

## SOMMAIRE

### Chapitre 1 : Environnement matériel et logiciel d'un système informatique

<b>I. Structure et fonctionnement d'un ordinateur .....</b>	<b>2</b>
<b>1. Composants de base.....</b>	<b>2</b>
a) Unités d'entrée.....	3
b) Unités de sortie .....	3
c) Unités de stockage (Mémoires auxiliaires) .....	3
d) Un système de bus .....	3
e) Ports de communication .....	4
<b>2. Unité centrale de traitement CPU : Le processeur.....</b>	<b>4</b>
<b>3. Mémoire centrale.....</b>	<b>5</b>
<b>4. Système binaire.....</b>	<b>6</b>
<b>5. Représentation des caractères dans la mémoire.....</b>	<b>8</b>
a) Le code ASCII (7 bits) .....	8
b) Le codage ISO 8859 (8 bits) .....	9
c) L'Unicode (16 bits ou 32 bits).....	9
<b>6. Représentation des nombres dans la mémoire .....</b>	<b>9</b>
a) Représentation d'un entier naturel.....	10
b) Représentation d'un entier relatif.....	10
c) Représentation d'un nombre réel .....	12
<b>7. Codage des images .....</b>	<b>13</b>
<b>II. Système d'exploitation d'un ordinateur.....</b>	<b>14</b>
<b>1. Définition et exemples de systèmes d'exploitation.....</b>	<b>14</b>
<b>2. Principales fonctionnalités.....</b>	<b>14</b>
a) Gestion du dialogue entre la machine et l'utilisateur .....	14
b) Gestion des fichiers et répertoires.....	15
c) Gestion des périphériques .....	15

# Environnement matériel et logiciel d'un système informatique

## I. Structure et fonctionnement d'un ordinateur

### 1. Composants de base

Un ordinateur est un dispositif électronique qui prend les données brutes en entrée de l'utilisateur et **traite ces données** sous le contrôle d'un jeu d'instructions (programme) pour aboutir au résultat désiré par l'utilisateur.

Un ordinateur est en général constitué d'une unité centrale entourée de périphériques d'entrée et de sortie. Une information entre dans l'unité centrale, elle est traitée puis ressort vers l'utilisateur. Il fonctionne suivant le principe Entrée-Traitement-Sortie.

La structure de base d'un ordinateur contient les éléments fondamentaux suivants :

- Unités d'entrée
- Unités de sortie
- Unité centrale de traitement CPU
- Mémoires centrales de traitement
- Unités de stockage
- Un système de bus

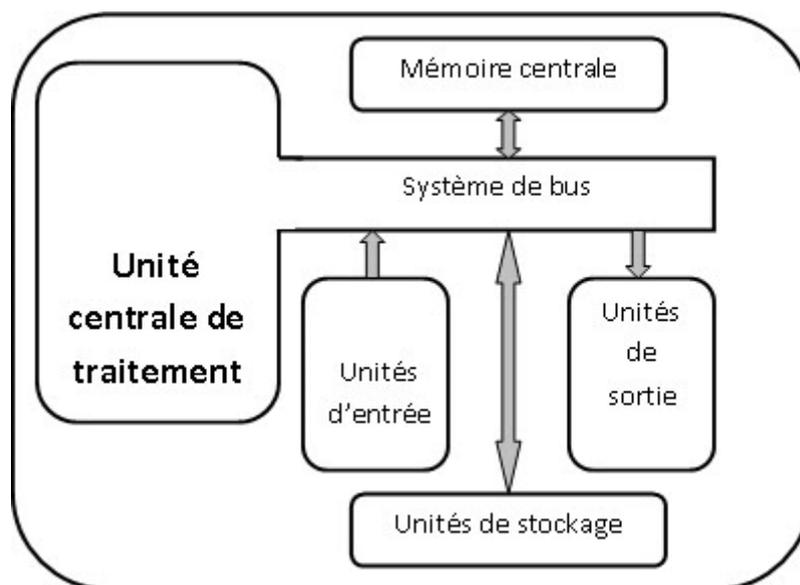


Schéma fonctionnel d'un ordinateur

### a) Unités d'entrée

Ces unités permettent à l'utilisateur d'entrer des données, des commandes et des programmes qui seront gérés par l'unité de traitement. Ces données, quelle que soit leur forme, sont traduites en configurations identifiables par l'ordinateur.

Exemples : Clavier, Souris, Scanner, Webcam, lecteur CD-ROM, ...

### b) Unités de sortie

Ces unités permettent à l'utilisateur de recevoir des résultats des calculs ou des manipulations de données effectués par l'ordinateur.

Exemples : Ecran, haut-parleurs, imprimante ...

### c) Unités de stockage (Mémoires auxiliaires)

Un ordinateur possède également un type de stockage qui conservent l'information de manière quasi permanente comme les disquettes, les disques durs, les clés USB,...

On distingue généralement :

Des supports magnétique : Disques durs, disquettes.

Des supports optiques : CD-ROM, DVD, Blue-Ray.

### d) Un système de bus

Le bus est le système de communication central d'un ordinateur il est composé essentiellement par des circuits intégrés, des connecteurs et des fils conducteurs. Pour simplifier les choses on peut le représenter comme un ensemble de conducteurs parallèles servant au transfert d'information entre les composants d'un ordinateur. Il peut être comparé à une autoroute reliant les différentes parties du système d'information qui se la partagent pour échanger des données. En général supervisé par le microprocesseur, un bus est souvent spécialisé dans le type d'information qu'il transfère. Ainsi, un groupe de circuits transporte les données, un autre transporte les adresses des données enregistrées dans la mémoire vive et un troisième véhicule les informations de contrôle qui veillent à ce que chaque partie de système utilise bien la partie d'autoroute qui lui est réservée.

Les bus se distinguent également par le nombre de bits qu'il transporte en parallèle. Un bus 32 bits véhicule 32 bits à la fois tandis qu'un bus 64 bits transporte 8 octets.

Il existe quatre types de bus :

#### **Bus de Données :**

Il transmet les données entre les différents composants de l'ordinateur.

#### **Bus d'adresses :**

Il sélectionne l'itinéraire qui doit être suivi par le bus de données pour transférer les données.

#### **Bus de commande :**

Elle décide que si les données doivent être écrites ou lues sur le bus de données.

#### **Bus d'extension :**

Il est utilisé pour connecter des périphériques de l'ordinateur tels que l'imprimante, modem et scanner avec le processeur.

## e) Ports de communication

Sur les micro-ordinateurs, tous les périphériques sont reliés à la carte mère par un connecteur que l'on insère :

- a) Soit dans un port directement soudé à la carte mère.

Exemple des principaux ports de connexion :

- Ports Usb, Série, Parallèle : qui permettent de connecter facilement des périphériques à l'ordinateur tels qu'une imprimante, une souris, un clavier, un disque dur externe, une webcam, un scanner, ...
- Les ports PCI, PCI Express : qui permettent de connecter facilement à la carte mère des cartes additionnelles telles que la carte son, la carte vidéo, la carte réseau.
- Les ports Séries ou Sérial-Ata : Pour connecter le(s) disque(s) dur(s) interne à l'unité centrale, le(s) lecteur(s) de disque(s) (Cd-Rom, Dvd-Rom, Blue Ray), ...

- b) Soit dans un port disponible sur une carte d'extension, elle-même enfichée (soudé ou insérer dans un port) sur la carte mère.

Exemple de principaux ports de connexion :

- Prise écouteurs, prise micro : Relié en générale à la carte son qui nous permettent de connecter un micro et des écouteurs (haut-parleur) à notre ordinateur.
- Port réseau (ou Lan) : Relié à la carte réseau et qui nous permet de nous brancher notre ordinateur au réseau internet (Intranet) ou externe (exemple : Internet).
- Les ports VGA ou DVI ou HDMI : relié à la carte graphique et qui nous permettent de relier notre ou nos écrans à l'ordinateur.

La carte d'extension étant amovible, il est facile de la remplacer en cas de panne ou d'évolution technologique (exemple : carte son, carte réseau, ...).

La plupart des périphériques sont amovibles, c'est-à-dire qu'ils peuvent être déconnectés de l'unité centrale sans empêcher celle-ci de fonctionner (Rmq : il faut parfois éteindre l'ordinateur avant de retirer le périphérique).

## ***2. Unité centrale de traitement CPU : Le processeur***

Le processeur est la pièce la plus importante dans un ordinateur. C'est la puce électronique qui représente le cerveau de votre machine.



Processeur AMD - Athlon



- Une puce est une petite boîte noire avec des pattes qu'on peut souder.

Il existe principalement (pour les particuliers) 2 marques de processeurs : Intel et AMD.

- La marque "Intel" avec les PENTIUM, CELERON, ATOM,...
- La marque "AMD" avec les K6 2, K6 3 et les Athlon.

Un microprocesseur comprend essentiellement :

- Une unité arithmétique et logique (UAL) qui effectue les opérations.
- Des registres qui permettent au microprocesseur de stocker temporairement des données.
- Une unité de contrôle qui commande l'ensemble du microprocesseur en fonction des instructions du programme.

Un microprocesseur se caractérise aujourd'hui par différentes fonctions :

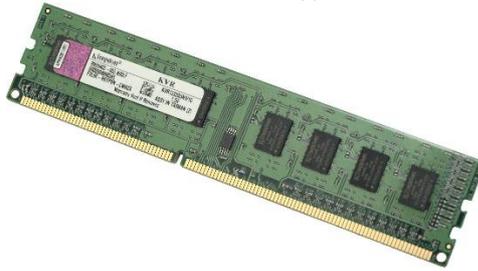
- Le jeu d'instructions qu'il est capable d'exécuter, pouvant aller de dizaines à des milliers d'instructions différentes.
- La vitesse de son horloge qui dicte le rythme de travail.
- La longueur de mot : 32bits, 64bits,....
- La mémoire cache : Une mémoire ultra rapide utilisée par le microprocesseur pour stocker des valeurs intermédiaires des calculs, et aussi pour enregistrer temporairement des copies de données provenant d'une source afin de diminuer le temps d'un nouvel accès.

### ***3. Mémoire centrale***

La mémoire centrale est un organe de l'ordinateur permettant d'enregistrer, de stocker et de restituer les informations. Elle se présente sous la forme de petites barrettes que l'on peut enficher dans des supports, appelés Slots.

La mémoire centrale d'un ordinateur est séparée en deux sections : la mémoire vive et la mémoire morte.

1. La mémoire vive (RAM Random Access Memory) :



Est une mémoire où on peut lire et écrire à volonté. Cette mémoire est dite « volatile » c'est-à-dire qu'elle perd son contenu dès qu'elle est hors tension.

La mémoire vive contient en plus des programmes servant à la gestion de l'ordinateur, le programme relatif à un traitement spécifique ainsi que les données qu'il requiert et les résultats qu'il génère.

2. La mémoire morte (ROM Read Only Memory) :



Est destinée uniquement à être lue ; en revanche, son contenu n'est pas altéré par une coupure de courant, c'est une mémoire « rémanente ».

La mémoire morte, programmée par le constructeur, est essentiellement utilisée pour contenir les logiciels de base servant au démarrage de l'ordinateur (Exemple BIOS).

Toutefois, il est possible d'écrire sur certains types de mémoire morte, désignés par les sigles PROM (Programmable Read Only Memory), EPROM (Erase Programmable Read Only Memory) et EEPROM (Electrically Erasable Programmable Read Only Memory) :

Une PROM est programmé une seule fois, et elle est dédiée à des fonctions spécifiques, comme par exemple des jeux.

Une EPROM est une EEPROM peuvent être effacées (respectivement par un rayonnement ultraviolet et par un courant électrique) puis reprogrammées pour autre usage.

#### 4. *Système binaire*

##### Base 10 et numération de position :

Dans le système décimal, nous utilisons 10 symboles : 0,1,2,3,4,5,6,7,8,9 et la position d'un symbole correspond au chiffre des unités, des dizaines, des centaines, etc.

$$1273 = 1 \times 10^3 + 2 \times 10^2 + 7 \times 10 + 3$$

### Codage binaire :

En informatique, la base la plus utilisée est la base 2 avec donc deux symboles, 0 et 1.

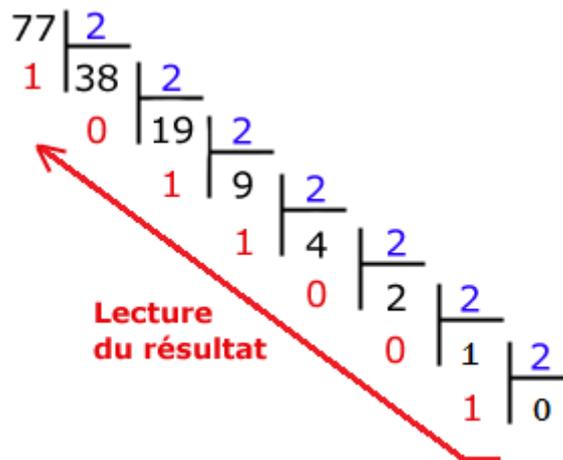
Le principe général de codage est alors le suivant : si  $(a_{n-1}, a_{n-2}, \dots, a_0)_2$  dénote un nombre en base 2, sa valeur décimale est :

$$a_{n-1} \times 2^{n-1} + a_{n-2} \times 2^{n-2} + \dots + a_0 \times 2^0$$

Remarque : le bit à la position 0, dit bit de poids faible, détermine à lui seul la parité de l'entier à représenter : s'il vaut 0, le nombre est pair, s'il vaut 1, le nombre est impair.

### Conversion d'un nombre décimal en binaire par divisions successives :

Pour convertir un nombre décimal en binaire on utilise la division successive :



$$77 = (1001101)_2$$

### Codage en binaire d'un nombre fractionnaire :

Méthode de calcul d'un nombre fractionnaire en binaire :

Exemple : 3,625

3 en binaire est 11

0,625 en binaire est de 0,101 calculé de la façon suivante :

$$0,625 * 2 = \boxed{1},25$$

$$0,25 * 2 = \boxed{0},5$$

$$0,5 * 2 = \boxed{1},0$$

$$\text{Donc : } (3,625)_{10} = (11,101)_2$$

## Opérations arithmétiques en binaire :

Nous pouvons définir les opérations élémentaires par analogie avec le système décimal. L'addition en binaire de deux nombres consiste à effectuer l'addition binaire sur les bits de même poids de chaque nombre reportant de droite à gauche les retenues successives.

Exemples :

$$\begin{array}{r} 1011 \\ + 0100 \\ \hline 1111 \end{array}$$

$$\begin{array}{r} 11 \\ 0010 \\ + 0111 \\ \hline 1001 \end{array}$$

$$\begin{array}{r} 111 \\ 0011 \\ + 0111 \\ \hline 1010 \end{array}$$

### 5. Représentation des caractères dans la mémoire

Le texte est constitué de caractères (lettre, chiffre, signe de ponctuation). Chaque caractère est représenté par un entier. Il existe de nombreux **codages des caractères** ; les principaux codages pour les occidentaux sont :

- 1) Le code ASCII (ISO 646)
- 2) Les codes ISO 8859-1 / ISO 8859-15
- 3) Le code **Unicode**

#### a) Le code ASCII (7 bits)

Code	Char	Code	Char	Code	Char	Code	Char	Code	Char	Code	Char
32	[space]	48	0	64	@	80	P	96	`	112	p
33	!	49	1	65	A	81	Q	97	a	113	q
34	"	50	2	66	B	82	R	98	b	114	r
35	#	51	3	67	C	83	S	99	c	115	s
36	\$	52	4	68	D	84	T	100	d	116	t
37	%	53	5	69	E	85	U	101	e	117	u
38	&	54	6	70	F	86	V	102	f	118	v
39	'	55	7	71	G	87	W	103	g	119	w
40	(	56	8	72	H	88	X	104	h	120	x
41	)	57	9	73	I	89	Y	105	i	121	y
42	*	58	:	74	J	90	Z	106	j	122	z
43	+	59	;	75	K	91	[	107	k	123	{
44	,	60	<	76	L	92	\	108	l	124	
45	-	61	=	77	M	93	]	109	m	125	}
46	.	62	>	78	N	94	^	110	n	126	~
47	/	63	?	79	O	95	_	111	o	127	[backspace]

L'ASCII (American standard code for information interchange) a été créé au début des années 60. Son principe consiste à associer à chaque lettre, chiffre ou caractère d'un clavier d'ordinateur un entier compris entre 0 et 127, donc représentable sur 7 bits. Avec ce code on peut représenter les chiffres, les lettres latines, et les principaux symboles de ponctuation. Exemple : lorsqu'on tape dans un texte sur la barre d'espace (space), l'ordinateur enregistre

dans sa mémoire vive (RAM) le code hexadécimal  $(20)_{\text{hex}}$ . Le codage binaire sur 7 bits correspondant est :  $(010\ 0000)_2 = (32)_{10}$

	<p><b>Remarque: comment insérer un caractère ASCII dans un document Word ?</b></p> <p>Outre la saisie d'un caractère au clavier, vous pouvez également utiliser la code de caractère du symbole comme raccourci clavier ou lorsqu'un symbole n'est pas disponible sur le clavier que vous utilisez.</p> <p>Pour insérer un caractère ASCII des tables ci-dessous, appuyez sur la touche Alt et maintenez-la enfoncée tout en tapant l'équivalent numérique décimal.</p> <p>Par exemple, pour insérer le symbole du point d'exclamation ! , appuyez sur la touche Alt et maintenez-la enfoncée tout en tapant 33 sur le pavé numérique.</p>
---	--

### b) Le codage ISO 8859 (8 bits)

Le codage ASCII suffit pour coder un texte anglais mais ne suffit pas pour les autres langues. Par exemple, les lettres accentuées ne figurent pas dans le code ASCII. Les code **ISO 8859** permettent de représenter les autres langues et consiste en une extension du code ASCII. Pour ce code à chaque caractère alphanumérique (lettre, chiffre, ponctuation etc..) est associé un nombre compris entre 0 et 255. On a donc besoin de 8 bits pour représenter l'ensemble des caractères.

Exemples des code ISO 8859 :

ISO 8859-1 : latin 1 : ou européen occidental

ISO 8859-2 : latin 2 : ou européen central

ISO 8859-6 : arabe

### c) L'Unicode (16 bits ou 32 bits)

D'autres codages ont été définis pour les autres langues : le chinois, l'arabe etc.. Ces codes sont compatibles avec l'ASCII mais ne le sont pas entre eux. Comment écrire un texte multilingue ? On a créé un **unique codage universel l'Unicode**. Les valeurs entières associées étaient initialement comprises entre 0 et 65235 donc codées sur 16 bits. Cependant ce codage souffre de nombreux défauts (car mis en place par l'homme). Maintenant il est codé sur 32 bits. Toutes les langues connues (sur Terre) sont représentées dans Unicode.

## 6. Représentation des nombres dans la mémoire

On appelle représentation (ou codification) d'un nombre la façon selon laquelle il est décrit sous forme binaire. La représentation des nombres sur un ordinateur est indispensable pour que celui-ci puisse les stocker, les manipuler. Toutefois le problème est qu'un nombre mathématique peut être infini (aussi grand que l'on veut), mais la représentation d'un nombre dans un ordinateur doit être faite sur un **nombre de bits prédéfini**.

Dans ce qui suit on va voir la méthode de représentation en binaire des trois types de base des nombres :

- Nombres entiers naturels (Appelé aussi nombres non signés)
- Nombres entiers relatifs (Appelé aussi nombre signés)
- Nombres réels

### a) Représentation d'un entier naturel

Un entier naturel est un entier positif ou nul. Le choix à faire (c'est-à-dire le nombre de bits à utiliser) dépend de la fourchette des nombres que l'on désire utiliser. Pour un codage sur 8 bits on peut coder les valeurs comprises entre 0 et 255. D'une manière générale un codage sur n bits pourra permettre de représenter des nombres entiers naturels compris entre 0 et  $2^n-1$ .

Pour représenter un nombre entier naturel après avoir défini le nombre de bits sur lequel on le code, il suffit de ranger chaque bit dans la cellule binaire correspondant à son poids binaire de la droite vers la gauche, puis on « remplit » les bits non utilisés par des zéros.

#### Exemples :

Codage de la valeur 5 sur 8 bits :

0	0	0	0	0	1	0	1
---	---	---	---	---	---	---	---

Codage de la valeur 255 sur 8 bits :

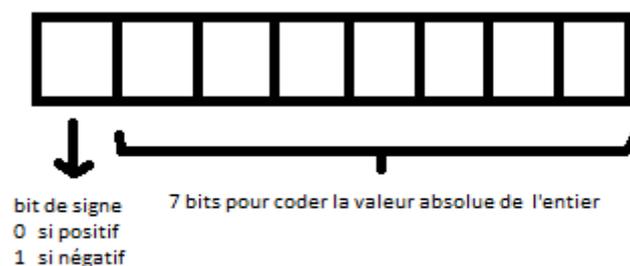
1	1	1	1	1	1	1	1
---	---	---	---	---	---	---	---

### b) Représentation d'un entier relatif

Un entier relatif est un entier pouvant être négatif. Il faut donc coder le nombre de telle façon que l'on puisse savoir s'il s'agit d'un nombre positif ou d'un nombre négatif, et il faut de plus que les règles d'addition soient conservées. L'astuce consiste à utiliser un codage que l'on appelle complément à deux.

- Codage d'un entier relatif positif :

Un entier relatif positif ou nul sera représenté en binaire (base 2) comme un entier naturel, à la seule différence que le bit de poids fort (le bit situé à l'extrême gauche) représente le signe. Il faut donc s'assurer pour un entier positif ou nul qu'il est à zéro (0 correspond à un signe positif, 1 à un signe négatif).



#### Codage d'un entier relatif sur 8 bits

Ainsi si on code un entier naturel sur 8 bits, le nombre le plus grand sera 01111111 (c'est-à-dire 127 en base décimale).

D'une manière générale le plus grand entier relatif positif codé sur n bits sera  $2^{n-1}-1$ .

- Codage d'un entier relatif négatif :

Pour pouvoir représenter les nombres négatifs, on change de convention de codage : un nombre négatif est le complément à deux de sa valeur positive.

Principe du complément à 2 :

1. On représente la valeur absolue du nombre négatif par la méthode vu en dessus
2. On complémente chaque bit (on inverse, c'est-à-dire que l'on remplace les zéro par 1 et vice-versa)
3. On ajoute 1

### Exemples :

On désire coder la valeur -5 sur 8 bits. Il suffit :

- d'écrire 5 en binaire : 0000101
- de complémenter à 1 : 11111010
- d'ajouter 1 : 11111011
- la représentation binaire de -5 sur 8 bits est 11111011

### Remarques:

Le bit de poids fort est 1, on a donc bien un nombre négatif.

Si on ajoute 5 et -5 (0000101 et 11111011) on obtient 0 (avec une retenue de 1...).

$$\begin{array}{r}
 5 \\
 + \\
 -5 \\
 \hline
 0
 \end{array}
 \qquad
 \begin{array}{r}
 0000101 \\
 + \\
 11111011 \\
 \hline
 10000000
 \end{array}$$

**Le 1 est en dépassement de capacité puisque le codage est sur 8 bits donc le résultat est bien : 0000000**

Sur 32 bits, on peut coder des nombres entiers signés allant de  $-2^{31}$  à  $2^{31} - 1$ .

Propriétés du complément à 2 :

$$\text{Comp2}(N) + N = 0$$

$$\text{Comp2}(\text{Comp2}(N)) = N$$

### c) Représentation d'un nombre réel

A la fin des années 70 chaque ordinateur avait sa propre représentation interne pour les nombres à virgule flottante. Or, l'arithmétique à virgule flottante possède certaines subtilités que le constructeur moyen ne maîtrisait pas forcément et certaines machines effectuaient certaines opérations de manière incorrecte. Pour remédier à cette situation, l'IEEE proposa un standard non seulement pour permettre les échanges de données en virgule flottante entre ordinateurs, mais aussi pour fournir un modèle rodé aux constructeurs, dont le fonctionnement était correct et maîtrisé. De nos jours, pratiquement tous les constructeurs ont des processeurs ou des coprocesseurs dédiés qui effectuent des calculs en virgule flottante et qui emploient la représentation au standard IEEE 754.

La norme IEEE 754 définit la façon de coder un nombre réel. Cette norme propose des formes pour coder les nombres à virgule flottant :

- Simple précision à 32 bits
- Double précision à 64 bits

La méthode consiste à réécrire le nombre sous la forme normalisée suivante :

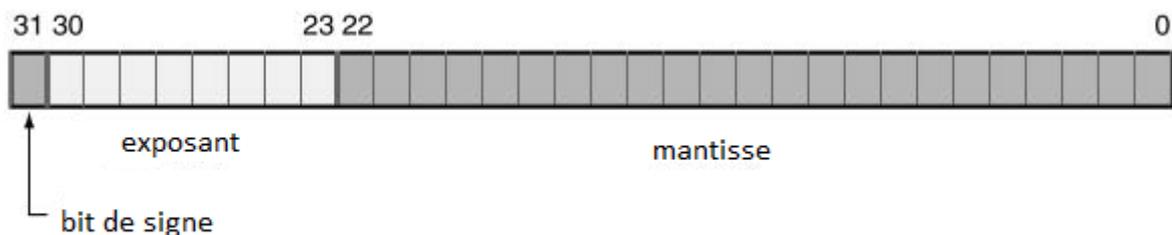
$$(+/-) 1.m 2^{(e-127)}$$

m : la mantisse du nombre

e : l'exposant du nombre

Sous sur 32 bits et définit trois composantes :

- Le signe est représenté par un seul bit, le bit de poids fort (celui le plus à gauche)
- L'exposant est codé sur les 8 bits consécutifs au signe
- La mantisse (les bits situés après la virgule) sur les 23 bits restants



#### IEEE 754 Simple précision

Certaines conditions sont toutefois à respecter pour les exposants :

- L'exposant 00000000 est interdit
- L'exposant 11111111 est interdit. On s'en sert toutefois pour signaler des erreurs, on appelle alors cette configuration du nombre NaN, ce qui signifie Not a number
- Il faut rajouter 127 (01111111) à l'exposant pour une conversion de décimal vers un nombre réel binaire. Les exposants peuvent ainsi aller de -127 à 128

### Exemples :

Soit à coder la valeur 525,5.

- 525,5 est positif donc le 1er bit sera 0.
- Sa représentation en base 2 est la suivante : 1000001101,1
- En normalisant, on trouve :  $1,0000011011 * 2^9$
- On ajoute 127 à l'exposant qui vaut 9 ce qui donne 136, soit en base 2 : 10001000
- La mantisse est composée de la partie décimale de 525,5 en base 2 normalisée, c'est-à-dire 0000011011.
- Comme la mantisse doit occuper 23 bits, il est nécessaire d'ajouter des zéros pour la compléter :

000001101100000000000000

- La représentation du nombre 525,5 en binaire avec la norme IEEE est donc :

0 1000 1000 000001101100000000000000

Voici un autre exemple avec un réel négatif :

Soit à coder la valeur -0,625.

- 1er bit vaut 1 car -0,625 est négatif
- 0,625 s'écrit en base 2 de la façon suivante : 0,101
- On souhaite l'écrire sous la forme  $1.01 * 2^{-1}$
- Par conséquent l'exposant vaut 11111110 car  $127 - 1 = 126$  (soit 11111110 en binaire)
- la mantisse est 010000000000000000000000 (seuls les chiffres après la virgule sont représentés, le nombre entier étant toujours égal à 1)
- La représentation du nombre 0,625 en binaire avec la norme IEEE est :  
1 1111 1110 010000000000000000000000

## 7. Codage des images

En informatique une couleur est représentés par trois valeurs des intensités des trois couleurs de base RGB (Red Green Blue). Chaque couleur est codée sur 8 bits donc l'intensité de chaque couleur varie de 0 à 255.

### Exemples :

Le Noir est codé par : (R=0, V=0, B=0)

Le Blanc est codé par : (R=255, V=255, B=255)

Le Jaune est codé par : (R=255, V=255, B=0)

Le Rouge est codé par : (R=255, V=0, B=0)

Un autre Rouge codé par : (R=250, V=0, B=0)

Avec ce codage on peut avoir 16 581 375 couleurs.

Coder une image en couleur consiste à coder chacun de ses pixels, pour chaque pixel, on code en binaire chacune des 3 valeurs des composantes R, V, B de la couleur du pixel. Le code binaire de l'image est obtenu en indiquant successivement pour chaque pixel le code binaire des 3 composantes. Si on code chaque composante sur 8 bits, chaque pixel sera donc représenté par 24 bits.

## II. Système d'exploitation d'un ordinateur

### 1. Définition et exemples de systèmes d'exploitation

Un système d'exploitation, ou *Operating System* (OS), est un logiciel qui, dans un appareil électronique, pilote les dispositifs matériels et reçoit des instructions de l'utilisateur ou d'autres logiciels (ou applications).

Pour démarrer un ordinateur un système d'exploitation est indispensable. On peut installer un ordinateur plusieurs systèmes d'exploitation mais on ne peut démarrer qu'avec un seul à la fois, un écran au démarrage nous invite à choisir le système avec lequel l'ordinateur démarre.

Dans le secteur informatique, les systèmes d'exploitation les plus répandus sont :

- Microsoft Windows (pour les PC) : Windows XP, Vista, 7, 8, 10
- Apple Mac OS (pour les ordinateurs d'Apple) : Tiger, MacOS X
- Unix (pour les serveurs) : Ubuntu, Mandriva, RedHat, ...
- Les systèmes embarqués (pour les téléphones, tv, voitures,...) : Android de Google, iOS de Apple, Windows Phone de Microsoft,...

### 2. Principales fonctionnalités

#### a) Gestion du dialogue entre la machine et l'utilisateur

La communication entre l'utilisateur et le S.E s'effectue par :

- Un langage de commande :

Dans ce cas l'utilisateur doit connaître le nom de la commande pour effectuer une tâche.

- Une interface graphique :

Utilisation d'environnement graphique, dans ce cas il suffit de choisir et cliquer sur des objets sous forme d'icône, ouvrir des fenêtres...

## b) Gestion des fichiers et répertoires

Pour être traité par ordinateur l'information est stockée dans des entités logiques appelées fichier.

Un fichier : Est un ensemble des données de mêmes types qui sont présentes de façon permanente sur une mémoire de masse.

Un fichier est caractérisé par :

- Un nom : formé d'une suite finie des caractères ;
- Une extension : (ou suffixe) sert à identifier le type du fichier.
- Date : la date de création ou de modification
- Taille : exprimé en octet.



Exemple d'extensions: TXT,Doc,xls,bmp,jpg,gif,mid,wav,mp3,avi,htm,exe,pas....

Un répertoire (ou un dossier) : Est un objet informatique pouvant contenir des fichiers et/ou des dossiers. Les répertoires peuvent être organisés sous forme d'arborescence.

Dossiers	Nom	Taille	Type	Date de modification
Bureau	Temp		Dossier de fichiers	07/08/2008 20:26
Mes documents	twain_32		Dossier de fichiers	27/07/2008 16:38
Poste de travail	Web		Dossier de fichiers	27/07/2008 14:50
Disquette 3 1/2 (A:)	WinSxS		Dossier de fichiers	27/07/2008 15:16
Disque local (C:)	_default	1 Ko	Raccourci pour le progra...	07/09/2002 01:00
Documents and Settir	bootstat.dat	2 Ko	Fichier DAT	08/08/2008 20:54
downloads	Bulles de savon.bmp	65 Ko	Image bitmap	07/09/2002 01:00
Program Files	clock.avi	81 Ko	Clip vidéo	07/09/2002 01:00
WINDOWS	control.ini	0 Ko	Paramètres de configuration	27/07/2008 14:52
addins	desktop.ini	1 Ko	Paramètres de configuration	07/09/2002 01:00
AppPatch	DtcInstall.log	1 Ko	Document texte	27/07/2008 15:17
Config	explorer	1 Ko	Windows Explorer Command	07/09/2002 01:00
Connection Wizar	explorer.exe	1 012 Ko	Application	04/08/2004 05:54
Cursors	FaxSetup.log	12 Ko	Document texte	27/07/2008 14:48

## c) Gestion des périphériques

Le S.E permet de communiquer avec les périphériques d'entrée/sortie et de définir le mode d'emploi et la configuration matérielle.

Gestion des applications :

Consiste à exécuter plusieurs programmes simultanément, le S.E contrôle le programme dès son chargement en mémoire centrale jusqu'à sa terminaison.

## Gestion de la mémoire centrale

Généralement, pour être exécuté un programme doit être chargé dans la RAM. Vu sa taille limitée le SE doit optimiser (bien organiser) son utilisation

La gestion de la mémoire consiste à remplir les fonctions suivantes:

- o Permettre le partage de la mémoire entre les processus
- o Protéger les zones mémoires utilisées
- o Récupérer les zones mémoires lorsque les processus terminent leur travail