

PROJEKAT

VI termin

Dr Nevena Radović

Multidimenzionalni nizovi

- Memorija je suštinski jednodimenzionalni entitet. Imajući to u vidu, multidimenzionalni niz se implementira kao set jednodimenzionalnih nizova.
- Postoje dva osnovna načina implementacije multidimenzionalni nizova:
 - implementacija po vrstama,
 - implementacija po kolonama.
- U nastavku ćemo se fokusirati na dvodimenzionalne nizove. Ovaj princip je dalje moguće na jednostavan način proširiti na veći broj dimenzija.

Dvodimenzionalni nizovi

- Dvodimenzionalni nizovi (matrice) se intenzivno koriste u višim programskim jezicima.
- Prilikom kreiranja matrice neophodno je definisati njene dimenzije:
 - › Broj vrsta
 - › Broj kolona
- Kako izgleda raspored elemenata u matrici, prilikom njenog definisanja u višim programskim jezicima?
- Raspored elemenata u matrici **A** dimenzija **(3,4)**:

A[2][0]	A[2][1]	A[2][2]	A[2][3]
A[1][0]	A[1][1]	A[1][2]	A[1][3]
A[0][0]	A[0][1]	A[0][2]	A[0][3]

Implementacija po vrstama

A[2][0]	A[2][1]	A[2][2]	A[2][3]
A[1][0]	A[1][1]	A[1][2]	A[1][3]
A[0][0]	A[0][1]	A[0][2]	A[0][3]

The diagram illustrates the row-major traversal of a 3x4 matrix A . The matrix A is shown as a 3x4 grid of cells, each containing an index $A[i][j]$. To the right of the matrix, a vertical list shows the sequence of indices and their corresponding matrix elements. Red arrows connect the elements in the matrix to their positions in the list, demonstrating the traversal order.

11	A[2][3]
10	A[2][2]
9	A[2][1]
8	A[2][0]
7	A[1][3]
6	A[1][2]
5	A[1][1]
4	A[1][0]
3	A[0][3]
2	A[0][2]
1	A[0][1]
0	A[0][0]

Implementacija po vrstama

- Kako konvertovati indekse dvodimenzionalnog niza (matrice) u indeks odgovarajućeg jednodimenzionalnog niza?
- Kod implementacije po vrstama memorijski offset se računa na sljedeći način:

adresa = početna_adresa + (index_vrste *broj_kolona + index_kolone) * mem_veličina_podatka

- Memorijska veličina podatka je izražena u bajtima.
- **Primjer:** Adresa podatka **A[1][2]**.
- **Rješenje:**

adresa = početna_adresa + (1*4 + 2)*4 = početna_adresa + (4 + 2)*4 = početna_adresa + 24

Implementacija po kolonama

A[2][0]	A[2][1]	A[2][2]	A[2][3]
A[1][0]	A[1][1]	A[1][2]	A[1][3]
A[0][0]	A[0][1]	A[0][2]	A[0][3]

11	A[2][3]
10	A[1][3]
9	A[0][3]
8	A[2][2]
7	A[1][2]
6	A[0][2]
5	A[2][1]
4	A[1][1]
3	A[0][1]
2	A[2][0]
1	A[1][0]
0	A[0][0]

Implementacija po kolonama

- Kako konvertovati indekse dvodimenzionalnog niza (matrice) u indeks odgovarajućeg jednodimenzionalnog niza?
- Kod implementacije po kolonama memorijski ofset se računa na sljedeći način:

addr = početna_adresa + (index_kolone * broj_vrstta + index_vrste) * mem_veličina_podatka

- Memorijска veličina podatka je izražena u bajtima.
- **Primjer:** Adresa podatka **A[1][2]**.
- **Rješenje:**

adresa = početna_adresa + (2*3 + 1)*4 = početna_adresa + (6 + 1)*4 = početna_adresa + 28

- **Primjer:** Za prikazanu kvadratnu matricu napisati proceduru koja će izračunati sumu elemenata sa glavne dijagonale, a rezultate prikazati pomoću konzole.

11	12	13	14	15
16	17	18	19	20
21	22	23	24	25
26	27	28	29	30
31	32	33	34	35

Deklaracija podataka

.data

matrica:

.word 11, 12, 13, 14, 15

.word 16, 17, 18, 19, 20

.word 21, 22, 23, 24, 25

.word 26, 27, 28, 29, 30

.word 31, 32, 33, 34, 35

dimenzija:

.word 5

suma:

.word 0

poruka:

.ascii "Suma elemenata na dijagonali = "

Text/kod zadatka

.text

.globl main

main:

Poziv procedure za sumiranje

la \$a0, matrica
lw \$a1, dimenzija
jal sumiranje
sw \$v0, suma

pocetna adresa
dimenzija

Prikazivanje rezultata

li \$v0, 4
la \$a0, poruka
syscall

print string

li \$v0, 1
lw \$a0, suma
syscall

print integer

Kraj programa

li \$v0, 10
syscall

terminate
system call

.end main

```
# adr = poc_adr + (index_vr * broj_kol + index_kol) * mem_vel_podatka
# Argumenti: $a0 – pocetna adresa, $a1 - dimenzije
# Rezultat se vraca u $v0 – suma elemenata sa glavne dijagonale
```

sumiranje:

li \$v0, 0

suma=0

li \$t1, 0

indeks i=0

Loop:

mul \$t3, \$t1, \$a1

racunanje adrese

add \$t3, \$t3, \$t1

mul \$t3, \$t3, 4

add \$t4, \$a0, \$t3

lw \$t5, (\$t4)

add \$v0, \$v0, \$t5

add \$t1, \$t1, 1

i=i+1

blt \$t1, \$a1, Loop

jr \$ra

```
[0040002c] 8c250064 lw $5, 100($1)
```

```
[0040003]
```

```
[0040003]
```

```
[0040003] Suma elemenata na dijagonali= 115
```

```
[0040003]
```

```
[0040004]
```

```
[0040004]
```

```
[0040004]
```

```
[0040005]
```

```
[0040005]
```

```
[0040005]
```

```
[0040005]
```

```
[0040006]
```

```
[0040006]
```

```
[0040006]
```

```
[0040006]
```

```
[0040007]
```

```
[0040007]
```

```
[0040007] 71615802 mul $11, $11, $1
```

Console

