**Services Informatiques aux Organisations**

**Fonctionnement d’un ordinateur  
« Codification et Architecture matérielle »**

Lycée Pasteur Mont Roland, Enseignement supérieur ⚫ 40, rue du Gouvernement ⚫ 39100 DOLE

SOMMAIRE

1. ) L'informatique 2

1.1. ) Le fonctionnement du matériel 2

1.2. ) Codification des informations 2

1.2.1. ) Table ASCII de base sur 7 bits 3

1.2.2. ) Table ASCII étendue 3

1.2.3. ) Évolution : 4

2. ) Présentation générale d'un ordinateur (Hardware) 5

2.1. ) Schéma de base 5

3. ) Correspondance des unités 6

3.1. ) Normes 6

3.2. ) Calculs 6

Objectifs :

1. Repérer les différents éléments d'un ordinateur.

|  |
| --- |
| 1. Connaître le schéma global d'un ordinateur |
| 1. Être capable de placer les différents composants dans le schéma |
| 1. Savoir comment fonctionne la codification dans un ordinateur |
| 1. Connaitre les différentes unités en informatique |

Fonctionnement d'un ordinateur

# ) L'informatique

Traitement automatisé de l'information.

Utilise 2 éléments :

1. Le Matériel
2. Le Logiciel

) Le fonctionnement du matériel

L'ordinateur est une machine électrique, nous sommes alors en présence de deux états :

Le courant passe --> valeur 1

Le courant ne passe pas --> valeur 0

L'ordinateur utilise le système, dit : système binaire.

Avec deux symboles 🡺 Circuits très efficaces et peu coûteux.

🡺 Système algébrique de logique établi au XIXème siècle par le mathématicien britannique Georges Boole.

« L'algèbre de Boole, ou calcul booléen, est la partie des mathématiques qui s'intéresse à une approche algébrique de la logique, vue en termes de variables, d'opérateurs et de fonctions sur les variables logiques, ce qui permet d'utiliser des techniques algébriques pour traiter les expressions à deux valeurs du calcul des propositions. Elle fut lancée en 1854 par le mathématicien britannique George Boole. Aujourd'hui, l'algèbre de Boole trouve de nombreuses applications en informatique et dans la conception des circuits électroniques. » Wikipedia

) Codification des informations

Pour distinguer les différentes informations 🡺 Codification

🡺 Utilisation de 8 valeurs binaires.

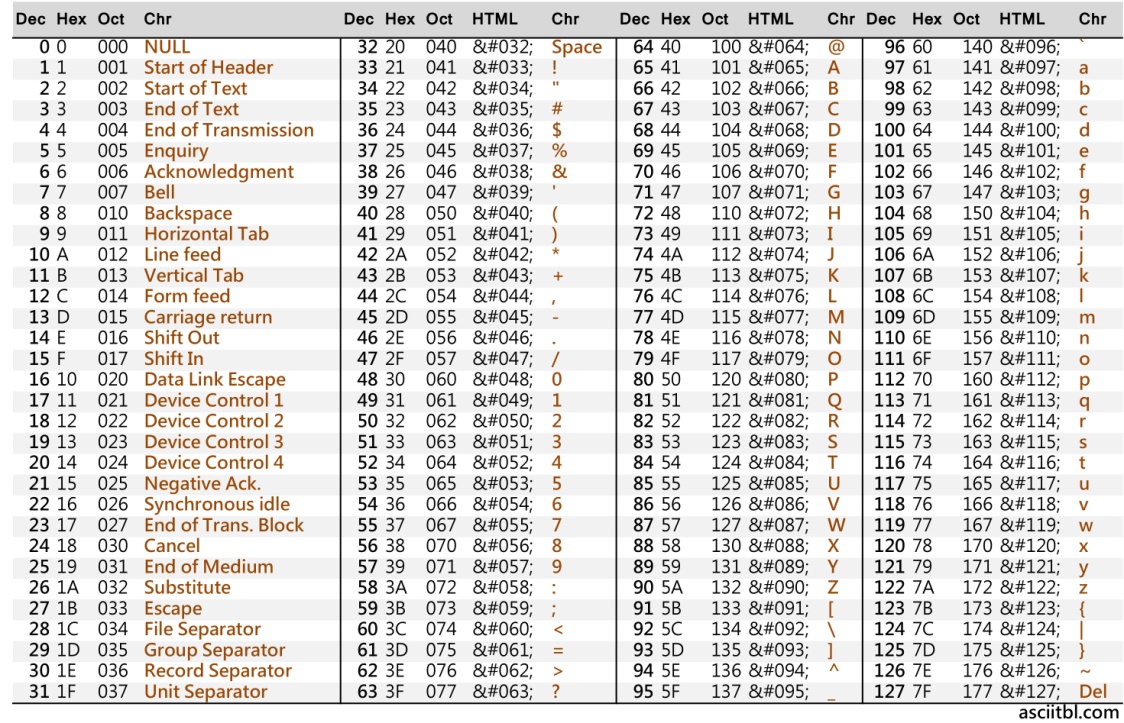
Chacune de ces valeurs est appelée 🡺 Bit

Un ensemble de huit bits est appelé 🡺 Octet

Avec un octet, on peut représenter 🡺 28 = 256

Pour la micro-informatique, la correspondance entre toutes les informations et leurs codes est donnée par la norme 🡺 ASCII = American Standard Code for Information Interchange

) Table ASCII de base sur 7 bits



Cette table contient :

|  |  |
| --- | --- |
| **Type caractères** | **Plages valeurs** |
| Code de commandes | De 0 à 31 et 127 |
| Caractères spéciaux | De 32 à 47, de 58 à 64, de 91 à 96 et de 13 à 126 |
| *Chiffres* | *De 48 à 57* |
| *Caractères majuscules* | *De 65 à 90* |
|  | *Caractères minuscules De 97 à 122* |

) Table ASCII étendue

Avec l'apparition des micro-ordinateurs et de la bureautique, nous sommes passés à une table ASCII étendue sur 8 caractères.

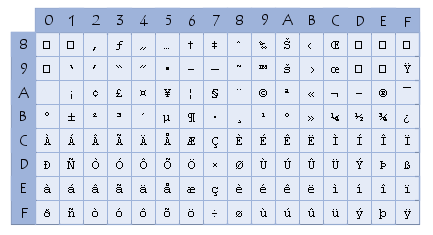
* Le code ASCII étendu OEM, c'est-à-dire celui qui équipait les premières machines de type IBM PC



Cette table commence à quelle valeur ? : 80

Pourquoi ? : 128 en binaire = 10 000 000 = 80 en héxadécimal

* Le code ASCII étendu ANSI, utilisé par les systèmes d'exploitation récents



) Évolution :

* Gestion du graphisme 🡺 utilisation des points ou pixel
* Gestion des caractères de toutes les langues 🡺 Unicode (de 16 à 32 bits pour coder plus d’informations, on utilise au maximum 20 ou 21 bits suivant la norme)

# ) Présentation générale d'un ordinateur (Hardware)

) Schéma de base



*Organes de sortie*

Mémoire cache

*Unité de traitement / processeur*

Interface

Périphériques

Bus

Bus

Organes d’entrée

*Mémoires auxiliaires*

*Organes de liaison*

*Mémoire centrale*

*Unité centrale*

# ) Correspondance des unités

) Normes

Il existe 2 normes de calcul des capacités en informatiques.

1. Soit en base 10 (exemple : KiloOctet ou Ko).
2. Soit en base 2 (exemple : KibiOctet ou Kio).

Vie courante 🡺 mélange des 2 normes (Surtout Windows qui utilise un ko pour un Kio).

Pour les autres utilisateurs, lorsque l’on respecte cette norme, cela pose un problème de compréhension.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Multiples d’octets tels que définis par** [**IEC 60027-2**](http://fr.wikipedia.org/w/index.php?title=IEC_60027-2&action=edit&redlink=1) | | | | | | |
| [**Préfixe SI**](http://fr.wikipedia.org/wiki/Pr%C3%A9fixe_du_syst%C3%A8me_international) (Base 10) | | |  | [**Préfixe binaire**](http://fr.wikipedia.org/wiki/Pr%C3%A9fixe_binaire) (Base 2) | | |
| **Nom** | **Symbole** | **Valeur** | **Nom** | **Symbole** | **Valeur** |
| kilooctet | ko | 103 | kibioctet | Kio | 210 |
| mégaoctet | Mo | 106 | mébioctet | Mio | 220 |
| gigaoctet | Go | 109 | gibioctet | Gio | 230 |
| téraoctet | To | 1012 | tébioctet | Tio | 240 |
| pétaoctet | Po | 1015 | pébioctet | Pio | 250 |
| exaoctet | Eo | 1018 | exbioctet | Eio | 260 |
| zettaoctet | Zo | 1021 | zébioctet | Zio | 270 |
| yottaoctet | Yo | 1024 | yobioctet | Yio | 280 |

Kio 2\*\*10 = 1 024 octets  
Mio 2\*\*20 = 1 048 576 octets  
Gio 2\*\*30 = 1 073 741 824 octets  
Tio 2\*\*40 = 1 099 511 627 776 octets  
Pio 2\*\*50 = 1 125 899 906 842 624 octets

) Calculs

Achat d'un disque de 100 Go :

* Combien contient-il d'octets ? 100 000 000 000 octets
* Quelle valeur une fois formaté ? 93,1 Gio
* Combien d'octets une fois formaté ? 99 965 363 814 octets environ

Achat d'un disque de 1 To :

* Combien contient-il d'octets ?
* Quelle valeur une fois formaté ?
* Combien d'octets une fois formaté ?

Achat d'un disque de 4 To :

* Combien contient-il d'octets ?
* Quelle valeur une fois formaté ?
* Combien d'octets une fois formaté ?