

Les différents systèmes de numération

Représentation des données pour un système numérique/informatique

Un ordinateur pourrait se résumer à un ensemble de commutateurs électrique pouvant prendre deux états :

- En fonction _____ Le courant passe
- Hors fonction _____ Le courant ne passe pas

Pour les différentes taches qu'ils effectuent de nos jours, les ordinateurs utilisent le système de numération binaire.

***Système de numération** : ensemble de symboles permettant de réaliser des opérations mathématiques. Par exemple, le langage binaire est constitué de 2 symboles, 0 et 1. Correspondant aux 2 états que peuvent prendre les commutateurs.*

Du fait que les humains fonctionnent avec le système décimal, l'ordinateur doit pouvoir effectuer cette traduction afin de pouvoir traiter les informations des utilisateurs. Ces nombres binaires sont exprimés en _____ Bits _____, qui

constituent la plus petite unité d'information d'un ordinateur.

Un groupe de 8 bits correspond à un _____ Octet _____ ou « BYTE » en anglais, qui représente un caractère de données. Pour un ordinateur, un octet représente également un emplacement de mémoire adressable.

Par exemple, la représentation binaire des caractères du clavier et des caractères de contrôle est donnée dans le tableau de l'*American Standard Code for Information Interchange* (ASCII) dont voici un extrait :

Décimal	Hexadécimal	Octal	Binaire	Caractères
0	0	000	00000000	NUL
1	1	001	00000001	SOH
2	2	002	00000010	STX
3	3	003	00000011	ETX
4	4	004	00000100	EOT
5	5	005	00000101	ENQ
42	2A	052	00101010	*
43	2B	053	00101011	+
44	2C	054	00101100	,
45	2D	055	00101101	-
46	2E	056	00101110	,
58	3A	072	00111010	:
59	3B	073	00111011	;
60	3C	074	00111100	<
61	3D	075	00111101	=
62	3E	076	00111110	>
63	3F	077	00111111	?
64	40	100	01000000	@
65	41	101	01000001	A
66	42	102	01000010	B
67	43	103	01000011	C
68	44	104	01000100	D
69	45	105	01000101	E

Extrait du tableau ASCII

Ce tableau nous présente les équivalences entre différents systèmes de numération que nous allons étudier par la suite. Si nous regardons la colonne « binaire », nous voyons que tous les caractères sont exprimés grâce à une combinaison de 8 bits pouvant prendre la valeur 0 ou la valeur 1.

Pour mieux comprendre le système ASCII, considérons un fichier écrit avec le bloc notes de Windows. La structure de ce fichier est juste une suite d'octet représentant les caractères saisis par l'utilisateur.

Du fait de la taille des informations contenues dans les ordinateurs actuels, différentes unités de mesure ont été mis en place :

Unité décimale	Valeur	Unité binaire	Valeur	Exemples
Bit (b)	Chiffre binaire 1 ou 0	Bit	Chiffre binaire 1	+5 volts ou 0 volts
Octet (o)	8 bits	Octet	8 bits	01001100 correspond à la lettre L en ASCII
Kilo-octet (Ko)	1 kilo-octet =1 000 octets	Kibi-octet (Kio)	1 kio-octet =1024 octets	mail type : 2ko premiers PC : 64Ko de Ram
Méga-octet (Mo)	1 méga-octet =1000 kilo-octets	Mebi-octet (Mio)	1 Mio = 1024 Kio =1048576 octets	disquette = 1,44 Mo cdrom = 700 Mo
Giga-octet (Go)	1 giga-octet =1000 méga-octets	Gibi-octet (Gio)	1 Gio = 1024 Mio =1073741824 octets	disque dur type = 500 Go
Téra-octet (To)	1 téra-octet =1000 giga-octets	Tébi-octet (Tio)	1 Tio = 1024 Gio = 1099511627776 octets	quantité théorique de données transmissibles par une fibre optique en 1 seconde

Les différentes unités de mesures

Les différents systèmes de numération

L'homme est habitué dès le plus jeune âge à utiliser un système de numération pour représenter des valeurs. Ce système comporte 10 symboles : 0 1 2 3 4 5 6 7 8 9 et se nomme « *système de numération décimal* ».

Ce système constitue la base du calcul pour les hommes, principalement parce que ces derniers ont 10 doigts. Nous utiliserons d'ailleurs ce système comme système de référence dans la suite du cours. Cependant, il existe d'autres systèmes de numération pouvant représenter des valeurs.

Une valeur est de ce fait une notion abstraite pouvant être exprimé selon différents systèmes :

Par exemple, nous savons qu'un ordinateur fonctionne avec des commutateurs électriques pouvant avoir 2 états : en fonction et hors fonction. L'ordinateur va donc utiliser un système de numération avec deux symboles : 0 et 1. C'est ce que l'on appelle le système binaire. Il fonctionne de manière analogue au système décimal sauf qu'il n'utilise que 2 symboles.

Exemple : 1011 qui équivaut en décimal à la valeur 11

Autres systèmes, le système hexadécimal, comportant 16 symboles 0 1 2 3 4 5 6 7 8 9 A B C D E F.

Les 6 lettres correspondent en décimal à A=10 B=11 C=12 D=13 E=14 F=15

Exemple : A2F54B qui équivaut en décimal à la valeur 10679627

Il est évident ici de l'utilité de disposer de plusieurs systèmes d'informations. Une fois que l'on est familiarisé avec ces différents systèmes, la valeur A2F54B est plus facile à manipuler ou à mémoriser que son équivalent décimal.

Voici les différents systèmes que nous utiliserons ainsi que leur spécificité (l'octal est peu utilisé) :

Nom	Symboles utilisés	Base	
binaire	0 1	2	b
octal	0 1 2 3 4 5 6 7	8	o
décimal	0 1 2 3 4 5 6 7 8 9	10	(d)
hexadécimal	0 1 2 3 4 5 6 7 8 9 A B C D E F	16	h

les différents systèmes de numération

Méthodes de conversion de base

Méthodes de conversion de base

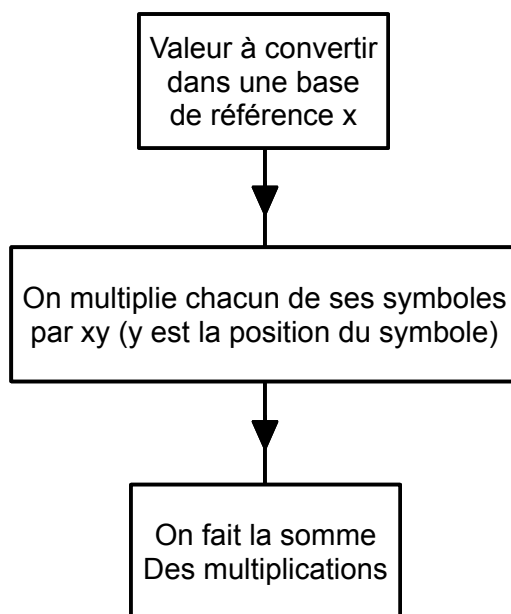
Conversion en base 10 (décimal)

Le système décimal repose sur les puissances de 10. Chaque symbole composant un nombre décimal représente une puissance de 10 ; chacun ayant pour exposant sa position dans le nombre en partant de la droite ; multiplié par le symbole occupant cette position.

Décomposition d'un nombre décimal

$$25642 = 2 \times 10^4 + 5 \times 10^3 + 6 \times 10^2 + 4 \times 10^1 + 2 \times 10^0$$

Pour convertir une valeur exprimée dans un système en une valeur utilisant le système de numération décimal, l'algorithme est le suivant :



algorithme de conversion d'un système en base x vers le système décimal

Exemple :

20165 (octal)

$$= 2 \times 8^4 + 0 \times 8^3 + 1 \times 8^2 + 6 \times 8^1 + 5 \times 8^0 = 8182 + 0 + 64 + 48 + 5 = \mathbf{8309(dec)}$$

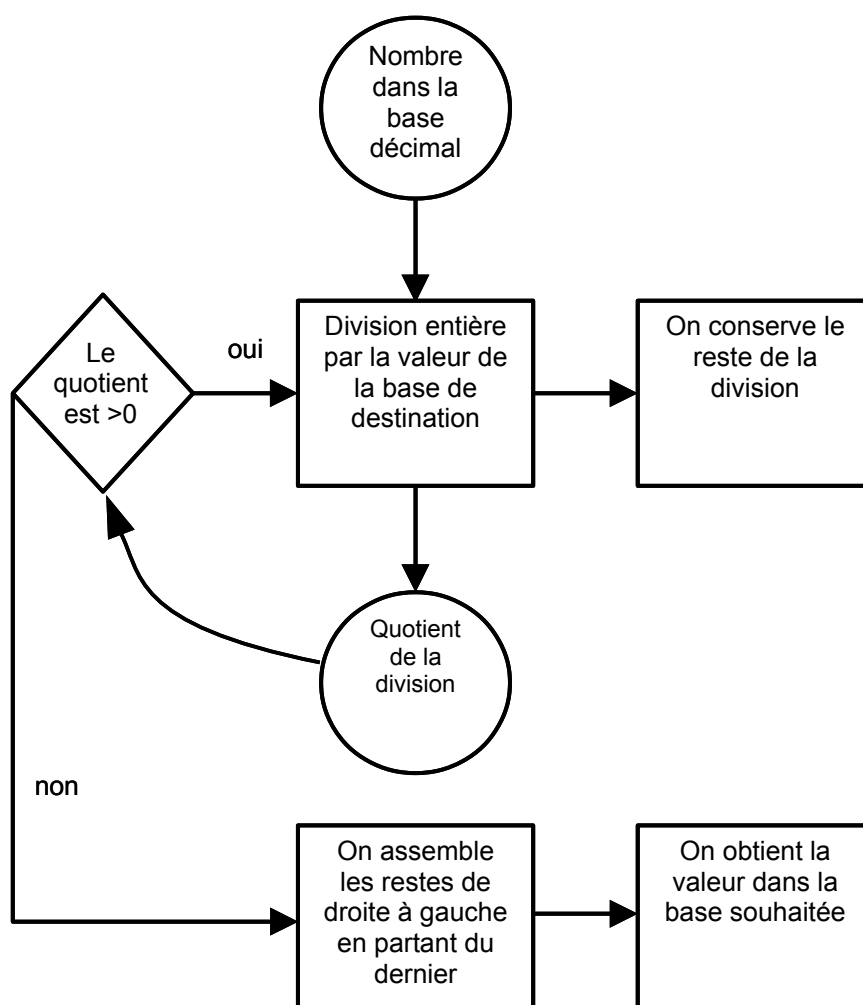
A2F54B (hexadécimal)

$$= A(10) \times 16^5 + 2 \times 16^4 + F(15) \times 16^3 + 5 \times 16^2 + 4 \times 16^1 + B(11) \times 16^0$$

$$= 10485760 + 131072 + 61440 + 1280 + 64 + 11 = \mathbf{10679627(dec)}$$

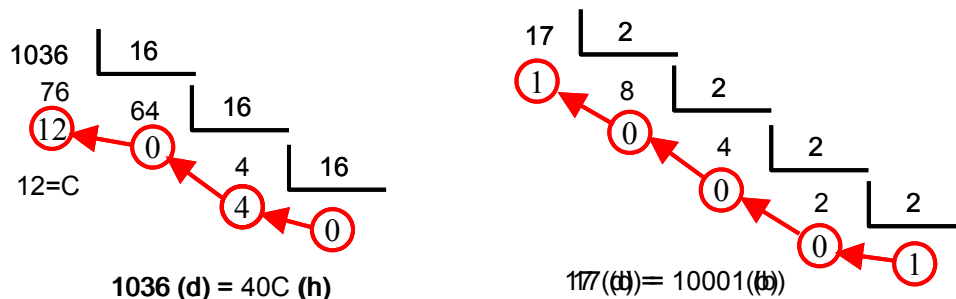
Conversion de base 10 (décimal) vers les autres bases

Nous allons maintenant voir comment effectuer l'opération inverse, à savoir convertir la base décimale vers les autres bases :



Algorithme de conversion d'un nombre décimal vers une autre base

Exemple : Conversion de 1036(dec) en hexadécimal et 17(dec) en binaire.



Conversion entre la base 2 (binaire) et la base 16 (hexadécimal)

Dans le domaine informatique l'utilisation du binaire s'est imposée par la nature des signaux électriques. Le fonctionnement des systèmes informatiques se résume par des "contacts" ouverts ou fermés. Niveaux logiques 5 volts et 0 volt par exemple.

La manipulation de nombres ou de données codés en binaire devient rapidement très difficile pour les informaticiens. On note que la base 16 a des caractéristiques intéressantes pour converser avec les systèmes binaires.

Correspondance binaire hexadécimal

Dec	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Bin	0	1	10	11	100	101	110	111	1000	1001	1010	1011	1100	1101	1110	1111
Hex	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F

On constate que le plus grand nombre que l'on peut écrire avec un caractère en hexadécimal est F correspond au plus grand nombre que l'on peut écrire avec 4 caractères en binaire (1111).

Conversion du binaire vers l'hexadécimal

Soit un nombre 1001011 binaire constitué de bits (binary digit) (chiffres constituant le nombre).

Repère du bit	6	5	4	3	2	1	0
Poids du bit	2^6	2^5	2^4	2^3	2^2	2^1	2^0
Nombre	1	0	0	1	0	1	1

Le nombre en binaire est décomposé en série de 4 caractères (4 bits) auxquels on affecte un poids de 2^0 à 2^3 .

Poids du bit	2^3	2^2	2^1	2^0		2^3	2^2	2^1	2^0
Nombre	0	1	0	0		1	0	1	1
Somme décimal	4					11			
Somme hexa.	4					B			

Un 0 à gauche a été ajouté pour la démonstration mais n'a aucune influence.

Nous pouvons dire que 1001011 en binaire est égal à 4B en hexadécimal.

Conversion de l'hexadécimal vers du binaire

L'opération inverse est assez simple à mettre en œuvre.

A partir d'un nombre en hexadécimal, constitué de n caractères. Il faudra au maximum $4 \times n$ bits pour l'écrire en binaire.

Exemple : C5(hexadécimal) est égal à 11000101 (binaire)

Nombre hexa.	C					5			
Valeur décimale	12=8+4					5=4+1			
Poids du bit	2^3	2^2	2^1	2^0		2^3	2^2	2^1	2^0
Nombre binaire	1	1	0	0		0	1	0	1

Exercices de conversion

Convertir

1) Convertir ces nombres en binaire :

a) (125)₁₀ :

b) (92)₁₀ :

c) (27)₁₀ :

d) (255)₁₀ :

2) Convertir ces nombres en décimal :

a) (0000 0110)₂ :

b) (0110 0101)₂ :

c) (1000 1110)₂ :

d) (1010 1111)₂ :

3) Convertir ces nombres en décimal :

- a) $(A1)_{16}$:
- b) $(F2)_{16}$:
- c) $(E2A)_{16}$:
- d) $(3B)_{16}$:
- e) $(14D)_{16}$:

4) Convertir ces nombres en binaire :

- a) $(1F)_{16}$:
- b) $(2C)_{16}$:
- c) $(9E)_{16}$:
- d) $(3B)_{16}$:
- e) $(B6)_{16}$:

5) Convertir ces nombres dans les bases demandées :

- a) $(18)_{10}$ en base 8:
- b) $(24)_{10}$ en base 7:
- c) $(44)_{10}$ en base 6: