



Универзитет „Св. Кирил и Методиј“ во Скопје
ФАКУЛТЕТ ЗА ИНФОРМАТИЧКИ НАУКИ И
КОМПЈУТЕРСКО ИНЖЕНЕРСТВО

ФАКУЛТЕТ ЗА ИНФОРМАТИЧКИ НАУКИ И КОМПЈУТЕРСКО ИНЖЕНЕРСТВО

Низи од знаци

Структурно програмирање

Текстуални низи вовед

- Многу јазици имаат стандардно дефиниран податочен вид текстуална низа (стринг)
 - Типични примери вклучуваат: Basic, Turbo Pascal, Scheme/Lisp, Java
 - Тоа значи дека сите детали околу користењето на низите се имплементирани во компајлерот и извршната околина
- Текстуални низи во C++: поддржува **три вида** текстуални низи
 - Низи од знаци во стил на јазикот C - наследени од јазикот C:
 - Постои договорна конвенција текстуалните низи да се сместуваат во низи од знаци.
 - Според конвенцијата крајот на текстуалната низа се означува специфично со `'\0'` или `0` (или NUL – стандардно име за ASCII `0`, што е **различно од специјалната вредност NULL за покажувачи**)
 - String - Низи од знаци во стил на C++ (објекти), што се дел од стандардната библиотека на C++.
 - String View (od C++17)

Каратктеристики на C и C++ стиловите на текстуални низи

- Која било „буквална текстуална низа“ (string literal) е текстуална низа во C стил
 - Пр. "Happy New Year"
- Манипулацијата со текстуалните низи во C стил е со регуларни функции (нема функции членки – повеќе за ова кога ќе работиме за класи)
- Треба внимателно да се управува со меморијата за текстуалните низи во C стил
- Може да се прави конверзија меѓу двата вида текстуални низи
 - `string("programming")` претвора од текстуална низа во C стил во текстуална низа во C++ стил
 - `string.c_str()` како резултат враќа текстуална низа во C стил од текстуална низа во C++ стил

String Literals

- Се состојат од незадолжителен префикс по кој следуваат нула или повеќе знаци во двојни наводници

- Примери:

"Hello, World!\n"

"123"

"ABCDEFGH"

Префикс	Literal	Тип
Нема	Тесен/мал	const char[]
u8	UTF-8	const char8_t[]
u	UTF-16	const char16_t[]
U	UTF-32	const char32_t[]
L	Широк	const wchar_t[]

- Соседните се спојуваат

□ Пр. "Program" "ming" ⇔ "Programming"

- На крајот имплицитно се додава null терминатор (NUL)

Декларација и иницијализација

- За да се декларира текстуална низа потребно е да се декларира низа од знаци или за низата да се декларира покажувач
 - формат:
`char niza[broj];`
`char *pniza;`
- Работата со низи од знаци потоа се сведува на работа со покажувачи на null терминирани низи од знаци (низи од знаци што завршуваат со NUL)
- NUL не е дел од низата. Тој е само маркер што означува каде е крајот на низата

Декларација и иницијализација

- Пример за декларирање и иницијализација на текстуална низа

```
char s1[10];
char *s2;
char s3[10] = "foo";
char s5[5] = "foo";
char a[]="hello\n";
const char* b="hello\n";
char *s4 = s3;
```

a[0] *b	a[1] *(b+1)	a[2] *(b+2)	a[3] *(b+3)	a[4] *(b+4)	a[5] *(b+5)	a[6] *(b+6)
h	e	l	l	o	\n	\0
104	101	108	108	111	10	0

Константни текстуални низи

- Константен стринг - `const char*`

- Се резервира меморија со фиксна големина, не се менува во текот на програмата

- `const char*` `pl = "C++";`

```
char *x="abcd";  
x[1]='q';  
Нема да работи!!!
```

- `const char*` `message = "This is an example of \n"`
`"a multiline message \n";`

Пример

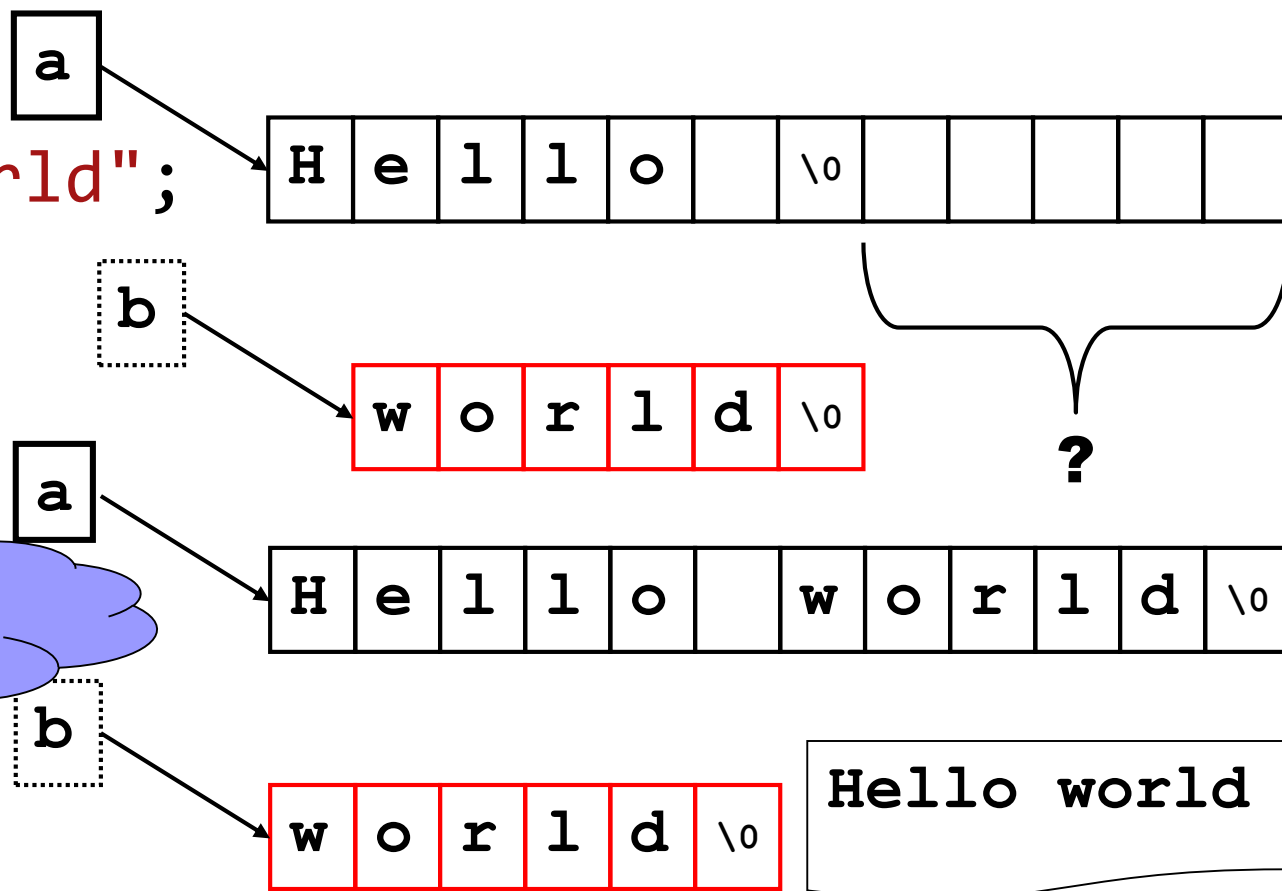
```
char a[12] = "Hello ";
```

```
char *b = "world";
```

```
strcat(a, b);
```

#include <cstring>

```
cout << a;
```



Пример

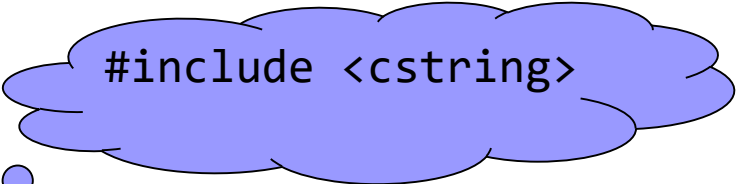
```
const int N = 10;  
char str[N];  
str = "My string";
```

- При преведување на програмата ќе се појави грешка:

- error: array type 'char[10]' is not assignable

- Решение:

```
const int N = 10;  
char str[N];  
strcpy(str, "My string");
```



```
#include <cstring>
```

Означување на крајот на текстуалните низи

- Што се случува ако низата не завршува со NUL?

```
char s1[10] = "foo";  
cout << s1;
```

Автоматски се
додава '\0' на крајот

- Ако се изврши претходниот код, тогаш на компјутерскиот екран ќе биде прикажано

foo

Означување на крајот на текстуалните низи

Забелешка:

- Ако `s1` не завршува со NUL, тогаш ќе се прикажат сите знаци додека не се најде на NUL или програмата ќе заврши со порака за грешка (segment fault etc.).

■ Пример:

```
char s1[10];
s1[0] = 'f';
s1[1] = 'o';
s1[2] = 'o';
cout << s1;
```

Sample output

```
fooâ-•VGot 2
,_VGot 2
Got 20
_VGot 2
_Got 16
Got 1
```

ПРАШАЊЕ:

Како се однесува `char s[3]="foo"`?
Нема да има место за NUL!

Текстуалните низи се покажувачи

```
char s1[10] = "foo";
```

```
char *s2 = "bar";
```

- Дали може да се изведе следното доделување?

```
s2 = s1;
```

ДА

- Дали може да се изведе следното доделување?

```
s1 = s2;
```

НЕ

Копирање на текстуални низи

```
char s1[10] = "foo";
```

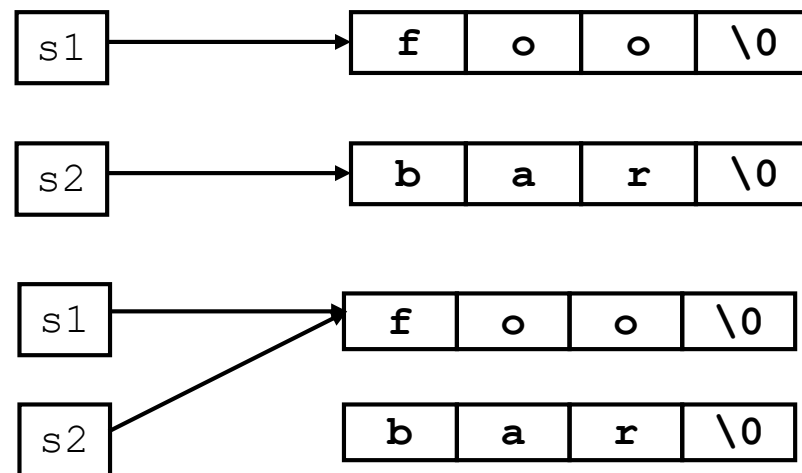
```
char *s2 = "bar";
```

- Дали наредбата `s2 = s1` овозможува копирање на содржината на текстуалните низи?

НЕ

- Но тоа сепак изгледа како да била извршена операцијата копирање
- Ако се печатат `s1` и `s2` се добива:

```
foo
foo
```



Внесување на текстуални низи (1)

■ `cin`

- Пр.

```
const int N = 10; char str[N];
cin >> str;
```

- ги игнорира празните места при читање на текстуалната низа (читање збор по збор)

■ `cin.getline()`

- Има два влезни аргумента: низата во која ќе се смести прочитаниот ред и максимален број на знаци (остава место за NUL)
- Чита додека не најде на знак за нов ред (**го чита, го отфрла и го заменува со NUL**) или додека не го достигне максималниот број на знаци
- Има и трет незадолжителен параметар - разграничувач (delimiter)
- Не ги игнорира празните места – овозможува читање на цел ред
- `const int N = 10; char str[N];`
`cin.getline(str,N);`

Внесување на текстуални низи (2)

■ `cin.get()`

- Една од варијантите наликува на `cin.getline()` и ги има истите влезни аргументи како и `cin.getline()`
- Чита додека не најде на знак за нов ред, **но не го чита** (тој останува како прв знак што следно би бил прочитан) или додека не го достигне максималниот број на знаци
- Не ги игнорира празните места – овозможува читање на цел ред
- `const int N = 10; char str[N];`
`cin.get(str,N);` //cita N-1 znak, edno mesto e potrebno za NUL

■ `char* fgets(char* str,int count,FILE* stream);`

- Чита најмногу **count-1** знак (додека не најде на EOF или `\n`) и ги сместува во низата кон која покажува `str`
- Ако најде на знак за нов ред, тогаш тој ќе биде дел од `str`, не ги игнорира празните места
- Потребно е да се вклучи `<cstdio>`
- Пр. `fgets(str, N, stdin);`

Внесување на текстуални низи – примери (1)

- Нека се дадени декларациите:

```
const int len = 10; char s[len];
```

- Пример со cin: cin >> s;

- Проблеми при читање со cin

```
int main()  
{
```

```
    const int MAX = 12;
```

```
    char name[MAX];
```

```
    char day[MAX];
```

```
    cout << "Enter your name:\n";
```

```
    cin >> name;
```

```
    cout << "Enter your favourite day of the week:" << endl;
```

```
    cin >> day;
```

```
    cout << name << ", your favourite day is " << day <<
```

```
    "." << endl;
```

```
    return 0; }
```

Што ако се внесе текстуална
низа подолга од MAX знаци?

Enter your name:

Albert Einstein

Enter your favourite day of the week:

Albert, your favourite day is Einstein.

!

Внесување на текстуални низи – примери (2)

- Нека се дадени декларациите:

```
const int len = 10; char s[len];
```

- Примери со `cin.getline()`:

```
cin.getline(s, len);  
cin.getline(s, len, '\n');  
cin.getline(s, len, '.');
```

- Примери со `cin.getline()`

```
int main(){  
    const int MAX = 20;  
    char name[MAX], day[MAX];  
    cout << "Enter your name:" << endl;  
    cin.getline(name, MAX);  
    cout << "Enter your favourite day of the week:" << endl;  
    cin.getline(day, MAX);  
    cout << name << ", your favourite day is " << day << "." <<  
endl;  
    return 0;}
```

Што ако се внесе текстуална
низа подолга од MAX знаци?

Enter your name:
Albert Einstein
Enter your favourite day of the week:
Saturday
Albert Einstein, your favourite day is
Saturday.

Внесување на текстуални низи – примери (3)

- Нека се дадени декларациите:

```
const int len = 10; char s[len];
```

- Примери со `cin.get()`:

```
cin.get(s, len);  
cin.get(s, len, '\n');  
cin.get(s, len, '.');
```

- Проблеми со `cin.get()` (од претходниот пример)

```
cin.get(name, MAX);  
cin.get(day, MAX);...
```

Овозможува да се провери како завршило читањето, односно дали бил прочитан цел ред или е исполнета низата каде се сместува прочитаната текстуална низа.

Што ако се внесе текстуална низа подолга од MAX знаци?

Нема да го прочита денот!

Решение

```
cin.get(name, MAX); // read the first line  
cin.get(); // read newline  
cin.get(day, MAX); // read the second line
```

или

```
cin.get(name, MAX).get();  
cin.get(day, MAX);
```

Внесување на текстуални низи – примери (4)

- Целиот пример:

```
#include <iostream>
using namespace std;
int main()
{
    const int MAX = 20;
    char name[MAX];
    char day[MAX];

    cout << "Enter your name:" << endl;
    cin.get(name, MAX).get();
    cout << "Enter your favorite day of the week:" << endl;
    cin.get(day, MAX).get();
    cout << name << ", your favorite day is " << day <<
    "." << endl;
    return 0;
}
```

Enter your name:
Albert Einstein
Enter your favorite day of the week:
Saturday
Albert Einstein, your favorite day is
Saturday.

Внесување на текстуални низи - примери (5)

```
const int len1 = 8, len2 = 12;
char s1[len1], s2[len2];
```

```
cin.getline(s1, len1);
if (cin.fail())
{
    cin.clear();
    cin.ignore(numeric_limits<streamsize>::max(), '\n');
}
cin.getline(s2, len2, '\n');
cout << s1 << endl << s2 << endl;
```

Го ресетира знаменцето за грешка

Ги отфрла сите останати знаци додека не наиде на знак за нов ред (вклучувајќи го и него)

Враќа true ако е поставено знаменцето за грешка

Што ако се
внесе подолга
текстуална
низа???

Влез

- This is a long string
- interesting

Излез

- This is
- interesting

Внесување на комбинација од нумерички и текстуални податоци

```
#include <iostream>
using namespace std;
int main()
{
    const int MAX = 20;
    char name[MAX];
    int year;

    cout << "What is your birth year?" << endl;
    cin >> year;
    //cin.get();
    cout << "What is your name?" << endl;
    cin.get(name, MAX);
    cout << "Year of birth: " << year << endl;
    cout << "Name: " << name << endl;
    return 0;
}
```



Внесување на текстуални низи со getchar()

```
#include <stdio>
#define N 10
char ch, str[N];
int i=0;
while((i<N-1)&&(ch=getchar())!='\n')
    str[i++]=ch;
str[i]='\0';
```

Или `ch = cin.get();`

- Ги чита и празните места се додека не се притисне ENTER или додека не се прочитаат потребниот број знаци (9)

Приказ на текстуални низи

■ Co puts()

□ Пример

```
char str[]="Hello World!";
```

```
puts(str); /* dodava oznaka za nov red po str */
```

#include <stdio>

■ Co cout

□ Пример

```
const int N = 20;
```

```
char str[N];
```

```
cin.getline(str,N);
```

```
cout << str;
```

```
cout << setw(15) << str << endl;
```

```
cout << setw(40) << str << endl;
```

Albert Einstein

Albert Einstein

Albert Einstein

Albert Einstein

Не дозволува
помалку од
должината на низата,
но може повеќе
места

Операции со текстуални низи

Датотеката `cstring (string.h)` ги содржи заглавјата на функциите што овозможуваат работа со текстуални низи

- **strlen(a)**: функцијата враќа број на знаци во низата `a`
- **strcmp(a,b)**: функцијата ги споредува низите `a` и `b`, и враќа вредноста 0 ако низите се еднакви, 1 ако првата низа е „поголема“ од втората, -1 ако првата низа е „помала“ од втората.

```
strcmp("abc", "abd") == -1;
```

```
strcmp("abf", "aba") == 1;
```

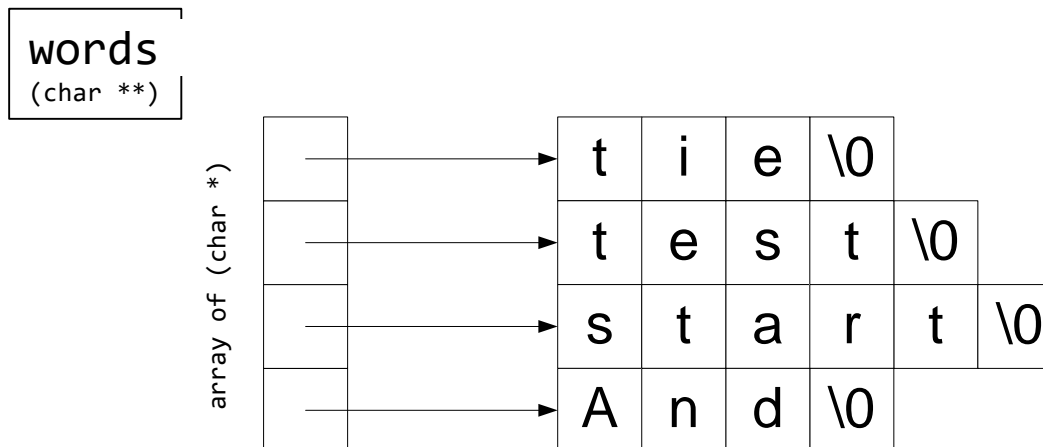
```
strcmp("abc", "abc") == 0;
```

- **strcpy(a,b)**: ја копира содржината на низата `b` во `a` притоа не проверувајќи дали `a` има место за сместување на `b`

```
char s1[10] = "foo";  
char *s2 = "foobar";  
strcpy(s2, s1);  
strcpy(s1, s2);
```


Низи од текстуални низи

```
char *words[]={ "tie", "test", "start", "And" };
```



Аргументи во main функцијата

```
#include <iostream>
using namespace std;
int main(int argc, char **argv)
{
    for(int i = 0; i < argc; i++){
        cout << argv[i] << endl;
    }
    return 0;
}
```

Аргументите во main функцијата претставуваат едноставен начин за пренесување на информација во програмата (имиња на датотеки, опции, итн.)

argc се однесува на бројот на аргументи

- цел број што го содржи бројот на аргументи што се пренесуваат во програмата

argv е низа од покажувачи кон текстуални низи во стил на C

- првиот елемент на овој вектор е името на програмата

argv[argc] секогаш е 0 (NULL pointer)

argv[1], argv[2], ..., argv[argc - 1] одговараат на опциите на командна линија

Користење на аргументите од командна линија

```
#include <iostream>

int main(int argc, char **argv) {
    int i;
    for(i = 0; i < argc; i++) {cout << argv[i] << endl;}
    return 0;
}
```

/home/joksim/tmp/sp/cm

- Телото на циклусот се повторува за секој аргумент од командната линија
- Излез од програмата

```
> g++ -o hello hello.cpp
> ./hello how are you today?
./hello
how
are
you
today?
```

```
argc=5
argv[0]="./hello"
argv[1]="how"
argv[2]="are"
argv[3]="you"
argv[4]="today?"
```

cstring (string.h)

Алтернатива: `std::fill ()`

String manipulation

- [`strcpy`](#) - copies one string to another
- [`strncpy`](#) - writes exactly n bytes to a string, copying from src or adding nulls
- [`strcat`](#) - appends one string to another
- [`strncat`](#) - appends no more than n bytes from one string to another

String examination

- [`strlen`](#) - returns the length of a string
- [`strcmp`](#) - compares two strings
- [`strncmp`](#) - compares a specific number of bytes in two strings
- [`strchr`](#) - finds the first occurrence of a byte in a string
- [`strrchr`](#) - finds the last occurrence of a byte in a string
- [`strspn`](#) - finds in a string the first occurrence of a byte not in a set of bytes
- [`strcspn`](#) - finds in a string the last occurrence of a byte not in a set of bytes
- [`strpbrk`](#) - finds in a string the first occurrence of a byte in a set of bytes
- [`strstr`](#) - finds in a string the first occurrence of a substring
- [`strtok`](#) - finds in a string the next occurrence of a token

Memory manipulation

- [`memset`](#) - fills a buffer with a repeated byte

Numeric conversions in the stdlib.h

- [`atof`](#) - converts a string to a floating-point value
- [`atoi`](#) - converts a string to an integer

Имплементација на некои функции во cstring **

```
void strcpy(char *s, char *t){
    int i=0; while((s[i]=t[i])!='\0') i++;
}
```

```
void strcpy(char *s, char *t){
    while((*s=*t)!='\0') { s++; t++; }
}
```

```
void strcpy(char *s, char *t){
    while(*s++=*t++!='\0');
}
```

```
void strcpy(char *s, char *t){
    while(*s++=*t++);
}
```

Имплементација на некои функции во cstring **

```
size_t strlen(const char *s){
    size_t n;
    for(n=0;*s!='\0';s++)n++;
    return n;
}
```

```
size_t strlen(const char *s){
    const char *p=s;
    while (*s) s++;
    return s-p;
}
```

```
char *strcat(char *s1, const char *s2){
    char *p=s1;
    while(*p) p++;
    while (*p++=*s2++);
    return s1;
}
```

Прашања?