

Chapitre 1: Introduction à l'informatique

1. Définition de l'informatique

Le terme informatique est composé de deux mots: *Information* et *Automatique*. Donc, l'informatique est l'ensemble des sciences et techniques relatives au *traitement automatique* de l'*information* par un ordinateur.

- *Information* : est un élément de connaissance humaine susceptible d'être représentée à l'aide d'un *système de codage* afin d'être conservée, traitée ou communiquée. Ils existent deux types d'informations: les *données* et les *instructions*.
 - ❖ *Donnée* : est une représentation de l'information en vue d'un traitement automatique, tels que les données d'un étudiant : *nom, prénom, âge et adresse*.
 - ❖ *Instruction* : est une forme d'information qui permet de décrire l'action qui devra être réalisée (exécutée) par l'ordinateur.
- *Traitement automatique de l'information* : est une suite d'opérations transformant une représentation de cette information en une autre représentation plus facile à manipuler ou à interpréter.

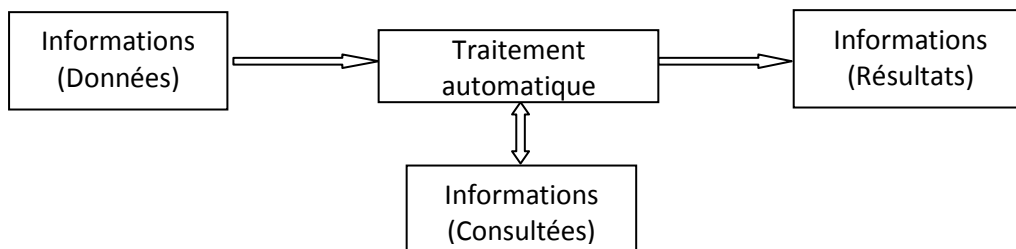


Figure 1. *Traitement d'information*

Exemples:

- Calcul du produit de deux nombres ($3*2=6$).
- Calcul de la moyenne des notes d'un module.
- Traduire des lignes de texte.
- ... etc.

2. Evolution de l'informatique et des ordinateurs

Un ordinateur lancé sur le marché dépasse l'ordinateur précédent. On parle à ce propos de l'évolution des ordinateurs, ou de générations d'ordinateurs. On peut classer les ordinateurs en quatre générations, qui sont:

a. *Première génération : les calculateurs à tube (à lampes)*

L'un d'entre eux, l'ENIAC (1944). Il a les dimensions d'une maison et pèse 30 tonnes et contient 18 000 lampes. Il pouvait réaliser deux additions en 3 microsecondes; mais toutes les 7 ou 8 minutes une lampe tombait en panne, et les ingénieurs passaient un temps fou à la localiser.

b. *Deuxième génération: les calculateurs à transistors*

Mis au point dans les années 50 aux États-Unis, les transistors sont capables d'effectuer le même travail que les lampes. Les transistors sont beaucoup moins chers, plus petits, plus fiables et plus rapides.

c. *Troisième génération: l'apparition des circuits intégrés*

Les circuits intégrés sont des combinaisons de transistors à l'intérieur d'une pastille de silicium appelée une *puce*. Son invention permet l'avènement de la micro-électronique. Ils permettent de construire des ordinateurs plus puissants et plus petits par rapport aux ordinateurs précédents.

d. *Quatrième génération: les microprocesseurs*

Au début des années 1970, tous les composants d'un ordinateur sont rassemblés dans une puce appelée *microprocesseur*. Grâce à l'ultra-miniaturisation des composants, un ancien ordinateur qui est très grand peut aujourd'hui tenir sur un morceau de silicium de quelques millimètres. On peut acheter désormais des ordinateurs de petite taille et pour le prix de quelques-unes des anciennes lampes. Les ordinateurs sont devenus plus petits, plus puissants et moins chers. On parle alors des micro-ordinateurs, utilisables même par les non-spécialistes.

3. Les systèmes de numération et le codage de l'information

3.1. Codage de l'information

- ✓ Les informations traitées par les ordinateurs sont de différentes natures:
 - Nombres, texte;
 - Images, sons, vidéos;
 - Programmes;
 - ... etc.
- ✓ Dans un ordinateur, elles sont toujours représentées sous forme d'une suite de **0** et de **1**. Ces 0 et 1 sont appelés **bits** ou **binary digits** (digits binaires).
- ✓ Un groupe de huit (8) bits est appelé **octet** ou **byte** (*en anglais*).

- ❖ *Le codage de l'information* : permet de passer d'une représentation (dite *externe*) d'une information à une autre représentation (dite *interne* : sous forme de bits) de la même information, suivant un ensemble de règles précises.

Exemple: la représentation externe du nombre trente cinq est **35**. La représentation interne de 35 sera une suite de 0 et 1 (**100011**).

➤ Les étapes de codage de l'information

En informatique, le codage de l'information s'effectue principalement en trois étapes :

1. Expression de l'information par une suite de nombres.
2. Codage des nombres sous forme binaire (suite de 0 et 1)
3. Chaque élément binaire est représenté par un **état physique**:
 - *Charge électrique (RAM : Condensateur-transistor) : Chargé (bit 1) ou non chargé (bit 0).*
 - *Magnétisation (Disque dur, disquette) : polarisation Nord (bit 1) ou Sud (bit 0).*
 - *Fréquences (Modem) : fréquence f_1 (bit 1) ou fréquence f_2 (bit 0).*

3.2. Les systèmes de numération

- ✓ Un système de numération décrit la façon avec laquelle les nombres sont représentés.
- ✓ Il est défini par ensemble de *symboles ou chiffres* (alphabet) et des *règles d'écritures*.
- ✓ Les systèmes de numération les plus utilisés sont le *système décimal*, le *système binaire*, le *système octal* et le *système hexadécimal*.

❖ Base, rang et poids

- ✓ La *base* de numération dans un système de numération correspond au *nombre de symboles ou de chiffres différents utilisés*. Par exemple, dans le système décimal la base est **10**, parce qu'on utilise *10 chiffres différents*, à savoir: **0 à 9**.
- ✓ Le *rang* d'un chiffre est *sa position dans le nombre* (numérotation de droite vers la gauche, en commençant par 0).
- ✓ On détermine le poids (p) d'un chiffre à partir de la base (b) et le rang (r), comme suit:

$$p = b^r$$

3.2.1. Le système décimal (base 10)

Le système décimal est le système le plus courant. On utilise dix chiffres différents pour représenter un nombre, qui sont les chiffres (**0** à **9**). Donc, la base b d'un système décimal est égale à **10**.

Exemple: soit le nombre **8357** en décimal (la base $b = 10$):

chiffre	8	3	5	7
rang	3	2	1	0
poids	10^3	10^2	10^1	10^0

Poids fort poids faible

Table 1

3.2.2. Le système binaire (base 2)

Le système binaire est le système utilisé par les ordinateurs. On utilise deux chiffres, qui sont **0** et **1**.

Chaque chiffre est appelé **bit**. Donc, la base b d'un système binaire est égale à **2**.

Exemple: soit le nombre **1101** en binaire (la base $b = 2$).

chiffre	1	1	0	1
rang	3	2	1	0
poids	2^3	2^2	2^1	2^0

Poids fort poids faible

Table 2

3.2.3. Le système octal (base 8)

Le système octal utilise les chiffres de **0** à **7**. Donc, la base b d'un système octal est égale à **8**. Chaque **3 bits** en base 2 sont codés par un seul chiffre en base 8, comme suit:

Chiffre octal	0	1	2	3	4	5	6	7
Code binaire	000	001	010	011	100	101	110	111

Table 3 : Correspondances octal-binaire.

Exemple: soit le nombre **5613** en octal (la base $b = 8$).

chiffre	5	6	1	3
rang	3	2	1	0
poids	8^3	8^2	8^1	8^0

Poids fort poids faible

Table 4

3.2.4. Le système hexadécimal (base 16)

Le système hexadécimal utilise les chiffres de **0** à **9** et les lettres de **A** à **F**. Donc, la base b d'un système hexadécimal est égale à **16**.

Les lettres de **A** à **F** correspondent, respectivement, aux nombres de **10** à **15**.

Chiffre hexadécimal	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
Code décimal	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

Table 5 : Correspondances hexadécimal-décimal.

Exemple: soit le nombre **F9E7** en hexadécimal (la base $b = 16$).

chiffre	F	9	E	7
rang	3	2	1	0
poids	16^3	16^2	16^1	16^0

Poids fort poids faible

Table 6

3.2.5. Le changement de base (transcodage)

Le changement de base est l'opération qui permet de passer de la représentation d'un nombre exprimé dans une base à la représentation du même nombre exprimé dans une autre base.

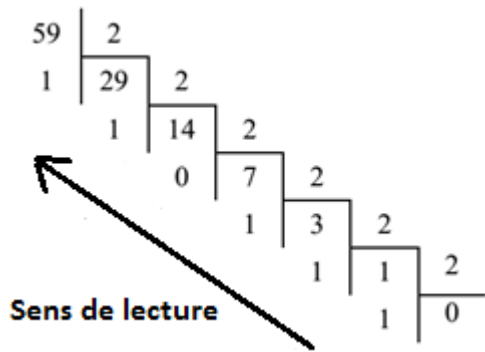
➤ Conversion d'un nombre en base 10 vers une base b

Pour convertir un nombre de la base 10 vers une base b quelconque, on suit la règle des divisions successives suivante:

- On divise le **nombre** par la base b ;
- puis le **quotient** par la base b ;
- ainsi de suite jusqu'à l'obtention d'un **quotient nul** (égale à 0);
- la suite des restes correspond aux chiffres du nombre en base b ;
- on obtient en premier le chiffre de poids faible et en dernier le chiffre de poids fort.

❖ Exemple: décimal vers binaire

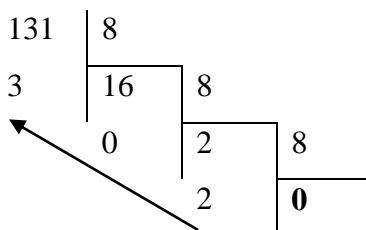
Conversion du nombre 59 écrit en base 10 vers la base 2



Résultat: $(59)_{10} = (111011)_2$

❖ **Exemple:** décimal vers octal

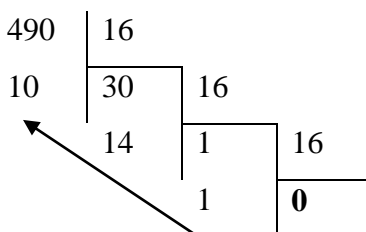
Conversion du nombre 131 écrit en base 10 vers la base 8



Résultat: $(131)_{10} = (203)_8$

❖ **Exemple:** décimal vers hexadécimal

Conversion du nombre 490 écrit en base 10 vers la base 16



Résultat: $(490)_{10} = (1EA)_{16}$

➤ **Conversion d'un nombre en base b vers la base 10**

Pour convertir un nombre N exprimé dans une base b quelconque vers la base 10, on applique la règle suivante:

$$(N)_{10} = \sum_{i=0}^{n-1} a_i * b^i$$

Ou bien,

$$N_{(10)} = a_0 * b^0 + a_1 * b^1 + a_2 * b^2 + \dots + a_{n-1} * b^{n-1}$$

- b est la base du nombre à convertir;
- n est le nombre de symboles du nombre exprimé en base b ;
- a_i est le chiffre ou le bit du rang i du nombre exprimé en base b ;

❖ **Exemple:** binaire vers décimal

Soit le nombre $N=(111011)_2$ à convertir vers la base 10.

$$\begin{aligned}(111011)_2 &= 1*2^0 + 1*2^1 + 0*2^2 + 1*2^3 + 1*2^4 + 1*2^5 \\ &= 1 + 2 + 0 + 8 + 16 + 32 \\ &= (59)_{10}\end{aligned}$$

$$(111011)_2 = (59)_{10}$$

❖ **Exemple:** octal vers décimal

Soit le nombre $N = (253)_8$ à convertir vers la base 10.

$$\begin{aligned}(253)_8 &= 3*8^0 + 5*8^1 + 2*8^2 \\ &= 3 + 40 + 128 \\ &= (171)_{10}\end{aligned}$$

$$(253)_8 = (171)_{10}$$

❖ **Exemple:** hexadécimal vers décimal

Soit le nombre $N = (A9C4)_{16}$ à convertir vers la base 10.

$$\begin{aligned}(A9C4)_{16} &= 4*16^0 + 12*16^1 + 9*16^2 + 10*16^3 \\ &= 4 + 192 + 2304 + 40960 \\ &= (43460)_{10}\end{aligned}$$

$$(A9C4)_{16} = (43460)_{10}$$

➤ **Conversion d'un nombre en base b_1 vers une base b_2**

Pour faire la conversion d'un nombre écrit en base b_1 ($b_1 \neq 10$) vers une base b_2 ($b_2 \neq 10$), on fait comme suit:

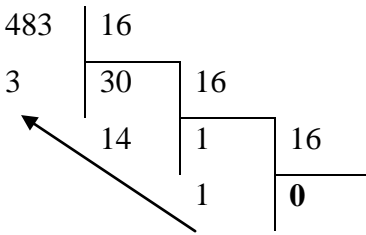
- Conversion du nombre en base b_1 vers la base 10.
- Conversion du nombre en base 10 vers la base b_2 .

❖ **Exemple:** $(743)_8 = (?)_{16}$

$$(743)_8 = (?)_{10}$$

$$(743)_8 = 3*8^0 + 4*8^1 + 7*8^2 = (483)_{10}$$

$(483)_{10} = (?)_{16}$



Résultat: $(743)_8 = (1E3)_{16}$

4. Principe de fonctionnement d'un ordinateur

Un ordinateur est constitué de deux parties, qui sont le **matériel**, référencé par le terme anglais **Hardware**, qui représente les éléments physiques de l'ordinateur, et le **logiciel** référencé par le terme **Software**.

4.1. Partie matérielle d'un ordinateur

La partie matérielle d'un ordinateur est composée de l'*unité centrale* et des *périphériques*.

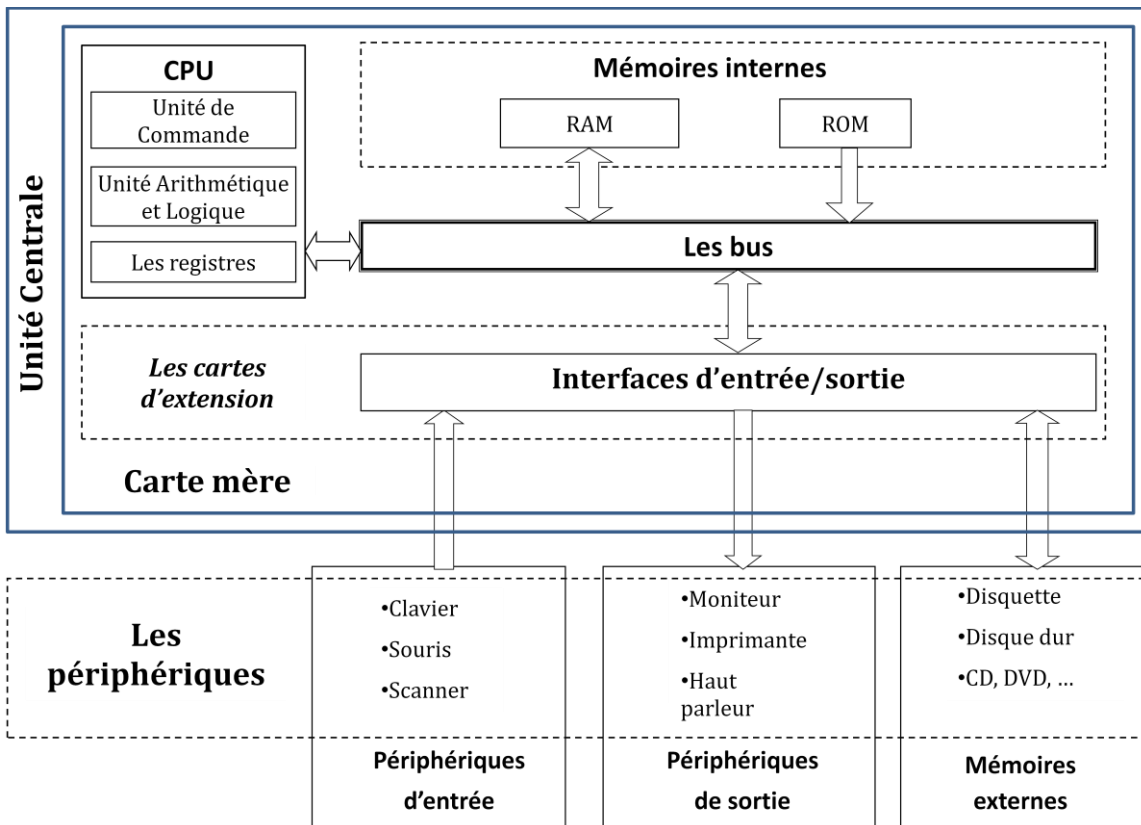


Figure 2. Composants matériels d'un ordinateur

A. L'unité centrale:

➤ **L'unité de traitement centrale (*Central Processing Unit*) CPU** : l'unité de traitement centrale ou le *microprocesseur* est:

- le cerveau de l'ordinateur;
- une puce électronique qui comporte des millions de transistors;
- capable d'interpréter et d'exécuter des instructions stockées en mémoire;
- capable de coordonner, contrôler et vérifier tous ce qui se passe dans la machine.
- Elle contient *l'Unité de Contrôle et Commande*, *l'Unité Arithmétique et Logique* et des *registres*.

✓ **Unité Arithmétique et Logique (UAL)**: c'est le composant central qui permet de réaliser tous les calculs. Elle permet d'effectuer:

- les opérations arithmétiques (addition, multiplication, division, soustraction) ;
- les opérations logiques de base (ET logique, OU logique, NON logique).

✓ **Unité de Commande et de Contrôle (UCC)**: elle est chargée de commander et de gérer tous les différents constituants de l'ordinateur, c.à.d:

- contrôler les échanges;
- chercher et décoder des instructions rangées en mémoire;
- gérer l'enchaînement des différentes instructions;
- ... etc.

✓ **Les registres**: petites mémoires très rapides qui stockent localement les informations traitées par le microprocesseur.

*Le microprocesseur est caractérisé par sa fréquence (vitesse) **en Hz** (nombre d'opérations que le microprocesseur peut effectuer en seconde), et la quantité de mémoire cache contenue dans le processeur et qui augmente ses performances.*

➤ **Mémoires internes**: on trouve principalement deux types de mémoires internes, qui sont: les *mémoires vives* (RAM) et les *mémoires mortes* (ROM). Elles sont moins rapides par rapport aux registres.

✓ **La mémoire vive (*Random Access Memory*) RAM**: est une mémoire:

- qui permet la lecture et l'écriture;
- sert à stocker **temporairement** les informations;
- **volatile** (perd son contenu dès qu'elle est hors tension);

✓ **La mémoire morte (Read Only Memory) ROM:** est une mémoire:

- permet uniquement la **lecture**;
- les informations qui sont contenue dans cette mémoire y sont écrites par le constructeur de l'ordinateur.

La mémoire est caractérisée par le temps d'accès (l'intervalle de temps entre la demande de lecture/écriture et la disponibilité de la donnée) et par ça capacité mesurée en octet.

1 Octet (O) = 1 Byte = 8 Bits.
1 Kilo-octet (Ko) = 1024 O = 2^{10} octets
1 Méga-octet (Mo) = 1024 Ko = 2^{20} octets
1 Giga-octet (Go) = 1024 Mo = 2^{30} octets
1 Téra- octet (To) = 1024 Go = 2^{40} octets

- **Les cartes d'extension:** ce sont des cartes offrant de nouvelles fonctionnalités ou de meilleures performances à l'ordinateur. Ils existent plusieurs types de cartes d'extension, comme la *carte graphique*, la *carte son*, la *carte réseau*, la *carte satellite*, ... etc.
- **La carte mère:** la carte mère est l'élément central d'un ordinateur.
 - elle assure l'interconnexion entre tous les autres composants.
 - elle comporte notamment des *emplacements* (ou « slots ») pour accepter le processeur et la RAM),
 - et des *interfaces* (ports) de connexion pour connecter les périphériques et les cartes d'extension.
- **Le bus:** il permet le transport des informations entre différents composants d'un ordinateur. Il peut, par exemple, servir à relier le microprocesseur à la mémoire centrale ou aux périphériques. Il est constitué d'un ensemble de *fils* (i.e. une nappe) ou de « *pistes* » sur la carte mère.
- **Interfaces (ports) de connexion :** elles servent à connecter les périphériques et les cartes d'extension à la carte mère. Ils existent des *ports internes* et des *ports externes*.
 - ✓ **Les ports internes :**
 - *PCI* et *PCI Express* pour connecter toutes les cartes d'extension;
 - *AGP* pour connecter les cartes graphiques;
 - *IDE* et *SATA* pour connecter les lecteurs de disques durs et lecteurs CD/DVD;
 - *Floppy* pour connecter les lecteurs de disquettes.
 - ✓ **Les ports externes :**
 - *PS2* pour connecter les claviers et les souris;
 - *VGA* pour connecter les écrans, les data show ...;
 - *HDMI* pour connecter les écrans à haute résolution;

- *Série* pour connecter des appareils de mesure, les routeurs, ... ;
- *Parallèle* pour connecter des vieilles imprimantes;
- *LAN* ou (*RJ45*) pour connecter le câble réseau;
- *eSATA* pour connecter les disques durs externes;
- *USB* sert à connecter un nombre générique de périphériques.

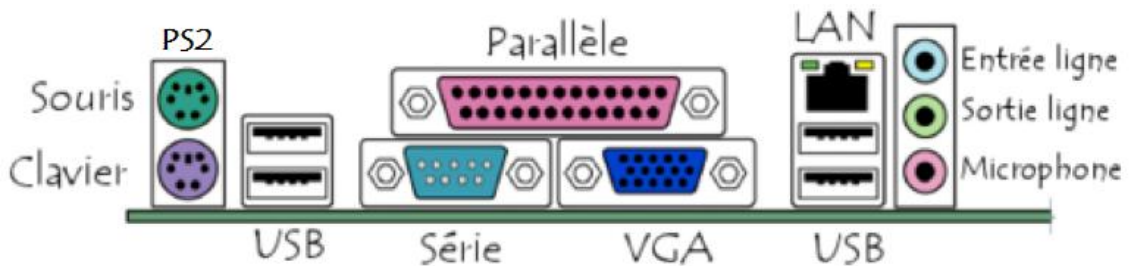


Figure 2. Ports externes d'un ordinateur

B. Les périphériques:

On désigne sous le terme de périphérique tout élément situé à la périphérie de l'ordinateur qui n'est pas nécessaire à son fonctionnement mais dont l'utilisateur a besoin pour communiquer avec l'ordinateur.

- ✓ **Les périphériques d'entrée** : permettent le transfert d'informations de l'extérieur vers l'ordinateur (Clavier, Souris, scanner, microphone, ... etc.).
- ✓ **Les Périphériques de sortie** : permettent le transfert d'informations de l'ordinateur vers l'extérieur (Ecran, imprimante, haut-parleur, ... etc.).
- ✓ **Les Périphériques d'entrée / sortie** : permettent à l'ordinateur d'échanger les informations dans les deux sens (lecteur de disquette, modem, graveur, ... etc.).
- ✓ **Les mémoires externes (périphériques de stockage)** : les mémoires externes appelées *mémoire de masse* ou *mémoires auxiliaires* sont des supports qui peuvent stocker d'une manière permanente les informations. Contrairement à la mémoire centrale, les mémoires secondaires sont lentes. (**Exemples** : le disque dur, la disquette, la clé USB, le CD-ROM, ... etc.).

4.2. Partie système (logiciel, software)

- ✓ La partie système est l'ensemble des *programmes* qui permettent aux utilisateurs de travailler avec un ordinateur.
- ✓ Un programme est un *enchaînement d'instructions*, écrit dans un *langage de programmation*, exécuté par un ordinateur, permettant de traiter un problème et de renvoyer des résultats.

On peut classer les logiciels en trois classes: les *systèmes d'exploitation*, les *logiciels d'application* et les *langages de programmation*.

A. Systèmes d'exploitation

Un système d'exploitation (OS: Operating System) sert à *exploiter* ou faire fonctionner une machine. C'est un ensemble de programmes permettant:

- La *liaison entre les éléments matériels* (unité centrale et périphériques) et les *logiciels d'application*. Donc, c'est grâce à l'OS que les logiciels d'application peuvent accéder aux ressources matérielles.
- La communication entre l'utilisateur et la machine, via une interface *graphique* ou *textuelle*.

Il existe plusieurs systèmes d'exploitation : *Windows, MS-DOS, Mac OS, Linux, Android, iOS, ...etc.*

❖ Composants d'un système d'exploitation

Un système d'exploitation est composé :

- d'un noyau (*kernel*);
- de bibliothèques dynamiques ;
- d'un ensemble d'outils système ;
- de programmes applicatifs de base.

i. **Le noyau** : représente les fonctions fondamentales du système d'exploitation:

- Gestion des processus: *création, suppression, ordonnancement et communication*.
- Gestion de la mémoire: *partage de la mémoire centrale entre les processus*.
- Gestion des E/S : *gestion de tous les périphériques (clavier, écran, disques, imprimantes, etc.)*;
- Gestion des fichiers: *création, modification, suppression et organisation des fichiers et des répertoires*.

ii. **Les bibliothèques dynamiques** : regroupent les opérations les plus utilisées dans les programmes informatiques, pour éviter la redondance de ces opérations dans tous les programmes. Certains systèmes ne proposent pas de bibliothèques dynamiques.

iii. **Les outils système** : ensembles des outils (programmes) qui permettent la configuration, l'administration et la maintenance du le système (*sauvegarde et restauration du système, défragmentation du disque, réparation des fichiers endommagés, identifier les problèmes, ... etc.*).

iv. **Les programmes applicatifs de base** : offrent des services à l'utilisateur (*calculatrice, éditeur de texte, navigateur web, ... etc.*).

B. Logiciels d'application

Les logiciels d'application sont choisis par l'utilisateur pour répondre à ses propres besoins ou faire des traitements particuliers. Parmi les logiciels d'application, on peut citer :

- Les logiciels de traitement de texte (saisie du texte, mise en forme, mise en page, correction orthographique et grammaticale, etc.). **Ex.** *Microsoft Word, Wordperfect, ... etc.*
- Les tableurs (manipulation de feuilles de calcul, production de tableaux, de graphiques, etc.), **Ex.** *Excel, Lotus, ... etc.*
- Les logiciels de présentation. **Ex.** *PowerPoint, ... etc.*
- Les systèmes de gestion de bases de données SGBD. **Ex.** *Access, Oracle, ... etc.*
- Les logiciels de dessin. **Ex.** *Paint, Autocad, ... etc.*
- Les logiciels de messagerie et de communication via un réseau ou via internet. **Ex.** *Outlook Express, Skype, ... etc.*

C. Langages de programmation

- ✓ En informatique, un langage de programmation sert à produire ou bien à développer des applications.
- ✓ Un langage de programmation, dit de *haut niveau*, est facilement compréhensible par un être humain, mais pas par la machine. **Ex.** Java, C, C++, Pascal, Fortran, PHP, HTML, ... etc.
- ✓ Un programme écrit dans un langage de programmation de haut-niveau est dit *code source*.
- ✓ Un ordinateur (processeur) ne comprend que le *langage machine* ou *code machine*, qui est une suite de bits.
- ✓ Donc, un programme écrit dans un langage de haut niveau doit être traduit en langage machine à l'aide d'un *compilateur* ou d'un *interpréteur* pour être exécuté par l'ordinateur (processeur).