

Funções que Manipulam Strings

Módulo 7 Aula 5

Linguagem C, o Curso Definitivo WR Kits

Autor: Dr. Eng. Wagner Rambo

Para passar uma *string* como parâmetro para uma função, pode-se recorrer ao mesmo método utilizado com vetores, pois utilizamos vetores para armazenar as *strings*. Para relembrar, confira a sintaxe no Box 1.

```
void func(char str[]); /* protótipo da função que recebe a string */
main()
{
    char s[30];          /* string declarada na função principal */
    func(s);             /* passando a string para a função */
} /* end main */
```

Box 1 - Passando a string como parâmetro para função.

No entanto, é mais comum utilizarmos o operador *. Este operador é a designação de um ponteiro em C, que iremos estudar detalhadamente no Módulo 8. Por enquanto, entenda que podemos utilizar * para passar tanto *strings* como vetores, para as funções que desenvolvermos. No Box 2 apresentamos um exemplo de função que recebe uma *string* str e também uma função que além de receber uma *string* também retorna um *string*.

```
void func(char *str); /* protótipo da função que recebe a string
                     com o operador *. A passagem através da
                     função principal, é exatamente igual ao
                     Box 1. */

char *sret(char *str); /* protótipo da função que recebe uma
                       string e também retorna uma string. */
```

Box 2 - Utilizando o operador * para passar a string para a função.

A biblioteca *string.h* apresenta diversas funções úteis para manipulação de *strings*, das quais, podemos destacar:

strcat: concatena duas *strings*.

strcmp: compara duas *strings*.

strcpy: copia o conteúdo de uma *string* para outra.

strlen: devolve o tamanho da *string* sem levar o caractere nulo em consideração.

Implementaremos manualmente estas e mais algumas *strings* para entendermos o seu total funcionamento.

Exercício proposto: Desenvolva o projeto de um interpretador para as seguintes instruções:

RUN – rodar o interpretador.

RUN SOUND – rodar o interpretador com “bips”.

REP – define o número de repetições de uma frase salva previamente com o comando TXT. Recebe como argumento o número de repetições desejadas.

TXT – comando para entrada de nova frase. O argumento é a frase-texto que o usuário quiser que o interpretador rode pelo número de repetições definido.

CLEAR – limpa a tela do console.

CLEAR ALL – limpa a tela e restaura variáveis.

EXIT – encerra o interpretador.

HELP – apresenta a versão atual do interpretador e breve explicação sobre os seus comandos.

Bibliografia: DAMAS, Luís; Linguagem C, décima edição.

Disponível em: <https://amzn.to/3nGdlbN>