

## Note sur l'écriture en base 2 et les changements de base

Le but de cette note est de motiver l'utilisation de bases autre que 10 dans le cadre de l'informatique, et de rappeler les opérations basiques de changement de base ainsi que les notations usuelles.

### 1 Introduction

Comme on le sait, les ordinateurs représentent l'information, au niveau le plus élémentaire, comme des collections de 0 et de 1. Cela est dû à leur conception. Le processeur d'un ordinateur est composé de transistors, des circuits permettant couper (0) ou laisser passer (1) une quantité de courant électrique.

De cette capacité de représenter deux états et de pouvoir opérer dessus, il découle naturellement que les ordinateurs travaillent en base 2. Un chiffre en base 2 (donc 0 ou 1) est appelé un *bit* (pour l'anglais *binary digit*). Un groupe de huit bits est appelé un *octet* ou *byte* en anglais. Attention donc à ne pas confondre *bit* et *byte*.

### 2 Bases et système de numération

Un système de numération est une façon de représenter les nombres. Il existe plusieurs méthodes. La plus utilisée, celle avec laquelle nous écrivons les nombres et la notation positionnelle. Dans cette notation, un nombre est composé d'une succession de symboles (les chiffres) multipliés par une puissance d'une valeur implicite et additionnés entre eux<sup>1</sup>. Cette valeur implicite est la *base*. La *base* détermine le nombre de symboles distincts à utiliser.

Dans la vie de tous les jours, nous utilisons la base 10. Cela signifie que les nombres sont formés de 10 symboles distincts : 0, 1, 2, 3, 4, 5, 6, 7, 8 et 9.

Un nombre tel que « 3117 » doit être compris comme :

$$3 \times 10^3 + 1 \times 10^2 + 1 \times 10^1 + 7 \times 10^0$$

En base 2, il n'y a que deux chiffres possibles 0 et 1. Un nombre tel que « 11001 » doit donc être compris comme :

$$1 \times 2^4 + 1 \times 2^3 + 0 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 = 25$$

Lorsque l'on utilise plusieurs bases dans le même contexte, il est souvent naturel d'indiquer la base dans laquelle un nombre doit être lu en indice de ce dernier, afin d'éviter toute ambiguïté. Par exemple, on peut écrire : «  $11001_2 = 25_{10}$  ».

### 3 Bases utilisées en informatique

Comme on l'a vu, les ordinateurs manipulent des quantités représentées en base 2. Cependant, représenter des nombres en bases 2 est fastidieux pour des humains : l'écriture des nombres est plus longue (on a moins de symboles, donc il faut en mettre plus). Une base couramment utilisée est la base 16. Dans cette base, on utilise 16 chiffres : 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E et F. Ainsi, le nombre « 12AC3 » doit être compris comme :

$$1 \times 16^4 + 2 \times 16^3 + 10 \times 16^2 + 12 \times 16^1 + 3 \times 16^0 = 76483$$

L'intérêt de la base 16 est double :

---

1. Il existe d'autres types de systèmes de numération. Par exemple, la *numération romaine* représente les valeurs des nombres par des symboles et additionne les valeurs  $CXXII = C + XX + II = 100 + 20 + 2 = 122$ .

— l'écriture des nombres est de longueur similaire à celle en base 10. Dans l'exemple ci-dessus

$$76483_{10} = 12AC3_{16} = 1\ 0010\ 1010\ 1100\ 0011_2$$

Le nombre s'écrit avec 5 chiffres en base 10 et 16, mais avec 17 chiffres en base 2.

— la conversion de la base 2 à la base 16 est très simple.

En effet, comme  $16 = 2^4$ , il suffit de grouper les bits d'un nombre écrit en binaire par paquets de 4 et d'appliquer une correspondance automatique :

<i>base 2</i>	<i>base 16</i>
0000	0
0001	1
0010	2
0011	3
0100	4
0101	5
0110	6
0111	7
1000	8
1001	9
1010	A
1011	B
1100	C
1101	D
1110	E
1111	F

Ainsi «  $1001\ 0100\ 1010\ 0010_2 = 94A2_{16}$  ».