



INSTITUTO FEDERAL
Sul-rio-grandense

Campus
Gravataí

Strings em C

Prof. Hunder Evertto Correa Junior



Curso Superior de Tecnologia em
Análise e Desenvolvimento
de Sistemas

Introdução

- Em C não existe um tipo de dado string, no seu lugar é utilizado uma matriz de caracteres. Uma string é uma matriz tipo char que termina com '\0'.
- Por essa razão uma string deve conter uma posição a mais do que o número de caracteres que se deseja.
- Constantes strings são uma lista de caracteres que aparecem entre aspas, não sendo necessário colocar o '\0', que é colocado pelo compilador.
- Strings são vetores de chars. Nada mais e nada menos. As strings são o uso mais comum para os vetores.

- `char nome[]="GREMIO"`
- Índice:

0	1	2	3	4	5	6
G	R	E	M	I	O	\0

Declarando Strings

- Para declarar uma string, podemos usar o seguinte formato geral:
- *char nome_da_string[tamanho];*
- *Ex.*
- `char nome[20];`
- Ou
- `char nome[]="Hunder Correa";`

Função gets()

- É utilizada para leitura de uma string através do dispositivo padrão, até que o ENTER seja pressionado. A função gets() não testa limites na matriz em que é chamada.
- Ex:
- main()
- {
- char str[80];
- gets(str);
- printf("%s",str);
- }

Função puts()

- Imprime uma string na tela.
- puts()
- Ex.
- puts("IFSUL");

Função `strlen()`

- A função **`strlen()`** retorna o comprimento da string fornecida. O terminador nulo não é contado. Isto quer dizer que, de fato, o comprimento do vetor da string deve ser um a mais que o inteiro retornado por **`strlen()`**.
- Sua forma geral é:
- *`strlen (string);`*

Exemplo

- `#include <stdio.h>`
- `#include <string.h>`
- `int main ()`
- `{`
- `int size;`
- `char str[100];`
- `printf ("Entre com uma string: ");`
- `gets (str);`
- `size=strlen (str);`
- `printf ("\n\nA string que voce digitou tem tamanho %d",size);`
- `return(0);`
- `}`

Função `strcat()`

- A função **`strcat()`** tem a seguinte forma geral:
- *`strcat (string_destino,string_origem);`*
- Concatena dois strings: A string de origem permanecerá inalterada e será anexada ao fim da string de destino.

Exemplo

- `#include <stdio.h>`
- `#include <stdlib.h>`
- `#include <string.h>`
- `main()`
- `{`
- `char str[80], str2[80];`
- `printf("Digite um nome:");`
- `gets(str);`
- `printf("Digite um Apelido:");`
- `gets(str2);`
- `printf("A expressão digitada foi %s",strcat(str,str2));`
- `}`

Resultado

Digite um nome:Zezinho

Digite um Apelido:trovador

A expressão digitada foi Zezinhotrovador

Process returned 40 (0x28) execution time : 20.052 s

Press any key to continue.

Comparando Strings

- **strcmp: int strcmp(const char *lhs, const char *rhs);**
- Essa função compara duas strings segundo sua ordem alfabética e retorna um inteiro.
- Se esse inteiro for negativo, é porque a primeira string é menor que a segunda.(a<b)
- Se retornar um inteiro positivo, é porque a segunda string é menor que a primeira.(a>b)
- Se retornar 0, é porque as strings são idênticas.(a==b)
- **Ex. If (strcmp(x,y)==0) // strings idênticas**

Biblioteca para Strings

- Para utilizarmos as funções para string é necessário incluirmos a biblioteca:
- `string.h`
- Ex.
- `#include <string.h>`

Exemplo

- Faça um programa que leia uma palavra e a imprima na tela:
- `#include <stdio.h>`
- `#include <stdlib.h>`
- `#include <locale.h>`
- `#include <string.h>`
- `main()`
- `{`
- `setlocale(LC_ALL, "Portuguese");`
- `char nome[15];`
- `printf("Digite um nome: ");`
- `gets(nome);`
- `puts(nome);`
- `system("PAUSE");`
- `}`