



*Conservatoire National des Arts et Métiers
FOD Ile de France*

LA CODIFICATION BINAIRE

28 octobre 2008

<i>Versions</i>	<i>Auteur</i>	<i>Commentaires</i>
<i>10 octobre 2003</i>	<i>Emile Geahchan</i>	<i>Version Initiale</i>
<i>31 octobre 2005</i>	<i>Emile Geahchan</i>	<i>Remise en forme</i>
<i>23 octobre 2010</i>	<i>Éric Bourgeois</i>	<i>Remise en forme ODT</i>

Sommaire

LA CODIFICATION BINAIRE.....	1
I- INTRODUCTION.....	2
II- LA NOTATION HEXADECIMALE.....	3
Tables de conversion décimal/binaire/hexadécimal :.....	4
Exercice 1.....	4
III- LA TABLE ISO LATIN-1.....	5
Exercice 2.....	6
Exercice 3.....	6
Editeur Hexadécimal.....	6
Annexe A : Corrigés des Exercices.....	7

Ce document est un support de cours à l'usage exclusif des auditeurs du Cnam dans le cadre de leur formation.

Ce document est protégé par le code de la propriété individuelle (livre 1er article L111-1 à L113-9).

Tout autre usage est interdit sans l'autorisation écrite du Cnam.

Histoire des Réseaux 1/7



I- INTRODUCTION

Les ordinateurs du marché sont des ordinateurs numériques (ou digitaux), il existe des ordinateurs analogiques mais ce sont des monstres de laboratoire.

Le système numérique utilisé par les ordinateurs du marché est le système binaire (base 2), et cela pour des raisons technologiques (les circuits électroniques à 2 états sont de loin les moins coûteux à performances égales), il existe des ordinateurs ternaires (base 3), décimaux (base 10) mais ce sont des monstres de laboratoire.

La codification informatique est donc une codification binaire :

- codification des instructions machine,
- codification des nombres,
- codification des caractères,
- numérisation puis codification des signaux analogiques : son, images, vidéo,
- d'une façon générale codification de tous les objets manipulés par les ordinateurs.

La référence aujourd'hui en termes de stockage ou de transmission de données est l'octet (8 bits). Stocker de l'information ou transmettre de l'information, revient donc à stocker ou à transmettre des octets.

Vous devez savoir manipuler des octets binaires, on utilise pour cela la notation hexadécimale.

Vous devez connaître la structure table de codage des caractères ISO Latin-1 c'est la table de codage par défaut pour les pages HTML.



II- LA NOTATION HEXADÉCIMALE

Lorsque l'on manipule des octets binaires,
il est bien plus pratique de représenter cette information à l'aide de chiffres hexadécimaux
(système de numération de base 16)

un chiffre hexadécimal correspond à 4 bits

donc un octet correspond à 2 chiffres hexadécimaux soit $2 \times 4 = 8$ bits.

Exemple : octet 0001 0000 = 10 (hexa)

■ Rappels Mathématiques

Système de numération de base 10 :

Nombre = $a_n \times 10^n + a_{n-1} \times 10^{n-1} + \dots + a_2 \times 10^2 + a_1 \times 10 + a_0$

exemple : $1956 = 1 \times 10^3 + 9 \times 10^2 + 5 \times 10 + 6$

on utilise les 10 chiffres : **0, 1, 2, 3, 4, 5, 6, 7, 8 et 9**

Système de numération de base 2 :

Nombre = $a_n \times 2^n + a_{n-1} \times 2^{n-1} + \dots + a_2 \times 2^2 + a_1 \times 2 + a_0$

exemple : $1100 = 1 \times 2^3 + 1 \times 2^2 + 0 \times 2 + 0$ et en décimal = $8 + 4 = 12$

$100 = 1 \times 2^2 + 0 \times 2 + 0$ et en décimal = 4

on utilise les 2 chiffres : **0 et 1**

Système de numération de base 16 :

Nombre = $a_n \times 16^n + a_{n-1} \times 16^{n-1} + \dots + a_2 \times 16^2 + a_1 \times 16 + a_0$

exemple : 20 hexa = $2 \times 16^1 + 0 = 32$ décimal

25 hexa = $2 \times 16^1 + 5 = 37$ décimal

on utilise les 16 chiffres : **0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E et F**



Tables de conversion décimal/binaire/hexadécimal :

<i>Unités hexadécimales</i>		
<i>Hexa</i>	<i>Binaire</i>	<i>Décimal</i>
0	0000	0
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5
6	0110	6
7	0111	7
8	1000	8
9	1001	9
A	1010	10
B	1011	11
C	1100	12
D	1101	13
E	1110	14
F	1111	15

<i>Dizaines hexadécimales</i>		
<i>Hexa</i>	<i>Binaire</i>	<i>Décimal</i>
0	0000 0000	0
10	0001 0000	16
20	0010 0000	32
30	0011 0000	48
40	0100 0000	64
50	0101 0000	80
60	0110 0000	96
70	0111 0000	112
80	1000 0000	128
90	1001 0000	144
A0	1010 0000	160
B0	1011 0000	176
C0	1100 0000	192
D0	1101 0000	208
E0	1110 0000	224
F0	1111 0000	240

Vous devez savoir convertir un octet binaire en hexadécimal puis en décimal.

Exemple

les 4 octets : **1100 0000** **1010 1000** **0000 0000** **0000 0001**
 se représentent en hexadécimal par : **C0** **A8** **00** **01**

Si l'on veut convertir chaque octet en décimal, on obtient :

C0 = 12x16 + 0x1 = 192
 A8 = 10x16 + 8x1 = 168
 0 = 0x16 + 0x1 = 0
 1 = 0x16 + 1x1 = 1

➤ un octet se traduit par un nombre décimal compris entre 0 et 255

■ **Exercice 1**

➤ Convertir en hexadécimal puis en décimal les octets suivants :

1010 0001 ; 0111 0000 ; 0000 1111 ; 1111 0000 ; 1100 0000 ; 1010 1000 ; 0000 0000 ; 0000 0001

➤ Représenter en binaire puis en hexadécimal les octets contenant les nombres décimaux suivants :

0, 1, 2, 4, 8, 16, 32, 64, 128, 255



III- LA TABLE ISO LATIN-1

Parmi tous les systèmes de codifications des caractères sur un octet ([ASCII](#), [ANSI](#), [EBCDIC](#), [pages codes MSDOS](#) etc.), ce sont les tables ISO 8859-X qui l'emportent sur le marché.

La première demi-table ISO Latin-1 (ou [ISO 8859-1](#)) correspond à l'ancien standard ASCII américain. Elle comporte les lettres latines majuscules et minuscules, les 10 chiffres ainsi qu'un certain nombre d'autres caractères usuels (opérateurs + - * / parenthèses etc.).

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

A = 41 hexa

7F hexa

* Les 20 (hexa; 32 en décimal) premiers caractères ainsi que le dernier caractère (7F hexa) sont des caractères spéciaux non imprimables :

exemple : « 0D » = retour chariot (CR Carriage Return)

« 0A » = saut de ligne (LF Line Feed).

La seconde demi-table ISO Latin-1 comporte les lettres accentuées utilisées en Europe de l'ouest ainsi que certains caractères spéciaux complémentaires.

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
8																
9																
A		¡	¢	£	¤	¥	¦	§	¨	©	ª	«	¬	•	®	¯
B	°	±	²	³	´	µ	¶	·	¸	¹	º	»	¼	½	¾	¿
C	À	Á	Â	Ã	Ä	Å	Æ	Ç	È	É	Ê	Ë	Ì	Í	Î	Ï
D	Ð	Ñ	Ò	Ó	Ô	Õ	Ö	×	Ø	Ù	Ú	Û	Ü	Ý	Þ	ß
E	à	á	â	ã	ä	å	æ	ç	è	é	ê	ë	ì	í	î	ï
F	ð	ñ	ò	ó	ô	õ	ö	÷	ù	ú	û	ü	ý	þ	ÿ	

é = E9 hexa

- Les 20(hexa) premiers caractères sont des caractères spéciaux non imprimables.



■ Exercice 2

- Coder en ISO Latin-1 : BONJOUR

■ Exercice 3

1. Utiliser le bloc-notes Windows(ou tout autre éditeur qui utilise la codification ISO Latin-1)
2. Créer un document test.txt contenant le texte :

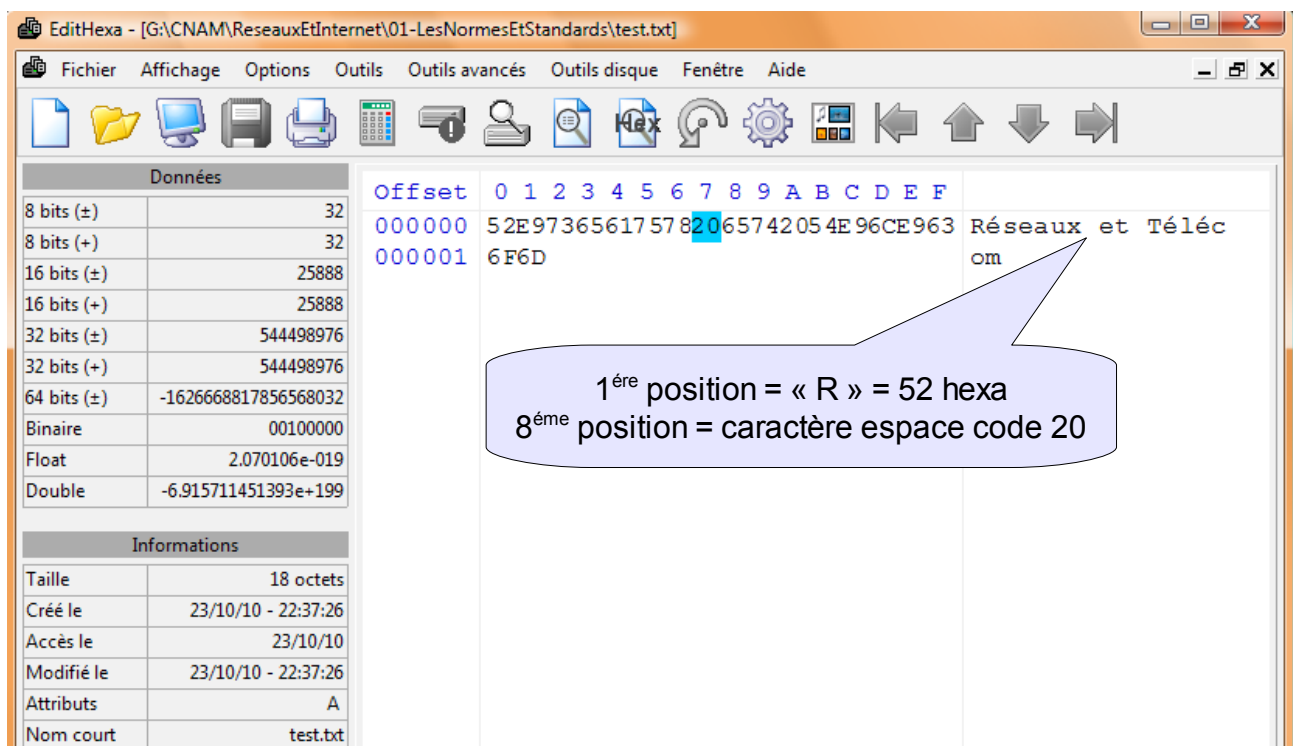
Réseaux et Télécom

3. Enregistrer test.txt
4. Télécharger et installer l'éditeur binaire CR-Hexact (ou tout autre éditeur binaire de votre choix)
5. Ouvrez le fichier test.txt avec l'éditeur binaire :

Vous verrez apparaître à gauche le [dump binaire](#) et à droite le texte en ISO Latin 1.

* Si vous êtes sous Unix vous pouvez utiliser la commande : `od -x test.txt`

Editeur Hexadécimal



Données	
8 bits (±)	32
8 bits (+)	32
16 bits (±)	25888
16 bits (+)	25888
32 bits (±)	544498976
32 bits (+)	544498976
64 bits (±)	-1626668817856568032
Binaire	00100000
Float	2.070106e-019
Double	-6.915711451393e+199

Informations	
Taille	18 octets
Créé le	23/10/10 - 22:37:26
Accès le	23/10/10
Modifié le	23/10/10 - 22:37:26
Attributs	A
Nom court	test.txt

```
Offset  0 1 2 3 4 5 6 7 8 9 A B C D E F
000000  5 2E973656175782065742054E96CE963  Réseaux et Téléc
000001  6F6D                                     om
```

1^{ère} position = « R » = 52 hexa
8^{ème} position = caractère espace code 20

EditHexa est un éditeur hexadécimal qui permet les opérations classiques

(recherche, modification, ajout de séquence, suppression, écrire un nombre...)

Vous avez aussi la possibilité d'éditer secteur par secteur un lecteur (pour un disque dur niveau logique et physique), la modification est en outre possible. Le programme permet aussi de récupérer la structure du secteur de boot et du Master Boot Record ainsi que la navigation dans les entrées du répertoire racine pour une partition FAT12/16/32

EditHexa permet de récupérer la structure (header, sections, fonctions importées et exportées, ressources ...) des exécutables 32/64 bits. Vous avez aussi la possibilité d'extraire sous forme de fichiers directement utilisables certaines ressources (wave, bmp, avi, curseur, icône)des exécutables 32 bits.



Annexe A : Corrigés des Exercices

Exercice 1

<i>Binaire</i>	<i>Hexa</i>	<i>calcul</i>	<i>Décimal</i>
1010 0001	A1	$(A \rightarrow 10 \times 16) + 1$	161
0111 0000	70	$(7 \times 16) + 0$	112
0000 1111	0F	$(0 \times 16) + 15 (F)$	15
1111 0000	F0	$(F \rightarrow 15 \times 16) + 0$	240
1100 0000	C0	$(C \rightarrow 12 \times 16) + 0$	192
1010 1000	A8	$(A \rightarrow 10 \times 16) + 8$	168
0000 0000	00	$(0 \times 16) + 0$	0
0000 0001	01	$(0 \times 16) + 1$	1

<i>Décimal</i>	<i>Binaire</i>	<i>Hexa</i>
0	0000 0000	00
1	0000 0001	01
2	0000 0010	02
4	0000 0100	04
8	0000 1000	08
16	0001 0000	10
32	0010 0000	20
64	0100 0000	40
128	1000 0000	80
255	1111 1111	FF

Exercice 2

B	O	N	J	O	U	R
42	4F	4E	4A	4F	55	52

Exercice 3

caractères non-imprimables OD (retour chariot) et le OA (saut de ligne) sont interprétés par des points.

