



CEFET – RJ / *Campus* Maria da Graça
Centro Federal de Educação Tecnológica
Celso Suckow da Fonseca – Rio de Janeiro



Prof. Cristiano Fuschilo

cristiano.fuschilo@cefet-rj.br

Linguagem e Técnicas de Programação



Bacharelado em
Sistemas de
Informação



COORDENAÇÃO DE
**Automação
Industrial**
Ensino de Qualidade



7ª Aula

Strings



String



- No C uma string é um vetor de caracteres terminado com um caractere nulo. O caractere nulo é um caractere com valor inteiro igual a zero (código ASCII igual a 0). O terminador nulo também pode ser escrito usando a convenção de barra invertida do C como sendo `'\0'`.
- Para declarar uma string, podemos usar o seguinte formato geral:

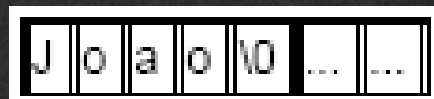




Formato Geral

```
char nome_da_string[tamanho];
```

- Isto declara um vetor de caracteres (uma string) com número de posições igual a tamanho. Note que, como temos que reservar um caractere para ser o terminador nulo, temos que declarar o comprimento da string como sendo, no mínimo, um caractere maior que a maior string que pretendemos armazenar. Vamos supor que declaremos uma string de 7 posições e coloquemos a palavra João nela. Teremos:



- O tamanho máximo da string que você pode entrar é uma string de 99 caracteres.



Características



- Note a forma como inicializamos a string str com os caracteres 'J' 'o' 'a' 'o' e '\0' simplesmente declarando `char str[10] = "Joao"` (uma cadeia de caracteres entre aspas)
- Diferenças caracter x string
`char letra = 'a';`
`char texto[10] = "Joao";`



Importante



- Devemos lembrar que o tamanho da string deve incluir o '\0' final. A biblioteca padrão do C possui diversas funções que manipulam strings. Estas funções são úteis pois não se pode, por exemplo, igualar duas strings:

```
String1 = string2; /* NAO faca isto */
```

- Fazer isto é um desastre.
- As strings devem ser igualadas elemento a elemento.



Copiando uma String

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

int main(int argc, char *argv[])
{
    int num, nr, flag, i;

    if (argc != 2) return 1;
    num = atoi(argv[1]);
    nr = (int)sqrt(num);
    if (num < 2)
        flag = 0;
    else
```



- o programa abaixo que serve para igualar duas strings (isto é, copia os caracteres de uma string para o vetor da outra) :

```
1  #include <stdio.h>
2
3  int main(){
4      char nome[31];
5      int tamanho;
6      printf("Digite um nome de ate 30 caracteres: ");
7      scanf("%s",&nome);
8      for(tamanho=0; tamanho < 31 && nome[tamanho] !='\0'; tamanho++);
9      printf("o tamanho da string %s eh: %d", nome, tamanho);
10
11
12 }
```



Funções Básicas para Manipulação de Strings

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

int main(int argc, char *argv[])
{
    int num, ar, flag, i;

    if (argc != 2) return 1;
    num = atoi(argv[1]);
    ar = (int)sqrt(num);
    if (num < 2)
        flag = 0;
    else
```



- As funções apresentadas nestas seções estão no arquivo cabeçalho string.h.
 - gets
 - A função gets() lê uma string do teclado. Sua forma geral é:
gets (nome_da_string);
 - Exemplo:

```
1  #include <stdio.h>
2  #include <string.h>
3
4  int main (){
5      char string[100];
6      printf ("Digite o seu nome: ");
7      gets (string); // scanf("%s", &string);
8      printf ("\n\n Ola %s", string);
9      return(0);
10 }
```



Funções Básicas para Manipulação de Strings

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

int main(int argc, char *argv[])
{
    int num, ar, flag, i;

    if (argc != 2) return 1;
    num = atoi(argv[1]);
    ar = (int)exp(num);
    if (num < 2)
        flag = 0;
    else
```



– strcpy

- A função strcpy() copia a string-origem para a string- destino. Seu funcionamento é semelhante ao da rotina apresentada na seção anterior. Sua forma geral é:

strcpy (string_destino,string_origem);

- Exemplo:

```
1  #include <stdio.h>
2  #include <string.h>
3
4  int main () {
5      char str1[100],str2[100],str3[100];
6      printf ("Entre com uma string: ");
7      gets (str1);
8      strcpy (str2,str1); /* Copia str1 em str2 */
9      strcpy (str3,"Voce digitou a string "); /* Copia "Voce digitou a string" em str3 */
10     printf ("\n\n%s%s",str3,str2);
11     return(0);
12 }
```



Funções Básicas para Manipulação de Strings

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

int main(int argc, char *argv[])
{
    int num, nr, flag, i;

    if (argc != 2) return 1;
    num = atoi(argv[1]);
    nr = (int)sqrt(num);
    if (num < 2)
        flag = 0;
    else
```



– strcat

- A string de origem permanecerá inalterada e será anexada ao fim da string de destino. A função strcat() tem a seguinte forma geral:
strcat (string_destino,string_origem);
- Exemplo:

```
1  #include <stdio.h>
2  #include <string.h>
3
4  int main () {
5      char str1[100],str2[100];
6      printf ("Entre com uma string: ");
7      gets (str1);
8      strcpy (str2,"Voce digitou a string ");
9      strcat (str2,str1); /* str2 armazenara' Voce digitou a string + o conteudo de str1 */
10     printf ("\n\n%s",str2);
11     return(0);
12 }
```



Funções Básicas para Manipulação de Strings

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

int main(int argc, char *argv[])
{
    int num, ar, flag, i;

    if (argc != 2) return 1;
    num = atoi(argv[1]);
    ar = (int)sqrt(num);
    if (num < 2)
        flag = 0;
    else
```



– strlen

- A função strlen() retorna o comprimento da string fornecida.
- Sua forma geral é:
strlen (string);
- Exemplo:

```
1  #include <stdio.h>
2  #include <string.h>
3
4  int main () {
5      int size;
6      char str[100];
7      printf ("Entre com uma string: ");
8      gets (str);
9      size=strlen (str);
10     printf ("\n\nA string que voce digitou tem tamanho %d",size);
11     return(0);
12 }
```



Funções Básicas para Manipulação de Strings

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

int main(int argc, char *argv[])
{
    int num, nr, flag, i;

    if (argc != 2) return 1;
    num = atoi(argv[1]);
    nr = (int)sqrt(num);
    if (num < 2)
        flag = 0;
    else
```



– strcmp

- A função strcmp() compara a string 1 com a string 2. Se as duas forem idênticas a função retorna zero. Se elas forem diferentes a função retorna não zero. Sua forma geral é:

strcmp (string1,string2);

- Um exemplo da sua utilização:

```
1  #include <stdio.h>
2  #include <string.h>
3
4  int main () {
5      char str1[100],str2[100];
6      printf ("Entre com uma string: ");
7      gets (str1);
8      printf ("\n\nEntre com outra string: ");
9      gets (str2);
10     if (strcmp(str1,str2))
11         printf ("\n\nAs duas strings são diferentes.");
12     else printf ("\n\nAs duas strings são iguais.");
13     return(0);
14 }
```



Matrizes de Strings



- Matrizes de strings são matrizes bidimensionais. Imagine uma string. Ela é um vetor. Se fizermos um vetor de strings estaremos fazendo uma lista de vetores. Esta estrutura é uma matriz bidimensional de chars. Podemos ver a forma geral de uma matriz de strings como sendo:

```
char nome_da_variável [num_de_strings] [compr_das_strings];
```

- Aí surge a pergunta: como acessar uma string individual? Fácil. É só usar apenas o primeiro índice. Então, para acessar uma determinada string faça:

```
nome_da_variável [índice]
```



Tabela ASCII no C



- O código ASCII (American Standard Code for Information Interchange) representa uma maneira de codificar caracteres na forma de valores inteiros.
- Neste código, os caracteres são mapeados para valores numéricos representáveis por sete dígitos binários (bits). Este código abrange 95 caracteres passíveis de impressão e 33 caracteres especiais utilizados, entre outros, no controle de comunicação entre computadores ou um computador e seus periféricos.



Tabela do padrão ASCII.



	00	16	32	48	64	80	96	112
0	NUL DLE			0	@	P	`	p
1	SOH DC1	!	1	A	Q	a	q	
2	STX DC2	"	2	B	R	b	r	
3	ETX DC3	#	3	C	S	c	s	
4	EOT DC4	\$	4	D	T	d	t	
5	ENQ NAK	%	5	E	U	e	u	
6	ACK SYN	&	6	F	V	f	v	
7	BEL ETB	'	7	G	W	g	w	
8	BS CAN	(8	H	X	h	x	
9	HT EM)	9	I	Y	i	y	
10	LF SUB	*	:	J	Z	j	z	
11	VT ESC	+	;	K	[k	{	
12	FF FS	,	<	L	\	l		
13	CR GS	-	=	M]	m	}	
14	SO RS	.	>	N	^	n	~	
15	SI US	/	?	O	_	o	DEL	

Para obter o código interno, em decimal, você soma o valor no início da linha onde se encontra o caractere desejado com o valor no topo de sua coluna.

Os 33 caracteres de controle são os de código 0 a 31 (**NUL** a **US**) e o de código 127 (**DEL**).

O código da letra **A**, por exemplo, é $1 + 64 = 65$.



ASCII Estendido



- A codificação "ASCII Estendido" utiliza 8 bits para representar caracteres. Dessa forma consegue codificar 256 caracteres.
- Os primeiros 128 correspondem exatamente aos representados acima. Os 128 caracteres que se seguem incluem letras acentuadas e outros símbolos. Execute o programa abaixo para gerar a "Tabela ASCII Estendida".



Programa ASCII

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

int main(int argc, char *argv[])
{
    int num, nr, flag, i;

    if (argc != 2) return 1;
    num = atoi(argv[1]);
    nr = (int)sqrt(num);
    if (num < 2)
        flag = 0;
    else
```



```
1  #include<stdio.h>
2
3  int main(void){
4      int i, j, k;
5      char c;
6      printf("Tabela ASCII Estendida\n\n");
7      for (i = 0; i < 26; i++){
8          for (j = 0; j < 10; j++){
9              k = i * 10 + j;
10             printf("%3d %c ", k, k);
11         }
12         putchar('\n');
13     }
14     printf("\nDigite:Computa%c%co!\n", 135, 198);
15     do
16         putchar(c = getchar());
17     while (c != '\n');
18     return(0);
19 }
```



"Mas que Diabos é Isso?"



Este outdoor está em linguagem "C", e essa sequência de caracteres forma a frase "Now Hiring" (estamos contratando), ou seja, precisam de pessoas que entendam da linguagem.



