






Codage des informations

I Le système de comptage en base 2 ou système binaire :

Dans notre système décimal usuel, lequel est un système positionnel, on utilise dix symboles appelés chiffres, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 pour compter un nombre d'objets. La position de chacun de ces chiffres précise s'il s'agit de paquets de 10, de 10 x 10, de 10 x 10 x 10.

Dans le système binaire, on utilise le même principe mais en constituant des paquets de 2, de 2 x 2, de 2 x 2 x 2 et ainsi de suite selon le principe décrit ci-dessous :

Codage de l'information en binaire		
Principe de comptage en base 2	→	Usage de deux symboles seulement : 0 et 1
Quantité à compter	Nombre en binaire	Le même nombre écrit sur 4 bits
	1	0 0 0 1
	1 0	0 0 1 0
	1 1	0 0 1 1
	1 0 0	0 1 0 0
	1 1 1	0 1 1 1

On pourrait ainsi qualifier les paquets de 2 comme des « deuzaines », tout comme les paquets de 10 étaient des dizaines, ceux de 2 x 2 des « quatraines » (analogue des centaines), ceux de 2 x 2 x 2 des « huitaines » etc... Ce principe de comptage ne nécessite que l'emploi de deux symboles, les chiffres 0 et 1.

II Conversion d'écriture d'un système à l'autre et codage des entiers en informatique

Dans un ordinateur, tout nombre entier naturel est représenté dans le système binaire. Lorsqu'un utilisateur tape par exemple le nombre entier 87 sur sa calculatrice, ce dernier est converti en binaire pour être stocké sur un certain nombre de bits (par exemple 32 bits soit 4 octets)

Si nous voulons nous même opérer une telle conversion, on procède par divisions successives :

$$\begin{array}{l} 87 : 2 = 43 : 2 = 21 : 2 = 10 : 2 = 5 : 2 = 2 : 2 = 1 : 2 = 0 \\ \text{restes : } \quad 1 \quad 1 \quad 1 \quad 0 \quad 1 \quad 0 \quad 1 \end{array}$$

L'écriture en binaire s'obtient en écrivant la suite des restes à l'envers (en partant de la droite donc).
Le nombre devient ainsi : 1010111

Dans un ordinateur où les nombres entiers seraient codés sur 4 octets soit, 32 bits, ce nombre serait représenté avec des zéros additionnels ainsi :

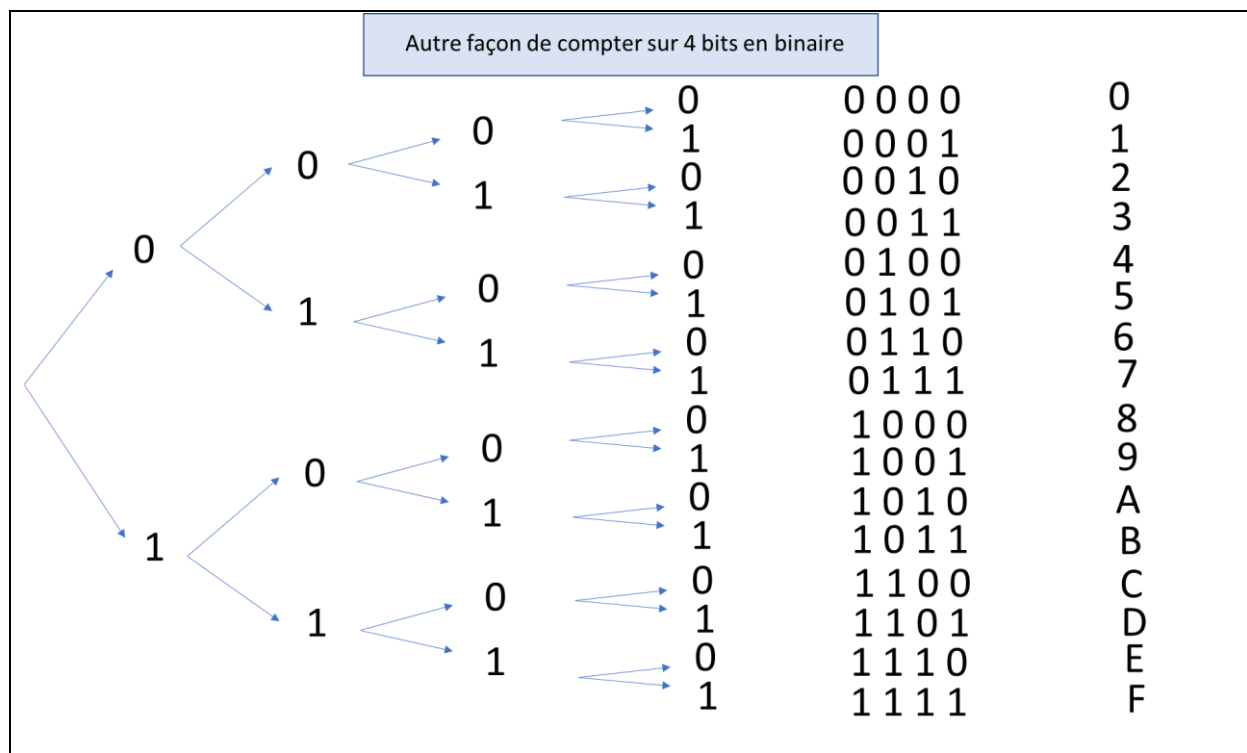
00000000 00000000 00000000 01010111

La conversion dans l'autre sens est plus facile. Par exemple, le nombre d'écriture binaire 1101, se convertit dans le système décimal ainsi :

$$1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 = 1 \times 8 + 1 \times 4 + 0 \times 2 + 1 \times 1 = 13$$

III Le codage d'un pixel

Dans une image de type bitmap (ou carte de bits) sont enregistrées trois intensités de couleurs pour trois couleurs dites primaires, rouge, vert et bleu. L'intensité lumineuse d'une couleur est représentée sur un octet, ce qui offre ainsi $2^8 = 256$ niveaux d'intensités différentes par couleur et donc $256 \times 256 \times 256$ soit environ 16 millions de couleurs. Pour plus de lisibilité, chaque octet est généralement affiché sous forme de deux blocs de 4 bits, chaque bloc pouvant représenter $2^4 = 16$ symboles différents, donc en particulier, les 16 chiffres d'un système de base 16 qualifié d'hexadécimal et dans lequel une dizaine est représentée par le symbole A. Voici les correspondances :



Par exemple : un pixel, dont le code couleur hexadécimal R V B est A1 B8 65 est représenté en mémoire informatique par le code binaire suivant :

10100001 10111000 01100101

lequel se traduit par des niveaux d'intensités (parmi les 256 possibles numérotés de 0 à 255) obtenu en convertissant ces nombres binaires dans le système décimal, ce qui donne pour le premier (rouge) :

$$10 \times 2^4 + 1 = 161$$

pour le second (vert) :

$$11 \times 2^4 + 8 = 184$$

et pour le troisième (bleu) :

$$6 \times 2^4 + 5 = 101$$

Le trois niveaux d'intensités sont alors :

$$161 \ 184 \ 101$$

Si on veut coder un pixel, on pourra donc au choix (dans les boites de dialogue permettant de changer les couleurs) , donner son code hexadécimal, ou ses trois intensités lumineuses comprises entre 0 et 255, ce qui donne, pour quelques couleurs remarquables :

	Code hexadécimal	Niveaux d'intensité
Blanc	FF FF FF	255 255 255
Gris clair	D0 D0 D0	208 208 208
Gris foncé	25 25 25	37 37 37
Noir	00 00 00	0 0 0
Rouge vif	FF 00 00	255 0 0
Magenta	FF 00 FF	255 0 255
Jaune	FF FF 00	255 255 0

IV Le codage d'une image : avec et sans compression

Une image (inscrite dans un rectangle) qui ne subit pas de compression est dans un format qualifié de bitmap ou carte de bits.

Son codage se résume à enregistrer le nombre de pixels dans sa longueur, le nombre de pixels dans sa largeur, puis les codes RVB de chacun de ses pixels. Ainsi, pour l'image qui suit (Aie, une épine sortie du petit doigt de l'auteur, qui je rassure les lecteurs, va bien) ouverte dans le logiciel Paint du menu Accessoires sous Windows, on peut lire :

$$\text{Longueur} = 3264 \text{ px (pixels)}, \quad \text{Largeur} = 2448 \text{ px}$$

Sachant qu'il faut 3 octets pour coder les trois couleurs RVB d'un pixel, on peut en déduire sa taille en pixels par la formule :

$\text{Taille en octets de l'image} = \text{Longueur} \times \text{Largeur} \times \text{Nombre d'octets par pixel}$
--

Et pour l'avoir en Mégaoctets (Mo), il faut diviser ce nombre par la valeur d'un Mo qui est 1 048 567, ce qui donne :

$$\text{Taille} = 3264 \times 2448 \times 3 : 1\,048\,567 = 22,9 \text{ Mo}$$



Cette image occupe beaucoup de place en mémoire. Une façon de réduire cette place est de la coder avec moins de couleurs, par exemple en utilisant seulement 1 octet pour chaque pixel, ce qui offre seulement 256 couleurs différentes possibles. La taille devient alors :

$$Taille = 3264 \times 2448 \times 1 : 1\,048\,567 = 7,6\ Mo$$

On constate que le rendu est légèrement affecté comme le montre l'image suivante, obtenue sous Paint en enregistrant la précédente au format 256 couleurs :



En revanche si on choisit d'enregistrer dans le format 16 couleurs, soit un demi octet par pixel, la taille est encore plus réduite mais le rendu très affecté :

$$Taille = 3264 \times 2448 \times 0,5 : 1\,048\,567 = 3,8 \text{ Mo}$$



Enfin, si on enregistre l'image initiale en format noir et blanc (dit monochrome), soit 1 bit donc 1/8 octet par pixel la taille est considérablement réduite et le rendu considérablement affecté :

$$Taille = 3264 \times 2448 \times 1 : 8 : 1024 = 975,4 Ko \approx 0,1 Mo$$



Quand on traite des photos, de paysage ou de portrait par exemple, le souci du détail et du contraste fait qu'on rechignera à perdre de la qualité. Il faut alors recourir à une technique de compression d'image qui n'altère pas de façon visible cette qualité.

Un appareil photo ou une caméra fournit à la prise de vue une image numérique bitmap où chaque pixel est codé sur 3 octets, offrant ainsi 16 777 216 couleurs différentes (256 x 256 x 256). La compression consiste à utiliser une méthode dite transformée de Fourier sur les courbes de variation d'intensité de chaque couleur pour chaque ligne de pixels. Voilà l'image obtenue après enregistrement dans le **format jpeg**, un des plus efficaces pour la photo :



On ne parvient pas à faire de différence avec le bitmap initial. Mais en regardant la taille de l'image fournie par Paint (en bas de l'image) on trouve 3,2 Mo, ce qui est un gain considérable, que l'on peut caractériser par un taux de compression défini par :

$$\begin{aligned} \text{Taux de compression} &= \frac{\text{Nombre d'octets supprimés}}{\text{Nombre d'octets initial}} \\ &= \frac{\text{Nombre d'octets initial} - \text{Nombre d'octets final}}{\text{Nombre d'octets initial}} \end{aligned}$$

Ce qui donne dans notre cas :

$$\text{Taux de compression} = \frac{22,9 - 3,2}{22,9} = 0,86 = 86 \%$$

A noter qu'il existe un autre format d'image compressé, le format **gif**, lequel pour notre image donne une taille légèrement supérieure de 3,5 Mo.

V Le codage de texte

Un fichier texte est un fichier constitué d'une chaîne de caractères. Chaque caractère est codé sur 1 octet soit 8 bits. On peut donc coder $2^8 = 256$ caractères différents qui comprennent les 26 lettres de l'alphabet, les chiffres de 0 à 10 mais aussi d'autres caractères, tous ces caractères étant qualifiés de **caractères ASCII**.

Voici la liste de ces caractères et le code (compris entre 0 et 255) qui leur est attribué :

ASCII	Caractère	ASCII	Caractère	ASCII	Caractère	ASCII	Caractère	ASCII	Caractère	ASCII	Caractère	ASCII	Caractère	ASCII	Caractère
0	NUL	32	Space	64	@	96	`	128	€	160		192	À	224	à
1	SOH	33	!	65	A	97	a	129	Ⓜ	161	ı	193	Á	225	á
2	STX	34	"	66	B	98	b	130	,	162	ƒ	194	Â	226	â
3	ETX	35	#	67	C	99	c	131	f	163	€	195	Ã	227	ã
4	EOT	36	\$	68	D	100	d	132	»	164	π	196	Ä	228	ä
5	ENQ	37	%	69	E	101	e	133	...	165	¥	197	Å	229	å
6	ACK	38	&	70	F	102	f	134	†	166	ı	198	Æ	230	æ
7	BEL	39	'	71	G	103	g	135	‡	167	§	199	Ç	231	ç
8	BS	40	(72	H	104	h	136	ˆ	168	ˆ	200	È	232	è
9	HT	41)	73	I	105	i	137	§	169	©	201	É	233	é
10	LF	42	*	74	J	106	j	138	Š	170	±	202	Ê	234	ê
11	VT	43	+	75	K	107	k	139	<	171	«	203	Ë	235	ë
12	FF	44	,	76	L	108	l	140	œ	172	¬	204	Ï	236	ï
13	CR	45	-	77	M	109	m	141	Ⓜ	173	Ⓜ	205	Í	237	í
14	SO	46	.	78	N	110	n	142	Ž	174	®	206	Î	238	î
15	SI	47	/	79	O	111	o	143	Ⓜ	175	ˆ	207	Ï	239	ï
16	DLE	48	0	80	P	112	p	144	Ⓜ	176	°	208	Ï	240	ï
17	DC1	49	1	81	Q	113	q	145	‘	177	±	209	Ñ	241	ñ
18	DC2	50	2	82	R	114	r	146	,	178	²	210	Ò	242	ò
19	DC3	51	3	83	S	115	s	147	“	179	³	211	Ó	243	ó
20	DC4	52	4	84	T	116	t	148	”	180	´	212	Ô	244	ô
21	NAK	53	5	85	U	117	u	149	•	181	µ	213	Õ	245	õ
22	SYN	54	6	86	V	118	v	150	-	182	¶	214	Ö	246	ö
23	ETB	55	7	87	W	119	w	151	-	183	·	215	×	247	×
24	CAN	56	8	88	X	120	x	152	~	184	.	216	Ø	248	ø
25	EM	57	9	89	Y	121	y	153	™	185	ı	217	Ù	249	ù
26	SUB	58	:	90	Z	122	z	154	š	186	e	218	Ú	250	ú
27	ESC	59	;	91	[123	{	155	>	187	»	219	Û	251	û
28	FS	60	<	92	\	124		156	œ	188	¼	220	Ü	252	ü
29	GS	61	=	93]	125	}	157	Ⓜ	189	½	221	Ý	253	ý
30	RS	62	>	94	^	126	~	158	ž	190	¾	222	Þ	254	þ
31	US	63	?	95	_	127	DEL	159	ÿ	191	¿	223	ß	255	ÿ

Pour l’obtenir dans word, il suffit de maintenir la touche alt enfoncée et de taper le numéro du code sur quatre chiffres en ajoutant éventuellement des zéros. Par exemple, pour obtenir la lettre h, il faut faire alt 0104