Engenharia de Computação Fundamentos de Programação

Aula 13 – Strings

Prof. Muriel de Souza Godoi muriel@utfpr.edu.br







Strings

- Um dos usos mais comuns de vetores é como strings de caracteres.
- Em C, uma string é definida como um vetor de caracteres que é terminado por um caractere nulo ('\0');
- Na declaração do vetor de caracteres, sempre precisamos adicionar mais uma unidade no tamanho do vetor para o caractere nulo.
- Exemplo de declaração e inicialização:

```
//vetor de 11 caracteres, mas apenas 10 úteis
char str[11] = "teste";
0 1 2 3 4 5 6 7 8 9 10

t e s t e \0
```

Lendo uma string do teclado

- gets(nome);
 - **Problema**: Não segura, pode passar o tamanho máximo;
 - Solução: Não utilizar. Função descontinuada.
- scanf("%s", nome);
 - Problema: Não lê espaços e não garante tamanho;
 - Solução: scanf("%20[^\n]", nome);
 - Recebe no máximo 20 caracteres, exceto '\n'. Não limpa buffer.
- fgets(nome, 20, stdin);
 - Problema: Garante o tamanho correto, mas salva também o '\n' final;
 - Solução: nome[strcspn(nome, "\n")]='\0'e setbuf;
 - Localiza a posição do '\n' e escreve '\0' no local. Não limpa buffer.

Exemplo de String

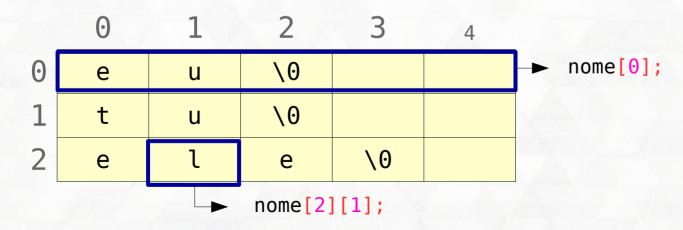
Recebe uma string de até 20 caracteres

```
char nome[21];
printf("Digite seu nome: ");
fgets(nome, 21, stdin);
nome[strcspn(nome, "\n")]='\0';
setbuf(stdin, NULL);
printf("Nome informado: %s \n", nome);
printf("Primeira letra do nome: %c \n", nome[0]);
```

Vetor Bidimensional para Strings

- Um vetor bidimensional (matriz) de caracteres, pode ser interpretado como:
 - Cada linha da matriz é um índice que representa uma String;
 - O número de coluna representa a quantidade de caracteres que cada String possui.
- É possível acessar somente uma linha ou somente um caractere referenciando os índices, conforme abaixo:

```
char nome[3][5];
```



Vetor de Strings - Exemplo

```
int l;
int nomes[3][5];
                                                   \0
                                              u
//Preenchendo os nomes
                                         +
                                                   \0
                                              u
for (l=0; l < 3; l++){
  printf("Nome[%i]:",l);
                                                         \0
                                         e
                                                    е
  fgets(nomes[l], 5, stdin);
  nomes[l][strcspn(nomes[l],"\n")]='\0';
  setbuf(stdin, NULL);
}//for
//Exibindo os nomes
for (l=0; l < 3; l++){}
  printf("Nome[%i]: %s\n",l,nomes[l]);
}//for
```

Biblioteca Strings

 Permite obter informações e manipular strings mais facilmente

Biblioteca Adicional#include <string.h>

Verifique relação completa de funções em:

http://www.cplusplus.com/reference/cstring/

\$ man string

```
strlen(<string>);
```

Retorna a quantidade de caracteres utilizados, ou seja, antes do \0

```
char nome[21];
int tamanho;
printf("Digite seu nome: ");
fgets(nome, 21, stdin);
nome[strcspn(nome, "\n")]='\0';
setbuf(stdin, NULL);
tamanho = strlen(nome);
printf("Nome informado: %s \n", nome);
printf("Tamanho no nome: %d \n", tamanho);
```



Caracteres especiais podem ocupar mais de um byte, podendo gerar erro no valor retornado

```
strcpy(<destino>,<origem>);
```

Copia o conteúdo da string origem na string destino

```
char nome[21];
char novonome[21];
printf("Digite seu nome: ");
fgets(nome, 21, stdin);
nome[strcspn(nome, "\n")]='\0';
setbuf(stdin, NULL);
strcpy(novonome, nome);
printf("Nome informado: %s \n", novonome);
```

```
strcat(<destino>,<origem>);
```

Concatena o conteúdo da string **origem** na string **destino**, ou seja, junta o conteúdo da string origem ao final da string destino

```
char nome[21] = "João";
char sobrenome[21] = " da Silva";

strcat(nome, sobrenome);
printf("Nome concatenado: %s \n", nome);
```

strcmp(<string1>,<string2>);

- Compara se o valor da string1 é igual o valor da string2.
 - Retorna 0 caso os dois valores sejam iguais;
 - Retorna um valor positivo se string1 < string2;
 - Retorna um valor negativo se string1 > string2.

```
char nome1[21] = "Joao";
char nome2[21] = "Joao";

if(strcmp(nome1, nome2) == 0){
    printf("0s nomes são iguais");
} else {
    printf("0s nomes NÃO são iguais");
}// else
```



A função **strcasecmp** compara as *strings* desconsiderando maiúsculas e minúsculas

Exercícios - Strings

- 1) Crie um programa para armazenar 10 nomes em um vetor e imprimir uma lista numerada
- 2) Faça um programa que armazene o nome e salário de 5 empregados. Em seguida calcule um aumento de 8% nos salários e exiba a nova folha de pagamentos.
- 3) Desenvolva um programa que armazene o nome e o telefone de 5 pessoas. Ao digitar a posição desejada, o programa deve exibir o nome e telefone daquela posição. O programa finaliza ao receber a entrada -1.

Exercícios – Funções de Strings

- 4) Faça um programa que receba um nome, ele deve perguntar novamente caso o nome tenha tamanho inferior a 5 caracteres. Ao receber um nome com 5 caracteres ou mais o programa exibe o nome e finaliza.
- 5) Faça um programa que receba uma frase e depois exiba quantas vezes cada vogal aparece. Ele deve contar considerando o tamanho da String.
- 6) Faça um programa que receba um nome e um sobrenome. Ele deve construir uma <u>nova string</u> no formato americano (Sobrenome, Nome) Exemplo: Silva, José
- 7) Escreva um programa que receba uma sigla de um estado da região Sul ou Sudeste e exiba o nome completo do estado correspondente.