

Técnicas de Programação

Cadeias de Caracteres (Strings)

Fábio Duncan de Souza

Instituto Federal Fluminense

Técnicas de
Programação

Fábio Duncan

Caracteres

Cadeia de
Caracteres
(Strings)

Strings na
Linguagem C

Biblioteca de
Manipulação
de Strings

Referências

- 1 Caracteres
- 2 Cadeia de Caracteres (Strings)
- 3 Strings na Linguagem C
- 4 Biblioteca de Manipulação de Strings



Caracteres



Técnicas de Programação

Fábio Duncan

Cadeia de Caracteres (Strings)

Strings na Linguagem C

Biblioteca de Manipulação de Strings

Referências

- Um byte pode ser interpretado como um caractere (letra, dígito, sinal de pontuação, etc.).
- Tem por base a tabela ASCII, uma tabela arbitrária mas universalmente aceita que associa um caractere a cada byte. (ASCII é a sigla do American Standard Code for Information Interchange.)
- A tabela ASCII usa apenas bytes cujo primeiro bit é 0 e portanto, tem apenas 128 linhas.



Tabela ASCII

Técnicas de
Programação

Fábio Duncan

Caracteres

Cadeia de
Caracteres
(Strings)

Strings na
Linguagem C

Biblioteca de
Manipulação
de Strings

Referências

ASCII control characters		
00	NULL	(Null character)
01	SOH	(Start of Header)
02	STX	(Start of Text)
03	ETX	(End of Text)
04	EOT	(End of Trans.)
05	ENQ	(Enquiry)
06	ACK	(Acknowledgement)
07	BEL	(Bell)
08	BS	(Backspace)
09	HT	(Horizontal Tab)
10	LF	(Line feed)
11	VT	(Vertical Tab)
12	FF	(Form feed)
13	CR	(Carriage return)
14	SO	(Shift Out)
15	SI	(Shift In)
16	DLE	(Data link escape)
17	DC1	(Device control 1)
18	DC2	(Device control 2)
19	DC3	(Device control 3)
20	DC4	(Device control 4)
21	NAK	(Negative acknowl.)
22	SYN	(Synchronous idle)
23	ETB	(End of trans. block)
24	CAN	(Cancel)
25	EM	(End of medium)
26	SUB	(Substitute)
27	ESC	(Escape)
28	FS	(File separator)
29	GS	(Group separator)
30	RS	(Record separator)
31	US	(Unit separator)
127	DEL	(Delete)

ASCII printable characters			
32	space	64	@
33	!	65	A
34	"	66	B
35	#	67	C
36	\$	68	D
37	%	69	E
38	&	70	F
39	'	71	G
40	(72	H
41)	73	I
42	*	74	J
43	+	75	K
44	,	76	L
45	-	77	M
46	.	78	N
47	/	79	O
48	0	80	P
49	1	81	Q
50	2	82	R
51	3	83	S
52	4	84	T
53	5	85	U
54	6	86	V
55	7	87	W
56	8	88	X
57	9	89	Y
58	:	90	Z
59	;	91	[
60	<	92	\
61	=	93]
62	>	94	^
63	?	95	_
96	`		
97	a		
98	b		
99	c		
100	d		
101	e		
102	f		
103	g		
104	h		
105	i		
106	j		
107	k		
108	l		
109	m		
110	n		
111	o		
112	p		
113	q		
114	r		
115	s		
116	t		
117	u		
118	v		
119	w		
120	x		
121	y		
122	z		
123	{		
124			
125	}		
126	~		

- É comum usar atalhos verbais ao falar de caracteres ASCII
 - Por exemplo, em vez de dizer o caractere A pode-se dizer o caractere 65, pois o byte que corresponde ao A na tabela ASCII vale 65 em notação binária
- Além dos noventa e cinco caracteres normais, o alfabeto ASCII contém trinta e três caracteres especiais, conhecidos como caracteres de controle.
 - Esses caracteres não são símbolos tipográficos como os outros
 - São representados utilizando uma barra invertida seguida de um dígito ou letra.
- A tabela ASCII não contém letras com acentos e cedilha.
 - Para a representação deste são utilizados padrões de codificação mais amplos que incluem também os caracteres ASCII, como o Latin1, UTF-8 e Unicode.



Tabela ASCII Estendida - (ISO 8859-1 / Latin-1)

Técnicas de
Programação

Fábio Duncan

Caracteres

Cadeia de
Caracteres
(Strings)

Strings na
Linguagem C

Biblioteca de
Manipulação
de Strings

Referências

Extended ASCII characters							
128	Ç	160	á	192	Ł	224	Ó
129	û	161	í	193	ł	225	ô
130	é	162	ó	194	Ŧ	226	Ô
131	ä	163	ú	195	ŧ	227	Ö
132	ä	164	ñ	196	—	228	ö
133	à	165	Ñ	197	†	229	Õ
134	â	166	ª	198	ä	230	μ
135	ç	167	º	199	Ä	231	þ
136	ê	168	¿	200	£	232	ƒ
137	ë	169	©	201	ƒ	233	Ú
138	è	170	¬	202	ƒ	234	Û
139	ÿ	171	½	203	ƒ	235	Ü
140	ı	172	¼	204	ƒ	236	ý
141	ı	173	ı	205	=	237	Ÿ
142	Ä	174	«	206	±	238	ˆ
143	Å	175	»	207	±	239	˜
144	É	176	¶	208	ð	240	≡
145	æ	177	§	209	Ð	241	±
146	Æ	178	■	210	É	242	≡
147	ó	179	ƒ	211	Ê	243	¼
148	ö	180	ƒ	212	Ë	244	¶
149	õ	181	À	213	ı	245	§
150	ü	182	Á	214	ı	246	+
151	ù	183	Â	215	ı	247	°
152	ÿ	184	©	216	ı	248	°
153	Ö	185	ƒ	217	ı	249	°
154	Û	186	ƒ	218	ı	250	°
155	ø	187	ı	219	ı	251	°
156	£	188	ı	220	ı	252	°
157	Ø	189	€	221	ı	253	°
158	x	190	¥	222	ı	254	°
159	f	191	ı	223	ı	255	nbsp



Cadeia de Caracteres (Strings)



Conceitos Básicos

Técnicas de
Programação

Fábio Duncan

Caracteres

Cadeia de
Caracteres
(Strings)

Strings na
Linguagem C

Biblioteca de
Manipulação
de Strings

Referências

- Na programação de computadores, uma cadeia de caracteres ou string é uma sequência de caracteres, geralmente utilizada para representar palavras, frases ou textos de um programa.
- Uma string é uma série de caracteres tratada como uma unidade simples;
- Em uma string podem estar incluídas letras, dígitos e caracteres especiais como +, -, *, \$, dentre outros;
- Exemplos:
 - "Adalberto Fritz" (um nome)
 - "Rua Falcon 222" (um endereço)
 - "Campos dos Goytacazes, Rio de Janeiro" (uma cidade e um estado)
 - "(071)555-1212" (um número de telefone)

- As cadeias de caracteres podem ser expressas tanto na forma literal, como através de algum tipo de variável;
- Strings literais ou constantes strings (constantes alfanuméricas) são escritas comumente entre aspas duplas;
- Existem linguagens de programação onde o tipo string pode ter o seu conteúdo alterado, bem como linguagens onde o tipo string é imutável;
- Uma cadeia de caracteres é vista como sendo um tipo de dado e normalmente é implementada através de um arranjo de bytes que armazena os elementos da cadeia em sequência, utilizando alguma codificação preestabelecida.

```

1  Início
2      caractere nome[10]
3      inteiro tamanhoNome
4
5      escreva ("Entre com o nome:")
6      leia(nome)
7      se nome = "Maria" então
8          escreva("O nome digitado foi Maria.")
9      senão
10         escreva("O nome digitado não foi Maria.");
11      fim se
12      nome ← "Neusa"
13      escreva("Novo conteúdo de nome:",nome)
14  Fim do Algoritmo

```

Strings na Linguagem C

Strings na Linguagem C

Técnicas de
Programação

Fábio Duncan

Caracteres

Cadeia de
Caracteres
(Strings)

Strings na
Linguagem C

Biblioteca de
Manipulação
de Strings

Referências

- Uma string em C é formada por uma estrutura de dados que contém caracteres sequencialmente armazenados na memória, e que termina com o caractere NULL (`'\0'`);
- Uma string é acessada por meio do endereço da primeira posição da estrutura de dados;
- O valor de uma variável string é na verdade o endereço de memória do primeiro caractere existente na estrutura.



Strings na Linguagem C

char nome[10]

nome

7ffc25a01360

Memória RAM

		:
nome[0]	M	7ffc25a01360
nome[1]	a	7ffc25a01361
nome[2]	r	7ffc25a01362
nome[3]	i	7ffc25a01363
nome[4]	a	7ffc25a01364
nome[5]	\0	7ffc25a01365
nome[6]		7ffc25a01366
nome[7]		7ffc25a01367
nome[8]		7ffc25a01368
nome[9]		7ffc25a01369
		7ffc25a0136A
		:

Exemplos:

Inicialização da String

```
char nome[10];  
char nome[10] = "Maria";  
char nome[10] = {'M','a','r','i','a','\0'};
```

Leitura e impressão da string na tela

```
scanf("%s", nome); //Sem o &  
printf("%s", nome);
```

Alteração e impressão do 5º caractere da string

```
nome[4] = 'o';  
printf("%c", nome[4]);
```

Impressão do endereço da string

```
printf("%p", nome);
```

Obs: No momento da definição do tamanho da string deve-se levar em conta a presença do caractere de finalização da estrutura (\0), que sempre ocupará um espaço a mais.

Biblioteca de Manipulação de Strings

Biblioteca de Manipulação de strings - string.h

Técnicas de
Programação

Fábio Duncan

Caracteres

Cadeia de
Caracteres
(Strings)

Strings na
Linguagem C

Biblioteca de
Manipulação
de Strings

Referências

- A biblioteca de manipulação de strings fornece funções úteis para a manipulação deste tipo de dado;
- Para utilizar funções da biblioteca de manipulação de strings, deve-se incluir o arquivo de cabeçalho `<string.h>`;
- Exemplos de funções:
 - Comparação de strings
 - `strcmp(string1, string2)`
Compara a `string1` com a `string2`
A função `strcmp` retorna um inteiro maior que zero se a `string1` for maior do que a `string2`
A função `strcmp` retorna zero se a `string1` for igual a `string2`
A função `strcmp` retorna um inteiro menor do que zero se a `string1` for menor do que a `string2`
Obs: cabe lembrar que os valores se baseiam na tabela ASCII.

- Exemplos de funções:
 - Cópia de strings
 - `strcpy(string1, string2)`
Copia o valor de `string2` para `string1`.
 - Determinação do comprimento de uma string
 - `strlen(string1)`
Retorna um valor inteiro correspondente ao total de caracteres de `string1`.
Obs: Não inclui o caractere nulo.
 - Diferentes funcionalidades úteis para manipulação de strings e de caracteres podem ser encontradas em `string.h` e `ctype.h`.
- Como pôde ser percebido os operadores de atribuição e relacionais não podem ser utilizados com strings, pois estas são estruturas de dados e não tipos primitivos.

```

1 #include<stdio.h>
2 #include<string.h>
3
4 int main() {
5     char nome[10];
6     int tamanhoNome;
7
8     printf("Entre com o nome:");
9     scanf("%s",nome);
10    tamanhoNome = strlen(nome);
11    printf("A string tem %d letras.\n", tamanhoNome);
12    if (strcmp(nome, "Maria") == 0) {
13        printf("O nome digitado foi Maria.\n");
14    }
15    else {
16        printf("O nome digitado não foi Maria.\n");
17    }
18    strcpy(nome, "Neusa");
19    printf("Novo conteúdo de nome: %s.\n",nome);
20    return 0;
21 }

```

- ◀ ◻ ▶ ◀ ◻ ▶ ◀ ≡ ▶ ◀ ≡ ▶ ≡ ↺ 🔍 ↻ 19/25

Detalhes da Entrada de Strings via Teclado

Técnicas de
Programação

Fábio Duncan

Caracteres

Cadeia de
Caracteres
(Strings)

Strings na
Linguagem C

Biblioteca de
Manipulação
de Strings

Referências

- `scanf`
 - Pressupõem que o fluxo de entrada é dividido em uma sequência de tokens
 - Por padrão, os tokens são separados por espaço em branco, tabulação ou nova linha
 - Utiliza a sequência de controle `%s` para a realização da leitura
 - A leitura ocorre até que seja encontrado um caractere delimitador do token

Detalhes da Entrada de Strings via Teclado

Técnicas de
Programação

Fábio Duncan

Caracteres

Cadeia de
Caracteres
(Strings)

Strings na
Linguagem C

Biblioteca de
Manipulação
de Strings

Referências

- scanf
 - Pode ser utilizado com os seguintes modificadores:
 - Campo largura
 - Um número inteiro decimal positivo que controla o número máximo de caracteres a serem lidos, adicionado após o caractere %
 - Ex: `scanf("%20s", nome);` Será lido um nome com no máximo 20 caracteres
 - Definição dos caracteres permitidos na leitura ([])
 - Entre colchetes podem vir todos os caracteres válidos para serem lidos
 - A leitura é encerrada quando um caractere inválido é encontrado
 - Ex: `scanf("%20[aeiou]s", nome);` Somente será permitida a entrada das vogais minúsculas

- scanf
 - Pode ser utilizado com os seguintes modificadores (cont):
 - Definição dos caracteres não permitidos na leitura ([^])
 - Entre colchetes podem vir todos os caracteres inválidos para a leitura
 - A leitura é encerrada quando um caractere inválido é encontrado
 - Ex: scanf("%20[^aeiou]s", nome); Não serão permitidas as entradas das vogais minúsculas

- scanf

- O buffer de teclado permanecerá com os dados que foram digitados e não armazenados na variável
- Para a leitura de uma próxima string ou um char, na maioria da vezes, faz-se necessário o esvaziamento do buffer de teclado
- Formas de se esvaziar o buffer
 - Adicionar um caractere de espaço em branco antes do caractere %
 - Descarta do buffer os caracteres não imprimíveis antes da leitura de uma string ou de um char
 - Ex: `scanf("%s", palavra);`
 - Utilizar o caractere *
 - Faz a leitura dos caracteres e os descarta
 - Ex: `scanf("%*[^\\n]");` Faz a leitura dos caracteres até encontrar o caractere enter e os descarta
 - Ex: `scanf("%*c");` Descarta o último char lido



Detalhes da Entrada de Strings via Teclado

- `fgets`
 - Função para realizar a leitura de uma string disponível em um arquivo
 - Quando utilizado com o arquivo pré-definido `stdin` (entrada padrão), viabiliza a leitura do teclado
 - Formato: `fgets (string, tamanhoMáximoString, stdin);`
 - Tem como parâmetros:
 - A string a ser lida
 - O tamanho máximo da cadeia de caracteres
 - O nome do arquivo (para o nosso caso o arquivo identificado por `stdin`)
 - Armazena o caractere `\n` dentro da própria string lida, necessitando por vezes tratar esta característica
- Obs: Além dos comandos `scanf` e `fgets` existe o comando `gets` (sem o `f`). Porém este se encontra desatualizado, possui falhas e não deve ser utilizado.



Referências Bibliográficas

Técnicas de
Programação

Fábio Duncan

Caracteres

Cadeia de
Caracteres
(Strings)

Strings na
Linguagem C

Biblioteca de
Manipulação
de Strings

Referências



Harvey M Deitel and Paul J Deitel.
Como programar em C.
LTC, 1999.



Renato Cardoso Mesquita.
Curso de linguagem C.
Universidade Federal de Minas Gerais, 1998.



Wikimedia.
Linguagem de programação.
Wikipédia: a enciclopédia livre, 2020.