



V Représentation d'un texte en machine

Nous avons appris à représenter les nombres, entiers ou flottants, en machine. Nous allons maintenant nous intéresser à la représentation des textes.

Contrairement au nombre, on est ici confronté au problème de l'harmonisation puisque il existe beaucoup d'alphabets différents, chaque langue ayant ses propres caractéristiques.

1) Le code ASCII

Le code ASCII a été conçu pour des textes écrits en anglais et permet de représenter 128 caractères, donc des mots de 7 bits. En fait, chaque caractère est codé sur un octet, le premier bit étant 0

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]

L'inconvénient majeur ici est l'absence de codage possible pour les lettres accentuées. C'est pour cela qu'a été créée une extension à ce code, s'adaptant aux particularités des langues de l'Europe occidentale, même si le 'œ' a été oublié !!!!

122 z	123 {	124	125 }	126 ~	127 □	128 €	129 □	130 ,
131 f	132 „	133 ...	134 †	135 ‡	136 ^	137 %	138 Š	139 <
140 €	141 □	142 Ž	143 □	144 □	145 \	146 /	147 \"	148 "
149 •	150 –	151 –	152 ~	153 ™	154 š	155 >	156 œ	157 □
158 ž	159 Ÿ	160	161 ¡	162 ¢	163 £	164 ¨	165 ¥	166 ¡
167 \$	168 ¨	169 ©	170 º	171 «	172 ¬	173	174 ®	175 ¯
176 °	177 ±	178 º	179 º	180 ´	181 µ	182 ¶	183 ·	184 ,
185 ¹	186 °	187 »	188 ¼	189 ½	190 ¾	191 ÷	192 À	193 Á
194 Â	195 Ã	196 Ä	197 Å	198 Æ	199 Ç	200 È	201 É	202 Ê
203 Ë	204 Ì	205 Í	206 Î	207 Ï	208 Ð	209 Ñ	210 Ò	211 Ó
212 Ô	213 Õ	214 Ö	215 ×	216 Ø	217 Ù	218 Ú	219 Û	220 Ü
221 Ý	222 Þ	223 ß	224 à	225 á	226 â	227 ã	228 ä	229 å
230 æ	231 ç	232 è	233 é	234 ê	235 ë	236 ì	237 í	238 î
239 ï	240 ð	241 ñ	242 ò	243 ó	244 ô	245 õ	246 ö	247 ÷
248 ø	249 ù	250 ú	251 û	252 ü	253 ý	254 þ	255 ÿ	

On obtient ainsi une table quasi complète qui correspond à la norme **ISO-8859-1**, aussi appelée *latin-1*

2)Unicode

Cependant le problème de la concordance entre différentes langues n'était pas réglé. Avec le développement des communications mondiales entre machine, il fallut trouver un codage capable de faire coexister des textes européens, chinois, russe...

Un format universel a émergé, Unicode qui contient près de 110000 caractères. Il faut alors 32 bits pour coder un caractère !

Cependant, Unicode est juste une table de codage et ne s'occupe pas de la façon dont la machine encode les caractères. La longueur d'encodage est variable suivant les normes utilisées. Ainsi avec la norme **UTF-8**, les caractères les plus courants sont codés sur un octet, de façon compatible avec le code ASCII.

Voici la représentation binaire suivant le nombre d'octets utilisés

Nbre octets codant	Format de la représentation binaire
1	0xxxxxxx
2	110xxxxx 10xxxxxx
3	1110xxxx 10xxxxxx 10xxxxxx
4	11110xxx 10xxxxxx 10xxxxxx 10xxxxxx

Si l'encodage UTF 8 est compatible avec la norme ASCII, il ne l'est pas avec la norme ISO-8859-1



3) Exercices

1) Coder votre prénom (en commençant avec une majuscule) à l'aide du code ASCII.

Quels sont selon vous les avantages et les inconvénients de ce codage ?

2) Trouvez le texte représenté en ASCII binaire par la suite de bits

010011100010111001010011001011100100100100101110

3) Ecrire un programme en langage python qui permet de coder un mot par la méthode de Vigenère.

https://fr.wikipedia.org/wiki/Chiffre_de_Vigenère

4) Ouvrez le site <https://apps.timwhitlock.info/unicode/inspect> et entrez cette phrase

Maître Corbeau, sur un 🌳 perché,

Précisez pour chaque caractère sur combien d'octet il est codé et donnez la représentation binaire du mot Maître.