATOTEKE SIFARA.

Upis u postojeću datoteku vrši se na identičan način, kao i upis u novu datoteku. Jedina razlika je što se RBS ne postavlja na početnu vrednost 1, već se iz DATOTEKE SIFARA učitava prvi slog koji sadrži redni broj poslednjeg upisanog sloga, i taj broj uvećan za 1 postaje početna vrednost za RBS kod upisa u postojeću datoteku. Posle upisa svakog sloga, takodje se vrši ažuriranje DATOTEKE SIFARA, na isti način kao i kod upisa u novu datoteku. Posle završenog ponovnog upisivanja, prvi slog DATOTEKE SIFARA sadrži redni broj poslednjeg upisanog sloga, pri ponovnom upisivanju, a ostali slogovi ključeve pretraživanja svih upisanih slogova DATOTEKE PODATAKA. Obratite pažnju na činjenicu, da redni broj sloga u DATOTECI PODATAKA i redni broj sloga DATOTEKE SIFARA, koji sadrži ključ pretraživanja tog sloga nisu isti, već se razlikuju za 1. Npr. ako neki slog iz DATOTEKE PODATAKA ima RBS = 3, njegov ključ pretraživanja biće smešten u DATOTECI SIFARA u slog čiji je redni broj RBS = 4. Ovo je posledica toga, što je prvi slog DATOTEKE SIFARA, rezervisan da pamti redni broj poslednjeg upisanog sloga u DATOTEKU PODATAKA.

Učitavanje datoteke

1730

Formiranje poruke na ekranu "UCITAVANJE DATOTEKE".

1740

Formiranje ekrana sa spiskom datoteka u upotrebi i mogućnošću promene tih datoteka, GOSUB 3400.

1750 - 1850

Otvaranje kanala 15, kanala 6 sekvencijalne datoteke, kanala 5 relativne datoteke, provera postojanja relativne datoteke 5 sa takvim imenom.

1860 - 1940

Učitavanje informacija potrebnih za formiranje ekrana za učitavanje, iz sekvencijane datoteke 6. Zatvaranje otvorenih kanala.

1950 - 2040

Formiranje ekrana sa menijem za učitavanje koji nudi mogućnost učitavanja svih slogova i učitavanje izabranog sloga. Izbor aktivnosti zavisi od izbora broja 1 ili 2, koristeći GOSUB 3170.

Ako je B = 1 izvršava se GOSUB 2090.

Ako je B = 2 izvršava se GOSUB 2350.

2090 - 2310

Ovaj potprogram vrši učitavanje svih slogova. Učitavanje svih slogova vrši petlja, kroz koju se prolazi onoliko puta koliko ukupno ima slogova u DATOTEKI PODATAKA. Taj broj se čuva kao prvi slog DATOTEKE SIFARA. Zbog toga, prvo se učitava prvi slog DATOTEKE SIFARA u promenljivu M\$. Granicu brojača J, petlje definiše URBS = VAL(M\$). Posle zatvaranja relativne datoteke 7, otvara se relativna datoteka 5. Petlja koja počinje linijom 2160 a završava linijom 2260. omogućava učitavanje sadržaja jednog sloga i prikaz sadržaja tog sloga na ekranu. Linija 2270 omogućava nastavljanje ili prekidanje učitavanja u zavisnosti od izbora tipke RETURN ili slova K.

2350 - 2710

Ovaj potprogram vrši učitavanje jednog izabranog sloga. Izbor sloga uslovljen je unosom ključa pretraživanja u promenljivu IIDN\$, i brojem karaktera ključa koji se unosi u promenljivu KP. Iz DATOTEKE SIFARA 7, učitava se sadržaj prvog polja, ukupan broj slogova DATOTEKE PODATAKA. Petlja koja počinje linijom 2420 a završava linijom 2460, služi za pronalazjenje sloga u DATOTEKI SIFARA, koji odgovara unetom ključu. Kada je takav slog pronađen, on se učitava i njegov sadržaj štampa na ekranu, linije 2540 - 2660. Linija 2670 omogućava povratak u petlju, ako se pritisne RETURN tipka i na taj način nadju i ostali slogovi, koji imaju isti ključ pretraživanja, ili povratak u osnovni meni ako se pritisne K.

Azuriranje datoteke**3530**

Formiranje poruke na ekranu "AZURIRANJE DATOTEKE".

3535

Formiranje ekrana sa spiskom datoteka, koje su trenutno u upotrebi, i mogućnošću promene datoteka, sa 60SUB 3400.

3540 - 3578

Otvaranje kanala 15, kanala 6 sekvencijalne datoteke, kanala 5 relativne datoteke, provera postojanja relativne datoteke 5 sa takvim imenom.

3579 - 3587

Učitavanje informacija potrebnih za formiranje ekrana za ažuriranje, iz sekvencijalne datoteke 6. Zatvaranje otvorenih kanala.

3640 - 3650

Formiranje ekrana za unosenje ključa pretraživanja u promenljivu IIDN\$ i broja karaktera ključa u promenljivu KP.

3660 - 3700

Učitavanje ukupnog broja slogova DATOTEKE PODATAKA iz relativne DATOTEKE SIFARA.

3710 - 3840

Petlja koja služi za pronalazjenje rednog broja sloga DATOTEKE SIFARA, koji odgovara zadatom ključu pretraživanja. Kada je taj slog pronađen, ukoliko postoji, vrši se izlazak iz petlje i odlazak na liniju 3850. Ako takav slog ne postoji, štampa se na ekranu odgovarajuća poruka i vrši unosenje novog ključa pretraživanja.

3850 - 3980

Formiranje ekrana za ažuriranje. Učitavanje sadržaja sloga koji treba ažurirati, i prikaz tog sadržaja na ekranu. Linije 3950 - 3961 omogućavaju unosenje novog sadržaja sloga na istom ekranu.

3981 - 3987

Ponovno upisivanje izmenjenog sloga, polje po polje, u relativnu DATOTEKU PODATAKA.

3988

Povećanje rednog broja sloga za 1 i ažuriranje DATOTEKE SIFARA sa 60SUB 5000. Ovo ažuriranje se razlikuje od ažuriranja koje se vrši kada se upisuje u DATOTEKU PODATAKA po tome, što se ažuriraju ključevi slogova, ukoliko su promenjeni, ali ne i prvi slog koji sadrži ukupan broj slogova DATOTEKE PODATAKA.

3990 - 4030

Nastavljanje ili prekidanje ažuriranja u zavisnosti od izbora tipke RETURN ili slova K.

Ostali potprogrami detaljno su objašnjeni ili kroz gornji tekst ili u ranijim poglavljima.

VAŽNE NAPOMENE

1. Kada radite sa bazama podataka i koristite ovaj ili sličan program obavezno koristite dve odvojene diskete, jednu za program i drugu za datoteke. Ubacite u disk prvo disketu sa programom i napunite program. Zatim izvadite programsku disketu, ubacite u disk disketu za datoteku. Tada startujte program i izvršavajte sve aktivnosti programa.
2. Ako menjate program vodite računa da u jednom trenutku ne mogu biti istovremeno otvorene dve relativne datoteke. Ako se to desi, ova greška zablokirace vam kompjuter, tako da posle toga necete moći ništa više da radite. Jedini izlaz ce biti da isključite i ponovo uključite kompjuter.

```

5 DIM IME$(30),SIME$(30),DP(30)
10 POKE 53280,0:POKE 53281,0:PRINT "{GRN}"
40 A$="NEMA DATOTEKE SA TAKVIM IMENOM"
42 PRINT "{CLR}{RVS ON}                                {RVS OFF}"
45 PRINT "{RVS ON} PLM-SOFT                                C-1985 {RVS OFF}"
46 PRINT "{RVS ON}                                {RVS OFF}"
47 PRINT "{CUR DN}{CUR DN}      RAD SA BAZAMA PODATAKA"
50 PRINT "{CUR DN}{CUR DN}{CUR DN}{CUR DN}{CUR DN}{CUR DN}{CUR DN}
}{CUR DN}{CUR DN} DA LI SU VAM POTREBNE INSTRUKCIJE {D/N}"
60 GOSUB 3170
70 IF B$="D" THEN GOSUB 3240
80 REM OSNOVNI MENI
90 PRINT "{CLR}{CUR DN}{CUR DN}      {RVS ON}      OSNOVNI MENI      {RV
S OFF}"
100 PRINT "{CUR DN}{CUR DN}KREIRANJE  NDVE DATOTEKE...1"
110 PRINT "{CUR DN}{CUR DN}UPISIVANJE U DATOTEKU.....2"
120 PRINT "{CUR DN}{CUR DN}UCITAVANJE DATOTEKE.....3"
125 PRINT "{CUR DN}{CUR DN}AZURIRANJE DATOTEKE.....4"
130 PRINT "{CUR DN}{CUR DN}{CUR DN}KRAJ RADA.....5"
140 PRINT "{CUR DN}{CUR DN}{CUR DN}{CUR DN}{RVS ON} IZABERITE BRO
J AKTIVNOSTI {RVS OFF}"
150 REM
160 GOSUB 3170
170 B=VAL(B$)
180 IF B<1 OR B>5 THEN GOTO 80
190 IF B=1 THEN GOSUB 300
200 IF B=2 THEN GOSUB 850

```

```

210 IF B=3 THEN GOSUB 1720
215 IF B=4 THEN GOSUB 3500
220 IF B=5 THEN GOTO 240
230 GOTO 80
240 REM
250 PRINT "{CLR}{CUR DN}{CUR DN}{CUR DN} KRAJ RADA PROGRAMA"
260 END
270 REM *****
280 REM KREIRANJE NOVE DATOTEKE
290 REM *****
300 REM
310 PRINT "{CLR}{CUR DN}{RVS DN} KREIRANJE NOVE DATOTEKE {RVS OF
F}"
320 PRINT "{CUR DN}{CUR DN} DSNOVNI PARAMETRI"
330 PRINT "{CUR DN}{CUR DN}{CUR DN}{CUR DN}{CUR DN}{CUR DN}{CUR D
N}{CUR DN}{CUR DN}{CUR DN} <RETURN> ZA DALJE"
350 GOSUB 3170
360 INPUT "{CLR}{CUR DN}{CUR DN} UNESITE BROJ POLJA ";BP
370 DS=2
380 FOR J=1 TO BP
385 PRINT "{CLR} IME POLJA DO B KARAKTERA"
390 PRINT "{CUR DN}{CUR DN} IME POLJA.....";J;" ";
400 INPUT IME$(J)
410 PRINT "{CUR DN} DUZINA POLJA...";J;" ";
420 INPUT DP(J)
430 DS=DS+DP(J)+2
440 IF DS<255 THEN NEXT J:GOTO 470
450 PRINT "{CLR}{CUR DN}{CUR DN} PREDUGACAK SLOB-UNESITE KRACA PO
LJA"
460 GOTO 300
470 PRINT "{CLR}{RVS DN} UNESITE NAZIVE DATOTEKA {RVS OFF}"
480 INPUT "{CUR DN}{CUR DN} DATOTEKA PODATAKA.....";IDAT$
482 INPUT "{CUR DN}{CUR DN} IME DATOTEKE PDLJA.....";POLJA$
484 INPUT "{CUR DN}{CUR D
N} IME DATOTEKE BIFARA ....";BDAT$
490 IDAT$=LEFT$(IDAT$,16)
491 PDLJA$=LEFT$(POLJA$,16)
492 SDAT$=LEFT$(SDAT$,16)
500 PRINT "{CLR}{CUR DN}{CUR DN}{RVS ON} SADA SE VRSI PRIPREMA Z
A {RVS OFF}"
505 PRINT "{CUR DN}{CUR DN}{RVS ON} KREIRANJE DATOTEKE {RVS OFF
}"
510 PRINT "{CUR DN}{CUR DN}{CUR DN}{CUR DN}{CUR DN}{CUR DN}{RVS O
N} SACEKAJTE MALO {RVS OFF}"
511 OPEN 7,B,7, SDAT$+",L,"+CHR$(DP(1)+2)
512 CLOSE 7
520 OPEN 6,B,6,PDLJA$+",S,W"
530 PRINT#6,DS
540 PRINT#6,BP
550 FOR J=1 TO BP
560 PRINT#6,IME$(J)
570 PRINT#6,DP(J)
580 NEXT J
590 CLOSE 6
600 OPEN 15,B,15
610 OPEN 5,0,5, IDAT$+",L,"+CHR$(DS)
620 PRINT#15,"P"CHR$(5)CHR$(0)CHR$(1)
630 PRINT#5,"KRAJ"
640 GOSUB 2750
650 CLOSE 5;CLOSE 15
660 RETURN
670 REM *****
680 REM UPISIVANJE U DATOTEKU
690 REM *****

```

```

700 REM
710 PRINT "{CLR}{CUR DN}{CUR DN}{RVS ON} UPIS U DATOTEKU {RVS D
FF}"
720 FOR T=1 TO 1000:NEXT:GOSUB 3400
730 OPEN 15,8,15
740 OPEN 6,8,6,POLJA$+"",S,R"
750 BOSUB 2750
760 OPEN 5,8,5,IDAT$
770 BOSUB 2750
780 IF B=62 OR B=70 THEN GOTO 800
790 RETURN
800 REM
810 PRINT "{CLR}{CUR DN}{CUR DN}"; A$
820 FOR I=1 TO 2000:NEXT I
830 CLDGE 5: CLOSE 6: CLOSE 15
840 GOTO 700
850 REM MENI
860 PRINT "{CLR}{CUR DN}{CUR DN}{RVS ON} MENI ZA UPIS U DATOTEKU
{RVS OFF}"
870 PRINT "{CUR DN}{CUR DN}UPIS U NOVU DATOTEKU.....1"
880 PRINT "{CUR DN}{CUR DN}UPIS U POSTOJECU DATOTEKU...2"
890 PRINT "{CUR DN}{CUR DN}{CUR DN}{CUR DN}{CUR DN}{CUR DN}{CUR D
N}{CUR DN}{CUR DN}{CUR DN} IZABERITE BROJ"
900 REM
910 BOSUB 3170
920 B=VAL(B$)
930 IF B<1 OR B>2 THEN GOTO 850
940 IF B=1 THEN BOSUB 700:GOSUB 1010
950 IF B=2 THEN BOSUB 700:GOSUB 1330
960 CLOBE 5: CLOSE 15
970 RETURN
980 REM *****
990 REM UPIS U NOVU DATOTEKU
1000 REM *****
1010 REM
1020 INPUT#6,DS
1030 INPUT#6,BP
1040 FOR J=1 TO BP
1050 INPUT#6,IME$(J)
1060 INPUT#6,DP(J)
1070 NEXT J
1080 CLOSE 6
1090 RBS=1
1100 BOSUB 2870
1110 REM NOVI$LOS
1120 PRINT "{CLR}UNESITE {RVS ON} Y {RVS OFF} ZA KRAJ PODATAKA"
1130 BR=1
1140 PRINT "{CUR DN}{CUR DN} REDNI BROJ SLOGA ";RBS
1150 FOR J=1 TO BP
1160 PRINT "{CUR DN}{CUR DN}"IME$(J)TAB(10);:INPUT SIME$(J)
1170 BOSUB 2950
1180 IF LEFT$(SIME$(J),1)="Y" THEN GOTO 1280
1190 BOSUB 2870
1200 PRINT#15,"P"CHR$(5)CHR$(L)CHR$(H)CHR$(BR)
1210 PRINT#5,SIME$(J)
1220 BR=BR+2+DP(J)
1230 BOSUB 2750
1240 ;
1250 NEXT J
1251 BOSUB 3010
1260 RBS=RBS+1
1270 GOTO 1110
1280 REM
1290 RETURN

```

```

1300 REM *****
1310 REM   UPIS U POSTOJECU DATOTEKU
1320 REM *****
1330 REM
1340 INPUT#6,DS
1350 INPUT#6,BP
1360 FOR J=1 TO BP
1370 INPUT#6,IME$(J)
1380 INPUT#6,DP(J)
1390 NEXT J
1400 CLOSE 6
1410 CLOSE 5
1420 OPEN 7,8,7,SDAT$
1430 PRINT#15,"P"CHR$(7)CHR$(1)CHR$(0)CHR$(1)
1440 GOSUB 2870
1450 INPUT#7,M$
1460 RBS=1+VAL(M$)
1470 GOSUB 2870
1480 CLOSE 7
1490 OPEN 5,8,5,IDAT$
1500 REM
1510 BR=1
1515 PRINT "{CLR} UNESITE {RVS ON} Y {RVS OFF} ZA KRAJ PODATAKA"
1520 PRINT "{CUR DN}{CUR DN} REDNI BROJ SLOBA   ";RBS
1530 FOR J=1 TO BP
1540 PRINT "{CUR DN}{CUR DN}";IME$(J);TAB(10);
1550 INPUT SIME$(J)
1560 GOSUB 2950
1570 IF LEFT$(SIME$(J),1)="Y" THEN BOTD 1670
1580 GOSUB 2870
1590 PRINT#15,"P"CHR$(5)CHR$(L)CHR$(H)CHR$(BR)
1600 PRINT#5,SIME$(J)
1610 BR=BR+2+DP(J)
1620 GOSUB 2750
1630 NEXT J
1640 GOSUB 3010
1650 RBS=RBS+1
1660 GOTO 1500
1670 REM
1680 RETURN
1690 REM *****
1700 REM   UCITAVANJE DATOTEKE
1710 REM *****
1720 REM
1730 PRINT "{CLR}{CUR DN}{CUR DN}{RVS DN} UCITAVANJE DATOTEKE {R
VS OFF}"
1740 FOR T=1 TO 1000:NEXT:GOSUB 3400
1750 OPEN 15,8,15
1760 OPEN 6,8,6,POLJA$+",S,R"
1770 GOSUB 2750
1780 OPEN 5,8,5,IDAT$
1790 GOSUB 2750
1800 IF G=62 OR G=70 THEN GOTO 1820
1810 GOTO 1860
1820 REM
1830 PRINT "{CLR}{CUR DN}{CUR DN}"; A$
1840 FOR I=1 TO 2000: NEXT I
1850 CLOSE5;CLOSE6;CLOSE15;GOTO 1720
1860 REM
1870 INPUT#6,DS
1880 INPUT#6,BP
1890 FOR J=1 TO BP
1900 INPUT#6,IME$(J)
1910 INPUT#6,DP(J)

```

```

1920 NEXT J
1930 CLOSE 6
1940 CLOSE 5
1950 REM MMENI
1960 PRINT "{CLR}{CUR DN}{RVS DN} MENI ZA UCITAVANJE DATDTEKE {
RVS OFF}"
1970 PRINT "{CUR DN}{CUR DN}UCITAVANJE SVIH SLOGOVA....1"
1980 PRINT "{CUR DN}{CUR DN}UCITAVANJE IZABRANOG SLOGA..2"
1990 PRINT "{CUR DN}{CUR DN}{CUR DN}{CUR DN}{CUR DN}{CUR DN}{CUR
DN} IZABERITE BRDJ"
2000 GOSUB 3170
2010 B=VAL(B$)
2020 IF B<1 OR B>2 THEN GOTO 1720
2030 IF B=1 THEN GOSUB 2090
2040 IF B=2 THEN GOSUB 2350
2050 RETURN
2060 REM *****
2070 REM UCITAVANJE SVIH SLOGOVA
2080 REM *****
2090 REM
2100 OPEN 7,B,7,SDAT$
2110 PRINT#15,"P"CHR$(7)CHR$(1)CHR$(0)CHR$(1)
2120 INPUT#7,M$
2130 URBS=VAL(M$)
2140 CLOSE 7
2150 OPEN 5,B,5,IDAT$
2160 FOR RBS=1 TO URBS
2170 GOSUB 2070
2180 PRINT"{CLR}{CUR DN}{CUR DN} SADRZAJ SLOGA "; RBS
2190 PRINT#15,"P"CHR$(5)CHR$(L)CHR$(H)CHR$(1)
2200 BR=1
2210 FOR J=1 TO BR
2220 PRINT#15,"P"CHR$(5)CHR$(L)CHR$(H)CHR$(BR)
2230 INPUT#5,SIME$
2240 PRINT"{CUR DN}" IME$(J);TAB(10)SIME$
2250 BR=BR+2+DP(J)
2260 NEXT J
2270 PRINT "{CUR DN} <RETURN> ZA DALJE ILI *K* ZA KRAJ"
2280 GOSUB 3170
2290 IF B$<>"K" THEN NEXT RBS
2300 CLOSE 5;CLOSE 15
2310 RETURN
2320 REM *****
2330 REM UCITAVANJE IZABRANOG SLOGA
2340 REM *****
2350 PRINT "{CLR}{RVS DN} UCITAVANJE IZABRANOG SLOGA PO KLJUCU {R
VS OFF}"
2355 INPUT "{CUR DN}{CUR DN} KLJUC PRETRAZIVANJA...";IIDN$
2360 INPUT "{CUR DN}{CUR DN} BRDJ KARAKTERA KLJUCA.";KP
2370 IIDN$=LEFT$(IIDN$,KP)
2380 OPEN 7,B,7,SDAT$
2390 PRINT#15,"P"CHR$(7)CHR$(1)CHR$(0)CHR$(1)
2400 INPUT#7,M$
2410 URBS=VAL(M$)
2420 FOR RBS=2 TO URBS+1
2430 GOSUB 2070
2440 PRINT#15,"P"CHR$(7)CHR$(L)CHR$(H)CHR$(1)
2450 INPUT#7,M$
2460 IF IIDN$=LEFT$(M$,KP) THEN GOTO 2540
2465 REM
2470 NEXT RBS
2475 CLOSE 7;CLOSE 15;GOTO 80
2480 CLOSE 7
2490 OPEN 5,B,5,IDAT$

```

```

2500 PRINT "(CLR) NEMA SLOGA SA TAKVOM SIFROM"
2510 PRINT "(CUR DN) PROBAJTE PONOVO"
2520 FOR Z=1 TO 2000:NEXT Z
2530 CLOSE 5:GOTO 2350
2540 REM
2550 RBS=RBS-1
2560 PRINT"(CLR)(CUR DN)(CUR DN) SADRZAJ SLOGA "; RBS
2570 GOSUB 2870
2580 CLOSE 7
2590 OPEN 5,B,5, IDAT$
2600 BR=1
2610 FOR J=1 TO BP
2620 PRINT#15,"P"CHR$(5)CHR$(L)CHR$(H)CHR$(BR)
2630 INPUT#5,SIME$
2640 PRINT "{CUR DN}" IME$(J);TAB(10)SIME$
2650 BR=BR+2+DP(J)
2660 NEXT J
2670 PRINT "(CUR DN) <RETURN> ZA DALJE ILI *K* ZA KRAJ"
2680 GOSUB 3170
2690 IF B$="K" THEN CLOSE 5:CLOSE 15:GOTO 80
2700 RBS=RBS+1:CLOSE 5:OPEN 7,8,7,SDAT$
2710 GOTO 2465
2720 REM *****
2730 REM          PROVERA GRESKE
2740 REM *****
2750 REM
2760 INPUT#15,B,PB$,S,B
2770 IF B<20 THEN GOTO 2820
2780 IF B=50 THEN GOTO 2820
2790 IF B=62 THEN GOTO 2820
2800 IF B=70 THEN GOTO 2820
2810 PRINT PB$:STOP:RETURN
2820 REM
2830 RETURN
2840 REM *****
2850 REM          RBS=L+H*256
2860 REM *****
2870 REM
2880 L=RBS:H=0
2890 IF L>255 THEN H=INT(L/256)
2900 L=L-256*H
2910 RETURN
2920 REM *****
2930 REM          KONTROLA DUZINE POLJA
2940 REM *****
2950 REM
2960 SIME$(J)=LEFT$(SIME$(J),DP(J))
2970 RETURN
2980 REM *****
2990 REM          AZURIRANJE BIFARA - 1
3000 REM *****
3010 REM
3020 CLOSE 5
3030 OPEN 7,B,7,SDAT$
3040 PRINT#15,"P"CHR$(7)CHR$(1)CHR$(0)CHR$(1)
3050 PRINT#7,RBS:RBS=RBS+1
3060 GOSUB 2870
3070 PRINT#15,"P"CHR$(7)CHR$(L)CHR$(H)CHR$(1)
3080 PRINT#7,SIME$(1)
3090 RBS=RBS-1
3100 BGSUB 2750
3110 CLOSE 7
3120 OPEN 5,B,5, IDAT$
3130 RETURN

```



```

3140 REM *****
3150 REM      PAUZA SA GET NAREDBOM
3160 REM *****
3170 REM
3180 GET B$:IF B$="" THEN GOTO 3180
3190 RETURN
3200 I
3210 REM *****
3220 REM      INSTRUKCIJE
3230 REM *****
3240 REM
3250 PRINT "(CLR)(CUR DN)(CUR DN)          INSTRUKCIJE"
3260 PRINT "(CUR DN)(CUR DN) OVIM PROGRAMOM MOGUCE JE RADITI SA"
3270 PRINT "  VISE RELATIVNIH DATOTEKA.KAD SE SA-"
3280 PRINT "  STOJI IZ SLEDECIH AKTIVNOSTI: "
3290 PRINT "(CUR DN)  1. FORMIRANJE NOVIH DATOTEKA"
3300 PRINT "(CUR DN)  2. UCITAVANJE DATOTEKA"
3310 PRINT "(CUR DN)  3. DODAVANJE NOVIH SLOBOVA DATOTECI"
3320 PRINT "(CUR DN)  4. AZURIRANJE DATOTEKE"
3330 PRINT "(CUR DN) AKTIVNOST SE BIRA IZ OSNOVNOG MENIA"
3340 PRINT "(CUR DN)(CUR DN) PRITISNI NEKU TIPKU ZA DALJE"
3350 GOSUB 3170
3360 IF B$("<") THEN GOTO 80
3400 REM *****
3410 REM      IMENA DATOTEKA
3420 REM *****
3430 PRINT "(CLR) (RVS ON) NAZIVI DATOTEKA U UPOTREBI (RVS OFF)

3440 IF W(">") THEN 3470
3441 PRINT "(CUR DN)(CUR DN) DATOTEKA PDDATAKA..";IDAT$
3442 PRINT "(CUR DN)(CUR DN) DATOTEKA POLJA.....";POLJA$
3443 PRINT "(CUR DN)(CUR DN) DATOTEKA SIFARA....";SDAT$
3450 INPUT "(CUR DN)(CUR DN)(CUR DN)(CUR DN) NOVE DATOTEKE ....DA
/NE...";ID$
3460 IF OD$(">")"DA" THEN RETURN
3470 INPUT "(CUR DN)(CUR DN) DATOTEKA PODATAKA.....";IDAT$
3480 INPUT "(CUR DN)(CUR DN) DATOTEKA POLJA.....";POLJA$
3490 INPUT "(CUR DN)(CUR DN) DATOTEKA SIFARA.....";SDAT$
3495 W=1:RETURN
3500 REM *****
3510 REM      AZURIRANJE DATOTEKE PODATAKA
3520 REM *****
3530 PRINT"(CLR)(CUR DN)(CUR DN)(RVS ON)  AZURIRANJE DATOTEKE (R
VB OFF)"
3535 FOR T=1 TO 1000:NEXT:GOSUB 3400
3540 OPEN 15,8,15
3545 OPEN 6,8,6,POLJA$+","S,R"
3550 GOSUB 2750
3555 OPEN 5,8,5,IDAT$
3560 GOSUB 2750
3565 IF 8=62 DR 6=70 THEN 80TD 3575
3570 80TD 3579
3575 REM
3576 PRINT "(CLR)(CUR DN)(CUR DN)"; A$
3577 FOR I=1 TO 2000: NEXT I
3578 CLOSE5:CLOSE6:CLOSE15:80TD 3530
3579 REM
3580 INPUT#6,DS
3581 INPUT#6,8P
3582 FOR J=1 TO 8P
3583 INPUT#6,IME$(J)
3584 INPUT#6,DP(J)
3585 NEXT J
3586 CLOSE 6
3587 CLOSE 5

```

```

3630 REM
3640 PRINT "{CLR}{CUR DN}{CUR DN}{RVS ON} AZURIRANJE IZABRANOG SLO
GA PO KLJUCU {RVS OFF}"
3645 INPUT "{CUR DN}{CUR DN} KLJUC PRETRAZIVANJA...";IIDN$
3650 INPUT "{CUR DN}{CUR DN} BROJ KARAKTERA KLJUCA.";KP
3660 IIDN$=LEFT$(IIDN$,KP)
3670 OPEN 7,8,7,SDAT$
3680 PRINT#15,"P"CHR$(7)CHR$(1)CHR$(0)CHR$(1)
3690 INPUT#7,M$
3700 URBS=VAL(M$)
3710 FOR RBS=2 TO URBS+1
3715 D=0
3720 GOSUB 2870
3730 PRINT#15,"P"CHR$(7)CHR$(L)CHR$(H)CHR$(1)
3740 INPUT#7,M$
3750 IF IIDN$=LEFT$(M$,KP) THEN O=1:GOTO 3850
3760 REM
3770 NEXT RBS
3780 IF O=1 THEN CLOSE 7:CLOSE 15:GOTO 3540
3790 CLOSE 7
3810 PRINT "{CLR} NEMA SLOGA SA TAKVOM SIFROM"
3820 PRINT "{CUR DN} PROBAJTE PONOVO"
3830 FOR Z=1 TO 2000:NEXT Z
3840 CLOSE:GOTO 3630
3850 REM
3860 RBS=RBS-1
3870 PRINT "{CLR}{CUR DN}{CUR DN} SADRZAJ SLOGA "; RBS
3890 GOSUB 2870
3900 CLOSE 7
3910 OPEN 5,8,5,IDAT$
3920 BR=1
3930 FOR J=1 TO BP
3940 PRINT#15,"P"CHR$(5)CHR$(L)CHR$(H)CHR$(BR)
3950 INPUT#5,SIME$(J)
3960 PRINT "{CUR DN}TAB(10)SIME$(J)
3961 PRINT "{CUR UP}"IME$(J);:INPUT SIME$(J)
3962 GOSUB 2950
3970 BR=BR+2+DP(J)
3980 NEXT J
3981 BR=1
3982 FOR J=1 TO BP
3983 PRINT#15,"P"CHR$(5)CHR$(L)CHR$(H)CHR$(BR)
3984 PRINT#5,SIME$(J)
3985 BR=BR+2+DP(J)
3986 GOSUB 2750
3987 NEXT J
3988 RBS=RBS+1:GOSUB 5000
3990 PRINT "{CUR DN} <RETURN> ZA DALJE ILI *K* ZA KRAJ"
4000 GOSUB 3170
4010 IF B$="K" THEN CLOSE 5:CLOSE 15:GOTO 80
4020 RBS=RBS+1:CLOSE:OPEN 7,8,7,SDAT$
4030 GOTO 3760
5000 REM *****
5010 REM AZURIRANJE 8IFARA - 2
5020 REM *****
5040 CLOSE 5
5050 OPEN 7,8,7,SDAT$
5060 GOSUB 2870
5070 PRINT#15,"P"CHR$(7)CHR$(L)CHR$(H)CHR$(1)
5080 PRINT#7,SIME$(1)
5090 RBS=RBS-1
5100 GOSUB 2750
5110 CLDSE 7
5120 OPEN 5,8,5,IDAT$
5130 RETURN

```

* BAZAPODATAKA

PROGRAM SUPERBASE 64

U ovom poglavlju ćemo vam predstaviti jedan od najboljih aplikativnih programa za rad sa bazama podataka, SUPERBASE 64. Nemamo nameru da objašnjavamo kako se koristi ovaj program, jer bi za to bila potrebna još jedna knjiga ove veličine. Želimo da vam damo sumarni pregled svih mogućnosti ovog programa, da biste mogli da uporedite rad sa datotekama, kada koristite ovaj program i kada koristite naredbe i komande standardnog BASICA-a.

RAD SA PROGRAMOM SUPERBASE 64

SUPERBASE program se isključivo može koristiti sa diskom, jer njegove mogućnosti daleko prevazilaze mogućnosti kasetofona. Ukoliko se koristi sa jednom disk jedinicom, baza podataka može da ima maksimalno 15 datoteka i neograničen broj programa vezanih za rad sa tim datotekama. Broj slogova u svakoj datoteci je teorijski neograničen, međutim praktično je ograničen na 170K, koliki je maksimalni kapacitet jedne diskete za 1541 disk jedinicu. Moguće je koristiti dve disk jedinice, i u tom slučaju skoro sva ograničenja nestaju.

SUPERBASE može da se koristi na dva načina:

- 1 - korišćenjem MENIJA pomoću kojih se sve komande unose direktno
- 2 - formiranjem programa, koji se sastoje od komandi, koje se direktno unose u prethodnom načinu

Programski jezik koji se koristi u ovom programu, je mešavina BASIC komandi i naredbi i sopstvenih komandi SUPERBASE programa.

Sve mogućnosti programa date su u dva osnovna MENIJA. MENI 1 daje vam mogućnost izbora sledećih aktivnosti:

- 1 - ENTER
- 2 - SELECT
- 3 - FIND
- 4 - OUTPUT
- 5 - CALC
- 6 - REPORT
- 7 - EXECUTE
- 8 - HELP

MENI 2 daje vam mogućnost izbora sledećih aktivnosti:

- 1 - FILE
- 2 - FORMAT
- 3 - BATCH
- 4 - SORT
- 5 - PROG
- 6 - MAINTAIN
- 7 - MEMO
- 8 - HELP

Kada koristite ovaj program uvek imajte na umu osnovne pojmove vezane za rad sa bazom podataka:

- BAZA PODATAKA
- DATOTEKA
- SLOG
- POLJE

BAZA PODATAKA predstavlja skup DATOTEKA u kojima se čuvaju razni podaci. Ove datoteke mogu biti povezane ili nepovezane, što zavisi od osnovnih principa koji se usvajaju na početku rada sa bazama podataka. Da biste lakše shvatili ove principe dajemo primer za oba slučaja.

Primer povezanih datoteka

Ako obradjujete plate u jednoj radnoj organizaciji, princip rada bio bi sledeći:

- svakom radniku dodeliti sifru po kojoj se raspoznaje
- grupisati potrebne podatke u posebne grupe od kojih će biti formirane datoteke
- formirati datoteku ličnih podataka radnika sa imenom LICNI PODACI
- formirati datoteku sa podacima o broju radnih časova, i ostalim podacima potrebnim za računanje plate, pod imenom RADNI PODACI
- formirati datoteku sa podacima o izostancima (bolovanje, vojne vežbe i sl.), pod imenom IZOSTANCI
- formirati datoteku sa podacima o odbicima (obavezni doprinosi, krediti i sl.), pod imenom ODBICI
- formirati datoteku sifara koja sadrži glavnu sifru radnika i sifre koje određenog radnika povezuju sa ostalim datotekama

Sve ove datoteke cine BAZU PODATAKA koja služi za obracun plata, tako da ime ove baze podataka može biti OBRACUN PLATA.

Primer nepovezanih datoteka

BAZA PODATAKA koja sadrži nepovezane datoteke je npr. baza koja sadrži sledeće datoteke:

- datoteku podataka o svim vašim prijateljima i poslovnim partnerima (lični podaci, adrese, brojevi telefona, profesija i sl.)
- datoteku podataka o vašim zbirkama (ploča, knjiga, maraka, značaka i sl.)
- datoteku podataka o vašem kućnom budžetu, troškovima i sl.

SUPERBASE vam dopušta da u jednu bazu podataka (DATABASE) smestite maksimalno 15 datoteka (FILE). S druge strane dopušta vam da formirate neograničen broj baza podataka, koje će te smestati na različite diskete. Moguće je čak, da vam jedna baza podataka bude smestena na više disketa.

DATOTEKE su formirane od SLDGVA čiju strukturu slobodno formirate, kombinujući POLJA. Taj postupak je olaksan činjenicom da su unapred definisani tipovi polja.

KEY - SIFRA POLJE je polje po kome se prepoznaje slog (vrši izbor sloga, azuriranje, sortiranje i sl.). SUPERBASE dozvoljava mogućnost postojanja slogova sa istom sifrom (DUPLICATE KEY). Medjutim, preporučuje se kod poslovne primene korišćenje jedinstvene sifre. Sifra može biti azbučni podatak (npr. ime, prezime i sl.), ili brojni podatak (npr. broj racuna, redni broj i sl.).

TEXT - TEKST POLJE sadrži bilo kakav tekst. Npr. polje koje sadrži prezime, ime, adresu i sl.

DATE - DATUM POLJE sadrži datum. Npr. datum rođenja, datum prodaje neke robe, datum prispeća narudzbine i sl.

NUMERIC - NUMERICKO POLJE sadrži neku brojnu vrednost. Npr. cena robe, broj artikala i sl.

CONSTANT - KONSTANTNO POLJE sadrži neku konstantnu vrednost. Npr. porez za neke artikle, kamate kredita i sl.

RESULT - REZULTAT POLJE sadrži rezultat nekog izračunavanja, u kome učestvuju ostala numerička polja. Npr. izračunavanje ukupnog iznosa na osnovu broja artikala, pojedinačne cene artikala i sl.

CALENDAR - KALENDAR POLJE sadrži datum koji se dobija kao rezultat preračunavanja nekog drugog datuma.

KREIRANJE BAZE PODATAKA

Prvo što treba da uradite je da kreirate BAZU PODATAKA. Na programskoj disketi SUPERBASE 64 nalazi se program START-UP koji se sam startuje na početku rada. Svrha mu je da postavi osnovne parametre za SUPERBASE program i omogući kreiranje baze podataka (detajno objašnjenje ovog programa daćemo nešto kasnije). Ovaj program formira početni ekran i traži od vas da unesete prvi podatak IME BAZE PODATAKA. Kada unesete to ime, vrši se kreiranje baze podataka i automatski startuje opcija FILE za kreiranje DATOTEKA. Sledeći podatak koji treba da unesete je IME DATOTEKE. Zatim se vrši kreiranje datoteke i automatski startuje opcija FORMAT u kojoj se definišu imena polja, tipovi polja i dužina polja, jednom rečju, oblik i struktura SLOGA. Posle definisanja sloga, vrši se popunjavanje slogova konkretnim podacima koristeći opciju ENTER. Ako želite da kreirate još jednu datoteku ponovite postupak:

- izaberite FILE opciju i unesite ime datoteke
- automatski će te dobiti format opciju, u kojoj definišete strukturu sloga
- izaberite ENTER opciju za popunjavanje slogova podacima

Ako se setite primera za rad sa bazom podataka u poglavlju RELATIVNE DATOTEKE, i uporedite gornji postupak sa svim naredbama koje ste morali da koristite, da biste kreirali jednu datoteku, videćete čega vas sve oslobadja rad sa SUPERBASE programom. Za kreiranje datoteke u SUPERBASE programu, potrebno je da znate da koristite komande:

FILE, FORMAT, ENTER

Dok za rad sa relativnim datotekama morate dosta znati o radu sa diskom, korišćenju kontrolnih kodova, radu sa relativnim datotekama i poznavati u potpunosti sve BASIC komande.

Medjutim, ako ste sve ove pojmove savladali i uspešno ih koristite, vi ste samim tim naučili osnovne principe rada sa

bazama podataka, i bice vam mnogo lakse da maksimalno efikasno koristite gotove programe, kao sto je SUPERBASE. Ako se bez ovih prethodnih znanja odmah upustite u rad sa gotovim programom, moze vam se desiti da mehanicki izvršavate postupke koje vam program omogućava, a da niste uopste svesni na cemu su oni bazirani. Stoga smatramo, da je osnovno znanje o radu sa diskom, datotekama i principima rada sa bazama podataka bitan preduslov za uspesno koriscenje gotovih programa. Ako savladate sve sto vam ova knjiga nudi, posle toga, rad sa SUPERBASE programom bice vam lak i u potpunosti razumljiv.

PRETRAZIVANJE DATOTEKA

Jedna od glavnih aktivnosti je pretraživanje datoteka po raznim kriterijumima. To pretraživanje vrši se u SUPERBASE programu pomoću opcija SELECT i FIND.

Prvo sto treba da uradite je da izaberete datoteku iz baze podataka, koju zelite da pretražujete, koristeći opciju FILE. Zatim, izaberite opciju SELECT i dobićete SELECT MENI koji vam daje razne mogućnosti pretraživanja:

- učitavanje prvog sloga, sledećeg, ... itd. redom do kraja datoteke, i prikaz sadržaja sloga na ekranu
- kada na ekranu imate jedan slog možete tražiti da vidite prethodni, sledeći, prvi slog u datoteci, poslednji slog u datoteci (opcije NEXT, PREVIOUS, FIRST, LAST)
- možete izabrati slog i po zadatoj sifri sloga (opcija KEY)
- možete izabrati slog i po bilo kom drugom kriterijumu vezanom, za sva ostala polja u slogu (opcija MATCH)

Pretraživanje datoteke pomoću opcije FIND takodje može da se vrši po zadatom kriterijumu vezanom za bilo koje polje u slogu. Dodatna prednost ove opcije je, da se grupa slogova koji zadovoljavaju kriterijum izdvaja automatski u posebnu datoteku i pamti kao posebna datoteka.

AZURIRANJE DATOTEKE

Promena sadržaja slogova može da se vrši na više načina:

- promena sadržaja odredjenog polja u jednom slogu. Moguće je promeniti sadržaje svih polja sem SIFRE SLOGA (KEY).

- Opcija koja to omogućava je REPLACE iz SELECT MENIJA
- dodavanje novih polja i njihovih sadržaja, opcija ADD iz SELECT MENIJA
 - istovremena promena određenih polja u svim slogovima datoteke, opcija BATCH iz MENIJA 2

SORTIRANJE DATOTEKE

Posle kreiranja datoteke i popunjavanja slogova podacima automatski se vrši sortiranje datoteke po sadržaju ŠIFRE sloga (KEY). Ako je šifra azbučni podatak, sortiranje se vrši po engleskom alfabetu. Ako je šifra numerički podatak, sortiranje se vrši po vrednostima od manjih ka većim.

Postoje i druge mogućnosti sortiranja, po drugim kriterijumima. Sve te mogućnosti daje vam opcija SORT. Posle sortiranja, nova sortirana datoteka automatski se pamti kao nova datoteka.

PRIKAZ DATOTEKA NA EKRANU I STAMPACU

Opcija DISPLAY daje vam mogućnost prikaza svih informacija o datotekama na ekranu dok opcija PRINT ima tu istu funkciju za štampač. Opcije ACROSS i DWN daju vam mogućnost prikaza (i na ekranu i na štampaču):

- liste svih slogova (sadržaji svih polja u slogu)
- liste svih slogova koja sem sadržaja polja sadrži i imena polja

Šem toga tu je i opcija REPRT iz MENIJA 2, koja ima izvanredne mogućnosti a isključivo je namenjena formiranju izveštaja na štampaču.

U delu knjige, STAMPAC nalazi se posebno poglavlje koje se odnosi na korišćenje raznih tipova štampača iz SUPERBASE programa. Ukoliko ste imali problema vezanih za ovu primenu nadamo se da će vam to poglavlje biti korisno i da će rešiti vaše probleme.

FORMIRANJE PROGRAMA

Sve ove aktivnosti koje ste izvršavali u direktnom modu, koristeći određene tipke i slova na tastaturi kompjutera, možete pisati u programskom modu. Program koji pišete veoma je sličan po obliku standardnom BASIC programu. Pisanje programa omogućava

programski editor, PROG koji se poziva iz MENIJA 2. Napisani program se može kao i standardni BASIC program zapamtiti na disketi pomoću komande:

SAVE "IME PROGRAMA"

i kasnije ponovo učitati komandom:

LOAD "IME PROGRAMA"

Izvršavanje programa omogućava opcija EXECUTE iz MENIJA 2.

PREGLED MOGUĆNOSTI SUPERBASE 64

Dosad smo nabrojali osnovne mogućnosti ovog programa, koje se odnose na rad sa bazom podataka odnosno, datotekama. Nabrojimo i ostale mogućnosti:

- 1 - povezivanje datoteka iz jedne baze podataka i formiranje novih datoteka ili izveštaja, koji se sastoje od podataka uzetih iz više datoteka
- 2 - komunikacija sa drugim programima, koja podrazumeva EXPORT i INPORT komunikaciju. Datoteke i baze podataka kreirane u drugim programima (gotovim i samostalno napravljenim), mogu biti unete u rad sa ovim programom i na njima se izvršavati sve aktivnosti SUPERBASE programa (EXPORT). Datoteke ili pojedinačni podaci iz datoteka mogu biti ulaz za druge programe, npr. za tekst procesore. Tekst procesor EASYSCRIPT može na jednostavan način da komunicira sa SUPERBASE programom. Zasto vam je to potrebno? Npr. ako želite da napisete pisma vašim poslovnim partnerima kojim ih pozivate na poslovni sastanak, sve informacije potrebne za ta pisma možete direktno uzeti iz datoteke podataka o poslovnim partnerima koja je kreirana u SUPERBASE programu. Više vam nije potrebna klasična beležnica, telefonski imenik, klasična pisaać mašina, indigo ili mašina za kopiranje. Sve što vam je potrebno je, vaš kompjuterski sistem (C-64, disk i štampač), i da otkucate osnovni tekst pisma. Dobicete pojedinačna pisma za svakog vašeg partnera sa njegovim sopstvenim podacima. Ovo je samo jedan primer. Sigurno da se i sami možete setiti još mnogo slučajeva gde možete primeniti ovu mogućnost SUPERBASE programa
- 3 - rad sa SUPERBASE programom olakšavaju vam HELP ekrani. To su pomoćni ekrani sa osnovnim informacijama o svim aktivnostima programa i načinu korišćenja tih aktivnosti
- 4 - moguće je zapamtiti nekoliko ekrana pomoću MEMO opcije i

RAD SA BAZOM PODATAKA

Na kraju dajemo jedan primer programa koji koristi SUPERBASE komande i mogućnosti. Ovaj program omogućava izvršavanje istih aktivnosti kao i program dat u poglavlju BAZA PODATAKA. Razlika je u tome što se u ovom programu maksimalno koriste mogućnosti programa SUPERBASE 64.

```

5 rem *** inicijalizacija ***
10 brkon:
20 display @1,2" "
30 display @1,3" "@+" "
40 display @1,4" "@+" Plm-soft c-1985 "
50 display @1,5" "@+" "
60 display @4,10"rad sa bazama Podataka"
65 display @0
70 ask @1,14" da li su vam Potrebne instrukcije(d/n)"
   :P$
80 if P$="d"gosub 1100
90 if P$="n"goto 110
100 goto 20
110 rem *** osnovni meni ***
120 display @1,1" "
130 display @2,2" "@+" osnovni meni "
140 display @3,4"kreiranje datoteke.....1"
150 display @3,5"upisivanje u datoteku....2"
160 display @3,6"ucitavanje datoteke.....3"
170 display @3,7"azuriranje datoteke.....4"
180 display @3,8"kraj rada.....5"
185 display @0
190 ask @3,12"izaberite broj aktivnosti ";a
200 on a gosub 300,500,700,900
210 goto 10
300 rem *** kreiranje datoteke ***
310 display @1,1" "
320 display @1,2" "@+" kreiranje nove datoteke "
330 display @3,4" osnovni Parametri"
340 display @3,10" <return> za dalje"
345 display @0
350 wait
360 display @1,1" "
370 display @1,1"unesite ime baze Podataka "
375 display @0
380 ask @4,2" ";a$
385 database a$,8,0:rem kreiranje baze Podataka

```

```
390 file:rem superbase trazi ime datoteke
400 format:rem definisanje oblika slova
410 return
500 rem *** upisivanje u datoteku ***
510 display @1,1" "
520 display @1,3" "@" upisivanje u datoteku"
530 display @2,5"upis u novu datoteku.....1"
540 display @2,6"upis u Postojecu datoteku...2"
545 display @0
550 ask @2,9"izaberite broj ";s
560 on 990sub 570
565 goto 500
570 file:rem izbor datoteke
590 enter:rem unosenje Podataka u slovove
600 return
700 rem *** učitavanje datoteke ***
710 display @1,1" "
720 display @1,2" "@" meni za učitavanje "
730 display @2,4"učitavanje svih slova.....1"
740 display @2,6"učitavanje izabrano9 slova...2"
745 display @2,8"kraj rada.....3"
750 display @0
760 ask @2,11"izaberite broj ";q
770 on 990sub 800,850,880
780 goto 700
790 file
800 output display all records
810 return
840 file
850 file
855 select key:rem unosenje sifre slova
860 select o
870 return
880 goto 110
900 rem *** azuriranje ***
910 file
920 select key
930 select r
940 return
```

S T A M P A C

Teško je preporučiti u kom poretku je najbolje nabavljati I/O uređaje, o kojima smo pisali ili ih samo pomenuli u ovoj knjizi. Ova nedoumica se uglavnom odnosi na kućnu upotrebu kompjutera, jer je cena diska i štampača veća od cene samog C-64. Šta više, mnoge vrste flopi diskova i štampača, značajno ga prevazilaze u ceni. U tom smislu, treba pažljivo odmeriti želje, mogućnosti i potrebe. Što se tiče poslovne primene, smatramo da tu ne bi smelo biti nedoumice. Mislimo da cena jednog kompletnog sistema ne bi trebalo da bude problem ni za manje radne organizacije, ako se uzme u obzir za koje i kakve namene se odvajaju znatno veća materijalna sredstva. S druge strane, korist koju može doneti primena ovakvih mikrosistema, značajno prevazilazi uložena sredstva, kako zbog brže i ažurnije obrade podataka, tako i zbog pripreme zaposlenih za prelazak na veće i bolje sisteme, kojima cene svaki dan padaju.

Štampač prevashodno služi za:

- štampanje programa
- štampanje raznih podataka u obliku tabela i grafikona
- štampanje teksta
- specijalne namene

TIPOVI ŠTAMPACA

Postoji mnogo vrsta štampača i podelu je moguće izvršiti u odnosu na razne kriterijume, kao što su:

- brzina štampača
- način stampe
- grafičke mogućnosti, ...

Uobičajeno je da se štampači svrstavaju u odnosu na tip stampe, pa postoje:

- MATRICNI ŠTAMPACI
- LEPEZASTI ŠTAMPACI ("DAISYWHEEL")
- PLOTERI
- INK-JET ŠTAMPACI

Ogroman je raspon u ponudi štampača. Izbor ove periferijske I/O jedinice pre svega zavisi od vaših potreba za kvalitetom i brzinom stampe, u odnosu na vaše mogućnosti. Ne treba mnogo napomin-

jati, da što su zahtevi veći to je i stampac skuplji. Podjimo zato od tipova stampaca koji se redje sreću u našim uslovima, zbog njihove cene.

MATRICNI STAMPACI se najčešće sreću u konfiguracijama sa kompjuterom, jer je u ovoj grupi moguće naći i vrlo jeftine, ali i vrlo skupe stampace. Danas postoje matricni stampaci, koji nose LQ oznaku kvaliteta, kao što je EPSON LQ 1500, i samim tim spadaju u najvišu klasu matricnih stampaca. Srednja klasa ovih stampaca nosi oznaku kvaliteta NLQ (Near Letter Quality), što znači da im je kvalitet skoro kao kod standardnih pisacih mašina. U najnižoj klasi, samim tim i najjeftinijoj, se nalaze stampaci koji se najčešće sreću u našim uslovima.

U glavi matricnog stampaca se nalazi 7-11 vertikalno poredjanih iglica, koje pomoću elektromagneta a prema unapred programiranom poretku, preko trake za kucanje stvaraju otisak željenog slova ili grafičkog simbola.

Kako cilj ovog poglavlja o stampacima, nije da vam omogući izbor stampaca, to se nećemo detaljnije upustati u principe rada svake pomenute grupe, sa analizom uporednih karakteristika. Za sve one koji bi želeli da saznaju više informacija o ovoj temi kao i da vide uporedne preglede raznih vrsta stampaca, preporučujemo sledeće članke:

STAMPACI - "RACUNARI U VASOJ KUĆI", br. 1

STAMPACI U AKCIJI - "RACUNARI U VASOJ KUĆI", br. 2

kao i uporednu tabelu stampaca datu u časopisu "MOJ MIKRO", januar 1985.

S druge strane, mi bismo hteli da vam pružimo određene informacije koje su vezane za princip rada C-64 i raznih tipova stampaca.

CBM STAMPACI

U ovu grupu stampaca, svrstavamo sve one stampace, koji su specijalno namenjeni za rad sa C-64 i VIC-20. To su pre svih:

1. C-1525
2. SEIKOSHA GP 100 VC
3. MPS 801
4. MPS 802

Od ova četiri nabrojana stampaca, prva tri imaju skoro identične osobine, s tim što je MPS 801 nešto brži u odnosu na prva dva. MPS 802 je najkvalitetniji od nabrojanih stampaca. Pogledajmo osnovne karakteristike ovih stampaca:

	C-1525	S-6P-100VC	MPS 801	MPS 802
MATRICA	7x6	7x6	7x6	8x8
BRZINA (karak./sec)	30	30	50	60
DEFORMISANA SLOVA	DA	DA	DA	NE
STAMPANJE:				
- jednosmerno	DA	DA	DA	-
- dvosmerno	NE	NE	NE	DA
PAPIR:				
- perforiran	DA	DA	DA	DA
- običan A4	NE	NE	NE	DA
- u traci	NE	NE	NE	DA
REDEFINISANJE				
KARAKTERA	NE	NE	NE	NE

Tab. K - 1 Uporedne karakteristike CBM stampaca

Ono što izdvaja ove stampace od ostalih je jednostavnost povezivanja (vidi sl. A - 2). Uz stampac dobijate DIN kabl, koji uključujete u SERIAL PORT na C-64 ili u SERIAL PORT na disku (ako ga imate) i celo povezivanje je gotovo. To znači, da vam nije potreban interfejs niti bilo kakvi specijalni dodaci.

Sledeća prednost ovih stampaca je njihova potpuna kompatibilnost i transparentnost sa C-64, kao i sa većinom profesionalnih programa pisanih za C-64. To se odnosi i na dve vrlo korisne naredbe iz SIMONS BASIC-a:

- COPY - kopira ekran visoke rezolucije na stampac
- HRDCPY - kopira kompletan standardni ekran na stampac (tzv. hard kopija)

S druge strane mora se priznati, da ovi stampaci (sa izuzetkom modela MPS 802) imaju i dosta slabosti, po pitanju njihovih mogućnosti u stampi i potrebne brzine. Osnovna slabost im je deformisanje slova u stampi, zbog male matrice koju koriste za definisanje slova (7x6), zbog koje se i velika i mala slova kucaju od iste osnovne linije.

Poseban problem je, što ni na jednom od pobrojanih modela ne možete definisati YU karaktere, programskim putem.

Iz prakse znamo da mnogi korisnici budu zavedeni cinjenicom da se na ovim stampacima može definisati 1 KARAKTER, pa iz toga pogrešno izvedu zaključak da je i na ovim tipovima stampaca moguće definisati YU set karaktera. Zato, ako su vam u radu neophodni YU simboli, nemojte kupovati ove stampace, dodajte par stotina maraka za model koji ima mogućnost definisanja korisničkog seta karaktera.

Rad sa CBM stampacima

Rad sa stampacima je relativno jednostavan. Ako ste savladali rad sa kasetofonom a pogotovo sa diskom, onda ce vam rad sa stampacem biti zaista jednostavan.

Da biste radili sa stampacem, morate imati osnovna znanja o:

- radu sa kompjuterom
- programiranju u BASIC-u
- radu sa datotekama
- komuniciranju sa I/O uredjajima

Ako sa stampacem radite isključivo, sa tekst procesorom, onda su vam prethodna znanja potrebna u minimalnoj meri, ali zato morate detaljno prouciti uputstvo za rad sa tekst procesorom. Ono o čemu cemo sada govoriti, odnosi se na upotrebu stampaca iz BASIC-a.

U osnovi gledano, stampanje je prenos podataka sa ekrana na stampac. Da biste to uradili, potrebno je primeniti par BASIC komandi, koje smo često upotrebljavali u ovoj knjizi.

To je pre svega, naredba OPEN. Pre nego sto nastavite da citate, pogledajte odeljak SEKVENCIJALNE DATOTEKE NA KASETI, da biste se upoznali sa naredbama koje cemo dalje koristiti.

Naredba OPEN otvara kanal za komunikaciju sa I/O uredjajem. Zato ako zelite da pošaljete neku komandu stampacu, morate otvoriti datoteku preko koje cete komunicirati sa stampacem:

OPEN A,B,C

- A - broj kanala, bilo koji broj od 1 do 127
- B - broj uredjaja, za stampac je B = 4 ili B = 5
- C - sekundarna adresa

Preporuka je da za stampac koristite identicne brojeve za A i B parametre, pa je u tom slucaju za B = 4 i A = 4. Na stampacu se nalazi prekidač kojim reguliste da li ce stampac biti uredjaj broj 4 ili broj 5.

Sekundarna adresa odredjuje mod stampe:

C = 0 - velika slova/graficki simboli

C = 7 - mala slova/ velika slova

Kod MPS 802 sekundarnom adresom mozete definisati niz elemenata o čemu cemo posebno pisati.

Uz naredbu OPEN treba odmah dodati i komandu:

CMD A

Ovom komandom naredjujete C-64 da sve poruke i stampu preusmeri sa ekrana na stampac. Kada upotrebite ovu komandu, naredba PRINT više ne stampa na ekran, nego na stampac.

Ako ne upotrebite komandu CMD, tada se stampanje izvršava naredbom:

PRINT# A, podaci

Ova naredba je identicna naredbi PRINT, samo što se podaci stampaju na stampacu umesto na ekranu. Uz ovu naredbu potrebno je dati sledece napomene:

- u ovoj naredbi ne mogu biti kontrolni kodovi za stampanje na ekranu
- TAB naredba ne funkcioniše na stampacu, jer je ekran C-64 sirine 40 karaktera, a ovi stampaci su sirine 80 karaktera

Kao što morate otvoriti kanal za komunikaciju sa stampacem, naredbom OPEN, istom logikom morate i zatvoriti ovaj kanal, nakon završenog stampanja. Ovaj postupak morate pogotovo postovati ako ste komandom CMD preusmerili izlaz na stampac, a sada zelite da se vratite na ekran.

Zato uvek na završetku stampanja uradite sledece:

PRINT# A : CLOSE A

čime ste zatvorili kanal za komunikaciju sa stampacem i oslobodili ga za neku drugu primenu, do sledeceg stampanja.

Pogledajte nekoliko primera ispravnog i neispravnog rada sa štampačem:

ISPRAVNO

```
OPEN 4,4  
PRINT# 4, "ISPRAVNO"  
CLOSE 4
```

```
OPEN 4,4 : CMD 4  
PRINT "ISPRAVNO"  
PRINT# 4 : CLOSE 4
```

```
OPEN 4,4  
PRINT# 4, "ISPRAVNO"  
CMD 4: PRINT "STAMPANJE"  
PRINT# 4 : CLOSE 4
```

NEISPRAVNO

```
OPEN 4,4 : CMD 4  
PRINT "POGRESNO"  
CLOSE 4
```

```
OPEN 4,4  
PRINT# 4, "POGRESNO"  
CMD 4: PRINT "STAMPANJE"  
CLOSE 4
```

Štampanje listinga programa pisanih u BASIC-u možete dobiti na sledeći način:

```
OPEN 4,4 : CMD 4 : LIST : PRINT# 4 : CLOSE 4
```

Kontrolni kodovi i upravljanje štampačem

Svi štampači imaju manje ili više specijalnih mogućnosti, koje možete koristiti u programskom ili direktnom modu.

Da bismo saopštili štampaču, koju mogućnost želimo da koristimo, potrebno je naredbom PRINT (uz CMD naredbu) ili naredbom PRINT# poslati potrebne kontrolne ASCII kodove, pomoću naredbe CHR\$. Pogledajmo kako se to radi. Ako želimo da u zaglavlju listinga odštampamo naziv programa duplim slovima, uradićemo to na sledeći način:

```
10 OPEN 4,4 : CMD 4  
20 PRINT CHR$(14)  
30 PRINT "LISTING PROGRAMA"  
40 PRINT CHR$(15)  
50 LIST  
60 PRINT# 4 : CLOSE 4
```

Ako startujemo ovaj program, na štampaču ćemo dobiti sledeći listing:

LISTING

```

10 OPEN 4,4 : CMD 4
20 PRINT CHR$(14)
30 PRINT "LISTING PROGRAMA"
40 PRINT CHR$(15)
50 LIST
60 PRINT# 4 : CLOSE 4

```

Kao što vidite naslov listinga je odštampan sa karakterima duple širine, a listing programa sa normalnim karakterima. Kako smo to postigli?

Princip rada je jednostavan i primenljiv na većini štampača, zato pogledajmo program:

- 10 - otvaranje komunikacije sa štampačem i preusmeravanje kompletnog štampanja na štampač, naredbom CMD 4
- 20 - slanje kontrolnog koda 14, koji saopštava štampaču da sve što dalje sledi štampa sa karakterima duple širine
- 30 - štampanje naslova listinga slovima duple širine
- 40 - pošto želimo da listing štampano slovima normalne širine, moramo poslati štampaču kontrolni kod 15, (PRINT CHR\$(15)), koji ga vraća u normalni mod štampe
- 50 - program izlistavamo naredbom LIST
- 60 - nakon završetka štampanja, morate vratiti izlaz C-64 sa štampača na ekran, naredbom PRINT# 4 : CLOSE 4

Možemo rezimirati: da biste u odredjenom momentu koristili neku od mogućnosti štampača, potrebno je da mu prethodno pošaljete kontrolni kod naredbom PRINT CHR\$(broj kontrolnog koda) ili naredbom PRINT# 4, CHR\$(broj kontrolnog koda).

Pogledajmo sada koje mogućnosti imaju nabrojani CBM štampači, kao i njihove kontrolne kodove:

KOD	OBJASNJENJE
CHR\$(10)	pomeranje papira za jedan red
CHR\$(13)	CARRIAGE RETURN (kontrolni kod za kraj reda, automat.)
CHR\$(8)	prelazak u graficki mod
CHR\$(14)	štampanje karaktera duple širine
CHR\$(15)	štampanje karaktera normalne širine
CHR\$(16)	pozicioniranje štampe 0-79 karaktera (zamena za TAB)
CHR\$(27)	pozicioniranje štampe 0-479 pozicije kolika je maksimalna rezolucija u jednom štampanom redu

CHR\$(26) ponavljanje izabranog grafičkog simbola
 CHR\$(145) set karaktera: velika slova/grafički simboli
 CHR\$(17) set karaktera: mala slova/velika slova
 CHR\$(18) stampa u riverznom modu (bela slova na crnoj pozadini)
 CHR\$(146) otkacivanje riverznog moda i prelaz u normalni mod

**Tab. K - 2 Kontrolni kodovi za stampace
 S-6P-100VC, C-1525 i MPS 801**

Primenu kontrolnih kodova za izbor određene mogućnosti stampaca, prikazacemo na nekoliko primera.

Pozicioniranje stampe

Da biste odredili stampacu, od koje pozicije u odnosu na levu ivicu zelite da pocne stampanje, koristite komandni kod 16, CHR\$(16), iza koga mora slediti broj pozicije dat u znacima navoda:

```
10 OPEN 4,4 : REM OTVARANJE KANALA ZA STAMPAC
20 FOR I = 1 TO 4
30 PRIN# 4, "0123456789";
40 NEXT I
50 PRINT# 4, CHR$(10) : REM STAMPA SE JEDAN PRAZAN RED
60 PRINT# 4, CHR$(16) "7"; : REM STAMPA POCINJE OD 7. KOL.
70 PRINT# 4, "COMMODORE I/O"; : REM STAMPA TEKST
80 PRINT# 4, CHR$(16) "25"; : REM STAMPA POCINJE U ISTOM
  REDU ALI OD 25. KOLONE
90 PRINT# 4, "STAMPAC" : REM STAMPA TEKST
100 CLOSE 4
```

Ako startujete ovaj program na stampacu cete dobiti:

```
0123456789012345678901234567890123456789
      COMMODORE I/O      STAMPAC
```

Napominjemo da znaci interpukcije , i ; imaju u formatizovanju izvestaja na stampacu, istu funkciju kao i u PRINT naredbi za ekran, a to se odnosi i na naredbu SPC.

Pozicioniranje stampe u odnosu na apsolutnu adresu

Komandni kod 16 vam omogucava da pozicionirate stampu u odnosu na broj karaktera (0-79), koliko staje na list hartije. Komandni kod 27 vam dozvoljava da vrsite fino pozicioniranje. Kako na list hartije staje 80 karaktera, sirine 6 tacnica, to na formatu A4

imate 0 - 479 tačkica, kao maksimalnu rezoluciju koju daju ovi stampaci (to ne važi za MPS 802). Pozicioniranje teksta možete izvršiti na bilo koju tačkicu iz datog raspona.

Ako želite da štampa počne tačno na sredini stranice, stampacu morate poslati sledeću SERIJU komandnih kodova:

CHR\$(27) - kao znak da želite fino pozicioniranje
 CHR\$(16) - kao znak da sledi pozicija od koje želite štampu
 i zatim dva komandna koda koji definišu adresu pozicije
 CHR\$(LB) - za naš slučaj LB = 0
 CHR\$(HB) - za naš slučaj HB = 239

Adresu pozicioniranja proračunavate na već poznat način iz prethodnih tekstova o disku. Samo da vas podsetimo, ako želite štampu od 400-te pozicije tada je:

$$HB = \text{INT} (400/256) \text{ a } LB = 400 - HB * 256$$

Grafika na stampacima

Što se tiče grafike na ovim stampacima, nju možete programirati prelaskom u grafički mod pomoću komadnog koda 8.

Da biste programirali štampanje određene tačkice (pixel), u jednoj liniji morate znati rad sa binarnim brojevima. U uputstvu za vaše stampace imate date primere kako je moguće štampati željene grafičke simbole. Medjutim, grafima. U uputstvu za vaše stampace imate date primere kako je moguće štampati željene grafičke simbole. Medjutim, grafički mod je interesantan zbog mogućnosti prenošenja slike u grafici visoke rezolucije (kod C-64 to iznosi 320x200 tačkica na ekranu) na štampač. To znači da dijagrame, slike i crteže koje radite u ekranu visoke rezolucije možete odštampati na papir. Programiranje jednog takvog zadatka, podrazumeva detaljno poznavanje strukture ekrana visoke rezolucije i načina prenošenja na štampač.

U SIMONS BASIC-u imate naredbu COPY koja štampa ekran visoke rezolucije. Pomoću ove naredbe je odštampan YU karakter, koji je dat u poglavlju DEFINISANJE YU KARAKTERA (sl L - 1). Kada ste napravili neku sliku ili dijagram u ekranu visoke rezolucije, i vratili se u tekst ekran, dovoljno je da otkucate COPY i zatim pritisnete tipku RETURN, i štampač će vam odštampati kompletan ekran visoke rezolucije.

Ako pokušate da napravite program u BASIC-u, koji obavlja isti

zadatak kao naredba COPY u SIMONS BASIC-u, kopiranje ekrana visoke rezolucije na štampač, može potrajati i 20 minuta. Naredba COPY kopira ekran visoke rezolucije za nekoliko minuta.

Mora se napomenuti da naredba COPY ne funkcioniše sa drugim tipovima štampača kao što su EPSON ili GEMINI.

Štampač MPS 802

Od pomenuta četiri CBM štampača, najkvalitetniji je MPS 802 što se vidi iz datih karakteristika. Sve što smo prethodno govorili važi i za ovaj štampač. Međutim on ima još neke mogućnosti, koje možete dobiti definisanjem sekundarne adrese C, u OPEN A,B,C naredbi. MPS 802 ima veoma dobru osobinu, da pamti format štampe koji želite, i da ga prema potrebi koristite. Pogledajmo šta možemo definisati sekundarnom adresom:

SEKUNDARNA ADRESA	OBJASNJENJE
0	set velika slova/grafički simboli
1	štampa podatke prema prethodno definisanom formatu
2	pamti format štampe
3	definise broj linija po stranici
4	omogućava štampanje poruka o eventualnim greškama u radu sa štampačem
5	definisanje jednog simbola
6	postavljanje praznina između linija
7	set mala slova/velika slova
9	sprečava štampanje gresaka
10	resetuje štampač na stanje pri uključivanju

Tab. K - 3 Sekundarne adrese za štampač MPS 802

Formatizovanje izveštaja i tabela je zadatak koji MPS 802 obavlja sa lakoćom, za razliku od prva tri CBM štampača. Pokazimo to na primeru:

```
10 OPEN 2,4,2
20 OPEN 1,4,1
25 OPEN 4,4,0
30 PRINT# 2, "A AA AAA 9999"
40 B$ = CHR$(29)
50 PRINT# 1, "CBM";B$;"CBM";B$;"CBM";B$;1985.45
60 PRINT# 4, "CBM";B$;"CBM";B$;"CBM";B$;1985.45
70 CLOSE 2 : CLOSE 1 : CLOSE 4
```

- 10 - sekundarna adresa 2 u naredbi OPEN ukazuje stampacu da treba da zapamti format stampe, koji ce se definisati
- 20 - sekundarna adresa 1 u naredbi OPEN ukazuje stampacu, da sve ono sto dodje do stampaca kroz kanal 1 mora da stampa prema zadatom formatu
- 25 - pošto je sekundarna adresa jednaka 0 stampac ce podatke, koji dodju kroz kanal 4, odstampati tacno onako kako su poslali
- 30 - naredbom PRINT# 2 smo definisali format liniju, tako da su prva tri podatka tzv. string promenljive (alfanumericke), a poslednji podatak mora biti broj
- 40 - komandni kod za preskakanje blanko znaka
- 50 - pošto ova PRINT# 1 naredba šalje podatke kroz kanal 1, koji je pridružen sekundarnoj adresi 1, stampanje se obavlja prema zadatom formatu
- 60 - PRINT# 4 naredba šalje podatke stampacu kroz kanal 4, koji je pridružen sekundarnoj adresi 0, koja kazuje stampacu da podatke stampa na način kako je ukazano u samoj PRINT# 4 naredbi, bez korišćenja format linije
- 70 - zatvaraju s svi otvoreni kanali

Ako startujete ovaj program na stampacu MPS 802 cete dobiti:

```
C CB CBM 1985
CBM CBM CBM 1985.5
```

Kao što vidite MPS 802 je u stanju da pamti format liniju, po kojoj zelite da stampate vase tabele, i da kad je to potrebno, stampa sa format linijom ili bez nje. Format linija može biti duzine do 79 karaktera i u toku programa je možete menjati onoliko puta koliko vam je potrebno.

Namerno se zadržavamo na mogućnostima formatizovanja tabela, obzirom da je to redja osobina i kod boljih stampaca. Zato vam dajemo pregled FORMAT POLJA koja možete koristiti u vašim tabelama sa string i numerickim podacima.

Kao što vidite iz sledece tabele mogućnosti formatizovanja tabela kod MPS 802 stampaca su zaista velike. U definisanju formata tabele (princip dat u prethodnom programu) možete upotrebiti bilo koju od ponudjenih kombinacija format polja. Razmak izmedju kolona u tabeli pravite pomocu blanko znakova.

FORMAT POLJE	PODATAK	IZLAZ NA STAMPACU
AAAAAAA	COMMODORE	COMMODO
AAAAAAA	CBM	CBM
\$\$\$	99	\$99
\$9999	99	\$ 99
\$99.99	63	\$63.00
\$99.99-	-63	\$63.00-
\$99.99	-63	\$63.00
\$99.99-	63	\$63.00
S\$99.99	63	+\$63.00
ZZZZ	63	0063
ZZ.999	63	63.00
ZZZ.99	63	063.00
999.99	63	63.00
.99	63	.**
.99	0.001	.00
S.99	2.7E-02	+.02
Z.999	2.7E-02	0.027
Z.999-	-2.7E-02	0.027-

Tab. K - 4 Format polja koja mozete definisati na MPS 802

INTERFEJSI

Ako su vaši zahtevi za kvalitetom stampe visoki, i želite da nabavite znatno bolji (i skuplji stampac) prvi problem sa kojim se srecete je **KAKO POVEZATI STAMPAC SA C-64**. Za razliku od CBM stampaca (COMMODORE uvek okreće vodu na svoj mlin), ostali stampaci zahtevaju RS-232 ili CENTRONICS interfejs. Drugim recima, uz željeni stampac morate nabaviti i odgovarajući interfejs (od 100 MM pa na dalje).

Problem se ne završava samo na ovoj činjenici. Da bi napravio i svoju kompatibilnost (koja u osnovi ima komercijalni potez rodaje sopstvenih I/D uređaja), COMMODORE je smislio CBM ASCII od (neki ga zovu PET ASCII kod). Jednostavno receno, COMMODORE nije postovao standardni americki kod za razmenu informacija ASCII - American Standard Code for Information Interchange), pa sam nije dovoljno da kupite standardni RS-232 ili CENTRONICS interfejs, već morate da kupite ove interfejse specijalno namenjene za C-64. Kod RS-232 interfejasa je potrebno vrsiti i korekciju napona.

Problem leži u činjenici da svi dobri stampaci koriste standardni ASCII kod, pa kad povezujete ove stampace sa C-64 (bilo sa RS-232 bilo sa CENTRONICS interfejsom) interfejs mora pored redovnih obaveza, da vrši konverziju CBM ASCII kodova u standardni ASCII kodove. Za krajnjeg korisnika, ova činjenica ne mora biti značajna, ako mu nije žao maraka, koje je morao dati za ovu dodatnu opciju.

Iz prakse nam je poznato, da mnogi neiskusni korisnici ne obraćaju pažnju na činjenicu, da li je za dotični stampac potreban interfejs ili ne. Zato, od prodavca uvek tražite da vam uz stampac proda i odgovarajući interfejs za C-64.

USER PORT je mesto gde se RS-232 ili CENTRONICS interfejs najčešće priključuju (vidi poglavlje POVEZIVANJE PERIFERIJSKIH UREDJAJA SA C-64). Međutim, CENTRONICS interfejs je moguće priključiti i na SERIAL PORT, pa vam USER PORT ostaje slobodan, za priključenje MODEMA ili nekog drugog I/O uređaja.

RS-232 je jeftiniji interfejs od CENTRONICS-a, ali morate uzeti u obzir, da su mnogi tekst procesori pripremljeni za paralelni prenos podataka. Zato, preporučujemo da se ipak opredelite za stampac koji ima CENTRONICS priključak. Ako vam se čini da je cena za CENTRONICS interfejs previsoka, a elektronika vam nije slabija strana, pročitajte članak "CENTRONICS INTERFEJS ZA C-64", ("MOJ MIKRO", maj 1985., str. 26, Ciril Kraševac). Od materijala vam je potrebno:

- 24-pinski konektor za USER PORT
- 36-kontaktni CENTRONICS konektor za uključenje u stampac
- 1,5-2 m 11-žilnog kabla

Nakon vrlo jednostavnog spajanja, možete odmah vezati npr. EPSON FX 80 ili GEMINI-ja i koristiti ga pomoću VIZAWRITE tekst procesora, koji u sebi ima programiran protokol za rad sa CENTRONICS interfejsom. Ako želite da radite sa stampacem iz BASIC-a, morate ukucati kratak program (dat u članku), za inicijalizaciju rada C-64 za rad sa stampacem preko USER PORTA.

Za sve one koji imaju stampace sa RS-232 ili CENTRONICS interfejsom, a žele da koriste sjajan program za obradu podataka SUPERBASE 64, napisali smo posebno poglavlje SUPERBASE 64 I STAMPAC, obzirom na poseban protokol koji zahteva ovaj program u radu sa pomenutim stampacima.

TRIPPLER CENTRONICS interfejs

Svi listinzi u ovoj knjizi stampani su na stampacu EPSON FX 80+, preko TRIPPLER CENTRONICS interfejsa. Rad sa ovim interfejsom je krajnje jednostavan, a pruža izvanredne mogućnosti, i može da posluži kao ilustracija principa rada sa interfejsom. Najveća mu je prednost, što ste u mogućnosti da u listinzima definišete konverziju kontrolnih kodova u PRINT naredbama. Na CBM stampaci-
ma, kontrolne kodove u listinzima dobijate kao graficke, riverzne simbole, koji su često nečitki i nejasni. Ako pogledate bilo koji listing u ovoj knjizi primetićete da nema nikakvih grafickih simbola. Umesto toga u PRINT naredbama imate:

CLR - kao kontrolni kod za brisanje ekrana
 CUR DN - kontrolni kod za preskakanje jednog reda
 RVS ON - početak riverznog štampanja
 RVS OFF - prekid riverznog štampanja
 GRN - kontrolni kod za zelena slova i sl.

Na ovaj način listinzi su pregledni i nedvosmisleni za upotrebu. Konverziju kontrolnih kodova u listinzima, pretvaranje CBM u standardni ASCII kod, kao i još neke dodatne mogućnosti, ostvaruje ROM koji se nalazi u ovom interfejsu. TRIPPLER uzima podatke za štampu sa SERIAL PORT-a, a sopstveno napajanje stujom (5V), ostvaruju preko USER PORT-a. To znači da ovaj interfejs za svoj rad koristi oba PORT-a.

SEKUNDARNA ADRESA	KONVERZIJA KODOVA	CR+LF	M/V SLOVA	BEZ KONVERZIJE
BEZ	-	*	*	*
0	-	*	*	*
1	-	-	-	*
2	*	*	-	-
3	*	-	-	-
4	-	*	-	*
5	-	-	-	*
6	*	*	*	-
7	-	*	*	-

CR - CARRIAGE RETURN - kontrolni kod za kraj reda
 LF - LINE FEED - pomeranje štampača za jedan red
 M/V SLOVA - mala slova/velika slova

Tab. K - 5 Mogućnosti TRIPPLER CENTRONICS interfejsa u zavisnosti od sekundarne adrese

Protokol rada sa interfejsom regulisete kroz OPEN 4,4,C naredbu. Izborom vrednosti sekundarne adrese C, regulisite izbor mogucnosti TRIPPLER interfejsa u odnosu na stampac koji posedujete. Ako zelite konverziju kodova u listinzima, koristite sekundarne adrese 2, 3 ili 6.

Kod nekih stampaca (slucaj sa FX 80+), kod listanja programa, TRIPPLER ne salje LF kod, posle CR koda, pa bi se ceo listing stampao na jednoj liniji. Da bi se to prevazišlo prebacite mikroprekidac 2-4 u EPSONU za automatski LF na ON. Zbog toga kod izlistavanja programa morate koristiti sekundarne adrese 1, 3 ili 5, na primer:

```
OPEN 4,4,3 : CMD 4 : LIST : PRINT#4 : CLOSE 4
```

izlistava program na EPSON FX 80, sa konverzijom kontrolnih kodova.

Na kraju ovog odeljka o interfejsima, moramo reci i jednu napomenu koja nije nevažna. Iako su interfejsi dodatni trosak na ionako skupu I/O opremu, imajte na umu da u nekoj buducnosti mozda predjete na neki drugi kompjuter. RS-232 ili CENTRONICS interfejsi postaju standardna dodatna oprema vecine novih modela, sto znaci da vas stampac mozete odmah koristiti sa novim kompjuterom, sto se za CBM stampace ne moze reci.

EPSON FX 80+ I NJEMU SLICNI STAMPACI

O matricnim stampacima vise kategorije, kao sto su EPSON, GEMINI, LOGITEC i sl., dosta je pisano kako u stranim casopisima, tako i u domacim (pogledajte RACUNARE 1 i 2, MOJ MIKRO, maj 1985., i januar 1985.). U tim clancima su date opste osobine ovih stampaca, pa da ne bismo opterecivali knjigu vec dosta poznatim stvarima, nabrojimo osnovne osobine EPSON FX 80+, kao primer, sta jedan dobar stampac moze da uradi:

KONTROLA FORMATA

- definisanje razmaka izmedju linija u obliku:
1/8, 7/72, 1/6, n/216, n/72 inča
- definisanje duzine stranice u incima ili linijama
- preskakanje perforacije papira
- vertikalna tabulacija
- definisanje leve i desne margine
- horizontalna tabulacija

VARIJACIJE STAMPE

- tri osnovne veličine slova:
 - PICA 80 na format A4
 - ELITE 96 na format A4
 - CONDENSED 132 na format A4
- sedam modaliteta stampe

SET KARAKTERA: USA, FR, GDR, GB, D, S, I

DEFINISANJE KORISNICKOG SETA KARAKTERA

GRAFIKA

- normalne gustine
- dvostruka gustina
- četverostruka gustina

RAZNO

- programsko definisanje zvučnog signala za opomenu
- pomeranje glave za jedno slovo unapred
- inicijalizacija štampača
- detekcija kraja papira
- vraćanje glave za štampanje u osnovnu poziciju
- izbor pravca štampanja, u jednom ili u oba smera
- smanjenje brzine štampanja na polovinu (ako želite da smanjite buku pri štampanju)
- prebacivanje na rad sa uredjajem za dostavljanje pojedinačnih listova hartije

Pogled na ovaj pregled dela mogućnosti štampača EPSON FX 80+, dovoljno vam govori šta sve možete uraditi sa ovim štampačem. Upravljanje ovim štampačem se svodi na slanje određene ESCAPE sekvence kontrolnih kodova, kojim definišete štampaču, koju od njegovih funkcija želite da koristite. ESCAPE sekvenca podrazumeva prethodno slanje koda 27, pomoću koga štampač zna, da iza toga slede kontrolni kodovi. Pošto u ESCAPE sekvenci često imate upotrebu malih slova, potrebno je u OPEN 4,4,C naredbi, sekundarnom adresom C definisati da želite da radite u modu malih/velikih slova. U malom programu koji dajemo, je prikazano definisanje potrebnih parametara, pomoću kojih smo štampali sve listinge date u ovoj knjizi.

Sekundarne adrese 6 i 2 u programskim linijama 3 i 8 su odabrane u odnosu na TRIPPLER CENTRONICS interfejs (vidi tab. K - 5), što ne mora odgovarati vašem interfejsu, pa se morate konsultovati sa priručnikom koji ste dobili uz interfejs (obično vrlo kratka uputstva).

Najsigurnije je da za slanje kontrolnih kodova odaberete .set mala/velika slova:

```

1 ESC$=CHR$(27)
2 REM IZBOR SETA KARAKTERA MALA/VELIKA SLOVA
3 OPEN 4,4,6
4 REM LEVA MARGINA OD 15-TE KOLONE
5 PRINT# 4, ESC$; "1"; CHR$(15)
6 REM IZBOR ELITE SLOVA
7 PRINT# 4, ESC$; "!"; CHR$(1)
8 REM DESNA MARGINA NA 80-TOJ KOLONI
9 PRINT# 4, ESC$; "Q"; CHR$(80)
10 REM RAZMAK IZMEDJU LINIJA 1/8 INCA
11 PRINT# 4, ESC$; "0"
12 REM POJACAVANJE STAMPE DOUBLE-STRIKE MODALITETOM
13 PRINT# 4, ESC$; "6"
14 REM DETEKCIJA KRAJA PAPIRA ZA FORMAT A4
15 PRINT# 4, ESC$; "9"
16 CLOSE 4

```

Listing K - 1 Definisanje parametara stampaca EPSON FX 80+

Princip rada sa ostalim stampacima slicnim EPSON-u, kao sto je GEMINI, identican je prethodno objasnjenom principu rada. Jedino se u ESCAPE sekvencama upotrebljavaju neka druga slova ili brojke, ili su ESCAPE sekvence formirane na nesto drukciji nacin. Ako dozvolite jednu subjektivnu ocenu, smatramo da ova knjiga dovoljno govori sta moze da uradi jedan kvalitetan stampac, kao sro je EPSON FX 80+. Kompletan slog knjige uradjen je na WORD STAR tekst procesoru i stampan na stampacu EPSON FX 80+, a kvalitet i formu ocenite sami.

DEFINISANJE YU KARAKTERA

Mnogi korisnici, mogu i da prezale, ako nemaju YU karaktere na ekranu, ali zato bez ostatka zele da im stampani izveštaju budu sa YU karakterima (c,C,c,C,s,S,2,2). Problem lezi u cinjenici, da vecina stampaca koji su kod nas u upotrebi, na zalost, nemaju mogucnost definisanja karaktera. Kod takvih stampaca, jedino je resenje hardverska zamena ROM-a karaktera, sa EPROM cipom, koji bi pored ostalih karaktera sadrzao i YU karaktere. Kao sto smo vec ranije rekli, pitanje je ekonomske opravdanosti takvog zahvata, jer uz dodatnu investiciju i dalje imate stampac skromnih karakteristika, a za iste pare ste mogli dobiti znatno bolji stampac, kod koga se softverski mogu definisati YU karakteri.

Obzirom, da smo više puta bili u situaciji da i iskusnijim korisnicima, definišemo YU karaktere na njihovim stampacima, želimo da i ostalim čitaocima ove knjige, pomognemo da reše ovaj

Princip definisanja karaktera na stampacu, sastoji se iz tri glavne celine:

1. Kopiranje ROM karaktera u RAM memoriju stampaca
2. Zamena odgovarajućeg karaktera novim karakterom
3. Prebacivanje stampaca sa korišćenja ROM memorije na RAM memoriju

Sve tri celine se obavljaju programski, s tim što je kod nekih tipova GEMINI stampaca, potrebno prebaciti odgovarajući mikroprekidac, da bi stampac "znao" da treba da koristi karaktere iz RAM memorije. Kompletan postupak definisanja YU karaktera objasnićemo za stampace FX 80 (princip je isti i za ostale serije EPSON stampaca) i GEMINI SG 10/15

Kopiranje ROM karaktera u RAM memoriju stampaca

Da bi moderni stampaci obavljali sve one funkcije koje smo pomenuli oni u sebi sadrže:

- mikroprocesor
- ROM memoriju
- RAM memoriju
- interni časovnik (Timer) itd.

Drugim rečima, sve elemente koje inače ima i kompjuter. U ROM memoriji stampac ima zapisan operativni sistem, koji mu omogućava željeni rad. S druge strane, u ROM memoriji se nalaze zapisani oblici svih karakterà koje stampac može stampati. Kao što znate, ROM memorija se ne može menjati (sem kod EPROM cipova), pa stampac ima i određenu RAM memoriju.

RAM memorija ima više funkcija. Deo nje se koristi kao "medjuskladiste" - bafer u vezi između kompjutera i stampaca. Na taj način se znatno ubrzava štampanje. RAM služi i za pamćenje formata štampe koji želite. Međutim, ono što nas najviše interesuje, u RAM memoriju se mogu prepisati svi karakteri iz ROM memorije. Ova sjajna mogućnost, nam dozvoljava promenu nekih karaktera u karaktere koji nama trebaju.

TIP STAMPACA ESCAPE SEKVENCA ZA KOPIRANJE ROM-KARAKTERA U RAM

EPSON FX 80+ ESC\$: ":"; CHR\$(0); CHR\$(0); CHR\$(0)

GEMINI 10/15 ESC\$: "*"; CHR\$(0)

Napomena: ESC\$ = CHR\$(27)

Kao što vidite potrebno je kompjuteru poslati odgovarajući ESCAPE sekvencu i on će sve svoje karaktere iz ROM-a, prepisati u RAM. Sada smo spremni za najvažniji poduhvat, menjanje odgovarajućeg karaktera.

Zamena odgovarajućeg karaktera novim

TIP STAMPACA ESCAPE SEKVENCA MENJANJA ODGOVARAJUĆIH KARAKTERA

EPSON FX 80+ ESC\$: "&"; CHR\$(0); CHR\$(n); CHR\$(m);
CHR\$(a); CHR\$(P1); CHR\$(P2);...CHR\$(P11)

GEMINI 10/15 ESC\$: "*"; CHR\$(1); CHR\$(n); CHR\$(m);
CHR\$(a); CHR\$(P1); CHR\$(P2);...CHR\$(P11)

Ove ESCAPE sekvence izgledaju samo na prvi pogled komplikovane. Zato pogledajmo kako se određuju pojedini parametri u ovim ESCAPE sekvencama:

- n - predstavlja ASCII kod karaktera koji želimo da zamenimo. Pretpostavimo da @ hocete da zamenite sa YU karakterom C. U priručniku vašeg stampaca imate tabelu ASCII kodova svih karaktera. Tamo ćete videti da karakter @ ima ASCII kod 64, pa bi za n uzeli vrednost 64
- m - kod EPSON-a možete definisati niz ASCII simbola koje želite da zamenite. Zbog razmestaja simbola koji se zamenjuju YU karakterima, jednostavnije je menjati simbol po simbol, pa je zbog toga m = n
- a - pomoću ovog parametra možete regulisati tzv. proporcionalno stampanje i upotrebu 9-te iglice u glavi stampaca. Vrednost ovog parametra možete dobiti iz sledeće formule:

$$a = S * 128 + P * 16 + K \quad \text{gde je}$$

S = 0 - upotrebljava se 9 iglica u glavi stampaca, slovo se spušta ispod osnovnog nivoa ostalih slova

S = 1 - ne upotrebljava se 9 iglica

P - početna pozicija matrice za stampanje

K - krajnja pozicija matrice za stampanje

Pošto mi nećemo koristiti proporcionalno štampanje i 9-tu iglicu u glavi štampača (da bi definisani YU karakteri ostali u istoj liniji sa ostalim slovima) tada je:

$$a = 1 * 128 + P * 16 + K$$

P1 - P11 - su podaci koje dobijate iz matrice 8x11 u kojoj definišete vaše YU karaktere. Ovaj deo posla najčešće obeshrabruje manje iskusne korisnike, da se upuste u definisanje YU karaktera. Pogledajmo kako se to radi na primeru definisanja slova C (sl. L - 1). Potrebno je nacrtati matricu 8x11 polja, kao na datoj slici. Zatim, dolazi do izražaja vaš umetnički talenat, da u matricu ove veličine upišete odgovarajući YU karakter.

Jedino pravilo koje treba da poštuju, je da ne smete koristiti dve susedne tačke po horizontali. Između tačaka po horizontali mora biti bar jedno prazno polje. Kada ste završili zadatak, prostim sabiranjem dolazite do potrebnih P1-P11 podataka, gledano s leva u desno.

	0	60	66	0	194	0	194	0	66	26	0
128											
64											
32											
16											
8											
4											
2											
1											

PRITISNI RETURN ZA DALJE

Sl. L - 1 Matrica 8x11 za definisanje karaktera

Na primeru, našeg slova C, izračunavanje P1-P11 vršilo bi se na sledeći način:

P1 = 0
P2 = 4 + 18 + 16 + 32 = 60
P3 = 2 + 64 = 66
P4 = 0
P5 = 2 + 64 + 128 = 194
P6 = 0 (neko uzima P6 = 32)
P7 = 2 + 64 + 128 = 194
P8 = 0
P9 = 2 + 64 = 66
P10 = 4 + 32 = 36
P11 = 0

Kao sto vidite postupak nije previse komplikovan. Sada imamo sve komponente i parametre da karakter @ zamenimo sa karakterom C.

EPSON FX 80+

```
10 OPEN 4,4
20 ESC$=CHR$(27)
30 REM KOPIRANJE ROM KARAKTERA U RAM MEMORIJU
40 PRINT# 4, ESC$; ":"; CHR$(0); CHR$(0); CHR$(0)
50 REM ZAMENA KARAKTERA @ SA YU KARAKTEROM C
60 PRINT# 4, ESC$; "&"; CHR$(0); CHR$(64); CHR$(64);
70 PRINT# 4, CHR$(139);
80 FOR I = 1 TO 11
90 READ P : PRINT# 4, CHR$(P)
100 NEXT I
110 REM PREBACIVANJE NA KORISCENJE RAM KARAKTERA
120 PRINT# 4, ESC$; "%"; CHR$(1); CHR$(0)
130 CLOSE 4
140 DATA 0,60,66,0,194,0,194,0,66,36,0
```

GEMINI

```
10 OPEN 4,4
20 ESC$=CHR$(27)
30 REM KOPIRANJE ROM KARAKTERA U RAM MEMORIJU
40 PRINT# 4, ESC$; "*"; CHR$(0)
50 REM ZAMENA KARAKTERA @ SA YU KARAKTEROM C
60 PRINT# 4, ESC$; "*"; CHR$(1); CHR$(64); CHR$(64);
70 PRINT# 4, CHR$(139);
80 FOR I = 1 TO 11
90 READ P : PRINT# 4, CHR$(P)
100 NEXT I
110 REM PREBACIVANJE NA KORISCENJE RAM KARAKTERA
120 PRINT# 4, ESC$; "$"; CHR$(1)
130 CLOSE 4
140 DATA 0,60,66,0,194,0,194,0,66,36,0
```

Posle startovanja ovih programa vasi stampaci ce umesto @ karaktera stampati C karakter.

U programskoj liniji 120 smo koristili ESCAPE sekvencu, koja stampacu kazuje da li zelimo RAM ili ROM karaktere. To znaci da po potrebi mozete koristiti i @ karakter iz ROM memorije i C karakter iz RAM memorije.

TIP STAMPACA . ESCAPE SEKVENCA ZA KORISCENJE RAM-a ILI ROM-a

EPSON FX 80+ ESC\$; "%"; CHR\$(m); CHR\$(0)
GEMINI 10/15 ESC\$; "\$"; CHR\$(m)

m = 0 - ROM karakteri
m = 1 - definisani RAM karakteri

Na identican nacin mozete definisati i ostale YU karaktere. Ako zelite da napravite program koji ce odjednom definisati sve YU karaktere (uz pomoc EKRANSKOG EDITORA koga dajemo) bitno je napomenuti, da se kopiranje ROM karaktera u RAM izvršava samo na pocetku programa, a da posle ponavljate postupak definisanja novih karaktera, na nacin kako je prethodno objašnjeno.

Postoje odredjene preporuke koje ASCII simbole zameniti kojim YU karakterima. Medjutim, ove preporuke padaju u vodu ako zeleite da vam na istoj tipki na C-64 bude i veliko i malo slovo YU seta karaktera. Mi vam sada necemo davati nikakve preporuke, jer to zavisi od interfejsa koji upotrebljavate, a neki tekst procesori npr. VIZAWRITE, redefinisu odredjene tipke. Ako zelite da na mestu @ i <SHIFT>@ imate c i C, onda uradite sledece:

```
10 OPEN 4,4,7
20 PRINT# 4, "@"; "<SHIFT>@"
30 CLOSE 4
```

Na stampacu cete dobiti odgovarajuće simbole iz standardnog ASCII seta stampaca. Jedino sto vam ostaje da uradite je da pogledate u ASCII tabelu stampaca i odredite potrebne ASCII brojeve dobijenih simbola. Te brojeve zatim stavite za n parametar u datom programu za redefinisavanje i tako cete na mestu @ i <SHIFT>@ na C-64 dobijati na stampacu c i C. Identican postupak ponovite za ostale tipke za C-64 za koje zelite, da vam budu zamena za YU karaktere. Kraj standardnih oznaka na tastaturi mozete zalepiti nalepnice sa oznakama YU karaktera.

EDITOR KARAKTERA

Da bismo olakšali zadatak definisanja YU karaktera i smanjili potrebu za crtanjem odgovarajućih matrica (zasto to raditi kad već imamo kompjuter), napravili smo EDITOR KARAKTERA. Kao što je prethodno već rečeno, program je napisan u SIMONS BASIC-u i možete ga koristiti samo kada je ovaj BASIC u C-64. Program je sastavljen iz dve celine:

- DEMO programa
- EDITORA

DEMO program daje ekrane svih YU karaktera korišćenih u ovoj knjizi (c,C,c,C,s,S,z,Z) na način kako je to prikazano na sl.L-1. Ako ste zadovoljni oblicima YU karaktera koje smo mi kreirali, tada vam jedino ostaje da sa ekrana zabeležite potrebne podatke za parametre P1-P11 i da ih ubacite u DATA liniju 120. Pored toga potrebno je da definišete parametar n, koji određuje koji ASCII simbol želite da zamenite novim YU karakterom.

EDITDR je namenjen onim korisnicima koji žele da kreiraju svoje YU karaktere (ako im se naši ne sviđaju) ili da kreiraju neke specijalne matematičke simbole, koje stampac ne poseduje. EDITOR je napravljen za matricu 8x11, tako da se može koristiti za sve stampace čija matrica ulazi u ovaj opseg. Princip rada je jednostavan:

- 1 - iz glavnog MENIJA izaberite rad sa EDITOROM
- 2 - C-64 vas prebacuje u ekran visoke rezolucije, i na ekranu crta matricu identicnu sa sl. L-1 (uostalom ova slika je prekopirana na stampac upravo iz EDITORA naredbom COPY, upotrebljen je stampac SEIKOSHA GP 100VC koji podržava ovu naredbu)
- 3 - u donjem levom polju ce se pojaviti slovo x, kao zamena za kursor. Pomeranje "kursora" po matrici ostvarujete pomocu tipki za pomeranje kursora. Program ne dozvoljava izlazak "kursora" van polja matrice
- 4 - za polje koje želite da koristite za vaše slovo ili simbol, potrebno je pritisnuti I, i tada u tom polju ostaje slovo x u drugoj boji kao znak da to polje predstavlja deo simbola koji definišete
- 5 - kada završite formiranje (kreiranje) željenog simbola, pritisnite K kao oznaku za KRAJ i C-64 ce svako obeleženo polje, oznakom, popuniti bojom i na vrhu ekrana izbaciti odgovarajuće brojeve, koje koristite za parametre P1-P11 u ESCAPE sekvenci za definisanje karaktera.

NAPOMENA

- 1 - ako pritisnete 1, dok se "kursor" nalazi u gornjem desnom polju matrice, EDITOR automatski izvršava zadatak kao da ste pritisnuli slovo K za KRAJ.
- 2 - ako "kursor" vratite na polje koje ste obeležili da zapamtite, morate ponovo pritisnuti broj 1. Ako to ne uradite, EDITOR briše oznaku koju ste prethodno postavili i to polje ostaje prazno. Na taj način, možete korigovati željeni simbol, dok ne budete zadovoljni njegovim izgledom

Nadamo se da će vam EDITOR KARAKTERA sa svojim DEMO programom, uz objašnjenja data u prethodnom poglavlju, omogućiti da bez ikakvih problema sami definišete YU set karaktera za svoj štampač. Za sve eventualne probleme koje možete imati autori knjige će vam rado pomoći.

```

1 PROC MENI
2 PRINT CHR$(147);PRINT "      OSNOVNI MENI  "
3 PRINT " DEMO PROGRAM .....1"
4 PRINT " EDITOR KARAKTERA.....2"
5 PRINT " K R A J.....3"
6 INPUT " ZELJENU AKTIVNOST ";IZ$
7 N=VAL(IZ$);IF N<1 OR N>3 THEN EXEC MENI
8 IF N<>1 THEN CALL SLEDECI
9 EXEC DEMO
10 CLR:NRM:EXEC MENI
11 PROC SLEDECI
12 IF N=3 THEN PRINT " ";END
13 EXEC EDITOR
14 EXEC FARBA
15 TEXT 1,190,"PRITISNI RETURN ZA DALJE",1,1,10:PAUSE 220
16 CLR:NRM:EXEC MENI
17 PROC DEMD
18 HIRES 6,7
19 TEXT 20,40,"DEMD PROGRAM ZA YU-KARAKTERE",1,6,10
20 TEXT 30,100,"EPSON FX80+ I SEMINI SB 10/15",1,4,8
21 TEXT 40,170,"COMMODRE I/D C-1985",1,3,10:PAUSE 7
22 EXEC SLIKA
23 DATA 0,60,66,0,66,0,194,0,66,36,0
24 DATA 0,20,34,0,34,0,98,0,34,0,0
25 DATA 0,60,66,0,194,0,194,0,66,36,0
26 DATA 0,20,34,0,98,0,98,0,34,0,0
27 DATA 0,36,82,0,210,0,210,0,82,12,0
28 DATA 0,16,42,0,106,0,106,0,42,4,0
29 DATA 0,0,66,4,202,0,210,32,66,0,0
30 DATA 0,0,34,4,98,0,98,16,34,0,0
31 END PROC
32 PROC SLIKA
33 PRINT " ";RESTORE
34 FOR R=1 TO 0:XX=49:HIRES 6,7
35 EXEC MATRICA
36 FOR L=0 TO 10
37 READ D(L);P(L)=D(L);EXEC BINARNI
38 XX=XX+22:NEXT L
39 TEXT 1,190,"PRITISNI RETURN ZA DALJE",1,1,10
40 PAUSE 255:NEXT R:END PROC
41 PROC BINARNI
42 CSET 2:YY=170:FOR Y=0 TO 7
43 R1=MOD(D(L),2);D(L)=DIV(D(L),2)
44 IF R1=1 THEN PAINT XX-4,YY-4,1
45 YY=YY-20
46 NEXT Y:P$=STR$(P(L))
47 IF LEN(P$)=2 THEN RD=7:CALL PRVI
48 IF LEN(P$)=3 THEN RD=11:CALL PRVI
49 RD=15
50 PROC PRVI
51 TEXT XX-RD,YY,STR$(P(L)),1,1,6
52 END PROC
53 PROC MATRICA
54 HIRES 6,7:FOR H=39 TO 281 STEP 22
55 LINE H,20,H,100,1:NEXT H
56 FOR H=20 TO 100 STEP 20
57 LINE 39,H,281,H,1:NEXT H:F=165
58 FOR H=0 TO 7

```

```

58 FOR H=0 TO 7
59 B=2^H:TEXT 4,F,STR$(B),1,2,8:F=F-20:NEXT H
60 END PROC
61 PROC EDITOR
62 EXEC MATRICA
63 XX=49:L=0:T=0
64 REPEAT
65 ZZ=170:T=0
66 LOOP
67 TEXT XX-4,ZZ-4,"X",1,1,1:
68 PROC DRUGI
69 GET A$:IF A$="" THEN CALL DRUGI
70 IF A$=CHR$(145)THEN TEXT XX-4,ZZ-4,"X",2,1,1:R(T,L)=0:T=T+1:ZZ=ZZ-
20:
71 IF A$=CHR$(17)THEN TEXT XX-4,ZZ-4,"X",2,1,1:R(T,L)=0:T=T-1:ZZ=ZZ+
0:
72 IF T=-1 THEN T=7:ZZ=30
73 IF A$="1" THEN EXEC MEMORIJA
74 IF A$=CHR$(157)THEN EXEC MANJE
75 IF A$=CHR$(29) THEN EXEC VISE
76 IF A$="K" THEN EXEC BROJ
77 IF A$="K" THEN END PROC
78 EXIT IF T>7
79 END LOOP
80 EXEC BROJ
81 XX=XX+22:L=L+1
82 UNTIL L>10
83 END PROC
84 PROC MANJE
85 TEXT XX-4,ZZ-4,"X",2,1,1:XX=XX-22
86 IF XX<49 THEN XX=49:END PROC
87 R(T,L)=0:EXEC BROJ
88 L=L-1:END PRDC
89 PROC VISE
90 TEXT XX-4,ZZ-4,"X",2,1,1:XX=XX+22
91 IF XX>278 THEN XX=269:END PROC
92 R(T,L)=0: EXEC BROJ
93 L=L+1:END PROC
94 PROC BROJ
95 D(L)=1*R(0,L)+2*R(1,L)+4*R(2,L)+8*R(3,L)+16*R(4,L)
96 D(L)=D(L)+32*R(5,L)+64*R(6,L)+128*R(7,L)
97 P(L)=D(L):END PROC
98 PRDC FARBA
99 XX=49:FOR L=0 TO 10
100 P(L)=D(L):EXEC BINARNI
101 XX=XX+22:NEXT L
102 END PROC
103 PROC OKVIR
104 LINE 0,0,320,0,1:LINE 0,0,0,200,1
105 LINE 0,200,320,200,1:LINE 320,200,320,0,1
106 HIRES 6,7:CHAR 100,100,65,1,10:PAUSE 10
107 PRDC MEMORIJA
108 LOW COL 0,7,0
109 TEXT XX-4,ZZ-4,"X",1,1,1:R(T,L)=1:T=T+1:ZZ=ZZ-20:LOW COL 6,7,6
110 END PROC

```

Listing K - 1 Editor za definisanje simbola i YU karaktera
* EDITOR

SUPERBASE 64 I STAMPAC

Ovo poglavlje preporučujemo da pročitate samo ako imate prethodnog iskustva u radu sa SUPERBASE 64. Ako dosad niste radili sa ovim programom pročitate prethodno poglavlje SUPERBASE 64 koje detaljnije govori o ovom programu.

Iz prakse nam je poznato, da mnogi korisnici SUPERBASE 64 imaju odredjenih problema sa stampanjem ako ne koriste stampace specijalno namenjene za C-64 (SEIKOSHA GP 100 VC ili MPS serija). To znači, da ne znaju kako se dobijaju listinzi i izlazni rezultati na stampacu, koji je priključen na C-64 preko RS-232 ili CENTRONICS intrefejsa. Kao što ste mogli pročitati u poglavlju o stampacima, C-64 u redovnom radu šalje signale stampacu preko serijskog izlaza. Zato, ako ste stampac vezali preko interfejsa koji signale uzima sa USER PORT-a, to morate definisati na početku rada se SUPERBASE-64 da bi ovaj program "znao" gde da šalje signale za stampu.

Problem je u tome, što su korisnici navikli na tekst procesore kao što su VIZAWRITE ili EASYSCRIPT kod kojih se vrlo jednostavno definišu parametri stampaca, koji se koristi. Pošto je SUPERBASE tip programskog jezika, to definisanje parametara stampaca morate izvršiti u direktnom ili programskom modu, koristeći odredjene naredbe.

Najjednostavnije je to učiniti kroz START program, koji SUPERBASE uvek poziva na početku svoga rada, a koji se nalazi na programskoj disketi SUPERBASE programa. U ovom programu su pored ostalog, definisani i parametri stampaca. Medjutim, parametri su podešeni za rad sa CBM stampacima, ali vas zato niko ne sprečava da ovaj program prilagodite vašem stampacu i da ga kao takvog zapamtite. Tako ćete kasnije, kad god budete startovali SUPERBASE imati definisane parametre za vaš sistem. Pored ostalog u ovaj program možete uključiti i odredjene menije za rad ili čak i šifru za predstavljanje sistemu pre početka rada.

Objasnjenje programa START.P

100 BRKDN je naredba koja omogućava da prekinete izvršenje programa RUN/STOP tipkom, kad to želite. BRKOFF radi suprotno. GOSUB 400 - odlazak u potprogram na programsku liniju 400.

```

100 BRKON : GOSUB 400
110 DISPLAY @11,3 "-----"
120 DISPLAY @11,4 " " @+"SUPERBASE 64"
130 DISPLAY @11,5 "-----"
140 DISPLAY @1,17 "-----"
150 DISPLAY @1,19 "-----"
160 DISPLAY @0
170 ASK &16 @1,18 " ENTER DATABASE NAME :";A$
180 DATABASE A$,8,0
190 FILE
200 NEW
400 REM
410 LMARG 1:RMARG 80
420 PLEN 66:TLEN 60
430 PDEV 4:PDEF 0
440 LFEED 0:CONT 1
450 SPACE 0:ACROSS
460 SCREEN 0
470 RETURN

```

Listing J - 1 START.P program za inicijalizaciju

- 110-160 Slika na ekranu, koju dobijate nizom naredbi DISPLAY (zamena za naredbu PRINT, u SUPERBASE PRINT služi samo za štampač).
- 170 Unosite naziv baze podataka sa kojom želite da radite i smestate ga u promenljivu A\$.
- 180 Pozivanje postojeće baze podataka ili pozivanje postupaka za kreiranje nove baze podataka.
- 190 FILE - SUPERBASE vas pita sa kojom datotekom podataka želite da radite ili vam daje mogućnost da kreirate novu datoteku, ako ubacite ime datoteke koje nema na disketi.
- 200 Naredbom NEW se briše program.

- 410 LMARG 1 služi za postavljanje leve margine na štampaču na vrednost 1
 RMARG 80 služi za postavljanje desne margine na štampaču na vrednost 80
 Ova dva parametra možete menjati u zavisnosti od vaših želja i mogućnosti štampača sa kojim radite.
- 420 PLEN 66 služi za postavljanje dužine papira na 66 linija, pod pretpostavkom da radite sa A4 formatom. Ako je format manji, vi definišete manji broj linija i obrnuto.
 TLEN 60 služi za definisanje dužine teksta na jednoj stranici.
- 430 PDEV 4,7,0 je izuzetno značajna naredba, jer pomoću nje definišete parametre štampača sa kojim radite. Pogledajmo od čega zavisi taj broj.

NACIN PRIKLJUCENJA STAMPACA

	Serijski izlaz (IEEE)	RS-232	CENTRONICS
PDEV	4,7,0	X,Y,0	0

SERIJSKI IZLAZ (IEEE) - PDEV 4,7,0 za CBM štampače.

RS-232 INTERFEJS - ako koristite ovaj interfejs za vaš štampač, morate detaljno proučiti priručnik uz štampač da biste izračunali vrednost X,Y pomoću sledećih podataka:

- BAUD RATE** - mera za brzinu prenosa podataka, meri se u bitovima u sekundi

DATA WORD LENGHT - dužina jednog karaktera (bajta) koji se prenosi (7 ili 8 bitova)

NUMBER OF STOP BITS - broj stop bitova koji slede posle svakog podatka (bajta)

HANDSHAKE - princip komunikacije štampača i kompjutera

PARITY - metoda provere ispravnosti prenosa svakog bajta
- Za svaki od gore opisanih elemenata koristite datu tabelu. U priručniku za vaš štampač koji koristi RS-232 interfejs morate imati ove podatke. Pažljivo ih zabeležite i iz tabela upišite odgovarajuće vrednosti.
- Saberite dobijene vrednosti za BAUD RATE, DATA WORD LENGHT i BROJ STOP BITOVA i dobijena vrednost je parametar X u PDEV naredbi.

4. Saberite vrednosti za PARITY i HANDSHAKE i dobijenu vrednost unesite kao Y parametar u PDEV naredbi.

BAUD RATE	VREDNOST	DATA WORD LENGHT	VREDNOST	BRDJ STOP BITOVA	VREDNOST
50	1	8	0	1	0
75	2	7	32	2	128
110	3	6	64		
134.5	4	5	96		
150	5				
300	6	PARITY	VREDNOST	HANDSHAKE	VREDNOST
600	7	nema	0	0 - 3	0
1200	8	odd(neparan)	32	1 - X	1
1600	9	even(paran)	96		
2400	10	mark transmit	160		
3600	11	space transmit	224		
4800	12				

Tab. J - 1 Osnovni parametri za stampac sa RS-232 interfejsom

Uzmimo sledeci primer:

BAUD RATE	300	VREDNOST	6
DATA WORD LENGHT	8		0
BRDJ STOP BITOVA	1		0

		X =	6
PAITY - ODD (NEPARANAN)			32
HANDSHAKE 0 - 3			0

		Y =	32

Sada u START program u liniji broj 430 ubacite sledecu naredbu, PDEV 6,32,0.

Ako stampac ni sad ne bude radio, najbolje je da uzmete vrednost za BAUD RATE nesto manju od propisane, jer proizvođač ponekad da nesto vecu brzinu prenosa podataka nego sto stampac u realnim uslovima može da primi. Promena vrednosti brzine prenosa podataka (BAUD RATE) može da utice na HANDSHAKE (sporazumevanje C-64 i stampaca), pa vam autori SUPERBASE 64 preporučuju da jednostavno malo eksperimentisete sa Y vrednošću, dok ne dobijete ispravnu

stampu.

CENTRONIC INTERFEJS

Za stampac koji je priključen preko ovog interfejsa treba ubaciti PDEV 0, ako je interfejs priključen na USER PORT.

U programskoj liniji broj 430 imate još jednu naredbu tesno povezanu sa stampacem, koja određuje tip stampaca koji koristite, a to je PDEF naredba.

PDEF 0

CBM stampaci specijalno radjeni za C-64 (MPS serija, SEIKOSHA 6P-100 VC i sl.). Nije moguće dobiti graficke simbole, jer se pre stapanja svake linije šalje kontrolni kod CHR\$(17), kao znak da se vrši stapanje u modu mala/velika slova.

PDEF 1

EPSON MX 80(FX 80), GEMINI i sl. - pravi ASCII kodove.

PDEF 2

DAISYWHEEL (lepezni) stampaci: DIABLO, GUME i sl. - pravi ASCII kodovi.

PDEF 5

EPSON MX 80 - CBM ASCII kodove.

PDEF 6

DAISYWHEEL (lepezni) - CBM ASCII kodove.

Vase je samo da izaberete pravilnu PDEF naredbu u odnosu na stampac koji imate i da odaberete kakav tip ASCII kodova zelite da SUPERBASE šalje na vas stampac (prave ili CBM ASCII kodove).

440 LFEED - Ako ne zelite da vam stampac posle svakog odštampanog reda pomera hartiju za još jedan red, nego zelite stampu u narednom redu, onda definišite LFEED 0. Ako zelite obrnuto definišite LFEED 1.

CONT - Ovim parametrom definišete da li zelite kontinualnu stampu ili da stampac ceka da ubacite nov list hartije.

CONT 1 služi za kontinualnu štampu a CONT 0 za pojedinačne listove hartije.

- 450 SPACE - Odredjuje razmak izmedju dve linije koje se stampaju. SPACE 0 je normalni razmak izmedju dve linije. SPACE 1 je dupli razmak. Ovaj parametar mozete birati izmedju parametara 0 i 3 u zavisnosti od toga koliki razmak zelite izmedju linija.
- ACROSS - Daje listu slogova na ekranu ili stampacu bez imena polja u kojima se nalaze odgovarajuci sadržaji (suprotno od DOWN).
- 460 SCREEN - SUPERBASE odredjuje ekran 0 sa SCREEN 0 (moguće je imati ekrane od 0 do 3), kao ekran za komunikaciju.
- 470 RETURN - Naredba BASIC-a koja vraća izvršenje podprograma na liniju 110.

Kao što vidite kroz START program su definisani osnovni parametri za početak rada sa SUPERBASE 64, a pre svega parametri vezani za štampač. Rezimirajmo na kraju njihovu ulogu:

- LMARG L - definisanje leve margine
- RMARG R - definisanje desne margine, L<R, a zavise od štampača i seta karaktera koji koristite
- PLEN D - dužina stranice u linijama, za A4 format D=66
- TLEN T - dužina teksta na jednoj stranici, T<D
- LFEED F - F=1 prazan red posle svakog odštampanog reda
F=0 ne šalje prazan red posle CHR\$(13)
- CDNT S - S=0 kontinualna štampa
S=1 pauza posle svake odštampane strane za ubacivanje lista hartije
- SPACE R - razmak izmedju redova
R=0 normalan razmak
R=1 dupli razmak
R=2 tri razmaka
R=3 četiri razmaka
- PDEV X,Y,0 broj uredjaja i sekundarna adresa u zavisnosti od porta sa koga uzimate podatke. Ako imate potrebne podatke iz specifikacije štampača uradite po datom uputstvu, a ako nemate pokušajte sa sledecim vrednostima:

TIP KOMPJUTERA	PDEV	NACIN PRIKLJUCENJA STAMPACA		
		Serijski izlaz (IEEE)	U S E R RS-232	P O R T CENTRONICS
C-64	PDEV	4,7,0 ili 4,255,4(a)	2,8,0(b)	0(c)
700	PDEV	4,255,0	2,8,0(b)	0
8096	PDEV	4,255,0	4,255,0(d)	0

a - 4,7,0 : standardni CBM stampaci.
 4,255,0 : stampaci koji zahtevaju interfejs kao sto su
 (C-3022, C-4022, C-8023, C-6400)

b - pogledajte uputstvo dato u prethodnom tekstu ako imate
 zeljene podatke

c - CENTRONICS prikljucen na USER PORT

d - potreban RS-232 interfejs, koji se prikljucuje na serijski
 izlaz (IEEE)

PDEF T - tip stampaca koji koristite

T=0 CBM stampaci

T=1 EPSON MX-80, GEMINI i sl.-pravi ASCII kodovi

T=2 DAISYWHEEL stampaci - pravi ASCII kodovi

T=5 EPSON MX-80, GEMINI i sl.-CBM ASCII kodovi

T=6 DAISYWHEEL stampaci - CBM ASCII kodovi

Nadamo se da ce vam dati podaci i informacije o odnosu SUPERBASE
 i raznih tipova stampaca, omoguciti da iskoristite ovaj sjajan
 program.

RAD SA TEKST PROCESORIMA

U osnovi gledano TEKST PROCESORI rade isto sto i daktilografkinje, samo mnogo bolje. Glavna razlika medju njima je sto tekst procesor piše pomoću elektronskih impulsa, umesto slova na papiru. Tekst procesori to mogu da urade, jer postoje kompjuteri koji su programirani da vam omoguće da kucate tekst, ispravljate ga i tek kada ste u potpunosti zadovoljni, stampate ga na papir. Ovo su reći ARTHUR NAIMAN-a pisca poznate knjige INTRODUCTION TO WORDSTAR, za još poznatiji tekst procesor WORDSTAR. Ono sto je interesantno, to je da ARTHUR NAIMAN nije kompjuterski stručnjak, već naprotiv poznati pisac i žurnalista, sto dovoljno govori u prilog tezi da su tekst procesori kompjuterski programi namenjeni za krajnju (tzv. korisnicku upotrebu). Mnogo ljudi ima pogrešnu predstavu da je za kucanje teksta na kompjuteru potrebno neko programersko znanje, sto je daleko od istine. Uz malo osnovne kompjuterske škole i poznavanja daktilografije, svako može da nauči za par dana da radi i na najboljim tekst procesorima. U stvari, sto su tekst procesori bolji to je lakše s njima raditi i lakše je naučiti ih koristiti.

To vam otvara neslućene mogućnosti u radu sa tekstem, bilo da pišete knjige, naućne radove i sl.:

- menjanje reći ili celih fraza bilo gde u tekstu
- pomeranje celih blokova tekstova sa jednog mesta na drugo, to podrazumeva kako pomeranje, tako i kopiranje istih pasusa bez ponovne potrebe za kucanje
- taćno pronalaženje željene reći ili fraze u tekstu
- ćuvanje gotovih tekstova za kasniju upotrebu ili preformatizovanje
- provera sintakse za tekstove pisane na najpoznatijim svetskim jezicima
- povezivanje tekst procesora sa programima za baze podataka, omogućuje krajnje jednostavno pisanje zamornih cirkularnih pisama

Sve ove osobine poznatih tekst procesora, doskora su bile dostupne samo na većim sistemima, medjutim tekst procesori koji su se pojavili za C-64 kao sto su VIZAWRITE i EASYSCRIPT, mogu da se porede sa najpoznatijim tekst procesorima. Iz lićnog iskustva znamo, da oba ova tekst procesora mogu ravnopravno da se porede i sa takvim programom, kao sto je WORDSTAR. Zato nemojte zaliti za vremenom koji ste utrosili da savladate rad sa ovim procesorima jer ce vam se višestruko isplatiti, a naućiti rad sa njima mnogo je jednostavnije nego prosećno naućiti programirati u BASIC-u.

UPUTSTVO ZA KORIŠĆENJE PROGRAMA IZ KNJIGE I KASETE

Uputstvo za korišćenje programa sa kasete koju možete kupiti uz knjigu i programa koji su dati u knjizi je krajnje jednostavno. Svi listinzi programa koji se nalaze na kaseti označeni su * i uz zvezdicu je dato ime programa pod kojim je program snimljen na kasetu. Programi na kaseti su snimljeni prema redosledu obradivanja pojedinih poglavlja i prvenstveno su namenjeni olakšanju praćenja sadržaja ove knjige, kako ne biste morali gubiti značajan deo vremena na njihovom ukucavanju, pri čemu redovito dolazi do odredjenih grešaka.

Svi listinzi programa štampani su na EPSON FX 80+ štampaču preko TRIPPLER CENTRONICS interfejsa, koji vrši konverziju kontrolnih kodova za štampu na ekranu u okviru naredbe PRINT. Na taj način, je izbegnuto štampanje istih kodova u obliku grafičkih simbola, koji vas često mogu zbuniti. Da biste znali pravilno da koristite konvertovane kodove, dajemo vam uporedni pregled oznaka kontrolnih kodova u listinzima i kako možete do njih doći preko tastature:

OZNAKA KONTROLNOG KODA

TASTATURA

CLR	pritisnite	SHIFT i CLR/HOME
RVS ON	pritisnite	CTRL i 9
RVS OFF	pritisnite	CTRL i 0
CUR DN	pritisnite	CRSR (dole/gore)
CUR UP	pritisnite	SHIFT i CRSR (dole/gore)
CUR L	pritisnite	SHIFT i CRSR (levo/desno)
BLK	pritisnite	CTRL i 1
RED	pritisnite	CTRL i 3
GRN	pritisnite	CTRL i 6

S A D R Z A J

UVDDNA RAZMATRANJA.....	A - 3
Uvod.....	A - 3
Periferijski uređjaji.....	A - 4
Povezivanje periferijskih uređjaja sa C-64....	A - 7
KASETOFON.....	B - 11
Uputstvo za rad.....	B - 11
Učitavanje programa.....	B - 13
Snimanje programa.....	B - 25
Verifikacija snimljenih programa.....	B - 31
Programi za brzo učitavanje i snimanje.....	B - 31
Brzo traženje programa na kaseti.....	B - 32
DATOTEKE.....	C - 35
Organizacija podataka.....	C - 35
Sekvencijalne datoteke.....	C - 36
Sekvencijalne datoteke na kaseti.....	C - 36
Smještanje podataka.....	C - 40
Smještanje string podataka.....	C - 40
Postavljanje separatora između podataka.....	C - 43
Smještanje numeričkih podataka.....	C - 44
Učitavanje podataka INPUT# naredbom.....	C - 45
Učitavanje podataka GET# naredbom.....	C - 46
RAD SA DISKOM.....	D - 51
Disk jedinica.....	D - 51
Disketa.....	D - 55
Formatizacija nove diskete.....	D - 56
Komande za rad sa programima.....	D - 57
Komande diska.....	D - 62
DOS SUPDRT program.....	D - 68
SEKVENCIJALNE DATOTEKE NA DISKETI.....	E - 70
SLUCAJNE DATOTEKE.....	F - 72
Rad sa slučajnim datotekama.....	F - 73
Naredbe i komande.....	F - 74
BAM I DIREKTORIJA.....	G - 84
Format diskete.....	G - 84
Blok nula.....	G - 86
Blok jedan.....	G - 89
Put kroz kroz 18-tu stazu.....	G - 94
Komande SAVE,LOAD i SCRATCH.....	G - 97

S A D R Z A J

RELATIVNE DATOTEKE.....	H	- 102
Rad sa relativnim datotekama.....	H	- 102
Pravila za rad.....	H	- 104
Kreiranje datoteke.....	H	- 105
Dodavanje novih slogova datoteci.....	H	- 110
Ucitavanje datoteke.....	H	- 112
Izmena sadrzaja datoteke - azuriranje.....	H	- 114
BAZA PODATAKA.....	I	- 116
Struktura i sadrzaj programa.....	I	- 116
Princip tri datoteke.....	I	- 119
Primena programa.....	I	- 121
Objasnjenje programa.....	I	- 129
SUPERBASE 64.....	J	- 145
Rad sa programom SUPERBASE 64.....	J	- 145
Kreiranje baze podataka.....	J	- 148
Pretrazivanje datoteka.....	J	- 149
Azuriranje datoteka.....	J	- 149
Sortiranje datoteke.....	J	- 150
Prikaz datoteka na ekranu i stampacu.....	J	- 150
Formiranje programa.....	J	- 150
Pregled mogucnosti SUPERBASE 64.....	J	- 151
Rad sa bazom podataka.....	J	- 153
STAMPAC.....	K	- 155
Tipovi stampaca.....	K	- 155
CBM stampaci.....	K	- 156
Interfejsi.....	K	- 166
EPSON FX 80+ i njemu slicni stampaci.....	K	- 169
Definisiranje YU karaktera.....	K	- 171
Editor karaktera.....	K	- 177
SUPERBASE 64 I STAMPAC.....	L	- 181
RAD SA TEKST PROCESORIMA.....	M	- 188
UPUTSTVO ZA KORISCENJE PROGRAMA IZ KNJIGE I KASETE.N	N	- 189