

УуБ



ФОН



Е



лабСи



па



Универзитет у Београду

факултет организационих наука

Катедра за софтверско инжењерство

лабораторија за софтверско инжењерство

ПРОГРАМИРАЊЕ I

ПРОГРАМИРАЊЕ I

Предавање #5

Проф. др **Саша Д. Лазаревић**, дипл. инж. инф.

sasa.lazarevic@fon.bg.ac.rs

Садржај предавања

Низови:

- 1) Дефинисање низова
- 2) Приступање елементима низа
- 3) Иницијализација низова
- 4) Ниске (низови знакова)
- 5) Вишедимензиони низови
- 6) Низови као аргументи функција: разлике и сличности са скаларним аргументима

Садржај предавања

Низови:

- 1) Дефинисање низова
- 2) Приступање елементима низа
- 3) Иницијализација низова
- 4) Ниске (низови знакова)
- 5) Вишедимензиони низови
- 6) Низови као аргументи функција: разлике и сличности са скаларним аргументима

Дефинисање низова 1/2

- ❖ **Низ** (array) је колекција података; садржи податке истог типа који се чувају у суседним меморијским локацијама. Ти подаци се називају елементи низа.
- ❖ Капацитет (димензија) низа = максималан број елемената које низ може да садржи
- ❖ Број елемената низа = тренутни број (искоришћених) елемената низа
- ❖ Величина низа = број бајтова (Бу) које заузима низ
- ❖ Елементима низа се приступа преко индекса.
- ❖ Елементи низа могу бити било ког типа, сем непотпуног типа.

Дефинисање низова 2/2

- ❖ И сам низ је неког типа; тај нови низ(овни) тип је одређен:
 - ▶ Бројем елемената који чине низ
 - ▶ Типом елемената који чине низ
- ❖ На пример:

```
int nizInt[8];    // celobrojni niz od 8 elemenata
float n[123];    // float niz od 123 elementa
```
- ❖ У програмском језику Ц низови могу да буду и непотпуни, уколико се не одређује број елемената низа. На пример:

```
int n;
int nepIntNiz [2 * n];    // nepotpuni celobrojni niz
double nepDblNiz[n];    // nepotpuni double niz
```

Садржај предавања

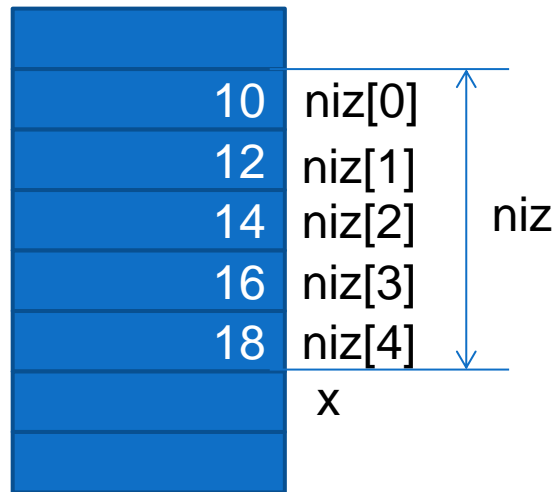
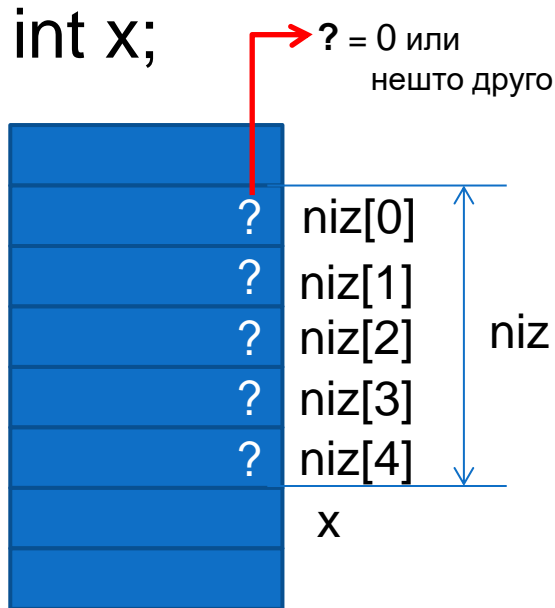
Низови:

- 1) Дефинисање низова
- 2) Приступање елементима низа
- 3) Иницијализација низова
- 4) Ниске (низови знакова)
- 5) Вишедимензиони низови
- 6) Низови као аргументи функција: разлике и сличности са скаларним аргументима

Декларисање низа и додела вредности елементима низа

```
#define DIM 5  
int niz[DIM];  
int x;
```

```
for (int i = 0; i < DIM; ++i)  
    niz[ i ] = 2 * i + 10;
```



ПОГРЕШНО ЈЕ :

niz[5] = 20;

Зашто?

Разликовати:

- syntax error
- run-time error

Приступање елементима низа остварује се путем оператора индексирања: []

Пристапање елем. низа помоћу показивача

- ❖ Други начин приступа елементима низа је помоћу показивача.
- ❖ Преводаилац име низа имплицитно конвертује у показивач на први елемент у свим изразима, изузев у операцијама са оператором `sizeof()`.
- ❖ На пример:
 - ▶ додела вредности 8 свим елементима низа `niz`, може се остварити наредбом:

```
for (int * pok = niz; pok < (niz + DIM); ++pok)  
    *pok = 8;
```
 - ▶ додела вредности {10,12,14,16,18} елементима низа `niz`, може се остварити наредбом:

```
for (int * pok = niz, i = 0; pok < (niz + DIM); ++pok, ++i)  
    *pok = 2 * i + 10;
```
- ❖ `niz + i` ::= адреса елемента са индексом `i` (тј. `&niz[i]`)
- ❖ `*(niz + i)` ::= вредност елемента са индексом `i` (тј. `niz[i]`)

Садржај предавања

Низови:

- 1) Дефинисање низова
- 2) Приступање елементима низа
- 3) **Иницијализација низова**
- 4) Ниске (низови знакова)
- 5) Вишедимензиони низови
- 6) Низови као аргументи функција: разлике и сличности са скаларним аргументима

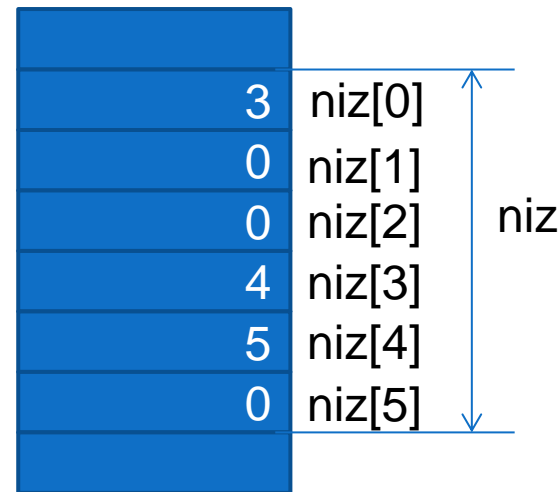
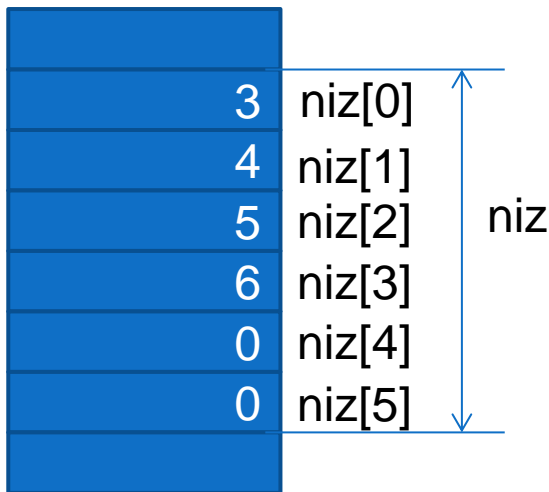
Иницијализација елемената низа 1/2

```
#define DIM 6
```

```
int niz[DIM] = {3, 4, 5, 6};
```

```
#define DIM 6
```

```
int niz[DIM] = {3, [DIM/2] = 4, 5};
```

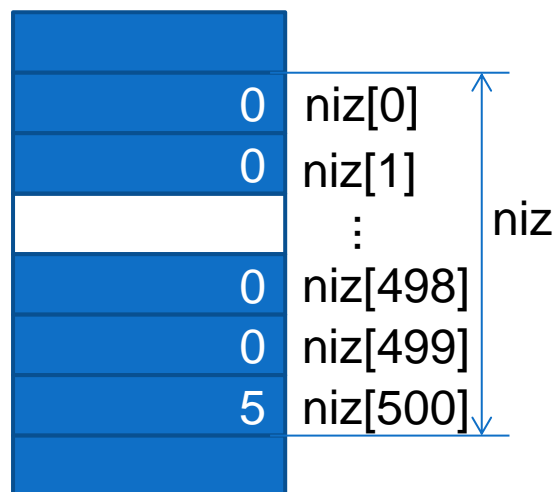


Иницијализација елемената низа 2/2

Ако је низ `niz` иницијализован следећом наредбом:

```
int niz[ ] = { [500] = 5 };
```

- Колико елемената има низ `niz`? 501
- Којим вредностима су иницијализовани елементи низа `niz`?



Садржај предавања

Низови:

- 1) Дефинисање низова
- 2) Приступање елементима низа
- 3) Иницијализација низова
- 4) **Ниске (низови знакова)**
- 5) Вишедимензиони низови
- 6) Низови као аргументи функција: разлике и сличности са скаларним аргументима

Ниске 1/2

- ❖ Ниска = низ знакова = знаковни низ = стринг (string)
- ❖ Ниска је:
 - ▶ непрекидна секвенца знакова која се завршава NULL знаком
- ИЛИ
- ▶ низ чији су елементи типа `char` и последњи елемент има вредност `'\0'`
- ❖ NULL знак = `'\0'`
- ❖ Дужина ниске = број знакова, не рачунајући завршни NULL знак
- ❖ Ниска не постоји као посебан тип у Ц-у, па је зато ниједан оператор не прихвата као операнд

Ниске 2/2

- ❖ Наредне две иницијализације су идентичне:

```
char str1[20] = "Dobar dan";
```

```
char str1[20] = {'D', 'o', 'b', 'a', 'r', ' ', 'd', 'a', 'n', '\0'};
```

- ❖ У претходном случају, дужина ниске је 9 (NULL знак се не узима у обзир), а низа је 20
- ❖ Ако се дефинише низ знакова не наводећи његову дужину, а потом се иницијализује, направиће се низ за један елемент дужи од ниске:

```
char str2[ ] = ", kolege.";
```

- ❖ У претходном случају, дужина ниске је 9, а низа је 10.

Зашто?

- ❖ Шта се исписује? `strcat()` – спаја ниске; `puts()` – приказује ниску на *stdout*
`strcat(str1, str2); // Мора: sizeof(str1) >= strlen(str1)+strlen(str2)+1. Зашто?`
`puts(str1);`

Садржај предавања

Низови:

- 1) Дефинисање низова
- 2) Приступање елементима низа
- 3) Иницијализација низова
- 4) Ниске (низови знакова)
- 5) **Вишедимензиони низови**
- 6) Низови као аргументи функција: разлике и сличности са скаларним аргументима

Вишедимензиони низови

❖ За домаћи

Садржај предавања

Низови:

- 1) Дефинисање низова
- 2) Приступање елементима низа
- 3) Иницијализација низова
- 4) Ниске (низови знакова)
- 5) Вишедимензиони низови
- 6) Низови као аргументи функција: разлике и сличности са скаларним аргументима

Низови као аргументи функција

- ❖ Када се као аргумент функције појави име низа, преводаца га имплицитно конвертује у показивач на први елемент низа. То значи да је параметар ф-је увек показивач на исити тип, као и тип елемента низа

- ❖ На пример:

```
void prikazi(int niz[ ], int brElem);
```

биће конвертовано у:

```
void prikazi(int * niz, int brElem);
```

- ❖ **Савет:** предност дефинисања параметара помоћу средњих заграда (тј. оператора адресирања) је у томе што је тако очигледно да ф-ја аргументе третира као показиваче на низ, а не само на појединачну променљиву која је типа елемента низа (`int` у примеру). Постоје и друге предности.

Шта ћете радити на вежбама

Следеће предавање

Операције над низовима:

- 1) агрегатне вредности елемената низа:
 - i. сума,
 - ii. производ и
 - iii. аритметичка средина;
- 2) екстремне вредности:
 - i. максимум и
 - ii. минимум;
- 3) ротација вредности елемената низа:
 - i. улево и удесно;
 - ii. за задати број позиција;
- 4) сортирање вредности елемената низа методом:
 - i. избора,
 - ii. мехурица,
 - iii. уметања;
- 5) тражење елемента низа који садржи задату вредност:
 - i. секвенцијално и
 - ii. бинарно;
- 6) фузија уређених (сортираних) низова.

УуБ



ФОН



Е



лабСи



па



Универзитет у Београду

факултет организационих наука

Катедра за софтверско инжењерство

лабораторија за софтверско инжењерство

ПРОГРАМИРАЊЕ I

ПРОГРАМИРАЊЕ I

Предавање #5

Проф. др **Саша Д. Лазаревић**, дипл. инж. инф.

sasa.lazarevic@fon.bg.ac.rs

Универзитет,
факултет,
катедра,
лабораторија,
предмет

