ASCII e OPERAÇÃO DE SOMA e SUBTRAÇÃO COM SISTEMA COMPUTACIONAL

```
   *  * * *
   1 1 0 1 1 1 0
-      1 0 1 1 1
  ----------------
= 1 0 1 0 1 1 1
```

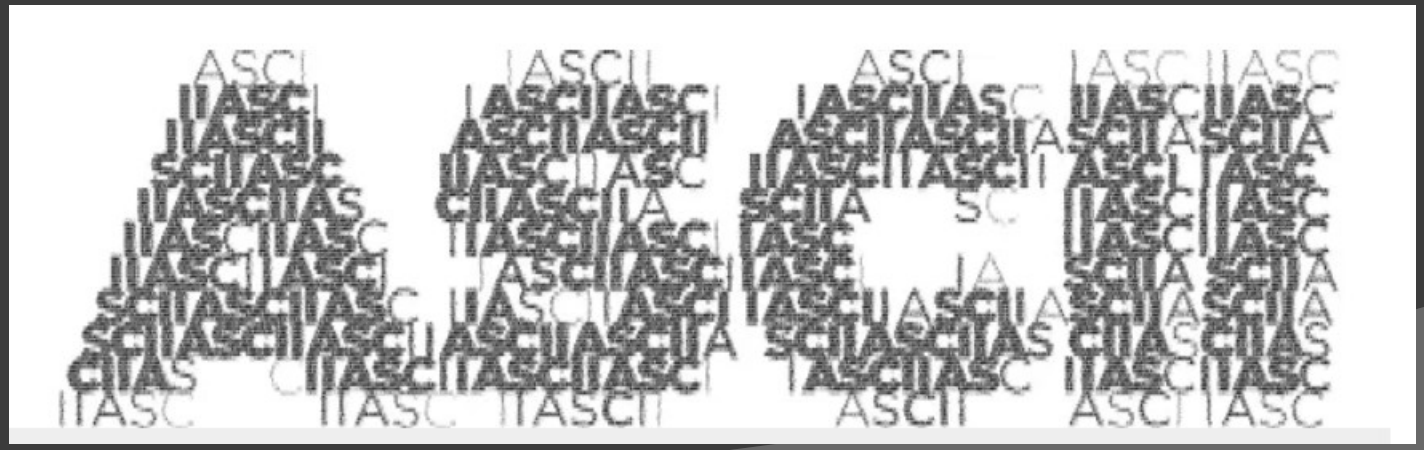
Como visto anteriormente o cálculo de conversão de bases está para responder às questões pertinentes à execução das especificações nas configurações de sistemas, comunicação remota e linguagem de máquina.

Hardware de Computadores

O que é o código ASCII e para que serve?

O ASCII é um código que foi proposto por Robert W. Bemer como uma solução para unificar a representação de caracteres alfanuméricos em computadores.

Antes de 1960 cada computador utilizava uma regra diferente para representar estes caracteres e o código ASCII nasceu para se tornar comum entre todas as máquinas.



De onde vem?

O nome ASCII vem do inglês *American Standard Code for Information Interchange* ou ”Código Padrão Americano para o Intercâmbio de Informação”. Ele é baseado no alfabeto romano e sua função é padronizar a forma como os computadores representam letras, números, acentos, sinais diversos e alguns códigos de controle.

No ASCII existem apenas 95 caracteres que podem ser impressos, eles são numerados de 32 a 126 sendo os caracteres de 0 a 31 reservados para funções de controle. Ou seja, funções de computador.

Alguns caracteres acabaram caindo em desuso pois eram funções específicas para computadores da época como o Teletype (máquinas de escrever eletro-mecânicas), fitas de papel perfurado e impressoras de cilindro.

A tabela...

Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char
0	0	[NULL]	32	20	[SPACE]	64	40	@	96	60	`
1	1	[START OF HEADING]	33	21	!	65	41	A	97	61	a
2	2	[START OF TEXT]	34	22	"	66	42	B	98	62	b
3	3	[END OF TEXT]	35	23	#	67	43	C	99	63	c
4	4	[END OF TRANSMISSION]	36	24	$	68	44	D	100	64	d
5	5	[ENQUIRY]	37	25	%	69	45	E	101	65	e
6	6	[ACKNOWLEDGE]	38	26	&	70	46	F	102	66	f
7	7	[BELL]	39	27	'	71	47	G	103	67	g
8	8	[BACKSPACE]	40	28	(	72	48	H	104	68	h
9	9	[HORIZONTAL TAB]	41	29	)	73	49	I	105	69	i
10	A	[LINE FEED]	42	2A	*	74	4A	J	106	6A	j
11	B	[VERTICAL TAB]	43	2B	+	75	4B	K	107	6B	k
12	C	[FORM FEED]	44	2C	,	76	4C	L	108	6C	l
13	D	[CARRIAGE RETURN]	45	2D	-	77	4D	M	109	6D	m
14	E	[SHIFT OUT]	46	2E	.	78	4E	N	110	6E	n
15	F	[SHIFT IN]	47	2F	/	79	4F	O	111	6F	o
16	10	[DATA LINK ESCAPE]	48	30	0	80	50	P	112	70	p
17	11	[DEVICE CONTROL 1]	49	31	1	81	51	Q	113	71	q
18	12	[DEVICE CONTROL 2]	50	32	2	82	52	R	114	72	r
19	13	[DEVICE CONTROL 3]	51	33	3	83	53	S	115	73	s
20	14	[DEVICE CONTROL 4]	52	34	4	84	54	T	116	74	t
21	15	[NEGATIVE ACKNOWLEDGE]	53	35	5	85	55	U	117	75	u
22	16	[SYNCHRONOUS IDLE]	54	36	6	86	56	V	118	76	v
23	17	[ENG OF TRANS. BLOCK]	55	37	7	87	57	W	119	77	w
24	18	[CANCEL]	56	38	8	88	58	X	120	78	x
25	19	[END OF MEDIUM]	57	39	9	89	59	Y	121	79	y
26	1A	[SUBSTITUTE]	58	3A	:	90	5A	Z	122	7A	z
27	1B	[ESCAPE]	59	3B	;	91	5B	[	123	7B	{
28	1C	[FILE SEPARATOR]	60	3C	<	92	5C	\	124	7C	|
29	1D	[GROUP SEPARATOR]	61	3D	=	93	5D	]	125	7D	}
30	1E	[RECORD SEPARATOR]	62	3E	>	94	5E	^	126	7E	~
31	1F	[UNIT SEPARATOR]	63	3F	?	95	5F	_	127	7F	[DEL]

Tabela de códigos ASCII (Foto: Reprodução/André Sugai)

Alguns exemplos de funções de controle seriam o LINE FEED que fazia com que a impressora avançasse seu papel, a função cancel e a função escape que até hoje é representada pela tecla ESC.

Caso você queira ver uma lista mais extensa dos caracteres ASCII visite o site Asciitable (asciitable.com).

E os desenhos?

Outro uso bem interessante dos códigos ASCII é para a criação de desenhos. Os códigos podem ser utilizados para representar qualquer tipo de imagem, coloridas ou não. Quem utilizava o mIRC talvez se lembre disso. Caso você queira experimentar algumas possibilidades em código ASCII hoje existem soluções online como o conversor de textos para ASCII como o Text to ASCII Art Generator ou o conversor de imagens Picascii.

No site Asciiart (asciiarte.com) é possível conferir uma galeria de imagens criadas utilizando o código ASCII, apesar de ser um código antigo ele ainda é muito útil e divertido.

Exemplo de ASCII Art com Mestre Yoda, de Start Wars (Foto: Reprodução/André Sugai)

O PROCESSAMENTO COM BINÁRIO

Os computadores têm suas características de processamento expressas em número de bits (8, 16, 32 ou 64).

Cada instrução enviada para o microprocessador pode ser formada por 1byte, 2 bytes, 3 bytes e 4 bytes. Assim, dependendo da instrução, são necessárias de 1 a 4 linhas de memória para armazená-la.

O espaço em disco ou memórias define-se como múltiplos de 1kbyte, em que 1kbyte é igual a 1024 bytes (2^{10}).

A tabela a seguir mostra os múltiplos do byte.

Múltiplos do Byte

Múltiplos do Byte	Abreviação	Valor
Quilobyte	KB	10^3 do byte (ou 1024 bytes)
Megabyte	MB	10^6 do byte (ou 1024 KB)
Gigabyte	GB	10^9 do byte (ou 1024 MB)
Terabyte	TB	10^{12} do byte (ou 1024 GB)
Petabyte	PB	10^{15} do byte (ou 1024 TB)

Álgebra booleana (de George Boole)

O sistema binário é base para a Álgebra booleana (de George Boole - matemático inglês), que permite fazer operações lógicas e aritméticas usando-se apenas dois dígitos ou dois estados (sim ou não, falso ou verdadeiro, tudo ou nada, 1 ou 0, ligado ou desligado).

A eletrônica digital e a computação estão baseadas no sistema binário e na lógica de Boole, o que permite representar por circuitos eletrônicos digitais (portas lógicas) os números, os caracteres e realizar operações lógicas e aritméticas.

Os programas de computadores são codificados sob forma binária e armazenados nas mídias (memórias, discos, etc.).

Um sistema numérico pode ser usado para realizar duas operações básicas: adição e subtração.

Pelo uso de adição e subtração, você pode então realizar multiplicações, divisões e qualquer outra operação numérica.

A figura a seguir resume as regras de adição com números binários.

Regra 1:	0 + 0 = 0	
Regra 2:	0 + 1 = 1	
Regra 3:	1 + 0 = 1	
Regra 4:	1 + 1 = 0	com transporte de 1
Regra 5:	1 + 1 + 1 = 1	com transporte de 1

Para ilustrar o processo de adição binária, vamos somar 1101 a 1101:

Transporte	1 1 0 1
Parcela	1 1 0 1
	+ 1 1 0 1
Soma	1 1 0 1 0

Operação com Binários

Soma de Binários

Para somar dois números binários, o procedimento é o seguinte:

Exemplo 1:

* *

$1\ 1\ 0\ 0_2$

$+\ \ 1\ 1\ 1_2$

$= 1\ 0\ 0\ 1\ 1_2$

Os números binários são base 2, ou seja, há apenas dois algarismos: 0 (zero) ou 1 (um). Na soma de 0 com 1, o resultado é 1.

Quando se soma 1 com 1, o resultado é 2, mas como 2 em binário é 10, o resultado é 0 (zero) e passa-se o outro 1 para a "frente", ou seja, para ser somado com o próximo elemento, conforme assinalado pelo asterisco.

Exercício..

Aula 05: Exercício

1. Realize as seguintes somas:
   - **a.** $1001_2 + 110_2$
   - **b.** $101011_2 + 101010_2$
   - **c.** $11111_2 + 1_2$
   - **d.** $1111_2 + 11_2$

Gabarito

1. Realize as seguintes somas:
   a. $1001_2 + 110_2 \rightarrow \mathbf{1111_2}$
   b. $101011_2 + 101010_2 \rightarrow \mathbf{1010101_2}$
   c. $11111_2 + 1_2 \rightarrow \mathbf{100000_2}$
   d. $1111_2 + 11_2 \rightarrow \mathbf{10010_2}$

Operação com binários

Subtração de Binários

```
          2 2
  0 2 0 0 0 2
  1 1 0 1 1 1 0
-     1 0 1 1 1
  ---------------
= 1 0 1 0 1 1 1
```

Quando temos 0 menos 1, precisamos "emprestar" do elemento vizinho. Esse empréstimo vem valendo 2 (dois), pelo fato de ser um número binário. Então, no caso da coluna 0 - 1 = 1, porque na verdade a operação feita foi 2 - 1 = 1. Esse processo se repete e o elemento que cedeu o "empréstimo" e valia 1 passa a valer 0. Os 1 marcam os elementos que "emprestaram" para seus vizinhos. Perceba que, logicamente, quando o valor for zero, ele não pode "emprestar" para ninguém, então o "pedido" passa para o próximo elemento e esse zero recebe o valor de 1.

Operação com binários

Subtração de Octais

Quando temos 0 menos 1 do octal, precisamos "emprestar" do elemento vizinho. Esse empréstimo vem valendo 8 (oito), pelo fato de ser um número octal. Então, no caso da coluna 0 - 1 = 7, porque na verdade a operação feita foi 8 - 1 = 7. Esse processo se repete e o elemento que cedeu o "empréstimo" e valia 1 passa a valer 0 do octal. Os *1* marcam os elementos que "emprestaram" para seus vizinhos. Perceba que, logicamente, quando o valor for zero, ele não pode "emprestar" para ninguém, então o "pedido" passa para o próximo elemento e esse zero recebe o valor de 1.

Operação com binários

Subtração de Hexadecimais

Quando temos 0 menos 1 do hexadecimal, precisamos "emprestar" do elemento vizinho. Esse empréstimo vem valendo 16 (dezesseis), pelo fato de ser um número hexadecimal. Então, no caso da coluna 0 - 16 = F, porque na verdade a operação feita foi 16 - 1 = 15. Esse processo se repete e o elemento que cedeu o "empréstimo" e valia 1 passa a valer 0 do hexadecimal. Os *1* marcam os elementos que "emprestaram" para seus vizinhos. Perceba que, logicamente, quando o valor for zero, ele não pode "emprestar" para ninguém, então o "pedido" passa para o próximo elemento e esse zero recebe o valor de 1.

Referências

- STALLINGS, Willian. Arquitetura e organização de computadores. 5. ed. Prentice Hall. São Paulo, 2006.
- TANENBAUM. Andrew S. Organização estruturada de computadores. 5. ed. Rio de Janeiro: LTC, 2007.
- MACHADO, Francis B.; MAIA, Luiz P. Arquitetura de sistemas operacionais. 4. ed. Rio de Janeiro: LTC, 2007.
- WEBER, Raul Fernando. Arquitetura de computadores pessoais. 2. ed. Porto Alegre: Sagra Luzzatto, 2003.
- _____. Fundamentos de arquitetura de computadores. 3. ed. Porto Alegre: Sagra Luzzatto, 2004.
- https://www.techtudo.com.br/noticias/noticia/2015/02/o-que-e-o-codigo-ascii-e-para-que-serve-descubra.html – acesso em 06/05/2020