

Une brève introduction sur ce qu'est un ordinateur

Ce qui suit n'est pas un véritable cours d'informatique à proprement dit. Il s'agit juste de vous donner quelques éléments vous permettant d'appréhender de façon sommaire le fonctionnement d'un ordinateur afin de mieux comprendre et accepter (sans vous énerver) les erreurs inévitables que vous rencontrerez dans l'écriture de vos futurs programmes et de percevoir le véritable apport de l'ordinateur pour l'exécution de certaines tâches.

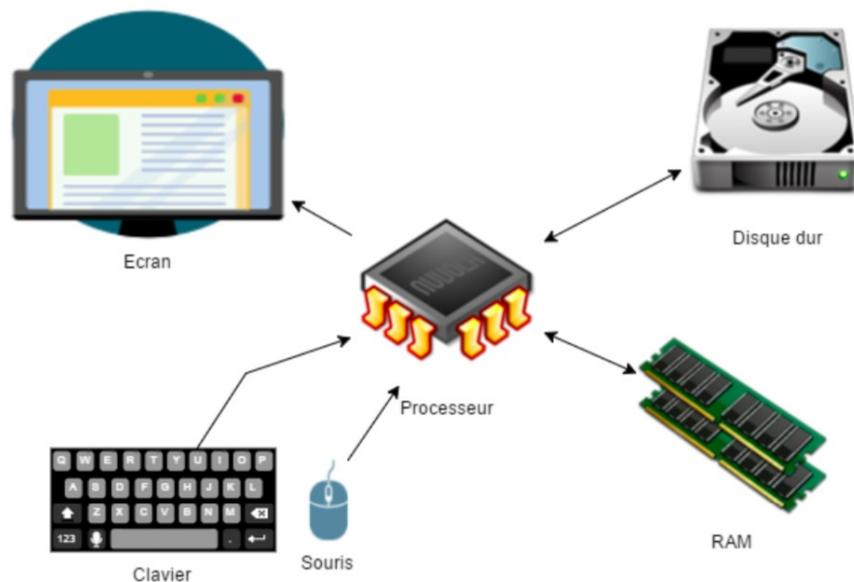
I Présentation générale d'un ordinateur

La structure de base d'un ordinateur (ou de tout autre appareil informatique : smartphone, appareil photo numérique, tablette...) est basée sur la schéma suivant :

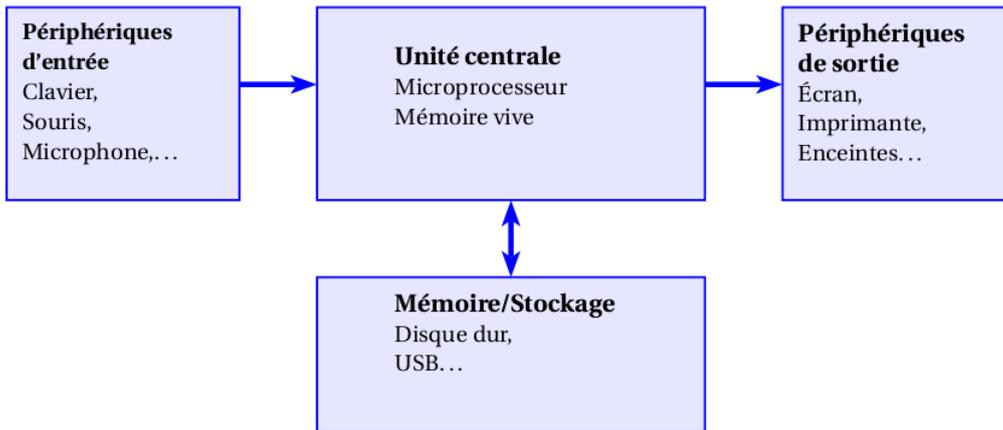


1. Les composants (le hardware)

a) Schéma général

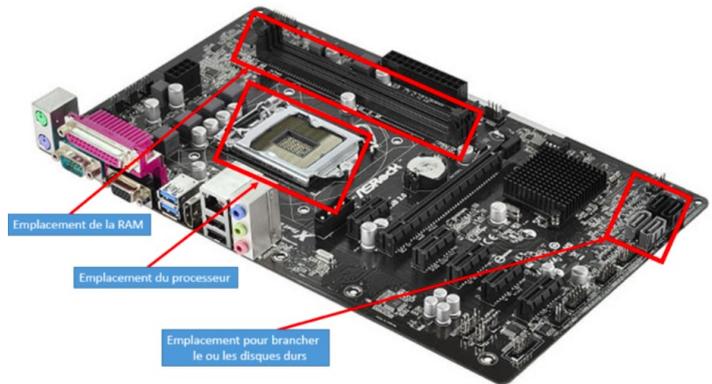


De manière plus schématique



- **Le disque dur** est un matériel qui va permettre d'enregistrer les données.
Nous pouvons le comparer à un bloc-notes où l'on viendrait écrire et lire les informations que nous voudrions conserver. Les données écrites sur le disque dur sont conservées même lorsque l'on éteint l'ordinateur.
- **La RAM (Random Access Memory)** est la mémoire (non permanente) de l'ordinateur, on pourrait la comparer à notre mémoire. Cette mémoire s'efface lorsque nous éteignons l'ordinateur.
- **Le processeur** : véritable « cerveau » de l'ordinateur.
Il contrôle l'ensemble et va calculer, traiter les différentes informations entrantes ou sortantes.

Ces composants sont tous à l'intérieur de l'ordinateur et rarement accessibles. Ils sont tous branchés sur une grande carte électronique : la « carte mère ».



- Le clavier, la souris sont des périphériques d'entrées. Les informations vont toujours dans le sens du périphérique vers le processeur.
- L'écran, une imprimante... sont des périphériques de sortie. Les informations vont toujours dans le sens du processeur vers le périphérique

b) Présentation détaillée de chacun des éléments

- **RAM** (Random Access Memory)

La RAM est une mémoire (volatile). Il s'agit d'une carte composée de puces électroniques. Ces cartes s'appellent des barrettes. L'avantage de la mémoire est que la lecture et l'écriture des informations se font très rapidement.

Cette mémoire s'efface dès que l'ordinateur est éteint.

- **Disque dur**

Puisque l'on ne peut pas tout stocker dans la RAM, il a fallu concevoir un système de stockage de données permanent : le disque dur (avant cela les données étaient enregistrées sur des cartes perforées, puis sur des bandes magnétiques).

L'avantage est que l'on peut stocker beaucoup d'informations et que ces informations restent enregistrées même si l'on éteint l'ordinateur. On y enregistre logiciels, photos, musiques, ...

- **Processeur : c'est le composant qui exécute les instructions.**

Un processeur est composé de transistors. Ce sont des petits composants électroniques qui agissent comme un interrupteur : ils vont laisser passer le courant ou pas en fonction de la valeur à l'entrée et de l'opération demandée : cet état peut être 1 ou 0 (allumé ou éteint, On ou Off).

Toutes ces opérations doivent être lancées à un rythme bien précis : ce rythme est cadencé par une horloge. Plus l'horloge tourne vite, plus le processeur pourra en traiter à la seconde. A chaque impulsion d'horloge, les transistors vont prendre un état 0 ou 1, ce qui va permettre de faire des calculs. On exprime ce cadencement en GHz (Giga Hertz). On peut parler d'un processeur à 4GHz soit capable d'effectuer 4×10^9 cycles d'opérations à la seconde !!

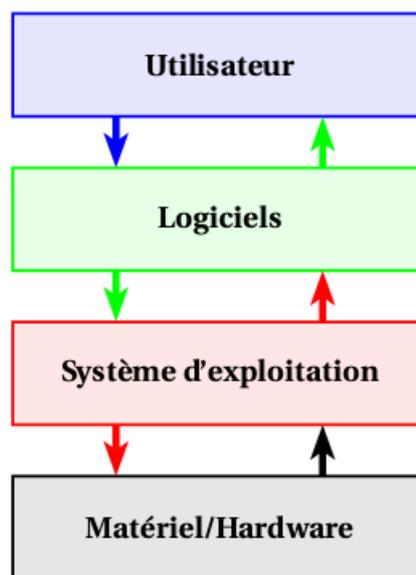
2. Fonctionnement de l'ordinateur

a) Le système d'exploitation

Lorsque l'ordinateur démarre, il commence par lancer un logiciel indispensable : le système d'exploitation (appelé aussi de façon générique l'OS (Operating System) de l'ordinateur) comme par exemple Windows, Linux ou MacOS. C'est l'OS qui va interpréter les commandes et en afficher les résultats. C'est en quelque sorte un intermédiaire entre l'utilisateur et le matériel.

Les OS modernes affichent généralement un écran d'accueil (appelé bureau) contenant toutes vos icônes. Ces icônes vont permettre de lancer les logiciels en cliquant dessus.

On peut résumer cela par le schéma suivant :



b) Que se passe-t-il quand on lance un logiciel ?

- Vous cliquez (ou double cliquez) sur l'icône.
- L'ordinateur va alors chercher dans le disque dur vers quel logiciel pointe cette icône.
- Il va lire le logiciel à partir du disque dur.
- Tout ce qui sera lu sera chargé dans la mémoire vive (RAM).
- A partir de la RAM, le processeur va exécuter le logiciel instruction après instruction.
- Une fois que vous fermerez le programme, celui-ci sera effacé de la RAM.

3. Comment dialoguer avec un ordinateur ?

a) Le langage machine

Pour faire exécuter une instruction, un programme à un ordinateur, il faut pouvoir communiquer avec lui. Cela se fait via le processeur qui ne va « comprendre » que des suites d'instructions du type on/off ou de manière plus « numériques » par des suites de 0 et de 1.

De manière simpliste, toute instruction, tout programme n'est qu'une suite de 0 et de 1 appelé bit (Binary Digit) qu'on regroupe en octet (groupe de 8 bits).

Une instruction est donc représentée par un nombre écrit dans un système binaire (nombre avec uniquement deux chiffres : 0 et 1).

« Le langage machine, ou code machine, est la suite de bits qui est interprétée par le processeur d'un ordinateur exécutant un programme informatique. C'est le langage natif d'un processeur, c'est-à-dire le seul qu'il puisse traiter. Il est composé d'instructions et de données à traiter codées en binaire. » *Wikipedia*

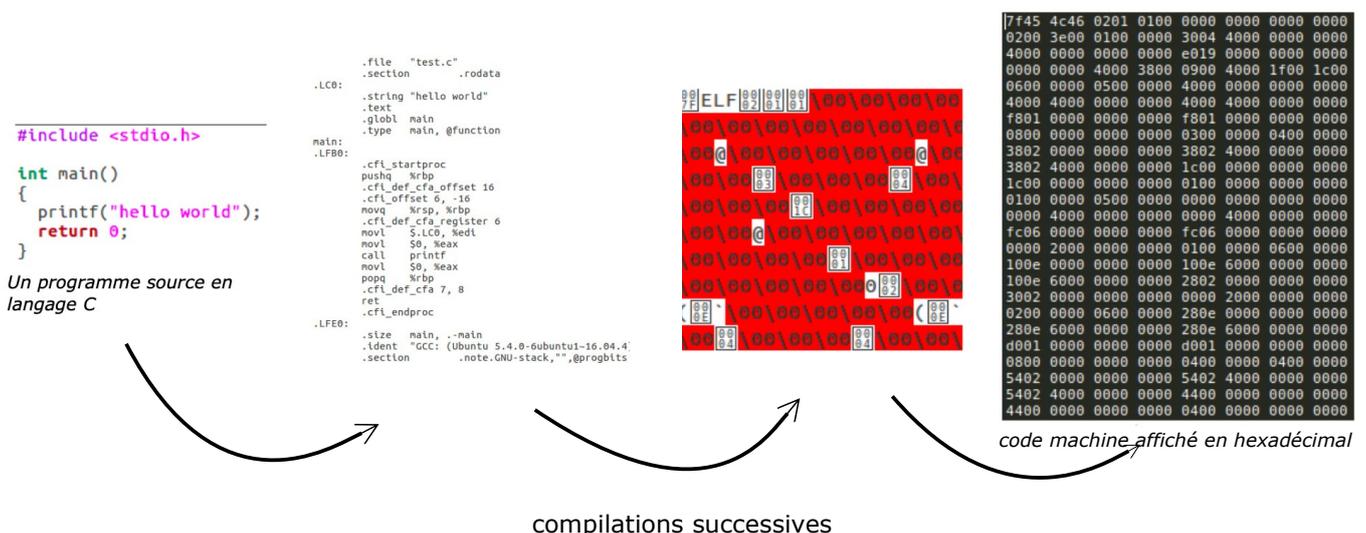
b) Langage de haut niveau

Le problème est que donner à l'ordinateur des instructions en binaire n'est pas chose aisée pour nous humains.

Le code machine que comprend le processeur est aujourd'hui généré automatiquement, généralement par le compilateur d'un langage de programmation.

Un compilateur est le terme utilisé pour désigner un programme qui transforme un code source écrit dans un langage de programmation appelé le langage source (par exemple Python) en un autre langage informatique appelé langage cible. Pour qu'il puisse être exploité par une machine, le compilateur traduit le code source, écrit dans un langage de haut niveau d'abstraction c'est-à-dire facilement compréhensible par l'humain, vers un langage de plus bas niveau, un langage d'assemblage ou langage machine.

Inversement, un programme qui traduit un langage de bas niveau vers un langage de plus haut niveau est un décompilateur.



II À propos du codage binaire

Base 10 et numérotation de position

Dans le système décimal, nous utilisons 10 symboles (chiffres) : 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 et la position d'un symbole correspond au chiffre des unités, des dizaines, des centaines, etc.

Exemple $1273 = 1 \times 10^3 + 2 \times 10^2 + 7 \times 10^1 + 3 \times 10^0$

En informatique, la base la plus utilisée est la base 2 avec donc deux chiffres : 0 et 1.

Le principe général de codage est alors le suivant :

si $a_{n-1} a_{n-2} \dots a_0$ représente un nombre en base 2 où $a_i = 0$ ou 1 pour $0 \leq i \leq n-1$, sa valeur décimale est :

$$a_{n-1} \times 2^{n-1} + a_{n-2} \times 2^{n-2} + \dots + a_0 \times 2^0$$

1. Donner l'écriture décimale des nombres suivants écrits en base 2.

1001 11101010

2. Donner l'écriture binaire des nombres suivants écrits en base 10. Vous écrirez les calculs nécessaires.

7 15 49 853

3. a) On donne un algorithme écrit en langage naturel :

```

n ← 0
C ← 3,4
Tant que C ≥ 1
    n ← n + 1
    C ← 0,8 × C
Fin Tant que
  
```

Que vaut C à la fin de l'algorithme ? Et n ?

- b) Proposer un algorithme écrit en langage naturel permettant :

- de convertir un nombre binaire en décimal
- de convertir un nombre décimal en binaire