

Linguagem e Técnicas de Programação



Manipulação de Strings

A biblioteca `<string.h>` em C fornece várias funções para manipular strings.

Abaixo estão as principais funções, com explicações e exemplos de uso:

Manipulação de Strings em C

Manipulação de Strings em C

1. strlen()

Descrição: Retorna o comprimento de uma string.

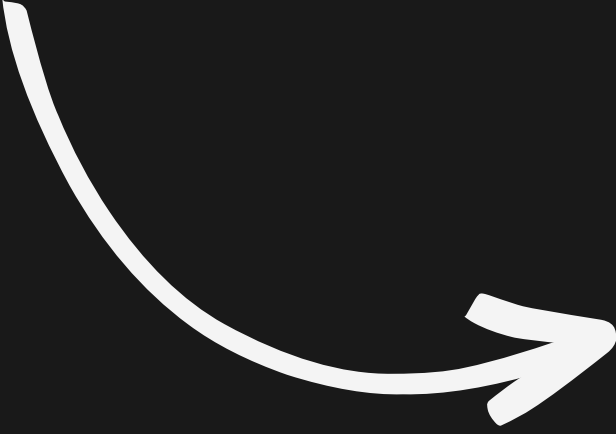


```
1 char str[] = "Hello";  
2 printf("Length: %lu\n", strlen(str));  
3 // Saída: Length: 5
```

Manipulação de Strings em C

2. strcpy()

Descrição: Copia uma string para outra.

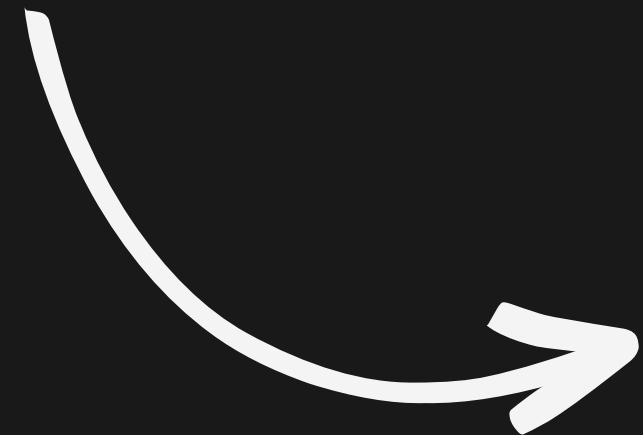


```
1 char source[] = "World";  
2 char destination[20];  
3 strcpy(destination, source);  
4 printf("%s\n", destination);  
5 // Saída: World
```

Manipulação de Strings em C

3. strcat()

Descrição: Concatena duas strings.




```
1 char str1[20] = "Hello, ";  
2 char str2[] = "World!";  
3 strcat(str1, str2);  
4 printf("%s\n", str1);  
5 //Saída: Hello, World!
```

Manipulação de Strings em C

4. strcmp()

Descrição: Compara duas strings.




```
1 if (strcmp("apple", "orange") < 0) {  
2     printf("apple vem antes de orange\n");  
3 }
```

Manipulação de Strings em C

5. strncpy()

Descrição: Copia n caracteres de uma string para outra.




```
1 char src[] = "World";  
2 char dest[20];  
3 strncpy(dest, src, 3);  
4 dest[3] = '\0'; // Adicionando terminador nulo manualmente  
5 printf("%s\n", dest); // Saída: Wor
```

Manipulação de Strings em C

6. strncat()

Descrição: Concatena n caracteres de uma string a outra.




```
6 char str1[20] = "Hello, ";  
7 char str2[] = "World!";  
8 strncat(str1, str2, 3);  
9 printf("%s\n", str1); // Saída: Hello, Wor
```


Manipulação de Strings em C

7. strncmp()

Descrição: Compara até n caracteres de duas strings.




```
6 | if (strncmp("apple", "applf", 4) == 0) {  
7 |     printf("Os primeiros 4 caracteres são iguais!\n");  
8 | }
```

Manipulação de Strings em C

8. strchr()

Descrição: Localiza a primeira ocorrência de um caractere em uma string.



```
char str[] = "Hello, World!";  
char *ptr = strchr(str, 'W');  
if (ptr) {  
    printf("Caractere encontrado: %c\n", *ptr);  
    // Saída: Caractere encontrado: W  
}
```

8. strchr()

O uso do asterisco * no item 8, na linha `char *ptr = strchr(str, 'W');`, indica que `ptr` é um ponteiro para um caractere (`char`). Em C, um ponteiro é uma variável que armazena o endereço de memória de outra variável. No caso da função `strchr()`, ela retorna o endereço do primeiro caractere na string que corresponde ao caractere fornecido como argumento. Se o caractere não for encontrado, a função retorna `NULL`.

Manipulação de Strings em C

8. strchr()

Aqui está uma explicação passo a passo:

1. `char str[] = "Hello, World!";` – Esta linha declara e inicializa uma string.
2. `char *ptr = strchr(str, 'W');` – Esta linha faz o seguinte:
 - `strchr(str, 'W')` procura o caractere 'W' na string `str`. Se encontrado, retorna um ponteiro para a primeira ocorrência de 'W' em `str`.
 - `char *ptr` declara um ponteiro para um caractere. Este ponteiro armazenará o endereço retornado por `strchr()`.
3. `if (ptr) { printf("Caractere encontrado: %c\n", *ptr); }` – Esta linha verifica se `ptr` não é NULL (ou seja, o caractere foi encontrado). Se `ptr` não for NULL, imprime o caractere no endereço apontado por `ptr` usando `*ptr`.

**Manipulação
de Strings
em C**

Um ponteiro em C é uma variável que armazena o **endereço** de outra variável. Isso permite que você manipule a memória diretamente e acesse dados de forma mais flexível e eficiente. Aqui estão alguns conceitos chave sobre ponteiros:

O que é um
ponteiro (*)?


int *p; // p é um ponteiro para um inteiro

int x = 10; p = &x; // p agora armazena o endereço de x

Manipulação de Strings em C

9. strstr()

Descrição: Localiza a primeira ocorrência de uma substring em uma string.




```
6 char str[] = "Hello, World!";
7 char *ptr = strstr(str, "World");
8 if (ptr) {
9     printf("Substring encontrada: %s\n", ptr); // Saída: Substring encontrada: World!
10 }
11
```

10. strtok()

Descrição: Divide uma string em tokens com base em delimitadores.

Manipulação de Strings em C



```
char str[] = "Hello, World!";
char *token = strtok(str, ",");
while (token != NULL) {
    printf("%s\n", token);
    token = strtok(NULL, ",");
}
// Saída:
// Hello
// World!
```