

# Introduction à l'architecture des ordinateurs

CM1-3 : Codage de l'information

Mickaël Martin Nevot

V1.3.1



Cette œuvre de [Mickaël Martin Nevot](#) est mise à disposition selon les termes de la [licence Creative Commons Attribution - Pas d'Utilisation Commerciale - Partage à l'Identique 3.0 non transposé](#).

# Introduction à l'architecture des ordinateurs

- I. Présentation
- II. Architecture
- III. Réseaux
- IV. Codage de l'info.

# Architecture de référence

## Couches hautes

- Couche des applications
- Couche distribution

## Couche système/réseau

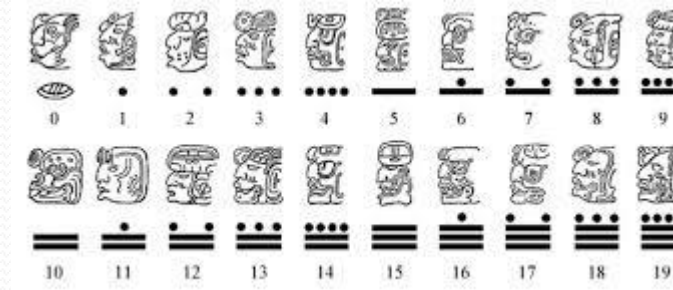
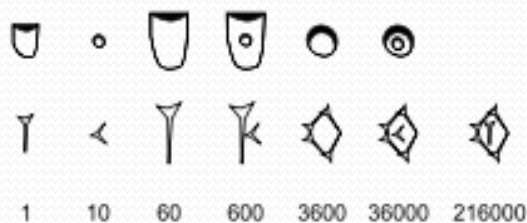
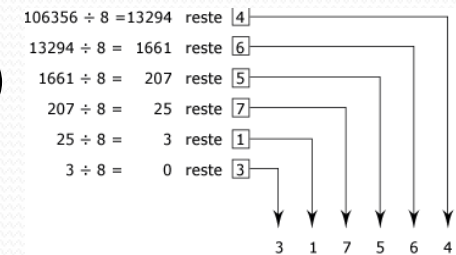
- Services additionnels
- Noyau

## Couches basses

- Couche du microcode et du langage machine
- Couche du matériel

# Rappel : bases

- Quinaire (5) : Mayas
- Octal (8) : informaticiens
- Duodécimale (12) : anglo-saxons av. 1960
- Vicésimale (20) : Mayas
- Sexagésimale (60) : Sumériens



# Rappel : bases

100111

→ 2<sup>0</sup> X 1 = 1

→ 2<sup>1</sup> X 1 = 2

→ 2<sup>2</sup> X 1 = 4

→ 2<sup>3</sup> X 0 = 0

→ 2<sup>4</sup> X 0 = 0

→ 2<sup>5</sup> X 1 = 32

±

Decimal ← 39

37 2

1 18 2

0 9 2

1 4 2

0 2 2

0 1

37 = 100101

Binaire

Binaire			Binaire		
Dec	Hex	8421	Dec	Hex	8421
0	0	0000	8	8	1000
1	1	0001	9	9	1001
2	2	0010	10	A	1010
3	3	0011	11	B	1011
4	4	0100	12	C	1100
5	5	0101	13	D	1101
6	6	0110	14	E	1110
7	7	0111	15	F	1111

Héxadécimal

# Bit et octet

- Octet : 8 bits

Représentation d'un OCTET

Numéros des "bits"

7	6	5	4	3	2	1	0
1	0	0	0	1	0	1	0

Valeurs en décimal

128	64	32	16	8	4	2	1
-----	----	----	----	---	---	---	---

138

décimal

10001010

binaire

byte ≈ octet

Préfixe	Symbole	Puissance	Multiplicateur
Yotta	Y	10 <sup>24</sup>	1.000.000.000.000.000.000.000.000
Zetta	Z	10 <sup>21</sup>	1.000.000.000.000.000.000.000.000
Exbi	Ei	2 <sup>60</sup>	1.152.921.504.606.846.976
Exa	E	10 <sup>18</sup>	1.000.000.000.000.000.000.000.000
Pebi	Pi	2 <sup>50</sup>	1.125.899.906.842.624
Peta	P	10 <sup>15</sup>	1.000.000.000.000.000.000.000.000
Tebi	Ti	2 <sup>40</sup>	1.099.511.627.776
Téra	T	10 <sup>12</sup>	1.000.000.000.000.000.000.000.000
Gibi	Gi	2 <sup>30</sup>	1.073.741.824
Giga	G	10 <sup>9</sup>	1.000.000.000.000.000.000.000.000
Mebi	Mi	2 <sup>20</sup>	1.048.576
Méga	M	10 <sup>6</sup>	1.000.000
Kibi	Ki	2 <sup>10</sup>	1.024
Kilo	k	10 <sup>3</sup>	1.000
Référence		10 <sup>0</sup>	1

Espace utilisé :	481 833 127 936 octets	448 Go
Espace libre :	493 551 214 592 octets	459 Go

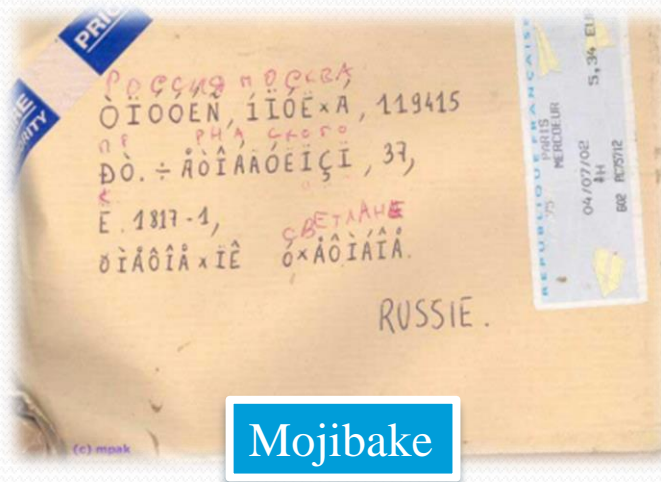
Erreur Windows (notamment) classique



# Codage des caractères

- ASCII (7 bits) →
- ISO-8859-1 / Latin-1 (8 bits)
- Unicode :
  - UTF-8 (1 à 4 octets)
  - UTF-16 (2 ou 4 octets)
  - UTF-32 (4 octets)

0 NUL	32 espace	64 @	96 `
1 SOH	33 !	65 A	97 a
2 STX	34 "	66 B	98 b
3 ETX	35 #	67 C	99 c
4 EOT	36 \$	68 D	100 d
5 ENQ	37 %	69 E	101 e
6 ACK	38 &	70 F	102 f
7 BEL	39 '	71 G	103 g
8 BS	40 (	72 H	104 h
9 HT	41 )	73 I	105 i
10 LF	42 *	74 J	106 j
11 VT	43 +	75 K	107 k
12 FF	44 ,	76 L	108 l
13 CR	45 -	77 M	109 m
14 SO	46 .	78 N	110 n
15 SI	47 /	79 O	111 o
16 SLE	48 0	80 P	112 p
17 GS1	49 1	81 Q	113 q
18 DC2	50 2	82 R	114 r
19 DC3	51 3	83 S	115 s
20 DC4	52 4	84 T	116 t
21 NAK	53 5	85 U	117 u
22 SYN	54 6	86 V	118 v
23 ETB	55 7	87 W	119 w
24 CAN	56 8	88 X	120 x
25 EM	57 9	89 Y	121 y
26 SIB	58 :	90 Z	122 z
27 ESC	59 ;	91 [	123 {
28 FS	60 <	92 \	124
29 GS	61 =	93 ]	125 }
30 RS	62 >	94 ^	126 ~
31 US	63 ?	95 _	127 ■



# Codage des images

Exemple : image  
1024 x 768

- Formats **matriciels**  
(composés de pixels) :

Abusivement appelé bitmap (format propriétaire)

- Monochrome (noir et blanc) :
  - Taille (exemple) : 786432 bits (96 Kio)
- Niveau de gris (8 bits)
  - Taille (exemple) : 770 Kio
- Couleurs indexées (8 bits)
  - Taille (exemple) : 770 Kio
- Couleurs vraies (24 bits)
  - Taille (exemple) : 2,25 Mio

- Formats **vectoriels** :

- Formes géométriques (équation mathématiques)

1	1	1	1	1	1	1	1	1	1
1	1	1	0	0	0	0	1	1	1
1	1	0	1	1	1	1	0	1	1
1	0	1	1	1	1	1	1	0	1
1	0	1	0	1	1	0	1	0	1
1	0	1	1	1	1	1	1	0	1
1	0	1	0	1	1	0	1	0	1
1	0	1	1	0	0	1	1	0	1
1	1	0	1	1	1	1	0	1	1
1	1	1	0	0	0	0	1	1	1
1	1	1	1	1	1	1	1	1	1





# Techniques de compression

- Sans perte (exemple RLE) : Documents, exécutable, etc.

30  $\Rightarrow$  14 =  
53% de compr.

- SSYYYYYYYYPHHHHHHHERRRRRRRRRRR  $\Rightarrow$  SS@9YP@7HE@10R

- Avec perte (exemple JPEG) : Images, sons, vidéos, etc.

- Image décomposée en carrés de 8 x 8 pixels
  - Suppression des fréquences auxquelles l'œil est peu sensible
  - Suppression des redondances spatiales
  - Compression RLE et Huffman

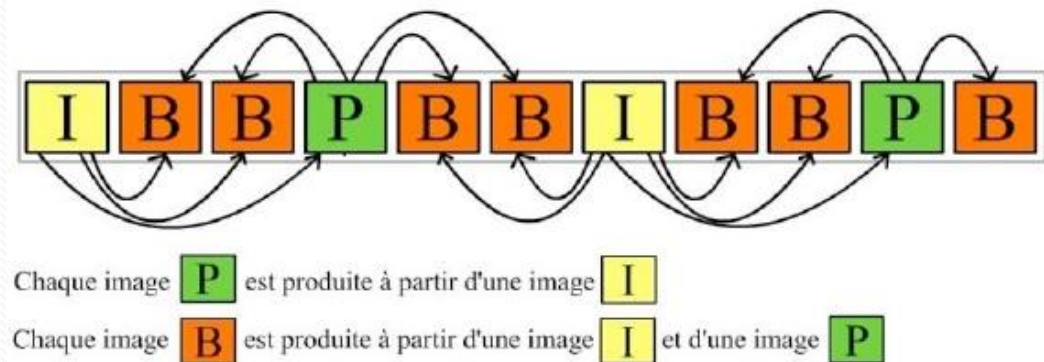
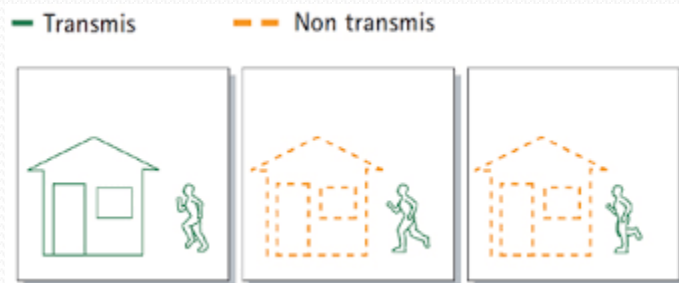
Paramètre : taux de  
compression (1% à 99%)



# Codage des vidéos

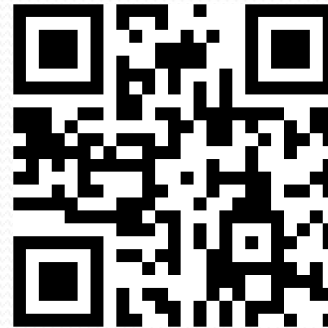
Exemple : vidéo de 15 minutes  
1024 x 768 couleurs vraies

- Nombre d'images par seconde (24, 25, 50, 60, 120, etc.)
- Sans compression : 22500 images de 2,25 Mio = 50,625 Gio
- M-JPEG : JPEG sur chaque image de la vidéo
- *Idem* avec suppression de redondances temporelles :
  - MPEG-1, MPEG-2, MPEG-4

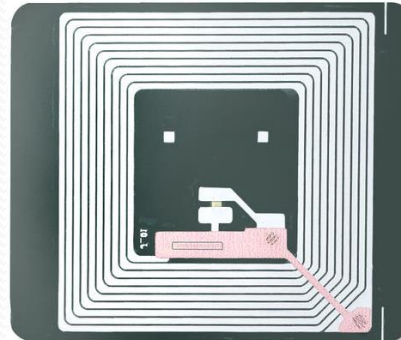


# Identification des objets

- Code barre
- Code QR
- RFID
  - Ondes radio
  - Visibilité inutile



Scanner un code QR c'est  
comme faire un clic IRL



# Crédits

## Auteur

Mickaël Martin Nevot

[mmartin.nevot@gmail.com](mailto:mmartin.nevot@gmail.com)



Carte de visite électronique

## Relecteurs

Cours en ligne sur : [www.mickael-martin-nevot.com](http://www.mickael-martin-nevot.com)

