

Содержание

string.h	3
strlen	3
strcmp	3
strcat	3
strncpy	4
strncpy_s	4
strstr	5

string.h

strlen

Возвращает количество символов строки (без учета конечного нулевого байта).

```
size_t strlen (const char* str);
```

strcmp

Функция принимает две строки и посимвольно сравнивает их. Результатом функции является целое число:

- Если строки равны, то возвращается 0
- Если первый несовпадающий символ первой строки идет в лексикографическом порядке до символа второй строки, то возвращается число меньше 0
- Если первый несовпадающий символ первой строки идет в лексикографическом порядке после символа второй строки, то возвращается число больше 0

```
int strcmp (const char* str1, const char* str2);
```

strcat

Функция strcat объединяет две строки в одну. Она имеет следующий прототип:

```
char *strcat(char *destination_str, const char *source_str);
```

Функция копирует символы из строки source_str в destination_str. При этом первая строка должна иметь достаточную длину, чтобы вместить вторую.

```
#include <stdio.h>
#include <string.h>

int main(void) {
    char destination[20];
    char source[20] = "Hello world";
    strcpy(destination, source);
    printf("%s\n", destination); // Hello world
    return 0;
}
```

strncpy

Стоит отметить, что эта функция копирует символы из `source` в `destination`, пока не встретит нулевой байт `\0`. Теоретически это может приводить к переполнению буфера. И для избежания этой проблемы есть другая функция - `strncpy_s`, которая в качестве третьего параметра принимает количество копируемых символов:

```
#include <stdio.h>
#include <string.h>

int main(void) {
    char destination[20];
    char source[20] = "Hello METANIT.COM!!";
    int n = 20; // количество копируемых символов
    strncpy(destination, source, 20);
    destination[n-1] = '\0';
    printf("%s\n", destination); // Hello METANIT.COM
    return 0;
}
```

Однако функция `strncpy` имеет другую проблему - скопированная строка НЕ заканчивается нулевым байтом, что опять же в будущем может привести к переполнению буфера. Поэтому после копирования следует вручную устанавливать нулевой байт

strncpy_s

```
errno_t strncpy_s(char * destination, size_t destmax, const char * source,
size_t srcmax);
<code>
```

Эта функция принимает дополнительные параметры - `destmax` - максимальное количество копируемых символов в буфер `destination` и `srcmax` - максимальное количество копируемых символов из строки `source`. В качестве результата функция возвращает информацию об ошибке:

```
<code c>
#include <stdio.h>
#include <string.h>

int main(void) {
    char destination[20];
    char source[] = "Hello METANIT.COM!";
    int n = 20; // количество копируемых символов
    strncpy_s(destination, n, source, n);
    printf("%s\n", destination); // Hello METANIT.COM
    return 0;
}
```

strstr

Функция strstr ищет в строке определенную подстроку. Она имеет следующий прототип:

```
char *strstr(const char* string, const char* substring);
```

Функция ищет подстроку substring в строке string и возвращает указатель на символ первого вхождения подстроки, то есть по сути адрес подстроки. Если от адреса первого символа строки вычесть адрес первого вхождения подстроки, то мы получим индекс подстроки в строке:

```
#include <stdio.h>
#include <string.h>

int main(void) {
    char text[20] = "Hello METANIT.COM!";
    char substring[14] = "METANIT.COM";
    char *substring_ptr = strstr(text, substring);
    // если подстрока найдена
    if(substring_ptr) {
        // вычисляем позицию подстроки в строке
        long position = substring_ptr - text;
        printf("Substring index: %ld\n", position);    // Substring index: 6
    } else { // если подстрока не найдена
        printf("Substring not found\n");
    }
    return 0;
}
```

From:

<https://wiki.radi0.cc/> - radi0wiki

Permanent link:

https://wiki.radi0.cc/c:c_ultimate_guide:string.h

Last update: **2025/11/09 12:07**

