

(<http://linguagemc.com.br>)

Tel. (11) 3477-5176

(11) 3508-1923

treinamento@intelletuale.com.br

[Home \(http://linguagemc.com.br\)](http://linguagemc.com.br)

[Nossa Escola \(http://www.cursosdeprogramacao.com.br/turmas\)](http://www.cursosdeprogramacao.com.br/turmas)

[Curso Presencial \(http://www.cursosdeprogramacao.com.br\)](http://www.cursosdeprogramacao.com.br)

[Contato \(http://linguagemc.com.br/fale-conosco/\)](http://linguagemc.com.br/fale-conosco/)

A biblioteca string.h

[Tweet \(https://twitter.com/share\)](https://twitter.com/share) 

Por: Eduardo Casavella

A biblioteca string.h da linguagem C, contém uma série de funções para manipular strings.

Neste artigo veremos como:

- Copiar strings em C usando strcpy e strncpy;
- Concatenar strings em linguagem C usando strcat e strncat;
- Descobrir o tamanho de uma string em C usando strlen();
- Comparar strings em C usando strcmp();

Construiremos exemplos de programas em C para exemplificar as funções mais utilizadas.

Nas funções a seguir, utilizamos o seguinte padrão para descrever a sintaxe:

string_destino: é a string que recebe a cópia

string_origem: é a string que contém a cadeia de caracteres que será copiada.

strcpy

Sintaxe:

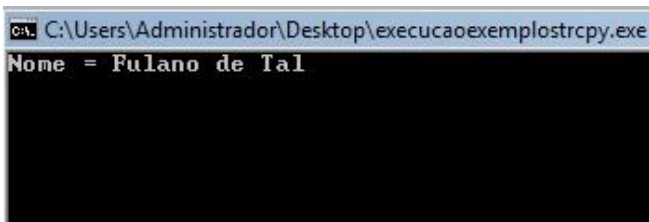
```
strcpy(string_destino, string_origem);
```

Realiza a cópia do conteúdo de uma variável a outra.

Obs: Ambas devem ser strings.

```
1  #include <stdio.h>
2  #include <string.h> //necessário para strcpy
3  #include <conio.h>
4  int main (void)
5  {
6      char nome[15];
7
8      strcpy(nome, "Fulano de Tal");
9      //strcpy(string_destino, string_origem);
10     //note que a string de destino é nome
11     //a string de origem é "Fulano de Tal"
12
13     printf("Nome = %s", nome);
14
15     getch();
16     return 0;
17 }
```

Tela de execução



(<http://linguagemc.com.br/wordpress/wp-content/uploads/2012/04/execucaoexemplostrcpy1.jpg>)

Execução do exemplo strcpy

strncpy

Sintaxe:

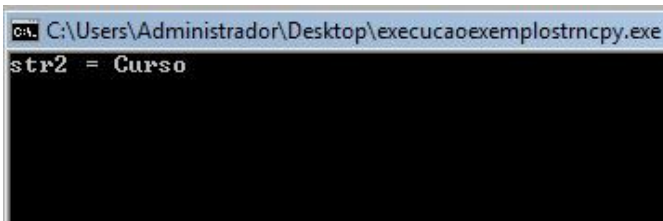
strncpy(string_destino, string_origem, tamanho);

Realiza a cópia do conteúdo de uma variável a outra, porém, deve ser especificado o tamanho a ser copiado.

Obs: Tanto a variável de destino como a de origem devem ser strings.

```
1  #include <stdio.h>
2  #include <string.h> //necessário para strncpy
3  #include <conio.h>
4  int main (void)
5  {
6      char str1[8] = "Curso C";
7      char str2[5];
8
9      strncpy(str2, str1, 5);
10
11     str2[5] = '\0';
12     printf("str2 = %s\n", str2);
13
14     getch();
15     return 0;
16 }
```

Tela de execução



(<http://linguagemc.com.br/wordpress/wp-content/uploads/2012/04/execucaoexemplostrncpy.jpg>)

Execução do exemplo strncpy

strcat

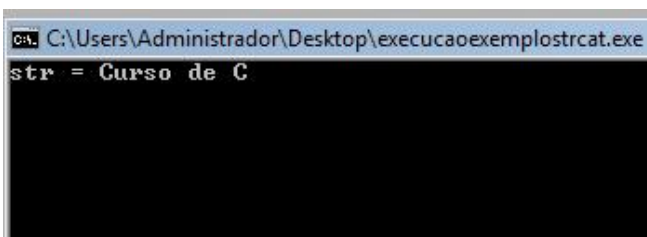
Sintaxe

`strcat(string_destino, string_origem);`

Realiza a concatenação do conteúdo de uma variável a outra.
Ambas devem ser strings.

```
1 //Exemplo strcat
2 #include <stdio.h>
3 #include <string.h> //necessário para strcat
4 #include <conio.h>
5 int main (void)
6 {
7     char str[10] = "Curso";
8     strcat(str, " de C");
9     //Concatena a string " de C" com o conteúdo da string str
10
11     printf("str = %s\n", str);
12     //Será exibido curso de C
13
14     getch();
15     return 0;
16 }
```

Tela de execução



(<http://linguagemc.com.br/wordpress/wp-content/uploads/2012/04/execucaoexemplostrcat.jpg>)

Execução do exemplo strcat

strncat

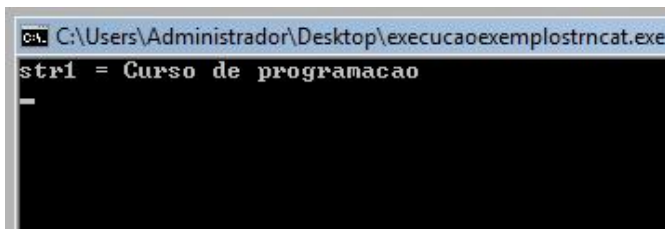
Sintaxe:

`strncat(string_destino, string_origem, tamanho);`

Realiza a concatenação do conteúdo de uma variável a outra,
porém, deve ser especificado o tamanho a ser concatenado.
Ambas devem ser strings.

```
1 //Exemplo strncat
2 #include <stdio.h>
3 #include <string.h> //necessário para strncat
4 #include <conio.h>
5 int main (void)
6 {
7     char str1[20] = "Curso";
8     char str2[17] = " de programacao C";
9
10    strncat(str1, str2, 15);
11    //concatena a string1 com 15 posições da string2
12
13    printf("str1 = %s\n", str1);
14    //Será exibido Curso de Programação
15
16    getch();
17    return 0;
18 }
```

Tela de execução



(<http://linguagemc.com.br/wordpress/wp-content/uploads/2012/04/execucaoexemplostrncat.jpg>)

Execução do exemplo strncat

strlen

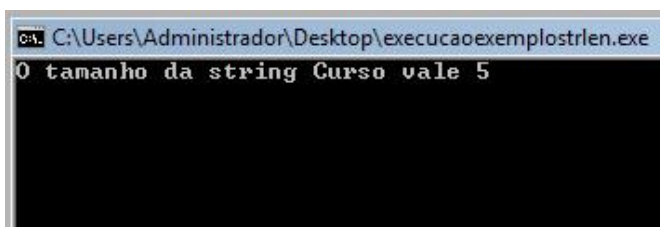
Sintaxe:

variável do tipo inteiro = strlen(string);

Determina o tamanho de uma string.

```
1 //Exemplo strlen
2 #include <stdio.h>
3 #include <string.h> //necessário para strlen
4 #include <conio.h>
5 int main (void)
6 {
7     char str[5] = "Curso";
8     int tamanho;
9
10    tamanho = strlen(str);
11
12    printf("O tamanho da string %s vale %d\n", str, tamanho);
13
14    getch();
15    return 0;
16 }
```

Tela de execução



(<http://linguagemc.com.br/wordpress/wp-content/uploads/2012/04/execucaoexemplostrlen.jpg>)

content/uploads/2012/04/execucaoexemplostrlen.jpg)

Execução do exemplo strlen

strcmp

Sintaxe:

variável do tipo inteiro = strcmp(string1, string2);

Compara o conteúdo de duas strings;

Possíveis valores de retorno:

0: conteúdo das strings são iguais

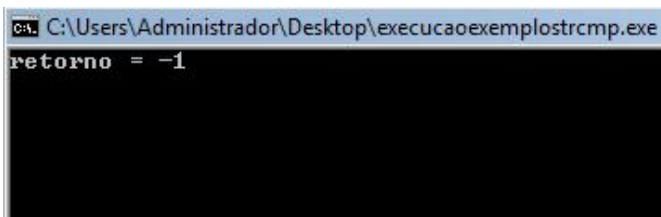
< 0: conteúdo da string1 é menor do que string2

> 0: conteúdo da string1 é maior do que string2

Exemplo de programa usando strcmp:

```
1  #include <stdio.h>
2  #include <string.h> //necessário para strcmp
3  #include <conio.h>
4  int main (void)
5  {
6      char str1[3] = "abc";
7      char str2[3] = "abd";
8      int retorno;
9
10     retorno = strcmp(str1, str2);
11     printf("retorno = %d\n", retorno);
12     //mostra o retorno da função strcmp
13
14     getch();
15     return 0;
16 }
```

Tela de execução



(<http://linguagemc.com.br/wordpress/wp-content/uploads/2012/04/execucaoexemplostrcmp.jpg>)

Execução do exemplo strcmp

strncmp

Sintaxe:

variável do tipo inteiro = strncmp(string1, string2, tamanho);

Também faz a comparação do conteúdo de duas strings, porém, deve ser especificado o tamanho a ser comparado;

Possíveis valores de retorno:

0: conteúdo das strings são iguais

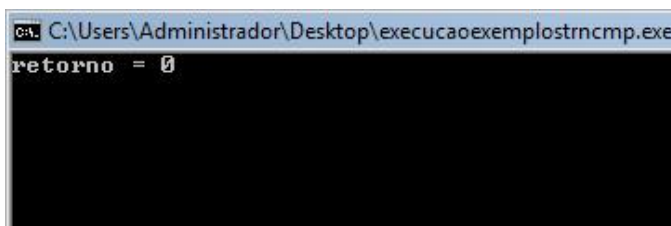
< 0: conteúdo da string1 é menor do que string2

> 0: conteúdo da string1 é maior do que string2

Exemplo de programa usando strncmp:

```
1 //Exemplo strncmp
2 #include <stdio.h>
3 #include <string.h> //necessário para strncmp
4 #include <conio.h>
5 int main (void)
6 {
7     char str1[10] = "Curso de C";
8     char str2[22] = "Curso de programação C";
9     int retorno;
10
11     retorno = strncmp(str1, str2, 5);
12
13     printf("retorno = %d\n", retorno);
14     getch();
15     return 0;
16 }
```

Tela de execução



(<http://linguagemc.com.br/wordpress/wp-content/uploads/2012/04/execucaoexemplostrncmp.jpg>)

Execução do exemplo strncmp

Obs: Os códigos dos programas utilizados neste artigo foram gentilmente cedidos pelo Prof. Fabio Schmitt. Até a Próxima!

Tweet (<https://twitter.com/share>) 

11 respostas a *A biblioteca string.h*