

MC102 - Algoritmos e programação de computadores

Aula 17: Manipulação de Strings

Cadeias de caracteres

- Uma cadeia de caracteres, mais conhecida como *string*, é uma sequência de letras e símbolos, onde os símbolos podem ser espaços em branco, dígitos e vários outros como pontos de exclamação e interrogação, símbolos matemáticos, etc.
- Em C, uma cadeia de caracteres é representada por um vetor de variáveis do tipo `char` e é terminada com o marcador `'\0'`.

Declarando uma cadeia de caracteres

```
char texto [TAMANHO + 1];
```

- Devemos utilizar uma posição além do tamanho desejado para que possa ser colocado o marcador '`\0`' no final da *string* quando essa tiver o tamanho máximo.

Lendo uma string do teclado

- Podemos ler uma string caracter a caracter, como faríamos com qualquer outro vetor, mas é mais simples ler a string inteira, utilizando o comando
- Note que não utilizamos o e comercial (&) para strings. Isso ocorre pois o nome de um vetor já é um endereço de memória (o endereço de memória do começo do vetor).

```
scanf ("%s", texto);
```

Veja o código: scanf.c

Lendo uma string do teclado

- Infelizmente, a leitura a partir do teclado utilizando o `scanf` lê somente até o primeiro espaço, ou seja, lê somente uma palavra, o que torna o seu uso desta forma um pouco restrito.
- Para contornar isso, podemos utilizar a função `gets`, que faz a leitura até encontrar o caracter de fim de linha (*enter*).

```
gets(texto);
```

Veja o código: `gets.c`

Lendo uma string do teclado

- Outra opção é explorar as outras possibilidades fornecidas pela função `scanf`. Por exemplo, a opção abaixo

```
scanf ("%^[^\n]");
```

lê uma cadeia de caracteres até encontrar um enter.

Veja o código: `scanf[].c`

Veja mais opções consultando: `man scanf`

Lendo uma string do teclado

- Aqui, deparamos com outro problema: tanto o comando `scanf` quanto o `gets` podem ler mais caracteres que os existentes na string, provocando erros.
- A solução aqui, é utilizar uma função que tenha o mesmo comportamento do `gets`, mas que permita limitar a leitura a um tamanho máximo. As opções são:

```
fgets (texto, 50, stdin);
```

```
scanf ("%50[^\n]");
```

Escrevendo uma string na tela

- Podemos escrever uma string na tela caracter a caracter, mas é mais simples escrever utilizando o comando `printf`, com o mesmo comando utilizado para lê-la (`%s`)

```
printf ("%s", texto);
```

Escrevendo uma string na tela

- De forma análoga ao gets e fgets, temos o puts e fputs, que escrevem a string na tela.

```
puts (texto);  
fputs (texto, stdout);
```

Manipulando cadeias de caracteres

As cadeias de caracteres são tão importantes que existe uma biblioteca de funções só com comandos para ela, a biblioteca `string.h`. Entre as diversas funcionalidades oferecidas por esta biblioteca, podemos destacar:

- `strlen(texto)` — Retorna o tamanho da string `texto` em número de caracteres

Manipulando cadeias de caracteres

- `strcpy(destino, fonte)` — Copia a string fonte para a string destino
- `strcat(destino, fonte)` — Concatena a string fonte no fim da string destino

Manipulando cadeias de caracteres

- `strcmp(str1, str2)` — Compara duas cadeias e caracteres e retorna um valor
 - = 0 - se `str1` e `str2` forem iguais
 - < 0 - se `str1` for menor que `str2`
 - > 0 - se `str1` for maior que `str2`

Veja um exemplo de uso em `ordena.c`