



Ministère de l'Enseignement Supérieur et de
la Recherche Scientifique



Cours

INITIATION A L'INFORMATIQUE

Licence 1



05/07/2020



Table des matières

1	INTRODUCTION AU FONCTIONNEMENT DES ORDINATEURS.....	3
1.1	Généralités :.....	3
1.2	Un peu d'histoire :	4
1.3	Représentation de données :	5
2	SYSTEMES DE NUMERATION	6
2.1	Définition d'un système.....	6
2.2	Système décimal.....	6
2.3	Système binaire	6
2.4	Systèmes associés au système binaire	7
2.5	Passage d'un système a un autre	8
2.6	Passage du système octal au système décimal (respectivement binaire, hexadécimal)	10
2.7	Passage du système hexadécimal au système décimal (Respectivement binaire, octal)	10
2.8	Représentation d'un nombre fractionnaire	11
2.9	Operations arithmétiques	12
2.10	Codes binaires.....	16
2.11	Codes alphanumériques	22
2.12	Compléments sur les nombres binaires	23
3	ELEMENTS DE MATERIEL INFORMATIQUE OU HARDWARE	25
3.1	Introduction.....	25
3.2	Structure générale de l'ordinateur.....	26
3.3	Présentation des unités de base	26
3.4	Carte mère.....	29
3.5	Périphériques.....	33
3.6	Mémoires de masse ou mémoires auxiliaires	38
3.7	Notion des trois bus.....	39
4	ELEMENTS DE LOGICIEL INFORMATIQUE OU SOFTWARE	41
4.1	Niveau de logiciel.....	41
4.2	Logiciels systèmes.....	42
4.3	Système d'exploitation	43

4.4	Langages de programmation et leurs traducteurs.....	47
4.5	Logiciels d'application	48
5	BIBLIOGRAPHIE	54

1 INTRODUCTION AU FONCTIONNEMENT DES ORDINATEURS

1.1 Généralités :

Le terme informatique a été défini en France en 1967 par l'académie française, cette définition est la suivante :

« Science du traitement rationnel, notamment à l'aide de machines automatiques, de l'information, considérée comme le support de connaissances dans les domaines scientifique, économique et social ».

L'informatique est la contraction d'information et d'automatique. C'est la science du traitement automatique de l'information.

Un ordinateur est un ensemble d'équipements électroniques utilisé pour le traitement automatique de l'information.

Un ordinateur est un système à microprocesseur, système électronique utilisé pour le traitement de l'information.

L'ordinateur se repose sur 2 grandes familles qui sont :

Le matériel (Hardware) qui est la partie tangible ou physique de l'ordinateur.

Une partie immatérielle (Software) qui gère le fonctionnement de l'ordinateur, au moyen de programme.

La notion d'architecture des ordinateurs permet de comprendre le fonctionnement de cet outil en fonction de l'usage que l'on veut en faire. Le choix d'une architecture est toujours le résultat d'un compromis :

- entre performance et coût ;
- entre efficacité et facilité de construction ;
- entre performance d'ensemble et facilité de programmation ;
- entre fonctionnalité et logiciel

Un microprocesseur est un circuit intégré complexe. Il résulte de l'intérêt de l'intégration sur une puce de fonctions logiques combinatoires et séquentielles. Il est capable d'interpréter et d'exécuter les instructions d'un programme. Son domaine d'utilisation est vaste.

Le concept de microprocesseur a été créé par la société INTEL, elle-même créée en 1968. Mais le 1^{er} processeur est né en 1971, c'est le 4004 avec une unité de calcul 4 bits, 2300 transistors et fonctionnant à 108 KHz.



1.2 Un peu d'histoire :

Avant J.-C., les outils de calcul étaient l'abaque et le boulier.

1580 : Neper invente les logarithmes.

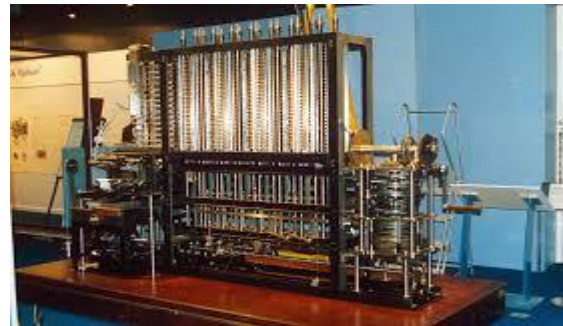
1632 : Oughred invente la règle à calcul.

1642 : Blaise Pascal crée la Pascaline qui pouvait traiter les additions et les soustractions.

1679 : Leibniz met au point l'arithmétique binaire et invente en 1694, une machine capable de traiter la multiplication et la division.

1728 : Falcon et Jacquard créent le métier à tisser utilisant les cartes perforées.

1833 : Charles Babbage crée une machine avec une unité de calcul, une mémoire, des registres et des entrées de données par carte perforée. Il est considéré comme le père de l'ordinateur.



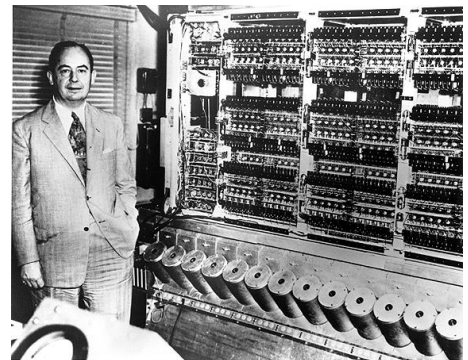
1854 : Boole met en place l'algèbre de Boole pour simuler 2 états (Vrai ou Faux).

1904 : invention du 1^{er} tube à vide à diode par John Fleming.

1938 : Shannon crée le lien étroit entre circuit électrique et l'algèbre de Boole. C'est lui qui définit le chiffre binaire **bit** (**B**inary **digi**T).

L'ENIAC (Electronic Numerical Integrator And Computer), est en 1945 le premier ordinateur entièrement **électronique** construit à partir d'une idée de John William Mauchly pour être **Turing-complet**. Il peut être reprogrammé pour résoudre, en principe, tous les problèmes calculatoires.

1945 : John Von Neumann crée une machine à programme enregistré appelé EDVAC (electronic discrete Variable Automatic Computer) opérationnel en 1952.



1947 : invention du transistor.

1950 : invention du langage **Assembleur**.

1953 : invention de la mémoire de tores de ferrite.

1957 : invention du 1^{er} langage de programmation, le FORTRAN.

1965 : **Gordon MOORE** écrit la loi de Moore qui dit que la complexité des circuits intégrés double tous les ans.

1967 : IBM construit le 1^{er} lecteur de disquette.

1968 : création du langage Pascal

1969 : début d'UNIX.

1971 : création de 1^{er} microprocesseur et de la 1^{ère} imprimante laser.

1973 : invention du disque dur.

1984 : création de l'Apple Macintosh.

Depuis les années 80, c'est la montée fulgurante des inventions informatiques.

1.3 Représentation de données :

Les informations traitées par un ordinateur peuvent être de différents types (texte, image, son, vidéo) mais elles sont toujours représentées et manipulées par l'ordinateur sous forme binaire. Toute information sera traitée comme une suite de 0 ou de 1. L'unité d'information est le chiffre binaire appelé bit.

Le code de l'information consiste à établir une correspondance entre la représentation externe (habituelle) de l'information (suite de caractère) et sa représentation interne dans la machine, qui est une suite de bit.

On utilise la représentation binaire car elle est simple et facile à réaliser techniquement en réalisant les états des transistors.

Pour se faire, il existe plusieurs tables de codification dont la plus utilisée est la table ASCII.

2 SYSTEMES DE NUMERATION

2.1 Définition d'un système

La représentation des nombres a toujours existé dans les cultures. Les premières représentations utilisaient des traits (I, II, III,..., X, L, M, C, ...).

Exemples de chiffres romains : 1=I, 2=II, 3=III, 4=IV, 5=V, 6=VI, 7=VII, 8=VIII, 9=IX, 10=X, 50=L, 100=C, 500=D, M=1000, 2020=MMXX, 2021=MMXXI

Les différents symboles sont appelés des chiffres. Le système décimal est le système utilisé par l'Homme. On notera que l'ordinateur utilise le système binaire et les systèmes associés tel que le système octal, le système hexadécimal, de même que les codes binaires comme le DCB (Décimal Codé Binaire).

2.2 Système décimal

Le système décimal ou base 10 utilise les 10 chiffres décimaux : 0, 1, 2, 3, 4, 5, 6, 7, 8 et 9. L'écriture d'un nombre dans ce système est une suite ordonnée de ces chiffres. Par exemple, 4520 est interprété comme suit :

10^3	10^2	10^1	10^0	
1000	100	10	1	
4	5	2	0	$\longrightarrow 4520_{10} = 4 \cdot 10^3 + 5 \cdot 10^2 + 2 \cdot 10^1 + 0 \cdot 10^0$

On remarque que le poids de chaque chiffre es multiplié par 10 au fur et à mesure qu'on avance de droite à gauche. Le poids de chaque chiffre est donc fonction de sa position dans l'écriture du nombre. Cette écriture se généralise pour la représentation dans l'écriture d'un nombre entier dans une base quelconque b, à savoir :

$$N = a_n b^n + a_{n-1} b^{n-1} + \dots + a_i b^i + \dots + a_1 b^1 + a_0 b^0$$

Où

Les a_i représentent les chiffres (symboles du système de numération, et, donc les valeurs sont comprises entre 0 et b-1 ;

Les b^i constituent les poids respectifs des chiffres a_i ; a_n correspond donc au chiffre de poids fort (b^n) et à celui de poids faible (b^0) ;

Les i sont les rangs ou place ou position des chiffres a.

2.3 Système binaire

Le système binaire ou base 2 utilise deux chiffres (symboles ou bits), 0 et 1.

Tout nombre écrit dans cette base est une suite de ces chiffres.

Ainsi, 1110_2 signifie comme précédemment :

$$\begin{array}{cccc}
 2^3 & 2^2 & 2^1 & 2^0 \\
 8 & 4 & 2 & 1 \\
 1 & 1 & 1 & 0
 \end{array}
 \longrightarrow 1110_2 = 1 * 2^3 + 1 * 2^2 + 1 * 2^1 + 0 * 2^0$$

Ici, le poids du chiffre est multiplié par 2, de droite à gauche.

2.4 Systèmes associés au système binaire

Deux systèmes de numération sont souvent utilisés en lieu et place du système binaire. Ce sont le système octal et le système hexadécimal.

Système octal

Le système octal ou système à base 8 utilise 8 chiffres, à savoir : 0, 1, 2, 3, 4, 5, 6, 7.

Tout nombre peut être représenté dans ce système à l'aide de ces 8 chiffres par exemple, 725_8 signifie :

$$\begin{array}{ccc}
 8^2 & 8^1 & 8^0 \\
 64 & 8 & 1 \\
 7 & 2 & 5
 \end{array}
 \longrightarrow 725_8 = 7 * 8^2 + 2 * 8^1 + 5 * 8^0 = 469_{10}$$

Signalons que le poids est multiplié par 8 de droite à gauche.

Système hexadécimal

Le système hexadécimal ou système à base 16 utilise 16 chiffres (symboles) :

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

Tout nombre représenté dans le système hexadécimal est donc une suite des symboles ci-dessus.

Par exemple, $40C_{16}$ signifie :

$$\begin{array}{ccc}
 16^2 & 16^1 & 16^0 \\
 256 & 16 & 1 \\
 4 & 0 & C
 \end{array}
 \longrightarrow 40C_{16} = 4 * 16^2 + 0 * 16^1 + C * 16^0 = 4 * 256 + 0 + 12 * 1 \quad (C = 12)$$

On remarque une correspondance entre les nombres binaires de 0000 à 1001 (code 0 à 9) et des combinaisons binaires de 1010 à 1111 qui prennent les lettres alphabétiques de A à F.

D'où, le tableau de correspondance récapitulatif suivant :

Décimal	Binaire	Octal	Hexadécimal
0	0000	0	0
1	0001	1	1

2	0010	2	2
3	0011	3	3
4	0100	4	4
5	0101	5	5
6	0110	6	6
7	0111	7	7
8	1000	10	8
9	1001	11	9
10	1010	12	A
11	1011	13	B
12	1100	14	C
13	1101	15	D
14	1110	16	E
15	1111	17	F

Tableau 1 : Tableau de correspondance

2.5 Passage d'un système a un autre

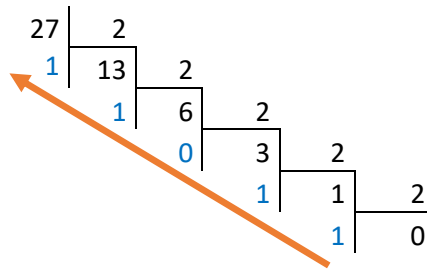
Il est possible de passer de l'une à l'autre des bases ci-dessus

Passage du système décimal au système binaire (respectivement octal, hexadécimal)

La conversion d'un nombre écrit en décimal en un nombre binaire (resp. octal. Resp. Hexadécimal) s'obtient en effectuant des divisions entières successives par 2 (resps. Par 8, resp. Par 16) jusqu' à l'obtention d'un quotient nul. L'équivalent binaire (resp. octal, resp. hexadécimal) est l'ensemble des différents restes pris du dernier au premier (le chiffre de poids fort est le dernier reste, celui de poids faible est le premier reste).

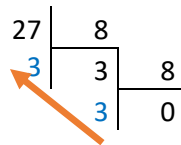
Exemple 1 : convertir le nombre entier 27_{10} en binaire

$$27_{10} = 11011_2$$



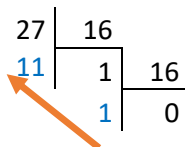
Exemple 2 : convertir le nombre entier 27_{10} en octal

$$27_{10} = 33_8$$



Exemple 3 : convertir le nombre entier 27_{10} en hexadécimal

$$27_{10} = 1B_{16}$$



On rappelle B vaut 11

Passage du système binaire au système décimal (respectivement octal, hexadécimal).

Passage Binaire – Décimal

La conversion d'un nombre binaire en décimal s'obtient en additionnant les puissances de 2 correspondances aux bits de valeur 1.

Exemple : Convertir le nombre binaire 1101_2 en décimal

$$\begin{array}{cccc}
 2^3 & 2^2 & 2^1 & 2^0 \\
 1 & 1 & 0 & 1
 \end{array}
 \quad
 \begin{aligned}
 1_2 &= 1 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 \\
 &= 8 + 4 + 0 + 1 = 13_{10}
 \end{aligned}$$

Passage Binaire – Octal

La conversion d'un nombre binaire en octal se fait en effectuant un remplacement, de droite à gauche, de 3 bits par chiffre octal correspondant. Si le nombre de bits n'est pas un multiple de 3, compléter le nombre par des zéros à gauche.

Exemple : Convertir le nombre binaire $110\ 111\ 101_2$ en octal.

2^2	2^1	2^0	2^2	2^1	2^0	2^2	2^1	2^0	→ Poids
1	1	0	1	1	1	1	0	1	→ Nombre binaire

6	7	5	→ Equivalent
$110111101_2 = 675_8$			

Passage Binaire – hexadécimal

La conversion d'un nombre binaire en hexadécimal se fait en effectuant un remplacement, de droite à gauche, de 4 bits par le chiffre hexadécimal correspondant. Si le nombre de bits n'est pas un multiple de 4, compléter le nombre par des zéros à gauche.

Exemple : Convertir le nombre binaire $1\ 1011\ 1101_2$ en hexadécimal

2^3	2^2	2^1	2^0	2^3	2^2	2^1	2^0	2^3	2^2	2^1	2^0	→ Poids
0	0	0	1	1	0	1	1	1	1	0	1	→ Nombre binaire
1				B				D				→ Equivalent

$110111101_2 = 0001\ 1011\ 1101_2 = 1BD_{16}$

2.6 Passage du système octal au système décimal (respectivement binaire, hexadécimal)

Passage Octal – Décimal

La conversion d'un nombre octal s'obtient en additionnant les poids (puissances de 8) multiplié par le chiffre correspondant.

Exemple : Convertir 675_8 en décimal

8^2	8^1	8^0	
6	7	5	$6 \cdot 8^2 + 7 \cdot 8^1 + 5 \cdot 8^0 = 445_{10}$

Passage Octal – Binaire

Pour convertir un nombre octal en binaire, il suffit d'explorer chaque chiffre octal en son équivalent binaire sur 3 bits.

Exemple : Convertir 675_8 en binaire

On a $6_8 = 110_2$ $7_8 = 111_2$ $5_8 = 101_2$

Par suite, $675_8 = 110111101_2$

2.7 Passage du système hexadécimal au système décimal (Respectivement binaire, octal)

Passage Hexadécimal – Décimal

La conversion d'un nombre hexadécimal s'obtient en additionnant les poids (puissances de 16) multipliés par le chiffre correspondant

Exemple : Convertir 491_{16} en décimal

16^2	16^1	16^0
--------	--------	--------

$$4 \quad 9 \quad 1 \quad 4 \cdot 16^2 + 9 \cdot 16^1 + 1 \cdot 16^0 = 1169_{10}$$

Passage hexadécimal – Binaire

Pour convertir un nombre hexadécimal en binaire, il suffit d'écarter chaque chiffre hexadécimal en son équivalent binaire sur 4 bits.

Exemple : Convertir $3D6_{16}$ en binaire

$$\text{On a : } 3_{16} = 0011_2 \quad D_{16} = 1101_2 \quad 6_{16} = 0110_2$$

$$\text{Par suite, } 3D6_{16} = 001111010110_2$$

Passage hexadécimal - octal

Il n'existe pas de passage direct d'un nombre octal en son équivalent hexadécimal. On peut passer de l'octal en binaire ou en décimal, puis obtenir en suite l'équivalent hexadécimal.

2.8 Représentation d'un nombre fractionnaire

Cas Binaire

Pour représenter un nombre fractionnaire en binaire, on procède par des multiplications successives de parties fractionnaires jusqu'à l'obtention d'une partie fractionnaire nulle ou jusqu'au nombre de bits désirés. Le résultat est constitué par les parties prises de première à la dernière.

Exemple : Convertir le nombre fractionnaire $0,675_{10}$ en binaire.

	Partie entière		Fraction	Chiffre Binaires
$0,675 \cdot 2 =$	1	+	0,350	$a_1 = 1$
$0,350 \cdot 2 =$	0	+	0,700	$a_2 = 0$
$0,700 \cdot 2 =$	1	+	0,400	$a_3 = 1$
$0,400 \cdot 2 =$	0	+	0,800	$a_4 = 0$

$$\text{Par suite, } 0,675_{10} = (0, a_1 a_2 a_3 a_4)_2 = 0,1010_2$$

Dans une base quelconque b

On généralise le procédé ci-dessus dans le cas d'une base quelconque b. Les chiffres étant constitués par les parties entières dans la base. Il est à remarquer que ces chiffres doivent être compris entre 0 et b – 1.

Exemple : Convertir $0,513_{10}$ en octal à l'aide de quatre chiffres.

	Partie entière		Fraction	Chiffre Binaire
$0,513 \cdot 8 =$	4	+	0,104	$a_1 = 4$
$0,104 \cdot 8 =$	0	+	0,832	$a_2 = 0$
$0,832 \cdot 8 =$	6	+	0,656	$a_3 = 6$

$$0,656 \times 8 = \quad 5 \quad + \quad 0,248 \quad a_4 = 5$$

Par suite $0,513_{10} = 0,4065_8$

REPRESENTATION D'UN NOMBRE REEL

Un nombre réel est formé d'une partie entière et une partie fractionnaire. La représentation d'un tel nombre dans n'importe quel base b se fait par la juxtaposition des représentations des parties entières et fractionnaires.

Exemple 1 : Convertir $10,675_{10}$ en binaire

Le résultat est $10,675_{10} = (1010,1011)_2$

Exemple 2 : Représenter le nombre $408,5_{10}$ en octal

Résultat : $408,5_{10} = (630,4)_8$

2.9 Opérations arithmétiques

L'écriture d'un nombre dans la mémoire de l'ordinateur est une succession de chiffres binaire, ou bits. Du fait du fonctionnement des circuits logiques qui composent l'ordinateur, les opérations arithmétiques effectuées par l'unité arithmétique et logique le sont en binaire. Notons cependant qu'il est possible d'effectuer les différentes opérations arithmétiques dans n'importe quelle base. Ici, l'accent est mis sur le cas binaire.

Addition binaire

Notons la table d'addition binaire suivante :

$$0+0=0$$

$$0+1=1$$

$$1+0=1$$

$$1+1=0 \text{ plus } 1 \text{ de retenue dans la position suivante.}$$

De plus, l'addition binaire ne diffère pas fondamentalement de l'addition décimale. Tout se ramène à une gestion de la retenue comme c'est le cas en décimal.

Exemple : Additionner deux nombres binaires 00000100_2 et 00000101_2

$$\begin{array}{r} 00000100 \\ + 00000101 \\ \hline \end{array}$$

$$000001001_2 = 9_{10}$$

Multiplication binaire

La multiplication binaire est la suivante :

$$0 \times 0 = 0$$

$$0 \times 1 = 0$$

$$1 \times 0 = 0$$

$$1 \times 1 = 1$$

La multiplication de nombres entiers consiste en des décalages des bits du multiplicande vers la gauche par rapport au multiplicateur, suivi d'additions.

Exemple : multiplier le nombre 00001111_2 par 00001010_2

$$\begin{array}{r}
 00001111 \\
 * 00001010 \\
 \hline
 00000000 \\
 00001111 \\
 00000000 \\
 00001111 \\
 \hline
 00010010110 = 2^7 + 2^4 + 2^2 + 2^1 = 150_{10} \quad (15 * 10) = 150
 \end{array}$$

Soustraction et Notion de complément

Notons la table de soustraction binaire suivante :

$$\begin{array}{ll}
 0 - 0 = 0 & 0 - 1 = 1 \text{ avec 1 de retenue dans la position suivante} \\
 1 - 0 = 1 & 1 - 1 = 0
 \end{array}$$

Cependant, la soustraction de 2 nombres peut se ramener à une addition d'un des nombres avec l'opposé de l'autre. Cette propriété n'est pas souvent utilisée n'est en décimal. Cette méthode est possible du fait de la facilité de représenter le complément (l'opposé nombre négatif au besoin) de tout nombre positif dans une base donnée. De plus, un même circuit logique peut être l'addition et la soustraction :

On distingue pour tout système de numération en base b , deux types de complément :

Le complément à b , et

Le complément à $(b - 1)$.

Ainsi, dans le cas du système décimal, on parle de complément à 10 et de complément à 9. Il en est de même du complément à 2 et du complément à 1 dans le système binaire.

Notion complément à b

Considérons un nombre positif N écrit dans une base b . La partie entière du nombre N est composée de n chiffres.

Le complément à b du nombre est la différence de $b^n - N$ pour N différent de 0 et 0 pour $N = 0$.

Exemple 1 : Trouver le complément à 10 du nombre 52520_{10} .

Le nombre de chiffres de la partie entière est $n = 5$, la base étant 10, le complément est $(10^5 - 52520)_{10} = 47780_{10}$.

Exemple 2 : Trouver le complément à 10 du nombre 0.3267_{10} .

Ici, le nombre de chiffres de la partie entière est 0, le complément est $(10^0 - 0,3267)_{10} = 1 - 0,3267 = 0,6733_{10}$.

Exemple 3 : Trouver le complément à 2 du nombre 101100_2

Le nombre de bits de la partie entière est 6, la base est 2. Le complément est donc : $(2^6)_{10} - (101100)_2 = (1000000 - 101100)_2$.

Exemple 4 : Trouver le complément à 10 du nombre $(24,545)_{10}$.

Le nombre de chiffres de la partie entière est 2, la base étant 10, le complément est $(10^2 - 24,545)_{10} = 75,455_{10}$.

Soustraction en complément à b

La soustraction de deux nombres positifs M et N ($M - N$) dans une base se fait comme suit :

Trouver le complément à b de N et l'ajouter à M ;

Vérifier si le résultat admet une retenue de dépassement :

- Si oui, ignorer cette retenue ;
- Si non, prendre alors le complément à b du résultat, et mettre le - devant.

Exemple : Effectuer en complément à 10, les opérations suivantes :

Exemple 1 : $(72532 - 3250)_{10}$

Dans cette opération, $M = 72532$ et $N = 03250$

Le complément à 10 de N est $(10^5 - 03250)_{10} = 96750$

Par suite,

$M + C_{10}(N) = 72532 + 96750 = 169282$, il y a une retenue de dépassement, on l'ignore

Donc $72532 - 3250 = 69282$.

Exemple 2 : $(3250 - 72532)_{10}$

Dans cette opération, $M = 03250$ et $N = 72532$

Le complément à 10 de N est $(10^5 - 72532)_{10} = 27468$

Par suite,

$M + C_b(N) = 03250 + 27468 = 30718$, il n'y a pas de dépassement

il faut prendre alors le complément à 10 de 30718 et mettre le signe - devant.

$C_{10}(30718) = (10^5 - 30718)_{10} = 69282$, affectons le signe -

Ce qui donne comme résultat -69282.

Notion complément (b -1)

Soit un nombre positif N écrit dans une base b dont la partie entière est composée de n chiffres et la partie fractionnaire de m chiffres.

Le complément à (b - 1) du nombre N est égal à $(b^n - b^{-m} - N)$.

Exemple 1 : Trouve le complément à 9 du nombre 52520_{10} .

Le nombre de chiffres de la partie entière est n = 5, celui de la partie fractionnaire est 0, la base est 10.

Le complément à 9 est $(10^5 - 10^0 - 52520)_{10} = 99999 - 52520 = 47479$

Exemple 2 : Trouver le complément à 9 du nombre $(25,639)_{10}$.

Le complément est égal à : $(10^2 - 10^{-3} - 25,639)_{10} = 99,999 - 25,639 = 74,360$.

Exemple 3 : Trouve le complément à 1 du nombre 101100_2 .

Le nombre de bits de la partie entière est 6, la base est 2.

Le complément est donc $(2^6)_{10} - 1^0 - (101100) = (1000000 - 1 - 101100)_2 = 010011_2$.

Soustraction en complément à (b - 1)

La soustraction de deux nombres positifs M et N ($M - N$) dans une base b en complément (b - 1) se fait comme suit :

c) Trouver le complément à (b - 1) de N et l'ajouter à M ;

d) Vérifier si le résultat admet une retenue de dépassement :

- Si oui, l'ajouter au chiffre de poids faible ;
- Si non, prendre alors le complément à (b - 1) du résultat, et mettre le - devant.

Exemple : effectuer en complément à 9 les opérations suivantes :

Exemple 1 : $(72532 - 3250)_{10}$

Dans cette opération, M = 72532 et N = 03250

Le complément à 9, de N est $(10^5 - 10^0 - 03250)_{10} = 99999 - 03250 = 96749$.

$M + C_9(N) = 72532 + 96749 = 169281$, ajoutons la retenue au chiffre de poids faible, ce qui donne 69282

Donc en complément à 9, $M - N = 69282$

Exemple 2 : $(3250 - 72532)_{10}$.

Dans cette opération, M = 03250 et N = 72532

Le complément à 9 de N est

$C_9(72532) = 10^5 - 10^0 - 72532)_{10} = 100\,000 - 72532 = 27467$.

$M+C_9(N) = 03250 + 27467 = 30717$, pas de dépassement

il faut prendre alors le complément à 9 de 30717 et mettre le signe - devant.

$C_9(30718) = (10^5 - 10^0 - 30717)_{10} = 100\,000 - 30718 = 69282$, affectons le signe -

Ce qui donne comme résultat -69282.

Division binaire

La division binaire est semblable à la division, décimale. La table de division est la suivante :

0 / 0 et 1 / 0 impossible ou indéterminé

Exemple 1 :

$$\begin{array}{r} 1001 \overline{) 11} \\ \underline{-11} \\ 0011 \\ \underline{-11} \\ 0 \end{array}$$

$$\begin{array}{r} 1011 \overline{) 100} \\ \underline{-100} \\ 00110 \\ \underline{-100} \\ 0100 \\ \underline{-100} \\ 000 \end{array}$$

$$\begin{array}{r} 1011100 \overline{) 11.1} \\ \underline{-111} \\ 01001 \\ \underline{-111} \\ 00100 \\ \underline{-000} \\ 1000 \\ \underline{-111} \\ 001000 \\ \underline{-111} \\ 0001 \end{array}$$

Exemple 2 : Diviser 27_{10} par 4_{10}

On a : $27_{10} = 00011011_2$ et $4_{10} = 100_2$

D'où :

N. B. : Signalons que dans les ordinateurs, la multiplication et la division se font à l'aide de programmes.

2.10 Codes binaires

Notion sur les codes binaires

L'électronique numérique utilise des signaux à deux niveaux et des circuits à deux états stables. On établit une analogie entre les signaux numériques, les circuits logiques, et les chiffres binaires.

Un nombre binaire de n chiffres peut être représenté par n circuits logiques, chaque circuit générant un signal de sortie égal à 0 ou à 1.

Tout système numérique peut représenter ou manipuler des nombres binaires mais également des informations discrètes. Ainsi toute information discrète déduite d'un ensemble d'information peut alors être représenté par un code binaire. On peut ainsi coder la couleur bleue d'un spectre de couleurs, la lettre S de l'alphabet.

On appelle bit, un chiffre binaire qui prend en termes de code binaire, la valeur 0 ou 1. Par suite, le codage binaire de n éléments nécessite au minimum n bits, puisque le nombre d'arrangements de n bits est égal à 2^n

Ainsi :

Pour 4 éléments, on peut définir un code à 2 bits (00, 01, 10, 11) ;

Pour 8 éléments, il suffit de choisir un code à 3 bits (000, 001, 010, 011, 100, 101, 110, 111) entre 0 et $(2^n - 1)$.

Codes binaires pour les décimaux

La définition de codes binaires pour les chiffres décimaux nécessite l'utilisation de 4 bits au minimum. Les codes décimaux les plus usités sont :

- Le code 8421 (0-15)
- Le code DCB (Décimal Codé Binaire)
- Le code excédent -3
- Le code 84 – 2 – 1
- Le code 2421
- Le code biquinaire ou 5043210

Le code 8421

Le code 8421 est un code binaire pondéré qui prend en compte les chiffres décimaux. On entend par pondération l'attribution de poids correspondants aux différents bits suivant leur position.

Par exemple, le chiffre décimal représenté par le code 0110 correspond au chiffre décimal

$$0 \cdot 8 + 1 \cdot 4 + 1 \cdot 2 + 0 \cdot 1 = 6_{10}$$

Le Code DCB (Décimal Codé Binaire)

Le code DCB est un code binaire pondéré 8421 où tout chiffre décimal est représenté par 4 bits correspondants. Le nombre décimal 970_{10} Est codé en DCB par $100101110000_{\text{DCB}}$. Remarquons les codes en 8421 représentant les décimaux de 10 à 15 ne sont pas utilisés en DCB.

Le code DCB est beaucoup utilisé dans les compteurs numériques, dans la transmission et la réception des données.

Notons aussi que le codage d'un nombre en DCB est différent de sa représentation en binaire. L'ordinateur est capable de réaliser des opérations arithmétiques en DCB. L'ordinateur doit cependant réaliser automatiquement un ajustement dit « ajustement décimal » pour obtenir le bon résultat.

Exemple : Additionner en DCB les nombres 60_{10} et 55_{10} .

On a :

$$60_{10} = 01100000_{\text{DCB}}$$

$$+55_{10} = 01010101_{\text{DCB}}$$

$$\hline 10110101_{\text{DCB}}$$

Le résultat est incorrect. Le code DCB 1011 n'existe pas. Il faut donc réaliser une correction. La règle consiste à ajouter **0110** soit (6_{10}) au(x) code(s) incorrects dans le résultat.

$$\text{D'où} \quad 10110101_{\text{DCB}}$$

$$+0110$$

$$\hline 000100010101_{\text{DCB}} \longrightarrow 115_{10}$$

NB : Pour la soustraction, il faut retrancher 6_{10} (ajouter 1001) au (x) codes erronés hors du résultat.

Le code excédent – 3 (excédent moins 3)

C'est un code utilisé dans les anciens ordinateurs. Ce n'est pas un code pondéré. Il consiste à prendre le code DCB du nombre et y ajouter 3. Par exemple, le code excédent – 3 de 0 est $0000 + 0011$, c'est-à-dire 0011.

Le code 84 – 2 – 1 (84 moins 2 moins 1)

Le code 84 – 2 – 1 permet l'utilisation de pondération négative suivant la séquence 8, 4, -2, -1. Le code 0110 représente le nombre $0 * 8 + 4 * 1 + 2 * 1 - 1 * 0 = 2_{10}$

Le code biquinaire ou code 5043210

Le code biquinaire est un code à 7-bits avec la possibilité de détecter des erreurs. En effet, le code de chaque chiffre comprend 5 zéros et deux 1 permettant d'évaluer les pondérations (un et un seul 1 dans les 2 positions de gauche, et un et un seul 1 dans les 5 positions de droite). La présence d'une erreur est détectée lorsque lors d'une transmission d'information, le récepteur signale des zéros en moins ou en plus.

Le code 2421

C'est un code pondéré auto-complémentaire, c'est-à-dire que le code du complément à 9 du chiffre binaire est le complément à 1 du code binaire du chiffre concerné. Les différents Codes sont donnés dans le tableau ci-après.

Chiffre décimal	DCB	Excédent - 3	84 – 2 - 1	2421	Biquinaire
	8421				5043210

0	0000	0011	0000	0000	0100001
1	0001	0100	0111	0001	0100010
2	0010	0101	0110	0010	0100100
3	0011	0110	0101	0011	0101000
4	0100	0111	0100	0100	0110000
5	0101	1000	1011	1011	1000001
6	0110	1001	1010	1100	1000010
7	0111	1010	1001	1101	1000100
8	1000	1011	1000	1110	1001000
9	1001	1100	1111	1111	1010000

Tableau 2 : Tableau récapitulatif des codes décimaux

N. B. : Les codes de 10 à 15 ne sont pas représentés.

Codes détecteurs d'erreurs

L'information binaire peut être transmise par l'intermédiaire de ligne ou par des ondes radio. La présence du bruit dans le médium de communication peut altérer la valeur des bits et les transformer de 0 en 1 ou vice versa. Ceci impose la gestion d'un code d'erreur utile à la détection des erreurs durant la transmission. Notons que seule l'existence d'une erreur est signalée mais celle-ci ne peut être corrigée. Si les erreurs sont fréquentes, alors il est suggéré de tester le système pour un mauvais fonctionnement.

Le principe de code détecteur d'erreur est basé sur la gestion d'un bit appelé bit de parité. C'est un bit extra qu'il faut inclure dans un message de manière à obtenir un nombre total de 1 pair ou impair. On parle alors de parité paire si le nombre de 1 est impair. Le tableau suivant présente un ensemble de messages sur 4 bits et le bit de parité à ajouter dans les cas pair et impair.

Message	Bit de parité paire	Bit de parité impaire
0000	0	1
0001	1	0
0010	1	0

0011	0	1
0100	1	0
0101	0	1
0111	0	1
1000	1	0
1001	1	0
1010	0	1
1011	0	1
1100	1	0
1101	0	1
1110	1	0
1111	1	0
	0	1

Tableau 3 : Exemple de code correcteur

Leur principe de fonctionnement est le suivant :

Lors de la transmission d'un message d'un point à un autre, le message est envoyé au départ dans un « générateur de parité » qui génère le bit de parité en fonction du type de parité. Le message est alors combiné avec ce bit et envoyé vers le point de destination. A l'arrivée, tous les bits reçus sont examinés à travers un « testeur de parité » qui vérifie le type de parité. Si le résultat du test de parité indique une parité différente de celle du départ, alors il y a erreur. Cette méthode permet de détecter une erreur ou un nombre impair d'erreur, les nombres pairs n'étant pas détectable.

Code 2 dans 5

Le code 2 dans 5 qui permet de coder chaque chiffre décimal sur 5 bits, avec 2 et seulement 2 bits ayant la valeur 1. ($129_{10} = 001010011011000$ code 2 dans 5).

Chiffre décimal	Code 2 dans 5
0	00011
1	00101
2	00110
3	01001
4	01010
5	01100

6	10001
7	10010
8	10100
9	11000

Code reflété ou code Gray

Rappelons que les systèmes numériques traitent des informations binaires. Toute information continue ou analogique doit être convertie en une information binaire avant d'être traitée par un système numérique, ordinateur par exemple. La conversion se fait au moyen d'un convertisseur analogique – numérique. Le code plus adapté dans un tel cas est le code reflété (reflecté). La règle utilisée dans la définition d'un tel code est dans le codage d'un nombre donné, seul un bit change d'un nombre à un autre. Le tableau suivant donne les codes des nombres de 0 à 15.

Nombre Décimal	Code Reflecté
0	0000
1	0001
2	0011
3	0010
4	0110
5	0111
6	0101
7	0100
8	1100
9	1101
10	1111
11	1110
12	1010
13	1011
14	1001
15	1000

2.11 Codes alphanumériques

L'ordinateur ne traite pas que des nombres. Il est possible de traiter les lettres de l'alphabet. Ceux-ci doivent être représentés sous forme binaire. D'où la nécessité de définir un code binaire pour les caractères.

Un code alphanumérique est donc un code binaire destiné à représenter les 10 chiffres décimaux, les lettres de l'alphabet, et un certain nombre de caractères spéciaux. On constate que le nombre total d'éléments à coder doit être au moins supérieur à 36 (10 + 36 pour les lettres majuscules), soit au minimum 6 bits ($2^6 = 64$). Différents codes ont alors été définis. Signalons les plus usités qui sont donnés sous forme de table :

- Le code alphanumérique à 6-bits ou code interne ;
- le code ASCII à 7 bits plus 1 bit de parité ;
- le code EBCDIC à 8-bit.

Utilisation du code A.S.C.I.I.

ASCII vient de « American Standard Code for Information Interchange ». C'est un code universellement adopté qui permet de représenter sur 7 bits l'ensemble des caractères alphanumériques, des symboles et des commandes de transmission.

Pour connaître le code binaire correspondant à un symbole, il suffit de le localiser dans le tableau (voir ci-après) et de noter dans l'ordre :

les bits de la colonne b7, b6 et b5

puis les bits de la ligne b4, b3, b2 et b1.

Par exemple le caractère 'A' a pour code (1000001)₂ soit (41)₁₆.

Remarque Le code ASCII est aujourd'hui toujours utilisé, mais il tend à être remplacé par le standard unicode. Si on regarde bien le tableau, on s'aperçoit que la définition du code correspond bien à la langue américaine. Par contre toutes les langues ayant des accents passent à la trappe, par conséquent la nôtre.

Mais que dire du â allemand qui n'y est même pas représenté. Quant aux langues qui n'ont pas d'alphabet latin (comme l'arabe ou le chinois), il n'y a aucun moyen de les coder.

Aujourd'hui cependant le standard 'unicode' code chaque caractère sur 16 bits, ce qui laisse 65536 possibilités. Cela en laisse assez pour coder toutes les langues du monde (ou presque) ainsi que des caractères spéciaux.

Tableau du code ASCII

				b7 →	0	0	0	0	1	1	1	1
				b6 →	0	0	1	1	0	0	1	1
				b5 →	0	1	0	1	0	1	0	1
b4	b3	b2	b1		0	1	2	3	4	5	6	7
0	0	0	0	0	NUL	DLE	SP	0	@	P		p
0	0	0	1	1	SOH	DC1	!	1	A	Q	a	q
0	0	1	0	2	STX	DC2	"	2	B	R	b	r
0	0	1	1	3	ETX	DC3	#	3	C	S	c	s
0	1	0	0	4	EOT	DC4	\$	4	D	T	d	t
0	1	0	1	5	ENQ	NAK	%	5	E	U	e	u
0	1	1	0	6	ACK	SYN	&	6	F	V	f	v
0	1	1	1	7	BEL	ETB	'	7	G	W	g	w
1	0	0	0	8	BS	CAN	{	8	H	X	h	x
1	0	0	1	9	HT	EM	}	9	I	Y	i	y
1	0	1	0	A	LF	SUB	*		J	Z	j	z
1	0	1	1	B	VT	ESC			K	[k	{
1	1	0	0	C	FF	FS	,	<	L	\	l	
1	1	0	1	D	CR	GS	.	=	M]	m	}
1	1	1	0	E	SO	RS		>	N		n	~
1	1	1	1	F	SI	US	/	?	O		o	DEL

NUL	Nul	DLE	Echappement transmission
SOH	Début d'en tête	DC1	Commande d'appareil
STX	Début de texte	DC2	Commande d'appareil
ETX	Fin de texte	DC3	Commande d'appareil
EOT	Fin de transmission	DC4	Commande d'appareil
ENQ	Interrogation	NAK	Accusé de réception négatif
ACK	Acquittement	SYN	Synchronisation
BEL	Sonnerie ou alarme	ETB	Fin de bloc de transmission
BS	Espacement arrière	CAN	Annulation
HT	Tabulation horizontale	EM	Fin de support
LF	Interligne	SUB	Substitution
VT	Tabulation verticale	ESC	Echappement
FF	Présentation de formule	FS	Séparateur de fichier
CR	Retour chariot	GS	Séparateur de groupe
SO	Hors code	RS	Séparateur d'article
SI	En code	US	Séparateur de sous article
DEL	Oblitération		

2.12 Compléments sur les nombres binaires

Le complément à b des nombres est également nommé complément vrai (C_v) et le complément à b-1 est aussi appelé complément restreint (C_r). Il existe une relation entre le complément vrai et le complément restreint :

Complément vrai de N = complément restreint de N + 1

$$C_v(N) = C_r(N) + 1$$

Pour trouver le complément à 1 des nombres binaires, il suffit de complémenter tous les chiffres binaires. C'est-à-dire remplacer les 0 par les 1 et les 1 par les 0.

Exemple : $C_r(1101_2) = 0010_2$

$$C_v(1101_2) = C_r(1101_2) + 1_2 = 0010_2 + 1_2$$

Pour les nombres en signes et valeur absolue le bit le plus à gauche est utilisé comme bit de signe. C'est-à-dire 1 pour les nombres négatifs et 0 pour les nombres positifs.

Exemple : Décoder en SVA

$$0101 = + (1 * 2^2 + 0 * 2^1 + 1 * 2^0) = +5$$

$$1011 = - (0 * 2^2 + 1 * 2^1 + 1 * 2^0) = -3$$

3 ELEMENTS DE MATERIEL INFORMATIQUE OU HARDWARE

3.1 Introduction

Rappelons que le matériel informatique ou hardware est l'ensemble des éléments physiques qui composent l'ordinateur, à savoir :

- L'unité centrale ou « boîte noire » ;
- Le clavier, la souris
- L'écran, l'imprimante ;
- Les lecteurs de disques ou de disquettes ;
- Les disques, les disquettes, les cassettes, les bandes ;
- Les modems ;
- Les câbles d'alimentation électriques et/ou de transmission des données ;
- Les microphones, les hauts parleurs, les webcams, écouteurs ;
- Etc.

On peut récapituler, une liste de matériels dans la figure suivante :



Figure 1 : Quelques matériels informatiques

C'est donc une définition assez large et met en évidence les éléments autour de l'unité centrale ou ordinateur ; ce dernier étant une machine électronique capable de réaliser des opérations sur de l'information à travers une séquence d'instruction appelée programme.

3.2 Structure générale de l'ordinateur

Définition

Les composantes principales de l'ordinateur sont

Une Unité Centrale de Traitement (UCT) ou processeur ou microprocesseur comprenant l'Unité Arithmétique et logique (ALU), l'Unité de Commande (UC), et des Registres ;

Une Unité de Mémoire Centrale (UMC) ;

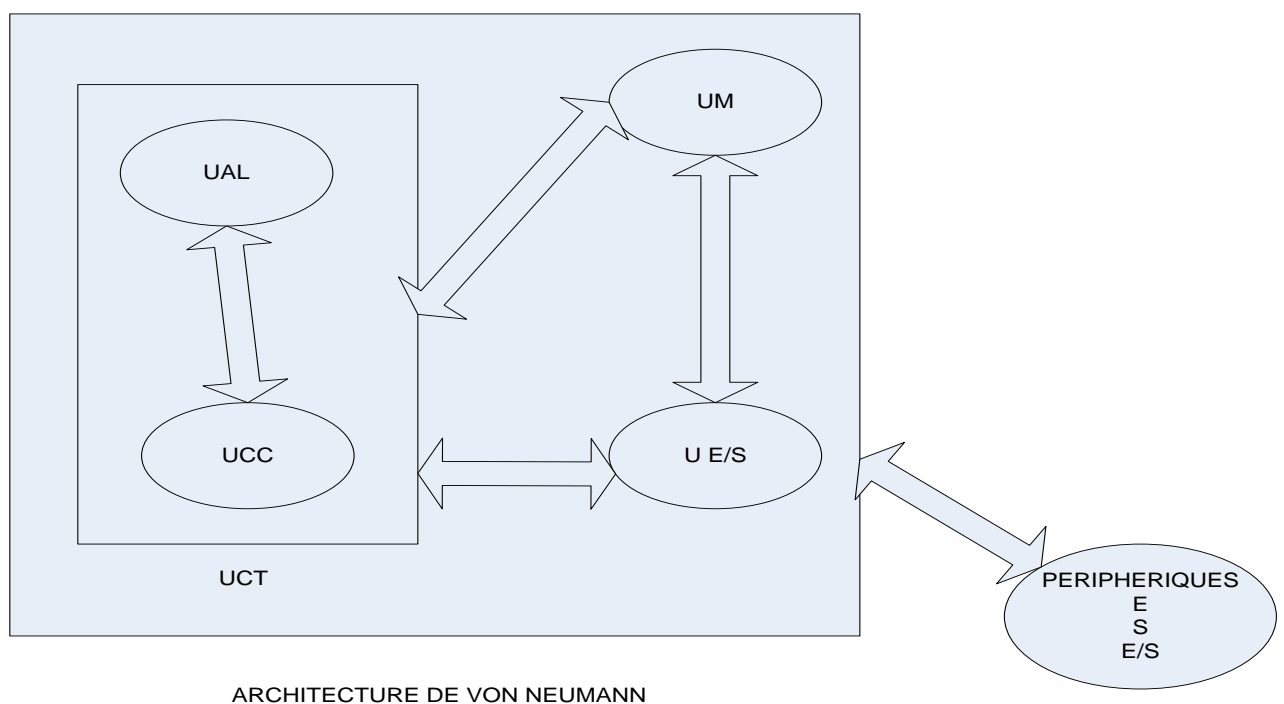
Des Unité d'Entrées (UE) permettant la connexion de périphériques d'Entrée (PE) ;

Des Unités de sorties (US) permettant la connexion de périphériques de Sortie (PS) ;

Des éléments de stockage de l'information (mémoires de masse ou mémoires auxiliaires).

Schéma général

La structure générale d'un ordinateur se présente comme ci-dessus :



ARCHITECTURE DE VON NEUMANN

Figure 2 : Architecture de Von Neumann

3.3 Présentation des unités de base

Unité de mémoire

L'unité de mémoire sert à conserver temporairement les informations utilisées par l'unité centrale. Deux types d'information peuvent être rangés (gardés) en mémoire. Ce sont :

Les données, c'est-à-dire les informations manipulées par le programme ;

Les instructions, à savoir les informations lues par l'unité de commande pour être exécutées.

La mémoire centrale constitue donc le point de transit de toute information à faire traiter par l'ordinateur. Elle est souvent construite avec un circuit intégré de type RAM (Random Access Memory). C'est une mémoire volatile, c'est-à-dire que les informations sont perdues à la suite d'une coupure de l'alimentation en courant de l'ordinateur.

La mémoire centrale mesure la capacité de stockage des informations de l'ordinateur. Elle s'exprime en bits ou en octets (8 bits) avec les multiples courants suivants :

Nom	Symbole	Valeur
Kilooctet	Ko	2^{10} octets = 1024 octets
Mégaoctet	Mo	2^{20} octets
Gigaoctet	Go	2^{30} octets
Téraoctet	To	2^{40} octets
Pétaoctet	Po	2^{50} octets
Exaoctet	Eo	2^{60} octets
zettaoctet	Zo	2^{70} octets
yottaoctet	Yo	2^{80} octets

N. B. : Le kilo informatique vaut 1024.

La fréquence d'un processeur est noté $F = 1/T$, avec F en Hz et T en s. Plus la fréquence augmente, plus le cycle de traitement T diminue. En conséquence ce temps est donné en sous- multiples.

Nom	Symbole	Valeur
seconde	S	1
déciseconde	Ds	10^{-1}
centiseconde	Cs	10^{-2}
milliseconde	ms	10^{-3}
microseconde	μ s	10^{-6}
nanoseconde	ns	10^{-9}

picoseconde	ps	10^{-12}
femtoseconde	fs	10^{-15}
attoseconde	as	10^{-18}
zeptoseconde	zs	10^{-21}
yoctoseconde	ys	10^{-24}

A côté de la mémoire centrale, il existe une mémoire à lecture seule appelée aussi mémoire morte, et qui contient le plus souvent les programmes réalisés par le constructeur. Elle est réalisée à l'aide de circuits intégrés de type ROM (Read Only Memory).

Unité Centrale de Traitement ou Processeur (UCT)

Le processeur est le cœur, le cerveau de l'ordinateur. Il contrôle le bon fonctionnement de l'ordinateur, d'une part, et coordonne les différentes activités à mettre en œuvre, d'autre part.

Le processeur ou microprocesseur est un circuit intégré à large ou à très large ultra large densité d'intégration (LSI : large scale integration ou VLSI : Very Large Scale Integration ou ULSI : Ultra Large Scale). Il contient les différents circuits électroniques nécessaires aux opérations à réaliser par l'ordinateur. C'est lui qui interprète les programmes et manipule les données pour produire les résultats désirés.

Il renferme deux unités fonctionnelles distinctes, à savoir l'unité de commande ou unité de contrôle et l'unité arithmétique et logique ou unité de traitement, et un ensemble de registres.

Il existe plusieurs fabricants de microprocesseurs dont les principaux sont intel et AMD. On peut citer Motorola, IBM, Sun Microsystem, Texas Instruments, Clones NEC, DEC, Compaq, Hewlett-Packard, Hitachi, ARM, etc.

L'exécution d'un programme dans une machine Von Neumann nécessite l'utilisation des 3 composants principaux UAL, Mémoire, et UC. Un logiciel, nommé système d'exploitation, contrôle le fonctionnement de ces 3 composants. Ce programme sera chargé dans la mémoire. Avant d'être chargé, le programme est stocké sur un périphérique de stockage secondaire (comme un disque). Dans cette organisation, la mémoire et les registres servent à la fois à stocker des instructions et des données. Son fonctionnement se déroule en 3 grandes étapes :

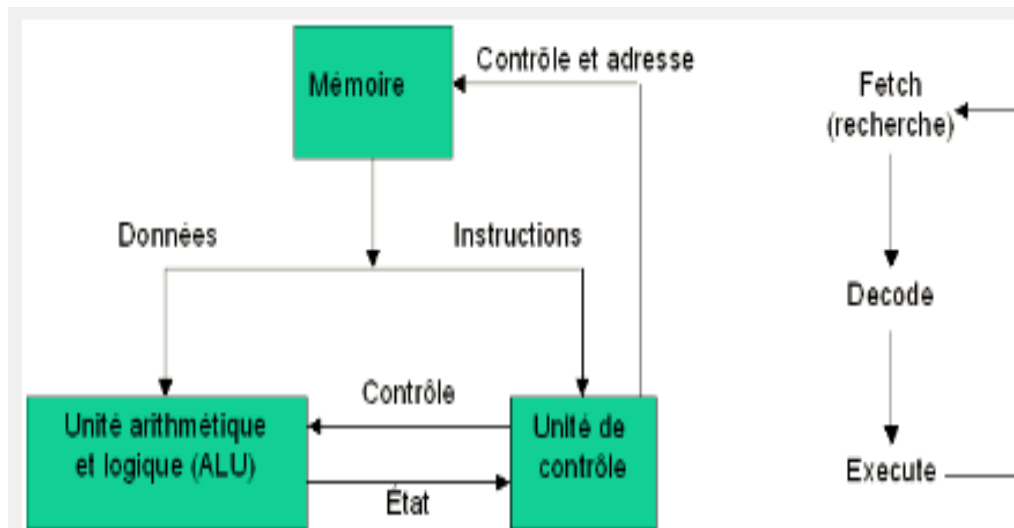


Figure 3 : Fonctionnement d'une architecture de Von Neumann

3.4 Carte mère

La carte mère est le support principal d'un ordinateur. Elle constitue le composant physique indispensable pour rassembler tous les composants d'un ordinateur. Pour un fonctionnement optimal, elle est alimentée de courant (secteur, pile) et ventilée pour éviter les échauffements de composants. Elle contient les bus, les chemins d'accès, les emplacements pour tous les composants électriques. Les bus permettent le déplacement des données, des commandes, des adresses entre les différents composants d'un ordinateur. Les chemins sont utilisés grâce à des contrôleurs (aiguilleurs). Les emplacements sont des endroits fixes pour clipser, enficher d'autres composants (circuits). Il est évident qu'une carte mère prévue à cet effet pourra accepter d'autres cartes (graphiques, réseaux,)

La carte mère s'adapte au processeur, à la mémoire vive (RAM), aux logements d'extension, à l'ensemble dissipateur thermique/ventilateur, à la puce du BIOS (**Basic Input Output System**), au jeu de composants chipset (interface E/S), aux interfaces de connexion, aux connecteurs internes et externes, aux différents ports et aux fils intégrés qui s'interconnectent aux composants de la carte mère.



Une carte mère d'un ordinateur

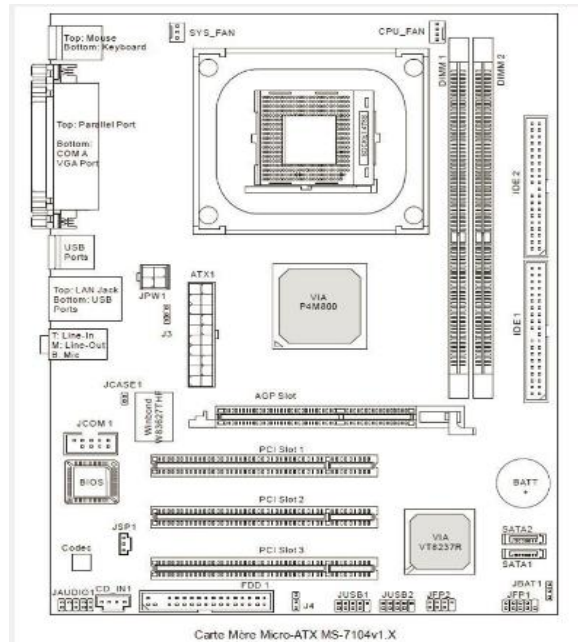


Schéma d'une carte mère

Figure 4 : Carte mère

Une carte mere est composéee de:

La carte mère est le principal constituant de l'ordinateur. C'est sur cette carte que sont connectés les autres éléments :

- Le microprocesseur (cerveau de l'ordinateur);
- La mémoire (RAM : *Random Access Memory*, la mémoire cache);
- Le disque dur, le lecteur de CD-ROM, le lecteur de disquettes;
- Les périphériques internes : carte de son, carte vidéo.

Microprocesseur

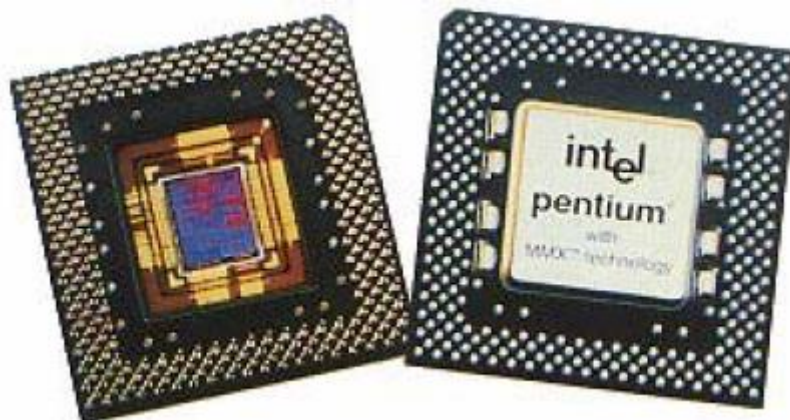
Le premier microprocesseur (Intel 4004) a été inventé en 1971. Depuis, la puissance des microprocesseurs augmente exponentiellement. Quels sont donc ces petits morceaux de silicium qui dirigent nos ordinateurs?

Le processeur (CPU) est le cerveau de l'ordinateur, c'est lui qui coordonne le reste des éléments, il se charge des calculs, bref, il exécute les instructions qui ont été programmées.

Toutes ces opérations sont des informations numériques. Les microprocesseurs utilisent des petits transistors pour faire des opérations de base; il y en a plusieurs millions sur un seul processeur.

Les principaux éléments d'un microprocesseur sont :

- une horloge qui rythme le processeur. À chaque TOP d'horloge, le processeur effectue une instruction. Ainsi plus l'horloge a une fréquence élevée, plus le processeur effectue d'instructions par seconde (MIPS : Millions d'instruction par seconde). Par exemple un ordinateur ayant une fréquence de 100 mégahertz (MHz) effectue 100 000 000 d'instructions par seconde;
- une unité de gestion des bus qui gère les flux d'informations entrant et sortant;
- une unité d'instruction qui lit les données, les décode puis les envoie à l'unité d'exécution ;
- une unité d'exécution accomplit les tâches données par l'unité d'instruction.



Le processeur travaille, en fait, grâce à un nombre très limité de fonctions comme des expressions logiques (ET, OU, NON, etc.), des expressions mathématiques (addition, soustraction, multiplication, etc.). Celles-ci sont directement câblées sur les circuits électroniques. Il est impossible de mettre toutes les instructions sur un processeur car celui-ci est limité par la taille de la gravure. Ainsi pour mettre plus d'instructions il faudrait un processeur ayant une très grande surface. Or le processeur est constitué de silicium et celui-ci coûte cher, et d'autre part il chauffe beaucoup. Le processeur traite donc les informations compliquées à l'aide d'instructions simples.

Mémoire cache

La mémoire cache permet au processeur de se «rappeler» les opérations déjà effectuées auparavant. Elle est utilisée par le microprocesseur pour conserver temporairement des instructions élémentaires. En effet, elle stocke les opérations effectuées par le processeur, afin que celui-ci ne perde pas de temps à recalculer des calculs déjà faits précédemment. La taille de la mémoire cache est généralement de l'ordre de 512 kilo-octets (Ko).

3.4 La mémoire vive

La mémoire vive, généralement appelée RAM (*Random Access Memory*, traduisez *mémoire à accès aléatoire*) ce qui signifie que l'on peut accéder instantanément à n'importe quelle

partie de la mémoire, permet de stocker des informations pendant tout le temps de fonctionnement de l'ordinateur. Par contre, cette mémoire est détruite lors de la mise hors-tension de l'ordinateur, contrairement à une mémoire de masse comme le disque dur qui garde les informations même lorsqu'il est hors tension. La mémoire vive contient les données et les instructions des applications en cours.

Mémoire morte (ROM)

Mémoire permanente contenant des microprogrammes enregistrés sur des puces électroniques de la carte mère (ou *mother board*) contenant les routines de démarrage du micro-ordinateur.

ROM (*Read Only Memory*, dont la traduction est *mémoire en lecture seule*) est appelée aussi parfois *mémoire non volatile*, car elle ne s'efface pas lors de la mise hors tension du système.

En effet, ces informations ne peuvent être stockées sur le disque dur étant donné que les paramètres du disque (essentiels à son initialisation) font partie de ces données vitales à l'amorçage.

Fentes d'extension

Les fentes ou «slots» d'extension sont des réceptacles dans lesquels on peut enficher des cartes.

Il en existe de trois types : les cartes ISA (les plus lentes fonctionnant en 16 bits), les cartes PCI (beaucoup plus rapides fonctionnant en 32 bits), et les cartes AGP (les plus rapides). Ils se branchent, grâce à des nappes, sur les broches prévues à cet effet sur la carte mère.

Disque dur

Le disque dur est l'organe du PC servant à conserver les données de manière permanente, contrairement à la RAM, qui s'efface à chaque redémarrage de l'ordinateur. Il a été inventé au début des années 50 par IBM.

Pour accéder de façon automatique à une case mémoire, il faut utiliser un compteur. Ce compteur est considéré comme un registre spécial : le CO (*Compteur Ordinal*). La valeur de ce compteur va désigner la case mémoire contenant la prochaine instruction à charger : cela correspond à l'adresse de la prochaine instruction. Le compteur ordinal contient une adresse, mais le processeur contient généralement d'autres registres qui permettent de stocker d'autres adresses utilisées au sein de notre programme assembleur. Ces registres sont appelés registres d'adresse.

Le système d'exploitation utilise les interfaces d'E/S pour extraire le programme du stockage secondaire et le charger dans la mémoire.

Unité de commande (UC)

L'unité de commande gère toute l'activité de l'ordinateur, et émet les ordres nécessaires à destination des autres éléments de l'ordinateur.

Unité Arithmétique et Logique (UAL)

L'unité arithmétique et logique assure la réalisation des opérations arithmétiques (+, -, *, /, etc.) et logiques élémentaires (et, ou, comparaison, test, etc.).

Registres

Les registres sont de très petites mémoires qui permettent le stockage local des données et des instructions.

Unité d'échanges ou unité d'Entrée/Sorties

La communication de l'unité centrale, de la mémoire avec le monde extérieur se fait par l'intermédiaire des unités d'échanges ou unité d'entrées /sorties. Ces unités servent de tampon entre l'ordinateur et les périphériques qui lui sont connectés. On distingue :

- Les unités d'entrées,
- Les unités de sorties,
- Les unités d'entrées et sorties

Unités d'entrée

Portes d'entrée de l'information dans l'ordinateur venant du monde extérieur, elles permettent d'introduire les programmes écrits dans un langage informatique tel que le PASCAL, les données. C'est à travers elles que sont connectés les périphériques d'entrée.

Unités de sortie

Porte de sortie de l'information en provenance de l'ordinateur sur le monde extérieur, elles reçoivent les données transformées ou résultats de l'ordinateur, et permettent la connexion des périphériques de sortie.

Unités d'entrée et sortie

Les unités d'entrée et sortie jouent à la fois les rôles d'entrée et de sortie ; on y connecte les périphériques dits d'entrée/sortie.

3.5 Périphériques

Un périphérique est un élément connecté à l'ordinateur afin d'établir une connexion entre ce dernier et le monde extérieur. Deux principales classes de périphériques existent :

Les périphériques réalisant des actions d'entrée et/ou sortie ;

Les périphériques réalisant des actions de stockage d'information appelés aussi mémoires de masse ou mémoires auxiliaires.

Il y a plusieurs connectiques tels que présentés ci-après.

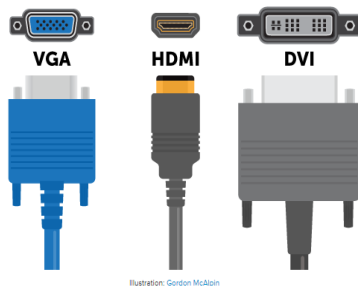


Figure 5 : Quelques connectiques

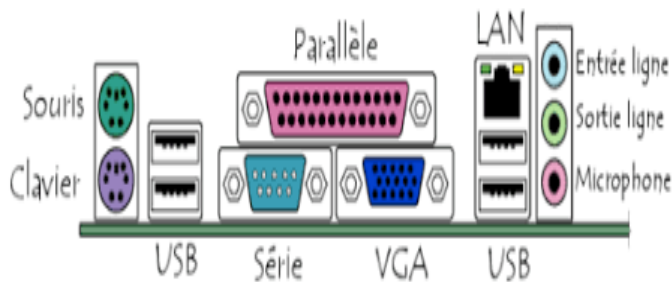


Figure 6 : Câbles et connecteurs



Figure 7 : Différents types de HDMI



Figure 8 : Différents ports USB



Figure 9 : Port RJ 45

Périphériques d'entrée - sortie

Périphérique d'Entrée

Un périphérique d'entrée est une unité qui permet d'introduire de l'information dans l'ordinateur. Sa fonction principale est la conversion de l'information naturelle (mot, phrase, symbole, nombre, etc.) en des codes binaires compréhensibles par l'ordinateur.

Exemple : Le clavier est le périphérique d'entrée le plus courant ; pour entrer l'information dans l'ordinateur, il suffit d'appuyer sur les touches désirées ; leur équivalent binaire est envoyé dans la mémoire centrale de l'ordinateur. On distingue plusieurs types de clavier ; les plus utilisés sont les claviers de type français (AZERTY) ou anglo-saxon (QWERTY).

Présentation du clavier :

Comme indiqué dans la fiche outil consacrée aux différents éléments d'un PC, la norme en France concernant les claviers est de type AZERTY. Il s'agit en fait des six premières touches alphabétiques de votre clavier. Un clavier azerty comporte 102 ou 105 touches de bases selon les modèles. Mais ce n'est pas pour autant que vous êtes limité à 102 caractères différents puisque chacune des touches peut être utilisée pour plusieurs caractères.

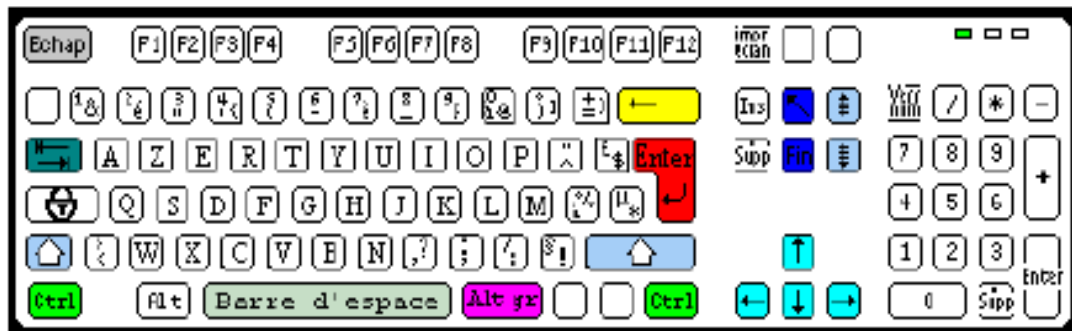


Figure 10 : clavier

Le caractère principal est accessible directement en pressant la touche 'Shift' et la touche en simultané.

Le caractère en bas à droite s'obtient en combinaison avec la touche 'Alt Gr' et la touche en simultané. Enfin le caractère en bas à gauche s'obtient directement en frappant la touche. Si le verrouillage majuscule (touche 'Caps Lock' représentée par un cadenas) est activé, la frappe d'une des touches vous donnera le caractère principal.

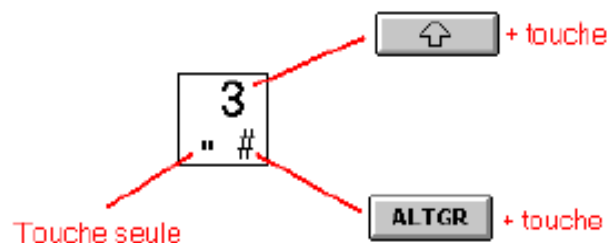


Figure 11 : Utilisation des touches du clavier

Périphérique de sortie

Un périphérique de sortie est une unité qui restitue de l'information provenant de l'ordinateur. Sa fonction principale est de convertir les codes binaires émis de l'ordinateur en une information naturelle compréhensible par l'être humain. Ce type de périphérique réalise donc l'opération inverse effectuée par le périphérique d'entrée.

Exemple 1 : L'écran est le périphérique de sortie le plus courant. Il sert à visualiser les informations. Les types d'écrans les plus répandus sont :

L'écran monochrome (une seule couleur) ;

L'écran couleur (environ 16 couleurs voir 256 couleurs) ;

L'écran graphique caractérisé par des petits points ou pixels et qui permet de représenter des figures, des courbes, des tableaux, des dessins, etc.

Exemple 2 : L'imprimante est un périphérique de sortie. On distingue les classes d'imprimantes suivantes :

- L'imprimante à impact qui donne des caractères semblables à ceux d'une machine à écrire. La qualité d'impression est très bonne mais c'est une imprimante très lente ;
- L'imprimante matricielle est une version améliorée de l'imprimante à impact. Les caractères sont « préfabriqués » et organisés sous la forme de matrice 7*7, ou 9*12, ou 10*9. Les améliorations sont : une vitesse d'impression élevée, une qualité d'impression moyenne, la possibilité d'imprimer des caractères graphiques.
- L'imprimante laser qui offre une qualité d'impression très supérieure (modification des charges électriques d'un tambour par un rayon laser filtré par le dessin de la feuille à imprimer).

N.B. : Des imprimantes ci-dessus, il existe des versions couleurs permettant d'imprimer du texte en couleur.



Figure 12 : Quelques imprimantes

Périphérique d'entrée et de sortie (Entrée/Sortie)

Un périphérique d'entrée/sortie est un périphérique qui réalise en même temps les opérations d'entrée et les opérations de sortie. Il convertit donc les informations naturelles en codes binaires et vice versa.

Exemple 1 : Le terminal est un périphérique composé :

- D'une unité d'entrée représentée par un clavier alphanumérique ou une souris ;
- D'une unité de sortie constituée par un écran ou une imprimante.

Il est généralement relié à un ordinateur central. Les informations sont introduites à l'aide du clavier et sont visualisées à l'écran. Les codes (ASCII) générés à partir du clavier sont transmis à l'ordinateur grâce à une interface d'entrée selon un mode de transmission sérielle (caractère après caractère) ou parallèle (par groupe de caractères en même temps). Les vitesses de transmission sont de l'ordre de 110 à 9600 bauds (le baud étant le nombre de caractère transmis par seconde).

Il existe des écrans à tube cathodique et des écrans plats. Les types les plus utilisés maintenant sont les écrans plats.

Dans les écrans plats, on a 4 technologies (seules les deux dernières sont utilisées en TV) :

DSTN : aussi appelé à matrice passive pour les anciens ordinateurs portables

TFT : appelé à matrice active qui permet une meilleure luminosité et un meilleur contraste. Elle augmente également l'angle de vision par rapport à l'affichage. C'est la technologie utilisée actuellement pour les écrans portables et pas les écrans bureautiques les moins chères.

LCD : permettant de nouveau un meilleur contraste mais également des dimensions supérieures, cette technologie est utilisée dans les écrans standards, avec une consommation un peu supérieure

LED : deux technologies qui sont dérivées du LCD et une troisième (OLED) est spécifique

Plasma : utilise l'illumination d'un gaz soumis à une charge électrique, pour les grandes dimensions.



Ecran plat

Ecran cathodique

Figure 13 : Quelques écrans

Exemple 2 : Le MODEM (MODulator –DEModulator) est un appareil qui permet l'établissement d'une liaison entre deux ordinateurs ou entre un ordinateur et un périphérique éloigné pour un transfert d'informations. La communication se fait par l'intermédiaire d'une ligne téléphonique. Le modulateur joue le rôle d'unité de sortie en convertissant les codes binaires issus de l'ordinateur en codes sonores. Ceux-ci sont transmis par téléphone à l'autre ordinateur où le démodulateur reconvertit ces codes sonores en codes binaires pour l'ordinateur récepteur.

3.6 Mémoires de masse ou mémoires auxiliaires

Définition

Une mémoire de masse est une mémoire externe de grande capacité destinée à emmagasiner une quantité importante d'information. Le traitement de cette information par l'ordinateur nécessite de la ramener dans la mémoire centrale de ce dernier. Les types de mémoires de masse sont :

- Le disque dur de capacité variable ;
- La disquette 3,5 pouces de haute densité (1.44MO) ;
- La disquette 5.25 pouces de haute densité (1.2MO) ;
- La bande ;
- Le Compact Dick ou CD.
- DVD

NB_ : A chaque type de mémoire de masse correspond un lecteur approprié.

Exemple de mémoire de masse

La carte perforée

La cassette standard de marché

Le disque, la disquette

La disquette a une capacité moindre que celle du disque. Elle se présente sous les formes courantes ayant des diamètres de $8,5''^{1/4}$, $3''^{1/2}$ pouces. Notons qu'un pouce vaut 2,54 centimètres.

NB. Le principe de conservation des informations est basé sur le concept de magnétisme : l'information est d'abord convertie en signaux électriques et est envoyée vers une aiguille aimantée. Un champ magnétique se crée alors à cet endroit. La direction du champ magnétique détermine l'information binaire à stocker (1 ou 0). Cette information y est permanente jusqu'à ce qu'elle soit détruite de manière électrique.

La bande magnétique

Cartouche

Disque optique numérique

Le disque optique numérique offre de meilleures capacités de stockage que celle rencontrées avec les mémoires magnétiques. Il utilise un rayon laser pour lire et écrire. On distingue les familles de disques optiques suivants :

Le CD-ROM (Compact Disc Rom) réalisés en usine ;

Le CD-WORM (Compact Disc write Once Read Only, CD réinscriptible une fois) ;

Le CD RW effaçable où l'on peut lire et écrire plusieurs fois.

Bulle magnétique

Quelques exemples de mémoires sont présentées ci-dessous.



Disquette 3"1/2

CD

DVD

Carte perforée

USB

Disque externe

Carte mémoire

Blu-ray

Figure 14 : Quelques matériels de stockage

3.7 Notion des trois bus

L'unité centrale et la mémoire sont physiquement reliées entre eux par un ensemble de fils ou lignes. Les unités d'échanges sont connectées au processeur de manière à transférer en parallèle un certain nombre de bits.

Définition de Bus

On appelle BUS l'ensemble de ces liaisons. La largeur du bus est le nombre de fils du bus (nombre de bits que l'on peut transférer en une seule fois). Le débit du bus est la vitesse de transfert du bus.

On distingue en général, trois types de bus externes :

- Le bus de données (Data Bus) ;
- Le bus adresses (Adress Bus) ;

- Le bus de commande ou bus de contrôle (Control Bus).

Types de Bus

Bus de Données (Data Bus)

Le bus de données (Data Bus) est un bus unidirectionnel qui permet le transfert des données de l'UCT au autres unités et vice versa. Sa largeur détermine le format ou la longueur des mots de l'ordinateur, c'est-à-dire le nombre de bits par mot.

Bus d'adresses (Address Bus)

Le bus d'adresses (Address Bus) est un unidirectionnel sur lequel le processeur fournit les adresses des unités auxquelles il désire avoir accès. Sa largeur détermine l'espace mémoire adressable. Par exemple, pour 16 lignes d'adresses, on peut accéder à 2^{16} mots mémoires.

Bus Contrôle (Control Bus)

. Le bus de commande ou bus de contrôle (control bus) est formé d'un ensemble de lignes utiles au bon fonctionnement de l'ordinateur. Citons par exemple, les lignes qui fournissent des informations relation aux opérations de lecture ou d'écriture dans la mémoire de celles indiquant l'état des unités d'échanges.

Schéma des trois Bus

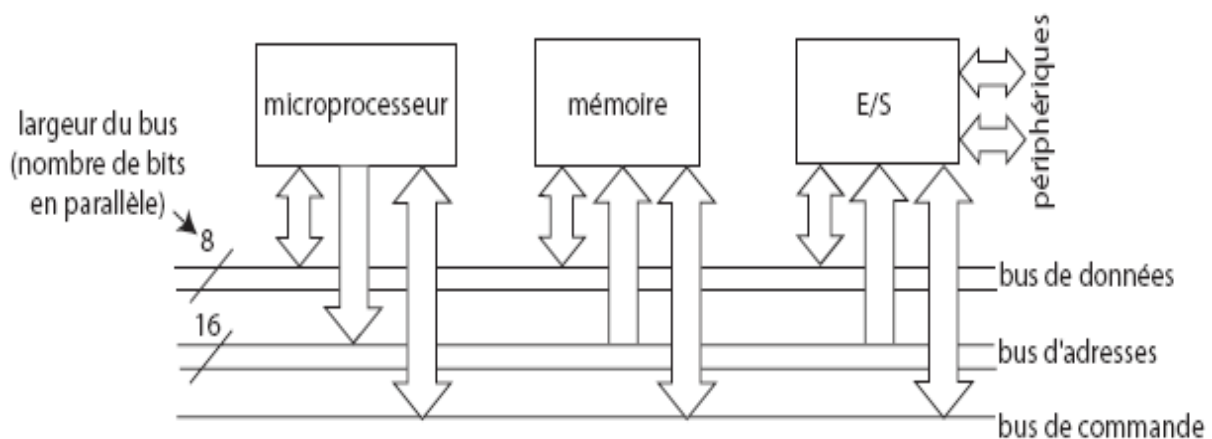


Figure 15 : schéma des trois bus

Remarque : les bus de données et de commande sont bidirectionnels, le bus d'adresse est unidirectionnel : seul le microprocesseur peut délivrer des adresses (il existe une dérogation pour les circuits d'accès direct à la mémoire, DMA).

4 ELEMENTS DE LOGICIEL INFORMATIQUE OU SOFTWARE

4.1 Niveau de logiciel

Le logiciel est l'ensemble des programmes destiné au fonctionnement du matériel et à la résolution de certains problèmes spécifiques. On peut distinguer 3 niveaux en partant du matériel. Chaque niveau assure un rôle et se réalise à partir de la base du logiciel de niveau inférieur.

Niveau 0 : Logiciel de Base ou système d'Exploitation

Le niveau 0 est le logiciel de base ou logiciel d'exploitation. C'est un ensemble de programme fournis pour son fonctionnement. Il a pour rôle de contrôler toute l'activité de l'ordinateur.

Niveau 1 : Logiciel outil et langage de programmation

Le niveau 1 est constitué par les logiciels – outils et langages de programmation. Ils permettent de traduire les fonctions (saisir de l'information, d'enregistrer, etc.,) en une forme utilisable par la machine. Des fonctions du système d'exploitation sont utilisées pour effectuer la conversion en instructions directement exécutables par la machine.

Niveau 2 : Logiciel d'application

Le niveau 2 est le logiciel d'application. Il est constitué par un ensemble de programmes destinés à la résolution d'un problème donné (comptabilité, gestion d'une école, gestion de personnel, ...). Ce logiciel est réalisé à l'aide d'un langage de programmation ou d'un logiciel –outil.

N.B : Les niveaux 0 et 1 constituent en fait le logiciel système.

Ces trois niveaux de logiciels sont implantés dans des matériels informatiques.

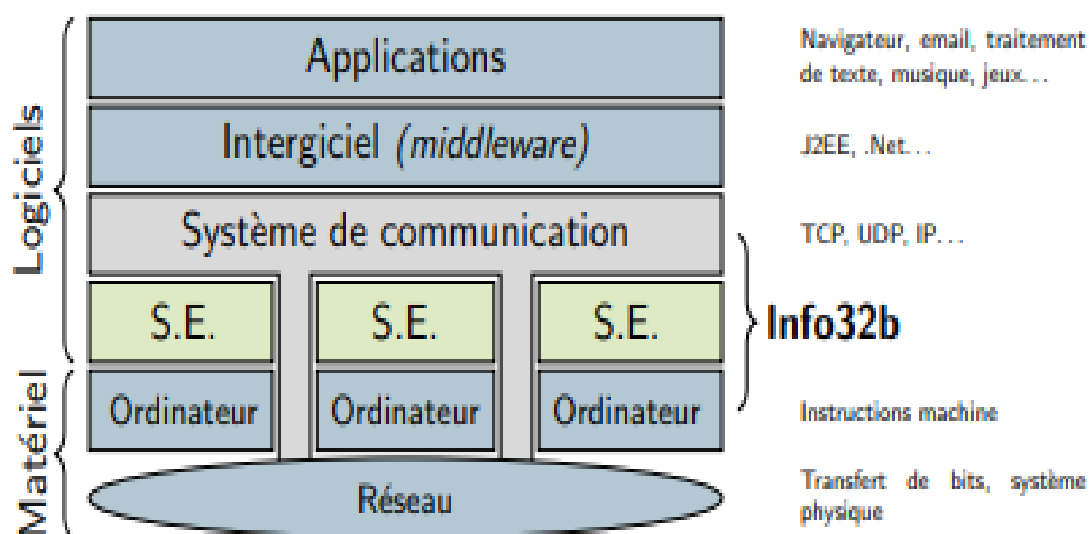


Figure 16 : Présentation de logiciels et de matériels

4.2 Logiciels systèmes

La réalisation d'un programme met en œuvre un ensemble d'outils logiciels. Cet ensemble constitue l'environnement de programmation. Le fonctionnement de cet environnement fait appel au système d'exploitation, et notamment au gestionnaire de fichiers. Les outils classiques ou utilitaires sont l'éditeur de texte, l'éditeur de liens, le chargeur, le traducteur, (assembleur, interpréteur, compilateur).

Utilitaire de base (logiciel-outil)

Editeur

Un éditeur de texte (text editor) est un programme interactif qui permet de créer et de modifier du texte à partir d'un clavier et de pouvoir le stocker dans un fichier. Les fonctions principales de l'éditeur sont :

La modification du texte

La suspension ou le remplacement de texte ;

La recherche de chaîne de caractères ;

La visualisation de texte ;

Le déplacement et le positionnement du curseur ;

Le déplacement de texte.

Pour le programme, l'éditeur sert donc à taper les programmes source ou bien à données utiles au programme. On distingue deux classes d'éditeur :

L'éditeur ligne, et

L'éditeur plein écran.

Remarquons que certains éditeurs favorisent la mise en œuvre de programmes et connus sous le nom d'éditeur syntaxique (éditeur PASCAL) alors que d'autres sont destinés au traitement de texte.

Chargeur

Le chargeur (Loader) lit le programme objet stocké sur une mémoire externe (bande, disque ou disquette) et le ramène dans la mémoire centrale pour son exécution.

Editeur de liens

L'éditeur de liens (Linker, Linkage Editor) est un programme qui réalise la combinaison de plusieurs programmes objet en un seul. Réaliser l'édition de lien.

Traceur (Débogueur)

Le traceur (Débogueur) permet de suivre l'exécution d'un programme et d'apporter les corrections nécessaires. Il facilite donc la mise au point du programme, c'est-à-dire la détection et la correction des erreurs.

Traducteur

Le traducteur (assembleur, interpréteur, compilateur) génère un code objet à partir du programme source écrit dans un langage informatique. Ce code peut être rangé sur une bande, une cassette, sur le disque dur ou la disquette.

4.3 Système d'exploitation

Généralités

Un système d'exploitation (S.E) ou Operating System (OS) en anglais est un ensemble de programmes fournis par le constructeur de l'ordinateur. Ces programmes contrôlent le fonctionnement de la partie matérielle. De plus, ils allouent à l'utilisateur les ressources nécessaires à l'exécution de son travail. Il est lié au type de processeur de la machine. On entend par ressources l'ensemble des éléments suivants :

- L'unité centrale,
- La mémoire centrale,
- Le disque dur, la disquette, les cassettes, etc.
- Les imprimantes,
- Les terminaux, les modems etc.

Le système d'exploitation est stocké sur disque. Au démarrage, la partie résidente, si elle n'est pas dans une mémoire ROM, est chargée dans la mémoire centrale. Une partie plus importante comprenant les autres fonctions reste stockée sur le disque, et introduite au fur et à mesure des besoins.

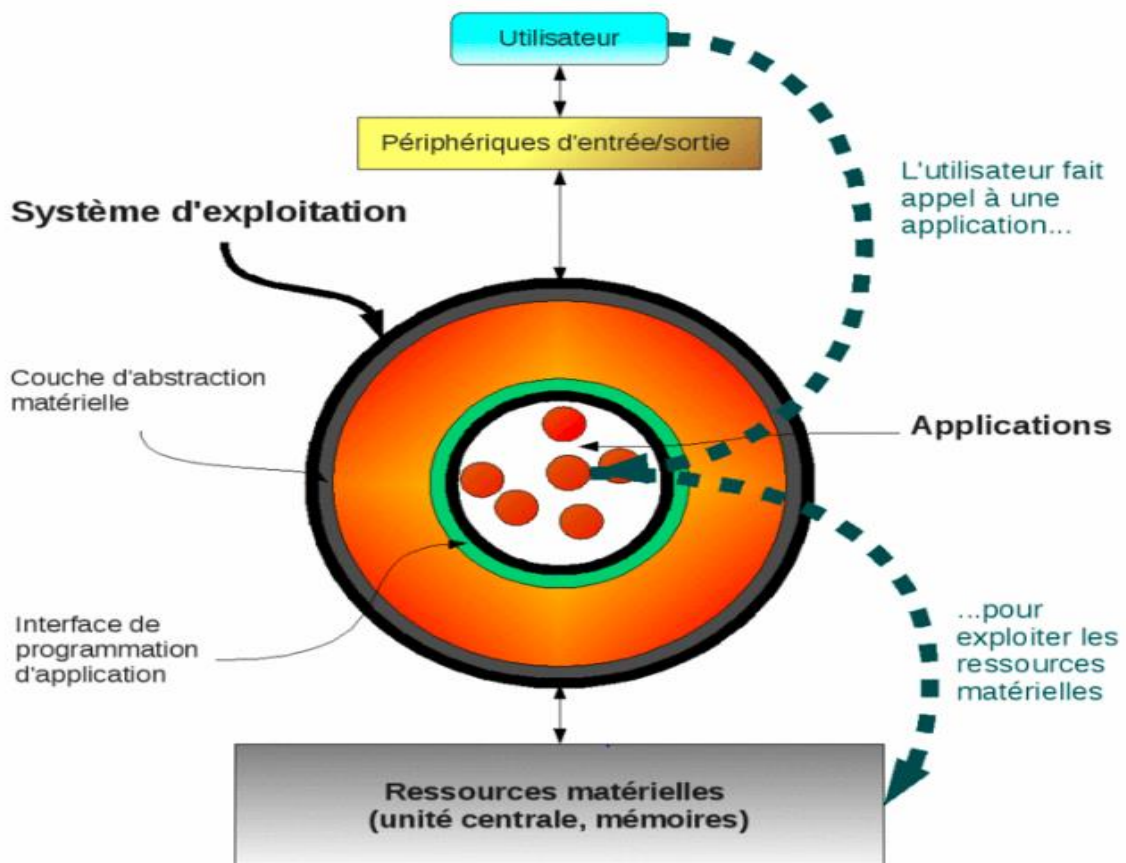


Figure 12 : Relations dans un système d'exploitation moderne.

Les types de systèmes d'exploitation

On distingue les différents types de S. E. ci-dessous :

- Le moniteur ;
- Le système Mono-utilisateur ou mono-tâche ou système par lots ou par train ;
- Le système Multitâches ;
- Le Système Multi-Utilisateurs ou système à temps Partagé ;
- Les systèmes actuels.
- Les systèmes distribués

Moniteur

Le moniteur est le système d'exploitation le « plus » primaire. Il ne prend en compte que quelques opérations courantes : charger un programme, écrire ou réaliser une lecture dans une adresse donnée, exécuter un programme...

Système mono-utilisateur

Le système mono-utilisateur ou système par lots ou par train (concept de la monoprogrammation) est un système simple et dédié à un seul utilisateur. Il permet donc à un seul utilisateur d'exécuter un seul programme.

Exemple : MS/DOS, FLEX, CP/M (Control Program for Microprocessors) DOS sous windows

Système multi-tâche

Le système Multi (concept de multiprogrammation) offre la possibilité d'exécuter plusieurs commandes ou tâches simultanément. Il permet, du point de vue de l'utilisateur, d'utiliser simultanément plusieurs programmes (modification de texte pendant l'impression, exécution d'un programme interactif : écran, clavier, traitement batch (grosse édition).

Exemple: Mac-Intoch, GEM, OS2 (Operating System 2)

Système multi-utilisateurs

Le système Multi- Utilisateurs ou système à Temps partagé permet à plusieurs personnes de travailler sur la machine à partir de terminaux différents ; chaque utilisateur a l'impression d'avoir à lui seul toutes les ressources de l'ordinateur. Il se rencontre dans les architectures d'informatique distribué (gros ordinateurs et terminaux et informatique) et réparties.

Exemple : UNIX, XENIX, VM, VMS, OS-9

Systèmes actuels

Les systèmes actuels sont des systèmes qui incorporent des modules spécifiques en vue de prendre en compte des fonctions de communication, de répartition, de transaction, etc.

Exemple : Net work Operating Système (NOVEL), distributed Operating Système, système à temps réel (OS-9), Système transactionnels (Banques et Compagnies de Transport).

Systèmes distribués

La baisse des prix du matériel informatique a permis, dans les années 1990, la création de systèmes informatiques composés de plusieurs ordinateurs, et donc plusieurs processeurs, plusieurs mémoires, et de nombreux périphériques. Un système distribué permet le partage des ressources entre les ordinateurs. Un utilisateur d'un ordinateur bon marché peut se servir de ressources coûteuses existant sur un autre ordinateur.

Mach, Amoeba, *Andrew*, Athena, et *Locus* sont des systèmes d'exploitation distribués. Ils ont tous été développés par des universités.

Principales fonctions

Les principales fonctions d'un système d'exploitation sont :

La gestion des travaux

Cette fonction est réalisée par le moniteur, et consiste à :

- La gestion de la succession des travaux

- La gestion de la mémoire en protégeant les données dans la partie (partition) de programme où elles sont traitées.

La gestion des entrées – sorties

La gestion des entrées/sorties gère les liaisons du processeur avec la mémoire centrale et les périphériques. Elle peut se résumer en :

- Envoi des ordres aux différents éléments ;
- Envoi ou réception de données ;
- Contrôle des opérations.

NB : Dans le cas des périphériques, cette gestion s'accompagne d'une gestion d'interruption par les messages (ordres ou données) en provenance des périphériques d'entrée (clavier ou souris).

Dans le cas de système complexe (multi-utilisateurs), les messages sont stockés dans des files d'attente (queues en anglais) de manière provisoire.

La gestion des fichiers

La gestion des fichiers est effectuée à l'aide de deux fonctions : une fonction de répartition du stockage des fichiers et une d'organisation hiérarchique logique sur des répertoires, d'une part, et des utilitaires de gestion de fichiers d'autre part.

La fonction de répartition de stockage des fichiers (programmes et données) a pour rôle d'optimiser le stockage entre la mémoire centrale (partie résidente) et les mémoires de masse (disque magnétiques et optiques).

La deuxième fonction réalise un classement logique (spécifie un emplacement physique) des fichiers à l'intérieur de répertoires hiérarchisés. Dans le système Multi-Utilisateurs, cette fonction peut attribuer des répertoires à des utilisateurs (users) avec des droits d'accès.

Les utilitaires de gestion de fichiers sont : effacement et copie des fichiers, création et suppression de fichiers, affichage des caractéristiques des fichiers et des répertoires, ...

Gestion des communications

Cette gestion inclut les communications du système avec l'utilisateur, les logiciels d'application, les langages de programmation et les logiciels outils.

Dans le cas de l'utilisateur, le système communique de façon inactive à travers des commandes qu'il interprète et exécute. La communication avec les logiciels dépend du type de ceux-ci (langages ou logiciels outils). Les traitements batch utilisent des commandes en différé (langage de commande ou Job Control Language : JCL) utilisant un fichier des travaux dans l'ordre des exécutions successives des différents programmes (jobs) à exécuter.

4.4 Langages de programmation et leurs traducteurs

Un langage est un moyen de communication entre les peuples. Il est souvent constitué d'un ensemble de symboles verbaux ou écrit, d'expressions, qui permettent d'échanger une idée ou une information. En informatique, un langage de programmation est l'ensemble de caractères, de symboles et de règles syntaxiques utilisées dans l'écriture des programmes. On distingue trois classes de langages de programmation :

Langages non évolués

Ce sont les langages proches de la machine. Ils se subdivisent en langage machine et en langage d'assemblage ou assembleur.

Langage machine

Le langage machine ou binaire est le langage de l'ordinateur. Il est composé des chiffres binaires 0 et 1. Tout programme écrit dans ce langage est sous forme binaire. Il est lié au processeur ou microprocesseur utilisé par l'ordinateur. Les adresses, les données, les instructions sont en binaire. Il est évident que l'utilisation d'un tel langage est fastidieuse. Les programmes y sont complexes. Cette complexité peut être réduite par l'utilisation des systèmes associés au système binaire (octal et hexadécimal).

Ici, le programmeur travaille directement avec l'unité centrale. La manipulation des registres, des entrées – sorties, et autres est à la portée de l'utilisateur.

Langage d'assemblage ou assembleur

Le langage d'assemblage ou assembleur est propre à chaque processeur. Il est formé d'instructions sous forme mnémotique (abréviations de 2 ou 3 lettres) correspondant aux opérations usuelles (lecture, écriture,) d'un processeur. Ce langage se veut une amélioration du langage machine. L'accès direct aux registres, aux entrées sorties, etc. est conservé. Par exemple, l'action de charger l'accumulateur B avec le nombre décimal 10 est exprimée en langage d'assemblage 6800 par LDA B # 10.

Tout programme en langage d'assemblage doit être traduit en langage machine à l'aide d'un assembleur. L'assembleur est un programme spécial qui permet de traduire en langage machine un programme écrit dans un langage d'assemblage. Le programme obtenu après traduction du programme source (programme de départ) est appelé programme objet. C'est ce dernier qui sera exécuté par l'ordinateur.

Langages évolués

La connaissance du microprocesseur et de son architecture est nécessaire pour maîtriser le langage d'assemblage ; tel n'est pas le cas pour les langages évolués.

Les langages évolués utilisent des instructions très proches du langage naturel. Ils sont orientés vers des applications spécifiques et ne dépendent pas de la machine utilisée. On distingue les langages interprétés et les langages compilés.

Langages interprétés : interpréteur

Tout programme écrit en un langage évolué dit interprété doit être traduit en langage machine par un « interpréteur ». Un interpréteur est un programme spécial qui permet de traduire ligne à ligne le programme source. Il doit résider dans la mémoire centrale pendant la traduction.

Exemples: BASIC: Beginners all-purpose Symbolic Instruction Code.

Langages compilés : compilateur

Tout programme écrit en un langage évolué dit compile doit être traduit en langage machine par un « compilateur ». Un compilateur est un programme spécial, proposé sur disquette ou tout autre support, qui traduit les programmes en langage machine. A la fin de la saisie du programme source, il est fait appel au compilateur (le ramener dans la mémoire centrale) pour la traduction en langage machine. Le programme objet est ensuite exécuté.

Exemple : PASCAL, langage c, PROLOG, FORTRAN : FORMula TRANslator (orienté scientifique) COBOL : Common Business Oriented Langage (orienté gestion), BASIC : Beginners All-purpose Symbolic Instruction Code.

Langages utilisateurs (très proche de l'utilisateur)

Ce sont des langages très proches du langage naturel, accessibles à des non informaticiens. On peut citer le langage d'interrogation de base de données SQL, ou les langages de quatrième génération (L4G) qui permettent l'obtention rapide des statistiques complexes : NOMAD).

4.5 Logiciels d'application

Généralités

Les logiciels d'application possèdent les caractéristiques suivantes :

Utilisation de menus ou d'icônes remplaçant les commandes de programme ;

Résolution d'un problème spécifique (traitement de texte, tracé de schéma, calcul : tableur, création et mise à jour de base de données, consultation de base de données (QBE), générateur d'écran, etc.) ;

Possibilité de réalisation de programmes.

Classification

Les principaux types de logiciels – outil sont :

Le traitement de texte

Le traitement de texte, ensemble de programmes facilitant la mise in œuvre des activités de bureau (note de service, courrier, rapport d'activité, compte rendu de réunion ou mission, etc.). Ces logiciels permettent :

De saisir du texte (frappe, modification et suppression de caractères) ;

De mettre en forme (mise en relief, tabulations, retraits, paragraphes, insertion de tableaux) ;

De déplacer du texte (couper, copier, coller) ;

De gérer des documents (recherche, chargement, affichage, édition, etc.) ;

D'imprimer tout ou partie de document.

Exemple : WORD, WORDERFECT, LATEX.

Tableurs

Le tableur permet d'organiser les informations sous forme de lignes et de colonnes. Des opérations de calcul ou de recherche y sont incorporées ;

Exemple : EXEL, LOTUS1-2-3.

Logiciels de construction de schémas

Les logiciels graphiques permettent de représenter les données numériques sous la forme de graphique (courbes, histogrammes, etc).

Exemple : GRAFTALK.

Logiciels de construction de schémas

Les logiciels de construction de schémas ou dessin. Ces logiciels permettent :

Le tracé de figures géométriques élémentaires (lignes, rectangle, ellipses, etc.) ;

L'association entre ces figures simples pour obtenir des figures complexes :

La possibilité d'introduire du texte dans ces schémas.

Systèmes Gestion de Bases de Données

Notion de base de données

Les systèmes de Gestion de fichiers (SGF) qui ont présentés des limites ont fait place aux bases de données dans les années 60.

Une BASE DE DONNÉES est une collection de données opérationnelles enregistrées sur des supports adressables et utilisés par les systèmes application (programmes) d'une organisation particulière.

En outre la collection de données est structurée indépendamment d'une application particulière. Elle est cohérente de redondance minimale et accessible simultanément par plusieurs utilisateurs.

Définition d'un Système de Gestion de Base de Données (SGBD)

Un SGBD est ensemble d'outils logiciels permettant la création et l'utilisation de Base de Données.

Exemple : ACCESS 2000 – ORACLE – FOX PRO.

Objectifs de l'utilisation d'une BASE DE DONNÉES et d'un SGBD

Un système d'informations peut être utilisé sans outil spécifique. On peut alors se demander quels sont les objectifs et les avantages de l'approche SYSTÈME DE GESTION DE BASE DE DONNÉES par rapport au fichier classique. La réponse tient en 9 points fondamentaux :

Indépendance physique

Indépendance logique

Manipulable par les non informaticiens

Administration centralisée des données

Non redondance des données

La cohérence des données

Les accès concurrents (partageabilité des données)

La sécurité de fonctionnement

Efficacité des accès aux données

Typologie des SGBD

Il existe actuellement 5 grands types de Bases de Données :

Le modèle hiérarchique et les systèmes de gestion de bases données hiérarchiques

Exemple : -IMS de IBM

Bases de données réseau ou Codasyl

Exemple : TOTAL, IDSII.

Les bases relationnelles

A l'heure actuelle se sont les plus utilisées. Les données sont représentées en table ou relation. Elles sont basées sur l'algèbre relationnelle et un langage de manipulation (SQL)

Exemple : Oracle 9.i, Access 2002, foxpro, Sql Serveur , etc.

Les bases déductives

Les données sont aussi représentées en table (prédicat). Le langage d'interrogation se base sur le calcul des prédicats et la logique du 1^{er} ordre. Le langage utilisé est Prolog ou datalog.

Les bases de données orientées objet

Les données sont représentées en temps qu'instance de classe hiérarchisée. Chaque donnée est active et possède ses propres méthodes d'interrogation et d'affectation. Elles permettent de prendre les applications nécessitant le recours à des objets complexes (sons, textes, images etc.)

Exemple : O2

N.B : la répartition du parc Des Systèmes De Gestion De Base De Données (SGBD) n'est pas équitable entre ses 5 types de bases. 75% sont relationnelles, 20% réseaux, 5% restant étant partagés étant partagés entre les bases déductives et objets.

Fonctions d'un système de gestion de base de données

Les composantes d'un environnement B.D.

Les données stockées dans les bases de données modélisent les objets du monde réel ou des relations entre ses données. La Base de Données peut être manipulée au moyen d'interface-utilisateur, de programme et de requête.

Architecture opérationnelle des systèmes de gestion de base de données

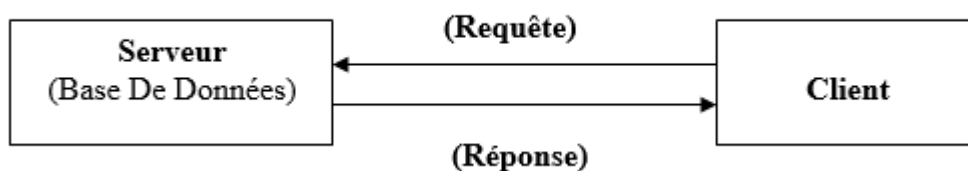


Figure 13 : Architecture client-serveur

NB : Dans la gestion quotidienne d'une Base De Données, C'est l'administrateur de B.D qui est responsable du bon fonctionnement de cette Base De Données.

Logiciels intégrés

Ce sont des paquetages (ensemble de modules distincts de programmes) utilisés dans la résolution de problèmes. Ils intègrent des fonctions de traitement de texte, de tableur, de grapheur, et de gestion de base de données.

Exemple : FRAMEWORK, LOTUS 1-2-3-, SYMPHONY, etc.

Logiciels mathématiques

En mathématiques, on peut citer les logiciels de statistiques, de simulation ou de calcul numérique.

Exemple : STAT-ITCF, MATHEMATICA, MATLAB, WIN'DESINGN.

Il existe plusieurs logiciels dans de nombreux domaines.

Présentation de quelques domaines d'applications et leurs logiciels spécifiques

Domaines	Spécifications	Exemples de logiciels
Texteur	Traitement de texte et bureautique	Word, word perfect, kword
Texteur scientifique	Traitement de texte scientifique	Latex, scientific word

Tableur	Tableaux, fonctions, graphique	Excel, lotus
PréAO	Présentation Assistée par ordinateur	PowerPoint
OCR	Reconnaissance optique de caractères	GOOCR, readiris, moredata
Compression fichiers	Compression et décompression de fichier	Winzip, winrar, QuickZip, TugZip
PAO	Publication Assistée par ordinateur	Publisher, Scribus, PageMaker
SGBD	Système de gestion de base de données	Access, oracle, MySQL
SIG	Géolocalisation, cartographie	ArcGIS, Mapinfo, Arcview
CAO	Construction Assistée par ordinateur	Autocad, FreeCad, Inventor
DAO	Dessin Assisté par ordinateur	Visio, DraftSight, TurboCAD
Gestion de Projets	Ordonnancement, planification et suivi	MS Project, GantProject, OpenProject
Client messagerie	Réception, envoi message, travail collaboratif	Outlook, Kmail, NetMail, WinMail
Comptabilité	Gestion comptable	Ciel Compta, SAGA, SAGE, Trompo
Architecture	Plan, design, métrique des matériaux	Archi3D, ArchiCAD
Photographie	Traitement de photo numérique	Photoshop, Gimp, UFRaw
Ludiciel	Jeux, aventures	FIFA, FlySimulator
Téléphonie sur IP	Téléphonie sur internet	Skype, Yahoo Messenger, kphone
Didacticiel	Logiciel éducatif	Express Scrib, visia math
Gravure	Gravure sur disque optique	Nero, Clone CD, XPburner
Multimédia	Lecture de fichiers multimédia	VLC, DragonPlayer, Totem
Statistiques	Calculs statistiques	Statview, R, SPSS, SPAD
Mathématiques	Calculs scientifiques	Matlab, scilab, mathématica, R

Navigateur	Affichage de page web	IE, Google Chrome, Opéra
Antivirus	Protection contre les virus	Avast, Essential security
Antispyware	Logiciel anti-espion	Spyware Terminator, Windows Defender, Spy Hunter
PGI	Progiciel de Gestion Intégrée	Adempiere, ERP5, OpenERP
Physique	Calculs et constructions en physique	CodeAster, CodeSaturne
Partage de fichiers	Partage de fichier sur internet	BitTorrent, eMule, Vuze, Deluge
Lecture et Ecriture de Fichier	Format de fichier	Ghostscript, PDFCreator, Xpdf
EDI	Environnement de Développement Intégré	CodeBlocks, eclipse, NetBeans

5 BIBLIOGRAPHIE

Livres :

Architecture et technologie des ordinateurs (Dunod) – *Paolo Zanella et Yves Ligier*

Technologie des ordinateurs et des réseaux (Dunod) – *Pierre-Alain Goupille*

Les microprocesseurs, comment ça marche ? (Dunod) – *T. Hammerstrom et G. Wyant*

Cours Web :

Architecture Avancée des ordinateurs (cours Supelec Rennes) – *Jacques Weiss*

Architecture des ordinateurs (cours IUT GTR Montbéliard) – *Eric Garcia*

Architecture des ordinateurs (cours IUT SRC Marne la Vallée) – *Dominique Présent*

Architecture des ordinateurs (cours Université Franche Compté) – *Didier Teifredo*

Architecture des ordinateurs (cours IUP STRI Toulouse)

Architecture des ordinateurs (cours Université de Sherbrooke) – *Frédéric Mailhot*

Architecture des ordinateurs (cours Polytechnique) – *Olivier Temam*

Architecture des ordinateurs (cours IUT GTR Villetaneuse) – *Emmanuel Viennet*

Architecture des ordinateurs (cours DEUG MIAS) – *Frédéric Vivien*

Architecture des Ordinateurs (cours Licence Informatique USTL) – *David Simplot*

Architecture des machines et systèmes Informatiques – *Joëlle Delacroix*

Architectures des processeurs (cours DEUST Nancy) – *Yannick Chevalier*

Architecture des systèmes à microprocesseurs – *Maryam Siadat et Camille Diou*

Architecture des systèmes à microprocesseurs (cours IUT Mesures Physiques) – *Sébastien Pillement*

Architecture Systèmes et Réseaux (cours DEUG 2ième année) – *Fabrice Bouquet*

Carte graphique (ENIC) – *Julien Lenoir*

Cours de réseau (cours EISTI) – *Bruno Péant*

Cours de réseaux (cours Maîtrise Informatique Université Angers) – *Pascal Nicolas*

Du processeur au système d'exploitation (cours DEUST Nancy) – Yannick Chevalier

Les réseaux : introduction (DESS DCISS) – Emmanuel .Cecchet

Les systèmes informatiques (cours CNAM) – Christian Carrez

Sites web :

Fonctionnement des composants du PC

<http://www.vulgarisation-informatique.com/composants.php>

Cours d'initiation aux microprocesseurs et aux microcontrôleurs

http://www.polytech-lille.fr/~rlitwak/Cours_MuP/sc00a.htm

Architecture des ordinateurs – Université Angers

<http://www.info.univ-angers.fr/pub/richer/ens/deug2/ud44/>

Les docs de Heissler Frédéric

<http://worldserver.oleane.com/heissler/>

X-86 secret

<http://www.x86-secret.com/>

Le cours hardware d'YBET informatique

<http://www.ybet.be/hardware/hardware1.htm>

Informa Tech

<http://informattech.online.fr/articles/index.php>