

UEMG - Universidade do Estado de Minas Gerais - Unidade Frutal

Curso - Sistemas de Informação, 1º Período, Noturno

Disciplina - Introdução à Programação

Docente - Profº Sergio Carlos Portari Junior

Discente - Carlos Augusto de Oliveira Alves

Trabalho - Trabalho de Vetores

Data - 24/08/2021

String em C utilizando vetores

O vetor na linguagem C é uma estrutura de dados utilizada para armazenar valores de um mesmo tipo. É importante lembrar que na linguagem C, um vetor é indexado a partir da posição 0, sendo assim a ultima posição do Vetor é n-1, sendo “n” o número de posições armazenadas.

Uma string é uma sequência de caracteres, usada para representar textos. Na linguagem C, a string é executada como um vetor, já que a linguagem não possui um tipo específico de dados para serem utilizados.

- **Declaração de uma String**

Em C, a declaração de uma string são denominadas pelo caractere (“ ”), e as aspas duplas (“...”) são usadas para declarar uma string constante. E para representar uma string devemos utilizar o vetor char, do tipo caracteres.

- **Exemplo:**

```
Char clientes[10];
```

Este comando irá criar uma variável a partir do vetor de caracteres (char), chamada clientes e o número entre colchetes é a quantidade de posições armazenadas dentro deste vetor.

Após criarmos o vetor, podemos atribuir um valor para ele:

```
char clientes[10] = "Carlos";
```

```
char clientes[10] = {'C', 'A', 'R', 'L', 'O', 'S'};
```

Nota-se que Carlos, possui apenas 6 caracteres e nosso vetor comporta até 9 caracteres mais o valor indexado ao nulo “\0”, sendo assim ainda temos 3 espaços livres.

Podemos também criar um vetor sem definir o valor para ele:

```
char clientes[] = "Carlos";
```

Assim, o próprio programa irá calcular os espaços armazenados automaticamente.

Podemos também criar strings variáveis com o “printf” e “scanf”:

```
#include <stdio.h>

char clientes[];

int main()
{
    printf ("Digite um nome para ser armazenado: ");
    scanf ("%s", clientes);
    printf ("\nO nome digitado foi: %s\n" , clientes);
    return 0;
}
```

- **Manipulação de String**

A manipulação das strings normalmente é feita pela biblioteca padrão C chamada “string.h”. As funções mais utilizadas são:

strcpy()

strcpy (dest, src)

Copia a string de origem (src) para o local indicado como destino (dest). Observe que a área de memória de destino deve ter sido previamente alocada, e ser grande o suficiente para conter toda a string.

```
#include <stdio.h>

#include <string.h>

int main (void)
{
    char nome[10];
    strcpy(nome, "Carlos");
    printf("Nome = %s", nome);
    return 0;
}
```

strcat()

strcat (dest, src)

Liga a string de origem ao final da string de destino, para espaço previamente alocado, e com espaço suficiente.

```
#include <stdio.h>#include <string.h>int main (void){char str[10] = "Curso";strcat(str, " de C");printf("str = %s\n", str);return 0;}
```

strchr()

strchr (str, c)

Retorna o endereço (um ponteiro) para a primeira ocorrência do caractere "c" na *string* "str", ou NULL se não encontrar.

```
#include <stdio.h>

#include <string.h>

int main ()

{

    const char str[] = "String";

    const char c = 'i';

    char *p;

    p = strchr(str, c); printf("Comecando a String por %c fica: %s", c, p);

    return 0;

}
```

strcmp()

strcmp (st1, st2)

Compara as duas strings indicadas, retornando 0 se forem iguais, <0 se st1<st2 e >0 se st1>st2.

```
#include <stdio.h>

#include <string.h>

int main (void)

{

    char str1[4] = "abc";

    char str2[4] = "abd";

    int retorno;

    retorno = strcmp(str1, str2);

    printf("retorno = %d\n", retorno);

    return 0;

}
```

strcmp()

strcmp (st1, st2)

Compara as duas strings indicadas (ignorando a diferença de valores dos caracteres em maiúsculo e minúsculo, retornando 0 se forem iguais, <0 se st1<st2 e >0 se st1>st2.

```
#include <stdio.h>

#include <string.h>

int main (void)

{

char str1[4] = "abc";

char str2[4] = "abd";

int retorno;

retorno = strcmp(str1, str2);

printf("retorno = %d\n", retorno);

return 0;

}
```

strlen()

strlen (str)

Informa o número de caracteres da string (sem considerar o " final).

```
#include <stdio.h>

#include <string.h>

int main (void)

{

char str[5] = "Curso";

int tamanho;

tamanho = strlen(str);

printf("O tamanho da string %s vale %d\n", str, tamanho);

return 0;

}
```

