

# Metode pristupa i organizacija datoteka

Usluge metoda pristupa i vrste organizacija datoteka

# Sadržaj

## ▶ Osnovna struktura datoteke

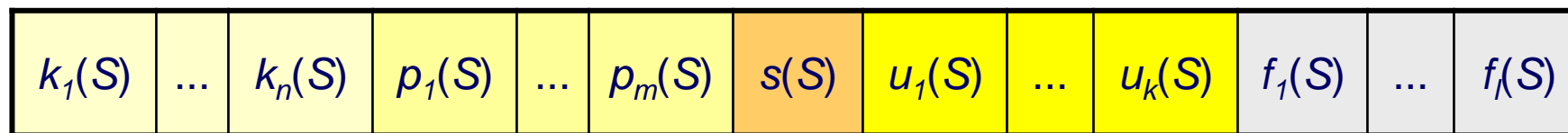
- ▶ Struktura datoteke kao niza blokova
- ▶ Metoda pristupa
- ▶ Parametri organizacije datoteka
- ▶ Vrste organizacije datoteka
- ▶ Opšte procedure nad datotekama
- ▶ Performanse obrade datoteke

# Osnovna struktura datoteke

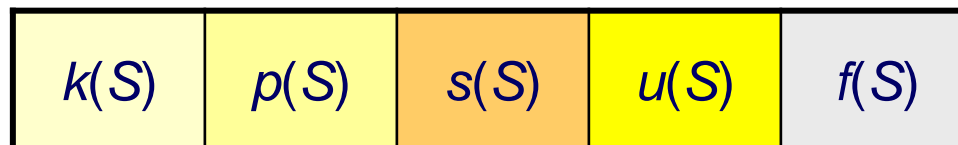
- ▶ Datoteka kao struktura slogova
  - ▶ organizovana nad tipom sloga
    - ▶ kao linearnom strukturom atributa
  - ▶ opšta struktura sloga datoteke kao FSP
    - ▶ uključuje podatke iz LSP i podatke o organizaciji FSP na eksternom memorijskom uređaju
      - ▶ informacija o strukturi tipa sloga
      - ▶ Informacija o vezama između slogova (opciono)
    - ▶ svaki slog predstavlja niz polja sa vrednostima atributa
- ▶ **format sloga**
  - ▶ pravila za strukturiranje i interpretaciju sadržaja sloga

# Osnovna struktura datoteke

## ► Opšta struktura sloga datoteke kao FSP



- $k_i(S)$  - polja vrednosti atributa primarnog ključa ( $n \geq 1$ )
  - $p_i(S)$  - polja vrednosti ostalih atributa ( $m \geq 0$ )
  - $s(S)$  - polje statusa sloga - indikator aktuelnosti sloga u LSP
  - $u_i(S)$  - polja pokazivača za memorisanje veza u LSP ( $k \geq 0$ )
  - $f_i(S)$  - kontrolna polja kod slogova varijabilne dužine ( $l \geq 0$ )
- Skraćena notacija



# Osnovna struktura datoteke

## ► Osnovna struktura datoteke

- $k(S)$  predstavlja jedinu obaveznu grupu polja
  - koja se sastoji od najmanje jednog polja
- često se posmatra kao linearna struktura slogova
  - uređena u rastućem ili opadajućem redosledu vrednosti primarnog ključa
- redosled polja u formatu sloga ne mora biti isti kao u opštoj strukturi sloga
  - dato je na prethodnom slajdu
  - pozicija kontrolnih polja uslovljena je njihovom semantikom

# Osnovna struktura datoteke

## ► Format polja sloga

- uslovljen specifikacijom domena odgovarajućeg atributa, odnosno primenjenim tipom podatka

## ► Vrste polja u slogovima

### ► polja konstantne dužine

- nije potrebno memorisati informaciju o granicama polja

### ► polja promenljive dužine

- potrebno je memorisati informaciju o granicama polja
  - koristi se kontrolno polje  $f_i(S)$
- tehnike
  - navođenjem aktuelne dužine polja u kontrolnom polju, neposredno ispred sadržaja polja
  - navođenjem specijalne oznake kraja polja u kontrolnom polju, neposredno nakon sadržaja polja

# Osnovna struktura datoteke

## ► Vrste slogova prema dužini

### ► slogovi konstantne dužine

- sva polja u svakom slogu su konstantne dužine
- nije potrebno memorisati informaciju o granicama sloga

### ► slogovi promenljive dužine

- postoji barem jedno polje promenljive dužine u slogu
- potrebno je memorisati informaciju o granicama sloga

- koristi se kontrolno polje  $f_i(S)$

### ► tehnike

- navođenjem aktuelne dužine sloga u kontrolnom polju, neposredno ispred ostalog sadržaja kompletnog sloga
- navođenjem specijalne oznake kraja sloga u kontrolnom polju, neposredno nakon ostalog sadržaja sloga
- uvođenjem posebne indeksne strukture sa rednim brojevima bajtova koji ukazuju na početke slogova

# Osnovna struktura datoteke

- ▶ Karakteristike vrsta slogova prema dužini
  - ▶ **slogovi konstantne dužine**
    - ▶ pojavljuju se u praksi
    - ▶ homogena struktura
    - ▶ jednostavnije pristupanje podacima i ažuriranje podataka
    - ▶ lakša i preciznija procena performansi obrade podataka
    - ▶ manja efikasnost upotrebe memorijskog prostora
  - ▶ **slogovi promenljive dužine**
    - ▶ izuzetno često se pojavljuju u praksi
    - ▶ nehomogena struktura
    - ▶ teže pristupanje podacima i ažuriranje podataka
    - ▶ teža i nepreciznija procena performansi obrade podataka
    - ▶ veća efikasnost upotrebe memorijskog prostora

# Osnovna struktura datoteke

- ▶ Vrste slogova prema ponavljanju vrednosti
  - ▶ **slogovi s ponavljajućim grupama**
    - ▶ višestruko pojavljivanje vrednosti atributa u jednom slogu
      - ▶ kada je dozvoljeno da jedna vrednost atributa bude predstavljena kao niz vrednosti istog tipa
      - ▶ primer
        - ▶ tip entiteta: *Student({StdID, ..., StdStrJez}, {StdID})*
        - ▶ tip podatka: *(StdStrJez : string(55))*
        - ▶ moguća vrednost, tri puta navedena (kao tri polja)
        - ▶ 'engleski', 'francuski', 'nemački'
    - ▶ moraju uvek biti slogovi varijabilne dužine
  - ▶ **slogovi bez ponavljajućih grupa**
    - ▶ nije dozvoljeno višestruko pojavljivanje vrednosti atributa
    - ▶ moguće je uvek projektovati tip sloga bez ponavljajućih grupa
      - ▶ primenom odgovarajućih projektantskih tehnika

# Osnovna struktura datoteke

- ▶ Polja pokazivača u strukturi sloga
  - ▶ predstavljaju adrese lokacija u memorijskom prostoru
- ▶ Vrste adresa lokacija
  - ▶ **apsolutna (mašinska) adresa**
    - ▶ strukturirana prema adresnom prostoru jedinice diska
  - ▶ **relativna adresa**
    - ▶ predstavlja redni broj lokacije
    - ▶ može biti praćen rednim brojem podlokacije
    - ▶ primeri
      - ▶ *rbr. bloka, rbr. bajta, (rbr. bloka, rbr. bajta u bloku)*
      - ▶ *rbr. sloga (u linearnoj strukturi), (rbr. bloka, rbr. sloga u bloku)*
  - ▶ **simbolička (asocijativna) adresa**
    - ▶ vrednost ključa

# Osnovna struktura datoteke

- ▶ Vrste adresa lokacija
  - ▶ **apsolutna (mašinska) adresa**
    - ▶ praktično se ne koristi u organizaciji datoteka
    - ▶ stvara zavisnost od fizičkih karakteristika uređaja
    - ▶ ne zahteva transformaciju
  - ▶ **relativna adresa**
    - ▶ vrlo često se koristi u organizaciji datoteka
    - ▶ obezbeđuje nezavisnost od fizičkih karakteristika uređaja
    - ▶ zahteva jednu ili više transformacija do apsolutne adrese
      - ▶ na nivou metode pristupa i/ili sistemskih poziva
  - ▶ **simbolička (asocijativna) adresa**
    - ▶ često se koristi u organizaciji datoteka
    - ▶ zahteva transformaciju u relativnu adresu
      - ▶ na nivou metode pristupa

# Sadržaj

- ▶ Osnovna struktura datoteke
- ▶ Struktura datoteke kao niza blokova
- ▶ Metoda pristupa
- ▶ Parametri organizacije datoteka
- ▶ Vrste organizacije datoteka
- ▶ Opšte procedure nad datotekama
- ▶ Performanse obrade datoteke

# Struktura datoteke kao niza blokova

## ▶ Blok (logički blok)

- ▶ kao organizaciona jedinica podataka
  - ▶ predstavlja niz slogova
  - ▶ ima konstantni kapacitet
    - ▶ tipične veličine: 2KB, 4KB, 8KB, 16KB
    - ▶ najčešće predstavlja celobrojni umnožak kapaciteta fizičkog bloka

## ▶ Odnos blok - fizički blok

- ▶ uobičajeno, jedan blok predstavlja niz od  $2^n$  ( $n \geq 0$ ) fizičkih blokova
  - ▶ nije nemoguće da kapacitet bloka bude
    - ▶ jednak kapacitetu fizičkog bloka ili
    - ▶ manji od kapaciteta fizičkog bloka

# Struktura datoteke kao niza blokova

## ► Opšta struktura (format) bloka

- zaglavlje bloka i niz slogova

$A_i$

Zaglavlje bloka	$A_i^1$	...	$A_i^j$	...	$A_i^f$
	$s_1$	...	$s_j$	...	$s_f$

- $A_i$  - adresa bloka (najčešće iskazana kao relativna)
- $A_i^j$  - realtivna adresa  $j$ -tog sloga u  $i$ -tom bloku ( $i, j$ )
- $f$  - faktor blokiranja - broj slogova u bloku
- zaglavlje bloka - neobavezna kategorija
  - obuhvata podatke vezane za FSP datoteke, npr.
    - različita polja pokazivača
    - broj slogova u bloku
    - indeks na početke slogova

# Struktura datoteke kao niza blokova

## ► Vrste blokova

### ► blokovi sa slogovima promenljive dužine

- više slogova može biti smešteno u jedan blok
- dozvoljeno je i da veličina jednog sloga premaši kapacitet bloka
  - tada se vrši ulančavanje blokova jednog sloga

### ► blokovi sa slogovima konstantne dužine

- homogena struktura bloka i datoteke
- svaki blok datoteke sadrži uvek isti broj slogova
- $f$  - faktor blokiranja datoteke
- $B$  - ukupan broj blokova datoteke
- $N$  - ukupan broj slogova u LSP datoteke

$$B = \lceil (N + x) / f \rceil$$

- $x$  - broj dodatno upotrebljenih specijalnih slogova

# Struktura datoteke kao niza blokova

## ► Proračun potrebnog kapaciteta datoteke

- moguć u slučaju primene blokova sa slogovima konstantne dužine

- $K_s$  - kapacitet sloga

- predstavlja zbir kapaciteta svih polja (konstantne dužine)

- $K_b$  - kapacitet bloka (unapred zadata konstanta)

- $K_z$  - kapacitet zaglavlja bloka (zavisi od organizacije)

$$f = \lfloor (K_b - K_z) / K_s \rfloor$$

- $K_d$  - kapacitet datoteke (zavisi od organizacije)

$$K_d = BK_b + W_d$$

- $W_d$  - kapacitet STD za datoteku

# Struktura datoteke kao niza blokova

- ▶ **Struktura datoteke kao niza blokova**
  - ▶ linearna struktura blokova datoteke
    - ▶ svaki blok datoteke obuhvata niz slogova datoteke
- ▶ **Strogo strukturirana datoteka**
  - ▶ strogo tipizovana datoteka sa pridruženom semantikom
  - ▶ organizovana kao struktura nad skupom slogova

# Struktura datoteke kao niza blokova

## ► **Zaglavlje datoteke**

- potrebno proširenje osnovne strukture datoteke
- uvodi se specijalni slog na početku datoteke
  - sa podacima o organizaciji datoteke i formatu bloka i sloga datoteke
  - uključuje podatke
    - broj slogova i/ili blokova u datoteci
    - dužina i format sloga
    - pozicija polja ključa u slogu
    - pokazivači na početke spregnutih struktura slogova ili blokova

# Struktura datoteke kao niza blokova

## ► Oznaka kraja datoteke

- načini označavanja kraja datoteke u osnovnoj strukturi
  - (A) uvođenjem specijalnog sloga za oznaku kraja datoteke
    - zapisuje se na kraju strukture, iza poslednjeg sloga u LSP, u prvu slobodnu lokaciju memorijskog prostora datoteke
  - (B) uvođenjem specijalne oznake kraja u polje pokazivača
    - navodi se u polju pokazivača logički narednog sloga  $u(S)$ , kod poslednjeg sloga u LSP
  - (C) vođenjem posebne evidencije zauzetosti prostora
    - memorisanjem npr. broja zauzetih lokacija u prostoru dodeljenom datoteci
    - u pomoćnoj strukturi podataka
  - (D) kraj datoteke je kraj prostora dodeljenog datoteci
    - ne uvodi se poseban mehanizam za označavanje kraja datoteke

# Sadržaj

- ▶ Osnovna struktura datoteke
- ▶ Struktura datoteke kao niza blokova
- ▶ Metoda pristupa
- ▶ Parametri organizacije datoteka
- ▶ Vrste organizacije datoteka
- ▶ Opšte procedure nad datotekama
- ▶ Performanse obrade datoteke

# Metoda pristupa

- ▶ Paket programa (rutina) za podršku usluga visokog nivoa
  - ▶ upravljanje strogo strukturiranim datotekama
    - ▶ upravljanje organizacijom i memorisanjem polja, slogova i blokova
      - ▶ različite vrste usluga u pogledu organizacije sloga i bloka
  - ▶ upravljanje baferima metode pristupa
    - ▶ viši nivo "baferisanja" u odnosu na nivo sistemskih bafera

# Metoda pristupa

- ▶ Paket programa (rutina) za podršku usluga visokog nivoa
  - ▶ podrška različitih vrsta organizacije datoteka
    - ▶ podrška različitih načina memorisanja logičkih veza i adresiranja (pristupa podacima)
      - ▶ u strogo strukturiranim datotekama
  - ▶ vođenje brige o kategorijama
    - ▶ zaglavlje datoteke
    - ▶ početak i kraj datoteke
    - ▶ tekući pokazivač, indikator aktuelnosti
      - ▶ iskazan kao relativna adresa bloka ili sloga na kojem se sprovodi operacija (ili alternativno, na kojem je izvršena prethodna operacija)
  - ▶ podrška izgradnje specijalnih pomoćnih struktura za poboljšanje efikasnosti obrade podataka

# Metoda pristupa

- ▶ Paket programa (rutina) za podršku usluga visokog nivoa
  - ▶ podrška opštih postupaka upravljanja sadržajem datoteka
    - ▶ kreiranje, traženje, pretraživanje, ažuriranje i reorganizacija
  - ▶ koristi ili uključuje usluge niskog nivoa izabranog OS
    - ▶ u zavisnosti od mesta i načina implementacije metode pristupa
- ▶ obezbeđuje nezavisnost aplikativnog programa od usluga niskog nivoa OS
  - ▶ obezbeđuje preslikavanje strogo strukturirane datoteke u FSP niza fizičkih blokova
  - ▶ obezbeđuje transformacije relativne adrese sloga ili bloka datoteke u relativnu adresu bajta ili fizičkog bloka

# Metoda pristupa

- ▶ Upravljanje strogo strukturiranim datotekama
  - ▶ podrška organizacije slogova i polja
    - ▶ konstantne dužine
    - ▶ promenljive dužine
  - ▶ podrška različitih (alfanumeričkih, datumskih, bitorijentisanih, multimedijalnih) tipova podataka
  - ▶ podrška različitih kodnih rasporeda
  - ▶ konverzije podataka
    - ▶ iz tipa podatka programske promenljive u tip podatka atributa datoteke i obratno
    - ▶ iz tipa podatka atributa strogo strukturirane datoteke u niz bajtova i obratno

# Metoda pristupa

- ▶ Upravljanje strogo strukturiranim datotekama
  - ▶ usluge razmene podataka sa aplikativnim programom
    - ▶ **na nivou sloga**
      - ▶ grupisanje slogova u blokove pri upisu podataka
      - ▶ rastavljanje bloka na slogove pri čitanju podataka
      - ▶ održavanje tekućeg pokazivača kao relativne adrese sloga
        - ▶ *redni broj sloga u datoteci*
      - ▶ i njegova transformacija u oblik
        - ▶ *(redni broj bloka u datoteci, redni broj sloga u bloku)*
    - ▶ **na nivou bloka**
      - ▶ razmena sadržaja kompletnih logičkih blokova između aplikativnog programa i datoteke
      - ▶ održavanje tekućeg pokazivača kao relativne adrese bloka, u obliku
        - ▶ *redni broj bloka u datoteci*

# Metoda pristupa

- ▶ Upravljanje strogo strukturiranim datotekama
  - ▶ usluge pristupa podacima iz aplikativnih programa
    - ▶ **sekvencijalni (redosledni) pristup**
      - ▶ slogovima ili blokovima datoteke
        - ▶ najčešće slogovima datoteke
      - ▶ automatski održavaju (inkrementiraju) vrednost tekućeg pokazivača
        - ▶ pri operacijama učitavanja / zapisivanja podataka
    - ▶ **direktni pristup**
      - ▶ slogovima ili blokovima datoteke
      - ▶ zahtevaju eksplicitno zadavanje vrednosti tekućeg pokazivača
        - ▶ rednog broja sloga ili bloka datoteke
        - ▶ pri operacijama pozicioniranja
    - ▶ **dinamički (kombinovani) pristup**
      - ▶ kombinacija direktnog i sekvencijalnog pristupa

# Metoda pristupa

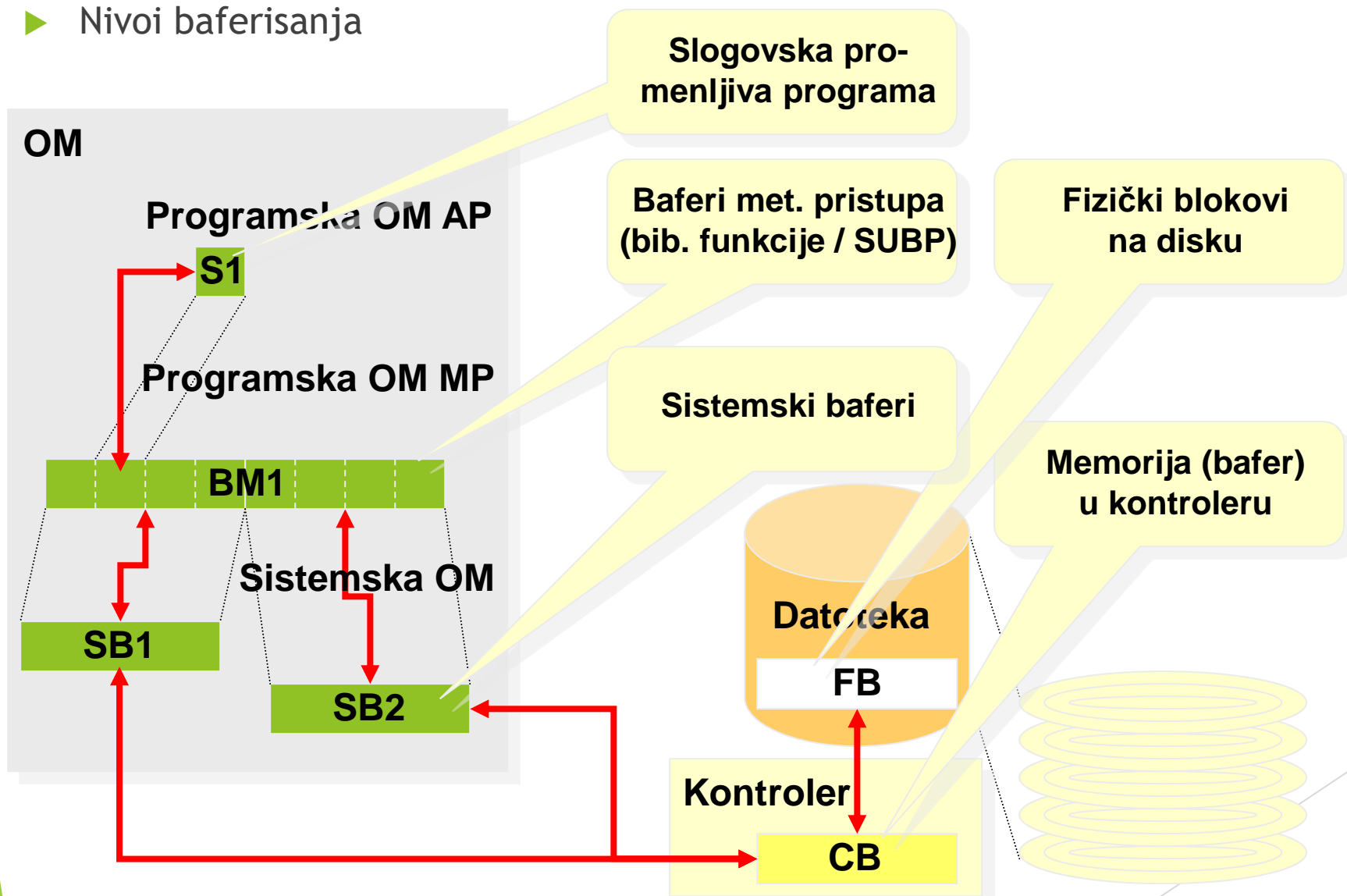
- ▶ Upravljanje strogo strukturiranim datotekama
  - ▶ pozivi rutina metode pristupa
    - ▶ otvaranje i zatvaranje datoteke
    - ▶ učitavanje i ispisivanje sadržaja sloga ili bloka
    - ▶ pozicioniranje na slog ili blok datoteke
    - ▶ ispitivanje statusa datoteke
    - ▶ kreiranje, brisanje datoteke, itd.
      - ▶ preuzimaju parametre poziva iz pozivajućeg okruženja - aplikativnog programa
        - ▶ putanja i naziv datoteke (otvaranje datoteke)
        - ▶ oznaka datoteke (rutine nad otvorenom datotekom)
        - ▶ oznaka promenljive u radnoj zoni programa
      - ▶ prosleđuju u pozivajuće okruženje informacije o statusu izvršenja rutine metode pristupa
        - ▶ osnova za obradu izuzetaka

# Metoda pristupa

- ▶ Upravljanje baferima metode pristupa
  - ▶ okruženje u kojem je implementirana metoda pristupa brine o zadacima upravljanja baferima
    - ▶ alociranje i dealociranje bafera
    - ▶ vođenje evidencije o sadržaju bafera
  - ▶ tri nivoa "baferisanja" podataka datoteke u OM
    - ▶ **nivo sistemskih bafera**
      - ▶ kojim upravlja OS
    - ▶ **nivo bafera metode pristupa**
      - ▶ kojim upravlja okruženje u kojem je implementirana metoda pristupa
    - ▶ **nivo lokacija promenljivih u aplikativnom programu**
      - ▶ kojim upravlja aplikativni program

# Metoda pristupa

## ► Nivoi baferisanja



# Metoda pristupa

- ▶ Okruženja koja uključuju metode pristupa
  - ▶ **operativni sistem**
  - ▶ **programski jezik sa pridruženim paketima (bibliotekama) funkcija**
  - ▶ **sistem za upravljanje bazama podataka**
- ▶ Neki servisi metode pristupa mogu biti implementirani direktno u aplikativnom programu

# Metoda pristupa

## ► Okruženja koja uključuju metode pristupa

### ► **operativni sistem**

- najčešće stariji operativni sistemi "mainframe" računara
- nije bila vidljiva eksplicitna podela na usluge OS niskog i visokog nivoa
  - servisi metode pristupa, tj. fajl sistema, vidljivi kao "monolitna" struktura
  - prvi SUBP nastaju na temelju eksplicitne upotrebe servisa takvih metoda pristupa
- podržavali upravljanje blokovima i baferima metode pristupa
  - Block = Control Interval (CI)
  - dozvoljavaju eksplicitno deklarisanje formata bloka (time i kapaciteta bafera) na nivou pojedinačne datoteke
- pozivi servisa metode pristupa često kombinovani s programskim jezikom Cobol ili PL/I (IBM)

# Metoda pristupa

- ▶ Okruženja koja uključuju metode pristupa
  - ▶ **programski jezik sa pridruženim paketima (bibliotekama) funkcija**
    - ▶ praktično svaki savremeni programski jezik (npr. C, Java) pruža određene usluge metode pristupa, koje mogu biti
      - ▶ ugrađene u sam jezik ili
      - ▶ uključene u određene pakete (biblioteke) funkcija
        - ▶ "upakovane" i isporučene zajedno s kompajlerom i razvojnim okruženjem (npr. za C, stdio.h) ili
        - ▶ isporučene nezavisno od samog jezika / kompajlera
    - ▶ eksplicitno koriste usluge niskog nivoa izabranog OS
    - ▶ najčešće pružaju samo usluge na nivou sloga datoteke
      - ▶ upravljanje blokovima i baferima sakriveno od aplikativnog programa
        - ▶ baferi bibliotečkih funkcija su konstantnog kapaciteta (u C definisan u stdio.h) i nedostupni aplikativnom programu

# Metoda pristupa

- ▶ Okruženja koja uključuju metode pristupa
  - ▶ **sistem za upravljanje bazama podataka**
    - ▶ svaki SUBP obavezno obezbeđuje usluge metode pristupa
      - ▶ najčešće, ne koriste se direktno iz aplikativnih programa
      - ▶ već su na raspolaganju drugim modulima unutar SUBP
    - ▶ eksplicitno koristi usluge niskog nivoa izabranog OS
      - ▶ mada je moguće, u specifičnim situacijama, da SUBP "zaobiđe" usluge niskog nivoa OS
        - ▶ tada SUBP direktno upravlja fizičkom razmenom podataka između datoteke na disku (raw device) i OM
  - ▶ podržava upravljanje blokovima i baferima metode pristupa
    - ▶ Database Block
    - ▶ dozvoljava eksplicitno deklarisanje kapaciteta bloka / bafera na nivou instalacije SUBP
      - ▶ uniformno za sve datoteke kojima upravlja SUBP

# Sadržaj

- ▶ Osnovna struktura datoteke
- ▶ Struktura datoteke kao niza blokova
- ▶ Metoda pristupa
- ▶ Parametri organizacije datoteka
- ▶ Vrste organizacije datoteka
- ▶ Opšte procedure nad datotekama
- ▶ Performanse obrade datoteke

# Parametri organizacije datoteka

## ► Organizacija podataka

- projekat logičke strukture obeležja (LSO)
- projekat i implementacija FSP, u kontekstu
  - isprojektovane LSO i
  - systemske arhitekture
- sa ciljevima da se obezbede
  - zadovoljenje korisničkih zahteva
  - uslovi za efikasnu obradu podataka

## ► Rezultat organizovanja podataka

- sistem baze podataka ili
- sistem datoteka

# Parametri organizacije datoteka

## ► Projekat i implementacija FSP

- izbor načina dodele lokacija slogovima
- izbor načina memorisanja logičkih veza između slogova u LSP
- projektovanje osnovnih struktura podataka
- projektovanje pomoćnih struktura podataka
- proračun i rezervisanje potrebnog prostora na eksternim memorijskim uređajima
- smeštanje slogova sa vezama na eksterne memorijske uređaje
- proračun, praćenje i analiza performansi postupaka obrade podataka

# Parametri organizacije datoteka

## ► Organizacija datoteke

- projektovanje LSO svodi se na projektovanje tipa entiteta  $N(Q, C)$ , tj. tipa sloga
- izbor vrste organizacije datoteke (osnovnih i pomoćnih struktura) zavisi od vrednosti parametara
  - način dodele lokacija slogovima
    - uslovljava i način evidentiranja slobodnog i zauzetog prostora u datoteci
  - način memorisanja logičkih veza između slogova u LSP

# Parametri organizacije datoteka

## ► Način dodele lokacija slogovima (DLS)

### ► moguće vrednosti parametra DLS

- (A) svaki novi slog upisuje se na kraj datoteke, kao fizički susedan u odnosu na poslednji slog datoteke

- ako se prostor datoteke dinamički alocira (povećava pri upisu)

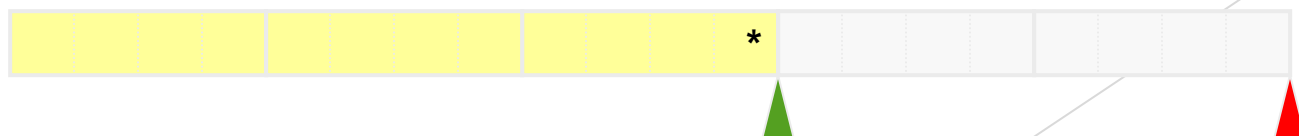
- samo poslednji blok može biti delimično popunjen

- svi ostali blokovi su kompletno popunjeni



- ako se prostor datoteke statički (unapred) alocira

- poslednji upisani slog datoteke deli prostor datoteke na kompletno zauzet i kompletno slobodni deo

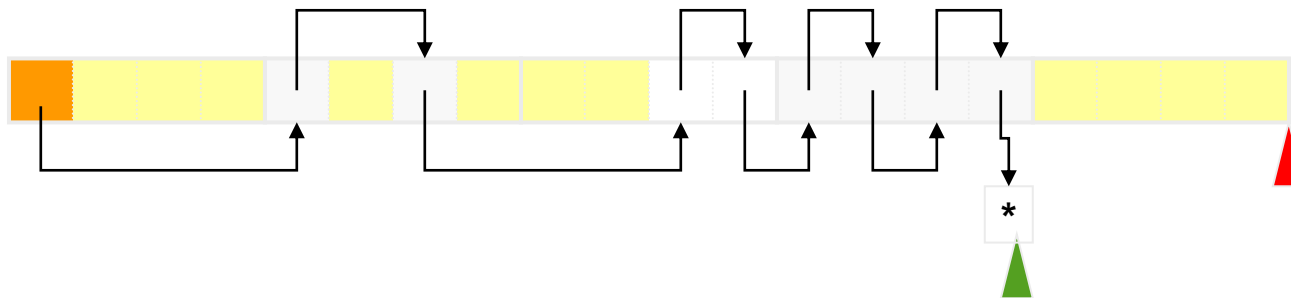


# Parametri organizacije datoteka

## ► Način dodele lokacija slogovima (DLS)

### ► moguće vrednosti parametra DLS

- (B) svaki novi slog dobija prvu slobodnu lokaciju iz spregnute linearne strukture slobodnih lokacija
  - prostor datoteke uvek se statički (unapred) alokira
  - ne izaziva ga svaki  $f$ -ti upis novih slogova u datoteku
  - indeks na listu slobodnih lokacija memoriše se u zaglavlju datoteke



# Parametri organizacije datoteka

## ► Način dodele lokacija slogovima (DLS)

### ► moguće vrednosti parametra DLS

- (C) svaki novi slog dobija slobodnu lokaciju čija relativna adresa predstavlja funkciju vrednosti ključa

- prostor datoteke uvek se statički alocira

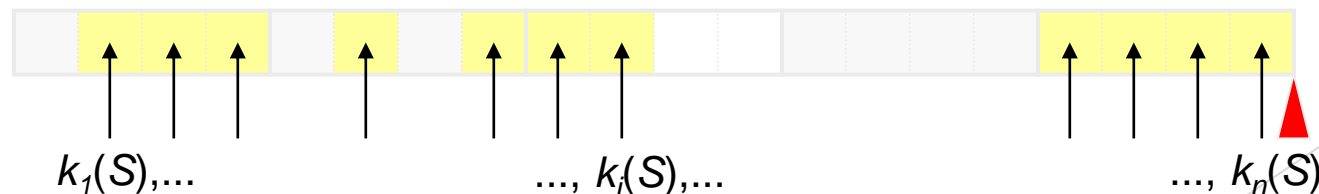
- nezavisno od upisa novih slogova u datoteku

- moguće kad su u pitanju isključivo slogovi konstantne dužine

- funkcija transformacije vrednosti ključa

- hash (analitička) transformacija ili

- tabelarno zadata, uz upotrebu pomoćne strukture



# Parametri organizacije datoteka

## ▶ Način memorisanja logičkih veza (MLV)

- ▶ između slogova u LSP
- ▶ moguće vrednosti parametra MLV
  - ▶ (1) fizičkim pozicioniranjem
    - ▶ logički susedni slogovi smeštaju se u fizički susedne lokacije
  - ▶ (2) pomoću pokazivača kao relativnih adresa
    - ▶ pokazivač memoriše relativnu adresu logički susednog sloga
    - ▶ (2a) polja pokazivača ugrađena u osnovnu strukturu
      - ▶ svaki slog osnovne strukture datoteke proširuje se barem jednim poljem pokazivača
    - ▶ (2b) polja pokazivača ugrađena u pomoćne strukture
      - ▶ uvodi se barem jedna pomoćna, tzv. indeksna struktura
      - ▶ često oblika stabla
      - ▶ sa formatom sloga u obliku para ili n-torke

*(polja identifikatora sloga, polja pokazivača)*

# Parametri organizacije datoteka

## ▶ Način memorisanja logičkih veza (MLV)

- ▶ između slogova u LSP
- ▶ moguće vrednosti parametra MLV
  - ▶ (3) logičke veze se ne memorišu
    - ▶ u FSP ne postoje podaci o logički susednim slogovima
    - ▶ podaci o logički susednim slogovima se jedino mogu generisati
      - ▶ putem posebnih programa, na zahtev korisnika
      - ▶ programi za uređivanje (sortiranje) datoteka

# Sadržaj

- ▶ Osnovna struktura datoteke
- ▶ Struktura datoteke kao niza blokova
- ▶ Metoda pristupa
- ▶ Parametri organizacije datoteka
- ▶ Vrste organizacije datoteka
- ▶ Opšte procedure nad datotekama
- ▶ Performanse obrade datoteke

# Vrste organizacije datoteka

## ► Vrste organizacija datoteka

### ► **osnovne organizacije**

- organizacija datoteke svodi se na osnovnu organizaciju
- FSP nad skupom slogova organizovana je u jednoj memorijskoj zoni
  - često, to je i jedna datoteka operativnog sistema

### ► **složene organizacije**

- dobijaju se kombinovanjem osnovnih organizacija
- FSP uključuju barem dve memorijske zone
  - što mogu biti i barem dve datoteke operativnog sistema
  - osnovna FSP može biti raspoređena u jednu, ili u više od jedne zone
  - mogu se pojaviti pomoćne strukture podataka, smeštene u posebnim zonama

# Vrste organizacije datoteka

## ► Vrste organizacija datoteka

### ► osnovne organizacije datoteka

- serijska (pile, heap)
- sekvencijalna (sequential)
- spregnuta (chain)
- rasuta (hash) sa jedinstvenim memorijskim prostorom
  - direktna
  - relativna
  - statička rasuta
  - dinamička rasuta

### ► složene organizacije

- rasute (hash) sa zonom prekoračenja
- statičke indeksne (indeks-sekvencijalna)
- dinamičke indeksne (organizacije sa B-stablom)

# Vrste organizacije datoteka

## ► Osnovne organizacije datoteka

<b>DLS</b> \ <b>MLV</b>	<b>1</b> [ 1   2   3   4   ... ]	<b>2a</b> [ [ 3   1   4   2 ] ] 	<b>3</b> [ 3   1   4   2   ... ]
<b>A</b> 			
<b>B</b> 			
<b>C</b> 			

## sekvencijalna

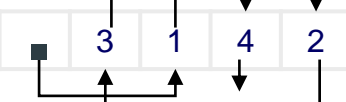
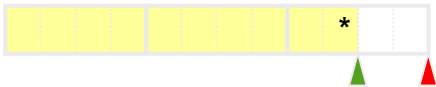
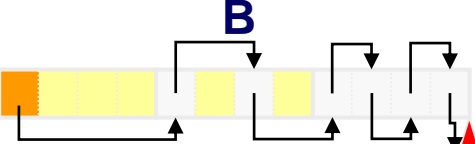

**serijska**

**spregnuta**

**rasuta**

# Vrste organizacije datoteka

## ► Osnovne organizacije datoteka

DLS \ MLV	1	2a	3
	1 2 3 4 ...		3 1 4 2 ...
<b>A</b> 	sekvencijalna		serijska
<b>B</b> 		spregnuta	
<b>C</b> 			rasuta

# Vrste organizacije datoteka

## ► Složene organizacije datoteka

- rasute (hash) sa zonom prekoračenja
  - primarna zona - osnovna struktura
    - osnovna rasuta organizacija
  - zona prekoračenja - nastavak osnovne strukture
    - spregnuta ili serijska organizacija
- statičke indeksne (indeks-sekvencijalne)
  - primarna zona - osnovna struktura
    - sekvencijalna organizacija
  - zona prekoračenja - nastavak osnovne strukture
    - spregnuta organizacija
  - zona indeksa - pomoćna struktura
    - spregnuta organizacija
      - sprezanje u obliku n-arnog stabla traženja

# Vrste organizacije datoteka

## ► Složene organizacije datoteka

- dinamičke indeksne (organizacije sa B-stablom)
  - primarna zona - osnovna struktura
    - serijska ili spregnuta organizacija
  - zona indeksa - pomoćna struktura
    - spregnuta organizacija
      - sprezanje u obliku jedne od varijanti B-stabla

# Vrste organizacije datoteka

- ▶ Navedene vrste organizacije pojavljuju se u praksi kao
  - ▶ **fizičke organizacije datoteka**
    - ▶ u sistemima datoteka
      - ▶ svaka datoteka u sistemu datoteka pojavljuje se kao jedna ili više posebnih OS datoteka
  - ▶ **fizičke organizacije tabela**
    - ▶ u sistemima baza podataka
      - ▶ svaka tabela BP može biti distribuirana u više datoteka podataka kojima upravlja SUBP
      - ▶ u jednoj datoteci podataka kojom upravlja SUBP može biti smešteno više tabela BP

# Sadržaj

- ▶ Osnovna struktura datoteke
- ▶ Struktura datoteke kao niza blokova
- ▶ Metoda pristupa
- ▶ Parametri organizacije datoteka
- ▶ Vrste organizacije datoteka
- ▶ Opšte procedure nad datotekama
- ▶ Performanse obrade datoteke

# Opšte procedure nad datotekama

- ▶ Vrste postupaka (operacija) nad LSP datoteke
  - ▶ formiranje datoteke
  - ▶ pristupanje u datoteci
  - ▶ traženje u datoteci
  - ▶ pretraživanje u datoteci
  - ▶ obrada datoteka
  - ▶ ažuriranje datoteke
  - ▶ reorganizacija datoteke

# Opšte procedure nad datotekama

## ► **Formiranje datoteke**

- postupak kreiranja FSP datoteke
  - sa smeštanjem slogova na eksterni memorijski uređaj
  - saglasno projektovanoj organizaciji
  - na osnovu sadržaja neke druge strukture podataka
    - uključuje preuzimanje podataka iz drugih datoteka ili
    - direktno zadavanje podataka od strane korisnika
- dve vrste datoteka
  - datoteke koje se formiraju u posebnom postupku
    - najčešće: sekvencijalna, spregnuta, statičke rasute, statičke indeksne
  - datoteke koje se formiraju u redovnom postupku ažuriranja (upisa novih slogova)
    - najčešće: serijska, indeksne s B stablima, dinamička rasuta

# Opšte procedure nad datotekama

## ► **Pristupanje u datoteci**

- postupak pozicioniranja na željenu lokaciju
  - sloga ili bloka datoteke
- vrste pristupa
  - sekvencijalni pristup
    - automatsko održavanje relativne adrese tekućeg pokazivača
    - operacija se odnosi na neposredno susednu lokaciju u odnosu na lokaciju na kojoj je obavljena prethodna operacija
  - direktni pristup
    - eksplicitno zadavanje relativne adrese tekućeg pokazivača koji ukazuje na lokaciju nad kojom će se realizovati neka operacija
  - dinamički
    - kombinacija sekvencijalnog i direktnog pristupa
- **NAPOMENA:** sekvencijalni i direktni pristup i sekvencijalna i direktna organizacija datoteke nisu isti pojmovi

# Opšte procedure nad datotekama

## ► Traženje u datoteci

- algoritam **AT**:  $\text{dom}(K) \rightarrow \text{Ind} \times A \times S$ 
  - za zadatu vrednost argumenta traženja
    - vrednost ključa iz domena,  $a \in \text{dom}(K)$
  - u stanju je da generiše i vrati, po potrebi, u program
    - indikaciju uspešnosti traženja  $\text{Ind} = \{\text{true}, \text{false}\}$ 
      - ako je slog nađen, traženje je uspešno (true)
      - ako slog nije pronađen, traženje je neuspešno (false)
    - relativnu adresu mesta zaustavljanja traženja
      - iz skupa svih adresa u adresnom prostoru datoteke **A**
      - koji, po potrebi, uključuje i adresu prve naredne lokacije nakon kraja datoteke
  - sadržaj sloga na mestu zaustavljanja traženja
    - ili specijalnu vrednost, ako je reč o nepostojećem slogu

# Opšte procedure nad datotekama

## ► Traženje u datoteci

► algoritam AT:  $\text{dom}(K) \rightarrow \text{Ind} \times A \times S$

- nisu uvek svi nabrojani izlazni parametri potrebni aplikativnom programu
- specifični algoritmi traženja na izlazu će generisati samo vrednosti onih parametara koji su stvarno neophodni aplikativnom programu
- moguće svrhe primene algoritma traženja
  - da bi se utvrdilo ima li traženog sloga u datoteci ili nema
    - npr. radi provere ispunjenosti uslova za upis novog ili brisanje postojećeg sloga sa zatom vrednošću ključa
  - da bi se utvrdila adresa na kojoj se traženi slog nalazi
    - jer je potrebno direktno pozicioniranje na datu adresu
  - da bi se preneo sadržaj traženog sloga u aplikativni program
    - npr. radi daljih potreba obrade podataka

# Opšte procedure nad datotekama / Traženje u datoteci

- ▶ algoritam AT:  $\text{dom}(K) \rightarrow \text{Ind} \times A \times S$

## POČETAK TRAŽENJA

generisanje početne relativne adrese traženja

postoji potreba za nastavak traženja  $\leftarrow$  DA

**RADI** *petlja\_traženja* **DOK** postoji potreba za nastavak traženja

čitanje sadržaja sloga s tekuće adrese

**AKO** argument traženja = vrednost ključa tekućeg sloga **TADA**

traženje uspešno, postoji potreba za nastavak traženja  $\leftarrow$  NE

**INAČE**

**AKO** postoje uslovi za nastavak traženja **TADA**

generisanje naredne relativne adrese traženja

**INAČE**

traženje neuspešno, postoji potreba za nastavak traženja  $\leftarrow$  NE

**KRAJ AKO**

**KRAJ AKO**

**KRAJ RADI** *petlja\_traženja*

**KRAJ TRAŽENJA**

# Opšte procedure nad datotekama

## ► Traženje u datoteci

- metode traženja s obzirom na vrstu postupka
  - linearno traženje
    - moguće u sekvencijalnim, serijskim i rasutim organizacijama
  - binarno traženje
    - isključivo moguće u sekvencijalnim organizacijama
  - traženje praćenjem pokazivača
    - isključivo moguće u spregnutim organizacijama
      - u osnovnim strukturama i
      - uz korišćenje pomoćnih struktura
  - traženje metodom transformacije argumenta u adresu
    - $h: \text{dom}(K) \rightarrow A$
    - moguće u rasutim organizacijama

# Opšte procedure nad datotekama

## ► Traženje u datoteci

### ► vrste traženja s obzirom na predistoriju traženja

#### ► traženje slučajno odabranog sloga (tso)

- izbor početne adrese traženja je unutrašnje pitanje algoritma
  - ne zavisi od mesta zaustavljanja prethodnog traženja, niti od toga da li je postojalo prethodno traženje
- moguće u svim organizacijama datoteka

#### ► traženje logički narednog sloga (tln)

- početna adresa traženja predstavlja adresu na kojoj je zaustavljeno prethodno traženje
  - moguće je ako je prethodno postojalo barem jedno traženje
- svaka naredna adresa traženja može biti samo adresa logički narednog sloga
- moguće u organizacijama u kojima se vode podaci o logički narednom slogu

# Opšte procedure nad datotekama

## ► Pretraživanje u datoteci

► algoritam AP:  $\text{dom}(\text{LogUslov}) \rightarrow P(S)$  ili  $P(A)$

► za zadatu vrednost argumenta pretraživanja

► zadati logički uslov

► u stanju je da generiše i vrati, po potrebi, u program

► skup slogova koji zadovoljavaju logički uslov pretraživanja

►  $P(S)$  - partitivni skup skupa slogova  $S$  ili

► skup adresa slogova koji zadovoljavaju logički uslov pretraživanja

►  $P(A)$  - partitivni skup skupa adresa iz adresnog prostora

► pretraživanje je uspešno, ako je skup slogova koji zadovoljava zadati uslov neprazan

# Opšte procedure nad datotekama

## ► Pretraživanje u datoteci

- algoritam **AP**:  $\text{dom}(\text{LogUslov}) \rightarrow P(S)$  ili  $P(A)$ 
  - logički uslov zahteva definisanje sintakse zapisivanja
  - predstavlja logički izraz, moguća upotreba
    - logičkih izraza i operandada
    - relacionih izraza i operandada
    - tipskih izraza i operandada
      - aritmetičkih, alfanumeričkih, datumskih
    - u ulozi operandada mogu se pojaviti atributi tipa sloga, konstante i funkcije primenjene nad izrazima
  - neki specijalni tipovi logičkih uslova
    - uslovi konjuktivnog tipa:  $A_1 = a_1 \wedge \dots \wedge A_n = a_n$
    - uslovi disjunktivnog tipa:  $A_1 = a_1 \vee \dots \vee A_n = a_n$
- **sekundarni ključ**
  - niz obeležja strukture po kojem se vrši pretraživanje

# Opšte procedure nad datotekama

## ▶ **Ažuriranje datoteke**

- ▶ postupak dovođenja LSP datoteke u sklad sa izmenjenim stanjem klase entiteta u realnom sistemu
- ▶ osnovne operacije
  - ▶ **upis novog sloga u datoteku**
    - ▶ zahteva prethodno neuspešno traženje
    - ▶ može iziskivati premeštanje određenog broja drugih slogova
  - ▶ **modifikacija vrednosti neprimarnih atributa sloga**
    - ▶ zahteva prethodno uspešno traženje
    - ▶ uobičajeno, zabranjuje se modifikacija vrednosti obeležja primarnog ključa
      - ▶ po kojem je uspostavljena osnovna organizacija
  - ▶ **brisanje postojećeg sloga iz datoteke**
    - ▶ zahteva prethodno uspešno traženje
    - ▶ može iziskivati premeštanje određenog broja drugih slogova

# Opšte procedure nad datotekama

## ► Ažuriranje datoteke

### ► vrste brisanja

#### ► logičko brisanje sloga iz datoteke

- svodi se na izmenu vrednosti polja statusa sloga
- iz statusa aktuelnog u status neaktuelnog sloga
- neaktuelni slog i dalje zauzima lokaciju u memorijskom prostoru
- lokacije neaktuelnih slogova oslobađaju se reorganizacijom

#### ► fizičko brisanje sloga iz datoteke

- dovodi do izmene sadržaja bloka u kojem se nalazio izbrisani slog
- može izazvati pomeranje drugih slogova iz jednih u druge lokacije
- dovodi do oslobađanja jedne lokacije sloga u memorijskom prostoru
  - to ne mora obavezno biti lokacija izbrisnog sloga

# Opšte procedure nad datotekama

## ► Obrada datoteka

- algoritamski iskazani niz operacija nad LSP jedne ili više datoteka
  - sa ciljem svrsishodne transformacije podataka datoteka
  - moguća primena operacija
    - pristupa slogovima
      - jedina obavezna vrsta operacija u obradi datoteka
    - traženja i pretraživanja
    - ažuriranja (unosa, brisanja i modifikacije)
    - generisanja (izračunavanja) novih podataka

# Opšte procedure nad datotekama

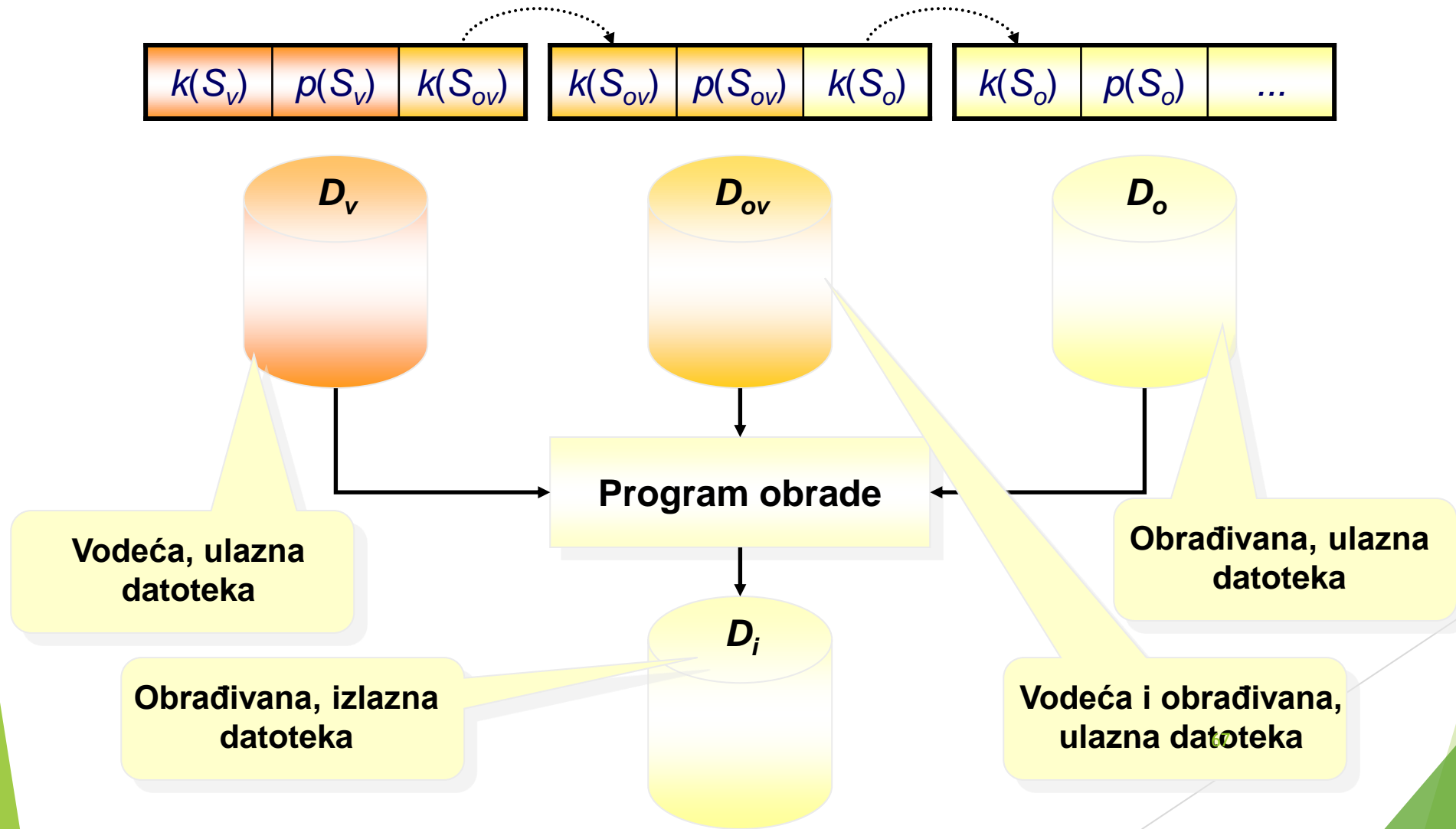
- ▶ **Obrada datoteka**
- ▶ Uloge datoteka u obradi
  - ▶ podela prema vrstama primenjenih operacija u obradi
    - ▶ **ulazna datoteka**
      - ▶ datoteka u kojoj se isključivo vrše čitanja
    - ▶ **izlazna datoteka**
      - ▶ datoteka u koju se isključivo zapisuju novi slogovi u obradi
    - ▶ **ulazno-izlazna datoteka**
      - ▶ datoteka u kojoj se vrše i čitanja i ažuriranja slogova

# Opšte procedure nad datotekama

- ▶ **Obrada datoteka**
- ▶ Uloge datoteka u obradi
  - ▶ podela prema ulozi u traženjima slogova
    - ▶ **vodeća datoteka**
      - ▶ datoteka koja isključivo generiše argumente traženja ili pretraživanja slogova tokom obrade
      - ▶ barem jedna ulazna datoteka u obradi mora biti vodeća
    - ▶ **obrađivana datoteka**
      - ▶ datoteka u kojoj se isključivo vrše traženja ili pretraživanja, na osnovu generisanih argumenata
    - ▶ **vodeća i obrađivana datoteka**
      - ▶ datoteka sa obe uloge
        - ▶ vodeća za neku drugu, obrađivanu i
        - ▶ obrađivana, u odnosu na neku vodeću

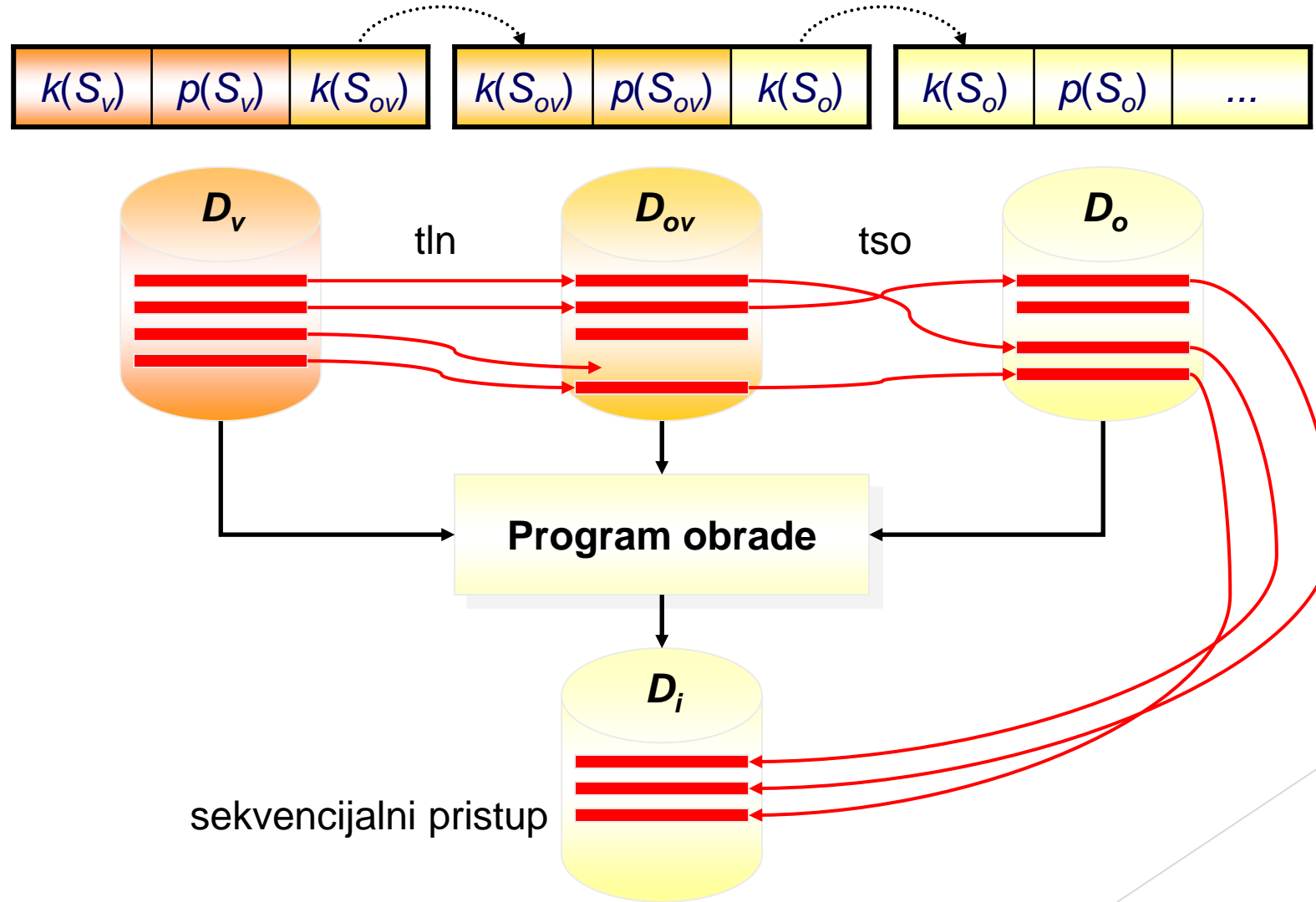
# Opšte procedure nad datotekama

## ► Obrada datoteka



# Opšte procedure nad datotekama

## ► Obrada datoteka



# Opšte procedure nad datotekama

## ► Obrada datoteka

- vrste obrade, prema načinima traženja slogova u obrađivanoj datoteci
  - **direktna obrada**
    - u svakom narednom koraku obrade zahteva se traženje slučajno odabranog sloga (tso)
  - **redosledna (sekvencijalna) obrada**
    - u svakom narednom koraku obrade zahteva se
      - traženje logički narednog sloga (tln) i/ili
      - sekvencijalni pristup fizički susednoj lokaciji

# Opšte procedure nad datotekama

## ► Reorganizacija datoteke

- ponovno formiranje datoteke
- u cilju dovođenja u sklad FSP sa novim stanjem LSP
- motivacija
  - operacije ažuriranja vrše izmene u LSP koje FSP nekada ne prati na odgovarajući način
  - to dovodi do degradacije performansi rada sa datotekom
  - primeri
    - nagomilavanje neaktuelnih, logički izbrisanih slogova koji zauzimaju lokacije u FSP
    - nagomilavanje lanaca slogova
    - neizbalansiranost podataka s obzirom na postojeću indeksnu strukturu
    - prevelika fragmentacija slobodnog prostora

# Opšte procedure nad datotekama

## ► Reorganizacija datoteke

- organizacije koje traže povremenu reorganizaciju
  - sekvencijalna
  - spregnuta
  - statička rasuta
  - statička indeksna
- organizacije koje ne traže povremenu reorganizaciju
  - serijska
    - reorganizacija nije neophodna
  - indeksna s B stablom
    - reorganizacija se sprovodi dinamički i lokalizovana je
  - dinamička rasuta
    - reorganizacija se sprovodi dinamički i lokalizovana je

# Sadržaj

- ▶ Osnovna struktura datoteke
- ▶ Struktura datoteke kao niza blokova
- ▶ Metoda pristupa
- ▶ Parametri organizacije datoteka
- ▶ Vrste organizacije datoteka
- ▶ Opšte procedure nad datotekama
- ▶ Performanse obrade datoteke

# Performanse obrade datoteke

- ▶ Mere podobnosti datoteke sa zadatom organizacijom da participira u obradi
  - ▶ kao vodeća ili obrađivana
  - ▶ u redoslednoj ili direktnoj obradi
- ▶ **Idealna organizacija datoteke**
  - ▶ zahteva tačno onoliko lokacija koliko sadrži slogova
    - ▶ faktor popunjenosti 100%
  - ▶ zahteva najviše jedan pristup za tso i tln
  - ▶ zahteva najviše jedan pristup za pretraživanje
    - ▶ po bilo kom zadatom uslovu
  - ▶ zahteva jedan pristup za bilo koju operaciju ažuriranja
  - ▶ nikada ne zahteva reorganizaciju

# Performanse obrade datoteke

- ▶ Izbor vrste organizacije datoteke
  - ▶ predstavlja kompromisno rešenje
    - ▶ nemoguće je da jedna vrsta organizacije zadovolji sve navedene zahteve
    - ▶ favorizacija jednih često defavorizuje druge zahteve
    - ▶ uzimaju se u obzir potrebe i uticajnost aplikativnih programa
  - ▶ favorizuju se željene mere performansi, u odnosu na zauzeće memorijskog prostora
    - ▶ cena memorisanja po jedinici kapaciteta sve niža

# Performanse obrade datoteke

- ▶ Ukupno vreme traženja ili pretraživanja slogova
  - ▶ zavisi od
    - ▶ broja i vremena pristupa blokovima na jedinici diska
      - ▶ broj dominantno opredeljen vrstom organizacije datoteke
        - ▶ u slučaju pretraživanja i prirodom logičkog uslova
      - ▶ broj (za velike datoteke) zavisi od karakteristika fajl sistema OS
      - ▶ vreme dominantno opredeljeno karakteristikama diska (~ 10ms)
    - ▶ vremena prenosa bloka sa diska u OM
      - ▶ dominantno opredeljeno karakteristikama diska i sprežnog podsistema (< 1ms)
    - ▶ broja i vremena upoređivanja argumenta sa vrednošću ključa
      - ▶ broj dominantno opredeljen vrstom organizacije datoteke
        - ▶ u slučaju pretraživanja i prirodom logičkog uslova
      - ▶ vreme dominantno opredeljeno karakteristikama OM i CPU (~ 10ns)

# Performanse obrade datoteke

## ► Mere za ocenu performansi

### ► (A)

- broj pristupa blokovima
- broj upoređivanja argumenta i vrednosti ključa

### ► (B)

- srednji broj
- broj u najgorem slučaju (apsolutni broj)

### ► (C)

- traženje logički narednog sloga
- traženje slučajno odabranog sloga
- operacije ažuriranja (upis, brisanje, modifikacija)

### ► (D)

- uspešna operacija
- neuspešna operacija

# Performanse obrade datoteke

## ► Mere za ocenu performansi

		Broj pristupa		Broj upoređivanja	
		apsolutni	srednji	apsolutni	srednji
tIn	uspešno	√	√	√	√
	neuspešno	√	√	√	√
tso	uspešno	√	√	√	√
	neuspešno	√	√	√	√
upis	uspešno	√	√	√	√
	neuspešno	√	√	√	√
brisanje	uspešno	√	√	√	√
	neuspešno	√	√	√	√
modifikacija	uspešno	√	√	√	√
	neuspešno	√	√	√	√

# Sadržaj

- ▶ Osnovna struktura datoteke
- ▶ Struktura datoteke kao niza blokova
- ▶ Metoda pristupa
- ▶ Parametri organizacije datoteka
- ▶ Vrste organizacije datoteka
- ▶ Opšte procedure nad datotekama
- ▶ Performanse obrade datoteke

# Pitanja i komentari



Kraj prezentacije

# Metode pristupa i organizacija datoteka

Usluge metoda pristupa i vrste organizacija datoteka