Le codage numérique du texte

D. Larrieu, P. Lucaud

Da ASCII code !

Le code ASCII est l'un des plus anciens codes utilisés pour représenter du texte en informatique. ASCII signifie (American Standard Code for Information Interchange).

Il se base sur un tableau contenant les caractères les plus utilisés **en langue anglaise** : les lettres de l'alphabet en majuscule (de A à Z) et en minuscule (de a à z), les dix chiffres arabes (de 0 à 9), les signes de ponctuation (point, virgule, point-virgule, guillemet, parenthèses, etc.), quelques symboles et certains caractères spéciaux invisibles (espace, retour-chariot, tabulation, retour-arrière, etc.).

Les créateurs de ce code ont limité le nombre de ses caractères à 128, c'est-à-dire 2^7, pour qu'ils puissent être codés avec seulement 7 bits : les ordinateurs utilisaient des cases mémoire de un octet, mais ils réservaient toujours le 8^e bit pour le contrôle de parité (c'est une sécurité pour éviter les erreurs, qui étaient très fréquentes dans les premières mémoires électroniques).

Exemple : Le caractère A est codé en ASCII par le nombre 65 (dans notre système décimal habituel), qui correspond en binaire au nombre 1000001.

Chaque caractère d'un texte codé en ASCII occupe alors un octet.
Un texte de 5000 caractères occupe donc 5 ko.

Code ASCII étendu

Longtemps après, les mémoires devenant plus fiables et de nouvelles méthodes plus sûres que le contrôle de parité ayant été inventées, le code ASCII ne suffisait plus. En effet, il avait l'inconvénient très gênant de ne contenir que les lettres non accentuées, ce qui pouvait suffire en anglais, mais pas dans les autres langues (comme le français et l'espagnol par exemple). Il manquait aussi des caractères qui pouvaient être utiles comme ceux servant à fabriquer des bordures de tableaux (plus jolies que celles formées de points d'exclamation et de traits d'union) etc.

C'est pourquoi le code ASCII fut étendu sur 8 bits, ce qui permit d'obtenir un tableau de 256 caractères (2^8) au lieu de 128. Il fut alors possible d'écrire des lettres comme é, è, ç, à, ù, ô, æ, œ, ñ, etc. Microsoft a adopté le code ANSI qui correspond au code ISO-Latin1.

Autres jeux de caractères

Le code *Unicode* est un système de codage des caractères sur 16 bits mis au point en 1991. Le système Unicode permet de représenter n'importe quel caractère par un code sur 16 bits, indépendamment de tout système d'exploitation ou langage de programmation. Il regroupe ainsi la quasi-totalité des alphabets existants (arabe, arménien, cyrillique, grec, hébreu, latin, ...) et est compatible avec le code ASCII.

***Activités* :** La phrase : « Enfin ! Je peux t'aider à comprendre ce qui s'est passé. » a une taille de 55 octets (il faut compter les espaces, l'apostrophe, le point final ...).

L'expérience peut être faite en utilisant un éditeur de texte simple comme le bloc-notes de Windows. Il suffit d'écrire le texte, puis de l'enregistrer sous format .txt et ensuite de vérifier la taille en octets du fichier obtenu (ce qui peut se faire en cliquant d'abord avec le bouton droit sur l'icône du fichier puis sur "Propriétés"). On peut écrire la même chose dans un éditeur de texte plus élaboré (Writer ou Word) et se rendre compte que la taille du fichier obtenu n'est pas du tout la même. Y a-t-il une explication ?

.....

Voici maintenant une phrase codée en binaire :

01000010 01110010 01100001 01110110 01101111 00101100 00100000 01110100 01110101 00100000 01100001 01110011 00100000 01110000 01110010 01100101 01110011 01110001 01110101 01100101 00100000 01110100 01101111 01110101 01110100 00100000 01110100 01110010 01101111 01110101 01110110 11101001 00101110 00101110 00101110

Saurais-tu la reconstituer à l'aide de la table ci-dessous?

.....

Que te manque-t'il ?.....

Voici un site sympathique : http://nickciske.com/tools/binary.php

En l'utilisant comme il faut tu trouveras à quelle lettre correspond le code qui n'est pas dans la table.

Quel est le code de la lettre à ?....., ù ?....., ô ?.....

Ecris une phrase très courte (une injonction) en français : .....

Code la en binaire : .....

.....

Décimale	Binaire	Valeur	Explication
000	00000000	NUL	NULL Character
001	00000001	SOH	Start of Header
002	00000010	STX	Start of Text
003	00000011	ETX	End of Text
004	00000100	EOT	End of Transmission
005	00000101	ENQ	Enquiry
006	00000110	ACK	Acknowledgement
007	00000111	BEL	Bell
008	00001000	BS	Backspace
009	00001001	HT	Horizontal Tab
010	00001010	LF	Line Feed
011	00001011	VT	Vertical Tab
012	00001100	FF	Form Feed
013	00001101	CR	Carriage Return
014	00001110	SO	Shift Out
015	00001111	SI	Shift In
016	00010000	DLE	Data Link Escape
017	00010001	DC1	Device Control 1 (XON)
018	00010010	DC2	Device Control 2
019	00010011	DC3	Device Control 3 (XOFF)
020	00010100	DC4	Device Control 4
021	00010101	NAK	Negative Acknowledgement
022	00010110	SYN	Synchronous Idle
023	00010111	ETB	End of Transmission Block
024	00011000	CAN	Cancel
025	00011001	EM	End of Medium
026	00011010	SUB	Substitude
027	00011011	ESC	Escape
028	00011100	FS	File Separator
029	00011101	GS	Group Separator
030	00011110	RS	Record Separator / Request to Send
031	00011111	US	Unit Separator
032	00100000	SP	Space
033	00100001	!	exclamation mark
034	00100010	"	Double quote
035	00100011	#	Number sign / hash sign
036	00100100	$	Dollar sign
037	00100101	%	Pourcent
038	00100110	&	Ampersand
039	00100111	'	Simple quote
040	00101000	(	Left parenthesis / Opening parenthesis
041	00101001	)	Right parenthesis / Closing parenthesis
042	00101010	*	Asterisk
043	00101011	+	Plus
044	00101100	,	Comma
045	00101101	-	Minus / Dash
046	00101110	.	Dot
047	00101111	/	Forward slash
048	00110000	0	
049	00110001	1	
050	00110010	2	
051	00110011	3	
052	00110100	4	
053	00110101	5	
054	00110110	6	
055	00110111	7	
056	00111000	8	
057	00111001	9	
058	00111010	:	Colon
059	00111011	;	Semi-colon
060	00111100	<	Less than
061	00111101	=	Equal sign
062	00111110	>	Greater than
063	00111111	?	Question mark
064	01000000	@	AT symbol
065	01000001	A	
066	01000010	B	
067	01000011	C	
068	01000100	D	
069	01000101	E	
070	01000110	F	
071	01000111	G	
072	01001000	H	
073	01001001	I	
074	01001010	J	
075	01001011	K	
076	01001100	L	
077	01001101	M	
078	01001110	N	
079	01001111	O	
080	01010000	P	
081	01010001	Q	
082	01010010	R	
083	01010011	S	
084	01010100	T	
085	01010101	U	
086	01010110	V	
087	01010111	W	
088	01011000	X	
089	01011001	Y	
090	01011010	Z	
091	01011011	[	Left bracket / Opening bracket
092	01011100	\	Back slash
093	01011101	]	Right bracket / Closing bracket
094	01011110	^	Caret / Circumflex
095	01011111	_	Underscore
096	01100000	`	Back quote
097	01100001	a	
098	01100010	b	
099	01100011	c	
100	01100100	d	
101	01100101	e	
102	01100110	f	
103	01100111	g	
104	01101000	h	
105	01101001	i	
106	01101010	j	
107	01101011	k	
108	01101100	l	
109	01101101	m	
110	01101110	n	
111	01101111	o	
112	01110000	p	
113	01110001	q	
114	01110010	r	
115	01110011	s	
116	01110100	t	
117	01110101	u	
118	01110110	v	
119	01110111	w	
120	01111000	x	
121	01111001	y	
122	01111010	z	
123	01111011	{	Left brace / Opening brace
124	01111100	|	Vertical bar
125	01111101	}	Right brace / Closing brace
126	01111110	~	Tilde
127	01111111	DEL	Delete

Code ASCII