

Fakultet tehničkih nauka, DRA,  
Novi Sad

Predmet:  
Organizacija podataka

dr Vladimir Ivančević  
Nikola Todorović  
Vladimir Jovanović

# Serijska organizacija datoteka

# Serijska datoteka

- 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

# Serijska datoteka

- Osnovna struktura
  - redosled memorisanja slogova najčešće prema hronološkom redosledu njihovog nastanka
  - slogovi mogu, a i ne moraju, biti blokirani

# Serijska datoteka

- Primer serijske 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	$p(S_{13})$	*			

# Zadatak 1

- Napisati Python program koji će omogućiti rad sa podacima o evidentiranim prispećima zatvorenika u Gradski zatvor. Za svaki dolazak novog zatvorenika u **neblokiranoj serijskoj datoteci** beleži se:
  - evidencioni broj (do 8 cifara)
  - šifra zatvorenika (tačno 7 karaktera)
  - datum i vreme dolaska
  - oznaka ćelije u koju će zatvorenik biti smešten (tačno 5 karaktera)
  - dužina kazne u mesecima (do 480 meseci)

# Zadatak 1

- Omogućiti
  - odabir datoteke
  - formiranje datoteke
  - pretragu datoteke
  - unos novog sloga
  - ispis svih slogova
  - ažuriranje sloga
    - direktna obrada serijske datoteke
  - brisanje sloga
    - Logičko brisanje

# Zadatak 2

- Implementirati fizičko brisanje nad prethodnom datotekom



# Zadatak 3

- Implementirati blokiranu serijsku datoteku nad istim slogom.
  - Sa svim operacijama

# Sekvencijalna organizacija datoteka

# Sekvencijalna datoteka

- 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

# Sekvencijalna datoteka

- Osnovna struktura
  - logički susedni slogovi smeštaju se u fizički susedne lokacije
    - rastuće uređenje po vrednostima ključa -> slog sa najmanjom vrednošću ključa smešta se u prvu lokaciju
  - naziva se i fizički sekvencijalnom organizacijom

# Sekvencijalna datoteka

- 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  $f$  slogova
    - poželjno da faktor blokiranja  $f$  bude što veći

# Sekvencijalna datoteka

- Osnovna struktura
  - 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 datoteka

- Primer sekvencijalne datoteke

$A_1$					
	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	$p(S_{13})$	*		

# Zadatak 1

- Napisati *Python* program koji će omogućiti rad sa podacima o evidentiranim prispećima zatvorenika u Gradski zatvor. Za svaki dolazak novog zatvorenika u **sekvencijalnoj datoteci** sa faktorom blokiranja **f = 3** beleži se:
  - evidencioni broj (do 8 cifara)
  - šifra zatvorenika (tačno 7 karaktera)
  - datum i vreme dolaska
  - oznaka ćelije u koju će zatvorenik biti smešten (tačno 5 karaktera)
  - dužina kazne u mesecima (do 480 meseci)



# Zadatak 1

- Implementirati:
  - formiranje datoteke
  - unos novog sloga
  - ažuriranje sloga
  - brisanje sloga
    - logičko
  - pretraga po ključu
  - reorganizacija datoteke
  - ispis svih slogova

# Zadatak 2

- Implementirati sekvencijalnu datoteku kod koje se slogovi unose pomoću serijske datoteke izmena.
  - Sve greške koje se pojave tokom funkcionisanja programa treba da budu smeštene u serijsku datoteku grešaka

# Zadatak 3

- Implementirati sekvencijalnu datoteku čiji parametri (faktor blokiranja, postojanje serijske datoteke izmena, putanje do datoteka) se zadaju putem posebnog programa. Nakon zadavanja ovih parametara, sekvencijalna datoteka treba da automatski funkcioniše sa zadatim parametrima.