



Sistemas de Numeração

1 – Sistemas de numeração

Sistemas de numeração são sistemas para representação de números. Historicamente vários sistemas de numeração foram desenvolvidos de acordo com os avanços das técnicas matemáticas e da própria civilização. Uma das hipóteses de que tenhamos desenvolvido o sistema decimal de numeração está relacionada à contagem com os 10 dedos das mãos.

Na era digital acostumamo-nos a ouvir falar de *bits* ou *bytes*. Sabemos que **bit** é a menor unidade de informação na memória do computador (0 ou 1, aberto ou fechado, falso ou verdadeiro). Um **byte** corresponde a 8 bits, mas um byte consegue armazenar um código que corresponde somente a um único número, letra ou símbolo. A combinação de cadeias de bits é que permite representar diferentes números agrupando-se na forma de bytes.

É importante ter uma boa noção de notação e bases numéricas quando queremos estudar Computação porque isso tem muitos aproveitamentos na área.

Notação e bases

A quantidade de algarismos disponíveis num sistema de numeração designa-se de base.



Decimal (base 10)

0, 1, 2, 3, 4, 5, 6, 7, 8, 9



Binário (base 2)



0, 1



Octal (base 8)

0, 1, 2, 3, 4, 5, 6, 7



Hexadecimal (base 16)


0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F

O sistema decimal é o sistema comumente utilizado por nós seres humanos e é constituído por dez algarismos: **0,1,2,3,4,5,6,7,8,9**.

Sabemos que no sistema decimal cada algarismo tem um valor posicional, ou seja, cada algarismo tem um peso de acordo com a sua posição na representação de um dado valor.

Por exemplo, o número 237 na base 10 pode ser decomposto em:

Peso	10^2	10^1	10^0
	2	3	7
	↙	↓	↘
	2×10^2	3×10^1	7×10^0
	$= 237$		

O **sistema binário** é o sistema mais utilizado por máquinas atualmente uma vez que os sistemas digitais trabalham internamente  em dois estados (ligado/desligado, verdadeiro/falso, aberto/fechado). O sistema binário utiliza os símbolos: **0, 1**, sendo cada símbolo designado por bit (*binary digit*). Um equipamento digital é um grande manipulador de 0's e 1's.

O **sistema octal** é um sistema de numeração de base 8, isto é, utiliza 8 símbolos (**0, 1, 2, 3, 4, 5, 6, 7**) para a representação de um determinado valor.

O **sistema hexadecimal** é muito utilizado na programação de microprocessadores (espécie de pequeno computador), especialmente nos equipamentos de estudo e sistemas de desenvolvimento. Utiliza os símbolos: **0, 1, 2, 3, 4, 5, 6, 7, 8, 9** do sistema decimal e ainda as letras **A, B, C, D, E, F**.

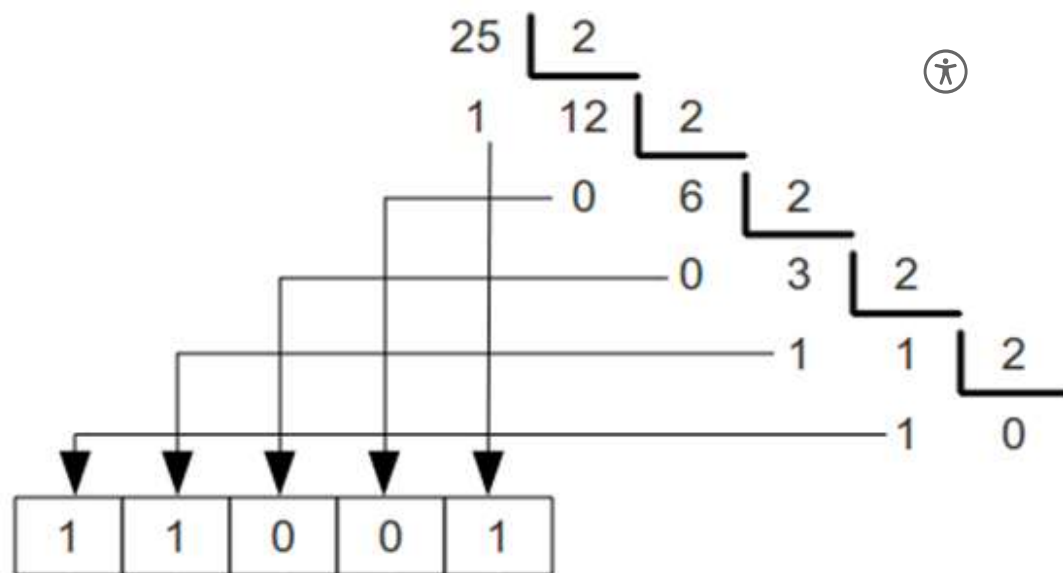
Importante notar as equivalências: A=10, B=11, C=12, D=13, E=14 e F=15.

2 – Conversões de base decimal

Conversão de decimal para binário

Basta dividir sucessivamente por 2 o número decimal e guardar os quocientes que vão sendo obtidos, até que o quociente de uma das divisões seja 0.

O resultado é a sequência de baixo para cima de todos os restos obtidos, conforme ilustra a figura em que é apresentado 25 na base 10 que se torna 11001 na base 2.



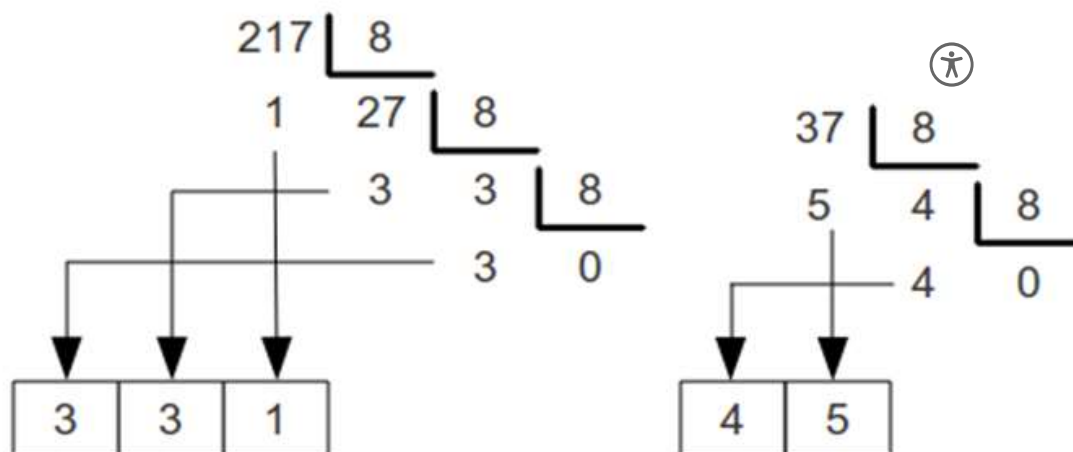
A relação entre números decimais e binários é extremamente comum e importante, dada resumidamente pela tabela abaixo para alguns valores:

Decimal	Binário		Decimal	Binário
0	0		11	1011
1	1		12	1100
2	10		13	1101
3	11		14	1110
4	100		15	1111
5	101		16	10000
6	110		17	10001
7	111		18	10010
8	1000		19	10011
9	1001		20	10100
10	1010	

Conversão de decimal para octal

Neste caso o processo de conversão se dá por divisões sucessivas por 8.

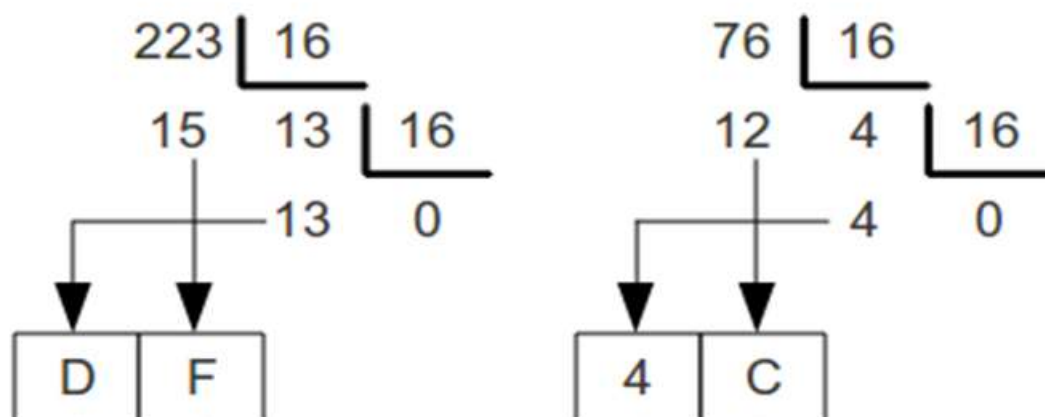
O resultado é a sequência de baixo para cima de todos os restos obtidos, conforme as imagens abaixo em que transformamos o número 217 (decimal) que vira 331 em octal e o número 37 (decimal) que vira 45 (octal)



Conversão de decimal para hexadecimal

Desta vez o processo se dá por sucessivas divisões por 16, aproveitando-se o resto conforme abaixo.

O número 223(10) corresponde a DF(16) e o número 76(10) corresponde a 4C(16).



A tabela abaixo apresenta uma série de números em ordem crescente escritos em decimal e a representação dos mesmos na base hexadecimal:

Decimal	Hexa		Decimal	Hexa		Decimal	Hexa
0	0		11	B		22	16
1	1		12	C		23	17
2	2		13	D		24	18
3	3		14	E		25	19
4	4		15	F		26	1A
5	5		16	10		27	1B
6	6		17	11		28	1C
7	7		18	12		29	1D
8	8		19	13		30	1E
9	9		20	14		31	1F
10	A		21	15		32	20

3 - Conversões de base binária, octal e hexadecimal

Conversão de binário para decimal:

É apenas resultado da soma do dígito (0 ou 1) multiplicado pela base 2 com o expoente de acordo com sua posição, começando do zero a partir da direita. Neste caso o número 11001 (base 2) corresponde a 25 (base 10), conforme ilustrado abaixo:

1	1	0	0	1
1×2^4	1×2^3	0×2^2	0×2^1	1×2^0
$16 + 8 + 0 + 0 + 1 = 25$				

1	1	0	1	0	1
1	1	0	1	0	1
1×2^2	1×2^1	0×2^0	1×2^2	0×2^1	1×2^0
4 + 2 + 0			4 + 0 + 1		
$6_{10} = 6_8$			$5_{10} = 5_8$		
65					

	1	1	1	1	1
0	1	1	1	1	1
0×2^2	1×2^1	1×2^0	1×2^2	1×2^1	1×2^0
$0 + 2 + 1$			$4 + 2 + 1$		
$3_{10} = 3_8$			$7_{10} = 7_8$		
37					

Conversão de binário para octal:

Neste caso, os dígitos do número binário são separados em grupos de 3 bits da direita para a esquerda. Cada grupo de 3 bits é um dígito em octal. Ao final, une-se os resultados.

Caso o número de dígitos do número binário não seja múltiplo de 3, completa-se os dígitos à esquerda com zeros (0).

Veja o exemplo abaixo para conversão do número 110101 (base binária) para o número 65 (base octal) e o número 11111 (base 2) para 37 (base 8).

Binário -> hexadecimal

Conversão de binário para hexadecimal:

Separa-se o número binário em grupos de 4 bits, da direita para a esquerda. Em seguida, transforma-se cada grupo de 4 bits em hexadecimal. Ao final,

simplesmente une-se os resultados em um só. Caso o número de dígitos do número binário não seja múltiplo de 4, completa-se os dígitos à esquerda com zeros (0).


Veja o exemplo abaixo para conversão do número 01011011 (base 2 ou binária) para o número 5B (base 16 ou hexa) e o número 111101 (base 2) para 3D (base 16):

0	1	0	1	1	0	1	1
0	1	0	1	1	0	1	1
0×2^3	1×2^2	0×2^1	1×2^0	1×2^3	0×2^2	1×2^1	1×2^0
$0 + 4 + 0 + 1 = 5_{10}$				$8 + 0 + 2 + 1 = 11_{10}$			
$5_{10} = 5_{16}$				$11_{10} = B_{16}$			
5B							

		1	1	1	1	0	1
0	0	1	1	1	1	0	1
0×2^3	0×2^2	1×2^1	1×2^0	1×2^3	1×2^2	0×2^1	1×2^0
$0 + 0 + 2 + 1 = 3_{10}$				$8 + 4 + 0 + 1 = 13_{10}$			
$3_{10} = 3_{16}$				$13_{10} = D_{16}$			
3D							

4 – Conversão de bases

Conversão octal para decimal

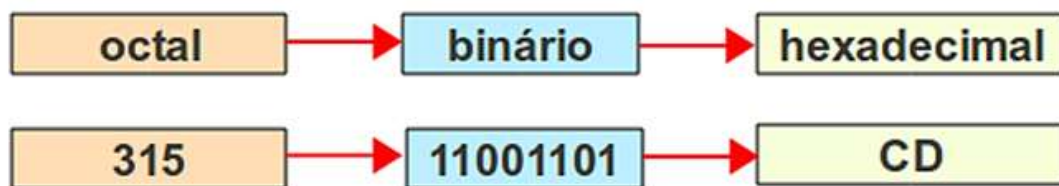
Obtida através da soma dos dígitos do número octal multiplicados nela base 8 elevada à posição colunar do dígito, começando em 0 da c  , para a esquerda. Observe os exemplos para os números 331 (base 8) e 45 (base 8).

3	3	1
3×8^2	3×8^1	1×8^0
$192 + 24 + 1 = 217$		

4	5
4×8^1	5×8^0
$32 + 5 = 37$	

Conversão de octal para hexadecimal

Transforma-se primeiro o octal em binário e em seguida o binário em hexadecimal:



Conversão hexadecimal para decimal

Realizada através da soma dos dígitos hexadecimais multiplicados pela base 16 elevada à posição colunar contando da direita para a esquerda, começando em 0, de forma semelhante à conversão de binários em decimais:

1	0	A
1×16^2	0×16^1	10×16^0
$256 + 0 + 10 = 266$		

C	B
12×16^1	11×16^0
$192 + 11 = 203$	

Note que os caracteres que definem os dígitos hexadecimais A, B e C foram substituídos pelos valores equivalentes em decimais 10, 11 e 12, de acordo com a tabela da lição anterior para a realização do cálculo.

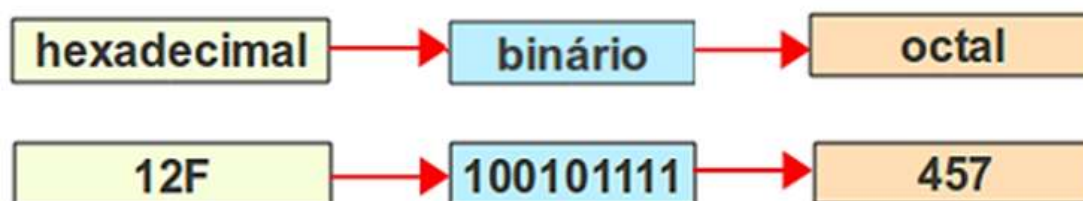
Conversão hexadecimal para binário

Decompõem-se o número hexadecimal diretamente em binários de 4 dígitos. Os zeros mais à esquerda do resultado binário podem ser omitidos:

1				2				F			
$0 + 0 + 0 + 1 = 1$				$0 + 0 + 2 + 0 = 2$				$8 + 4 + 2 + 1 = 15$			
0×2^3	0×2^2	0×2^1	1×2^0	0×2^3	0×2^2	1×2^1	0×2^0	1×2^3	1×2^2	1×2^1	1×2^0
0	0	0	1	0	0	1	0	1	1	1	1
			1	0	0	1	0	1	1	1	1

Conversão hexadecimal para octal

Transforma-se primeiro o hexadecimal em binário e em seguida o binário em octal:



4 – Grandezas e tabela ASCII

Sempre vale a pena rever a ideia de grandezas e suas nomenclaturas:

Nome	Simbolo	Tamanho
Byte	B	8 Bit
KiloByte	KB	1024 Byte
MegaByte	MB	1024 KByte
GigaByte	GB	1024 MByte
TeraByte	TB	1024 GByte
PetaByte	PB	1024 TByte
ExaByte	EB	1024 PByte
ZettaByte	ZB	1024 EByte
YottaByte	YB	1024 ZByte

Tabela ASCII

A tabela ASCII representa o equivalente de uma série de símbolos e caracteres presentes em inúmeros programas e aplicações:

Dec	Hx	Oct	Char	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr
0	0	000	NUL (null)	32	20	040	 	Space	64	40	100	@	@	96	6f	177	`	`
1	1	001	SOH (start of heading)	33	21	041	!	!	65	41	101	A	A	97	61	178	a	a
2	2	002	STX (start of text)	34	22	042	"	"	66	42	102	B	B	98	62	179	b	b
3	3	003	ETX (end of text)	35	23	043	#	#	67	43	103	C	C	99	63	180	c	c
4	4	004	EOT (end of transmission)	36	24	044	$	\$	68	44	104	D	D	100	64	181	d	d
5	5	005	ENQ (enquiry)	37	25	045	%	%	69	45	105	E	E	101	65	182	e	e
6	6	006	ACK (acknowledge)	38	26	046	&	&	70	46	106	F	F	102	66	183	f	f
7	7	007	BEL (bell)	39	27	047	'	'	71	47	107	G	G	103	67	184	g	g
8	8	010	BS (backspace)	40	28	050	((72	48	110	H	H	104	68	185	h	h
9	9	011	TAB (horizontal tab)	41	29	051))	73	49	111	I	I	105	69	186	i	i
10	A	012	LF (NL line feed, new line)	42	2A	052	*	*	74	4A	112	J	J	106	6A	187	j	j
11	B	013	VT (vertical tab)	43	2B	053	+	+	75	4B	113	K	K	107	6B	188	k	k
12	C	014	FF (NP form feed, new page)	44	2C	054	,	,	76	4C	114	L	L	108	6C	189	l	l
13	D	015	CR (carriage return)	45	2D	055	-	-	77	4D	115	M	M	109	6D	190	m	m
14	E	016	SO (shift out)	46	2E	056	.	.	78	4E	116	N	N	110	6E	191	n	n
15	F	017	SI (shift in)	47	2F	057	/	/	79	4F	117	O	O	111	6F	192	o	o
16	10	020	DLE (data link escape)	48	30	060	0	0	80	50	120	P	P	112	70	193	p	p
17	11	021	DC1 (device control 1)	49	31	061	1	1	81	51	121	Q	Q	113	71	194	q	q
18	12	022	DC2 (device control 2)	50	32	062	2	2	82	52	122	R	R	114	72	195	r	r
19	13	023	DC3 (device control 3)	51	33	063	3	3	83	53	123	S	S	115	73	196	s	s
20	14	024	DC4 (device control 4)	52	34	064	4	4	84	54	124	T	T	116	74	197	t	t
21	15	025	NAK (negative acknowledge)	53	35	065	5	5	85	55	125	U	U	117	75	198	u	u
22	16	026	SYN (synchronous idle)	54	36	066	6	6	86	56	126	V	V	118	76	199	v	v
23	17	027	ETB (end of trans. block)	55	37	067	7	7	87	57	127	W	W	119	77	200	w	w
24	18	030	CAN (cancel)	56	38	070	8	8	88	58	130	X	X	120	78	201	x	x
25	19	031	EM (end of medium)	57	39	071	9	9	89	59	131	Y	Y	121	79	202	y	y
26	1A	032	SUB (substitute)	58	3A	072	:	:	90	5A	132	Z	Z	122	7A	203	z	z
27	1B	033	ESC (escape)	59	3B	073	;	:	91	5B	133	[[123	7B	204	{	{
28	1C	034	FS (file separator)	60	3C	074	<	<	92	5C	134	\	\	124	7C	205	|	
29	1D	035	GS (group separator)	61	3D	075	=	=	93	5D	135]]	125	7D	206	}	}
30	1E	036	RS (record separator)	62	3E	076	>	>	94	5E	136	^	^	126	7E	207	~	~
31	1F	037	US (unit separator)	63	3F	077	?	?	95	5F	137	_	_	127	7F	208		DEL

Source: www.LookupTables.com

Dica

Um ponto importante a destacar é que existem inúmeras ferramentas de conversão de números entre diferentes bases. Nosso estudo até aqui teve o objetivo de mostrar como é o processo de conversão no sentido de você entender melhor o significado e a importância das bases numéricas. Uma delas é apresentada abaixo (<https://www.cjdinfo.com.br/utilitario-conversor-bases-numericas#:~:text=CJDinfo%20%2D%20Conversor%20de%20Bases%20Num%C3%A9ricas>).



Conversor de Bases Numéricas

Esta página faz a conversão de bases numéricas de 2 a 36, incluindo *binário*(2), *octal*(8), *decimal*(10) e *hexadecimal*(16), utilizando as rotinas genéricas apresentadas no item [Soluções](#).

Entre com a Base e o Valor de Entrada. Indique a Base de Saída e Clique em Converter.

Valores muito altos poderão provocar *Overflow* ou perda de dados na conversão.

Base de Entrada:

Valor de Entrada:

Base de Saída:

Valor de Saída:

Converter



Referência Bibliográfica

BROOKSHEAR, J.G. **Ciência da Computação: uma visão abrangente**. Porto Alegre: Bookman, 2013.

FORBELLONE, A.L.V. & EBERSPACHER, H. F. **Lógica de Programação – A Construção de Algoritmos e Estruturas de Dados**. 3ª. Edição. São Paulo, SP: Prentice Hall, 2005.

Ir para exercício