



## **Serijska i sekvencijalna organizacija datoteke**

---

*Struktura, formiranje, traženje,  
obrada, ažuriranje, primena i  
ocena*

# Sadržaj

---

- Serijska organizacija datoteke
- Sekvencijalna organizacija datoteke

# Vrste organizacije datoteka

- Osnovne organizacije datoteka

DLS \ MLV	1	2a	3
A		sekvencijalna	serijska
B		spregnuta	
C			rasuta

# Serijska organizacija datoteke

---

- **Osnovna struktura**

- slogovi smešteni jedan za drugim
  - u sukcesivne memorijske lokacije
- fizička struktura ne sadrži informacije o vezama između slogova logičke strukture datoteke
- ne postoji veza između vrednosti ključa sloga i adrese lokacije u koju je smešten
- redosled memorisanja slogova najčešće prema hronološkom redosledu njihovog nastanka
- slogovi mogu, a i ne moraju, biti blokirani

# Serijska organizacija datoteke

- Primer - mala serijska datoteka -  $D_{ser}$

- slogova  $N = 13$
- faktor blokiranja  $f = 3$
- slogovi
  - dvojke  $(k(S_i), p(S_i))$
  - $k(S_i)$  - vrednost ključa
  - $p(S_i)$  - konkretizacija ostalih obeležja sloga  $S_i$  ( $i = 1, \dots, 13$ )
- fizički blokovi
  - adrese (relativne)  $A_i = i$ :  
 $A_1, A_2, A_3, A_4$  i  $A_5$
- oznaka kraja datoteke: \*

$A_1$					
	34	$p(S_1)$	07	$p(S_2)$	03 $p(S_3)$

$A_2$					
	15	$p(S_4)$	19	$p(S_5)$	29 $p(S_6)$

$A_3$					
	64	$p(S_7)$	43	$p(S_8)$	23 $p(S_9)$

$A_4$					
	27	$p(S_{10})$	13	$p(S_{11})$	49 $p(S_{12})$

$A_5$					
	25	$p(S_{13})$	*		

# Serijska organizacija datoteke

---

- **Formiranje serijske datoteke**

- serijska datoteka se generiše
  - najčešće u postupku obuhvata podataka
- slogovi
  - formiraju se prenosom podataka sa različitih izvora
    - izvorna dokumenta
    - uređaji i softveri za očitavanje vrednosti (u realnom vremenu)
  - upisuju se jedan za drugim u sukcesivne memorijske lokacije
  - svaki novi slog se upisuje na kraj datoteke
- rezultat obuhvata podataka
  - neblokirana ili
  - blokirana serijska datoteka

# Serijska organizacija datoteke

---

- **Formiranje serijske datoteke**

- obuhvat podataka

- proces sa zadatkom da obezbedi inicijalno memorisanje ispravnih podataka

- osnovna aktivnost: upis podataka na medijum, izvršava je

- čovek-operator, koristeći
      - program sa odgovarajućim UI-jem za formatiranje podataka i
      - odgovarajuće U/I uređaje, ili
    - specijalizovani softver sa odgovarajućim hardverskim uređajima

# Serijska organizacija datoteke

---

- **Formiranje serijske datoteke**
  - UI programa za obuhvat podataka – format program
    - opis formata dokumenta – raspored polja (layout), ekranska forma
    - pravila navigacije – pomeranja kursora između polja
    - opisi i formatiranje sadržaja polja
    - specijalne kontrole sadržaja polja
    - dozvoljene operacije nad sadržajima polja



# Serijska organizacija datoteke

- UI programa za obuhvat podataka
  - opis formata dokumenta – raspored polja (layout)

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

# Serijska organizacija datoteke

---

- UI programa za obuhvat podataka
  - pravila navigacije – pomeranja kursora između polja
    - mogućnost pristupa poljima putem miša ili tastature
    - redosled obilaska polja u navigaciji
      - putem tipke <TAB>, <ENTER>, ili automatskoj

# Serijska organizacija datoteke

---

- UI programa za obuhvat podataka
  - opisi i formatiranje sadržaja polja
    - fiksni tekst (naslov) i tekstuelno uputstvo sa opisom polja
    - tip polja (alfa, numeričko, alfanumeričko, datumsko)
    - način vizuelizacije sadržaja
      - tekstuelno polje, padajuća lista, combo-box, radio-grupa
      - polje skrivenog sadržaja
      - format maska za numeričke ili datumske vrednosti
      - zadavanje vizuelnih atributa polja na formi
  - maksimalni broj znakova koji je moguće uneti
  - način poravnavanja sadržaja polja (levo, desno, centrirano)
  - način preloma sadržaja tekstuelnih polja (bez preloma, prelom na jedan znak, prelom na celu reč)

# Serijska organizacija datoteke

---

- UI programa za obuhvat podataka
  - specijalne kontrole sadržaja polja
    - obaveznost unošenja barem jednog znaka u polje
    - obaveznost kompletnog popunjavanja sadržaja polja
    - kontrola na dozvoljeni opseg (domen) vrednosti
    - kontrola po modulu – samo za numerička polja
      - broju od  $n$  cifara pridružuje se jedna kontrolna cifra koja se izračunava primenom posebnog algoritma na broj od  $n$  cifara
    - proverava da li se uneti podatak nalazi u tabeli / listi dozvoljenih vrednosti

# Serijska organizacija datoteke

---

- UI programa za obuhvat podataka
  - dozvoljene operacije nad sadržajima polja
    - ručni unos sadržaja polja
    - modifikacija postojećeg sadržaja polja
    - brisanje sadržaja jednog, izabranih, ili svih polja
    - dupliciranje sadržaja jednog, izabranih, ili svih polja
      - u cilju povećava produktivnost operatera
      - kada se isti ili skoro isti sadržaj polja više puta ponavlja

# Serijska organizacija datoteke

---

- **Formiranje serijske datoteke**

- obuhvat podataka – vreme obavljanja

- **u realnom vremenu**

- na mestu i u trenutku nastanka podataka

- **u odloženom režimu (naknadno)**

- nakon određenog intervala vremena od nastanka podataka

- po pravilu ga realizuje operater koji nije evidentirao izvorne podatke, na osnovu manuelno izrađenih dokumenata

- verifikacija

- **verifikacija**

- postupak suštinske provere ispravnosti unetih podataka (ručni, ili automatizovan)

- npr. drugi operater ponovo unosi jednom već unete podatke koristeći isti izvorni dokument

- npr. korišćenje inteligentnih softverskih rešenja

# Serijska organizacija datoteke

- **Traženje sloga u serijskoj datoteci**

- traženje slučajno odabranog sloga
  - ne postoji funkcionalna veza između vrednosti ključa i adrese lokacije sloga
  - traženje logički narednog = traženje slučajno odabranog
- primena metode linearnog traženja
  - počinje od početka datoteke
  - pristupanje sukcesivno memorisanim blokovima i slogovima
- uspešno traženje, ukupan broj pristupa  $R_u$ :

$$1 \leq R_u \leq B$$

- neuspešno traženje, ukupan broj pristupa  $R_n$ :

$$R_n = B$$

- $B$  – ukupan broj blokova serijske datoteke

# Serijska organizacija datoteke

- **Traženje sloga u serijskoj datoteci**

- ukupan broj blokova datoteke:  $B = \left\lceil \frac{N+1}{f} \right\rceil$ 
  - $N$  – broj slogova
  - $f$  – faktor blokiranja
  - $+ 1$  - zbog specijalnog sloga sa oznakom kraja datoteke
- uspešno traženje, očekivani (srednji) broj pristupa
  - verovatnoća traženja bilo kog sloga datoteke je ista,  $1 / N$

$$\bar{R}_u = \frac{B}{N} \left( N - \frac{f(B-1)}{2} \right)$$

- kada  $f \mid N$ , tj.  $B = N / f + 1$ , ili kada je  $N \gg 1$ , tada:

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



# Serijska organizacija datoteke

- **Traženje sloga u serijskoj datoteci**

- uspešno traženje, ukupan broj upoređivanja argumenta traženja i vrednosti ključa  $U_u$ :

$$1 \leq U_u \leq N$$

- uspešno traženje, očekivani (srednji) broj upoređivanja argumenta traženja i vrednosti ključa
  - verovatnoća traženja bilo kog sloga datoteke je ista,  $1 / N$

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

- neuspešno traženje, ukupan broj upoređivanja argumenta traženja i vrednosti ključa  $U_n$ :

$$U_n = N$$

# Serijska organizacija datoteke

---

- **Obrada serijske datoteke**

- vrste

- direktna obrada
    - redosledna obrada
      - ukoliko se ide na sekvencijalni pristup slogovima u hronološkom redosledu

- može se koristiti kao vodeća u režimu direktne obrade

- može se koristiti kao vodeća u redoslednoj obradi datoteke čiji ključ sadrži

- kao svoj strani ključ
    - kada je uređena saglasno neopadajućim vrednostima tog stranog ključa

# Serijska organizacija datoteke

---

- **Obrada serijske datoteke**

- program koji vrši 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 se koriste kao argumenti za traženje u serijskoj datoteci metodom linearnog traženja
- u režimu direktne obrade
  - sukcesivni slogovi vodeće datoteke sadrže slučajno odabrane vrednosti ključa obrađivane serijske datoteke
  - traženje je, ponovo, linearno
- traženje logički narednog i slučajno odabranog sloga serijske datoteke
  - obavlja se identično, krećući od prvog sloga datoteke

# Serijska organizacija datoteke

- **Obrada serijske datoteke**

- putem vodeće datoteke od  $N_v = N_v^u + N_v^n$  slogova
  - $N_v^u$  slogova inicira uspešna traženja
  - $N_v^n$  slogova inicira neuspešna traženja
- inicira ukupan prosečni broj traženja

$$\bar{R}_{uk} = N_v^u \bar{R}_u + N_v^n \bar{R}_n$$

$$\bar{R}_{uk} \approx N_v^u \frac{B}{2} + N_v^n B$$

- broj pristupa se ne razlikuje za slučaj direktne i redosledne obrade

# Serijska organizacija datoteke

- Primer - mala serijska datoteka -  $D_{ser}$

- slogova  $N = 13$
- faktor blokiranja  $f = 3$

- Direktna obrada

- Sadržaj vodeće datoteke:
- 64, 21, 8, 3

- Redosledna obrada

- Sadržaj vodeće datoteke:
- 3, 8, 21, 64

$A_1$

34	$p(S_1)$	07	$p(S_2)$	03	$p(S_3)$

$A_2$

15	$p(S_4)$	19	$p(S_5)$	29	$p(S_6)$

$A_3$

64	$p(S_7)$	43	$p(S_8)$	23	$p(S_9)$

$A_4$

27	$p(S_{10})$	13	$p(S_{11})$	49	$p(S_{12})$

$A_5$

25	$p(S_{13})$	*			

# Serijska organizacija datoteke

- **Ažuriranje serijske datoteke**

- **upis novog sloga**

- u prvu slobodnu lokaciju na kraju datoteke
- mora mu prethoditi jedno neuspešno traženje
- jednostavan, ali zahteva veliki broj pristupa

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

- svaki  $f$ -ti put neophodno je proširiti datoteku novim blokom

# Serijska organizacija datoteke

---

- **Ažuriranje serijske datoteke**
  - **brisanje postojećeg sloga**
    - mora mu prethoditi jedno uspešno traženje
    - najčešće samo logičko – izmenom statusa aktuelnosti sloga
    - fizičko brisanje bi zahtevalo veliki broj pristupa
  - **modifikacija sadržaja postojećeg sloga**
    - mora mu prethoditi jedno uspešno traženje
  - **očekivani broj pristupa za**
    - logičko brisanje ili
    - modifikaciju sadržaja sloga

$$\bar{R}_d = \bar{R}_u + 1$$

# Serijska organizacija datoteke

---

- **Oblasti primene i ocena karakteristika**
  - pogodne kao male datoteke
    - kada mogu stati cele u OM
    - zbog veoma velikog broja pristupa potrebnog za pronalaženje logički narednog ili slučajno odabranog sloga
    - druge vrste organizacije donose samo mala poboljšanja u efikasnosti obrade malih datoteka
  - serijska organizacija podataka u kombinaciji sa indeksnim strukturama
    - veoma pogodna za direktnu obradu
    - osnovna fizička struktura relacionih baza podataka
  - serijska datoteka kao rezultat obuhvata podataka
    - polazna osnova za izgradnju datoteka sa drugim vrstama organizacije podataka



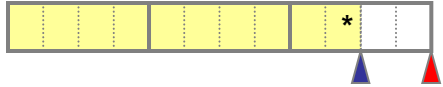

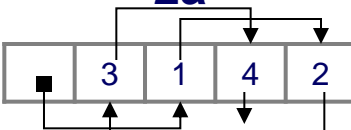

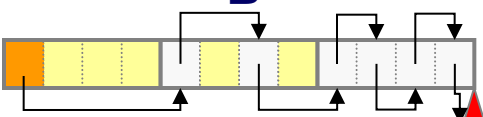
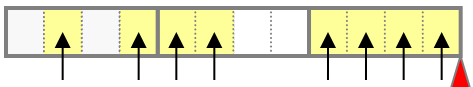
# Sadržaj

---

- Serijska organizacija datoteke
- Sekvencijalna organizacija datoteke

# Vrste organizacije datoteka

- Osnovne organizacije datoteka

<div>MLV \ DLS</div>	1	2a	3
<div>A</div> 	<div>1</div> 	<div>2a</div> 	<div>3</div> 
<div>B</div> 	<div>sekvencijalna</div>	<div>spregnuta</div>	
<div>C</div> 			<div>rasuta</div>

# Sekvencijalna organizacije datoteke

---

- **Osnovna struktura**

- slogovi su smešteni sukcesivno jedan za drugim
- logički susedni slogovi smeštaju se u fizički susedne lokacije
  - postoji informacija o vezama između slogova logičke strukture podataka datoteke, ugrađena u fizičku strukturu
  - realizovana kao linearna logička struktura podataka
    - smeštanjem sloga sa većom vrednošću ključa u lokaciju sa većom adresom
  - rastuće uređenje po vrednostima ključa  $\Rightarrow$  slog sa najmanjom vrednošću ključa smešta se u prvu lokaciju
- naziva se i fizički sekvencijalnom organizacijom

# Sekvencijalna organizacije datoteke

---

- **Osnovna struktura**

- veza između memorisanih vrednosti ključa  $k(S)$  i adresa lokacija
  - nije ugrađena u strukturu datoteke
  - ne predstavlja bilo kakvu matematičku funkciju
- slogovi se smeštaju u blokovima od po  $f (\geq 1)$  slogova
  - poželjno da faktor blokiranja  $f$  bude što veći
- savremeni OS (*Unix*) i programski jezici (C, C++, *Java*) podržavaju samo sekvencijalni način pristupa
  - korisnicima je ostavljeno da naprave svoje sopstvene sekvencijalne metode pristupa

# Sekvencijalna organizacije datoteke

## • Primer - mala sekvencijalna datoteka - $D_{sek}$

- slogova  $N = 13$
- faktor blokiranja  $f = 3$
- slogovi
  - isti sadržaj kao i  $D_{ser}$
  - dvojke  $(k(S_i), p(S_i))$
  - $k(S_i)$  - vrednost ključa
  - $p(S_i)$  - konkretizacija ostalih obeležja sloga  $S_i$  ( $i = 1, \dots, 13$ )
  - oznaka kraja datoteke: \*
  - indeksi  $i$  ( $i = 1, \dots, 13$ ) ukazuju na logički redosled smeštanja slogova

$A_1$

03	$p(S_1)$	07	$p(S_2)$	13	$p(S_3)$

$A_2$

15	$p(S_4)$	19	$p(S_5)$	23	$p(S_6)$

$A_3$

25	$p(S_7)$	27	$p(S_8)$	29	$p(S_9)$

$A_4$

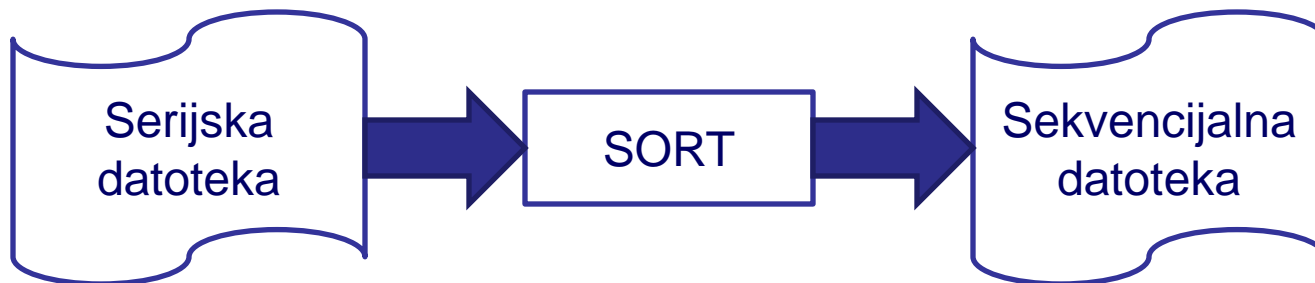
34	$p(S_{10})$	43	$p(S_{11})$	49	$p(S_{12})$

$A_5$

64	$p(S_{13})$	*			

# Sekvencijalna organizacije datoteke

- **Formiranje sekvencijalne datoteke**
  - najčešće sortiranjem serijske datoteke
  - saglasno rastućim ili opadajućim vrednostima ključa



# Sekvencijalna organizacije datoteke

---

- **Traženje sloga u sekvencijalnoj datoteci**

- logički narednog ili
- slučajno odabranog

- **traženje slučajno odabranog sloga**

- moguća primena metoda
  - linearnog traženja
  - binarnog traženja
- nema praktičnog smisla ako je datoteka velika i smeštena na eksterni memorijski uređaj
- ima praktičnog smisla ako je cela datoteka smeštena u OM
  - nju, u tom slučaju, može predstavljati
    - » neka linearna struktura nad skupom slogova ili
    - » blok neke druge datoteke, npr. indeks-sekvencijalne

# Sekvencijalna organizacije datoteke

---

- **Traženje sloga u sekvencijalnoj datoteci**
  - **traženje logički narednog sloga**
    - linearnom metodom traženja
    - počevši od prvog, fizički susedni blokovi se učitavaju u OM
    - u centralnoj jedinici se vrši upoređivanje argumenata traženja i vrednosti ključa sukcesivnih slogova dok se
      - traženi slog ne pronađe
      - argument traženja ne postane manji od vrednosti ključa sloga
      - ne dođe do kraja datoteke
    - traženje novog, logički narednog sloga, započinje od sloga na kojem se prethodno traženje zaustavilo
      - tekućeg sloga datoteke



# Sekvencijalna organizacije datoteke

- **Traženje sloga u sekvencijalnoj datoteci**
  - traženje logički narednog sloga
  - broj pristupa pri uspešnom i pri neuspešnom traženju

$$0 \leq R \leq B - i$$

- $i$  - redni broj tekućeg bloka u odnosu na početak
- broj poređenja argumenata traženja i vrednosti ključeva slogova, pri uspešnom i neuspešnom traženju

$$1 \leq U \leq N - i + 1$$

- $i$  - redni broj tekućeg sloga

# Sekvencijalna organizacije datoteke

- **Obrada sekvencijalne datoteke**

- redosledna
- direktna

- **direktna obrada**

- ima smisla ako je sekvencijalna datoteka mala, tako da se može smestiti u operativnu memoriju
- performanse obrade malo se razlikuju od performansi obrade serijske datoteke

$$\bar{R}_{uk} = N_v^u \bar{R}_u + N_v^n \bar{R}_n$$

$$\bar{R}_{uk} \approx N_v^u \frac{B}{2} + N_v^n \frac{B}{2}$$

# Sekvencijalna organizacije datoteke

- **Obrada sekvencijalne datoteke**
  - **vodeća datoteka u direktnoj i redoslednoj obradi**
    - česta upotreba
    - sukcesivno učitavanje fizički susednih slogova, počevši od prvog pa do poslednjeg
    - ukupan broj pristupa, kada se sekvencijalna datoteka koristi kao vodeća u obradi

$$R_{uk} = B = \left\lceil \frac{N+1}{f} \right\rceil$$

# Sekvencijalna organizacije datoteke

---

- **Obrada sekvencijalne datoteke**
  - **redosledna obrada**
    - iterativan proces
    - vodeća datoteka generiše logički naredne vrednosti ključa za traženje u obrađivanoj, sekvencijalnoj datoteci
  - svaki korak obrade = traženje logički narednog sloga
    - vrši se metodom linearnog traženja
  - svaki blok datoteke učitava se u OM samo jedanput
  - vodeća datoteka sadrži  $N_v$  ( $N_v \geq 1$ ) slogova
    - uključuje vrednost ključa veću ili jednaku najvećoj vrednosti ključa u obrađivanoj datoteci

# Sekvencijalna organizacije datoteke

- **Obrada sekvencijalne datoteke**

- redosledna obrada
- ukupan broj pristupa

$$R_{uk} = B$$

- srednji broj pristupa po jednom traženju

$$\bar{R} = \frac{B}{N_v}$$

- što je  $\bar{R}$  manji, obrada je efikasnija
- poželjniji veći faktor blokiranja  $f$  i veći broj traženja  $N_v$
- ukupan broj upoređivanja  $U \geq N + N_v$

- srednji broj upoređivanja  $\bar{U} \geq \frac{N}{N_v} + 1$

# Sekvencijalna organizacije datoteke

---

- **Ažuriranje sekvencijalne datoteke**
  - **upis novog sloga**
    - pronalaženje mesta upisa novog sloga – neuspešno traženje
      - lokacija sloga sa prvom većom vrednošću ključa od datog
    - pomeranje za jednu lokaciju udesno svih slogova sa vrednostima ključa većim od vrednosti ključa novog sloga
  - **brisanje postojećeg sloga**
    - prethodno pronalaženje sloga – uspešno traženje
    - pomeranje za jednu lokaciju ulevo svih slogova sa većom vrednošću ključa, ako se brisanje vrši fizički
  - **modifikacija sadržaja sloga**
    - prethodno pronalaženje sloga – uspešno traženje
  - upis i brisanje: ozbiljan problem ukupnog broja pristupa

# Sekvencijalna organizacije datoteke

- **Ažuriranje sekvencijalne datoteke**

- u režimu direktne obrade

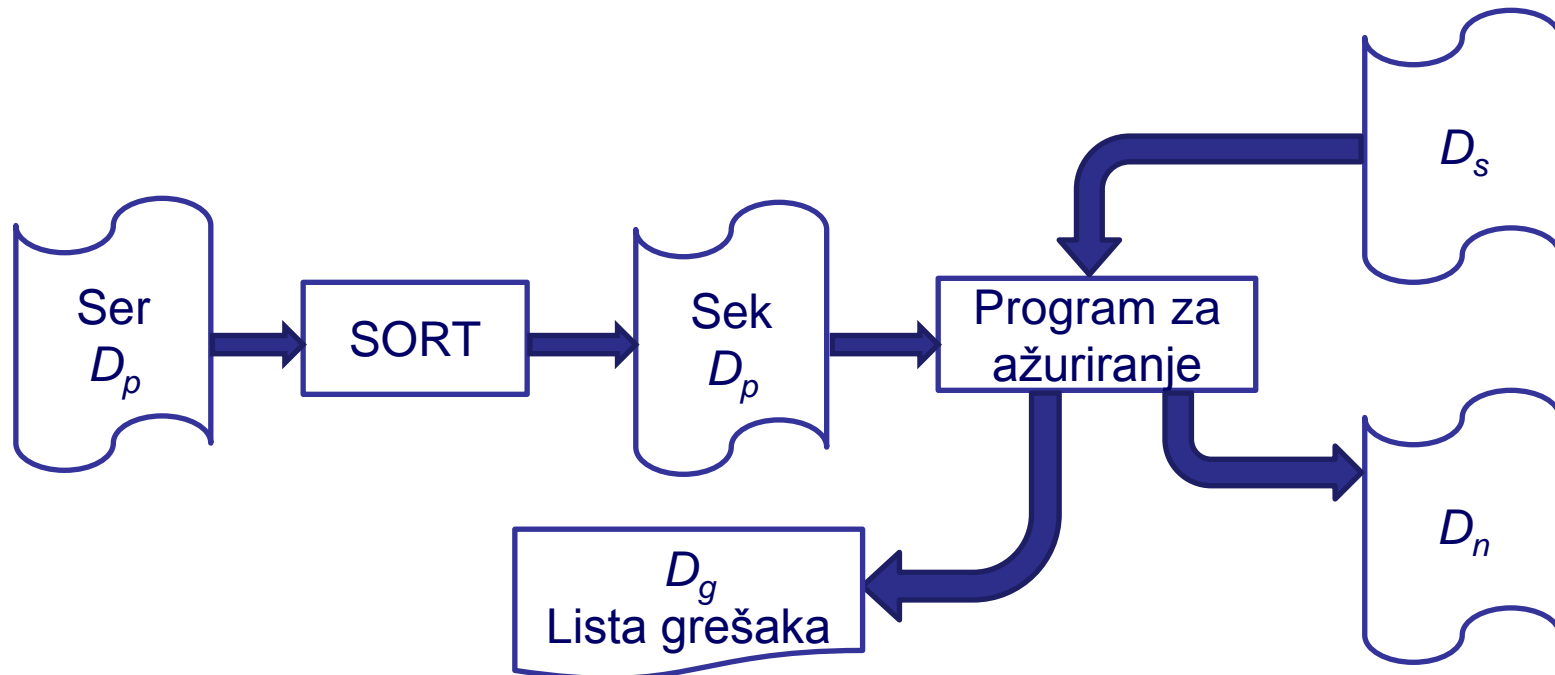
- u proseku, pomeranje polovine od ukupnog broja slogova za jednu lokaciju udesno (pri upisu) ili ulevo (pri brisanju) sloga
    - primenjuje se kada je kompletna datoteka smeštena u OM

- u režimu redosledne obrade

- poseban iterativni postupak
      - kreiranje potpuno nove datoteke, na osnovu postojeće
    - primeren kada se datoteka ne može kompletno smestiti u operativnu memoriju
    - datoteke i uloge u obradi
      - $D_s$  - obrađivana, ulazna (stara) sekvencijalna datoteka
      - $D_n$  - obrađena, izlazna (nova) sekvencijalna datoteka
      - $D_p$  - vodeća datoteka promena, serijska, ulazna
      - $D_g$  - datoteka grešaka, izlazna

# Sekvencijalna organizacije datoteke

- **Ažuriranje sekvencijalne datoteke**
  - u režimu redosledne obrade





# Sekvencijalna organizacije datoteke

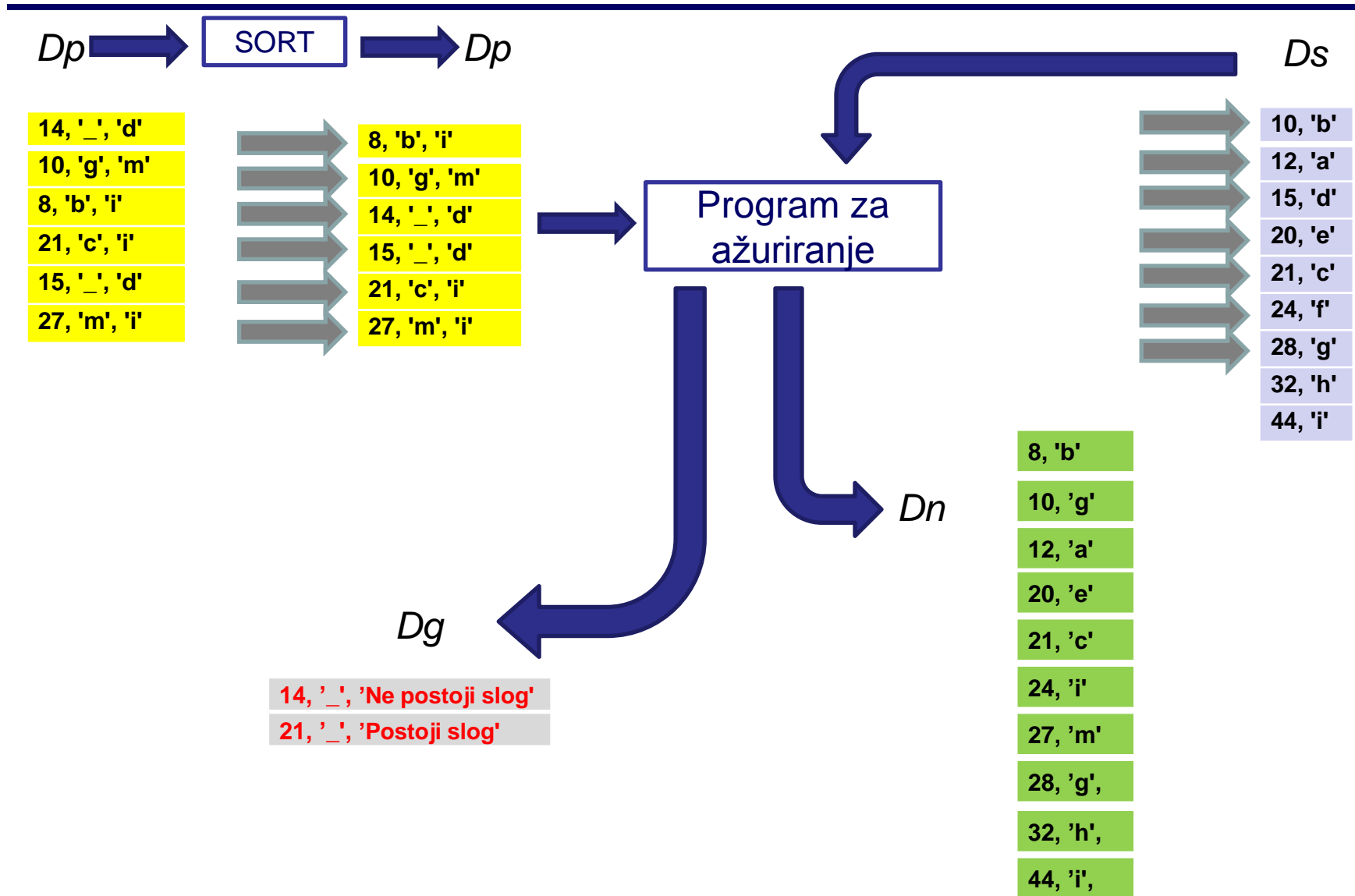
- **Ažuriranje sekvencijalne datoteke**
  - u režimu redosledne obrade
  - format sloga datoteke  $D_s$  i  $D_n$  identičan  $(k(S_i), p(S_i))$
  - format sloga datoteke promena  $D_p$ :  $(k(S_i), p_p(S_i), s_p(S_i))$ 
    - $s_p(S_i)$  - polje statusa izvršene operacije, moguće vrednosti:
      - $n$  – novi slog,  $m$  – podaci za modifikaciju,  $b$  – slog za brisanje
  - format sloga datoteke grešaka  $D_g$ :  $(k(S_i), p(S_i), s_g(S_i))$ 
    - $s_g(S_i)$  - polje opisa greške, moguće vrednosti ukazuju na:
      - pokušaj upisa već postojećeg sloga u datoteku
      - pokušaj brisanja ili modifikacije nepostojećeg sloga datoteke

# Sekvencijalna organizacije datoteke

---

- **Ažuriranje sekvencijalne datoteke**
  - u režimu redosledne obrade
    - sekvencijalni pristup sa učitavanjem slogova  $S_s(D_s)$  i  $S_p(D_p)$
    - upoređivanje vrednosti ključeva tekućih slogova
    - generisanje novih slogova  $S_n(D_n)$  na osnovu sadržaja tekućih slogova  $S_s$  i  $S_p$
    - upis slogova  $S_n$  u datoteku  $D_n$
  - dužina intervala između dva ažuriranja
    - određuje se tako da se tokom njega nakupi toliki broj promena koji bi opravdao pristupanje svim slogovima stare i generisanje nove datoteke
    - duži interval  $\Rightarrow$  veća efikasnost obrade, ali i duže vreme neusaglašenosti sadržaja datoteke sa realnim stanjem

# Sekvencijalna organizacije datoteke



# Sekvencijalna organizacije datoteke

- **Ažuriranje sekvencijalne datoteke**

- u režimu redosledne obrade

- datoteka promena  $D_p$  sadrži  $N_v = N_v^n + N_v^b + N_v^m$  slogova

- $N_v^n$  za upis,  $N_v^b$  za brisanje i  $N_v^m$  za modifikaciju

- i  $B_v$  blokova: 
$$B_v = \left\lceil \frac{N_v + 1}{f} \right\rceil$$

- postojeća datoteka  $D_s$  sadrži  $B_s$  blokova: 
$$B_s = \left\lceil \frac{N + 1}{f} \right\rceil$$

- nova datoteka  $D_n$  sadrži  $B_n$  blokova:

$$B_n = \left\lceil \frac{N + N_v^n - N_v^b + 1}{f} \right\rceil$$

# Sekvencijalna organizacije datoteke

---

- **Ažuriranje sekvencijalne datoteke**
  - u režimu redosledne obrade
  - srednji broj pristupa pri ažuriranju datoteke za jedno traženje logički narednog sloga

$$\bar{R} = \frac{B_v + B_s + B_n}{N_v}$$

# Sekvencijalna organizacije datoteke

---

- **Oblasti primene i ocena karakteristika**
  - prednosti
    - najpogodnija fizička organizacija za redoslednu obradu
    - ekonomično korišćenje memorijskog prostora
    - mogućnost korišćenja i magnetne trake i magnetnog diska, kao medijuma
  - nedostaci
    - nepogodnost za direktnu obradu
    - potreba sortiranja pri formiranju
    - relativno dugotrajan postupak ažuriranja

# Sekvencijalna organizacije datoteke

---

- **Oblasti primene i ocena karakteristika**
  - najpogodnija fizička organizacija za redoslednu obradu
    - režim redosledne obrade često se koristi u praksi, u paketnoj (batch) obradi podataka
    - posledica činjenice da su logički susedni slogovi smešteni u fizički susedne lokacije
    - učitavanjem jednog bloka u OM, pribavlja se  $f$  slogova koji najverovatnije učestvuju u narednim koracima obrade
    - poželjno je da  $f$  bude što veći
    - kada  $N_v \rightarrow N$ , tada  $\bar{R} \rightarrow 1 / f$ , te se s povećanjem  $f$  poboljšava efikasnost obrade

# Sadržaj

---

- Serijska organizacija datoteke
- Sekvencijalna organizacija datoteke



# Pitanja i komentari

---



## **Serijska i sekvencijalna organizacija datoteke**

---

*Struktura, formiranje, traženje,  
obrada, ažuriranje, primena i  
ocena*