Chapitre III : Architecture des ordinateurs

Objectifs du chapitre

- Identifier les différents composants matériels et logiciels de l'ordinateur
- Comprendre le fonctionnement de l'ordinateur dans ses mécanismes élémentaires
- Comprendre les principes de fonctionnement d'un ordinateur et son système d'exploitation

Dans ce chapitre on s'intéresse à l'architecture matérielle et logicielle de l'ordinateur. Les caractéristiques essentiels d'un ordinateur et ses composants sont définis. On va essayer aussi de comprendre le processus de traitement de l'information. Enfin, les différents logiciels et systèmes d'applications sont expliqués.

I.1 Composantes de base d'un ordinateur

L'ordinateur est une compilation de des éléments matériels et logiciels. Dans la suite de ce chapitre on va citer les composantes les plus importantes de l'ordinateur.

I.1.1 Partie matérielle :

Unité centrale

Cet élément comprend plusieurs composantes. Parmi ces composantes on peut souligner :

- **Carte mère** : elle est considérée comme la pièce principale d'un PC. C'est un circuit imprimé qui supporte la plupart des composants et des connecteurs nécessaires au fonctionnement d'un compatible PC. Elle permet aussi de connecter d'autres composants pour assurer le bon fonctionnement d'un ordinateur.

- **Processeur** : C'est le cerveau de l'ordinateur, c'est lui qui organise les échanges de données entre les différents composants (disque dur, mémoire RAM, carte graphique) et qui fait les calculs qui font que l'ordinateur interagit avec vous et affiche votre système à l'écran. Sa puissance est exprimée en Hz. Aujourd'hui, un processeur atteint les 3Ghz (Giga, milliards) et certains ordinateurs sont équipés de plusieurs processeurs.

  Le processeur est constitué d'un ensemble de circuits électroniques capables de lire et d'écrire des informations, et d'effectuer des opérations arithmétiques. Il possède une mémoire pour stocker des programmes, des données et d'autres informations. Dans le passé, la plupart des processeurs étaient construits autour d'une unité centrale de traitement (UC) constituée de nombreux transistors. Ces transistors peuvent être regroupés en portes logiques qui effectuent des opérations spécifiques telles que AND, OR, NAND, NOR, XOR, etc. Cependant, au cours des dix ou vingt dernières années, la plupart des processeurs ont été fabriqués à partir de circuits intégrés (ou puces) qui contiennent un grand nombre de portes logiques sur une seule puce.

  Dans la technologie PC, les processeurs sont en majeure partie fabriqués par Intel et AMD. Les dernières générations de processeurs s'appellent Pentium, Celeron, Duron ou Athlon. Le volume d'information qu'un processeur peut recevoir ou émettre (traitement externe) et celui qu'il peut traiter en un cycle de traitement (traitement interne) s'exprime en multiples de 8 bits.

- **Mémoire** : c'est l'unité de stockage, elle sauvegarde les informations. On distingue la mémoire principale (Secondary Storage Memory) et la mémoire secondaire (Primary Storage Memory).
  - ***La mémoire principale (ou Vive) RAM*** *(Random Access Memory)* **:** est une mémoire qui sert à stocker les données et les programmes qui sont en exécution ou en cours d'accès pendant l'utilisation. Elle est composée de circuits électroniques extrêmement

rapide et coûteuse. Les programmes et les données peuvent être stockés dans la RAM pendant l'utilisation de l'ordinateur. La RAM est une mémoire volatile (Toutes les informations seront perdues dès que l'ordinateur s'arrête. Exemples (SDRAM (Synchronous Dynamics RAM: 2 Encoches); DDRAM (Double Data Rate SDRAM: 1Encoche); DDR2 SDRAM (Double Data Rate two SDRAM); DDR3 SDRAM (Double Data Rate three SDRAM); XDR DRAM (XDimm Rambus RAM) ….).

La mémoire vive étant limitée en taille, il peut arriver que des programmes essaient de stocker plus de données que ne le permet la mémoire vive. Lorsque cela arrive, on parle de dépassement de pile (ou stack overflow).

La mémoire vive est dédiée à des données peu nombreuses qui nécessitent un accès rapide.

- ***La mémoire secondaire* :** C'est la mémoire de stockage de l'ordinateur. Elle conserve les données même si l'ordinateur est éteint. C'est par exemple le cas d'un disque dur, les disquettes, le CD ROM, le DVD ROM, la clé USB. Il existe plusieurs types de mémoire secondaire classées selon la possibilité de les modifier, comme par exemple :

- *ROM* (pour Read-Only Memory) : ce type de mémoire n'est pas effaçable et est utilisé, par exemple, pour stocker les instructions de démarrage d'un ordinateur. Les données stockées sont enregistrées par le constructeur et ne pourront pas être altérées.

- *EEPROM* (pour Electrically Erasable Programmable Read Only Memory) : ce type est le plus répandu - c'est celui des clés USB, des cartes SD, etc. La suppression et la modification des données est possible.

En général, la mémoire secondaire est dédiée à des données potentiellement volumineuses qui ne nécessitent pas un accès immédiat.

- **Carte graphique**

Une carte graphique ou carte vidéo (anciennement par abus de langage une carte VGA), ou encore un adaptateur graphique, est une carte d'extension d'ordinateur dont le rôle est de produire une image affichable sur un écran. La carte graphique envoie les images qu'elle possède dans sa mémoire à l'écran à une fréquence et dans un format qui dépendent d'une part de l'écran branché et du port sur lequel il est branché (grâce au Plug and Play) et de sa configuration interne d'autre part.

Périphériques

Un périphérique informatique est un dispositif connecté à un système de traitement de l'information central (ordinateur, console de jeu, etc.1) et qui ajoute à ce dernier des fonctionnalités.

- ***Périphériques d'entrée/sortie***

  Parmi les composants d'un ordinateur, on distingue **les périphériques de sortie** (moniteur, imprimante, haut-parleurs, graveur de CD), des **périphériques d'entrée** (clavier, souris, lecteur de CD/DVDRom, scanner, micro, webcam, manette de jeu, appareil photo et caméscope numériques). Il existe un cas particulier : le modem qui est un périphérique de sortie si l'on envoie des informations vers Internet et un périphérique d'entrée si l'on reçoit des informations d'Internet.

  Chaque composant joue un rôle bien précis : le microprocesseur reçoit et transmet des informations ; le disque dur stocke les informations ; les périphériques d'entrée analysent les informations fournies et les transmettent au microprocesseur ; les périphériques de sortie (imprimante, haut-parleurs, graveur) traduisent les informations reçues par les périphériques d'entrée, puis transmises au microprocesseur.

- ***Périphériques de stockage***

  Permettent de sauvegarder les informations. Lecteur de disquettes, disque dur, lecteur de cartouches de sauvegarde, lecteur de disques amovibles (Zip ou autre), lecteur de cédéroms ou DVD gravés.

- **Onduleur**

  C'est un dispositif de protection de l'ordinateur. Il régularise le courant électrique qui traverse l'ordinateur et possède les réserves d'énergie après coupure électrique.

- **Modem**

  Pour connecter l'ordinateur à Internet, vous avez besoin d'un modem. Un modem est un périphérique qui envoie et reçoit des données informatiques via une ligne téléphonique ou un câble à haut débit. Les modems sont parfois intégrés à l'unité système, mais les modems à haut débit sont généralement des composants séparés.

I.1.2 Partie logicielle

On peut décomposer la partie logicielle d'un ordinateur en quartes éléments : le BIOS, le système d'exploitation, les drivers et les logiciels.

- **BIOS** (Basic Input Output System)

Le BIOS est le premier programme chargé en mémoire dès que vous allumez votre ordinateur. Il assure plusieurs fonctions :

○ Il initialise tous les composants de la carte mère et de certains périphériques ; ○ Il identifie tous les périphériques internes et externes qui lui sont connectés ; ○ Si cela n'a pas déjà été fait il initialise l'ordre de priorité des périphériques d'entrée ; ○ Il démarre le système d'exploitation dans l'ordre croissant des périphériques disponibles.

Le BIOS s'exécute au démarrage de l'ordinateur. Il déclare les disques, configure les composants et recherche l'unité de booting, c'est-à-dire la partie où rechercher un système d'exploitation et finalement lance ce dernier. Dans la plupart des cas, l'OS (Operating System) se trouve dans le master boot record du disque dur et se charge en RAM.

- **Système d'exploitation**

C'est l'interface qui permet de faire le lien entre l'utilisateur, les programmes et les composants de l'ordinateur. Quand vous allumez votre ordinateur, c'est grâce à cela que vous voyez des fenêtres, que vous pouvez gérer vos fichiers, installer des programmes ou des périphériques externes.

Le système d'exploitation est la passerelle entre l'utilisateur, les ressources et les applications. Lorsqu'un programme est lancé, il ne communique pas directement avec un périphérique. Les instructions passent par le système d'exploitation, qui se charge de les transmettre au périphérique.

A l'aide d'un algorithme, le système d'exploitation gère l'allocation du processeur, le cerveau de la machine. Il intervient aussi dans la gestion de la mémoire vive, qui est limitée. Lorsqu'un manque d'espace libre est constaté, il crée des espaces libres appelés mémoire virtuelle.

Les 3 principaux systèmes d'exploitation sont : Windows, OS X et Linux. Ces systèmes d'exploitation ont évolué avec le temps, il en existe donc plusieurs versions.

- ***Windows*** a été créé par Microsoft, il est actuellement le plus répandu des 3. La version vendue actuellement est Windows 10 mais vous risquez de rencontrer d'anciennes versions : Windows 8, Windows 7, Windows Vista ou encore Windows XP. Ce système est vendu sur différentes marques d'ordinateurs (Acer,

Asus, Dell, HP, Sony, Toshiba...) ○ ***macOS*** (anciennement OS X) a été développé par la société Apple. Ce système d'exploitation n'est présent que sur les ordinateurs de la marque Apple (Macintosh). Ceux-ci sont facilement reconnaissables grâce au logo représentant une pomme. La version actuellement vendue est macOS Sierra, la version macOS

High Sierra devrait sortir à l'automne 2017.

- Enfin, ***Linux*** est le moins connu des 3 systèmes d'exploitation. Il est rarement installé par défaut sur un ordinateur. Gratuit et libre, il est surtout utilisé par ceux qui ont de bonnes connaissances en informatique. Une distribution Linux® est un système d'exploitation prêt à installer, conçu à partir d'un noyau Linux, et qui prend en charge des programmes et des bibliothèques. Chaque *version* d'un fournisseur ou d'une communauté est appelée *distribution*. Étant donné que le système d'exploitation Linux est Open Source et distribué sous la licence publique générale GNU, n'importe qui peut exécuter, étudier, modifier et redistribuer le code source, ou même vendre des copies du code modifié. Il diffère en cela des systèmes d'exploitation traditionnels, tels que Microsoft Windows, Unix et MacOS, qui sont propriétaires et bien plus difficiles à modifier.

• **Les logiciels**

(Voir chapitre I)

• **Les drivers**

Un driver, ou pilote en français, désigne un programme informatique (logiciel) particulier. Son rôle est de permettre la bonne liaison entre votre ordinateur et un périphérique (imprimante, webcam, scanner, etc.).

Certains périphériques ont en effet besoin d'être reconnus par un ordinateur pour pouvoir fonctionner correctement. Dans le cas d'une imprimante par exemple, le driver est le programme informatique qui apporte à l'ordinateur les informations indispensables pour pouvoir utiliser l'imprimante.

I.1.3 Démarrage de l'ordinateur

Lorsqu'on met un ordinateur sous tension, il « démarre ». Techniquement parlant, une toute première couche logicielle va être mise en œuvre. Cette couche, toujours la même a été inscrite directement sur le matériel : c'est le BIOS (*Basic Input/Output System*).
Ce premier système réalise un certain nombre de vérifications et de tests (présence des divers périphériques, volume de la mémoire...). Puis il passe la main au système d'exploitation luimême. Techniquement il lance donc le premier élément d'une chaîne de programmes constituant ce qu'on nomme globalement le système d'exploitation. Ce premier étage de logiciel est la première interface homme-machine. C'est le système d'exploitation qui donne à l'ordinateur son visage humain, qui permet le dialogue entre la machine et l'utilisateur. Selon le degré de convivialité du système. Le cas échéant, une couche logicielle supplémentaire peut être appelée avant les programmes d'application), l'interface graphique. Ce fut le cas lorsque Windows n'était pas encore un système à part entière, mais une simple interface graphique se greffant sur MS-DOS (*Microsoft Disk Operating System*). L'utilisateur a alors la possibilité de démarrer des logiciels d'applications

I.2 Codage de l'information

Les systèmes informatiques ne peuvent fonctionner que selon une logique à deux états telle que, de façon schématique, le courant passe ou ne passe pas. Les informations sont donc transformées en impulsions électriques. Ces impulsions sont le seul langage qui puisse être compris par l'ordinateur. Ces deux états logiques sont conventionnellement notés 1 ou 0, et déterminent une logique dite binaire. Toute information à traiter devra être représentée sous forme binaire.

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

L'octet (en anglais byte ou B majuscule dans les notations) est une unité d'information composée de 8 bits. Il permet par exemple de stocker un caractère comme une lettre ou un chiffre. Une unité d'information composée de 16 bits est généralement appelée mot (en anglais Word). Une

unité d'information de 32 bits de longueur est appelée mot double (en anglais double Word, dword). Beaucoup d'informaticiens ont appris que 1 kilooctet valait 1024 octets, mais en décembre 1998, l'organisme international IEC a statue sur la question1. Voici les unités standardisées :

• Un kilooctet (ko) = 10^3 octets

• Un megaoctet (Mo) = 10^6 octets

• Un gigaoctet (Go) = 10^9 octets

• Un teraoctet (To) = 10^{12} octets

• Un petaoctet (Po) = 10^{15} octets

• Un exaoctet (Eo) = 10^{18} octets

• Un zettaoctet (Zo) = 10^{21} octets

• Un yottaoctet (Yo) = 10^{24} octets

Ordres de grandeur

Un fichier texte → 50 Ko

Une image pour le web → 30 Ko

Une musique (mp3) → 4 Mo

Une photo → 6 Mo

Un film → 700 Mo a 2 Go

Une disquette → 1.4 Mo

Un CD → 700 Mo

Un DVD → 4.7 Go

Un Blu-ray → 25 Go

Une clé USB → 8 Go a 2 To

Un disque dur → 250 Go à 4 To

Il existe de nombreuses possibilités de codage de l'information, binaire, hexadécimal, ASCII, etc.

I.2.1 Codage des entiers

Pour représenter des entiers, on utilise aujourd'hui des systèmes de numération dits pondérés ou positionnelle.

La définition d'un système de numération pondéré repose sur trois notions :

- La « base » du système : c'est un nombre entier, noté B.

- Les « chiffres » ou « digits » du système : ce sont des caractères, tous différents, représentant chacun un élément de la base. Il y en a donc B au total, notés 0, 1, 2, 3, 4, ... Pour écrire un nombre, on associe plusieurs chiffres dans un ordre déterminé, par exemple : N =1354 ou N =4153, ainsi de suite.
- Le « poids » de chaque chiffre selon son rang (sa position dans l'écriture). Compté de la droite vers la gauche, ce poids vaut B_0 (c'est à dire 1) pour le premier chiffre, B_1 (c'est à dire B) pour le second chiffre, B_2 pour le troisième chiffre, etc. ...

On notera un nombre sous la forme : (ijkl)B où i,j,k et l sont des chiffres, et B la base du système.

Par exemple, Dans le système décimal, la base B est 10. Il y a 10 chiffres notés : 0, 1, 2, 3,4, 5, 6, 7, 8, 9.

Les nombres 3997 et 195 exprimés en décimal signifient :

$3997 = 3x10^3 + 9x10^2 + 9x10^1 + 7x10^0$

$195 = 1x10^2 + 9x10^1 + 5x10^0$

En base 10, on ne note pas $(3997)_{10}$ ou $(195)_{10}$, mais simplement 3997 ou 195.

Système décimal

Nous utilisons le système décimal (base 10) dans nos activités quotidiennes. Ce système est base sur dix symboles, de 0 à 9, avec une unité supérieure (dizaine, centaine, etc.) à chaque fois que dix unités sont comptabilisées. C'est un système positionnel, c'est-à-dire que l'endroit où se trouve le symbole définit sa valeur. Ainsi, le 2 de 523 n'a pas la même valeur que le 2 de 132. En fait, 523 est ≪ l'abréviation ≫ de $5 \cdot 10^2 + 2 \cdot 10^1 + 3 \cdot 10^0$. On peut selon ce principe imaginer une infinité de systèmes numériques fondes sur des bases différentes.

Système binaire

En informatique, outre la base 10, on utilise très fréquemment le système binaire (base 2) puisque l'algèbre booléenne est à la base de l'électronique numérique. Deux symboles suffisent : 0 et 1.

Dans ce système, la base B est 2. Il y a 2 chiffres notés : 0 et 1. Les nombres $(1101)_2$ et $(101,01)_2$ exprimés en binaire signifient :

$(1101)_2 = 1x2^0 + 0x2^1 + 1x2^2 + 1x2^3 = 1x1 + 0x2 + 1x4 + 1x8 = 13$ (en base 10).

Système octal

Le système de numération **octal** est le système de numération de base 8, et utilise les chiffres de 0 à 7. La numération octale peut être construite à partir de la numération binaire en groupant les chiffres consécutifs en triplets (à partir de la droite). Par exemple, la représentation binaire du nombre décimal 74 est 1001010, que l'on groupe en 1 001 010 ; ainsi, la représentation octale est 1 pour 1, 1 pour le groupe 001, et 2 pour le groupe 010, ce qui nous donne 112.

Système Hexadécimal

On utilise aussi très souvent le système hexadécimal (base 16) du fait de sa simplicité d'utilisation et de représentation pour les mots machines (il est bien plus simple d'utilisation que le binaire). Il faut alors six symboles supplémentaires : A (qui représente le 10), B (11), C (12), D (13), E (14) et F (15).

Le tableau ci-dessous montre la représentation des nombres de 0 à 15 dans les bases 10, 2 et 16.

Décimal	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Binaire	0	1	10	11	100	101	110	111	1000	1001	1010	1011	1100	1101	1110	1111
Hexadécimal	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F

I.2.2 Conversion entre systèmes

Le transcodage (ou conversion de base) est l'opération qui permet de passer de la représentation d'un nombre exprime dans une base a la représentation du même nombre mais exprime dans une autre base.

Binaire → Décimal

Pour un entier naturel N, codé en système binaire, $N_2 = (a_n, a_{n-1}, ..., a_0)_2$

L'expression générale en base 10 est N_{10} et s'écrit $N_{10} = \sum_{i=0}^{n} a_i \cdot 2_i$

(N_2 et N_{10} sont des expressions différentes de la même quantité N)

Exemple : Considérons l'entier naturel $N_2 = (1011)_2$ en système binaire.

La conversion décimale N_{10} pour cet entier est :

$N_{10} = 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0$

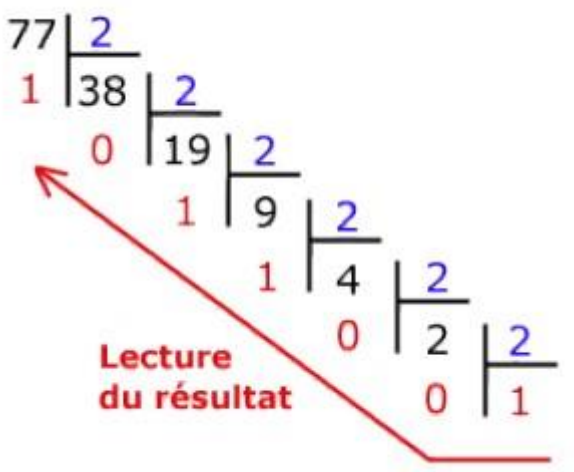
$N_{10} = 8 + 0 + 2 + 1 = 11$

Décimal → Binaire

La conversion inverse peut être facilement effectuée par récurrence.

Il s'agit de faire une suite de divisions euclidiennes par 2. On s'arrête quand on obtient un quotient inferieur a 2.

Le résultat sera la juxtaposition des restes. Le schéma ci-contre explique la méthode utilisée pour convertir l'entier 77 en binaire. 77 s'écrit donc en base 2 : 1001101.



Pour vérifier notre résultat, convertissons 1001101 en décimal à l'aide du schéma ci-dessous :

0 1 0 0 1 1 0 1

2^7 2^6 2^5 2^4 2^3 2^2 2^1 2^0

Le nombre en base 10 est $2^6 + 2^3 + 2^2 + 2^0 = 64 + 8 + 4 + 1 = 77$.

Binaire → Hexadécimal

Pour convertir un nombre binaire en hexadécimal, il suffit de faire des groupes de quatre bits (en commençant depuis la droite). Par exemple, convertissons 1001101 :

Binaire	0100	1101
Pseudo-décimal	4	13
Hexadécimal	4	D

Ainsi N_2 = 1001101 s'écrit donc en base 16 :

N_{16} = 4D.

Pour convertir d'hexadécimal en binaire, il suffit de lire ce tableau de bas en haut.

I.2.3 Codage des caractères

La prochaine étape, une fois qu'on sait stocker en machine des nombres, consiste à pouvoir stocker du texte, que ce soit le contenu d'un livre, un e-mail ou bien un mot de passe. Pour cela, faisons appel à l'abstraction : pour ne pas tout reprendre depuis le début, on peut utiliser le fait que nous savons désormais coder des entiers. Ainsi, on représente chaque

caractère imprimable (« A », « h », « $ », mais aussi « 7 » ou « + ») par un entier. Il s'agit donc de se mettre d'accord afin que tout le monde utilise un codage que les autres peuvent comprendre. Le codage le plus simple consiste à utiliser la table ASCII (c'est l'acronyme de *American Standard Code for Information Interchange*). Cette table contient 128 caractères chacun associé avec un code (décimal) entre 0 et 127 : son contenu est représenté dans la Figure 3. Par exemple, la lettre« A »est codée par l'entier 65. Dans la machine, cet entier est évidemment stocké en binaire : son code binaire est 1000001. Le symbole« # » est représenté par l'entier 35. La table ASCII contient également le code d'éléments utiles pour coder du texte ou des touches du clavier (la touche « escape », le « retour à la ligne » par exemple ou « l'espace », de codes ASCII respectifs 27, 13 et 32).

Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char
0	0	[NULL]	32	20	[SPACE]	64	40	@	96	60	`
1	1	[START OF HEADING]	33	21	!	65	41	A	97	61	a
2	2	[START OF TEXT]	34	22	"	66	42	B	98	62	b
3	3	[END OF TEXT]	35	23	#	67	43	C	99	63	c
4	4	[END OF TRANSMISSION]	36	24	$	68	44	D	100	64	d
5	5	[ENQUIRY]	37	25	%	69	45	E	101	65	e
6	6	[ACKNOWLEDGE]	38	26	&	70	46	F	102	66	f
7	7	[BELL]	39	27	'	71	47	G	103	67	g
8	8	[BACKSPACE]	40	28	(	72	48	H	104	68	h
9	9	[HORIZONTAL TAB]	41	29	)	73	49	I	105	69	i
10	A	[LINE FEED]	42	2A	*	74	4A	J	106	6A	j
11	B	[VERTICAL TAB]	43	2B	+	75	4B	K	107	6B	k
12	C	[FORM FEED]	44	2C	,	76	4C	L	108	6C	l
13	D	[CARRIAGE RETURN]	45	2D	-	77	4D	M	109	6D	m
14	E	[SHIFT OUT]	46	2E	.	78	4E	N	110	6E	n
15	F	[SHIFT IN]	47	2F	/	79	4F	O	111	6F	o
16	10	[DATA LINK ESCAPE]	48	30	0	80	50	P	112	70	p
17	11	[DEVICE CONTROL 1]	49	31	1	81	51	Q	113	71	q
18	12	[DEVICE CONTROL 2]	50	32	2	82	52	R	114	72	r
19	13	[DEVICE CONTROL 3]	51	33	3	83	53	S	115	73	s
20	14	[DEVICE CONTROL 4]	52	34	4	84	54	T	116	74	t
21	15	[NEGATIVE ACKNOWLEDGE]	53	35	5	85	55	U	117	75	u
22	16	[SYNCHRONOUS IDLE]	54	36	6	86	56	V	118	76	v
23	17	[ENG OF TRANS. BLOCK]	55	37	7	87	57	W	119	77	w
24	18	[CANCEL]	56	38	8	88	58	X	120	78	x
25	19	[END OF MEDIUM]	57	39	9	89	59	Y	121	79	y
26	1A	[SUBSTITUTE]	58	3A	:	90	5A	Z	122	7A	z
27	1B	[ESCAPE]	59	3B	;	91	5B	[	123	7B	{
28	1C	[FILE SEPARATOR]	60	3C	<	92	5C	\	124	7C	|
29	1D	[GROUP SEPARATOR]	61	3D	=	93	5D	]	125	7D	}
30	1E	[RECORD SEPARATOR]	62	3E	>	94	5E	^	126	7E	~
31	1F	[UNIT SEPARATOR]	63	3F	?	95	5F	_	127	7F	[DEL]

Table ASCII

Puisqu'on sait coder les caractères, on sait aussi coder une *chaine de caractères*, c'est-àdire du texte. Ainsi la phrase : *Dessine-moi un mouton* est codée par la séquence d'entiers :

68 101 115 115 105 110 101 45 109 111 105 32 117 110 32 109 111 117 116 111 110 46

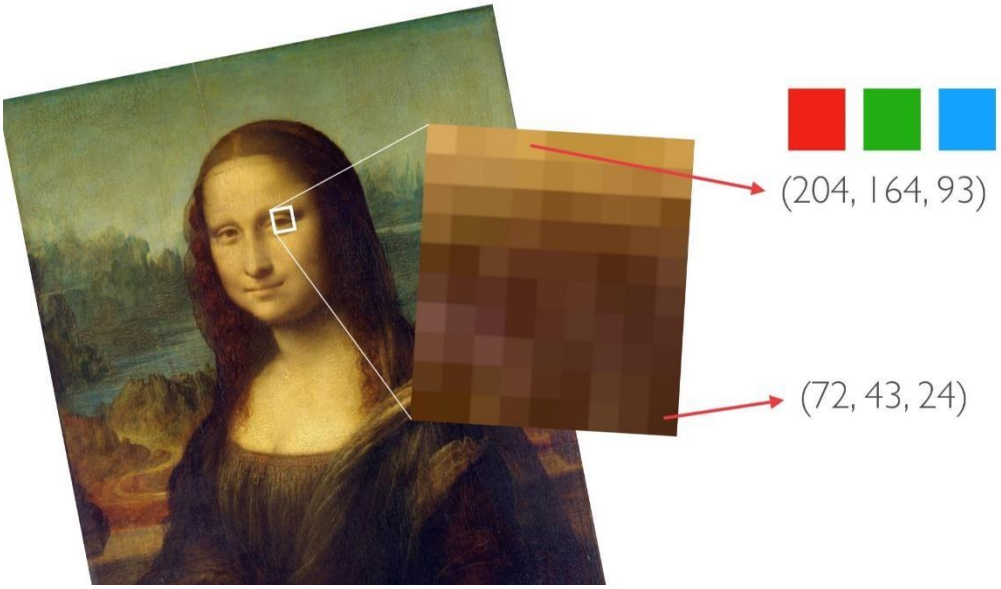
En réalité, évidemment, on ne stocke pas des entiers en décimal, mais bien des entiers codés en binaire, chacun sur 7 bits : par exemple, le tiret« - »est donc codé par la séquence de bits 0101101, ou` on a donc ajouté un 0 en premier pour utiliser les 7 bits à disposition. Puisque chaque caractère est codé sur 7 bits, on a donc pas besoin de séparer les codages de chaque caractère (comme ci-dessus).

Il existe d'autre façons de coder des caractères : citons par exemple d'autres tables, telles que l'UTF-8 (ou UTF-16, ou UTF-32) ou sa généralisation, l'Unicode, permettant de coder bien davantage de caractères (les caractères accentués ou les caractères d'autres alphabets que l'alphabet latin).

I.2.4 Codage des images

On sait désormais coder un roman, mais pas une bande dessinée : il nous manque la possibilité de coder des images. Un format simple (mais gourmand en espace) consiste à représenter une image comme un tableau bidimensionnel (une matrice) : chaque élément du tableau est alors appelé un *pixel*. Le codage d'un pixel consiste à représenter la couleur de la zone correspondante de l'image. Pour ce faire, on utilise généralement trois entiers qui représentent les quantités (entre 0 et 255) de rouge, de vert et de bleu dans la couleur.

Par exemple, dans la Figure ci-dessous, une zone de l'œil de la Joconde est agrandie : cette zone comporte 9 colonnes et 10 lignes de pixels. Deux de ces pixels sont détaillés à droite. Celui du haut a une couleur marron pale qui est codée par le triplet (204, 164, 93) : la couleur est donc constituée de rouge à hauteur de 204/(204 + 164 + 93) = 44,25%. Ainsi, la couleur verte à 100% sera représentée par le triplet (0, 255, 0).



Codage d'une image en format bitmap

Notons que $255 = 2^8 - 1$. Ainsi, la représentation de 255 en binaire est 11111111. Pour coder un entier entre 0 et 255, on a donc besoin de 8 bits. Cette quantité de bits se retrouve souvent en informatique :

On utilise les octets pour rendre également plus lisible la représentation des entiers. Par exemple, l'entier 10^8 a pour représentation binaire 101111101011110000100000000. C'est bien difficile à lire. De la même façon qu'on peut séparer les chiffres par groupes de 3 dans l'écriture décimale d'un nombre (par exemple, 100 000 000), on choisit de séparer les bits de la représentation binaire par groupes de 8 : 101 11110101 11100001 00000000. On utilise souvent l'octet comme unité de capacité mémoire de disques dans le commerce : 256 Go signifie ainsi 256 Giga octets, soit 256 milliards d'octets, ce qui représente donc 256×8 = 2048 milliards de bits.

Revenons au codage des images. Stocker une image de 1920 par 1200 pixels nécessite de représenter 1920 × 1200 = 2 304 000 pixels, chacun prenant 3 octets en mémoire : au total, cette image prend donc 6 912 000 octets (6,9 Mo), soit 55 296 000 bits, ce qui équivaut à 55 Mégabits. C'est beaucoup pour une simple image... En pratique, on ne stocke donc que rarement les images en format bitmap : on utilise plutôt des formats *compressés* qui essaient d'épargner de la mémoire en profitant de redondances dans l'image ou en supprimant des détails invisibles à l'œil nu.

I.2.5 Codage de videos

Toujours grâce au principe d'abstraction, on utilise le fait qu'on sait désormais coder une image pour pouvoir coder une vidéo comme une séquence d'images (environ 30 par secondes, par exemple, afin que l'œil ne puisse pas distinguer les images qui défilent). Là aussi, on n'utilise pas cette représentation naïve en pratique, mais des formats compressés permettent de gagner de la place en mémoire. Pour stocker une vidéo, il faut aussi pouvoir stocker du son ce qui est un problème plus complexe encore qu'on ne traitera pas dans ce cours.

I.3 Exercices

1. Quelle sont les plus grand nombres qu'on peut atteindre avec 8 bits, 16 bits , 32 bits et 64 bits ?
2. Quelle est la valeur décimale qui correspond à la valeur binaire 1110010101 ?
3. Quelle est la valeur binaire qui correspond à la valeur décimale 17 ?
4. Quelle est la valeur binaire qui correspond à la valeur Hexadécimale 4D1 ?
5. Convertir en Hexadécimal la valeur binaire 111011000100111011101
6. Une image de définition 480p est une image qui comporte 720 x 480 pixels. Calculer la taille théorique maximale d'une telle image.
7. Supposons qu'on a une vidéo de la qualité 4k (3840x2160 pixels) d'une durée de 30 secondes avec un framerate FPS = 60. Quelle est la taille théorique maximale de cette vidéo.