**Le système décimal**

Pour que vous compreniez le fonctionnement du binaire, et des systèmes de comptage en général (plus communément appelés **bases**), je vais commencer par faire une petite réintroduction à la base 10 que vous connaissez tous.

En effet, tout le monde sait compter en base 10 (décimal). Mais comment ça marche ? Comment est construit notre système ? Pour répondre à cette question à l'apparence simple, oubliez tout et reprenons depuis le début : comment avez-vous appris à compter à l'école ?

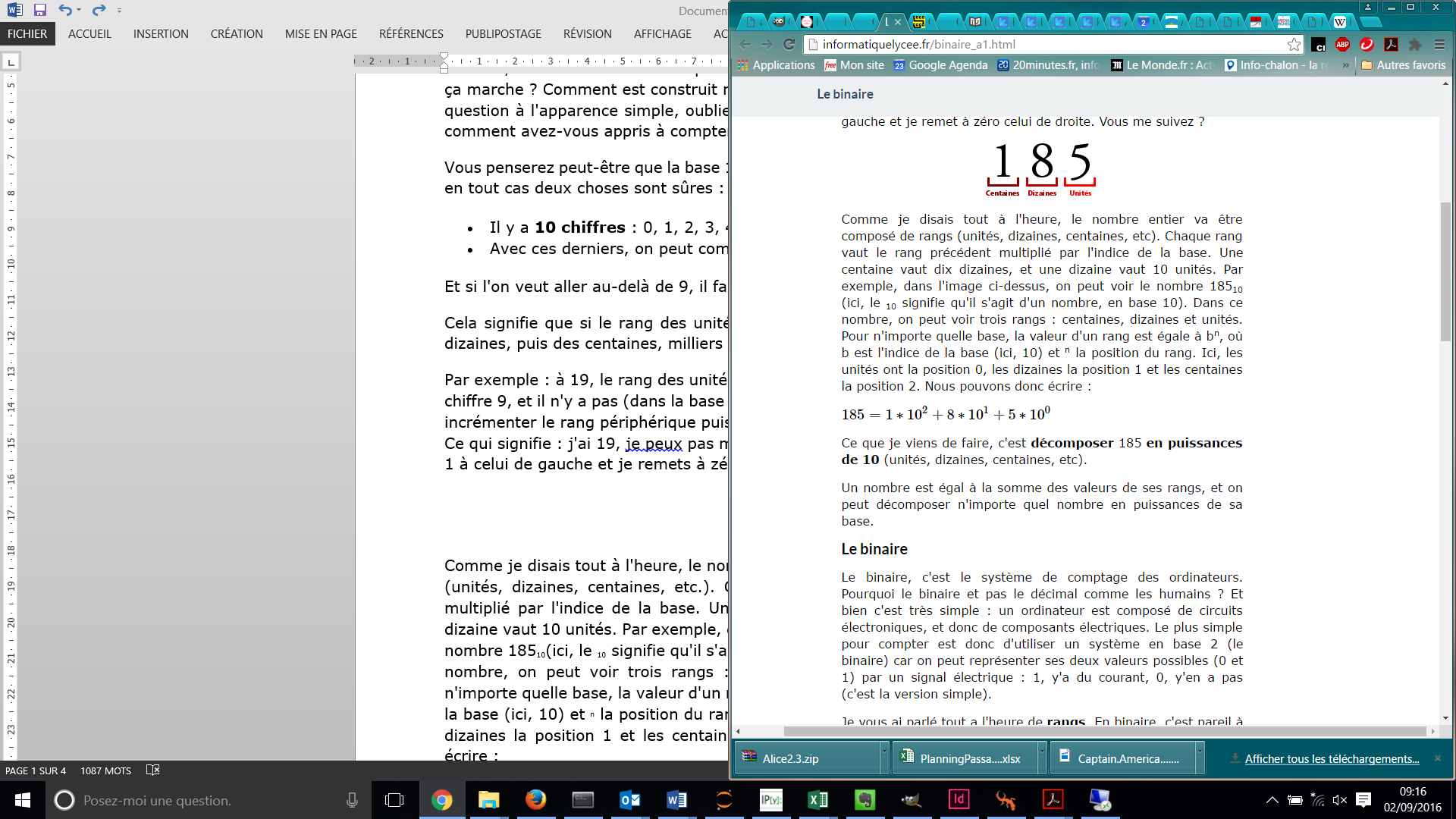
Vous penserez peut-être que la base 10 vient du fait qu'on a 10 doigts, mais en tout cas deux choses sont sûres :

* Il y a **10 chiffres** : 0, 1, 2, 3, 4, 5, 6, 7, 8, 9.
* Avec ces derniers, on peut compter jusqu'à 9.

Et si l'on veut aller au-delà de 9, il faut **changer de rang**.

Cela signifie que si le rang des unités est plein, il faut passer à celui des dizaines, puis des centaines, milliers et j'en passe.

Par exemple : à 19, le rang des unités est "saturé" (plein), car il contient le chiffre 9, et il n'y a pas (dans la base 10) de valeur plus élevée. Il faut donc incrémenter le rang périphérique puis réinitialiser l'état de celui des unités. Ce qui signifie : j'ai 19, je peux pas mettre plus de 9 à droite, donc j'ajoute 1 à celui de gauche et je remets à zéro celui de droite. Vous me suivez ?



Comme je disais tout à l'heure, le nombre entier va être composé de rangs (unités, dizaines, centaines, etc.). Chaque rang vaut le rang précédent multiplié par l'indice de la base. Une centaine vaut dix dizaines, et une dizaine vaut 10 unités. Par exemple, dans l'image ci-dessus, on peut voir le nombre 18510(ici, le 10 signifie qu'il s'agit d'un nombre, en base 10). Dans ce nombre, on peut voir trois rangs : centaines, dizaines et unités. Pour n'importe quelle base, la valeur d'un rang est égale à bn, où b est l'indice de la base (ici, 10) et n la position du rang. Ici, les unités ont la position 0, les dizaines la position 1 et les centaines la position 2.

Nous pouvons donc écrire : 185=1∗102+8∗101+5∗100

Ce que je viens de faire, c'est **décomposer** 185 **en puissances de 10** (unités, dizaines, centaines, etc).

Un nombre est égal à la somme des valeurs de ses rangs, et on peut décomposer n'importe quel nombre en puissances de sa base.

**Le binaire**

Le binaire, c'est le système de comptage des ordinateurs. Pourquoi le binaire et pas le décimal comme les humains ? Et bien c'est très simple : un ordinateur est composé de circuits électroniques, et donc de composants électriques. Le plus simple pour compter est donc d'utiliser un système en base 2 (le binaire) car on peut représenter ses deux valeurs possibles (0 et 1) par un signal électrique : 1, y'a du courant, 0, y'en a pas (c'est la version simple).

Je vous ai parlé tout a l'heure de **rangs**. En binaire, c'est pareil à la différence qu'on utilise le terme **bit** (sans e à la fin), qui est la contraction de "binary digit", littéralement "chiffre binaire". Par exemple, le nombre 10011 occupe 5 bits. Là où tout se complique, c'est que comme je l'ai expliqué, chaque rang en binaire ne peut avoir que deux valeurs (binaire = base 2) différentes : 0 ou 1. Pour la base 10, chaque rang représente une puissance de 10, pour la base 2, chaque rang occupe une puissance de 2. Comme un bon exemple vaut mieux qui looooong discours, voici comment compter en binaire jusqu'à 10 :

|  |  |  |
| --- | --- | --- |
| **Nombre en décimal** | **Nombre en binaire** | **Le pourquoi du comment** |
| 0 | 0 | Pour l'instant, ça va. |
| 1 | 1 | Là encore, c'est simple. |
| 2 | 10 | Le premier rang ayant été rempli, on passe au suivant ! |
| 3 | 11 | On re-remplit le rang 1. |
| 4 | 100 | Le rang 2 est plein, le rang 1 aussi, qu'à cela ne tienne, on passe au suivant. |
| 5 | 101 | On continue en suivant la même méthode. |
| 6 | 110 |
| 7 | 111 |
| 8 | 1000 | On commence le rang 4. |
| 9 | 1001 | On continue comme tout à l'heure. |
| 10 | 1010 |
| ... | | |

Retenez juste ceci : **entamer le rang suivant quand l'actuel est plein**.

Comme on a pu le voir, compter jusqu'à 10 ou 20 reste aisé, mais imaginons un instant que je vous demandasse d'écrire 185 en binaire ? Vous allez faire chaque rang, un par un ?

**Conversion décimal-binaire**

Pour l'instant, on n'a compté que jusqu'à 10. Mais pour les plus grands nombres, la méthode précédente peut se révéler fastidieuse. Sans plus attendre, la méthode !

**La méthode !**

Il existe bien sûr plusieurs méthodes de conversion, mais j'ai choisi la plus simple et la plus rapide. Il s'agit de la méthode euclidienne qui va sûrement vous rappeler les cours du PGCD en 4ème.

Cette méthode, en plus d'être facile à utiliser en programmation (c'est un algorithme) est une des meilleures lorsqu'il s'agit de traiter les grands nombres.

Voici la méthode :

* On prend le nombre en base 10 (forme normale).
* On le divise par 2 et on note le reste de la division (soit 1 soit 0)
* On refait la même chose avec le quotient précédent, et on met de nouveau le reste de côté.
* On réitère la division, jusqu'à ce que le quotient soit 0.
* Le nombre en binaire apparaît alors : il suffit de prendre tous les restes de bas en haut.

Et l'exemple :

* Le nombre est 185.
* 185÷2=92+1
* 92÷2=46+0
* 46÷2=23+0
* 23÷2=11+1
* 11÷2=5+1
* 5÷2=2+1
* 2÷2=1+0
* 1÷2=0+1

**Attention, il faut bien lire de bas en haut !**  
**185** en base 10 vaut donc **10111001** en binaire.

**Conversion binaire-décimal**

Je vous rassure tout de suite : c'est plus simple dans ce sens-là que dans l'autre.

Prenons un nombre au hasard, tel que 11010011. Il s'étale sur 8 rangs, et comme dit précédemment, chaque rang correspond à une puissance de deux.  
Le premier rang (en partant de la **droite**) est le rang 0, le second est le 1, etc.

Pour convertir le tout en décimal, on procède de la manière suivante : on multiplie par 20 la valeur du rang 0, par 21 la valeur du rang 1, par 22 la valeur du rang 2, [...], par 210 la valeur du rang 10, etc.

Après ça, il ne reste plus qu'à remplacer les puissances de 2 par leurs valeurs et de calculer la somme : (**Attention à bien partir de la droite !**)  
11010011binaire=1∗1+1∗2+0∗4+0∗8+1∗16+0∗32+1∗64+1∗128=211décimal