

Le codage de l'information

C'est quoi l'informatique ?

- L'informatique désigne l'automatisation du traitement de l'information par un système, concret (machine) ou abstrait.
- Dans son acceptation courante, l'informatique désigne l'ensemble des sciences et techniques en rapport avec le traitement de l'information.
- Dans le parler populaire, l'informatique peut aussi désigner ce qui se rapporte au matériel informatique (l'électronique) ou la bureautique.

- **Internet, emails, WWW, ...**
- **Outils bureautiques**
(traitements de texte, tableurs, ...)
- **Utilisation de logiciels scientifiques**
(Bioinformatique, informatique biomédicale, etc.)
- **Programmation, base de données**
- **Développement de logiciels, gestion de projets informatiques**



Difficulté croissante

Un ordinateur est un ensemble de composants électroniques modulaires, c'est-à-dire des composants pouvant être remplacés par d'autres composants ayant éventuellement des caractéristiques différentes, capables de faire fonctionner des programmes informatiques. On parle ainsi de « **hardware** » pour désigner l'ensemble des éléments matériels de l'ordinateur et de « **software** » pour désigner la partie logicielle.

Chaque élément est associé à une fonction:

Ecran (« voir »), DD (« écrire et lire »), souris (« faire des choix »), RAM (« mémoire »), lecteurs (DVD, CD: « retrouver des données»...), processeur (« calculer »), clavier (« taper des mots »), ... etc ... (imprimante, carte mère, joystick, scanner, bus, modem....).

Notion clef en informatique : « **l'information** »

Récapitulation de la liste précédente à quelques fonctions essentielles autour de cette notion :

- **Echanger** de l'information (clavier, souris, bus, écran, imprimante, ...).
- **Mémoriser** de l'information (DD/SSD, RAM, ROM, DVD, CD, clé USB, ...).
- **Calculer** de l'information (processeur).

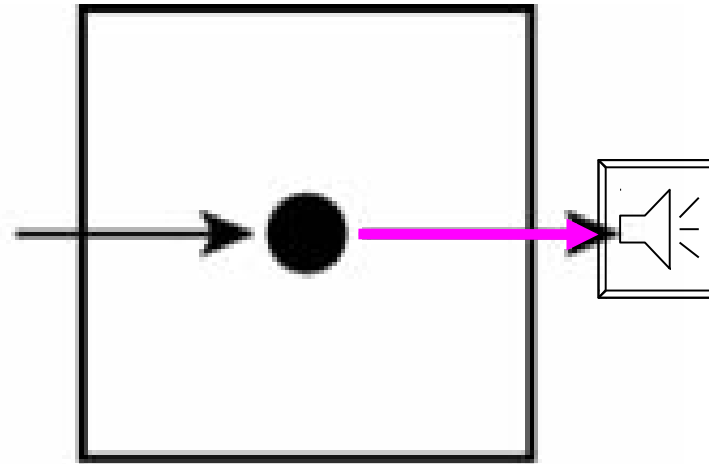
Construction de schémas simples modélisant :

- Une radio
- Une télévision/enregistreur
- Une calculatrice
- Un ordinateur

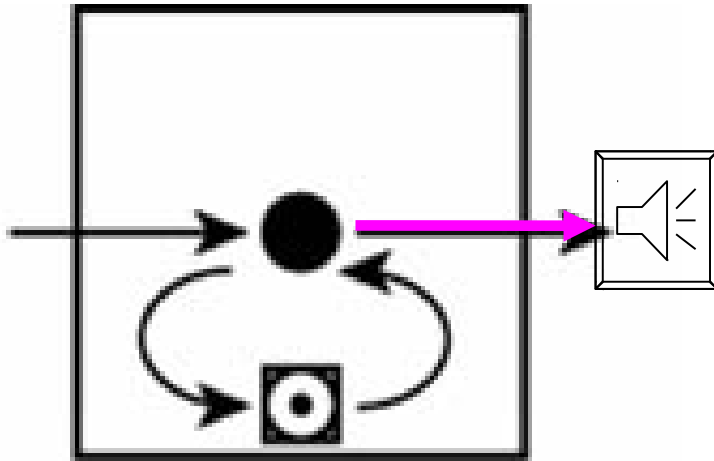
Définir des grandes notions :

- Le passage de l'information
- Unité de calcul
- Unité de stockage

- Information
- Unité de calcul
- Unité de stockage

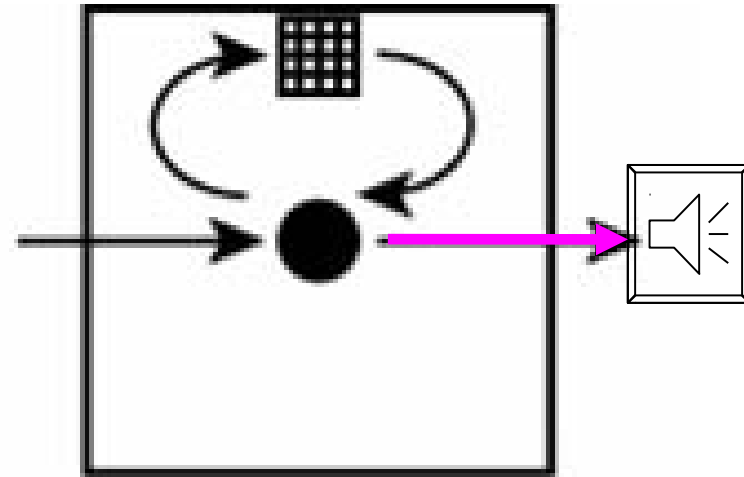


Radio



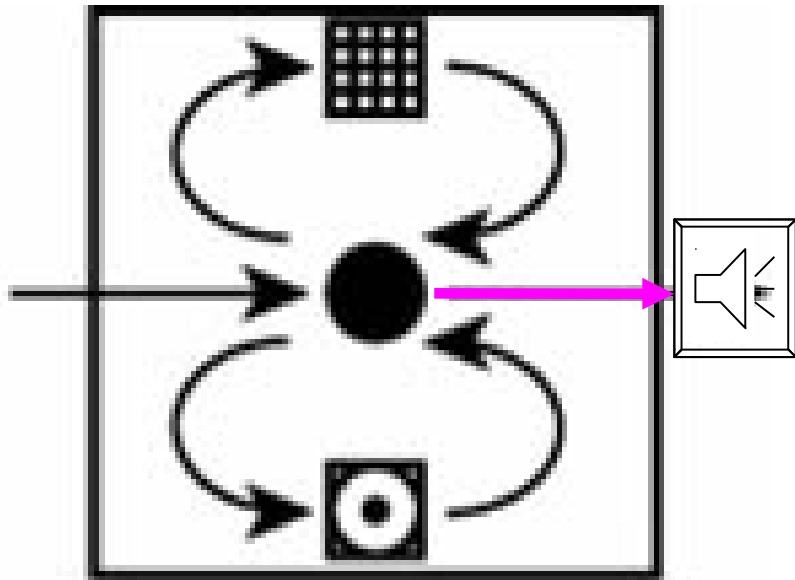
TV + enregistreur (DD box)

L'information arrive (antenne/cable), peut être retranscrite en temps réel (écran) ou enregistrée sur support et relue plus tard



Calculatrice (simple)

L'information arrive (clavier), le calcul s'effectue puis le résultat s'affiche (écran).



Ordinateur

L'information arrive (clavier, scanner, DVD, etc...), peut être affichée en temps réel (écran) ou mémorisée (DD, clé USB, etc...)

Des opérations peuvent être effectuées (processeur) puis le résultat stocké ou affiché (écran, imprimante, ...)

Les différentes mémoires

Quels types de mémoires trouve-t-on dans un ordinateur ?

Qu'est ce qui les différencie ?

Mémoire	Maintien	Rôle	Taille	Vitesse
Mémoires mortes (~ROM : read only memory)	Tout le temps (sans électricité)	PROM, EPROM : ex : lancer le système d'exploitation (= BIOS)	x ko - x Mo	Moyenne
Stockage magnétique	Tout le temps (sans électricité)	Stockage massif à long terme : disque dur, enregistreur à bandes	x Go - xx To	Lente (accès disque/bande)
Mémoires flash	Tout le temps (sans électricité)	Stockage à moyen terme : clé usb, disque SSD	x Go - x To	Moyenne
RAM (random acces memory)	Uniquement sous tension	Mémoire de travail des données et instructions en cours : ex. barrettes de mémoires	x-xx Go	Rapide
Cache	Uniquement sous tension	Idem RAM, mais optimisé : cache des CPUs	x ko - x Mo	Très rapide

Le processeur

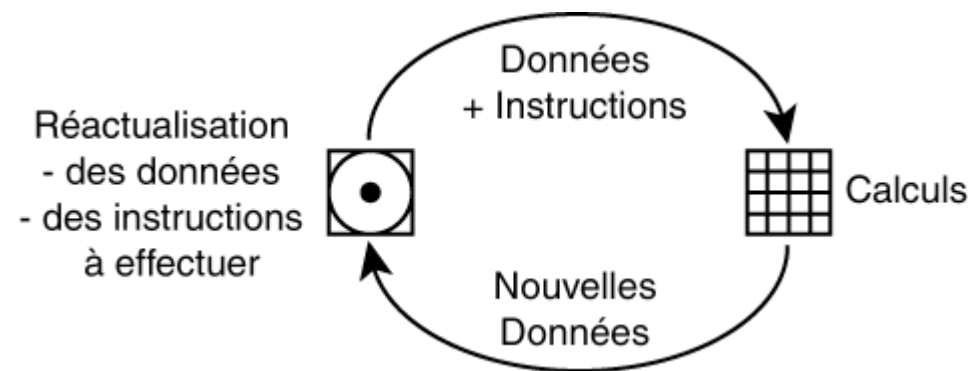
Quel est le rôle du processeur ?

(Intel, AMD, ARM, Cortex, etc...)

Calculer de nouvelles données (informations) avec des données (registres de données) et des instructions (registres d'instructions).

Les instructions commandent des portes logiques (des formes d'aiguillages) qui déterminent *comment* sont combinées les données entre elles.

Dialogue Processeur-Mémoire:



C'est le lien entre la machine et nous. Il existe différents SE (OS): Windows, Unix/Linux, MacOS, iOS, Android (=Unix), ...

Un système d'exploitation est un programme de base permettant de gérer :

- la mémoire et le processeur
- l'exécution des autres programmes
- la communication interne (le bus)
- les périphériques externes et donc le graphisme
- Accessibilité de l'information aux utilisateurs

Windows est à la fois un système d'exploitation et une interface graphique.

Le support de stockage (disque dur, clé usb, DVD, etc) d'un ordinateur conserve toutes les informations (textes, figures, films et programmes) dans des fichiers.

Afin de comprendre un peu mieux comment tout cela marche, quelques explications sont nécessaires.

Il est important de comprendre la différence entre :

- Le contenu d'un fichier
- La représentation d'un fichier

Le contenu d'un fichier est ce qui est vraiment écrit sur le disque

La représentation d'un fichier c'est son affichage à l'écran au moyen d'un utilitaire.

Le même fichier peut s'afficher différemment suivant l'utilitaire utilisé.

Les codages binaires et hexadécimaux.

Sur le disque un fichier est enregistré sous forme **binaire** : une série de zéros et de uns (001010101010101).

Comme cette notation est absolument impossible à lire pour nous, ce codage est généralement transformé en hexadécimal.

Dans la vie de tous les jours nous utilisons le système décimal, c'est à dire un codage avec dix chiffres de 0 à 9.

L'octet (=bytes)

Dans les ordinateurs, tout est stocké sous forme élémentaire dans des **bytes** (=octets en français) => un groupe de **huit chiffres** à la suite (soit zéro, soit un)

il y a 256 possibilités sur 1 octet : c'est à dire de 00000000 à (00000001, 00000010,...) à 11111111

L'hexadécimal

Pour pouvoir lire ces valeurs plus facilement on peut utiliser l'hexadécimal.

L'hexadécimal est un codage sur 16 caractères (caractères 0 à 9 et de A à F).

- Le nombre décimal 0 reste 0, ... 5 reste 5, ... 9 reste 9
- Le nombre décimal 10 devient donc A ...
- Le nombre décimal 15 devient F.

Donc 2 lettres hexadécimales sont équivalentes à 1 octet (car $16 * 16$ combinaisons = 256)

La visualisation d'un fichier

En hexadécimal

```
28 28 28 28 28 41 4A 36 32 37 39 30 39 20 2F 20
4D 61 72 69 6E 6F 6D 6F 6E 61 73 20 61 6C 6B 61
6C 69 70 68 69 6C 61 20 2F 20 55 31 3A 31 2E 30
2C 41 46 31 37 33 39 36 37 20 2F 20 4D 61 72 69
6E 6F 6D 6F 6E 61 73 20 63 6F 6D 6D 75 6E 69 73
20 2F 20 4B 54 30 39 32 33 3A 31 2E 30 29 3A 31
2E 30 2C 28 28 41 59 30 39 32 30 36 36 20 2F 20
4D 61 72 69 6E 6F 6D 6F 6E 61 73 20 70 72 6F 74
65 61 20 2F 20 31 35 34 3A 31 2E 30 2C 28 41 4A
32 33 38 35 39 37 20 2F 20 4D 61 72 69 6E 6F 6D
6F 6E 61 73 20 70 72 6F 74 65 61 20 2F 20 3A 31
2E 30 2C 28 41 42 30 37 34 31 39 34 20 2F 20 4D
61 72 69 6E 6F 6D 6F 6E 61 73 20 70 72 69 6D 6F
72 79 65 6E 73 69 73 20 2F 20 4B 4D 4D 20 33 36
33 34 3A 31 2E 30 2C 41 42 30 37 34 31 39 33 20
2F 20 4D 61 72 69 6E 6F 6D 6F 6E 61 73 20 70 72
69 6D 6F 72 79 65 6E 73 69 73 20 2F 20 4B 4D 4D
20 33 36 33 33 3A 31 2E 30 29 3A 31 2E 30 29 3A
30 2E 33 29 3A 31 2E 30 2C 4D 61 72 69 6E 6F 6D
6F 6E 61 73 20 70 6F 6E 74 69 69 20 2F 20 34 36
20 31 36 54 3A 31 2E 30 29 3A 31 2E 30 29 3A 31
2E 30 2C 28 41 42 30 30 36 37 37 30 20 2F 20 4D
61 72 69 6E 6F 73 70 69 72 69 6C 6C 75 6D 20 6D
65 67 61 74 65 72 69 75 6D 20 2F 20 48 37 54 3A
31 2E 30 2C 58 37 34 36 39 38 20 2F 20 56 69 62
72 69 6F 20 63 69 6E 63 69 6E 6E 61 74 69 65 6E
73 69 73 20 2F 20 41 54 43 43 20 33 35 39 31 32
54 3A 31 2E 30 29 3A 31 2E 30 29 3A 30 2E 35 2C
58 36 37 30 32 35 20 2F 20 4D 61 72 69 6E 6F 6D
6F 6E 61 73 20 76 61 67 61 20 2F 20 41 54 43 43
20 32 37 31 31 39 54 3A 31 2E 30 29 3A 31 2E 30
2C 41 46 30 36 33 30 32 37 20 2F 20 4D 61 72 69
6E 6F 6D 6F 6E 61 73 20 6D 65 64 69 74 65 72 72
61 6E 65 61 20 2F 20 41 54 43 43 20 37 30 30 34
39 32 54 3A 31 2E 30 29 3B 0D 0A
```

```
(((((AJ627909 /
Marinomonas alka
liphila / U1:1.0
,AF173967 / Mari
nomonas communis
/ KTO923:1.0):1
.O,((AYO92066 /
Marinomonas prot
ea / 154:1.0,(AJ
238597 / Marinom
onas protea / :1
.O,(ABO74194 / M
arinomonas primo
ryensis / KMM 36
34:1.0,ABO74193
/ Marinomonas pr
imoryensis / KMM
3633:1.0):1.0):
0.3):1.0,Marinom
onas pontii / 46
16T:1.0):1.0):1
.O,(AB006770 / M
arinospirillum m
egaterium / H7T:
1.0,X74698 / Vib
rio cincinnatien
sis / ATCC 35912
T:1.0):1.0):0.5,
X67025 / Marinom
onas vaga / ATCC
27119T:1.0):1.0
,AF063027 / Mari
nomonas mediterr
anea / ATCC 7004
92T:1.0);..
```

Conclusion :

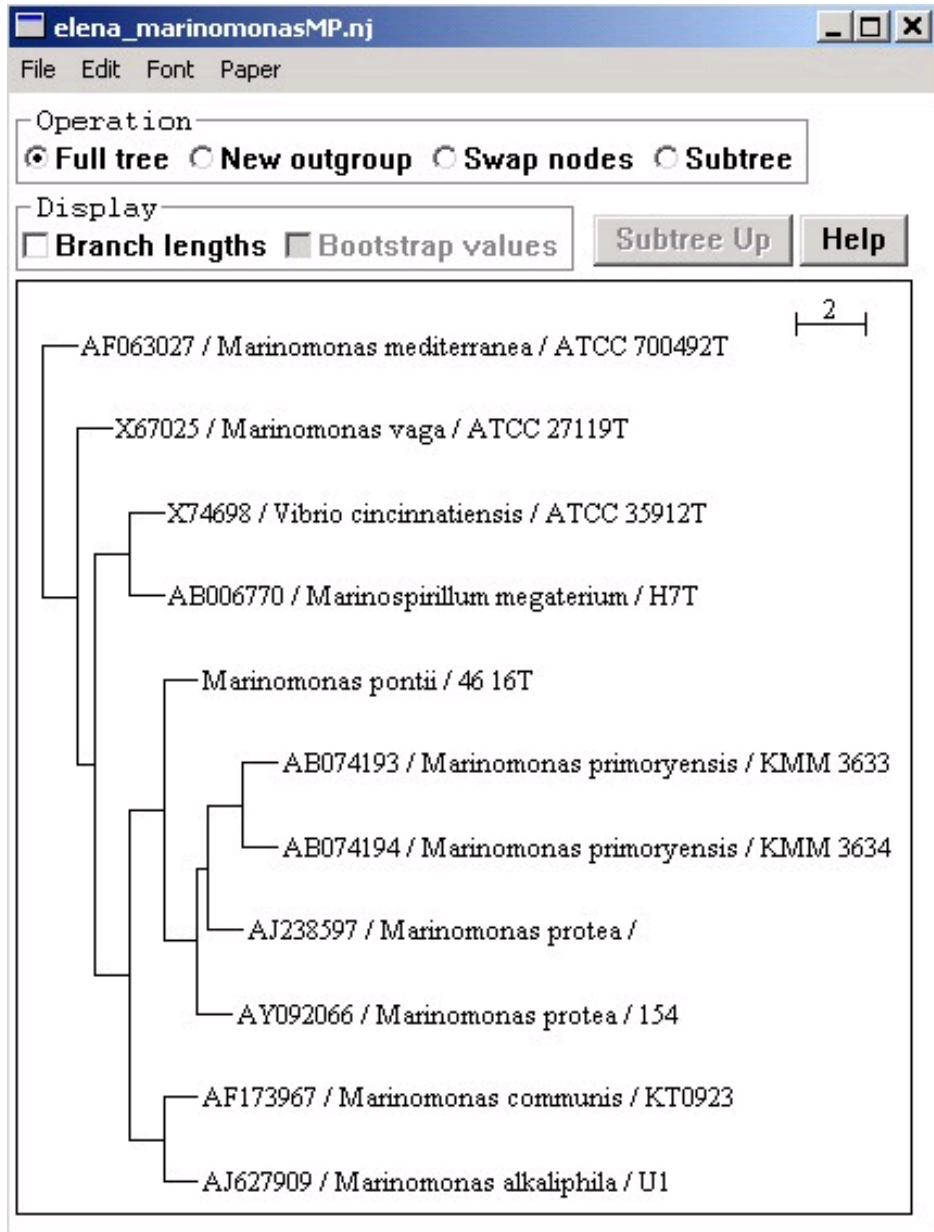
Un fichier incompréhensible

Le même fichier ouvert avec le text pad sous Windows:

```
(((((AJ627909 / Marinomonas alkaliphila / U1:1.0,AF173967 /  
Marinomonas communis / KT0923:1.0):1.0,((AY092066 / Marinomonas protea  
/ 154:1.0,(AJ238597 / Marinomonas protea / :1.0,(AB074194 / Marinomonas  
primoryensis / KMM 3634:1.0,AB074193 / Marinomonas primoryensis / KMM  
3633:1.0):1.0):0.3):1.0,Marinomonas pontii / 46  
16T:1.0):1.0):1.0,(AB006770 / Marinospirillum megaterium /  
H7T:1.0,X74698 / Vibrio cincinnatiensis / ATCC  
35912T:1.0):1.0):0.5,X67025 / Marinomonas vaga / ATCC  
27119T:1.0):1.0,AF063027 / Marinomonas mediterranea / ATCC 700492T:1.0);
```

Un fichier moins incompréhensible !

Finalement le même ouvert avec une application *ad hoc* :



Cette fois ci cela paraît plus compréhensible !

Il s'agit tout simplement d'un arbre qui représente les parentés entre certaines espèces de bactéries

Les flèches mettent en évidence des endroits où se situent la présence d'ancêtres communs à certaines espèces.

Il s'agit bel et bien du même fichier, mais ouvert avec des logiciels différents.

Ce qu'il faut retenir :

- Un fichier enregistré sur le disque n'est qu'une suite de 0 et 1.
- C'est le logiciel utilisé pour ouvrir ce fichier qui va le "traduire" afin de l'afficher sous un certain format.

L'extension du fichier (.txt, .doc, .pdf) n'est qu'une indication qui permet d'associer un utilitaire (par défaut) pour ouvrir ce fichier.

Si vous cliquez sur un fichier .txt, windows associe par défaut le texpad pour l'ouvrir.

Si vous cliquez sur un fichier .doc, Windows associe par défaut Word (Office) pour l'ouvrir.

- Vers la fin des années 30, Claude Shannon démontra qu'à l'aide de «contacteurs» (interrupteurs) fermés pour « vrai » et ouverts pour « faux » il était possible d'effectuer des opérations logiques en associant le nombre 1 pour « vrai » et 0 pour « faux ».
- Ce codage de l'information est nommé base binaire. C'est avec ce codage que fonctionnent les ordinateurs. Il consiste à utiliser deux états (représentés par les chiffres 0 et 1) pour coder les informations.

L'homme calcule depuis 2000 ans avant Jésus-Christ avec 10 chiffres (0, 1, 2, 3, 4, 5, 6, 7, 8, 9), on parle alors de base décimale (ou base 10). Toutefois dans des civilisations plus anciennes ou pour certaines applications actuelles d'autres bases de calcul ont et sont toujours utilisées :

- base sexagésimale (60), utilisée par les Sumériens. Cette base est également utilisée dans le système horaire actuel, pour les minutes et les secondes ;
- base vicésimale (20), utilisée par les Mayas ;
- base duodécimale (12), utilisée par les anglo-saxons dans leur système monétaire jusqu'en 1960 : un « pound » représentait vingt « shilling » et un « shilling » représentait douze « pences ». Le système d'heure actuel fonctionne également sur douze heures (notamment dans la notation anglo-saxonne) ;
- base quinaire (5), utilisée par les Mayas ;
- base binaire (2), utilisée par l'ensemble des technologies numériques.

Le terme bit (b avec une minuscule dans les notations) signifie « binary digit », c'est-à-dire 0 ou 1 en numérotation binaire (d'où : “digital” => numérique...).

Il s'agit de la plus petite unité d'information manipulable par une machine numérique. Il est possible de représenter physiquement cette information binaire par un signal électrique ou magnétique, qui, au-delà d'un certain seuil, correspond à la valeur 1

Avec un bit il est ainsi possible d'obtenir deux états : soit 1, soit 0. Grâce à 2 bits, il est possible d'obtenir quatre états différents (2×2) :

0	0
0	1
1	0
1	1

Avec 3 bits, il est possible d'obtenir huit états différents ($2 \times 2 \times 2$) :

Valeur binaire sur 3 bits	Valeur décimale
---------------------------	-----------------

000	0
001	1
010	2
011	3
100	4
101	5
110	6
111	7

Pour un groupe de n bits, il est possible de représenter 2^n valeurs.

Poids des bits

Dans un nombre binaire, la valeur d'un bit, appelée poids, dépend de la position du bit en partant de la droite. A la manière des dizaines, des centaines et des milliers pour un nombre décimal, le poids d'un bit croît d'une puissance de deux en allant de la droite vers la gauche comme le montre le tableau suivant :

Nombre binaire	1	1	1	1	1	1	1	1
Poids	$2^7 = 128$	$2^6 = 64$	$2^5 = 32$	$2^4 = 16$	$2^3 = 8$	$2^2 = 4$	$2^1 = 2$	$2^0 = 1$

<= 8 bits (= 1 octet) à "1"

<= $128+64+32+16+8+4+2+1$
= 255 en décimal

Conversions

Pour convertir un mot binaire en nombre décimal, il suffit de multiplier la valeur de chaque bit par son poids, puis d'additionner chaque résultat. Ainsi, le mot binaire 0101 vaut en décimal :

$$\begin{aligned} &2^3 \times 0 + 2^2 \times 1 + 2^1 \times 0 + 2^0 \times 1 \\ &= 8 \times 0 + 4 \times 1 + 2 \times 0 + 1 \times 1 \\ &= 5 \end{aligned}$$

L'octet (=Byte)

L'octet (en anglais byte ou B avec une majuscule dans les notations) est une unité d'information composée de 8 bits. Il permet par exemple de stocker un caractère, tel qu'une lettre ou un chiffre.

Ce regroupement de nombres par série de 8 permet une lisibilité plus grande, au même titre que l'on apprécie, en base décimale, de regrouper les nombres par trois pour pouvoir distinguer les milliers. Le nombre « 1 256 245 » est par exemple plus lisible que « 1256245 ».

Une unité d'information composée de 16 bits est généralement appelée mot (en anglais word).

Une unité d'information de 32 bits de longueur est appelée mot double (en anglais double word, d'où l'appellation dword).

Pour un octet, le plus petit nombre est 0 (représenté par huit zéros 00000000), et le plus grand est 255 (représenté par huit chiffres « un » 11111111), ce qui représente 256 possibilités de valeurs différentes.

$2^7 = 128$	$2^6 = 64$	$2^5 = 32$	$2^4 = 16$	$2^3 = 8$	$2^2 = 4$	$2^1 = 2$	$2^0 = 1$
0	0	0	0	0	0	0	0
1	1	1	1	1	1	1	1

Longtemps l'informatique s'est singularisée par l'utilisation de différentes valeurs pour les unités du système international. Ainsi beaucoup d'informaticiens ont appris que 1 kilooctet valait 1024 octets. Or, depuis décembre 1998, l'organisme international IEC a statué sur la question (<http://physics.nist.gov/cuu/Units/binary.html>). Voici donc les unités standardisées :

Un kilooctet (ko ou kB) = 1000 octets

Un Mégaoctet (Mo ou MB) = 1000 ko = 1 000 000 octets

Un Gigaoctet (Go ou GB) = 1000 Mo = 1 000 000 000 octets

Un Téraoctet (To) = 1000 Go = 1 000 000 000 000 octets

Attention ! De nombreux logiciels (parfois même certains systèmes d'exploitation) utilisent toujours la notation antérieure à 1998 pour laquelle :

Un kilooctet (ko) = $2^7 \times 1000 = 2^{10}$ octets = 1024 octets

Un Mégaoctet (Mo) = 2^{20} octets = 1024 ko = 1 048 576 octets

Un Gigaoctet (Go) = 2^{30} octets = 1024 Mo = 1 073 741 824 octets

Un Téraoctet (To) = 2^{40} octets = 1024 Go = 1 099 511 627 776 octets

Il est également utile de noter que la communauté internationale dans son ensemble utilise préférentiellement le nom de « byte » plutôt que le terme « octet » purement francophone. Cela donne les notations suivantes pour kilobyte, mégabyte, gigabyte et terabyte :

kB, MB, GB, TB

Notez l'utilisation d'un B majuscule pour différencier Byte et bit.

Les opérations arithmétiques simples telles que l'addition, la soustraction et la multiplication sont faciles à effectuer en binaire.

L'addition en binaire

L'addition en binaire se fait avec les mêmes règles qu'en décimale :

On commence à additionner les bits de poids faible (les bits de droite) puis on a des retenues lorsque la somme de deux bits de même poids dépasse la valeur de l'unité la plus grande (dans le cas du binaire : 1), cette retenue est reportée sur le bit de poids plus fort suivant...

Par exemple :

$$\begin{array}{r} \\ + \\ \hline \\ 1 \end{array}$$

La multiplication en binaire

La table de multiplication en binaire est très simple :

$$0 \times 0 = 0$$

$$0 \times 1 = 0$$

$$1 \times 0 = 0$$

$$1 \times 1 = 1$$

La multiplication se fait en formant un produit partiel pour chaque digit du multiplicateur (seuls les bits non nuls donneront un résultat non nul). Lorsque le bit du multiplicateur est nul, le produit partiel est nul, lorsqu'il vaut un, le produit partiel est constitué du multiplicande décalé du nombre de positions égal au poids du bit du multiplicateur.

Par exemple :

x		0	1	0	1
		0	0	1	0
		0	0	0	0
	0	1	0	1	
0	0	0	0		
-	0	1	0	1	0

Stockage des valeurs numériques

La représentation des nombres sur un ordinateur est indispensable pour que celui-ci puisse les stocker, les manipuler. Toutefois le problème est qu'un nombre mathématique peut être infini (aussi grand que l'on veut), mais la représentation d'un nombre dans un ordinateur doit être faite sur un nombre de bits prédéfini. Il s'agit donc de prédéfinir un nombre de bits et la manière de les utiliser pour que ceux-ci servent le plus efficacement possible à représenter l'entité.

Ainsi il serait idiot de coder un caractère sur 16 bits (65536 possibilités) alors qu'on en utilise généralement moins de 256...

Entier, double, nombre réel...

Le choix à faire (c'est-à-dire le nombre de bits à utiliser) dépend de la fourchette des nombres que l'on désire utiliser. C'est ce qu'on appelle en programmation le "type".

Par exemple les nombres entiers seront codés sur 1 jusqu'à 8 octets (dépend du compilateur ou de ce que définit l'utilisateur) .

Le type "reel" (ou "float" en anglais) seront codés sur beaucoup plus de bits, généralement utilisés pour les nombres décimaux à virgules.

Le code ASCII

La mémoire de l'ordinateur conserve toutes les données sous forme numérique. Il n'existe pas de méthode pour stocker directement les caractères. Dans les années 60, le code ASCII (American Standard Code for Information Interchange) est adopté comme standard.

Chaque caractère possède donc son équivalent en code numérique.

Il permet le codage de caractères sur 8 bits, soit 256 caractères possibles.

7	55	37
8	56	38
9	57	39
:	58	3A
;	59	3B
<	60	3C
=	61	3D
>	62	3E
?	63	3F
@	64	40
A	65	41
B	66	42
C	67	43
D	68	44
E	69	45
F	70	46
G	71	47
H	72	48
I	73	49
J	74	4A
K	75	4B
L	76	4C
M	77	4D

Le code ASCII n'est pas unique et dépend fortement de la plateforme utilisée !