Introdução à Computação Estrutura de Dados Composta -Unidimensional: Vetor de Caracteres

Prof. Ernesto Veiga ernestoveiga@ufg.br





String - Vetor de Caracteres



Na prática, as **strings** são usadas para representar **textos**.

Em linguagem C, ao contrário de outras linguagens (ex.: Java, Python, etc.), não existe um tipo de dados string nativo.

Para representar uma string em C, devemos criar um **vetor de caracteres**, ou seja um **vetor de valores do tipo char**.

O byte nulo '\0' é interpretado como uma sentinela que marca o fim da parte relevante do vetor

ASCII e as strings



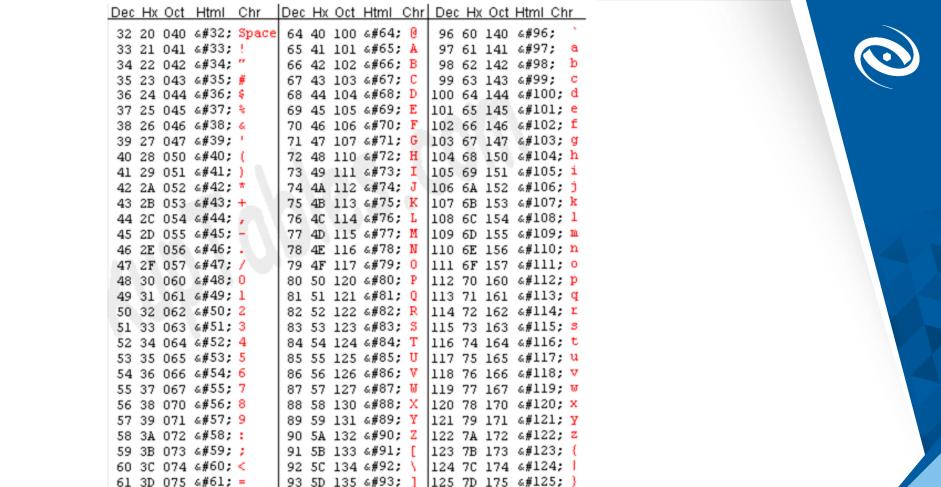
Cadeias de caracteres são representadas por strings.

É preciso criar uma correspondência entre caracteres e bytes.

A correspondência mais antiga e mais básica é a tabela ASCII, que associa a cada caractere do alfabeto ASCII um byte entre 00000000 e 01111111

Por exemplo, a cadeia de caracteres ABC é representada por 65 66 67.

```
char a;
a = 65;
printf(" %c", a);
saída: A
```



62 3E 076 > >

63 3F 077 4#63; ?

Source: www.LookupTables.com

94 5E 136 6#94; ^ | 126 7E 176 6#126; ~ 95 5F 137 6#95; _ | 127 7F 177 6#127; DEL

Recapitulando os usos de vetor...



Declaração de vetor:

float vetor[100];

Atribuindo valores ao vetor:

```
vetor[0] = 20; idade[5] = 22;
```

Preenchendo um vetor:

```
for(i=0; i<100; i++)
scanf("%f", &vetor[i]);
```

Mostrando os elementos do vetor:

```
for(i=0; i<100; i++)
printf("%f ", vetor[i]);
```



Como declarar uma string

Declaração de vetor:

float vetor[100]; int idade[20];

Declaração de string:

char nome[30];



Inicialização de uma string

Atribuindo valores ao vetor:

```
int vetor[0] = 20;
int idade[5] = {10, 20, 30, 40, 50};
```

Atribuindo valor à string:

```
char nome[30] = "Ernesto Fonseca Veiga";
char nome[30] = {97,195,167,195,163,111,0};
```

Saída: Ernesto Fonseca Veiga

Saída: ação

Obs.: o par de bits 195 167 representa o ç e o par de bits 195 163 representa o ã



Preenchendo uma string

Preenchendo um vetor:

```
for(i=0; i<100; i++)
scanf("%f", &vetor[i]);
```

Preenchendo uma string:

```
scanf ("%s", nome); // sem & scanf("%[^\n]", nome); // lê string com espaços (até achar '\n') fgets(nome, 29, stdin); //identificador, tamanho-1, stdin
```

Mostrando uma string



Mostrando os elementos do vetor:

```
for(i=0; i<100; i++)
printf("%f", vetor[i]);
```

Mostrando uma string:

```
for(count=0; count < 20; count++)
    printf("%c",curso[count]);</pre>
```

```
for (int i = 0; i < strlen(nome); i++) // imprime o nome, um char por vez putchar (nome[i]);
```

```
printf(" %s", nome);
puts(nome);
```

Manipulação de Strings



Em C temos uma biblioteca <string.h> para manipular essa estrutura

- int strlen(src): retorna o número de caracteres da string
- char * strcpy (dest, orig): copia a string da orig para o dest
- char * strncpy (dest, orig, n): copia n caracteres da orig para o destino
- int strcmp (st1, st2): compara duas strings e retorna 0 se forem iguais
- int strncmp (str1, str2,n): idem, mas compara apenas n caracteres
- char * strcat (dest, src): concatena a string src ao final de dest
- **char * strncat (strto, strfrom, n):** idem, mas concatena apenas n caracteres

Introdução à Computação Estrutura de Dados Composta -Unidimensional: Vetor de Caracteres

Prof. Ernesto Veiga ernestoveiga@ufg.br



