
RESUMO DE ESPECIFICADORES DE FORMATO DE ENTRADA E SAÍDA

B

APÊNDICE

B.1 Introdução

Este apêndice apresenta resumidamente especificadores de formato que fazem parte de strings de formatação utilizados pelas duas famílias de funções da biblioteca padrão de C que realizam entrada e saída formatadas:

- Família **printf** – da qual fazem parte as funções: **fprintf()**, **fwprintf()**, **printf()**, **snprintf()**, **sprintf()**, **swprintf()**, **vfprintf()**, **vwfprintf()**, **vprintf()**, **vsnprintf()**, **vsprintf()**, **vswprintf()**, **wprintf()** e **wvprintf()**.
- Família **scanf** – da qual fazem parte as funções: **fscanf()**, **fwscanf()**, **scanf()**, **sscanf()**, **swscanf()**, **vfscanf()**, **vwscanf()**, **vscanf()**, **vsscanf()**, **vswscanf()**, **vwscanf()** e **wscanf()**.

Os membros de cada uma destas famílias de funções utilizam quase os mesmos *strings* de formatação¹ e ambas as famílias utilizam especificadores de formato semelhantes. Apesar de haver grande semelhança entre os especificadores de formato utilizados por estas famílias de funções, fique atento, pois existem diferenças importantes, algumas das quais bastante sutis. Por exemplo, os especificadores `%d` e `%i` têm o mesmo significado

¹ A diferença entre os strings de formatação utilizados pelos membros de cada família está no uso ou não de strings extensos. Funções que possuem o caractere “w” em seu nome utilizam strings de formatação extensos.

para integrantes da família `printf`, mas este não é o caso para membros da família `scanf`. Por outro lado, a família `scanf` possui especificador de formato para o tipo `float`, mas o mesmo não ocorre com a família `printf`.

Neste apêndice, são apresentadas apenas tabelas resumidas contendo todos os especificadores de formato *simples* que podem ser usados com as famílias de funções `scanf` e `printf`, mas existem muito mais elementos que podem ser utilizados na composição de um especificador de formato para ambas as famílias. Estes componentes são detalhadamente apresentados no **Volume II**. As tabelas apresentadas aqui podem ser utilizadas como uma rápida referência para estes especificadores de formato.

B.2 Resumo de Especificadores de Formato da Família `printf`

A **Tabela 64** a seguir apresenta especificadores de formato simplificados que podem ser usados com membros da família `printf`. Nesta tabela, as colunas devem ser interpretadas do seguinte modo:

- **Especificador** – consiste em um tipo de especificador precedido por um prefixo. Todas as combinações possíveis de tipos de especificadores para a família `printf` são apresentados na **Tabela 64**.
- **Tipo de Valor** – como o valor a ser escrito é interpretado².
- **Valor Convertido** – conversão que é aplicada ao valor a ser impresso.
- **Base** – a base numérica na qual os valores numéricos são escritos. Não faz sentido para valores não-numéricos.
- **Precisão** – a precisão padrão assumida quando a precisão não é explicitamente especificada (v. **Volume II**).

² A exceção é o tipo de especificador `%n` (v. **Volume II**).

FAMÍLIA PRINTF				
ESPECIFICADOR	TIPO DE VALOR	VALOR CONVERTIDO	BASE	PRECISÃO
%a (C99)	double x	(double)x	10	6
%La (C99)	long double x	(long double)x	10	6
%A (C99)	double x	(double)x	10	6
%LA (C99)	long double x	(long double)x	10	6
%c	int x	(unsigned char)x		
%lc	wint_t x	(wchar_t)x		
%d	int x	(int)x	10	1
%hd	int x	(short)x	10	1
%ld	long x	(long)x	10	1
%hhd (C99)	int x	(signed char)x	10	1
%jd (C99)	intmax_t x	(intmax_t)x	10	1
%lld (C99)	long long x	(long long)x	10	1
%td (C99)	ptrdiff_t x	(ptrdiff_t)x	10	1
%zd (C99)	size_t x	(ptrdiff_t)x	10	1
%e	double x	(double)x	10	6
%Le	long double x	(long double)x	10	6
%E	double x	(double)x	10	6
%LE	long double x	(long double)x	10	6
%f	double x	(double)x	10	6
%Lf	long double x	(long double)x	10	6
%F (C99)	double x	(double)x	10	6
%LF (C99)	long double x	(long double)x	10	6
%g	double x	(double)x	10	6
%Lg	long double x	(long double)x	10	6
%G	double x	(double)x	10	6
%LG	long double x	(long double)x	10	6

%i	int x	(int)x	10	1
%hi	int x	(short)x	10	1
%li	long x	(long)x	10	1
%hhi (C99)	int x	(signed char)x	10	1
%ji (C99)	intmax_t x	(intmax_t)x	10	1
%lli (C99)	long long x	(long long)x	10	1
%ti (C99)	ptrdiff_t x	(ptrdiff_t)x	10	1
%zi (C99)	size_t x	(ptrdiff_t)x	10	1
%n	int *x			
%hn	short *x			
%ln	long *x			
%hhn (C99)	int *x			
%jn (C99)	intmax_t *x			
%lln (C99)	long long *x			
%tn (C99)	ptrdiff_t *x			
%zn (C99)	size_t *x			
%o	int x	(unsigned int)x	8	1
%ho	int x	(unsigned short)x	8	1
%lo	long x	(unsigned long)x	8	1
%hho (C99)	int x	(unsigned char)x	8	1
%jo (C99)	intmax_t x	(uintmax_t)x	8	1
%llo (C99)	long long x	(unsigned long long)x	8	1
%to (C99)	ptrdiff_t x	(size_t)x	8	1
%zo (C99)	size_t x	(size_t)x	8	1
%p	void *x	(void *)x		
%s	const char *x	(const char *)x		
%ls	const wchar_t *x	(const wchar_t *)x		
%u	int x	(unsigned int)x	10	1

%hu	int x	(unsigned short)x	10	1
%lu	long x	(unsigned long)x	10	1
%hhu (C99)	int x	(unsigned char)x	10	1
%ju (C99)	intmax_t x	(uintmax_t)x	10	1
%llu (C99)	long long x	(unsigned long long)x	10	1
%tu (C99)	ptrdiff_t x	(size_t)x	10	1
%zu (C99)	size_t x	(size_t)x	10	1
%x	int x	(unsigned int)x	16	1
%hx	int x	(unsigned short)x	16	1
%lx	long x	(unsigned long)x	16	1
%hhx (C99)	int x	(unsigned char)x	16	1
%jx (C99)	intmax_t x	(uintmax_t)x	16	1
%llx (C99)	long long x	(unsigned long long)x	16	1
%tx (C99)	ptrdiff_t x	(size_t)x	16	1
%zx (C99)	size_t x	(size_t)x	16	1
%X	int x	(unsigned int)x	16	1
%hX	int x	(unsigned short)x	16	1
%lX	long x	(unsigned long)x	16	1
%hhX (C99)	int x	(unsigned char)x	16	1
%jX (C99)	intmax_t x	(uintmax_t)x	16	1
%llX (C99)	long long x	(unsigned long long)x	16	1
%tX (C99)	ptrdiff_t x	(size_t)x	16	1
%zX (C99)	size_t x	(size_t)x	16	1
%%	Nenhum	(char)' %'		

Tabela 64: Especificadores de formato usados pela família printf

B.3 Resumo de Especificadores de Formato da Família scanf

A Tabela 65 a seguir apresenta especificadores de formato simplificados que podem ser usados com membros da família scanf. Nesta tabela, as colunas devem ser interpretadas do seguinte modo:

- **Especificador** – consiste em um tipo de especificador precedido por um prefixo. Todas as combinações possíveis de tipos de especificadores para a família scanf são apresentadas na **Tabela 65**.
- **Tipo de parâmetro** – o tipo do parâmetro, presente na lista de argumentos variáveis da função, que deve casar com o respectivo especificador.
- **Função de conversão** – função de conversão chamada para converter os caracteres lidos para o tipo da variável para onde o parâmetro respectivo aponta. Detalhes sobre o funcionamento das funções de conversão mencionadas são apresentados no **Volume II**.
- **Base** – a base numérica na qual os valores numéricos são lidos. Não faz sentido para valores não-numéricos. Quando este valor for 0, a base do número será determinada pelo formato do mesmo (v. **Volume II**).

FAMÍLIA SCANF			
ESPECIFICADOR	TIPO DE PARÂMETRO	FUNÇÃO DE CONVERSÃO	BASE
%a (C99)	float *x	strttof	10
%la (C99)	double *x	strtod	10
%La (C99)	long double *x	strtold	10
%A (C99)	float *x	strttof	10
%lA (C99)	double *x	strtod	10
%LA (C99)	long double *x	strtold	10
%c	char x[]		
%lc	wchar_t x[]		
%d	int *x	strtol	10
%hd	short *x	strtol	10
%ld	long *x	strtol	10

%hhd (C99)	int *x	strtoul	10
%jd (C99)	intmax_t *x	strtouimax	10
%lld (C99)	long long *x	strtoll	10
%td (C99)	ptrdiff_t *x	strtouimax	10
%zd (C99)	size_t *x	strtouimax	10
%e (Alteração/C99)	float *x	strtof	10
%le	double *x	strtod	10
%Le (Alteração/C99)	long double *x	strtold	10
%E (Alteração/C99)	float *x	strtof	10
%lE	double *x	strtod	10
%LE (Alteração/C99)	long double *x	strtold	10
%f (Alteração/C99)	float *x	strtof	10
%lf	double *x	strtod	10
%Lf (Alteração/C99)	long double *x	strtold	10
%F (C99)	float *x	strtof	10
%lF (C99)	double *x	strtod	10
%LF (C99)	long double *x	strtold	10
%g (Alteração/C99)	float *x	strtof	10
%lg	double *x	strtod	10
%Lg (Alteração/C99)	long double *x	strtold	10
%G (Alteração/C99)	float *x	strtof	10
%lG	double *x	strtod	10
%LG (Alteração/C99)	long double *x	strtold	10
%i	int *x	strtoul	0
%hi	short *x	strtoul	0
%li	long *x	strtoul	0
%hhi (C99)	int *x	strtoul	0
%ji (C99)	intmax_t *x	strtouimax	0
%lli (C99)	long long *x	strtoll	0

%ti (C99)	ptrdiff_t *x	strtoumax	0
%zi (C99)	size_t *x	strtoumax	0
%n	int *x		
%hn	short *x		
%ln	long *x		
%hhn (C99)	int *x		
%jn (C99)	intmax_t *x		
%lln (C99)	long long *x		
%tn (C99)	ptrdiff_t *x		
%zn (C99)	size_t *x		
%o	unsigned int *x	strtoul	8
%ho	unsigned short *x	strtoul	8
%lo	unsigned long *x	strtoul	8
%hho (C99)	unsigned int *x	strtoul	8
%jo (C99)	uintmax_t *x	strtoumax	8
%llo (C99)	unsigned long long *x	strtoull	8
%to (C99)	ptrdiff_t *x	strtoumax	8
%zo (C99)	size_t *x	strtoumax	8
%p	void **x		
%s	char x[]		
%ls	wchar_t x[]		
%u	unsigned int *x	strtoul	10
%hu	unsigned short *x	strtoul	10
%lu	unsigned long *x	strtoul	10
%hhu (C99)	unsigned int *x	strtoul	10
%ju (C99)	uintmax_t *x	strtoumax	10
%llu (C99)	unsigned long long *x	strtoull	10
%tu (C99)	ptrdiff_t *x	strtoumax	10
%zu (C99)	size_t *x	strtoumax	10

%x	unsigned int *x	strtoul	16
%hx	unsigned short *x	strtoul	16
%lx	unsigned long *x	strtoul	16
%hhx (C99)	unsigned int *x	strtoul	16
%jx (C99)	uintmax_t *x	strtoumax	16
%llx (C99)	unsigned long long *x	strtoull	16
%tx (C99)	ptrdiff_t *x	strtoumax	16
%zx (C99)	size_t *x	strtoumax	16
%X	unsigned int *x	strtoul	16
%hX	unsigned short *x	strtoul	16
%lX	unsigned long *x	strtoul	16
%hhX (C99)	unsigned int *x	strtoul	16
%jX (C99)	uintmax_t *x	strtoumax	16
%llX (C99)	unsigned long long *x	strtoull	16
%tX (C99)	ptrdiff_t *x	strtoumax	16
%zX (C99)	size_t *x	strtoumax	16
%[...]	char x[]		
%l[...]	wchar_t x[]		
%%	Nenhum		

Tabela 65: Especificadores de formato usados pela família scanf