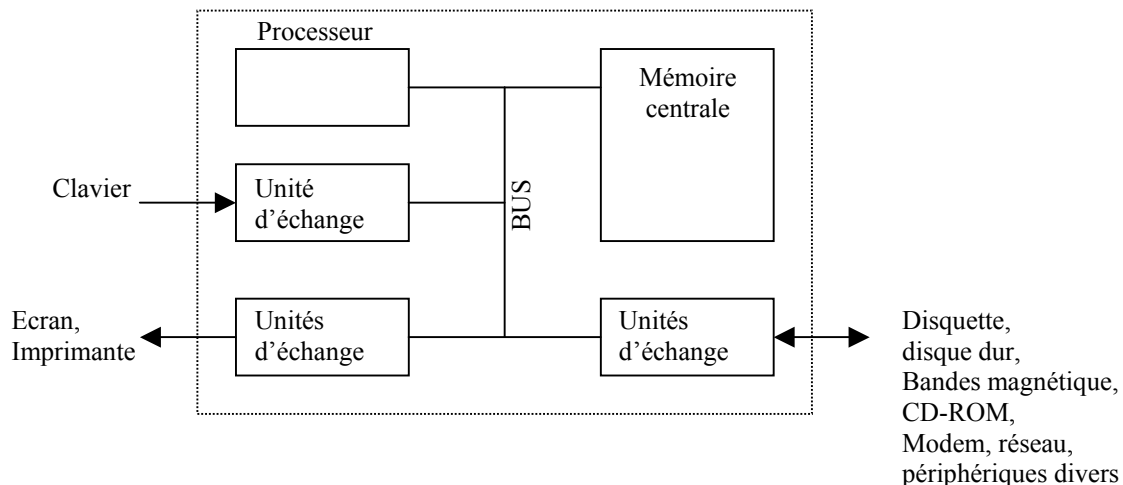


QUELQUES CONCEPTS INTRODUCTIFS

1- ARCHITECTURE ET FONCTIONNEMENT D'UN ORDINATEUR



La figure ci-dessus illustre l'architecture globale d'un ordinateur individuel. Ses principales composantes sont : le processeur ou unité centrale, la mémoire centrale, les unités d'échange. Des organes périphériques, connectés aux ports d'entrée/sortie de l'ordinateur, permettent l'interaction entre l'ordinateur et son environnement.

1.1- Codage de l'information

Le quantum d'information dans un ordinateur est le « *bit* » (Binary Digit). Un bit est une information binaire qui s'exprime physiquement par la présence ou l'absence d'une tension ou d'un courant électrique sur un fil de transmission d'information, ou l'état passant ou bloqué d'un transistor, ou ... Il est convenu de noter un bit 0 ou 1 (ce n'est qu'une notation).

Une unité d'information de n bits est appelée *mot*. Un mot de 8 bits est appelé *octet* (*byte*). Toutes les informations élémentaires que nous utilisons (chiffre, lettre, signe de ponctuation, symbole d'opération arithmétique, ...) sont codées en un ou plusieurs mots. Toutes les informations plus complexes (texte, image, son, ...) sont également représentées par des ensembles de mots.

Exemple. Supposons des mots de 8 bits : le nombre 5 aura pour code 00000101 ; le caractère '5' (i.e. la touche 5 du clavier) aura pour code 00110101.

La taille des mots est une constante pour un ordinateur donné. C'est toujours une puissance de 2. Elle croît au fil de l'évolution technologique. Aujourd'hui elle est typiquement de 32.

1.2- Les mémoires

L'homme possède deux grands types de mémoire : une *mémoire à court terme*, qui sert à stocker des étapes intermédiaires (par exemple mémoriser un numéro de téléphone lu le temps

de le composer, retenir l'emplacement de parking où on a garé sa voiture, ...), et une *mémoire à long terme*, qui mémorise les événements « inoubliables ». Les caractéristiques de ces deux types de mémoire sont très différentes :

- la mémoire à court terme est rapide (en enregistrement d'une information comme en restitution l'information mémorisée) mais limitée (en capacité de stockage) et volatile (si la mémorisation n'est pas entretenue, les données mémorisées se perdent)
- la mémoire à long terme a les caractéristiques presque inverses : elle est beaucoup moins volatile, bien moins limitée, mais souvent aussi plus lente.

La mémoire d'un ordinateur se divise aussi en mémoire à court terme et mémoire à long terme :

- la mémoire à court terme, dite *mémoire vive* ou *mémoire centrale*, est constituée de circuits mémoires situés près de l'organe de traitement (le processeur). C'est de la mémoire électronique rapide, d'assez faible capacité, et volatile (le contenu de la mémoire est perdu dès qu'on éteint l'ordinateur).
- la mémoire à long terme, dite *mémoire de masse*, est le plus souvent une mémoire magnétique (disque durs, disquettes, ...). Beaucoup plus lente que la mémoire centrale, elle est par contre permanente (sauf accident !) et facilement extensible.

La capacité d'une mémoire informatique est généralement exprimée en nombre de mots mémoire. Le préfixe « kilo » (K) vaut $2^{10} = 1\ 024$; le préfixe « méga » (M) vaut $1K \times 1K = 1024 \times 1024 = 1\ 048\ 576$; le préfixe « giga » (G) vaut $1M \times 1K = 1\ 073\ 741\ 824$.

1.3- Les périphériques

On distingue :

- périphériques de communication : clavier, souris, écran (terminal), imprimante, réseau, ...
- mémoires de masse : disquettes, disque dur, bandes magnétiques, ...

1.4- Les unités d'échange

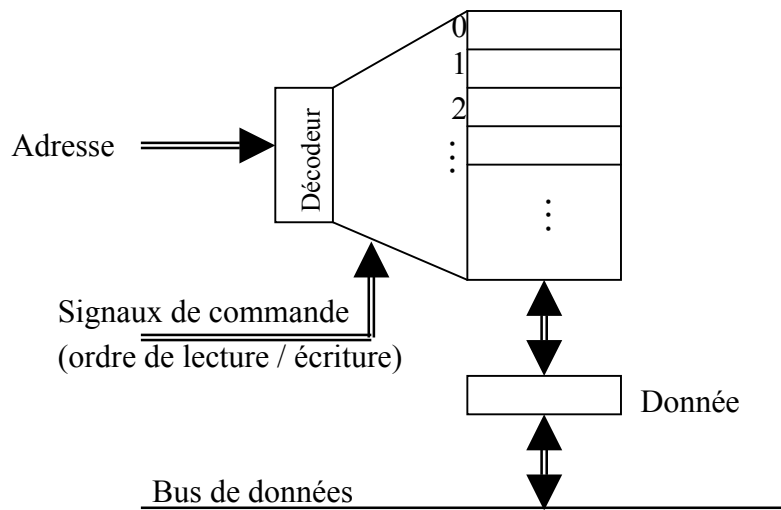
Les unités d'échange permettent la communication entre l'ordinateur et ses organes périphériques. Ce sont des interfaces qui ont un rôle de contrôle et de synchronisation. Ils peuvent posséder une certaine « intelligence » (avec un processeur interne) et comporter de la mémoire (mémoire tampon).

1.5- Les bus

Les échanges d'information entre les différents organes internes d'un ordinateur se font pas des « bus ». Un *bus* peut être vu comme un ensemble de n fils parallèles permettant de véhiculer des mots de n bits. On distingue typiquement : bus de données, bus d'adresses, et bus de commandes. Tous les échanges passant ces bus, ils constituent un goulot d'étranglement et ont un effet limitant sur la vitesse de l'ordinateur.

1.6- La mémoire centrale

La mémoire centrale sert à stocker les programmes et les données en cours de traitement. Elle organisée en *mots* rangés consécutivement, chaque *mot* étant repéré par une adresse unique.



1.7- Le processeur ou unité centrale (*CPU = Central Process Unit*)

Le processeur central est « l'âme » l'ordinateur. C'est lui qui effectue les calculs et qui contrôle l'exécution des programmes. Il comporte principalement :

- une unité de calcul (*Arithmetic and Logic Unit*)
- une unité de commande (*Control Unit*). C'est elle qui séquence l'exécution des *instructions (statements)* composant le programme. Elle boucle sur travail suivant :
 - a) lire l'instruction suivante en mémoire
 - b) la décoder
 - c) l'exécuter

1.8- Architecture logicielle

Le *matériel (hardware)* est l'ensemble des organes électroniques et plus généralement tout ce qui est physique, tangible.

Le *logiciel (software)* est l'ensemble des programmes définissant le fonctionnement d'un ordinateur, ainsi que leur phase de conception. Le terme logiciel réfère donc à la fois au processus d'analyse, à la programmation, et au produit qu'est le programme.

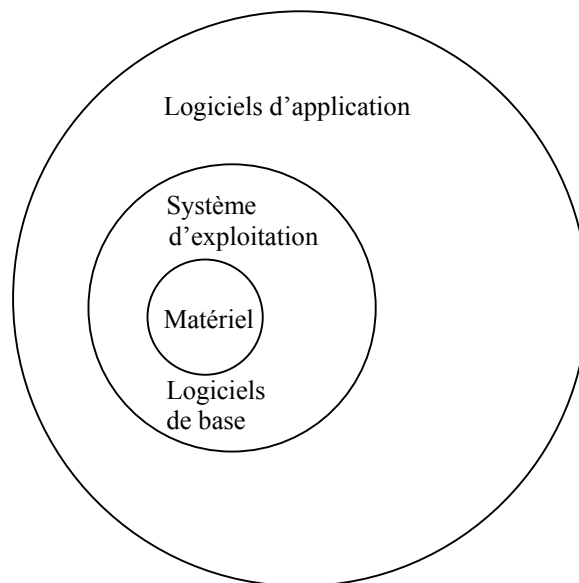
Le *système d'exploitation (Operating System)* est un programme, livré avec l'ordinateur, qui a pour fonction de rendre le matériel « exploitable ». Si l'ordinateur est capable de lire et comprendre ce qu'on lui demande à partir d'un clavier ou en cliquant sur une souris, s'il est capable d'afficher un message d'erreur en clair sur un écran, s'il est capable d'archiver une information dans une mémoire de masse et de la retrouver, ... , c'est qu'il y a un système d'exploitation qui automatise ces tâches sans que l'utilisateur ait à connaître l'architecture détaillée de l'ordinateur ni à dialoguer en langage électronique avec la machine. La gestion de la mémoire, et en particulier de la mémoire de masse, est l'une des principales tâches dont est chargé le système d'exploitation.

Deux des systèmes d'exploitation les plus courants sont : WINDOWS[®], de Microsoft[®], et UNIX[®] ou LINUX[®].

L'architecture logicielle d'un ordinateur est une architecture en couches :

- au plus proche du matériel : le système d'exploitation

- les logiciels de base : éditeurs de texte, compilateurs, bibliothèques d'utilitaires, ...
- au plus loin du matériel : les logiciels d'application (exemples : logiciel de comptabilité, logiciel d'analyse automatique d'images biologiques, ...)



2- INTRODUCTION A LA CONCEPTION DE PROGRAMMES

2.1- Les langages de programmation

On distingue classiquement 3 niveaux de langage. Du moins concis et moins évolué au plus concis et plus évolué :

- les langages machine (ou langages binaires). Ce sont des codes binaires directement interprétables par l'électronique du processeur. En définitive, tout programme finit par devoir être traduit en langage machine pour pouvoir être exécuté par le processeur. Les programmes en langage machine (on dit aussi programmes binaires) ne sont pas portables (i.e. exécutables) directement sur des processeurs d'autres familles.
- les langages assembleurs. Ils offrent des mnémoniques qui permettent de programmer de façon plus facile et plus lisible, mais qui restent néanmoins très facilement traduisibles en langage machine. Les langages assembleurs restent des langages orientés machine : les utiliser nécessite de connaître précisément l'architecture et le fonctionnement du processeur. Ils permettent d'écrire des programmes qui peuvent exploiter au mieux les possibilités techniques de la machine, mais, a contrario, ces programmes ne sont pas portables sur des processeurs d'autres familles.
- les langages évolués. Ils sont plus proches des langages mathématiques ou logiques, plus proches du langage naturel. Ils possèdent un vocabulaire, une grammaire, et le moyen d'étendre le langage en créant de nouveaux mots et en leur associant une sémantique. Ils ne nécessitent pas de connaître l'architecture détaillée de la machine. Offrant un certain degré d'abstraction, ils permettent de développer des programmes plus rapidement, et les programmes écrits sont assez

facilement portables. Les langages évolués sont des langages orientés application : le choix du langage est souvent fonction du domaine d'application du programme.

Exemple de niveaux de langage. Considérons un traitement qui consiste à additionner les valeurs de deux variables Y et Z, ajouter 1, et affecter le résultat à la variable X. Voici ce même traitement écrit dans les trois niveaux de langage :

En langage évolué (C) (1 seule instruction)	En langage assembleur (3 instructions)	En langage machine (3 instructions)
X = Y + Z + 1 ;	add Y, Z, r0 inc r0 move r0, X	00001010 00100101 00110101 00111101 00100011

Tous les langages permettent théoriquement de réaliser n'importe quel projet informatique. Cependant ils ont chacun leur domaine d'application privilégié. Exemples de langages évolués et de domaines d'application typiques :

Cobol	Gestion
Fortran	Calcul numérique
C	Système d'exploitation, logiciel de base
C++	Logiciel de base, grands projets
Java	Serveurs web, applettes, I.H.M., etc.
Lisp, Prolog	Intelligence artificielle

2.2- Les traducteurs de langage

Le processeur ne comprend qu'un seul langage : le langage de l'électronique numérique, le langage binaire. Les seuls programmes exécutables par le processeur sont donc des programmes en langage binaire. Tout programme écrit dans un autre langage de programmation (c'est toujours le cas en pratique) doit donc être préalablement traduit en langage binaire pour pouvoir être exécuté. Le programme écrit dans le langage de programmation est appelé *programme source* ; le programme obtenu après traduction en langage machine est appelé *programme binaire*. Ce travail de traduction est réalisé par un logiciel traducteur :

- un *assembleur* : programme en langage assembleur → programme binaire
- un *compilateur* : programme en langage évolué → programme binaire
- un *interprète* : {traduit & exécute} instruction par instruction