

Glava 8.

Serijska organizacija datoteke

U datoteci sa serijskom organizacijom podataka, slogovi su smešteni jedan za drugim u sukcesivne memoriske lokacije od početka memoriskog prostora dodeljenog datoteci. Fizička struktura datoteke ne sadrži informaciju o vezama između slogova logičke strukture datoteke, niti postoji funkcionalna veza između vrednosti ključa sloga i adrese lokacije u koju je smešten. Redosled memorisanja slogova je, najčešće, definisan hronološkim redosledom njihovog nastanka. Slogovi u datoteci mogu, a ne moraju biti blokirani.

A_1	<table border="1"><tr><td>34</td><td>$p(S_1)$</td><td>07</td><td>$p(S_2)$</td><td>03</td><td>$p(S_3)$</td></tr></table>	34	$p(S_1)$	07	$p(S_2)$	03	$p(S_3)$	A_2	<table border="1"><tr><td>15</td><td>$p(S_4)$</td><td>19</td><td>$p(S_5)$</td><td>29</td><td>$p(S_6)$</td></tr></table>	15	$p(S_4)$	19	$p(S_5)$	29	$p(S_6)$
34	$p(S_1)$	07	$p(S_2)$	03	$p(S_3)$										
15	$p(S_4)$	19	$p(S_5)$	29	$p(S_6)$										
A_3	<table border="1"><tr><td>64</td><td>$p(S_7)$</td><td>43</td><td>$p(S_8)$</td><td>23</td><td>$p(S_9)$</td></tr></table>	64	$p(S_7)$	43	$p(S_8)$	23	$p(S_9)$	A_4	<table border="1"><tr><td>27</td><td>$p(S_{10})$</td><td>13</td><td>$p(S_{11})$</td><td>49</td><td>$p(S_{12})$</td></tr></table>	27	$p(S_{10})$	13	$p(S_{11})$	49	$p(S_{12})$
64	$p(S_7)$	43	$p(S_8)$	23	$p(S_9)$										
27	$p(S_{10})$	13	$p(S_{11})$	49	$p(S_{12})$										
A_5	<table border="1"><tr><td>25</td><td>$p(S_{13})$</td><td>*</td><td></td><td></td><td></td></tr></table>	25	$p(S_{13})$	*											
25	$p(S_{13})$	*													

Slika 8.1.

Primer 8.1. Na slici 8.1 prikazana je geometrijska reprezentacija fizičke strukture jedne male serijske datoteke od $N = 13$ slogova sa faktorom blokiranja $f = 3$. Slogovi su prikazani kao dvojke ($k(S_i), p(S_i)$), gde je $k(S_i)$ vrednost ključa, a $p(S_i)$ predstavlja konkretnizacije ostalih obeležja sloga S_i ($i = 1, \dots, 13$). Za vrednosti ključa $k(S_i)$ pretpostavljeno je da su numeričke i da uzimaju vrednosti iz skupa prirodnih brojeva. Rastuće vrednosti indeksa i ukazuju na redosled smeštanja slogova u datoteku. Slogovi su smešteni u fizičke blokove sa adresama A_1, A_2, A_3, A_4 i A_5 . Blok sa adresom A_5 je samo delimično popunjen. Adresa A_i se može shvatiti kao

relativna adresa, tj $A_i = i$. U polje ključa prve slobodne lokacije upisan je specijalni znak „, kao oznaka kraja datoteke. □

8.1 FORMIRANJE DATOTEKE

Serijska datoteka se, najčešće, generiše u postupku obuhvata podataka. Slogovi se formiraju prenosom podataka sa izvornih dokumenata na pogodan memorijski medijum. Izvorni dokumenti su, po pravilu, uređeni saglasno hronološkom redosledu svog nastanka. Slogovi se upisuju jedan za drugim u sukcesivne memorijске lokacije. Svaki novi slog se upisuje na kraj datoteke. Rezultat obuhvata podataka predstavlja neblokirana ili blokirana serijska datoteka.

Obuhvat podataka je proces, čiji je zadatak da obezbedi ispravne podatke na memorijskom medijumu. Taj proces sadrži više aktivnosti. Osnovnu aktivnost predstavlja upis podataka na memorijski medijum. Ovu aktivnost, najčešće, izvršava čovek (operator) koristeći tastaturu i ekran terminala ili personalnog računara. Sama aktivnost se izvršava pod kontrolom takozvanog format programa. Format program, između ostalog, sadrži:

- opis formata, odnosno dokumenta čija se slika sa praznim poljima javlja na ekranu,
- uputstva za pomeranje kursora od jednog do drugog polja formata na ekranu,
- fiksne nazive polja,
- opise polja formata i
- karakteristike polja formata.

Prilikom obuhvata, podaci se putem tastature unose u polja formata i, nakon provere ispravnosti, prenose na memorijski medijum. Za svaku vrstu dokumenta pravi se poseban format program.

Opisi polja predstavljaju obavezan deo format programa. Za svako polje formata, opis polja sadrži podatke o:

- tipu polja (samo alfa, samo numeričko, alfnumeričko, nenegativno numeričko sa m decimalnih mesta, datum i slično) i o
- maksimalnom broju znakova, koji se može uneti u polje.

Karakteristike polja predstavljaju bitan, ali neobavezan sadržaj format programa. Definisanjem karakteristika polja se određuje:

- način uređivanja sadržaja polja,
- mogućnost dupliciranja sadržaja polja i
- postupak specijalne kontrole sadržaja polja.

Način uređivanja sadržaja polja se odnosi na:

- naznaku granice polja (leva ili desna) uz koju treba da se poravna sadržaj polja, ako je broj unetih znakova manji od maksimalno dozvoljenog, kao i kojim znakom treba da se, automatski, popuni ostatak polja,
- postojanje (ili nepostojanje) obaveze da se u polje unese bar jedan znak i
- postojanje (ili nepostojanje) obaveze da se polje kompletno popuni.

Dupliciranje sadržaja polja se primenjuje u slučajevima kada niz suksesivnih dokumenata poseduje polja sa istim sadržajem, što povećava produktivnost operatera.

U postupke specijalne kontrole spadaju:

- kontrola po modulu,
- kontrola na dozvoljeni opseg vrednosti,
- provera da li se uneti podatak nalazi u tabeli (ugrađenoj u program) sa dozvoljenim ili nedozvoljenim vrednostima,
- provera kontrolnih zbirova i slično.

Kontrola po modulu se sprovodi samo na numeričkim poljima. Svakom broju od n cifara pridružuje se još jedna, kontrolna cifra, tako da polje ima dužinu $n + 1$ cifara. Kontrolna cifra se izračunava primenom posebnog algoritma na broj od n cifara. Prepostavka je da se broj sa kontrolnom cifrom nalazi na izvornom dokumentu. Operater unosi, putem tastature, $n + 1$ cifara u odgovarajuće polje na formatu. Program, primenom istog algoritma na prvih n cifara, ponovo izračunava kontrolnu cifru i uporeduje je sa unetom. Ako su iste, smatra se da je sadržaj polja ispravno obuhvaćen, inače program signalizira grešku. Sledeći pojednostavljen primer ilustruje primenu kontrole po modulu.

Primer 8.2. Neka je dat broj 863 i neka se kontrolna cifra izračunava po modulu 9. Tada je kontrolna cifra 8. Ako operater unese 3638, format program će otkriti grešku (cifra 8 je zamjenjena cifrom 3). Međutim, ako se pri unosu permutuju cifre 8 i 6, pa se unese 6838, kontrola po modulu 9 to neće otkriti. Zato se u praksi koriste takvi algoritmi za izračunavanje kontrolne cifre, koji sa većom pouzdanošću otkrivaju moguće greške. ☐

Dalje razmatranje postupaka specijalne kontrole izlazi izvan okvira ovog teksta. Može se naći u literaturi, na primer [DA].

Tek kada sadržaji svih polja formata zadovolje uslove, definisane u programu, od njih se formira slog serijske datoteke.

Primer 8.3. Na slici 8.2 prikazan je jedan format. To je pojednostavljena slika kartona sa osnovnim podacima o radniku u kadrovskoj evidenciji preduzeća. Ekran je, horizontalnim linijama, podeljen u tri dela. Prazna polja gornjeg i donjeg dela ekrana popunjava program, a prazna polja srednjeg dela ekrana popunjava operater putem tastature. Nazivi polja srednjeg dela ekrana predstavljaju obeležja tipa sloga datoteke, čiji se naziv javlja u gornjem delu ekrana.

Saglasno ranije iznetom, opis i karakteristike polja sa nazivom **RADNA JEDINICA** (identifikacioni broj radne jedinice), bi mogli da budu:

- tip polja: numerik,
- dužina polja: 6,
- način uređivanja:
 - poravnavanje uz desnu ivicu,
 - mora se uneti bar jedan znak,
 - dozvoljeno dupliciranje,
 - kontrola na dozvoljene vrednosti u tabeli radnih jedinica.

KADROVSKA EVIDENCIJA podaci o radnicima		
Oznaka datoteke		Redni broj sloga
Preduzeće		Radna jedinica
Matični broj radnika		
Prezime	Ime	Datum rođenja
Opština	Adresa	
Radno mesto		Broj bodova
Pozicija kursora	Pritisni ENTER za upis	Tip polja

Slika 8.2.

U slučaju polja sa nazivom *Matični broj radnika*, bi, između ostalog, mogla biti predviđena primena funkcije specijalne kontrole po modulu, a za polje sa nazivom *Broj bodova*, bi mogla biti predviđena kontrola na opseg dozvoljenih vrednosti. Granice tog opsega proističu iz Pravilnika o sistematizaciji radnih mesta u preduzeću, čija oznaka se unosi u polje sa nazivom *Preduzeće*.

Kada je reč o polju sa nazivom *Ime*, za njega bi se moglo specificirati samo da je tip alfa, dužina, recimo, 15 i da treba da bude levo poravnato. □

Obuhvat podataka se može odvijati u realnom vremenu ili naknadno. U realnom vremenu se izvršava na mestu i u trenutku nastanka podataka. Tada se paralelno sa unosom podataka na memorijski medijum može generisati i odgovarajući dokument. Naknadni obuhvat se sprovodi nakon određenog intervala vremena od nastanka podataka. Po pravilu ga realizuje osoba (operator) koja nije evidentirala izvorne podatke. Vrši se na osnovu manuelno izrađenih dokumenata. Bitnu aktivnost naknadnog obuhvata podataka predstavlja i takozvano verificiranje.

Verificiranje predstavlja postupak u kojem, po pravilu, drugi operator ponovo unosi jednom već unete podatke koristeći isti izvorni dokument. Cilj postupka je da se otkriju i koriguju eventualne greške prvog operatera. Verificiranje se sprovodi nad samo onim poljima, za koja nije mogla da se definiše neka funkcija specijalne kontrole. Indikator potrebe verificiranja predstavlja jedan od parametara karakteristike polja.

8.2 TRAŽENJE U SERIJSKI ORGANIZOVANOJ DATOTECI

Traženje slogova se u serijski organizovanoj datoteci vrši primenom metode linearnog traženja. Traženje svakog sloga počinje od početka datoteke i vrši se pristupanjem sukcesivno memorisanim blokovima, dok se u jednom od njih ne pronade slog sa vrednošću ključa jednako argumentu traženja. Traženje logički narednog sloga vrši se uvek od početka datoteke, jer fizička struktura serijske datoteke ne poseduje informaciju o adresi lokacije logički narednog sloga. Isti postupak se primenjuje i u slučaju traženja slučajno odabranog sloga, jer između vrednosti ključa i adrese lokacije u koju je slog smešten ne postoji funkcionalna veza. Ako se sa B obeleži broj blokova u datoteci, broj pristupa R_u datoteci, u slučaju uspešnog traženja jednog sloga, uzima celobrojne vrednosti iz intervala $[I, B]$, odnosno

$$I \leq R_u \leq B.$$

Broj blokova u datoteci izračunava se po formuli $B = \left\lceil \frac{N+1}{f} \right\rceil$, gde $\lceil x \rceil$ predstavlja minimalan ceo broj ne manji od x , a broju slogova N se dodaje I zbog oznake kraja datoteke. Srednji broj pristupa \bar{R}_u datoteci, u slučaju uspešnog traženja bilo logički narednog bilo slučajno odabranog sloga, dat je matematičkim očekivanjem

$$(8.1) \quad \bar{R}_u = \sum_{i=I}^B i p_i,$$

gde je i redni broj pristupa, a p_i verovatnoća da se traženi slog pronade u i -tom pristupu. Ako je verovatnoća traženja svakog sloga ista, tada je

$$p_i = \frac{f}{N} \quad (i = I, 2, \dots, B-I) \text{ i } p_B = \frac{N-f(B-I)}{N},$$

jer se u poslednjem bloku može nalaziti manje od f slogova. Zamenom vrednosti za verovatnoću traženja sloga u (8.1) dobija se

$$(8.2) \quad \bar{R}_u = \frac{B}{N} \left(N - \frac{f(B-I)}{2} \right).$$

U slučaju da $f \mid N$, tada je $B = \frac{N}{f} + I$, te se za očekivani broj pristupa pri uspešnom traženju dobija

$$\bar{R}_u = \frac{B}{2}.$$

Nešto veća efikasnost pri uspešnom traženju bi se mogla postići primenom algoritma linearног traženja, ako bi se traženje vršilo, ne uvek od početka datoteke, već od tekućeg sloga (sloga na kojem se završilo prethodno traženje) ka kraju datoteke, a onda, po potrebi, od tekućeg sloga ka početku datoteke.

Da bi se utvrdilo da se neki slog ne nalazi u datoteci, mora se pristupiti svim blokovima i uporediti argumenat traženja sa vrednošću ključa svakog sloga. Tek ako se traženi slog

ne pronade ni u poslednjem bloku datoteke, može se konstatovati da je traženje neuspešno, odnosno da se traženi slog ne nalazi u datoteci. Broj pristupa datoteci pri neuspešnom traženju iznosi

$$(8.3) \quad R_n = B.$$

Ako je zakon raspodele verovatnoće traženja slogova neravnomoran, može se uvesti takvo uređenje u serijsku datoteku, da se najčešće traženi slogovi smeste na početak datoteke. Tada je, pri uspešnom traženju, srednji broj pristupa dat formulom (8.1), a rezultantne vrednosti su manje od onih koje daje formula (8.2). Treba zapaziti da ovo uređenje datoteke nije povezano sa logičkom strukturom podataka, niti uspostavlja vezu vrednosti ključa sa adresom lokacije na memorijском medijumu.

Broj upoređivanja U_u argumenta traženja i vrednosti ključa pri uspešnom traženju sloga u serijskoj datoteci uzima celobrojne vrednosti iz intervala $[I, N]$, odnosno

$$I \leq U_u \leq N.$$

Srednji broj upoređivanja \bar{U}_u , pri uspešnom traženju, dat je matematičkim očekivanjem

$$\bar{U}_u = \sum_{i=1}^N i p_i$$

gde je i redni broj upoređenja, a p_i verovatnoća da se traženi slog nađe pri i -tom upoređenju. Ako su verovatnoće traženja svih slogova iste, tj. za

$$p_i = \frac{1}{N}, \quad (i = 1, \dots, N),$$

dobija se

$$\bar{U}_u = \frac{N+1}{2}.$$

Pri neuspešnom traženju je

$$U_n = N.$$

8.3 OBRADA SERIJSKI ORGANIZOVANE DATOTEKE

Datoteka sa serijskom organizacijom se može obrađivati i u režimu redosledne i u režimu direktnе obrade. Takođe, može se koristiti kao vodeća u režimu direktnе obrade. U posebnom slučaju, kada serijska datoteka sadrži ključ neke druge datoteke i kada je uređena saglasno neopadajućim vrednostima tog (stranog) ključa, serijska datoteka se može koristiti kao vodeća u redoslednoj obradi datoteke čiji ključ sadrži*).

Program koji izvršava redoslednu obradu serijske datoteke učitava sukcesivne slogove vodeće datoteke. Svaki naredni slog vodeće datoteke sadrži logički narednu vrednost ključa obrađivane serijske datoteke. Te vrednosti ključa program koristi kao argumente za traženje u serijskoj datoteci. Traženje se vrši opisanom metodom linearног traženja. U režimu direktnе

* U posmatranom slučaju datoteka je serijska, jer ne sadrži informaciju o vezama između slogova u svojoj logičkoj strukturi podataka datoteke.

$$\begin{aligned}
 & \sum_{i=1}^{B-1} \frac{1}{N} \cdot i + \frac{N - f(0-1) \cdot B}{N} \\
 & f(0-1) \cdot \frac{B}{2N} + f(0-1) \cdot \frac{N - f(0-1)}{2N} = \frac{B(0-1) \cdot f + 2N - 2f(0-1)}{2N} = \frac{B}{N}
 \end{aligned}$$

obrade, sukcesivni slogovi vodeće datoteke sadrže slučajno odabrane vrednosti ključa obrađivane serijske datoteke. Traženje je ponovo linearne. Kao što je već rečeno, efikasnosti traženj logički narednog i slučajno odabranog sloga u serijski organizovanoj datoteci su iste, je traženje svakog sloga počinje od prvog sloga datoteke.

Da bi se serijski organizovana datoteka obradila uz pomoć vodeće datoteke od N_v : $N_v^u + N_v^n$ slogova, gde je N_v^u broj slogova koji inicira uspešno, a N_v^n broj slogova koji inicira neuspešno traženje u obrađivanoj datoteci, potrebno je u proseku

$$\bar{R}_{uk} = N_v^u \bar{R}_u + N_v^n \bar{R}_n \quad \frac{N_v^u \cdot B}{2} + N_v^n \cdot B \quad \checkmark$$

pristupa obrađivanoj datoteci. Broj pristupa datoteci je isti, bez obzira da li se radi o redoslednoj ili direktnoj obradi.

Primer 8.4. Ako vodeća datoteka generiše sledeći niz vrednosti ključa (03, 06, 13, 19, 25, 29, 49, 55, 64) pri redoslednoj obradi serijske datoteke sa slike 8.1, sedam traženja će biti uspešno, a dva neuspešna. Ukupni broj pristupa datoteci, pri toj obradi, iznosiće $R_{uk} = 31$. Pošto je, na osnovu (8.2) $\bar{R}_u = 2,692$, a na osnovu (8.3) $\bar{R}_n = 5$, za $N_v^u = 7$ i $N_v^n = 2$, dobija se na osnovu (8.4), $\bar{R}_{uk} \approx 28,85$. □

Primer 8.5. Pri redoslednoj obradi serijske datoteke od $N = 10\,000$ slogova sa faktorom blokiranja $f = 10$ vodećom datotekom od $N_v = 9\,300$ slogova, gde je $N_v^u = 8400$ i $N_v^n = 900$, potrebno je izvršiti $\bar{R}_{uk} = 5100900$ pristupa serijskoj datoteci, jer je $\bar{R}_u = 500$ i $R_n = 1001$.

Neka je $\bar{t} = 10$ msek srednje vreme pristupa i prenosa fizički narednog bloka sa jednog diska u operativnu memoriju. Vreme $\bar{T} = \bar{R}_{uk} \bar{t}$, potrebno za razmenu podataka između datoteke i programa u posmatranoj obradi, iznosi $\bar{T} \approx 14,18$ časova. Pri tome, očekivano vreme jednog uspešnog traženja $\bar{t}_u = \bar{R}_u \bar{t}$, iznosi $\bar{t}_u \approx 5$ sek. Za neuspešno traženje jednog sloga potrebno vreme iznosi $\bar{t}_n \approx 10$ sek, jer je $\bar{t}_n = R_n \bar{t}$. □

8.4 AŽURIRANJE DATOTEKE SA SERIJSKOM ORGANIZACIJOM

Ažuriranje datoteke sa serijskom organizacijom je u principu jednostavno, ali zahteva veliki broj pristupa datoteci. Da bi se upisao novi slog, potrebno je prvo proveriti da li se takav slog već ne nalazi u datoteci. Upisu novog sloga prethodi neuspešno traženje. Sam novi slog se upisuje u prvu slobodnu lokaciju na kraju datoteke. Za upis novog sloga potrebno je

$$R_i = \begin{cases} R_n + 1, & \text{za } f \mid (N+1) \\ R_n + 2, & \text{za } f \nmid (N+1) \end{cases}$$

pristupa serijskoj datoteci, pri čemu je R_n pristupa potrebno za realizaciju neuspešnog traženja jedan pristup za upis novog sloga, ako f ne deli $(N+1)$, a jedan pristup za upis novog sloga

$$\left(\frac{f(B-1)}{2} + N - \frac{2f(B-1)}{2} \right) \leq \frac{f(B-1)}{2} \quad \checkmark$$

jedan pristup za upis oznake kraja datoteke u naredni blok memorijskog prostora dodeljenog datoteci, ako f deli $(N + 1)$.

Verovatnoća da f deli $(N + 1)$ je $1/f$, a verovatnoća da f ne deli $(N + 1)$ je $(f - 1)/f$, te se za očekivani broj pristupa \bar{R}_i , potreban za upis jednog novog sloga u serijsku datoteku dobija

$$\bar{R}_i = R_n + 1 + \frac{1}{f}.$$

Brisanje nekog postojećeg sloga u datoteci zahteva njegovo prethodno pronalaženje. Samo brisanje je najčešće logičko, tako da se slog sa izmenjenim statusnim poljem upisuje natrag u datoteku. Izmena sadržaja postojećeg sloga takođe zahteva njegovo prethodno pronalaženje i upis u datoteku. Za brisanje ili modifikaciju potrebno je, u proseku,

$$\bar{R}_d = \bar{R}_u + I$$

pristupa datoteci, gde je \bar{R}_u dato formulom (8.2).



8.5 OBLASTI PRIMENE I OCENE KARAKTERISTIKA SERIJSKE DATOTEKE

Veoma veliki broj pristupa datoteci, potreban za pronalaženje bilo logički narednog bilo slučajno odabranog sloga, ograničava primenu serijske organizacije, u opisanom smislu, na samo veoma male datoteke. Kod malih datoteka, druge vrste organizacije donose, u apsolutnom iznosu, malo poboljšanje efikasnosti obrade. S druge strane, serijska organizacija podataka u kombinaciji sa tzv. indeksnim strukturama, predstavlja datoteku, veoma pogodnu za direktnu obradu. Takva struktura predstavlja osnovnu fizičku strukturu relacionih baza podataka.

Pošto se datoteka sa serijskom organizacijom dobija kao rezultat obuhvata podataka, serijska datoteka predstavlja polaznu osnovu za izgradnju datoteka sa drugim vrstama organizacije podataka. Ova činjenica zaslужuje da joj se pokloni nešto veća pažnja. Naime, u praksi se, prilikom obuhvata podataka, najčešće ne kontroliše da li među slogovima postoje i takvi, koji imaju iste vrednosti ključa. Saglasno tome, rezultat obuhvata podataka se ne može smatrati datotekom u onom smislu kako je taj pojam definisan u glavi 1.

Saglasno rečenom, pri korišćenju "serijske" datoteke za formiranje datoteke sa drugom vrstom organizacije, potrebno je vršiti proveru postojanja slogova sa istom vrednošću ključa. Pri formiranju sekvenčialne datoteke, ta provera se vrši prilikom sortiranja. Kada se datoteka sa određenom vrstom organizacije formira direktno od serijske, ta provera bi trebalo da se vrši u okviru postupka formiranja. U cilju pojednostavljenja opisa postupaka formiranja datoteka sa određenom vrstom organizacije u narednim poglavljima, provera postojanja slogova sa istom vrednošću ključa se podrazumeva, ali se eksplicitno ne navodi.