



INSTITUTO DE
INFORMÁTICA
UFG

Entrada e
Saída na
LinguagemC

Thierson
Couto Rosa

Organização

Introdução

Entrada-
Padrão

Funções da
Entrada-Padrão

Saída-Padrão

Entrada e Saída na LinguagemC

Thierson Couto Rosa

Instituto de Informática
Universidade Federal de Goiás



INSTITUTO DE
INFORMÁTICA
UFG

Entrada e
Saída na
Linguagem C

Thierson
Couto Rosa

Organização

Introdução

Entrada-
Padrão

Funções da
Entrada-Padrão

Saída-Padrão

1 Introdução

2 Entrada-Padrão

- Funções da Entrada-Padrão

3 Saída-Padrão



INSTITUTO DE
INFORMÁTICA
UFG

Entrada e Saída na Linguagem C

Entrada e
Saída na
Linguagem C

Thierson
Couto Rosa

Organização

Introdução

Entrada-
Padrão

Funções da
Entrada-Padrão

Saída-Padrão

- Na linguagem C, as operações de entrada e saída não fazem parte da linguagem (não há comandos de entrada e saída);
- As operações de entrada e saída são feitas por funções.
- A padronização da linguagem (ANSI) determina a existência de uma biblioteca padrão (libc) que todas implementações da linguagem devem ter.
- A libc implementa as funções de entrada e saída da linguagem C, entre outras funções, e armazena os códigos dessas funções em linguagem de máquina.



- É necessário que um programa declare as funções de entrada e saída antes de utilizá-las.
- Para evitar que o programador tenha que fazer isso em cada programa que ele escreve, essas declarações são mantidas em um arquivo de cabeçalho: **stdio.h**.
- O programador coloca a diretiva **#include<stdio.h>** no início do programa.
- O pré-processador, ao ler a diretiva, inclui o arquivo **stdio.h** que contém as declarações de funções de entrada e de saída da linguagem C no programa.
- Se você esquecer de incluir a diretiva acima no início do seu programa, o compilador acusará um erro de compilação do tipo: “identifier not declared”.



Entrada-Padrão e Saída-Padrão

Entrada e
Saída na
Linguagem C

Thierson
Couto Rosa

Organização

Introdução

Entrada-
Padrão

Funções da
Entrada-Padrão

Saída-Padrão

- A biblioteca libc especifica um grande conjunto de funções para as operações de entrada e saída a serem usadas com a linguagem C.
- Em seções mais avançadas do curso estudaremos várias funções de entrada e saída específicas para manipulação de arquivos.
- No momento, nos interessa apenas fazer programas que obtenham entrada do teclado e que gerem saída na tela do computador.
- Para isso estudaremos funções que implementam a entrada-padrão e a saída-padrão.



Entrada-Padrão - stdin

Entrada e
Saída na
Linguagem C

Thierson
Couto Rosa

Organização

Introdução

Entrada-
Padrão

Funções da
Entrada-Padrão

Saída-Padrão

- A libc implementa funções para um dispositivo virtual de entrada denominado *entrada padrão - stdin*
- A esse dispositivo podem ser associados quaisquer dispositivos físicos de entrada (disco, teclado, pen-drive);
- Quando nenhum dispositivo é explicitamente associado ao stdin pelo programador, a biblioteca considera que esse dispositivo é o teclado. Portanto, por padrão, a leitura é feita do teclado.



INSTITUTO DE
INFORMÁTICA
UFG

Entrada e
Saída na
Linguagem C

Thierson
Couto Rosa

Organização

Introdução

Entrada-
Padrão

Funções da
Entrada-Padrão

Saída-Padrão

Entrada-Padrão como um Arquivo Texto

- A entrada-padrão é tratada com um arquivo de tipo texto, isso é, uma sequência finita de caracteres.
- Sendo assim, as funções de entrada lêem sequências de caracteres na entrada e as convertem para os tipos de dados apropriados ao armazená-los na memória.
- Esses caracteres são codificados de acordo com a tabela ASCII. Essa tabela especifica como os caracteres são representados em binário, em um byte.
- Uma variável do tipo **char** armazena em um byte o valor binário correspondente ao caractere da tabela ASCII.



INSTITUTO DE
INFORMÁTICA
UFG

Tabela ASCII

Entrada e
Saída na
Linguagem C

Thierson
Couto Rosa

Organização

Introdução

Entrada-
Padrão

Funções da
Entrada-Padrão

Saída-Padrão

Tabela ASCII

Dec	Hex	Oct	Char	Dec	Hex	Oct	Char	Dec	Hex	Oct	Char	Dec	Hex	Oct	Char
0	0	0		32	20	40	[space]	64	40	100	@	96	60	140	`
1	1	1		33	21	41	!	65	41	101	A	97	61	141	a
2	2	2		34	22	42	"	66	42	102	B	98	62	142	b
3	3	3		35	23	43	#	67	43	103	C	99	63	143	c
4	4	4		36	24	44	\$	68	44	104	D	100	64	144	d
5	5	5		37	25	45	%	69	45	105	E	101	65	145	e
6	6	6		38	26	46	&	70	46	106	F	102	66	146	f
7	7	7		39	27	47	'	71	47	107	G	103	67	147	g
8	8	10		40	28	50	(72	48	110	H	104	68	150	h
9	9	11		41	29	51)	73	49	111	I	105	69	151	i
10	A	12		42	2A	52	*	74	4A	112	J	106	6A	152	j
11	B	13		43	2B	53	+	75	4B	113	K	107	6B	153	k
12	C	14		44	2C	54	,	76	4C	114	L	108	6C	154	l
13	D	15		45	2D	55	-	77	4D	115	M	109	6D	155	m
14	E	16		46	2E	56	.	78	4E	116	N	110	6E	156	n
15	F	17		47	2F	57	/	79	4F	117	O	111	6F	157	o
16	10	20		48	30	60	0	80	50	120	P	112	70	160	p
17	11	21		49	31	61	1	81	51	121	Q	113	71	161	q
18	12	22		50	32	62	2	82	52	122	R	114	72	162	r
19	13	23		51	33	63	3	83	53	123	S	115	73	163	s
20	14	24		52	34	64	4	84	54	124	T	116	74	164	t
21	15	25		53	35	65	5	85	55	125	U	117	75	165	u
22	16	26		54	36	66	6	86	56	126	V	118	76	166	v
23	17	27		55	37	67	7	87	57	127	W	119	77	167	w
24	18	30		56	38	70	8	88	58	130	X	120	78	170	x
25	19	31		57	39	71	9	89	59	131	Y	121	79	171	y
26	1A	32		58	3A	72	:	90	5A	132	Z	122	7A	172	z
27	1B	33		59	3B	73	;	91	5B	133	[123	7B	173	{
28	1C	34		60	3C	74	<	92	5C	134	\	124	7C	174	
29	1D	35		61	3D	75	=	93	5D	135]	125	7D	175	}
30	1E	36		62	3E	76	>	94	5E	136	^	126	7E	176	~
31	1F	37		63	3F	77	?	95	5F	137	_	127	7F	177	



INSTITUTO DE
INFORMÁTICA
UFG

Organização

Entrada e
Saída na
Linguagem C

Thierson
Couto Rosa

Organização

Introdução

Entrada-
Padrão

Funções da
Entrada-Padrão

Saída-Padrão

1 Introdução

2 Entrada-Padrão

- Funções da Entrada-Padrão

3 Saída-Padrão



Leitura de um Caractere

Entrada e
Saída na
Linguagem C

Thierson
Couto Rosa

Organização

Introdução

Entrada-
Padrão

Funções da
Entrada-Padrão

Saída-Padrão

`getchar()`

- A função `getchar()` não tem parâmetros e retorna um valor do tipo `int` contendo a representação em binário de um caractere lido, de acordo com a tabela ASCII.
- A função `getchar()` irá ler um caractere na entrada após o usuário digitar a tecla RETURN.
- Mesmo que o usuário tenha digitado vários caracteres, a função lê apenas o próximo caractere na entrada.



getchar() - Exemplo

Entrada e
Saída na
Linguagem C

Thierson
Couto Rosa

Organização

Introdução

Entrada-
Padrão

Funções da
Entrada-Padrão

Saída-Padrão

Lendo um Caractere

```
1 #include<stdio.h>
2
3 int main(){
4     int caractere;
5     printf("Digite uma tecla e
        tecla enter\n");
6     //leitura de um caractere
7     caractere=getchar();
8     printf("O caractere que voce
        digitou e: %c.\n",
        caractere);
9 }
```

Entrada:

HGJPPP



getchar() - Exemplo

Entrada e
Saída na
Linguagem C

Thierson
Couto Rosa

Organização

Introdução

Entrada-
Padrão

Funções da
Entrada-Padrão

Saída-Padrão

Lendo um Caractere

```
1 #include<stdio.h>
2
3 int main(){
4     int caractere;
5     printf("Digite uma tecla e
        tecle enter\n");
6     //leitura de um caractere
7     caractere=getchar();
8     printf("O caractere que voce
        digitou e: %c.\n",
        caractere);
9 }
```

Entrada:

HGJPPP

Saída:

O caractere que voce digitou e: H.



Leitura Formatada - scanf()

```
int scanf("formato", lista_de_enderecos_de_memoria)
```

- A função **scanf()** lê caracteres da entrada padrão e se a sequência lida estiver de acordo com o formato (primeiro parâmetro), a sequência é convertida em tipos definidos no formato e armazenada nos endereços de memória especificados nos demais parâmetros.
- A função **scanf()** pára a leitura quando o formato especificado é completamente atendido com os caracteres já lidos.



Leitura Formatada - scanf()

Formato

- O primeiro parâmetro da função é uma cadeia de caracteres entre aspas que determina:
 - O formato que a entrada deve atender para que a leitura ocorra;
 - A forma de conversão de partes da cadeia de caracteres para algum tipo da linguagem C.
- A cadeia de caracteres de formato pode conter:
 - Espaços e tabulações (que devem ocorrer na entrada)
 - Caracteres comuns (exceto %) que devem casar com os próximos **caracteres-não-espaço** na entrada.
 - Uma especificação de conversão: % seguido por um caracter *x*.



Leitura Formatada - scanf()

Especificação de conversão

Os caracteres *x* em uma especificação de conversão podem ser:

Caractere	Tipo de parâmetro a ser lido
d	Endereço para uma variável do tipo int
i	Endereço de um int - o número pode estar em octal (um 0 antes) ou hexadecimal (0x antes).
o	Endereço de um int - o número na entrada deve ser escrito em octal.
u	Endereço de um unsigned int .
h	Endereço de um int - o valor na entrada deve ser em hexadecimal.
e,f,g	Endereço de um float .

Um 'L' ou 'l' pode aparecer antes dos caracteres 'e', 'f', ou 'g' para indicar que a variável é um **long double**. Um 'l' ante de 'd', 'i' indica que a variável é um **long int**.



Leitura Formatada - scanf()

Exemplo

```
1 #include<stdio.h>
2 int main(){
3     /* Ler um inteiro, um float e um caractere na entrada. Suponha que
       entre o float e o int ha um ou mais espacos-brancos e que entre
       o float e o caractere ha um ou mais espacos-brancos.*/
4     int i;
5     float f;
6     char c;
7
8     scanf("%d %f %c", &i, &f, &c);
9     printf ("Os valores lidos sao: %d %f %c\n", i, f, c);
10
11     return(0);
12 }
```




Leitura Formatada - scanf()

Exemplo

```
1 #include<stdio.h>
2 int main(){
3     /* Ler um inteiro, um float e um caractere na entrada. Suponha que
       entre o float e o int ha um ou mais espacos-brancos e que entre
       o float e o caractere ha um ou mais espacos-brancos.*/
4     int i;
5     float f;
6     char c;
7
8     scanf("%d %f %c", &i, &f, &c);
9     printf ("Os valores lidos sao: %d %f %c\n", i, f, c);
10
11     return(0);
12 }
```

Exemplo de execução:

Entrada
23(10 espaços)5.4(espacos ou tabulações ou quebras de linha)H



Leitura Formatada - scanf()

Exemplo

```
1 #include<stdio.h>
2 int main(){
3     /* Ler um inteiro, um float e um caractere na entrada. Suponha que
       entre o float e o int ha um ou mais espacos-brancos e que entre
       o float e o caractere ha um ou mais espacos-brancos.*/
4     int i;
5     float f;
6     char c;
7
8     scanf("%d %f %c", &i, &f, &c);
9     printf ("Os valores lidos sao: %d %f %c\n", i, f, c);
10
11     return(0);
12 }
```

Exemplo de execução:

Entrada
23(10 espaços)5.4(espacos ou tabulações ou quebras de linha)H

Saída
Os valores lidos sao: 23 5.4 H



Leitura Formatada - scanf()

Exemplo

```
1 #include<stdio.h>
2 int main(){
3     /* Ler um inteiro, um float e um caractere na entrada. Suponha que
       entre o float e o int ha um ou mais espacos-brancos e que entre
       o float e o caractere ha um ou mais espacos-brancos.*/
4     int i;
5     float f;
6     char c;
7
8     scanf("%d %f %c", &i, &f, &c);
9     printf ("Os valores lidos sao: %d %f %c\n", i, f, c);
10
11     return(0);
12 }
```

Exemplo de execução:

Entrada
23(10 espaços)5.4(espacos ou tabulações ou quebras de linha)H

Saída
Os valores lidos sao: 23 5.4 H



Leitura Formatada - scanf()

Exemplo

```
1 #include<stdio.h>
2 int main(){
3     /* Ler um inteiro, um float e um caractere na entrada. Suponha que
       entre o float e o int ha um ou mais espacos-brancos e que entre
       o float e o caractere ha um ou mais espacos-brancos.*/
4     int i;
5     float f;
6     char c;
7
8     scanf("%d %f %c", &i, &f, &c);
9     printf ("Os valores lidos sao: %d %f %c\n", i, f, c);
10
11     return(0);
12 }
```

Exemplo de execução:

Entrada
23(10 espaços)5.4(espacos ou tabulações ou quebras de linha)H

Saída
Os valores lidos sao: 23 5.4 H



Leitura Formatada - scanf()

Onde Encontrar mais Informações

- As orientações acima são básicas sobre como usar a função `scanf()`.
- Há vários detalhes de como usar a função que a tornam poderosa para leitura de entrada com formato e que por questão de espaço e também por estarmos iniciando a aprendizagem da linguagem, foram omitidos.
- Uma importante fonte de informação de como usar a função `scanf()` pode ser encontrada em <http://www.cplusplus.com/reference/cstdio/scanf/>



Saída-Padrão - stdout

Entrada e
Saída na
Linguagem C

Thierson
Couto Rosa

Organização

Introdução

Entrada-
Padrão

Funções da
Entrada-Padrão

Saída-Padrão

- A libc implementa funções de saída para um dispositivo virtual de saída denominado *saída-padrão - stdout*
- A esse dispositivo podem ser associados quaisquer dispositivos físicos de saída (tela, impressora, disco rígido, pen-drive, cd, dvd, etc);
- Quando nenhum dispositivo é explicitamente associado ao stdout pelo programador, a biblioteca considera que esse dispositivo é a tela do computador. Portanto, por padrão, a escrita é feita na tela.



INSTITUTO DE
INFORMÁTICA
UFG

Entrada e
Saída na
Linguagem C

Thierson
Couto Rosa

Organização

Introdução

Entrada-
Padrão

Funções da
Entrada-Padrão

Saída-Padrão

Saída-Padrão como um Arquivo Texto

- A saída-padrão é tratada com um arquivo de tipo texto, isso é, gera uma sequência finita de caracteres.
- Sendo assim, as funções de saída convertem valores armazenados em variáveis, de acordo com os tipos das mesmas, em sequências de caracteres.
- Esses caracteres são codificados de acordo com a tabela ASCII.



Escrita de um Caractere

Entrada e
Saída na
Linguagem C

Thierson
Couto Rosa

Organização

Introdução

Entrada-
Padrão

Funções da
Entrada-Padrão

Saída-Padrão

```
int putchar ( int caractere );
```

- A função **putchar()** tem como parâmetro uma variável inteira ou uma constante caractere.
- A função **getchar()** converte o valor inteiro para **unsigned char** antes da impressão.



INSTITUTO DE
INFORMÁTICA
UFG

printf() - Exemplo

Entrada e
Saída na
Linguagem C

Thierson
Couto Rosa

Organização

Introdução

Entrada-
Padrão

Funções da
Entrada-Padrão

Saída-Padrão

Escrevendo um Caractere

```
1 #include<stdio.h>
2
3 int main(){
4     int caractere;
5     caractere = 'A';
6     //Lendo um caractere da entrada
7     caractere=getchar();
8     printf("Caractere Lido: ");
9     //Imprimindo o caractere lido
10    putchar(caractere);
11    putchar('\n');
12    return 0;
13 }
```

Entrada:

H



INSTITUTO DE
INFORMÁTICA
UFG

printf() - Exemplo

Entrada e
Saída na
Linguagem C

Thierson
Couto Rosa

Organização

Introdução

Entrada-
Padrão

Funções da
Entrada-Padrão

Saída-Padrão

Escrevendo um Caractere

```
1 #include<stdio.h>
2
3 int main(){
4     int caractere;
5     caractere = 'A';
6     //Lendo um caractere da entrada
7     caractere=getchar();
8     printf("Caractere Lido: ");
9     //Imprimindo o caractere lido
10    putchar(caractere);
11    putchar('\n');
12    return 0;
13 }
```

Entrada:

H

Saída:

Caractere Lido: H



Saída Formatada - printf()

int printf("formato", lista de expresoes)

- A função imprime a string *formato* na saída-padrão. Se a cadeia de formato contém um especificação de conversão (iniciando com %) O valor da expressão correspondente é convertido para uma cadeia de caracteres e impresso na posição onde a especificação ocorre na cadeia de formato.
- Exemplo: `printf ("%d e um numero inteiro\n", x);:`
 - irá substituir `%d` na cadeia de formato pela sequência de caracteres que representa o valor int armazeado em *x* (20).
 - saída: 20 e um numero inteiro;



Saída Formatada - printf()

Especificação de conversão

Os caracteres x em uma especificação de conversão $\%x$ podem ser:

Caractere	Valor impresso
d ou i	Valor inteiro com sinal da expressão
u	Valor inteiro sem sinal da expressão
x ou X	Valor inteiro da expressão sem sinal, convertido em hexadecimal.
f ou F	Valor float da expressão
e ou E	Valor da expressão em notação científica.
g ou G	A menor entre as representações %f e %e ou %F ou %E
c	imprime um caractere
s	imprime uma string (será estudado posteriormente)
a ou A	Imprime um valor float em hexadecimal.



Saída Formatada - printf()

Especificação de precisão

- No caso de um número float ou double, pode-se especificar o número x de casas decimais usando `%.xf` na especificação.
- Exemplo: suponha que pi seja do tipo float e esteja armazenando o valor 3.1416. Podemos imprimir o número com somente duas casas decimais:

```
printf("O valor de pi e: %.2fn", pi);
```
- saída : O valor de pi e: 3.14



INSTITUTO DE
INFORMÁTICA
UFG

Entrada e
Saída na
Linguagem C

Thierson
Couto Rosa

Organização

Introdução

Entrada-
Padrão

Funções da
Entrada-Padrão

Saída-Padrão

Saída Formatada - printf()

Exemplo

```
1 #include<stdio.h>
2 int main(){
3 float x=3.25291;
4 int y= 50;
5 char c = 'A';
6 printf("O valor de x e: %.3f, o valor de y e: %d e o valor de c e: %c\n", x
, y, c);
7 //Cuidado. O formato escolhido pode mudar a saída completamente:
8 printf("O valor de x e: %d, o valor de y e: %u e e o valor de c e: %d\n", x
, y, c);
9 }
```



Saída Formatada - printf()

Exemplo

```
1 #include<stdio.h>
2 int main(){
3 float x=3.25291;
4 int y= 50;
5 char c = 'A';
6 printf("O valor de x e: %.3f, o valor de y e: %d e o valor de c e: %c\n", x
, y, c);
7 //Cuidado. O formato escolhido pode mudar a saída completamente:
8 printf("O valor de x e: %d, o valor de y e: %u e e o valor de c e: %d\n", x
, y, c);
9 }
```

Execução do Programa:

Saída 1:

O valor de x e: 3.253, o valor de y e: -234 e o valor de c e: A



Saída Formatada - printf()

Exemplo

```
1 #include<stdio.h>
2 int main(){
3     float x=3.25291;
4     int y= 50;
5     char c = 'A';
6     printf("O valor de x e: %.3f, o valor de y e: %d e o valor de c e: %c\n", x
           , y, c);
7     //Cuidado. O formato escolhido pode mudar a saída completamente:
8     printf("O valor de x e: %d, o valor de y e: %u e e o valor de c e: %d\n", x
           , y, c);
9 }
```

Execução do Programa:

Saída 1:

O valor de x e: 3.253, o valor de y e: -234 e o valor de c e: A

Saída 2:

O valor de x e: -234, o valor de y e: 65 e e o valor de c e: 1071100579



INSTITUTO DE
INFORMÁTICA
UFG

Saída Formatada - printf()

Entrada e
Saída na
Linguagem C

Thierson
Couto Rosa

Organização

Introdução

Entrada-
Padrão

Funções da
Entrada-Padrão

Saída-Padrão

Onde Encontrar mais Informações

- Há vários detalhes de como usar a função que a tornam poderosa para escrever saída formatada que não apresentamos aqui. da linguagem, foram omitidos.
- Uma importante fonte de informação de como usar a função **printf()** pode ser encontrada em <http://www.cplusplus.com/reference/cstdio/printf/>