



Arquitetura de Computadores em SI

- Painel
- Meus cursos
- Arquitetura de Computadores em SI 2021-2
- Codificação em binário
- Codificação Unicode e UTF-8

Codificação Unicode e UTF-8

Esta página faz uma rápida introdução aos conceitos de [caractere](#), [Unicode](#), [esquema de codificação](#), e [UTF-8](#).

Caracteres

Um *caractere* é um símbolo tipográfico usado para escrever texto em alguma língua. (Embora imperfeita, essa definição é suficiente para nossas necessidades.) Eis alguns exemplos de caracteres:

!" - 9 A B a b ~ À ã ç é ÿ Σ α — “

O número de caracteres usados pelas diferentes línguas do mundo é muito grande. O português usa apenas 127 caracteres e o inglês fica satisfeito com 94 desses. Mas não podemos nos limitar a essas duas línguas porque no mundo globalizado de hoje estamos expostos a muitas outras línguas, às vezes várias numa mesma sentença.

Para começar a organizar essa [Babel](#), é preciso dar *nomes* a todos os caracteres. O [consórcio Unicode](#) atribuiu *nomes numéricos* (conhecidos como [code points](#)) a mais de 1 milhão de caracteres. Segue uma minúscula amostra da lista de caracteres e seus números:

número	
Unicode	caractere
33	!
34	"
45	-
57	9
65	A
66	B
97	a
98	b
126	~
192	À
227	ã
231	ç
233	é
255	ÿ
931	Σ
945	α
8212	—
8220	“

Nessa amostra, os nomes numéricos dos caracteres estão escritos em notação decimal. Em geral, entretanto, esses números são escritos em notação [hexadecimal](#). Além disso, é usual acrescentar o prefixo “U+” a cada número:

número	
Unicode	caractere
U+0021	!
U+0022	"
U+002D	-
U+0039	9



U+0041	A
U+0042	B
U+0061	a
U+0062	b
U+007E	~
U+00C0	À
U+00E3	ã
U+00E7	ç
U+00E9	é
U+00FF	ÿ
U+03A3	Σ
U+03B1	α
U+2014	—
U+201C	“

A lista completa de caracteres e seus números Unicode pode ser vista na página [List of Unicode characters](#) da Wikipedia ou na página [Unicode / Character reference](#) do Wikibooks.

O conjunto de todos os caracteres da lista Unicode pode ser chamado *alfabeto Unicode* e cada caractere desse alfabeto pode ser chamado *caractere Unicode*. (Se a pretensão do projeto Unicode for justificada, *todos* os caracteres de *todas* as línguas do mundo são caracteres Unicode.)

É cômodo usar atalhos verbais óbvios ao falar de caracteres. Por exemplo, em vez de dizer “o caractere A” podemos dizer

- “o caractere cujo nome é U+0041”, ou
- “o caractere cujo número é U+0041” ou, mais simplesmente,
- “o caractere U+0041”.

Caracteres ASCII

Os primeiros 128 caracteres da lista Unicode são os mais usados. Esse conjunto de caracteres vai de U+0000 a U+007F e é conhecido como *alfabeto ASCII*. Os elementos desse alfabeto serão chamados *caracteres ASCII*. O alfabeto ASCII contém letras, dígitos decimais, sinais de pontuação, e alguns caracteres especiais. A lista dos 128 caracteres ASCII e seus números Unicode está registrada na [Tabela ASCII](#).

Infelizmente o alfabeto ASCII não é suficiente para escrever texto em português, pois não contém letras com [sinais diacríticos](#).

Esquemas de codificação

Um *esquema de codificação* (= [character encoding](#)) é uma tabela que associa uma sequência de bytes com cada número Unicode, e portanto com cada caractere Unicode. Em geral, omitimos “esquema” e dizemos apenas “codificação” ou “código”.

A sequência de bytes associada com um caractere é o *código* do caractere. Esse código representa o caractere na memória do computador e em arquivos digitais.

Código ASCII

O código ASCII é muito simples: o número Unicode de cada caractere é escrito em [notação binária](#). Esse código é usado apenas para o alfabeto ASCII. Como o alfabeto tem apenas 128 caracteres, o código ASCII necessita de apenas 1 byte por caractere e o primeiro bit desse byte é 0. Segue uma amostra da tabela de códigos:

número Unicode	caractere	código ASCII	hexadecimal
U+0021	!	00100001	0x21
U+0022	"	00100010	0x22
U+002D	-	00101101	0x2D
U+0039	9	00100111	0x39
U+0041	A	01000001	0x41
U+0042	B	01000010	0x42
U+0061	a	01100001	0x61
U+0062	b	01100010	0x62
U+007E	~	01111110	0x7E

A última coluna traz o código ASCII escrito em notação hexadecimal.

(Por que não aproveitar *todos* os 8 bits de um byte? Com isso, poderíamos codificar 128 caracteres adicionais além dos 128 do alfabeto ASCII. O [código ISO-LATIN-1](#) faz exatamente isso, mas caiu em desuso.)

Código UTF-8

O alfabeto Unicode tem mais de 1 milhão caracteres. Portanto, o código de cada caractere precisaria de pelo menos 3 bytes se usássemos notação binária. Usar um número fixo de bytes por caractere não seria eficiente, já que 1 byte é suficiente para codificar os caracteres mais comuns. A solução é recorrer a um código *multibyte*, que emprega um número *variável* de bytes por caractere: alguns caracteres usam 1 byte, outros usam 2 bytes, e assim por diante.

O código multibyte mais usado é conhecido como UTF-8. Ele associa uma sequência de 1 a 4 bytes (8 a 32 bits) com cada caractere Unicode. Os primeiros 128 caracteres usam o velho e bom código ASCII de 1 byte por caractere. Os demais caracteres têm um código mais complexo. Veja uma minúscula amostra:

número				
Unicode	caractere	código UTF-8		hexadecimal
U+0021	!	00100001		0x21
U+0022	"	00100010		0x22
U+002D	-	00101101		0x2D
U+0039	9	00100111		0x39
U+0041	A	01000001		0x41
U+0042	B	01000010		0x42
U+0061	a	01100001		0x61
U+0062	b	01100010		0x62
U+007E	~	01111110		0x7E
U+00C0	À	11000011 10000000		0xC380
U+00E3	ã	11000011 10100011		0xC3A3
U+00E7	ç	11000011 10100111		0xC3A7
U+00E9	é	11000011 10101001		0xC3A9
U+00FF	ÿ	11000011 10111111		0xC3BF
U+03A3	Σ	11001110 10100011		0xCEA3
U+03B1	α	11001110 10110001		0xCEB1
U+2014	—	11100010 10000000 10010100	0xE28094	
U+201C	“	11100010 10000000 10011100	0xE2809C	

(A última coluna traz o código UTF-8 escrito em notação hexadecimal.)

A lista dos códigos UTF-8 de todos os caracteres Unicode pode ser vista na página [UTF-8 encoding table and Unicode characters](#) ou na página [Unicode / Character reference](#) do Wikibooks. Por exemplo, a [cadeia de caracteres](#) “ação” é representada em UTF-8 pela seguinte sequência de bytes:

0x61	0xC3	0xA7	0xC3	0xA3	0x6F
a		ç		ã	o

Todas as letras com [sinais diacríticos](#) usadas em português são representados em UTF-8 por apenas 2 bytes, o primeiro dos quais é 0xC3 (195 em notação decimal).

Como o número de bytes por caractere não é fixo, a decodificação de uma sequência de bytes que representa um texto não é fácil. Como saber onde termina o código de um caractere e começa o código do caractere seguinte?

Estrutura do código UTF-8

O código UTF-8 representa cada caractere por uma sequência de 1 a 4 bytes. O esquema de codificação UTF-8 foi construído de modo que os primeiros bits do código de um caractere dizem quantos bytes o código ocupa. Assim, se o primeiro bit é 0, e portanto o valor do primeiro byte é menor que 128, então esse é o único byte do caractere. Se o valor do primeiro byte pertence ao intervalo 192 .. 223 então o código do caractere tem dois bytes. E assim por diante.

Veja as tabelas para entender melhor....

Os primeiros 128 caracteres da lista Unicode (números U+0000 a U+007F) são representados por 1 byte cada. Os 1920 caracteres seguintes (números U+0080 a U+07FF) são codificados em 2 bytes. E assim por diante.

A tabela abaixo mostra a estrutura do código UTF-8. Na coluna esquerda, temos os intervalos de números Unicode, em notação hexadecimal. Na coluna direita, em notação binária, os correspondentes valores válidos dos bytes do código:

números Unicode	byte 1	byte 2	byte 3	byte 4
00000000 .. 0000007F	0xxxxxxx			
00000080 .. 000007FF	110xxxxx	10xxxxxx		
00000800 .. 0000FFFF	1110xxxx	10xxxxxx	10xxxxxx	
00010000 .. 0010FFFF	11110xxx	10xxxxxx	10xxxxxx	10xxxxxx

Agora, a mesma tabela, com os números Unicode e os intervalos de valores dos bytes de código escritos em notação decimal:

0 .. 127	000..127			
128 .. 2047	192..223	128..191		
2048 .. 65535	224..239	128..191	128..191	
65536 .. 1114111	240..247	128..191	128..191	128..191

Finalmente, a mesma tabela, com os números Unicode e os intervalos de valores dos bytes escritos em notação hexadecimal:

0 .. 7F	00..7F			
80 .. 7FF	C0..DF	80..BF		
800 .. FFFF	E0..EF	80..BF	80..BF	
10000 .. 10FFFF	F0..F7	80..BF	80..BF	80..BF

(Este conteúdo foi obtido e adaptado de <https://www.ime.usp.br/~pf/algoritmos/index.html>)

Última atualização: segunda, 13 Set 2021, 19:16

<

Seguir para...

▾

Codificação em binário

>

< Caracteres e Codificação ASCII

Manter contato

<https://www.rj.senac.br>



Obter o aplicativo para dispositivos móveis