

Strings em C

Aula 07

Marcos Silvano Almeida
Departamento de Computação
UTFPR Campo Mourão

Strings

- Representam o **tipo “texto”** em linguagens de programação
- Já utilizamos frequentemente valores literais de strings em C:
 - Literais strings são sempre envoltas por **aspas**.
 - **Valores literais** são aqueles que escrevemos diretamente no código.

Em todos estes casos, estamos escrevendo as strings (valores literais) diretamente no código.

```
printf("CALCULO DE MEDIA\n");
```

```
scanf("%d %d", &a, &b);
```

```
printf("O resultado é: %d", res);
```

Strings em C

- Uma string é armazenada como sequências de caracteres
 - Essencialmente, um **vetor de chars**
- Em C, uma string é um vetor de caracteres terminado em nulo
 - `NULL` \Leftrightarrow `0` \Leftrightarrow `'\0'` \Leftrightarrow falso
 - Comumente chamadas de strings terminadas em nulo

```
char str1[] = "String C";
```

Forma mais usada

Terminada em nulo

```
char str2[] = {'S','t','r','i','n','g',' ','C','\0'};
```

```
char str3[] = {'S','t','r','i','n','g',' ','C', 0};
```

```
char str4[] = {'S','t','r','i','n','g',' ','C', NULL};
```

Utilizando strings

- Deve-se sempre lembrar que a string em C nada mais é do que um vetor de caracteres com NULL (0 ou '\0') ao final.
 - O tamanho do vetor é sempre a quantidade de caracteres + 1 (NULL)
 - Quando utilizamos declaração com inicialização, o compilador gera um vetor com o tamanho adequado

```
// a string possui 13 de tamanho = 12 chars + '\0'
char str1[] = "João Ninguém";
printf("Bem vindo %s\n", str1);
```

Para imprimir uma variável string,
utilizamos o formatador %s no printf()

- Acesso à elementos da string: da mesma forma que no vetor

```
str[0] = 'P';    // modificar caractere da string
char c = str[3]; // acessar caractere da string
```

Varrendo strings

- Na maior parte das situações, vamos varrer a string com um laço **for**
 - Não é necessário conhecer a quantidade de chars, basta atentar para o char `'\0'`

```
char str[] = "Texto de teste de string em C";
```

```
// realizamos o laço até encontrar '\0' ou 0
```

```
for (int i = 0; str[i] != 0; i++) {
```

```
    if (str[i] == ' ') {
```

```
        str[i] = '-';
```

```
    }
```

```
}
```

```
printf("String: %s\n", str);
```

SAÍDA:

String: Texto-de-teste-de-string-em-C

Passando string para função

- Ao contrário de um vetor de valores, **não é necessário declarar um parâmetro** para o tamanho da string, basta percorrer até 0 ou '\0'

```
void spaceToDash(char str[]) { // str é uma referência à string passada à função
    for (int i = 0; str[i] != 0; i++) {
        if (str[i] == ' ') {
            str[i] = '-';
        }
    }
}

void main() {
    char str[] = "Texto de teste de string em C" ;
    spaceToDash(str);
    printf("String: %s\n", str);
    // Provoca falha de segmentação, pois um literal string não pode ser
    modificado
    spaceToDash("Olá mundo da linguagem C" );
```

Cuidado ao passar uma string literal para uma função que a modifica!