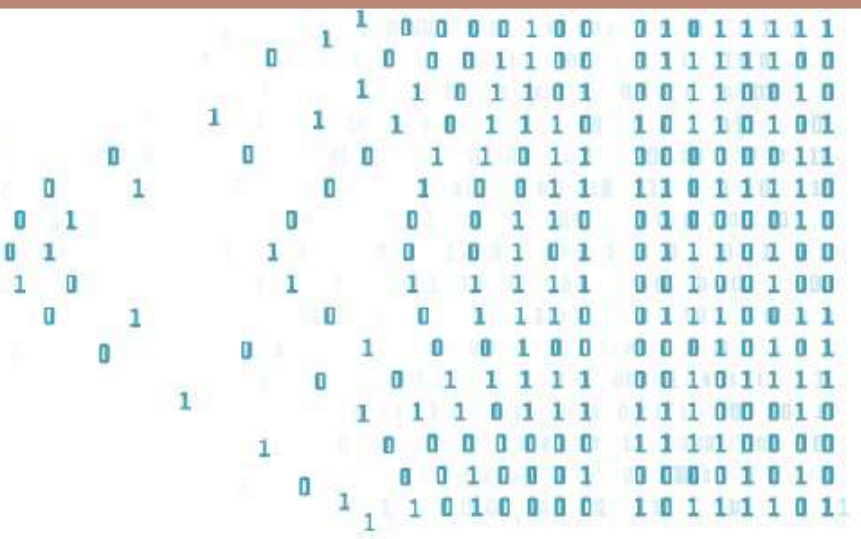
NSI | 04



REPRÉSENTATION D´UN TEXTE

[Marion SZPIEG]

Connaître différentes méthodes d´encodage des caractères

1. Introduction

Lorsqu'une machine affiche du texte à partir d'un document texte (word par exemple), voici les 3 étapes qui vous permettent de voir le texte à l'écran :

Remarque : il peut y avoir des soucis d'affichage à cause de

Zoom sur la 2e étape : l'encodage est une correspondance entre des paquets de nombres binaires et des caractères. Pour qu'un encodage soit bon, il faut qu'il vérifie les points suivants :

2. Un premier encodage très utilisé : ASCII

Dans les années 50, il commence à y avoir beaucoup de modèles de machines différentes, dans des marques différentes. Lors de leur conception, il n'y a pas eu d'harmonisation entre les différents constructeurs sur la façon d'encoder le texte. Il y a donc beaucoup d'encodages différents, tous incompatibles les uns avec les autres. Il était donc possible d'avoir un ordinateur relié à une imprimante, mais avec impossibilité d'imprimer puisque les 2 encodages n'étaient pas compatibles. Il apparaît rapidement le besoin d'une harmonisation globale entre les différents constructeurs.

Au début des années 60, la norme de codage de caractères ASCII (*American Statdard Code for Information Interchange*) est mise en place par l'ANSI (*American National Standards Institute*). Cet encodage, sur un octet, permet de représenter 128 caractères (voir le tableau).

On y trouve :

Déc	Hex	Oct	Binaire	Caractère	Déc	Hex	Oct	Binaire	Caractère	Déc	Hex	Oct	Binaire	Caractère	Déc	Hex	Oct	Binaire	Carac
0	0	0	0	NUL	32	20	40	100000	space	64	40	100	1000000	@	96	60	140	1100000	`
1	1	1	1	SOH	33	21	41	100001	!	65	41	101	1000001	A	97	61	141	1100001	a
2	2	2	10	STX	34	22	42	100010	"	66	42	102	1000010	B	98	62	142	1100010	b
3	3	3	11	ETX	35	23	43	100011	#	67	43	103	1000011	C	99	63	143	1100011	c
4	4	4	100	EOT	36	24	44	100100	$	68	44	104	1000100	D	100	64	144	1100100	d
5	5	5	101	ENQ	37	25	45	100101	%	69	45	105	1000101	E	101	65	145	1100101	e
6	6	6	110	ACK	38	26	46	100110	&	70	46	106	1000110	F	102	66	146	1100110	f
7	7	7	111	BEL	39	27	47	100111	'	71	47	107	1000111	G	103	67	147	1100111	g
8	8	10	1000	BS	40	28	50	101000	(	72	48	110	1001000	H	104	68	150	1101000	h
9	9	11	1001	HT	41	29	51	101001	)	73	49	111	1001001	I	105	69	151	1101001	i
10	0A	12	1010	LF	42	2A	52	101010	*	74	4A	112	1001010	J	106	6A	152	1101010	j
11	0B	13	1011	VT	43	2B	53	101011	+	75	4B	113	1001011	K	107	6B	153	1101011	k
12	0C	14	1100	FF	44	2C	54	101100	,	76	4C	114	1001100	L	108	6C	154	1101100	l
13	0D	15	1101	CR	45	2D	55	101101	-	77	4D	115	1001101	M	109	6D	155	1101101	m
14	0E	16	1110	SO	46	2E	56	101110	.	78	4E	116	1001110	N	110	6E	156	1101110	n
15	0F	17	1111	SI	47	2F	57	101111	/	79	4F	117	1001111	O	111	6F	157	1101111	o
16	10	20	10000	DLE	48	30	60	110000	0	80	50	120	1010000	P	112	70	160	1110000	p
17	11	21	10001	DC1	49	31	61	110001	1	81	51	121	1010001	Q	113	71	161	1110001	q
18	12	22	10010	DC2	50	32	62	110010	2	82	52	122	1010010	R	114	72	162	1110010	r
19	13	23	10011	DC3	51	33	63	110011	3	83	53	123	1010011	S	115	73	163	1110011	s
20	14	24	10100	DC4	52	34	64	110100	4	84	54	124	1010100	T	116	74	164	1110100	t
21	15	25	10101	NAK	53	35	65	110101	5	85	55	125	1010101	U	117	75	165	1110101	u
22	16	26	10110	SYN	54	36	66	110110	6	86	56	126	1010110	V	118	76	166	1110110	v
23	17	27	10111	ETB	55	37	67	110111	7	87	57	127	1010111	W	119	77	167	1110111	w
24	18	30	11000	CAN	56	38	70	111000	8	88	58	130	1011000	X	120	78	170	1111000	x
25	19	31	11001	EM	57	39	71	111001	9	89	59	131	1011001	Y	121	79	171	1111001	y
26	1A	32	11010	SUB	58	3A	72	111010	:	90	5A	132	1011010	Z	122	7A	172	1111010	z
27	1B	33	11011	ESC	59	3B	73	111011	;	91	5B	133	1011011	[	123	7B	173	1111011	{
28	1C	34	11100	FS	60	3C	74	111100	<	92	5C	134	1011100	\	124	7C	174	1111100	|
29	1D	35	11101	GS	61	3D	75	111101	=	93	5D	135	1011101	]	125	7D	175	1111101	}
30	1E	36	11110	RS	62	3E	76	111110	>	94	5E	136	1011110	^	126	7E	176	1111110	~
31	1F	37	11111	US	63	3F	77	111111	?	95	5F	137	1011111	_	127	7F	177	1111111	DEL

On remarque qu'il n'y a que 128 caractères alors que chacun d'entre eux est encodé sur un octet… On pourrait donc croire que le bit restant (celui de poids fort) ne sert à rien mais en fait non. Le bit de poids fort est le bit de parité. Lorsqu'on a un fichier texte, les bits de parité sont à 0 ou a 1 en fonction des caractères, mais on sait que la somme de ces bits de parité est paire (on l'appelle somme de contrôle). Ainsi, si la somme de contrôle des bits de parité est impaire, c'est qu'il y a eu un problème de transmission. Attention : si elle est paire, ça ne garantit pas qu'il n'y a pas d'erreur !!

Exemple : retrouver le message texte derrière ce code binaire, sachant que l'encodage utilisé est l'encodage ASCII :

01001100 11100001 01000000 11110110 11101001 11100101 00100001

3. Encodages ISO 8859

L'encodage ASCII ayant été inventé par un institut Américain, certains caractères comme les caractères accentués ainsi que les lettres de l'alphabet grec, cyrillique ou arabe ne sont pas représentables en machine, ce qui exclut une grande partie de la population mondiale !

Pour remédier à ce problème, dans les années 90, a été inventé un nouveau système d'encodage selon la norme ISO-8859 par l'organisation internationale ISO (International Organisation for Standardization). Les créateurs de cette nouvelle norme ont tenu à ce qu'un caractère ne soit encodé que sur un octet (sans bit de parité), ce qui laisse la possibilité d'encoder 256 caractères différents… ce qui est encore insuffisant ! Ils ont donc créé 16 tables différentes pour couvrir l'ensemble des symboles d'un grand nombre de langues.

Code ISO	Zone
8859-1 (latin-1)	Langue de l'Europe occidentale
8859-2 (latin-2)	Langues de l'Europe centrale ou de l'Est basées sur un alphabet romain
8859-3 (latin-3)	Langues de l'Europe du Sud (turc et maltais)
8859-4 (latin-4)	Langues de l'Europe du Nord (estonien, lituanien, groenlandais et sami)
8859-5	Langues basées sur l'alphabet cyrillique
8859-6	Langues basées sur l'alphabet arabe
8859-7	Langues basées sur l'alphabet grec
8859-8	Langues basées sur l'alphabet hébraïque
8859-9 (latin-5)	Langues turques et kurde (plus complet que latin-3 pour le turc)
8859-10 (latin-6)	Réarrangement du latin-4, mieux adapté aux langues nordiques
8859-11	Langue thaï
8859-12	Langues indiennes : devanagari (abandonné)
8859-13 (latin-7)	Langues baltes
8859-14 (latin-8)	Langues celtiques (irlandais, gaélique écossais, mannois (disparu) et breton)
8859-15 (latin-9)	Amélioration de 8859-15 abandonnant quelques symboles peu utilisés
8859-16 (latin-10)	Langues de l'Europe du Sud Est

Remarque : les 128 premiers caractères de chaque table sont les caractères de la table ASCII, les 128 suivants sont propres à la zone linguistique concernée.

On a donc caractères différents représentables.

4. Norme Unicode (encodages UTF-8, 16, 32)

Le système de l'encodage ISO 8859 permet de représenter un très grand nombres de caractères, mais il a deux inconvénients assez gênants :

Il a donc fallu une fois de plus faire évoluer l'encodage des caractères afin de remédier à ces problèmes. L'ISO a proposé un jeu universel de caractères appelé UCS (Universal Charactere Set) permettant de recenser tous les caractères existants dans toutes les langues humaines. L'organisation (privée à but non lucratif) consortium Unicode invente la norme Unicode, qui définit plusieurs techniques d'encodages appelés UTF-8, UTF-16 et UTF-32. UTF est un sigle signifiant Universal Transformation Format. Le nombre derrière représente le nombre de bits minimal pour encoder un caractère avec l'encodage en question.

Avec la norme Unicode, il y a un répertoire de 154 998 caractères permettant d'écrire dans plus de 150 langues ou dialectes.

Fonctionnement de l'encodage UTF-8 :

L'encodage UTF-8 utilise 1, 2, 3 ou 4 octets en respectant certaines règles :

- Un caractère en ASCII de base (appelé aussi US-ASCII) est codé sur un octet, dont le bit de poids fort est 0 :

Lettre	Valeur décimale	Codage ASCII	Codage UTF8
A			
Z			

- Pour les autres caractères, les bits de poids fort du premier octet forment une suite de 1 indiquant le nombre d'octets utilisés pour coder le caractère, puis d'un zéro. Les octets suivants commencent tous par 10 :

Représentation binaire UTF8	Signification
0xxxxxxx	1 octet codant de 1 à 7 bits
110xxxxx 10xxxxxx	2 octets codant de 8 à 11 bits
1110xxxx 10xxxxxx 10xxxxxx	3 octets codant de 12 à 16 bits
11110xxx 1011010 1011010 1011010	4 octets codant de 17 à 21 bits

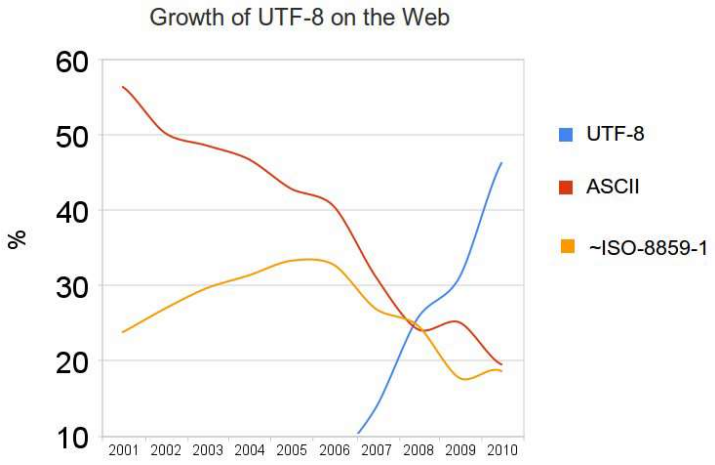
Exemples : les caractères é et € sont respectivement aux numéros 233 et 8364 en décimal. Retrouver leur encodage UTF-8.

Caractère é :

Caractère € :

5. Et aujourd’hui ?

Voici un graphe représentant l’évolution de l’utilisation de 3 encodages sur le Web entre 2000 et 2010 :



En 2016, le pourcentage d’utilisation de l’encodage UTF-8 sur le web était à 90 %, en 2019 il était de 93,1 % et en 2020 il était de 95,2 %.

Dans les deux extraits de code source de pages web ci-dessous, identifier, en les encerclant, dans quel encodage chacune s’affiche et grâce à quel balise/attribut.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"

<html xmlns="http://www.w3.org/1999/xhtml">

<head>
<meta http-equiv="Content-Type" content="text/html; utf-8" />
<title>Home</title>
<link rel="stylesheet" type="text/css" href="styl.css" />
```

```
<!DOCTYPE html>
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html;charset=iso-8859-15" />
    <title>iso-8859-15 Encoded Page</title>
  </head>
```