

Chapitre I

Introduction à l'Informatique

Sommaire

Chapitre 1 : Introduction à l'Informatique.....	3
I.1. Définition de l'Informatique.....	3
I.2. Evolution de l'informatique et des ordinateurs.....	3
I.3. Les systèmes de codage des informations.....	5
I.3.1. Le codage binaire.....	5
I.3.2. Autres Systèmes de numération.....	5
I.3.3. Conversion d'un nombre d'un système à un autre.....	6
I.4. Principe de fonctionnement d'un ordinateur.....	10
I.4.1. Structure générale d'un ordinateur.....	10
I.4.1.1. L'U.A.L (Unité Arithmétique et Logique : Circuit de Calcul).....	11
I.4.1.2. L'U.C. (Unité de contrôle : Circuit de commandes).....	11
I.4.1.3. La M.C. (La mémoire centrale : circuit de mémorisation).....	11
I.4.1.4. Les mémoires externes o auxiliaires (Support de stockage et de mémorisation)	12
I.4.1.5. Les périphériques d'entrée (ou organes d'entrée).....	12
I.4.1.6. Les périphériques de sortie (ou organes de sortie).....	13
I.4.2. Fonctionnement d'un Ordinateur.....	13
I.5. Partie Matériel d'un Ordinateur.....	14
I.6. Partie Système.....	17
I.6.1. Systèmes d'exploitation.....	18
I.6.2. Langages de programmation.....	19
I.6.3. Programmes d'application.....	19
I.7. Évaluation des expressions arithmétiques et logiques.....	19
I.7.1. Ordre de priorité.....	20
I.7.2. Expressions arithmétiques.....	20
I.7.3. Expressions logiques.....	22
I.7.4. Expressions arithmétiques et logiques en PASCAL.....	23

Cours Elearning :

<https://elearning.univ-bejaia.dz/course/view.php?id=7944>

Page facebook :

<https://www.facebook.com/InitiationAlgoProgrammation/>

La chaîne Youtube :

<https://www.youtube.com/c/AlgoProgrammation1èreAnnéeTechnologie>

Adapté par : Redouane OUZEGGANE
rouzeggane@gmail.com - redouane.ouzegane@univ-bejaia.dz

Chapitre 1 : Introduction à l'Informatique

I.1. Définition de l'Informatique

L'informatique est une branche qui s'occupe du domaine du traitement automatique de l'information. Le mot « INFORMATIQUE » est composé à partir des mots « INFORMATION » et « AUTOMATIQUE ». L'informatique a pour rôle :

- La conception et la construction des ordinateurs,
- Le fonctionnement et la maintenance des ordinateurs,
- Leur exploitation (utilisation des ordinateurs dans les différents domaines d'activités).

I.2. Evolution de l'informatique et des ordinateurs

- En 1643, BLAISE PASCAL invente la "PASCALINE" : machine mécanique capable de réaliser des additions et soustractions.
- En 1812, "CHARLES BABBAGE" a conçu une machine mécanique pouvant effectuer des calculs numériques compliqués (c'est une machine à base de cartes perforées). En 1860, il définit les grands principes des calculateurs électroniques.
- En 1885, "HERMANN HOLLERITH" (inventeur des cartes perforées) construit la première machine à cartes perforées et qui a servi dans l'opération de recensement de la population d'Amérique en 1890.

En 1946 le premier calculateur électronique est apparu. Il est baptisé "ENIAC" (Electronic Numerical Integrator and Computer). Il est construit sur le principe du binaire (0 et 1) : (le courant passe ou ne passe pas). C'est une machine électronique composée de :

- ~ 6000 relais (commutateurs) mécaniques
- ~ 1800 tubes électroniques
- ~ 70000 résistances

Et pèse environ 30 tonnes. Il est capable de réaliser ~5000 additions/seconde et ~3000

multiplications/seconde. C'est le premier ordinateur qui utilise le principe de "Programme enregistré" et constitue la première génération des ordinateurs.

1^{ère} Génération d'ordinateurs (1946 – 1957)

Ceux sont les ordinateurs construits sur la base de "tubes électroniques" comme :

- ENIAC (1946)
- IBM 701 (1952)
- Etc ...

2^{ème} Génération d'ordinateurs (1958 – 1963)

Ils sont apparus après l'invention du "Transistor" composant électronique capable de réguler le courant. Ces ordinateurs sont 100 plus rapides que ceux de la 1^{ère} génération et consomment moins d'énergie électrique et sont moins volumineux (occupent moins d'espaces).

Exemples :

- PDP -I (1^{ère} ordinateur de 2^{ème} génération (1960))
- IBM 7030 (1961)

3^{ème} Génération d'ordinateurs (1964-1971)

Après l'apparition du "Circuit Intégré", ces types d'ordinateurs utilisent les transistors et les circuits intégrés. Exemple de ces ordinateurs :

- IBM 360 (1964)
- DEC PDP-8 (1964)

4^{ème} Génération d'ordinateurs

Après l'apparition du "microprocesseur". Le premier microprocesseur est fabriqué par la société INTEL en 1971. Les ordinateurs (micro-ordinateur) qui sont construits autour de microprocesseur constituent les ordinateurs de la 4^{ème} génération.

Exemple de micro-ordinateurs :

- MICRAL 8008 (1973)

- ALTAIR 8008 (1973)
- APPLE1 (1976)
- IBM PC (1981)
- PENTUIM etc. ...

Actuellement l'évolution des ordinateurs tend à exploiter le laser, les fibres optiques, la biochimie ou ordinateur moléculaire.

I.3. Les systèmes de codage des informations

Toute sorte d'information manipulées par un ordinateurs (numériques, textuelles, images, sons, vidéos, etc.) est représentée par des séquences de deux chiffres : **0** et **1**. Ces deux chiffres sont désignés par *BIT* (*Binary digiT*). Donc un bit est soit **0** ou bien **1** qui est représenté par l'ordinateur par deux états électroniques : soit il y a présence d'une impulsion électrique (c'est l'état **1**), soit il y a absence d'impulsion électrique (c'est l'état **0**).

I.3.1. Le codage binaire

Le système de numération décimale est construite sur la base de 10 chiffres (0, 1, 2, 3, 4, 5, 6, 7, 8, 9) tandis que système de numération binaire est construite que sur la base de 2 chiffres (0, 1).

Un nombre en numération décimale (base 10) se décompose de la façon suivante :

$$(378)_{10} = 8 \times 10^0 + 7 \times 10^1 + 3 \times 10^2$$

De même un nombre en numération binaire (base 2) se décompose comme suit :

$$(101101)_2 = 1 \times 2^0 + 0 \times 2^1 + 1 \times 2^2 + 1 \times 2^3 + 0 \times 2^4 + 1 \times 2^5 = 1 + 4 + 8 + 32 = 45$$

D'une façon générale, un nombre N s'écrit en système binaire comme suit :

$$N = (C_n C_{n-1} C_{n-2} \dots C_2 C_1 C_0)_2 = C_0 \times 2^0 + C_1 \times 2^1 + C_2 \times 2^2 + \dots + C_n \times 2^n = \sum_{i=0}^n C_i \times 2^i \quad / \text{tel-que : } 0 \leq C_i < 2$$

I.3.2. Autres Systèmes de numération

- Système de numération octale (base 8) : est utilise les 8 chiffres : 0, 1, 2, 3, 4, 5, 6, 7
- Système de Numération hexadécimale (base 16) : elle utilise les 16 chiffres et lettres

suivants : 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F

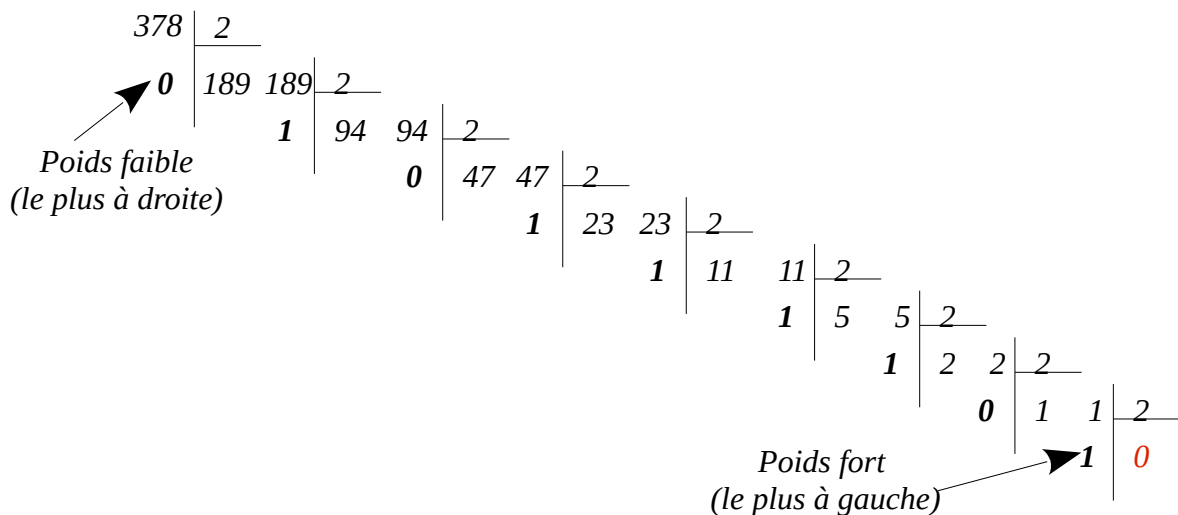
avec :

$$\begin{aligned}
 A &= (10)_{10} = (1010)_2 & B &= (11)_{10} = (1011)_2 & C &= (12)_{10} = (1100) \\
 D &= (13)_{10} = (1101)_2 & E &= (14)_{10} = (1110)_2 & F &= (15)_{10} = (1111)_2
 \end{aligned}$$

I.3.3. Conversion d'un nombre d'un système à un autre

a) Conversion Décimale → Binaire

Cette conversion est obtenu par divisions successives par 2 et on prend les restes dans le sens inverse. (le dernier reste est le poids fort, c'est-à-dire le plus à gauche, et le premier reste est le poids faible, c'est-à-dire le plus à droite). Voir l'exemple suivant :



$$(378)_{10} = (101111010)_2$$

Nous pouvons réaliser les divisions d'une façon linéaire, comme suit :

$$\begin{array}{ll}
 (378)_{10} = (?)_2 & \\
 378 : 2 = 189 & \text{reste } 0 \quad \leftarrow \text{Poids faible (le plus à droite)} \\
 189 : 2 = 94 & \text{reste } 1 \\
 94 : 2 = 47 & \text{reste } 0 \\
 47 : 2 = 23 & \text{reste } 1 \\
 23 : 2 = 11 & \text{reste } 1 \\
 11 : 2 = 5 & \text{reste } 1 \\
 5 : 2 = 2 & \text{reste } 1 \\
 2 : 2 = 1 & \text{reste } 0 \\
 1 : 2 = 0 & \text{reste } 1 \quad \leftarrow \text{Poids fort (le plus à droite)}
 \end{array}$$

b) Conversion Binaire → Décimal

Nous utilisons la formule vue dans la section 1.3.1. Prenons l'exemple suivant :

$$(111101)_2 = (?)_{10}$$

Chaque chiffre possède un poids (la puissance qu'il faut affecter à la base). Le chiffre du poids faible est le chiffre situé le plus à droite, et qui possède le poids **0**. Le chiffre du poids fort est le chiffre situé le plus à gauche, et qui possède le poids (**Nombre de chiffre -1**).

Dans l'écriture suivante, nous représentons les poids des chiffres :

$$\begin{array}{cccccc} \mathbf{5} & \mathbf{4} & \mathbf{3} & \mathbf{2} & \mathbf{1} & \mathbf{0} \\ (1 & 1 & 1 & 1 & 0 & 1)_2 & = & 1 \times 2^0 & + & 0 \times 2^1 & + & 1 \times 2^2 & + & 1 \times 2^3 & + & 1 \times 2^4 & + & 1 \times 2^5 \\ & & & & & & = & 1 & + & 0 & + & 4 & + & 8 & + & 16 & + & 32 \\ & & & & & & = & (61)_{10} & = & 61 \end{array}$$

c) Conversion Octal → Binaire

On code par groupes de trois chiffres binaires, dont chaque chiffre octal est remplacé par trois chiffres binaires, puisque $8 = 2^3$.

Voici le tableau de correspondance entre les chiffres « Octal » et les trois chiffres binaires :

Base 8	Base 2
0	000
1	001
2	010
3	011
4	100
5	101
6	110
7	111

Exemples :

$$(356)_8 = (?)_2$$

Dans le tableau précédent nous avons les correspondances suivantes :

$$6 \rightarrow 110 \quad 5 \rightarrow 101 \quad 3 \rightarrow 011$$

Donc : $(356)_8 = (011\ 101\ 110)_2$

d) Conversion Binaire → Octal

On code par groupe de 3 digits en procédant de droite vers la gauche. Chaque groupe de 3 chiffres binaires est remplacé par 1 chiffre octal.

$$(10\ 110\ 111)_2 = (?)_8$$

$$\begin{array}{ccc} \underbrace{010} & \underbrace{110} & \underbrace{111} \\ 2 & 6 & 7 \end{array}$$

Donc : $(10\ 110\ 111)_2 = (267)_8$

c) Conversion Hexadécimal → Binaire

On code par groupe de 4 chiffres binaires. Chaque chiffre hexadécimal est remplacé par 4 chiffres binaires. (puisque $16 = 2^4$). Nous utilisons le tableau suivant :

Base 16	Base 2	Base 16	Base 2
0	0000	8	1000
1	0001	9	1001
2	0010	A=10	1010
3	0011	B=11	1011
4	0100	C=12	1100
5	0101	D=13	1101
6	0110	E=14	1110
7	0111	F=15	1111

$$(1A5)_8 = (?)_2$$

En utilisant le tableau ci-dessus, on remplace 5 par 0101, A par 1010 et 1 par 0001, nous aurons :

$$(1A5)_8 = (0001\ 1010\ 0101)_2 = (1\ 1010\ 0101)_2$$

e) Conversion Binaire → Hexadécimal

On code par groupe de 4 digits en commençant de droite vers la gauche. Chaque groupe de 4 chiffres binaires est remplacé par 1 chiffre hexadécimal (utiliser le tableau précédent).

$$(0001\ 1010\ 0101)_2 = (?)_{16}$$

$$\underline{0001}\ \underline{1010}\ \underline{0101}$$

$$1\quad A\quad 5$$

$$(0001\ 1010\ 0101)_2 = (1A5)_{16}$$

I.4. Principe de fonctionnement d'un ordinateur

I.4.1. Structure générale d'un ordinateur

Pour comprendre le principe de fonctionnement d'un ordinateur, on doit tout d'abord expliquer sa structure logique. Cette structure nous permet d'expliquer, d'une manière sommaire, comment un programme est exécuté et comment ce dernier traite les données.

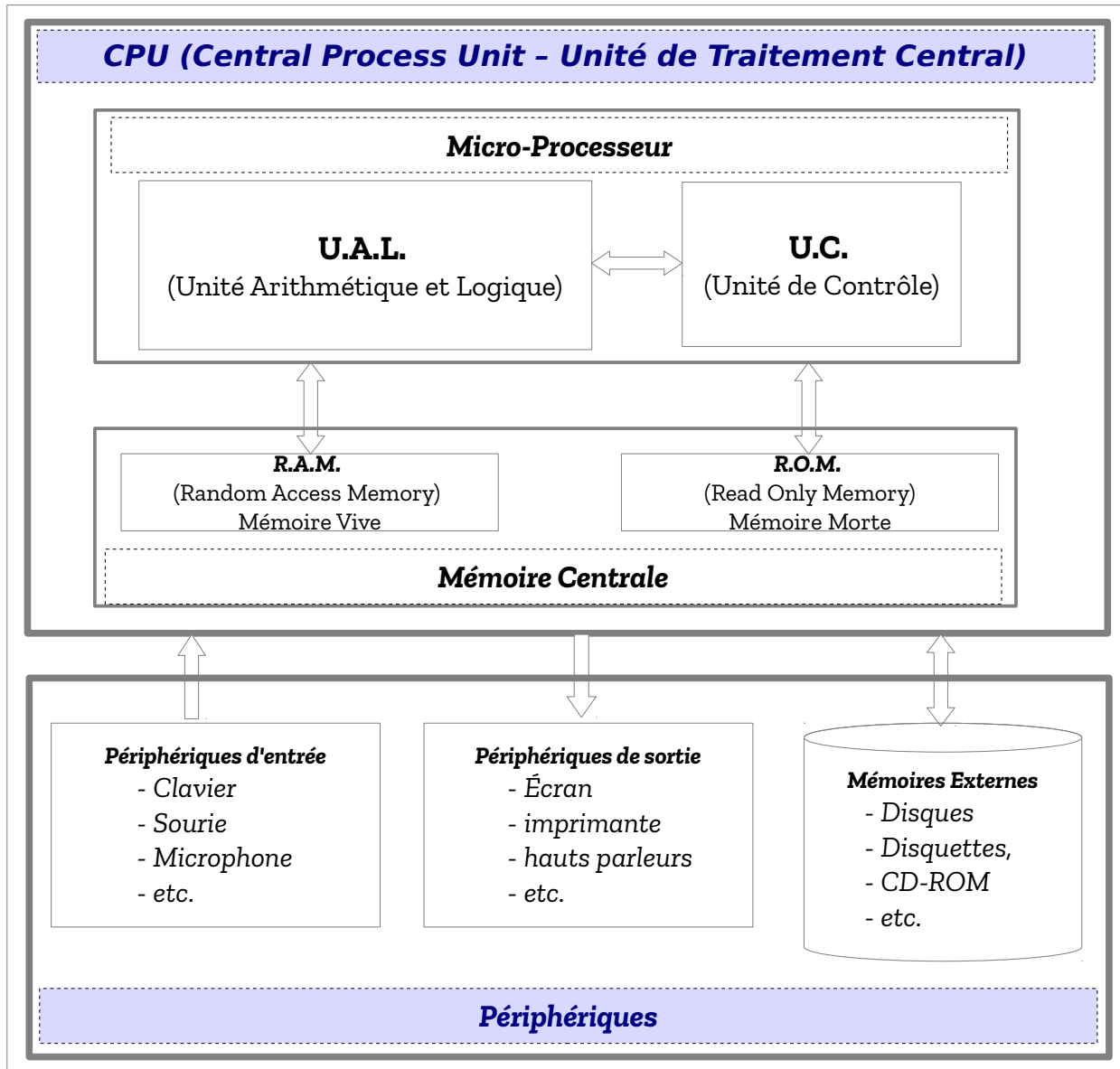


Figure I.1 : Structure d'un Ordinateur

C.P.U. : Central Process Unit (Unité de Traitement Centrale)

U.C. : Unité de contrôle ou de commande

U.A.L. : Unité Arithmétique et Logique

M.C. : Mémoire Centrale

Un ordinateur, d'une manière générale, est constitué d'un *CPU* (Central Process Unit) et des périphérique. Le *CPU* est constitué d'un *Micro-Processeur* (ou *Processeur*) et d'une *Mémoire Centrale*. Cette dernière est constituée d'une mémoire vive (*RAM* : Random Access Memory) et d'une mémoire morte (*ROM* : Read Only Memory). Le processeur se compose de deux unités : unité de contrôle et unité arithmétique et logique. Les périphériques sont composés de périphériques d'entrée, périphériques de sortie et et périphérique de sauvegardes (Mémoires externes). La figure I.1 illustre la structure générale d'un ordinateur :

I.4.1.1. L'U.A.L (Unité Arithmétique et Logique : Circuit de Calcul)

Elle consiste le circuit de calcul. Elle est chargée d'effectuer toutes les opérations arithmétique et logiques. Elle contient tous les circuits logique pour réaliser les différentes opérations arithmétique (Addition, soustraction, multiplication, division, etc.) et logiques (décalage, rotation, et logique, et binaire, ou logique, ou binaire, négation logique, complément, etc.).

I.4.1.2. L'U.C. (Unité de contrôle : Circuit de commandes)

Elle contrôle toutes les opérations qui s'effectuent dans l'ordinateur. C'est elle qui permet l'exécution d'un programme présent en mémoire centrale (M.C.)

I.4.1.3. La M.C. (La mémoire centrale : circuit de mémorisation)

- **RAM** : (Random Access Memory – Mémoire à Accès Aléatoire - Mémoire vive ou mémoire volatile) Elle a pour rôle de stocker le programme à exécuter ainsi que les données résultats. Un programme ne peut s'exécuter que s'il est chargé en mémoire centrale. Elle est constituée de plusieurs barrettes (barrettes RAM) . Cette mémoire est volatile, c'est à dire qu'elle s'efface par coupure de courant.

- **ROM** : (Read Only Memory – Mémoire en lecture seule – Mémoire morte) mémoire qui ne s'efface pas par coupure de courant. Cette dernière sert à conserver du code et des paramètres système nécessaire au fonctionnement de l'ordinateur (BIOS : Basic Input/Output System - Programme de base des entrées/sorties).

La mémoire centrale se mesure actuellement par milliers de méga-octets. On a :

- 1 bit : (état électronique 1 ou 0) = unité élémentaire de l'information (Binary Element)

- 1 octet = 1 caractère = 8 bits = 2^8 bits.

- 1 Kilo-octets = 2^{10} octets = 1024 octets

- 1 Mega-octets = 2^{10} Kilo-octets = 2^{20} octets

- 1 Giga-octets = 2^{10} Mega-octets = 2^{20} Kilo-octets = 2^{30} octets

I.4.1.4. Les mémoires externes o auxiliaires (Support de stockage et de mémorisation)

Elle servent à conserver des grandes quantités d'informations (fichiers de données, programmes, logiciels d'applications, système d'exploitation, *etc.*). Ceux sont des supports magnétiques, optique ou électroniques qui ne s'effacent pas par coupures de courant. On peut en citer les disques magnétiques, disquettes, cassettes, CD-ROM, flash-disque, *etc.*

Actuellement la capacité des diques magnétiques compte de dizaines de Giga-octets pour les ordinateurs personnels (P.C.)

$$1 \text{ Giga-octets} = 2^{30} \text{ octets} \gg 1 \text{ milliard d'octets}$$

I.4.1.5. Les périphériques d'entrée (ou organes d'entrée)

Ils servent à transmettre les informations à l'ordinateur. Exemples :

- Clavier (on des claviers AZERTY ou QWERTY : introduire des textes et des commandes)

- Microphone (introduire des sons et de la parole)

- Caméra (introduire de la vidéo)

- Lecteur optique (introduire des données à partir de CD ou DVD)

- Scanner (numérisé des photos)
- Tous de type de capteurs (Capteur de chaleurs, d'acidité, *etc.*)

I.4.1.6. Les périphériques de sortie (ou organes de sortie)

Ils servent à recevoir les information provenant de l'ordinateur vers l'extérieur. Ils permettent de traduire les informations de leurs formes codées (des 1 et 0) vers leurs formes naturelles. Exemple :

- L'écran
- L'imprimante
- Table traçante
- Synthétiseur vocal
- Synthétiseur musical
- Haut parleur, *etc.*

Remarque :

Les mémoires externes ou auxiliaires (Disques durs, disquettes, flash-disque, CD, DVD, *etc.*) sont considérées comme des périphériques d'entrée et de sortie en même temps.

I.4.2. Fonctionnement d'un Ordinateur

Tous les traitement réalisé par un ordinateur se fait via l'exécution d'un programme au niveau de CPU (Central Process Unit). Cette exécution se fait en suivant les étapes suivantes :

- ✓ Un programme, avant son exécution, sera tout d'abord chargé au niveau de la mémoire centrale ; (un programme est constitué de deux parties : *données* et *instructions*)
- ✓ Le micro-processeur récupère la première instruction du programme, réalise son décodage et l'exécute. Cette exécution, peut éventuellement récupérer des données de la mémoire central où écrire des données sur cette mémoire.
- ✓ Le micro-processeur réalise la même chose pour la deuxième instruction, et ainsi de suite, jusqu'à la dernière instruction du programme.
- ✓ Une fois le programme est terminé (l'exécution de la dernière instruction), l'espace de la mémoire central occupé par ce programme sera libérée.

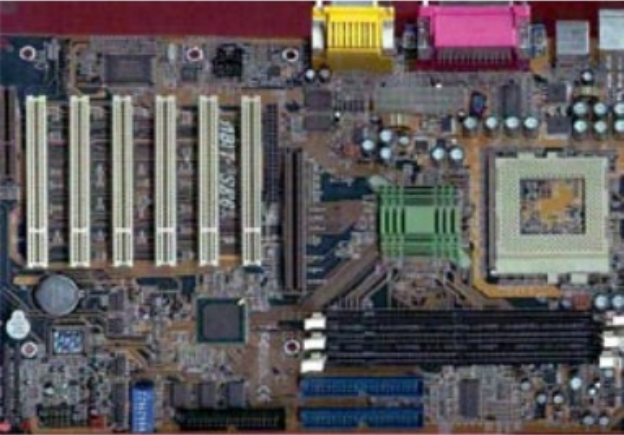
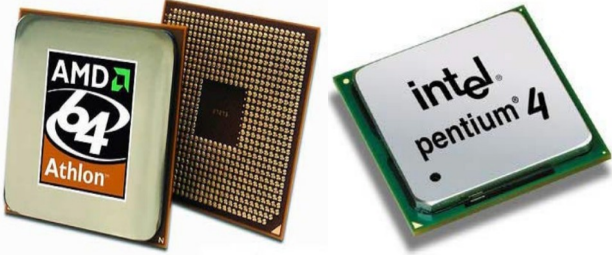
Tous les données manipulées par un programme sont chargées au niveau de la mémoire centrale, et comme on a vu, cette mémoire est volatile. Pour cela, les programme réalise en



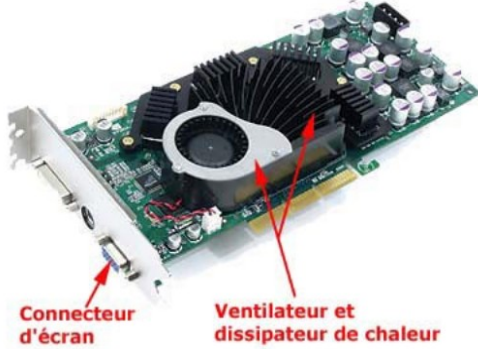
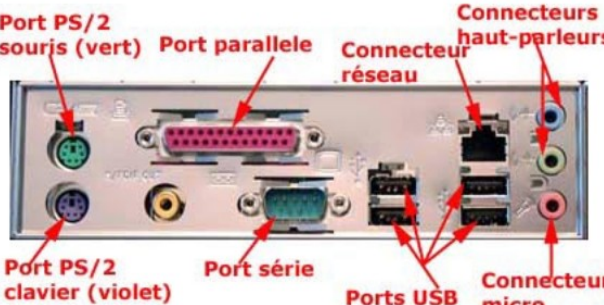
générale des opérations d'entrées/sorties vers les disques externes (sauvegardes et chargement de fichiers).

I.5. Partie Matériel d'un Ordinateur

Un ordinateur est composé de deux parties : partie matérielle (*Hardware*) et partie logicielle (*Software*). La partie hardware permet de réaliser les instructions et les commandes de la partie software.

Le tableau suivant illustre quelques composants matériels d'un ordinateur :

<i>Composant</i>	<i>Description</i>
<p><i>Carte Mère</i></p> 	<p>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 (carte graphique externe, carte son, carte réseau, etc.).</p>
<p><i>Processeur</i></p> 	<p>Processeur dont tous les éléments sont miniaturisés et rassemblés sur une puce en un seul ou en plusieurs circuits intégrés.</p> <p>Le processeur remplit les fonctions d'unité centrale dans un micro-ordinateur. Un ventilateur est utilisé pour réduire sa température souvent élevée et réduire les risques de surchauffe.</p>

<p>Mémoire Vive</p> 	<p>Mémoire du travail du processeur, tout programme exécuté doit être chargé sur la mémoire vive, dans laquelle les instructions peuvent lire ou écrire des données.</p>
<p>Disque Dur</p> 	<p>Support de mémoire de masse d'accès rapide qui est composé d'un ou de plusieurs disques magnétiques généralement solidaires d'une mécanique, et ayant une très grande capacité de stockage.</p>
<p>Carte graphique</p> 	<p>Carte d'extension servant à stocker et à interpréter les images reçues par l'ordinateur et à mettre à jour continuellement l'écran du moniteur. Elle n'est pas obligatoire sur certains ordinateurs qui en possèdent une incorporée dans la carte mère. Mais pour de meilleures performances il est conseillé d'en installer une.</p>
<p>Connecteurs et Ports</p> 	<p>Ces ports permettent de relier des périphériques externes (clavier, souris, etc.) au boîtier de l'unité centrale. Il y a plusieurs type de ports : Port USB, port parallèle, port série, etc.</p>

<p>Écran</p> 	<p>Périphérique de sortie, il permet d'afficher les toute sorte de données. Il y a deux mode d'affichage : mode texte et mode graphique.</p> <p>Il y a des écran tactile permettant d'émettre des commandes via de simples touches de doigts, dans ce cas, l'écran représente aussi un périphérique d'entrée.</p>
<p>Clavier</p> 	<p>Périphérique permettant d'entrer des données sous forme textuelle vers le CPU. Donc il représente un périphérique de sortie.</p>
<p>Sourie</p> 	<p>Permet de lancer des commandes à travers de simples cliques : clique gauche pour la sélection, double clique gauche pour exécuter un programme ou ouvrir un fichier, clique droit affiche le menu contextuel, <i>etc.</i></p>
<p>Imprimante</p> 	<p>Périphérique de sortie permettant d'afficher des données (textuelles, images ou graphes) sur du papier.</p>

Tableau I.1 : Composants matériels de l'ordinateur

La figure I.2 résume la partie hardware de l'ordinateur (ressources matérielles) :

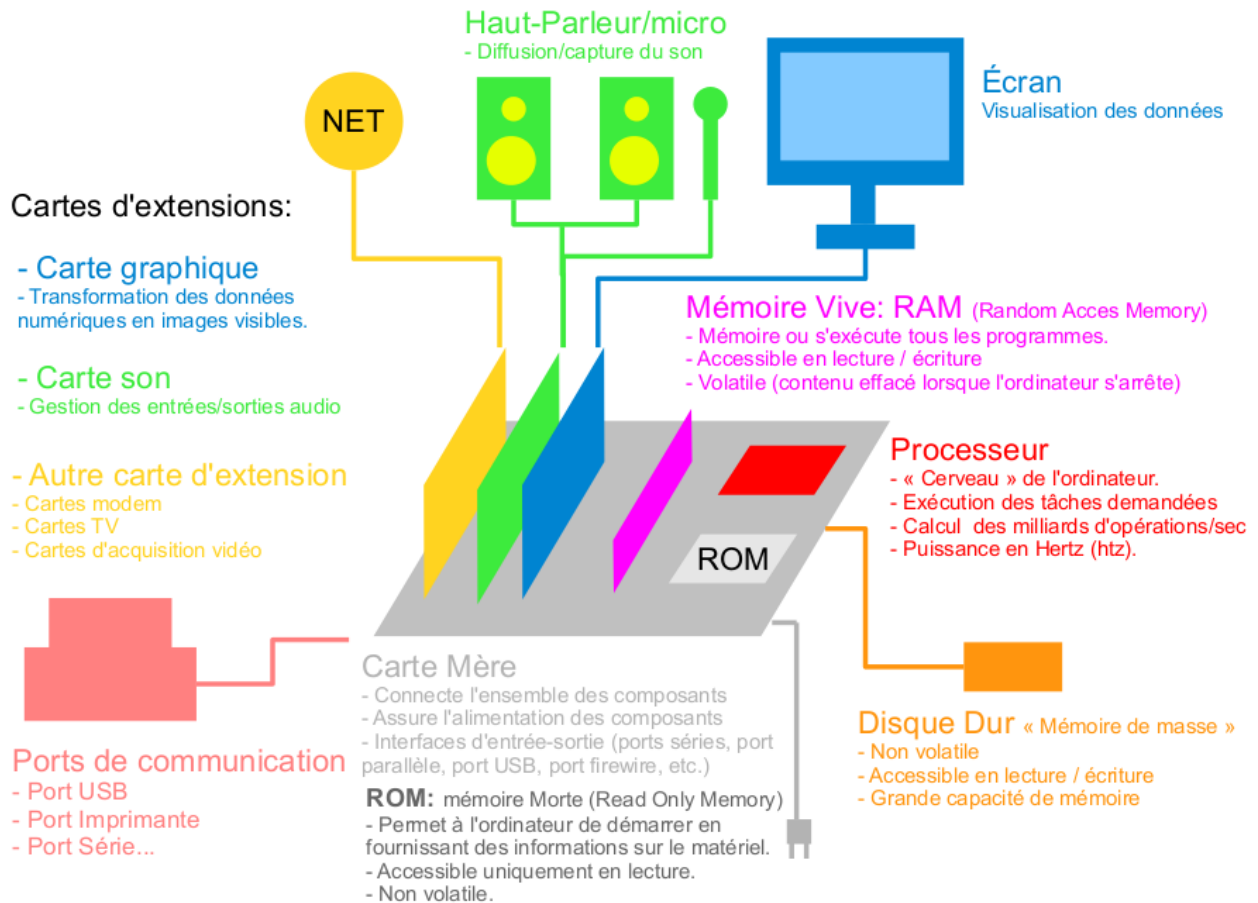


Figure I.2 : Partie matérielle de l'ordinateur (Hardware)

I.6. Partie Système

En plus de la partie matérielle, un ordinateur possède une partie logicielle (software). Cette partie est non tangible (par opposition au matériel). Elle contient des programmes, application et données, et elle est indispensable au fonctionnement d'un ordinateur que le matériel lui-même. Le software s'exécute dans la mémoire vive.

On peut diviser cette partie en trois grandes catégories :

- Système de base : ce sont les systèmes d'exploitation comme Windows, Linux, Mac OS, *etc.*
- Langages de Programmation : à fin de créer des applications et des logiciels, comme exemple de langages de programmation on trouve : C/C++, Java, Pascal, Matlab, *etc.*
- Logiciels d'application : comme suite de MS-Office (Word, Excel, PowerPoint, *etc.*),

PhotoShop, Auto CAD, Jeux, *etc.*

I.6.1. Systèmes d'exploitation

Actuellement, il y a plusieurs systèmes d'exploitation : Windows, Linux, Max OS, *etc.*

- Une système d'exploitation assure la liaison entre les utilisateurs, applications et les ressources matérielles de l'ordinateur ;
- il permet la gestion, la sauvegarde et l'organisation des informations au moyen d'une interface utilisateur compréhensible par l'homme (textes, icônes, images, graphes, *etc.*). Les informations manipulables sont enregistré sous forme de fichiers (contenu) que l'on range habituellement dans des dossiers (contenant) ;
- le système d'exploitation est sauvegardé sur le disque dur de l'ordinateur ;
- comme tous les programmes, le système d'exploitation s'exécute dans la mémoire vive.

Parmi les termes utilisés dans les systèmes d'exploitation on a :

- **Le fichier** : c'est l'unité de stockage des bases des données. Il existe de très nombreux types de fichiers qu'on peut différencier par leur extension.
- **Extension** : suffixe composé généralement de 3 lettres, ajouté au nom d'un fichier pour identifier son type. (ex: .mp3 .bmp .doc...)
- **Le dossier (ou répertoire)** : c'est un objet informatique pouvant contenir des fichiers et d'autres dossiers. Il sert au classement des données.
- **Application** : programme exécutable sur un ordinateur. Par exemple : Word, Firefox, messenger, winzip, les logiciels de jeu, *etc.*... Sous windows un fichier d'application comporte l'extension .exe
- **Document** : fichier créé par une application. Elle contient en générale une extension propre.
- **Arborescence** : Structure hiérarchisée des répertoires et des sous répertoires sur un disque dur.

I.6.2. Langages de programmation

Ce sont de langages utilisés par les développeurs d'applications et de logiciels à fin d'écrire des programmes. On trouve plusieurs langage de programmation, comme C/C++, PASCAL, Java, Fortran, Matlab, *etc.*

I.6.3. Programmes d'application

Ce sont des programmes exécutables spécialement écrits pour un système d'exploitation, et qui permettent de réaliser tout type de fonctions: traitement de texte, jeux et loisirs, retouche d'image, navigation internet, lecture de médias son, image, vidéo...

I.7. Évaluation des expressions arithmétiques et logiques

Cette section fait partie du chapitre 2, puisque c'est un concept lié à l'algorithmique et la programmation. Cependant, nous avons vu utile et logique de mettre ce point dans ce chapitre.

Dans l'évaluation des expressions, le compilateur (le programme qui traduit le programme de la forme code source vers le code binaire exécutable (code machine)) doit déterminer l'ordre d'évaluation des opérations.

Il y a deux types d'expressions :

- ✓ **Expressions arithmétiques** : résultat numérique (valeur entière ou réelle)
- ✓ **Expressions logiques** : résultat logique ou booléen (Vrai/Faux : True/False)

Avant de voir les niveaux de priorité, nous allons présenter quelques nouvelles opérations (que les étudiants peuvent ne pas connaître).

- **Unaire** : (moins unaire) opération unaire qui s'applique sur un seul opérant, par exemple : $-(3+2)$

Not : négation logique : **Not(True) = False** et **Not(False) = True**

DIV : division entière, par exemple : $10 \text{ DIV } 3 = 3$ ($10/3=3.33$) , $9 \text{ DIV } 4 = 2$
($9/4=2.25$)

MOD : reste de division, par exemple : $10 \text{ MOD } 3 = 1$, $15 \text{ MOD } 4 = 3$

AND : Et logique (voir I.7.3)

OR : Ou logique (voir I.7.3)

I.7.1. Ordre de priorité

Pour évaluer les expressions (arithmétiques et logiques) nous suivons les 6 niveaux de priorité suivants :

- 1) Commencer par les parenthèses (le plus profondes)
- 2) Fonctions : `sqr`, `sqrt`, `abs`, `exp`, `ln`, ...
- 3) - Unaire, `NOT`
- 4) \times / `DIV` `MOD` `AND`
- 5) $+$ - `OR`
- 6) $<$ $>$ \leq \geq $=$ \neq (en pascal `<>`)

I.7.2. Expressions arithmétiques

Les expressions arithmétiques sont des expressions qui donnent des résultats numériques. Dans les exemples qui suit, nous montrons comment indiquer l'ordre d'évaluation des opérations et les résultats intermédiaires et le résultat final :

$$(x + 3) / y \times (5 + y / (5 - y)) \quad \text{avec : } x=2 \text{ et } y = 4$$

La première chose, est qu'il faut indiquer l'ordre d'évaluation en numérotant les opérations, ceci sera réalisé en appliquant les 6 niveaux vu précédemment dans I.7.1.

Nous aurons ceci :

$$\begin{matrix} (2) & (5) & (6) & (4) & (3) & (1) \\ (x + 3) / y \times (5 + y / (5 - y)) \end{matrix}$$

5-y est l'opération la plus prioritaire puisque elle est dans les parenthèse les plus profondes (à chaque fois on ouvre une parenthèse, on augmente la priorité).

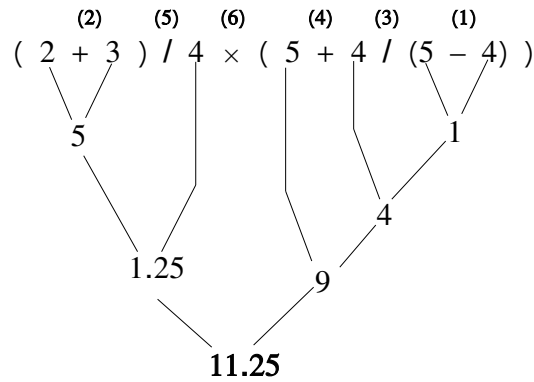
Par la suite, nous avons parenthèses du même niveau : $x+3$ et $5+y/(...)$, dans ce cas nous commençons par les parenthèses situé à gauche.

À chaque fois on trouve deux opérations du même niveau de priorité, celle à gauche deviens prioritaire.

Après avoir déterminer l'ordre, nous commençons à réaliser les calcules, comme suit :

$$\begin{array}{cccccc}
 (2) & (5) & (6) & (4) & (3) & (1) \\
 (x + 3) / y \times (5 + y / (5 - y))
 \end{array}$$

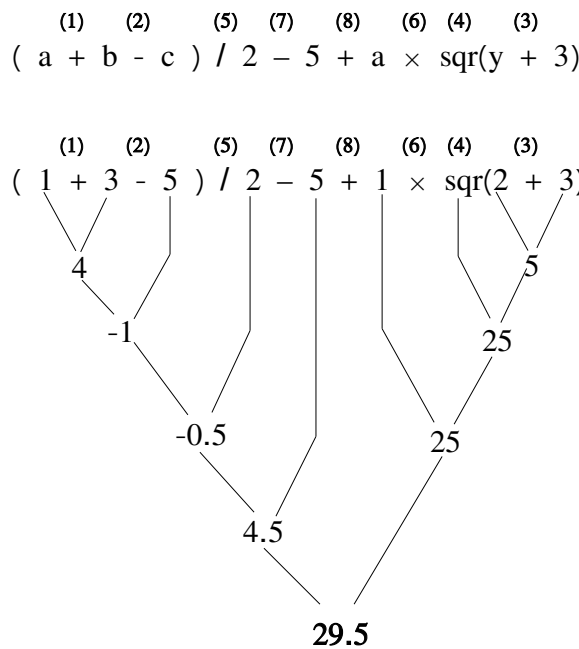
On remplace les variables x et y par leurs valeurs :



Donc, le résultat final de l'expression est : **11.25**

On prend un autre exemple, pour mieux expliquer :

$$(a + b - c) / 2 - 5 + a \times \text{sqr}(b + 3) \text{ avec } a = 1 \text{ b} = 2 \text{ et } c = 5$$



Le résultat final est : **29.5**

I.7.3. Expressions logiques

Avant de réaliser des exemples, nous allons expliquer comment les trois opérateurs logiques vu précédemment, sont calculés : NOT, AND et OR. Les trois tableaux suivants montrent comment ces trois opérateurs sont évalués (F : False, T : True) :

NOT	P	Not(P)
	F	T
	T	F

OR	P	Q	P OR Q
	F	F	F
	F	T	T
	T	F	T
	T	T	T

AND	P	Q	P AND Q
	F	F	F
	F	T	F
	T	F	F
	T	T	T

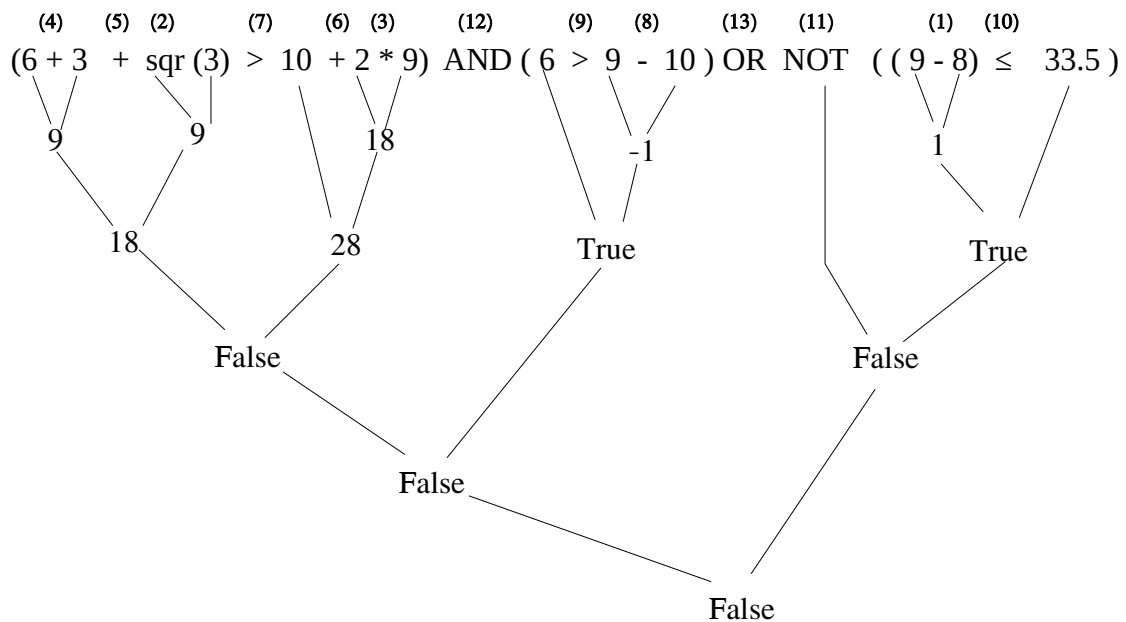
Prenons l'expression booléenne suivante, avec $x = 6$ et $y = 9$

$$(x + 3 + \text{sqr}(3) > 10 + 2 * y) \text{ AND } (x > y - 10) \text{ OR NOT } ((y-x) \leq 33.5)$$

La première chose, on indique l'ordre de priorité de chaque opération :

$$\begin{matrix} (4) & (5) & (2) & (7) & (6) & (3) & (12) & (9) & (8) & (13) & (11) & (1) & (10) \\ (x + 3 + \text{sqr}(3) > 10 + 2 * y) \text{ AND } (x > y - 10) \text{ OR NOT } ((y-x) \leq 33.5) \end{matrix}$$

On remplace les variables par leurs valeurs :



I.7.4. Expressions arithmétiques et logiques en PASCAL

L'écriture mathématique des formules (expressions logiques et arithmétiques et aussi les valeurs numériques) n'est pas la même que celle de la programmation. Dans cette section, nous allons voir comment ré-écrire ces expressions, ainsi que les valeurs numériques, en langage PASCAL.

Le tableau résume quelques expressions de base :

Exp.	PASCAL
$2a$	<code>2 * a</code>
$\frac{a}{b}$	<code>a / b</code>
a^2	<code>sqr(a)</code>
\sqrt{a}	<code>sqrt(a)</code>
$ a $	<code>abs(a)</code>
$\ln(a)$	<code>ln(a)</code>
$\log(a)$	<code>ln(a) / ln(10)</code>
e^a	<code>exp(a)</code>
x^n	<code>exp(n * ln(x))</code>

Pour les valeurs, nous prenons les exemples suivants :

Valeurs (Maths)	Valeurs (Pascal)
125,56	125.56
96,2 10^9	96.2E9
0,39 10^{-16}	0.39E-16