

Programação de Computadores I

Strings

String

- Em C, não existe um tipo de dado nativo chamado string. Em vez disso, uma string é simplesmente um vetor de caracteres (char) terminado por um caractere especial: o \0 (barra zero - caractere nulo).

Declaração e Inicialização

- Como as strings são vetores, você deve definir o tamanho máximo que o texto terá. Sempre reserve um espaço a mais para o caractere oculto \0, que indica ao compilador onde o texto termina.

```
char nome[10] = "Lucas"; // Armazena 'L', 'u', 'c', 'a', 's', '\0'  
char sobrenome[] = "Silva"; // O compilador define tamanho 6
```

Lendo uma string

- O Perigo do scanf tradicional
 - Se você usar `scanf("%s", variavel)`, o programa vai parar de ler o texto assim que encontrar o primeiro espaço em branco. Se o usuário digitar "Ana Maria", apenas "Ana" será armazenado na variável.
 - Para ler frases completas (com espaços), a forma correta e segura moderna é usar a função `fgets`.

Exemplo

```
#include <stdio.h>
int main() {
    char nome_completo[50];

    printf("Digite seu nome completo: ");

    // fgets(variavel, tamanho_maximo, stdin);
    // stdin significa que a leitura vem do teclado (standard input)
    fgets(nome_completo, 50, stdin);

    // Exibindo com %s (especificador de string)
    printf("Ola, %s de usar o fgets!\n", nome_completo);

    return 0;
}
```

Principais Funções de Manipulação (`<string.h>`)

- Para realizar operações comuns com textos (como copiar, comparar ou medir o tamanho), você deve incluir a biblioteca `#include <string.h>` no topo do código. As funções mais usadas são:
 - `strlen(str)`: Retorna o tamanho (comprimento) do texto.
 - `strcpy(destino, origem)`: Copia o texto da string origem para a string destino. (Atenção: em C você não pode fazer `texto1 = texto2`; diretamente).
 - `strcmp(str1, str2)`: Compara duas strings. Retorna 0 se elas forem exatamente iguais.

Ajuste de Leitura de string

- Ao ler uma string usando fgets, a string terá como último carácter lido o '\n' (quebra de linha), para retirar esse carácter usamos o comando:

```
nome_completo[strcspn(nome_completo, "\n")] = '\0';
```

Ajuste de Leitura de string

- A função `strcspn(nome_completo, "\n")` varre a string `nome_completo` procurando pela primeira ocorrência do caractere `'\n'` e retorna a posição (o índice) onde ele foi encontrado. Ao fazer `nome_completo[...] = '\0'`, você coloca o terminador de string exatamente em cima do `'\n'`, eliminando a quebra de linha de forma segura. Se o usuário digitar um texto longo que preencha todo o vetor (e o `'\n'` não for salvo), a função retornará o final da string e nada de errado acontecerá.

Exercício

- Validador de Senha
 - Você precisa criar um programa em C que valide a senha de um usuário. O programa deve conter uma senha mestre (secreta) gravada no código.
 - Sua missão é:
 1. Declarar uma string chamada senha_mestre e inicializá-la com o valor "C_e_Melhor".
 2. Criar uma segunda string para receber a tentativa de senha do usuário.
 3. Usar o fgets para ler a entrada do usuário e remover o \n usando a técnica do strcspn que acabamos de ver.
 4. Usar a função strcmp (da biblioteca <string.h>) para comparar as duas strings.
 5. Se as senhas forem iguais, exibir "Acesso Permitido". Caso contrário, exibir "Acesso Negado".