**UEMG - Universidade do Estado de Minas Gerais - Unidade Frutal**

**Curso - Sistemas de Informação, 1° Período, Noturno**

**Disciplina - Introdução à Programação**

**Docente - Prof° Sergio Carlos Portari Junior**

**Discente – José Luiz Vieira Batista**

**Trabalho - Trabalho de Matrizes e Vetores**

**Data – 27/08/2021**

**String baseadas em Vetores**

Uma string é uma sequência de caracteres que permite representar nomes, endereços e outras informações textuais. Em C são implementadas como vetores de caracteres terminados pelo caractere especial ”, que deve ser considerado quando medir o tamanho do vetor. As aspas duplas “...” são usadas para declarar strings constantes.

**Exemplo:**

char cod[10] ; // variável com até 9 caracteres mais o '\0'

char \*endereco ; // endereço que aponta para uma string, mas sem alocar espaço

endereco = "ARROZ" ; // string constante, com '\0' no fim

E também deve-se observar que uma string é um vetor de caracteres, sendo assim as duas declarações abaixo são equivalentes:

char cod[] = "ARROZ" ;

char cod[] = { 'A', 'R', 'R', 'O', 'Z'} ;

Nota-se que ARROZ, possui apenas 5 caracteres e nosso vetor comporta até 9 caracteres mais o valor indexado ao nulo “\0”, sendo assim ainda temos varios espaços livres.

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

char cod[] = “ARROZ”;

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

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

#include char cod[];

int main() { printf ("codigo para ser armazenado: ");

scanf ("%s", cod);

printf ("\nO codigo digitado foi: %s\n" , cod);

return 0; }

**Manipulação**

A manipulação de strings é geralmente feita através das funções disponíveis na biblioteca padrão C, que estão declaradas no cabeçalho ''string.h''.

**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(cod, "ARROZ");

printf("cod = %s", cod);

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] = "nome";

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.

#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;

}

**stricmp()**

stricmp (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.

#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;

}