



Linguagem de Programação

Biblioteca `<string.h>`

Gil Eduardo de Andrade





Biblioteca <string.h>

Introdução

- A biblioteca <string.h> disponibiliza funções para manipulação de *Strings* (conjunto de caracteres)em C;
- Dentre as funções disponibilizadas temos a possibilidade de obter o comprimento, comparar, copiar e concatenar (juntar) *Strings*;





Biblioteca <string.h>

Principais Funções

- ***int*** *strlen(char* a)* → Calcula o comprimento (quantidade de caracteres) de uma determinada *string*. Onde:
 - ***int***: valor inteiro de **retorno** que indica o total de caracteres calculado para *string a*.
 - ***char****: parâmetro (*string*) passado a função do qual desejar descobrir o comprimento;





Biblioteca <string.h>

Exemplo de Codificação: *strlen()*

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

int main() {

    int tam;
    // declara e aloca um ponteiro do tipo char
    char *p = malloc(30*sizeof(char));

    // solicita que o usuário digite uma palavra e armazena no ponteiro 'p'
    printf("Digite uma palavra: ");
    gets(p);

    // utiliza a função strlen para obter o tamanho desta palavra
    // armazenado o resultado na variável 'tam'
    tam = strlen(p);

    // Exibe o valor calculado pela função 'strlen()'
    printf("\nTamanho da palavra: %i", tam);

    printf("\n");
    return 0;
}
```



Biblioteca <string.h>

Exemplo de Codificação: *strlen()*

```
g113du4rd0@asus-ultrabook-g1l:/run/media/g113du4rd0/GEA/VirtualBox/Compartilhada/IFPR/20... x
Arquivo Editar Ver Pesquisar Terminal Ajuda
[g113du4rd0@asus-ultrabook-g1l pc_exdoc12]$ ./a.out
Digite uma palavra: gil eduardo

Tamanho da palavra: 11
[g113du4rd0@asus-ultrabook-g1l pc_exdoc12]$
```





Biblioteca <string.h>

Principais Funções

- **int** *strcmp(char* a, char* b)* → efetua a comparação de duas *strings*. Onde:
 - **int**: valor inteiro de **retorno** que indica qual foi o resultado da comparação entre *a* e *b*. Para **retorno**:
 - **= 0** → as *strings a* e *b* são iguais;
 - **> 0** → a *string a* é lexicograficamente maior que a *string b*;
 - **< 0** → a *string a* é lexicograficamente menor que a *string b*;
 - **char***: parâmetros (*string*) passados a função para serem comparados;





Biblioteca <string.h>

Exemplo de Codificação: *strcmp()*

```
int comp;
// declara e aloca dois ponteiros do tipo char
char *p1 = malloc(30*sizeof(char));
char *p2 = malloc(30*sizeof(char));

// solicita que o usuário digite duas palavras e as armazena nos
// ponteiros 'p1' e 'p2'
printf("Primeira palavra: ");
gets(p1);
printf("Segunda palavra: ");
gets(p2);

// utiliza a função strcmp para comparar as duas palavras
comp = strcmp(p1, p2);

if(comp == 0) {
    printf("São Iguais!");
}
else if(comp > 0) {
    printf("A primeira é maior que a segunda!");
}
else {
    printf("A primeira é menor que a segunda!");
}
```

Biblioteca <string.h>

Exemplo de Codificação: *strcmp()*

```
g1l3du4rd0@asus-ultrabook-g1l:/run/media/g1l3du4rd0/GEA/VirtualBox/Compartilhada/IF... x
Arquivo Editar Ver Pesquisar Terminal Ajuda
[g1l3du4rd0@asus-ultrabook-g1l pc_exdoc12]$ ./a.out
Primeira palavra: gil
Segunda palavra: gil
São Iguais!
[g1l3du4rd0@asus-ultrabook-g1l pc_exdoc12]$
```

```
g1l3du4rd0@asus-ultrabook-g1l:/run/media/g1l3du4rd0/GEA/VirtualBox/Compartilhada/IF... x
Arquivo Editar Ver Pesquisar Terminal Ajuda
[g1l3du4rd0@asus-ultrabook-g1l pc_exdoc12]$ ./a.out
Primeira palavra: eduardo
Segunda palavra: andrade
A primeira é maior que a segunda!
[g1l3du4rd0@asus-ultrabook-g1l pc_exdoc12]$
```




Biblioteca <string.h>

Principais Funções

- **int** *strncmp*(**char*** *a*, **char*** *b*, **int** *n*) → efetua a comparação de duas *strings* até um total de *n* caracteres. Onde:
 - **int**: valor inteiro de **retorno** que indica qual foi o resultado da comparação entre *a* e *b* até *n* caracteres. Para **retorno**:
 - = 0 → as *sub-strings* de *a* e *b* são iguais;
 - > 0 → a *sub-string* de *a* é lexicograficamente maior que a *string* de *b*;
 - < 0 → a *sub-string* de *a* é lexicograficamente menor que a *sub-string* de *b*;
 - **char***: parâmetros (*string*) passados a função para serem comparados;





Biblioteca <string.h>

Exemplo de Codificação: *strncmp()*

```
int comp, n;
// declara e aloca dois ponteiros do tipo char
char *p1 = malloc(30*sizeof(char));
char *p2 = malloc(30*sizeof(char));

// solicita que o usuário digite duas palavras e as armazena nos
// ponteiros 'p1' e 'p2'
printf("Primeira palavra: ");
gets(p1);
printf("Segunda palavra: ");
gets(p2);

// solicita ao o total de caracteres que deseja-se comparar
printf("Total de caracteres a serem comparados: ");
scanf("%i", &n);

// utiliza a função strncmp para comparar as sub-strings
comp = strncmp(p1, p2, n);

if(comp == 0) {
    printf("São Iguais!");
}
else if(comp > 0) {
    printf("A primeira é maior que a segunda!");
}
else {
    printf("A primeira é menor que a segunda!");
}
```

Biblioteca <string.h>

Exemplo de Codificação: *strncmp()*

```
g1l3du4rd0@asus-ultrabook-g1l:/run/media/g1l3du4rd0/GEA/VirtualBox/Compartilhada/IF... x
Arquivo Editar Ver Pesquisar Terminal Ajuda
[g1l3du4rd0@asus-ultrabook-g1l pc_exdoc12]$ ./a.out
Primeira palavra: camaleão
Segunda palavra: camarão
Total de caracteres a serem comparados: 4
São Iguais!
[g1l3du4rd0@asus-ultrabook-g1l pc_exdoc12]$
```

```
g1l3du4rd0@asus-ultrabook-g1l:/run/media/g1l3du4rd0/GEA/VirtualBox/Compartilhada/IF... x
Arquivo Editar Ver Pesquisar Terminal Ajuda
[g1l3du4rd0@asus-ultrabook-g1l pc_exdoc12]$ ./a.out
Primeira palavra: camaleão
Segunda palavra: camarão
Total de caracteres a serem comparados: 5
A primeira é menor que a segunda!
[g1l3du4rd0@asus-ultrabook-g1l pc_exdoc12]$
```



Biblioteca <string.h>

Principais Funções

- *strcpy(char* destino, char* origem)* → copia uma *string* de origem para outra *string* de destino. Onde:
 - *char* destino*: é a *string* que receberá o conjunto de caracteres a ser copiado de outra *string*;
 - *char* origem*: é a *string* que contém o caracteres que deseja-se copiar para outra *string*;





Biblioteca <string.h>

Exemplo de Codificação: *strcpy()*

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

int main() {

    // declara e aloca dois ponteiros do tipo char
    char *origem = malloc(30*sizeof(char));
    char *destino = malloc(30*sizeof(char));

    // solicita que o usuário digite uma palavra e armazena em 'origem'
    printf("Primeira uma palavra: ");
    gets(origem);

    // utiliza a função strcpy para copiar 'origem' para 'destino'
    strcpy(destino, origem);

    // apresenta o conteúdo copiado para 'destino'
    printf("Conteúdo copiado: %s", destino);

    printf("\n");
    return 0;
}
```



Biblioteca <string.h>

Exemplo de Codificação: *strcpy()*

```
g1l3du4rd0@asus-ultrabook-g1l:/run/media/g1l3du4rd0/GEA/VirtualBox/Compartilhada/IF... x
Arquivo Editar Ver Pesquisar Terminal Ajuda
[g1l3du4rd0@asus-ultrabook-g1l pc_exdoc12]$ ./a.out
Primeira uma palavra: gil eduardo
Conteúdo copiado: gil eduardo
[g1l3du4rd0@asus-ultrabook-g1l pc_exdoc12]$ █
```





Biblioteca <string.h>

Principais Funções

- *strncpy(char* destino, char* origem, int n)* → copia *n* caracteres de uma *string* de origem para outra *string* de destino. Onde:
 - *char* destino*: é a *string* que receberá o conjunto de caracteres a ser copiado de outra *string*;
 - *char* origem*: é a *string* que contém os caracteres que deseja-se copiar para outra *string*;
 - *int n*: é total de caracteres que devem ser copiados da *string* de origem para *string* de destino





Biblioteca <string.h>

Exemplo de Codificação: *strncpy()*

```
int n;
// declara e aloca dois ponteiros do tipo char
char *origem = malloc(30*sizeof(char));
char *destino = malloc(30*sizeof(char));

// solicita que o usuário digite uma palavra e armazena em 'origem'
printf("Primeira uma palavra: ");
gets(origem);
// solicita ao usuário quantos caracteres deseja copiar
printf("Total de caracteres a serem copiados: ");
scanf("%i", &n);

// utiliza a função strncpy para copiar n caracteres de 'origem'
// para 'destino'
strncpy(destino, origem, n);

// apresenta o conteúdo copiado para 'destino'
printf("Conteúdo copiado: %s", destino);

printf("\n");
return 0;
```


Biblioteca <string.h>

Exemplo de Codificação: *strncpy()*

```
g1l3du4rd0@asus-ultrabook-g1l:/run/media/g1l3du4rd0/GEA/VirtualBox/Compartilhada/IF... x
Arquivo Editar Ver Pesquisar Terminal Ajuda
[g1l3du4rd0@asus-ultrabook-g1l pc_exdoc12]$ ./a.out
Primeira uma palavra: gil eduardo
Total de caracteres a serem copiados: 6
Conteúdo copiado: gil ed
[g1l3du4rd0@asus-ultrabook-g1l pc_exdoc12]$
```

```
g1l3du4rd0@asus-ultrabook-g1l:/run/media/g1l3du4rd0/GEA/VirtualBox/Compartilhada/IF... x
Arquivo Editar Ver Pesquisar Terminal Ajuda
[g1l3du4rd0@asus-ultrabook-g1l pc_exdoc12]$ ./a.out
Primeira uma palavra: programação C
Total de caracteres a serem copiados: 8
Conteúdo copiado: programa
[g1l3du4rd0@asus-ultrabook-g1l pc_exdoc12]$
```



Biblioteca <string.h>

Principais Funções

- *strcat(char* destino, char* origem)* → concatena (junta) uma *string* de origem com outra *string* de destino. Onde:
 - *char* destino*: é a *string* que agregará ao seu conteúdo o conjunto de caracteres copiado de outra *string*;
 - *char* origem*: é a *string* que contém os caracteres que deseja-se concatenar com outra *string*;





Biblioteca <string.h>

Exemplo de Codificação: *strcat()*

```
// declara e aloca dois ponteiros do tipo char
char *origem = malloc(30*sizeof(char));
char *destino = malloc(60*sizeof(char));

// solicita que o usuário digite duas palavras e armazena em 'destino'
// e 'origem'
printf("Primeira palavra: ");
gets(destino);
printf("Segunda palavra: ");
gets(origem);

// utiliza a função strcat para concatenar 'destino' e 'origem'
strcat(destino, origem);

// apresenta o conteúdo copiado para 'destino'
printf("Conteúdo Concatenado: %s", destino);

printf("\n");
return 0;
```





Biblioteca <string.h>

Exemplo de Codificação: *strcat()*

```
g1l3du4rd0@asus-ultrabook-g1l:/run/media/g1l3du4rd0/GEA/VirtualBox/Compartilhada/IF... x
Arquivo Editar Ver Pesquisar Terminal Ajuda
[g1l3du4rd0@asus-ultrabook-g1l pc_exdoc12]$ ./a.out
Primeira palavra: paralele
Segunda palavra: pipedo
Conteúdo Concatenado: paralelepipedo
[g1l3du4rd0@asus-ultrabook-g1l pc_exdoc12]$
```





Biblioteca <string.h>

Principais Funções

- *strncat*(**char*** destino, **char*** origem, **int** n) → concatena (junta) **n** caracteres de uma *string* de origem com outra *string* de destino. Onde:
 - **char*** destino: é a *string* que agregará ao seu conteúdo o conjunto de caracteres copiado de outra *string*;
 - **char*** origem: é a *string* que contém os caracteres que deseja-se concatenar com outra *string*;
 - **int** n: é o total de caracteres que serão concatenados da *string* de origem na *string* de destino;





Biblioteca <string.h>

Exemplo de Codificação: *strncat()*

```
int n;
// declara e aloca dois ponteiros do tipo char
char *origem = malloc(30*sizeof(char));
char *destino = malloc(60*sizeof(char));

// solicita que o usuário digite duas palavras e armazena em 'destino'
// e 'origem'
printf("Primeira palavra: ");
gets(destino);
printf("Segunda palavra: ");
gets(origem);

// solicita ao usuário quantos caracteres deseja utilizar da string
// origem para concatenar com a string destino
printf("Total de caracteres a serem copiados: ");
scanf("%i", &n);

// utiliza a função strncat para concatenar 'destino' e 'origem'
strncat(destino, origem, n);

// apresenta o conteúdo copiado para 'destino'
printf("Conteúdo Concatenado: %s", destino);
```



Biblioteca <string.h>

Exemplo de Codificação: *strncat()*

```
g13du4rd0@asus-ultrabook-g1l:/run/media/g13du4rd0/GEA/VirtualBox/Compartilhada/IF... x
Arquivo Editar Ver Pesquisar Terminal Ajuda
[g13du4rd0@asus-ultrabook-g1l pc_exdoc12]$ ./a.out
Primeira palavra: paralele
Segunda palavra: pipedo
Total de caracteres a serem copiados: 4
Conteúdo Concatenado: paralelepipe
[g13du4rd0@asus-ultrabook-g1l pc_exdoc12]$
```



Operadores de Condição

Exemplos Utilizados no Documento

http://www.gileduardo.com.br/ifpr/lp/downloads/pc_exdoc12.zip

Mais Exemplos sobre o Conteúdo

http://www.gileduardo.com.br/ifpr/lp/downloads/pc_ex12.zip

Exercícios sobre o Conteúdo

http://www.gileduardo.com.br/ifpr/lp/downloads/lp_pratica12.pdf

