

INTRODUCTION A L'INFORMATIQUE

1-Définition de l'informatique

Le terme « informatique » date de 1962. Il vient de la contraction des mots « information » et « automatique ». L'informatique est la science du traitement automatique de l'information. L'histoire de l'informatique est justement marquée par la volonté des hommes d'automatiser certaines tâches longtemps réalisées à la main, en particulier le calcul

Voici quelques domaines d'application de l'informatique : L'enseignement, Le commerce, La médecine L'agriculture, ...etc

2-Evolution de l'informatique et des ordinateurs

C'est en 1642 que le philosophe et mathématicien Blaise Pascal construit la première machine à calculer (la Pascaline, aussi appelée roue Pascal), capable d'effectuer des additions et des soustractions.

Un peu plus tard, en 1833, l'Anglais Charles Babbage reprend ce principe et construit une machine encore plus élaborée que les machines à calculer de l'époque : la sienne est capable d'exécuter toutes les opérations et de stocker les résultats. C'est à son associée, la mathématicienne Ada Byron, que l'on doit un peu plus tard les principes de base de la programmation.

Selon les langages de programmation et la technologie utilisée, l'histoire de l'informatique est classée par génération :

La première génération (1946)

Construction du premier calculateur électronique ENIAC, utilisant le langage de programmation binaire.

La deuxième génération (1956)

Premier calculateur transistorisé TRADIC, apparition des langages de programmation Assembleur, Algol, Cobol.

La troisième génération (1964)

Premier ordinateur à circuits intégrés (IBM360), langages de programmation Basic, Pascal, C.

La quatrième génération (1976)

Naissance des micro-processeurs et des micro-ordinateurs de bureau, logiciels de bureautique.

□ **La cinquième génération (1984)**

Naissance des micro-ordinateurs familiaux et apparition des systèmes d'exploitations (MS_DOS, MacOS).

□ **La sixième génération (1992)**

Amélioration des fréquences des processeurs 33 MHZ, RAM 4Mo. Diffusion de Windows à grande échelle et exploration d'Internet par le grand public.

□ **Génération actuelle**

Apparition des ordinateurs portables puissants, réseaux sans fil,...

3- Les systèmes de codage des informations

Quelle que soit la nature de l'information traitée par un ordinateur (image, son, texte, vidéo), elle l'est toujours sous la forme d'un ensemble de nombres écrits en base 2, par exemple 01001011. Le terme bit (b minuscule dans les notations) signifie « binary digit », c'est-à-dire 0 ou 1 en numérotation binaire. Il s'agit de la plus petite unité d'information manipulable par une machine numérique. Il est possible de représenter physiquement cette information binaire par un signal électrique ou magnétique, qui, au-delà d'un certain seuil, correspond à la valeur 1.

L'octet (en anglais byte ou B majuscule dans les notations) est une unité d'information composée de 8 bits. Il permet par exemple de stocker un caractère comme une lettre ou un chiffre.

Une unité d'information composée de 16 bits est généralement appelée mot (en anglais word). Une unité d'information de 32 bits de longueur est appelée mot double (en anglais double word, d'où l'appellation dword). Beaucoup d'informaticiens ont appris que 1 kilooctet valait 1024 octets.

. Voici les unités standardisées :

- Un kilooctet (ko) = 2^{10} octets = 1024 octets
- Un mégaoctet (Mo) = 2^{20} octets = 1024 koctets
- Un gigaoctet (Go) = 2^{30} octets = 1024 Moctets
- Un téraoctet (To) = 2^{40} octets = 1024 Goctets
- Un pétaoctet (Po) = 2^{50} octets = 1024 Toctets

Pour qu'une information numérique soit traitée par un circuit, elle doit être mise sous forme adaptée à celui-ci. Pour cela Il faut choisir un système de numération, de nombreux systèmes de numération sont utilisés en technologie numérique ; tels que les systèmes : Décimal (base 10), Binaire (base 2), Tétral (base 4), Octal (base 8) et Hexadécimal (base 16).

Dans cette partie de cours on va étudier les systèmes décimal, binaire et hexadécimal ainsi que les conversions entre ces différentes bases de codage.

3-1 Le système décimal

Les nombres que nous utilisons habituellement sont ceux de la base 10 (système décimal). Nous disposons de dix chiffres différents de 0 à 9 pour écrire tous les nombres. D'une manière générale, **toute base N est composée de N chiffre de 0 à N-1.**

Soit un nombre décimal $N = 2348$. Ce nombre est la somme de 8 unités, 4 dizaines, 3 centaines et 2 milliers. Nous pouvons écrire

$$N = (2 \times 1000) + (3 \times 100) + (4 \times 10) + (8 \times 1)$$

$$\mathbf{2348 = (2 \times 10^3) + (3 \times 10^2) + (4 \times 10^1) + (8 \times 10^0)}$$

10 représente la base et les puissances de 0 à 3 le rang de chaque chiffre. Quelque soit la base, le chiffre de droite est celui des unités. Celui de gauche est celui qui a le poids le plus élevé. Cette écriture s'appelle **forme polynomiale.**

3-2 Le système binaire

Dans les domaines de l'automatisme, de l'électronique et de l'informatique, nous utilisons la base 2. Tous les nombres s'écrivent avec deux chiffres uniquement (0 et 1). De même que nous utilisons le système décimal parce que nous avons commencé à compter avec nos dix doigts, nous utilisons le binaire car les systèmes technologiques ont souvent deux états stables.

Un interrupteur est ouvert ou fermé, une diode est allumée ou éteinte, une tension est présente ou absente, une surface est réfléchissante ou pas (CD), un champ magnétique est orienté Nord-Sud ou Sud-Nord (disque dur). A chaque état du système technologique, on associe un état logique binaire.

La présence d'une tension sera par exemple notée 1 et l'absence 0. Le chiffre binaire qui peut prendre ces deux états est nommé "Bit" (Binary digit).

Avec un bit nous pouvons coder deux états

0
1

Avec deux bits nous pouvons coder quatre états

0	0
0	1
1	0
1	1

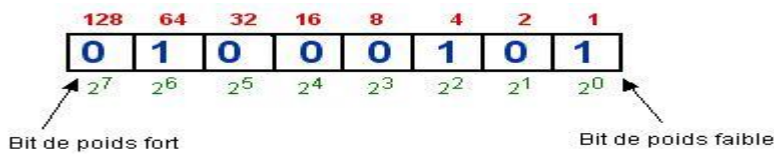
Avec trois bits nous pouvons coder huit états

0	0	0
0	0	1
0	1	0
0	1	1
1	0	0
1	0	1
1	1	0
1	1	1

A chaque nouveau bit, le nombre de combinaisons possibles est doublé. Ce nombre est égal à **2 puissance N** (N étant le nombre de bits).

Un groupe de bits est appelé un mot, un mot de huit bits est nommé un octet (byte). Avec un octet, nous pouvons écrire 2 puissance 8 = **256 nombres binaires de 0 à 255**

Description d'un octet.



Un 1 dans une case représente la valeur décimale qui est au dessus.

Correspondance entre binaire et décimal.

Conversion d'un nombre binaire en décimal.

Il suffit de faire la somme des poids de chaque bit à 1, c'est-à-dire écrire le nombre sous forme polynomiale. Le nombre ci dessus est égal à

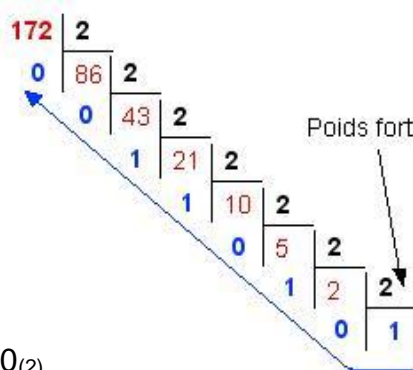
$$(01000101)_2 = (1 \times 2^0 + 1 \times 2^2 + 1 \times 2^6)_{10} = (69)_{10}$$

$$(111011)_2 = 1 \times 2^5 + 1 \times 2^4 + 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 = (61)_{10}$$

$$(10011.1101)_2 = 1 \times 2^4 + 0 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 + 1 \times 2^{-1} + 1 \times 2^{-2} + 0 \times 2^{-3} + 1 \times 2^{-4} = (19.75)_{10}$$

Conversion d'un nombre décimal en binaire (exemple : N = 172).

Méthode par divisions



$$172_{(10)} = 10101100_{(2)}$$

Conversion d'un nombre décimal à virgule

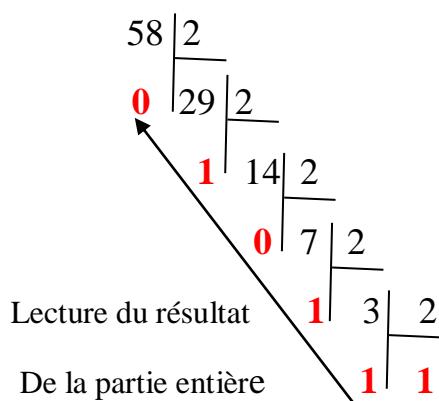
Pour convertir un nombre décimal à virgule dans une base B quelconque, il faut :

1-Convertir la partie entière en effectuant des divisions successives par **2** (comme nous l'avons vu précédemment).

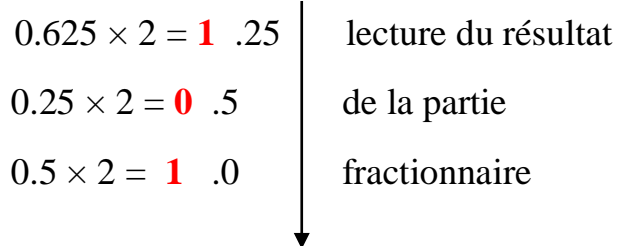
2-Convertir la partie fractionnaire en effectuant des multiplications successives par **2** et en conservant à chaque fois le chiffre devenant entier.

Exemple : Conversion du nombre (58,625) en base 2

Conversion de la partie entière



Conversion de la partie fractionnaire



(58.625)₁₀ = (111010.101)₂

Parfois en multipliant la partie fractionnaire par la base 2 ou n'importe quelle base B on n'arrive pas à convertir toute la partie fractionnaire. Ceci est dû essentiellement au fait que le nombre à convertir n'a pas un équivalent exacte dans la base B et sa partie fractionnaire est cyclique ou infinie.

Exemple : (0.15)₁₀=(?)₂

0.15 × 2 = 0 .3

0.3 × 2 = 0 .6

0.6 × 2 = 1 .2

0.2 × 2 = 0 .4

0.4 × 2 = 0 .8

0.8 × 2 = 1 .6

0.6 × 2 = 1 .2

0.2 × 2 = 0 .4

0.4 × 2 = 0 .8

0.8 × 2 = 1 .6

$$(0.15)_{10} = (0.00\underline{1001} \underline{1001})_2$$

On dit que le nombre $(0.15)_{10}$ est cyclique dans la base 2 de période **1001**.

3-3 Le système quaternaire (base 4) :

Ce système appelé aussi base 4 comprend quatre chiffres possibles {0, 1, 2, 3}.

Un nombre quaternaire peut s'écrire sous la forme polynomiale comme le montre les exemples suivant :

$$(2331)_4 = 2 \times 4^3 + 3 \times 4^2 + 3 \times 4^1 + 1 \times 4^0 = (189)_{10}$$

$$(130.21)_4 = 1 \times 4^2 + 3 \times 4^1 + 1 \times 4^0 + 2 \times 4^{-1} + 1 \times 4^{-2} = (29.5625)_{10}$$

Conversion d'un nombre décimal en base 4

$$(105.125)_{10} = (?)_4$$

<p>Partie entière</p> <div style="margin-left: 20px;"> $\begin{array}{r} 105 \quad \quad 4 \\ \hline (1) \quad 26 \quad \quad 4 \\ \hline (2) \quad 6 \quad \quad 4 \\ \hline (2) \quad \quad (1) \end{array}$ </div> <p>lecture du résultat</p>	<p>partie fractionnaire</p> <div style="margin-left: 20px;"> $\begin{array}{l} 0.125 \times 4 = (0) .5 \\ 0.5 \times 4 = (2) .0 \end{array}$ </div> <p>lecture du résultat</p>	\downarrow
$(105.125)_{10} = (1221.02)_4$		

3-4 Le système octal (base 8) :

Le système octal ou base 8 comprend huit chiffres qui sont {0, 1, 2, 3, 4, 5, 6, 7}. Les chiffres 8 et 9 n'existent pas dans cette base. Ecrivons à titre d'exemple, les nombres 45278 et 1274.6328 :

$$(4527)_8 = 4 \times 8^3 + 5 \times 8^2 + 2 \times 8^1 + 7 \times 8^0 = (2391)_{10}$$

$$(1274.632)_8 = 1 \times 8^3 + 2 \times 8^2 + 7 \times 8^1 + 4 \times 8^0 + 6 \times 8^{-1} + 3 \times 8^{-2} + 2 \times 8^{-3} = (700.8007813)_{10}$$

Conversion d'un nombre décimal en base 4

$(110.625)_{10} = (?)_8$

<p>Partie entière</p> $ \begin{array}{r l} 110 & 8 \\ \hline 6 & 13 \\ & \\ & 5 & 8 \\ & \\ & & 1 \end{array} $	<p>partie fractionnaire</p> $0.625 \times 8 = \textcircled{5}.0$
$(110.625)_{10} = (156.5)_8$	

3-5 Le système hexadécimal

La manipulation des nombres écrits en binaire est difficile pour l'être humain et la conversion en décimal n'est pas simple. On utilise aussi très souvent *le système hexadécimal* (base 16) du fait de sa simplicité d'utilisation et de représentation pour les mots machines (il est bien plus simple d'utilisation que le binaire).

Pour écrire les nombres en base 16 nous devons disposer de 16 chiffres, pour les dix premiers, nous utilisons les chiffres de la base 10, pour les suivant nous utiliserons des lettres de l'alphabet.

Décimal	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓
Hexadécimal	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F

Les règles sont ici aussi les mêmes que pour le décimal.

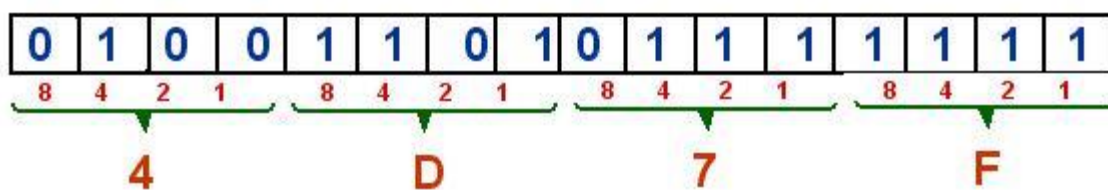
$$\begin{aligned}
 A3F_{(16)} &= (A \times 16^2) + (3 \times 16^1) + (F \times 16^0) \\
 A3F_{(16)} &= (10 \times 256) + (3 \times 16) + (15 \times 1) \\
 A3F_{(16)} &= 2560 + 48 + 15 = 2623_{(10)}
 \end{aligned}$$

Correspondance entre binaire et hexadécimal.

La conversion du binaire en hexadécimal est très simple, c'est d'ailleurs la raison pour laquelle nous utilisons cette base.

Il suffit de faire correspondre un mot de quatre bits (quartet) à chaque chiffre hexadécimal.

Conversion d'un mot de 16 bits entre binaire et hexadécimal

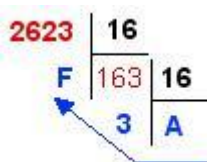


)

$$4D7F_{(16)} = 0100110101111111_{(2)}$$

Correspondance entre décimal et hexadécimal.

La méthodes par divisions s'applique comme en binaire (exemple : N = 2623).



$$2623_{(10)} = A3F_{(16)}$$

En général pour faire La conversion d'un nombre d'une base quelconque B1 vers une autre base B2 il faut passer par la base 10. Mais si la base B1 et B2 s'écrivent respectivement sous la forme d'une puissance de 2 on peut passer par la base 2 (binaire) :

Base quaternaire (base 4) : $4=2^2$ chaque chiffre quaternaire se convertit tout seul sur 2 bits.

Base octale (base 8) : $8=2^3$ chaque chiffre octal se convertit tout seul sur 3 bits.

Base hexadécimale (base 16) : $16=2^4$ chaque chiffre hexadécimal se convertit tout seul sur 4 bits.

Exemples :

$$(1\ 0\ 2\ 2\ 3)_4 = (01\ 00\ 10\ 10\ 11)_2$$

$$(6\ 5\ 3\ 0.7)_8 = (110\ 101\ 011\ 000.111)_2$$

$$(101\ 010\ 100\ 111\ 000)_2 = (5\ 2\ 4\ 7\ 0)_8$$

$$(1101\ 1000\ 1011\ 0110.011)_2 = (D\ 8\ B\ 6.6)_{16}$$

$$(11\ 10\ 01\ 00\ 10.101)_2 = (3\ 2\ 1\ 0\ 2.22)_4$$

3-6 Les nombres signés

Nous avons jusqu'à présent parlé de nombres entiers naturels. Ils ne peuvent par nature qu'être positifs ou nuls. Envisageons maintenant les nombres entiers relatifs ou autrement dit, munis d'un signe '+' ou '-'. En décimal,

+1, +2, +3 etc. sont des nombres positifs. Ils sont supérieurs à 0 ($n > 0$)

-1, -2, -3 etc. sont des nombres négatifs. Ils sont inférieurs à 0 ($n < 0$)

De même en binaire,

+1, +10, +11, +100, +101 etc. sont des nombres binaires positifs,

-1, -10, -11, -100, -101 etc. sont des nombres binaires négatifs.

Le problème est que les circuits électroniques digitaux ne peuvent enregistrer que des 0 ou des 1 mais pas de signes + ou -. Le seul moyen est alors de convenir que si un nombre est susceptible d'être négatif on lui réserve un bit pour indiquer le signe.

Plusieurs méthodes sont utilisées pour représenter les nombres négatifs dans un ordinateur, parmi lesquelles nous citons : la représentation en signe et valeur absolue (SVA), le complément à 1 (CP1) et le complément à 2 (CP2). On cherche une représentation binaire des entiers négatifs pour que l'addition de deux nombres entiers relatifs fonctionne, c'est la représentation complément à 2 (CP2).

Comment calculer les codes des nombres négatifs en CP2 sur 8 bits:

Le calcul se fait en trois étapes :

1° calcul du code binaire du nombre sur 8 bits

2° Calcul du complément à 1 = Remplacer tous les 0 par des 1 et tous les 1 par des 0.

3° Calcul du complément à 2 = Ajouter 1 au complément à 1

Exemples : 1-comment écrire -4 en CP2 ?

$$+4 = 0000\ 0100_{(2)}$$

Le complément à 1 de ce code est CP1(+4)=1111 1011₍₂₎

Ajoutons 1 à ce code pour obtenir son complément à 2

$$1111\ 1011_{(2)} + 1 = 1111\ 1100_{(2)} = (-4)_{CP2}$$

2-comment écrire -24 en CP2 ?

Le nombre $+24 = (11000)_2 = (0001 \mathbf{1000})_{CP2}$

Le complément à 1 de ce code est $CP1(+24) = 1110 0111$

Ajoutons 1 à ce code pour obtenir son complément à 2

$1110 0111_{(2)} + 1 = \mathbf{1}110 1000_{(2)} = (-24)_{CP2}$

ainsi $(-24)_{10} = (1110 \mathbf{1000})_{CP2}$

Le bit le plus à gauche du code CP2 est celui qui va représenter le signe. Signe négatif si ce bit vaut 1, signe positif quand ce bit vaut 0.

Le plus grand nombre signé sur 8 bits est **+127 (01111111)**

Le plus petit nombre signé sur 8 bits est **-128 (10000000)**

-128 à +127 => 256 combinaisons (2 puissance 8)

Ce qui s'applique sur 8 bits s'applique aussi sur 4, 16, ... bits.

Notifications : L'addition binaire :

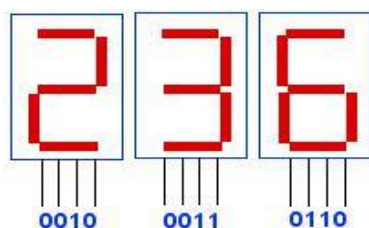
L'addition de deux bits binaire se réalise selon la spécification suivante :

bit1	bit2	résultat	retenue
0	+	0	0
0	+	1	0
1	+	1	0
1	+	0	1

L'addition binaire de deux nombres s'effectue bit à bit de droite à gauche, en reportant les retenues

Le décimal codé binaire (code DCB)

Ce codage est destiné à l'affichage de valeurs décimales, chaque digit doit être codé en binaire sur 4 bits (unités, dizaines, centaines ...).



Ce codage ne permet aucun calcul, il est uniquement destiné à la saisie et à l'affichage de données

Le codage ASCII

Le binaire permet de coder les nombres que les systèmes informatiques peuvent manipuler. Cependant, l'ordinateur doit aussi utiliser des caractères alphanumériques pour mémoriser et transmettre des textes. Pour coder ces caractères, on associe à chacun d'entre eux un code binaire, c'est le codage ASCII (American Standard Code for Information Interchange).

4-Principe de fonctionnement d'un ordinateur

Lorsque l'ordinateur démarre, il commence par lancer un logiciel spécial que l'on appelle l'OS (par exemple Windows, Linux ou MacOS, etc... Sans cet OS, vous ne pouvez pas utiliser l'ordinateur et l'ordinateur ne serait qu'une machine qui ne servirait pas à grand chose. C'est l'OS qui va interpréter vos commandes et vous afficher les résultats.

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.

Voici comment ça marche quand vous lancez un logiciel étape par étape.

1. Vous cliquez (ou double cliquez) sur l'icône.
2. L'ordinateur va alors chercher dans le disque dur vers quel logiciel pointe cette icône.
3. Il va lire le logiciel à partir du disque dur car celui-ci est enregistré dessus.
4. Tout ce qui sera lu sera chargé dans la mémoire vive (RAM).
5. A partir de la RAM, le processeur va exécuter le logiciel instruction après instruction.
6. Une fois que vous fermerez le programme, celui-ci sera effacé de la RAM.
7. Mais bien évidemment, il reste présent sur le disque dur pour une prochaine utilisation.

La RAM, le processeur et le disque dur sont des composants de l'unité centrale de l'ordinateur.

5-Partie matériel d'un ordinateur

On peut citer parmi les composants d'un ordinateur :

- La carte mère
- Le processeur
- La mémoire vive
- Le disque dur
- Le lecteur et/ou graveur de CD-Rom et/ou DVD-Rom

carte mère

Carte principale sur laquelle on retrouve tous les composants nécessaires au fonctionnement d'un ordinateur, dont le processeur, la mémoire centrale, les bus et les connecteurs d'extension destinés à recevoir des cartes d'extension et gère les différentes interfaces avec les ports USB pour les périphériques externes.

Le processeur

Souvent qualifié de cerveau de l'ordinateur, le processeur permet de manipuler et de traiter les données qui lui sont fournies. Sa puissance a une influence sur la vitesse d'exécution de vos logiciels et des opérations que vous effectuez.

La mémoire vive

Partenaire de travail du processeur, la mémoire vive (appelée aussi **RAM**) stocke temporairement les données à traiter par le processeur. Ainsi, plus il y a de mémoire de disponible, plus il est possible d'y conserver des données temporairement (ce qui évite d'accéder au disque dur qui est plus lent).

La mémoire vive est vidée à chaque arrêt ou redémarrage de votre ordinateur.

Le disque dur

Contrairement à la mémoire vive, les données stockées sur un disque dur sont permanentes et ne sont pas effacées à l'arrêt de votre ordinateur. C'est donc sur un disque dur que votre système d'exploitation (Linux, MacOS ou Windows), vos logiciels et vos documents sont conservés. C'est donc un espace de stockage permanent, où vous pouvez copier et supprimer des données à volonté.

Le lecteur et/ou graveur de CD-Rom et/ou DVD-Rom

Définition lecteur :

Appareil permettant la lecture, à l'aide d'un rayon laser, des informations enregistrées sur un CD-ROM ou un DVD-Rom

Les lecteurs de CD-ROM et DVD-Rom peuvent être internes (c'est-à-dire intégrés dans l'ordinateur) ou externes (se présentant sous la forme d'un boîtier autonome).

Définition graveur :

Appareil permettant l'enregistrement de disques compacts, à l'unité ou en petite série, par gravure au laser sur un CD ou DVD

Il y a d'autres composants tels que la carte graphique pour l'affichage sur l'écran, la carte son pour gérer les entrées (microphones) et les sorties (hauts parleurs..), la carte réseau qui permet le transfert des données, entre un ordinateur personnel et un réseau ou entre un serveur et un réseau. Permet également un accès à internet dans un réseau domestique ou d'entreprise.

6-Partie système

Nous avons présenté l'ordinateur comme un ensemble de circuits électroniques interconnectés. Pour établir la communication entre ces éléments matériels et l'homme, considéré comme l'utilisateur, il est nécessaire d'avoir une interface appropriée qui rende possible la communication homme-machine, d'où le concept de système d'exploitation.

Le système d'exploitation (*SE*, en anglais *Operating System* ou *OS*) est un ensemble de programmes responsables de la liaison entre les ressources matérielles d'un ordinateur et les applications de l'utilisateur (traitement de texte, jeu vidéo, ...).

Il existe sur le marché des dizaines de systèmes d'exploitation différents, souvent livrés avec l'appareil informatique. C'est le cas de Windows, Mac OS, Irix, Symbian OS, Linux (pour lequel il existe de nombreuses distributions) ou Android.