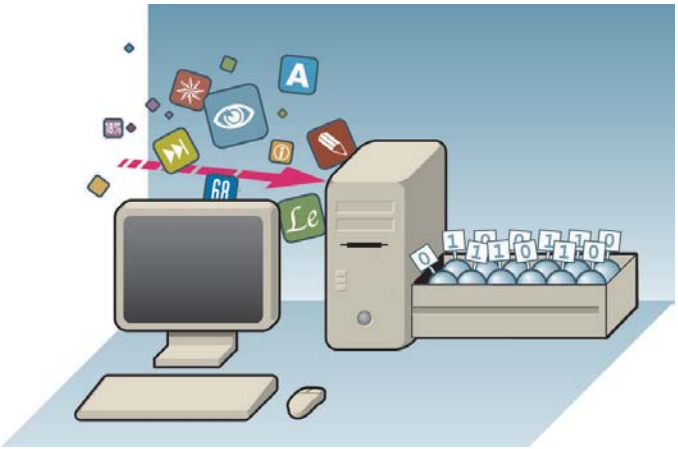
Bits & Bytes

CASES LUXEMBOURG
CYBERWORLD AWARENESS AND SECURITY ENHANCEMENT STRUCTURE

Résumé

Une machine, notamment un ordinateur ou encore un serveur, peut être considérée comme un matériel informatique composé d'équipements électroniques. Ces composants peuvent prendre uniquement deux états et ce sur base de leur signal électrique. Le traitement des données, via une machine, repose exclusivement sur ce concept de numérisation.

Table des matières



1 Qu'est-ce que la notation binaire ?

Lorsque le codage de l'information se fait sur base de deux états (0 ou 1), on parle de base binaire. L'être humain travaille quant à lui avec 10 chiffres de référence (0 à 9), on parle dans ce cas de base décimale.

1.1 Le Bit

Le « Bit - Binary Digit » représente 0 ou 1 en numérotation binaire. Il s'agit donc de la plus petite unité d'information traitable par une machine.

1 bit	peut représenter 2 états : 0 - 1
2 bits	peuvent représenter 22 soit 4 états différents : 00-01-10-11
3 bits	peuvent représenter 23 soit 8 états différents : 000-001-010-011-100-101-110-111
n bits	peuvent représenter 2n états différents

1.2 L'Octet

L'octet est une unité d'information composée de 8 bits, cela permet de stocker des caractères, tels qu'une lettre, un chiffre, etc... En général, on peut dire qu'un octet correspond à un caractère.

8 bits peuvent représenter 28 états soit 256 états différents - il a donc une valeur entre 0 et 255

1.3 Le Byte

Le Byte est l'équivalent anglais de l'octet.

1.4 Kilo, Méga...

Dans toutes les sciences, le kilo représente un facteur de multiplication de 1000 ou 10^3. En informatique on parle de :

1 ko	1 kilo-octet	1 000 octets	**1 kB**	1 kilo-Byte
1 Mo	1 Méga-octet	1 000 000 octets	**1 MB**	1 Mega-Byte
1 Go	1 Giga-octet	1 000 000 000 octets	**1 GB**	1 Giga-Byte
1 To	1 Tera-octet	1 000 000 000 000 octets	**1 TB**	1 Tera-Byte

REMARQUE :

Historiquement, les informaticiens ont utilisé un multiplicateur 1024 pour le kilo, mais depuis décembre 1998 l'organisme international IEC (International Engineering Consortium) a fixé la norme à 1000 comme dans les autres disciplines scientifiques.

Attention !

La différence entre le kilo-bit (habituellement utilisé pour quantifier des bandes passantes exprimées en «kbps» (kilo-bit par seconde) et le kilo-Byte (habituellement utilisé pour quantifier le stockage) exprimé en «kB» (kilo-Byte) n'est pas très visible.

suite au verso

www.cases.lu

Bits & Bytes

CASES LUXEMBOURG
CYBERWORLD AWARENESS AND SECURITY ENHANCEMENT STRUCTURE

suite

2 Qu'est-ce que la notation hexadecimale ?

Vu le caractère illisible pour l'être humain des nombres exprimés en binaire, il a été nécessaire d'introduire une base hexadécimale (base 16).

Afin d'utiliser une base supérieure à notre base 10 usuelle, il a donc été nécessaire d'introduire 6 lettres de l'alphabet.

Base10 Décimale		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Base2 Binaire	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
Base16 Hexadécimale		1	2	3	4	5	6	7	8	9	A	B	C	D	E	F

Pour convertir un octet en hexadécimale il faut partager l'octet en deux groupes de 4 bits et traduire chaque groupe de 4 bits en sa valeur hexadécimale.

REMARQUE :

La calculette intégrée à MS Windows en mode scientifique permet aisément de passer de base 10 en binaire, ou en hexadécimal.

3 Quels sont les contrôles d'erreurs ?

L'utilisation de composants électroniques pour stocker et transmettre des informations est très pratique, mais le signal électrique transmis peut subir des perturbations, cela rend nécessaire l'utilisation de contrôle de validité des données. C'est pourquoi il existe des mécanismes dont le rôle sera de garantir l'intégrité des données. La plupart des systèmes de contrôle d'erreur sont bâtis sur l'ajout d'information « checksum- somme de contrôle » permettant de vérifier la validité des données.

3.1 Le contrôle de parité

Ce système consiste à utiliser un bit de l'octet en tant que bit de parité, il restera donc 7 bits pour l'encodage des données et le 8ème bit pour le contrôle.

Bit de parité	1	1	0	0	0	1	1	0

Le nombre de bits égal à 1 est pair, le bit de parité est donc positionné à « 0 »

Bit de parité	1	1	0	0	0	1	1	1

Si un bit de données est basculé par erreur durant la transmission, le nombre de bits égal à 1 sera impair et donc le bit de parité ne correspondra plus à la parité de l'objet.

Malheureusement, ce mécanisme ne permet de détecter que les erreurs en nombre impair et ne détecte donc pas toutes les erreurs.

3.2 Le contrôle de parité croisé

Ce modèle ne consiste pas uniquement à contrôler l'intégrité des données d'un caractère mais à contrôler l'intégrité des bits de parité d'un bloc de caractères.

Lettre	Codage sur 7 bits	LRC
H	1001000	
E	1000101	1
L	1001100	1
O	1001111	1
VRC	1000010	

Le VRC (Vertical Redundancy Check)
Contrôle vertical de redondance. Désigne la parité appliquée à un mot et non pas à la suite des mots (parité longitudinale).

Le LRC (Longitudinal Redundancy Check)
Contrôle longitudinal de redondance : système de détection d'erreurs par parité s'appliquant à la totalité d'un bloc, par opposition à la parité « verticale » qui s'applique à chaque mot de ce bloc.

3.3 Le contrôle de redondance cyclique

Ce mécanisme consiste à protéger des blocs de données en ajoutant un code de contrôle. Ce code « CRC » contient des éléments redondants par rapport aux données transmises de manière à permettre la détection des erreurs, mais également de les réparer.

www.cases.lu

Bits & Bytes

CASES LUXEMBOURG
CYBERWORLD AWARENESS AND SECURITY ENHANCEMENT STRUCTURE

suite

4 Quels sont les codages des données ?

Afin de pouvoir utiliser les notations définies plus haut, il est encore nécessaire de définir des tables de codage permettant de traduire toutes les informations numériques ou alphabétiques sous forme de valeurs numériques exploitables par une machine.

4.1 Le code EBCDIC

Le code EBCDIC (Extended Binary Coded Decimal Interchange Code) a été développé par IBM et utilisé principalement sur les «Mainframe» (gros calculateurs localisés dans les centres informatiques) n'a jamais séduit le marché de la micro informatique.

4.2 Le code ASCII

Au début des années 1960, le code ASCII (American Standard Code for Information Interchange) voit le jour, et est adopté comme standard.

Le code ASCII de base permettait de représenter les caractères sur 7 bits et offrait donc la possibilité de coder 128 caractères différents.

Les codes 0 à 31	Caractères de contrôle (saut de page,...)
Les codes 33 à 47	Ponctuation
Les codes 48 à 57	Les chiffres 0 à 9
Les codes 58 à 64	Ponctuation
Les codes 65 à 90	Lettres majuscules
Les codes 91 à 96	Ponctuation
Les codes 97 à 122	Lettres minuscules
Les codes 123 à 127	Ponctuation et suppression

REMARQUE :

Le code ASCII ayant été inventé par des anglophones il ne comprend aucun caractère accentué.

4.3 Le code ASCII étendu

Afin de répondre aux besoins internationaux, (caractères accentués et autres caractères propres à certaines langues), il a été mis au point un code ASCII étendu. Un jeu de ce code ASCII a été défini par langue, le code ASCII généralement utilisé au Luxembourg est appelé ISO LATIN15.

Afin de supporter tous les nouveaux caractères, ce code ASCII étendu utilise 8 bits offrant donc des valeurs allant de 0 à 255.

Cependant, la liaison des codes à une langue définie pose de gros problèmes de paramétrage des périphériques tels que les imprimantes, mais surtout représente un énorme obstacle à l'échange de documents.

4.4 Le code UNICODE

Début des années 1990, suite au développement des environnements graphiques, MS Windows et autres, le code UNICODE fit son apparition.

Ce système de codage sur 16 bits (2 octets) utilise des codes de 0 à 65 535 et permet de coder tous les alphabets et les caractères indépendamment du système d'exploitation et du langage de programmation.

5 Quels sont les liens utiles ?

- http://www.asciitable.com
- http://www.unicode.org

www.cases.lu