

TD séance n° 10

Multimédia : Texte

Ce TD a pour vocation de prendre un peu de recul sur la notion de multimédia que vous utilisez largement. Au lieu de nous intéresser aux usages des outils multimédia (comme vous avez déjà pu le faire au travers d'activités liées au B2i), nous allons nous concentrer sur les aspects plus fondamentaux afin de comprendre comment l'information est stockée dans un ordinateur pour ces différents types de fichiers multimédia.

Ce TD est donc le premier d'une série de 3 sur le multimédia. Nous débutons par le texte ce qui nous permettra de voir comment celui est enregistré dans les fichiers que vous manipulez, puis nous continuerons par le son et enfin les images.

1 Introduction générale



C'est quoi le multimédia ?



Le Multimédia qualifie l'intégration de plusieurs moyens de représentation de l'information, tels que textes, sons, images fixes ou animées.

1.1 Objets multimédias

Nous appellerons ces différents moyens de représenter de l'information multimédia, des « objets multimédias ». Les objets multimédias de base sont donc : du texte, du son, de l'image et de la vidéo.

Au moins un type de dispositif d'acquisition permet de créer un objet multimédia (ex. un enregistrement sonore) puis au moins un type de dispositif de restitution permet de le rendre accessible à l'utilisateur (ex. par le biais de haut-parleurs). Vous les connaissez bien pour les utiliser fréquemment :

Type d'objet multimédia	Dispositif d'acquisition	Dispositif de restitution
Du texte	Clavier, ...	Ecran, Imprimante, ...
Du son	Microphone, ...	Haut-parleurs, ...
De l'image	Appareil photo, scanner, ...	Ecran, imprimante, ...
De la vidéo	Caméra, ...	Ecran, haut-parleurs, vidéoprojecteur...

Tableau 1: Types de multimédia et dispositifs d'acquisition et de restitution

1.2 Objet multimédia composites

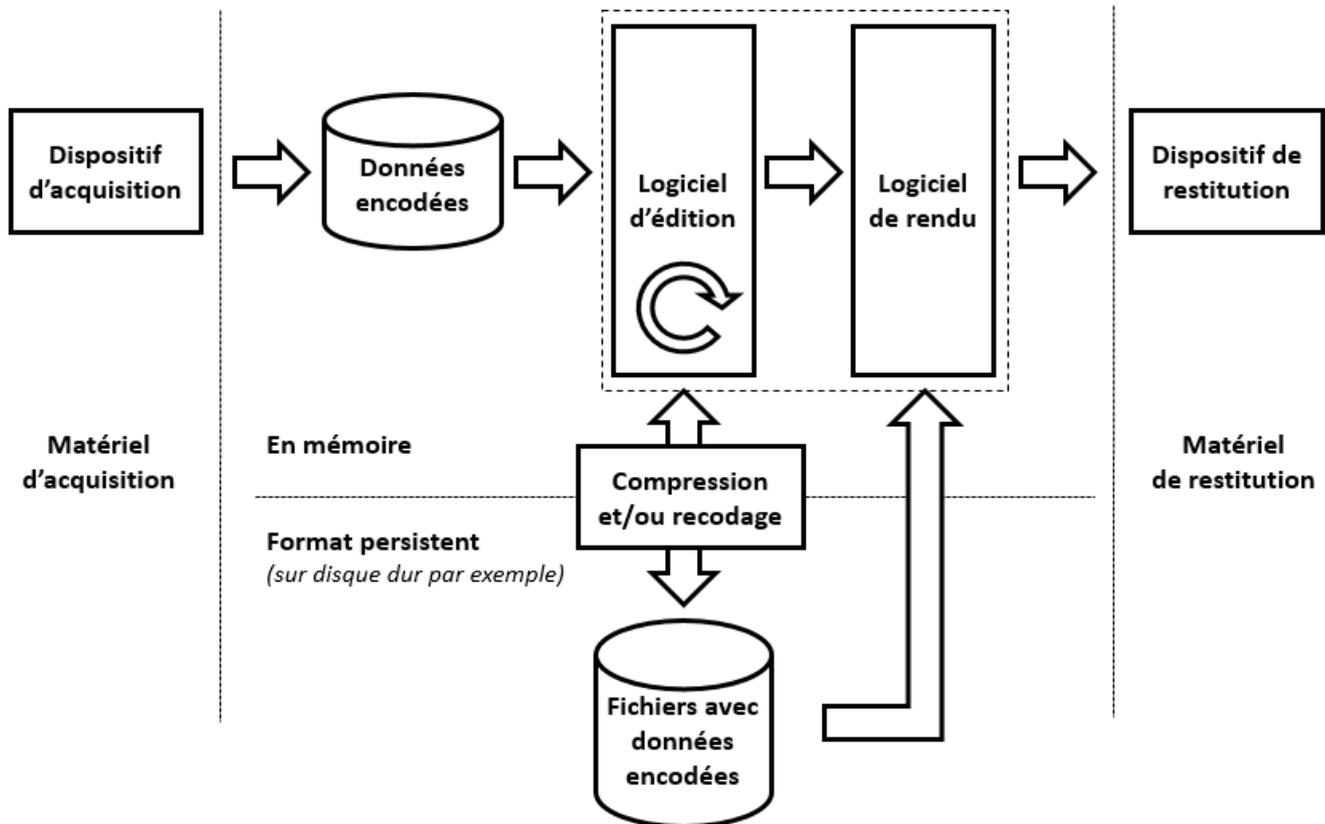
L'intégration de plusieurs objets multimédias de base se fera dans un **objet multimédia dit composite**.

Remarquons que la vidéo devrait être considérée comme un objet multimédia composite (images animées et son), même si elle est communément considérée comme un objet multimédia de base.

1.3 Chaîne de traitements multimédia : les grands principes

Une « chaîne de traitements multimédia » est l'ensemble des traitements et modifications possibles sur un objet multimédia entre le dispositif d'acquisition et le dispositif de restitution. Le schéma général d'une chaîne de traitements multimédia peut-être représenté ainsi :

TD séance n° 10 Multimédia : Texte



Elle est constituée de différents traitements modifiant l'objet multimédia. Ce dernier peut être sauvegardé dans des fichiers, sous différents formats (format des fichiers avec des extensions qui permet à l'utilisateur de les identifier facilement), à chaque étape (ex. pour le son d'un fichier .wav vers un fichier .mp3).

Voici les détails des différents éléments de la chaîne de traitement :

- Les dispositifs d'acquisition ont été pour la plupart déjà énumérés dans le tableau ci-dessus pour les différents types d'objets multimédias.
- Données avec un codage de base sont celles qui contiennent l'information enregistrée depuis le dispositif d'acquisition sans traitement.
- Les traitements de compression et/ou recodage peuvent avoir plusieurs objectifs. Le premier est de recoder l'information dans les multiples formats plus adaptés aux logiciels qui vont la lire. Le deuxième est de réduire la taille de l'information stockée dans un fichier (la compression d'un fichier en .zip par exemple) Une fois compresser et/ou recoder, l'information sera enregistrée dans des fichiers avec un encodage différent de celui de base.
- Les fichiers avec un autre encodage sont ceux qui contiennent l'information enregistrée sous différents formats après traitements dans la chaîne.
- Le logiciel d'édition permet alors de lire un fichier multimédia avec un encodage approprié et permet à l'utilisateur de le modifier et de l'enregistrer à nouveau sous forme de fichier.
- Le logiciel de rendu est celui qui va permettre à d'autres utilisateurs de lire le fichier final une fois toutes les modifications de l'utilisateur effectuées sur le logiciel d'édition. Il peut ainsi rendre l'objet multimédia définitif accessible à tout utilisateur (pour le son : par exemple avec Real Player, Winamp ...).
- Les dispositifs de restitution ont été pour la plupart déjà énumérés dans le tableau ci-dessus.

TD séance n° 10 Multimédia : Texte

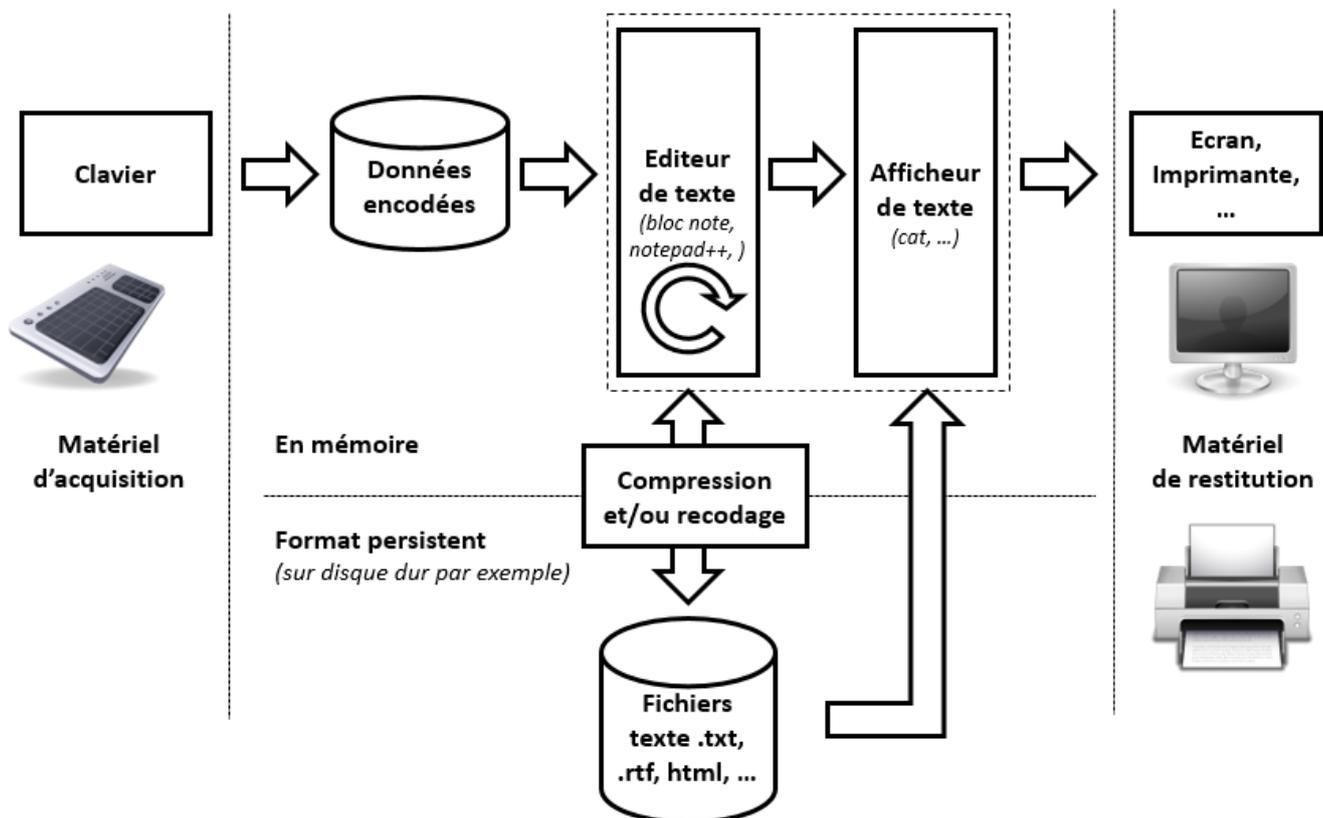
Finalement, la majeure partie des logiciels que nous utilisons implémentent ces chaînes de traitement pour les différents types de multimédias. Elles sont parfois « insolites ». En faisant preuve d'un peu d'imagination, on peut même en imaginer des nouvelles....

Mais commençons pas le premier type de données que nous allons étudier : le texte et son encodage.

2 Multimédia : Texte

2.1 Chaîne de traitements pour le texte

Nous allons donc étudier une chaîne de traitement pour manipuler du texte depuis sa saisie jusqu'à son rendu définitif (ex. son impression ou son rendu à l'écran). Vous trouverez ci-dessous les spécificités d'une chaîne de traitement pour texte.



2.2 Codage de texte

Comme vous pouvez vous en douter, les codages de texte sont plus ou moins riches, selon les informations qu'on ajoute au texte pour gérer la mise en forme (ex. dans les fichiers .rtf) et parfois même pour y insérer des images (ex. dans les fichiers .doc et .docx de Word). Remarquons que dans ce dernier cas, nous manipulons alors un objet multimédia composite...

Et oui tout n'est pas simple... Mais la première question fondamentale concernant le texte est : comment sont représentés les caractères dans un ordinateur. Soulevons un peu le voile sur le codage de l'information.

3 Codage de l'information

Pour mieux comprendre, regardons comment l'information en général est codée puis stockée dans un ordinateur.

TD séance n° 10

Multimédia : Texte

3.1 Codage de base de l'information dans un ordinateur

L'unité de base pour le stockage de l'information dans un ordinateur est le bit (qui signifie en anglais « *binary digit* »). Dans un bit, on peut stocker une information très simple : soit 1 soit 0 (vrai / faux, oui / non, ying / yang, ...). On parle alors de système binaire.

3.1.1 Bits et Octets

Pour stocker et représenter une information plus complexe que cette information très simple, on va regrouper les bits par paquets de 8 ; on appellera un paquet de 8 bits, 1 octet (octet vient du latin et du grec signifiant octo, signifiant 8).

Avec un octet (ou « *Byte* » en anglais), on pourra représenter une information plus complexe, avec toutes les possibilités de combinaisons de 0 et de 1 pour chacun des 8 bits. On pourra alors coder par exemple des nombres entre 0 et 255 (soit 256 valeurs = 2^8). Et si on a besoin de stocker des valeurs encore plus grandes, on fera des paquets de 2 octets (16 bits), 4 octets (32 bits) ou 8 octets (64 bits).

Le symbole utilisé pour l'unité de mesure de l'octet dans le système international d'unités est le « o » minuscule.

3.1.2 Multiples normalisés de l'octet

Beaucoup d'informaticiens ont appris que 1 kilo = 1024. Pourquoi 1024 ? Car $2^{10} = 1024$. Cependant cette tradition viole les normes en vigueur pour les unités de mesure. Une norme a été créée en 1998 pour respecter les normes internationales en vigueur à savoir 1 kilo = $10^3 = 1000$. Donc dans le système international d'unités, la dénomination est :

Nom unité	Symbole	Valeur	Nombre d'octets
Un kilooctet	ko	10^3 octets	1.000 octets
Un mégaoctet	Mo	10^6 octets	1.000.000 octets
Un gigaoctet	Go	10^9 octets	1.000.000.000 octets
Un téraoctet	To	10^{12} octets	1.000.000.000.000 octets
Un pétaoctet	Po	10^{15} octets	1.000.000.000.000.000 octets
Un exaoctet	Eo	10^{18} octets	1.000.000.000.000.000.000 octets
Un zettaoctet	Zo	10^{21} octets	1.000.000.000.000.000.000.000 octets
Un yottaoctet	Yo	10^{24} octets	1.000.000.000.000.000.000.000.000 octets

Pour conserver cette habitude acquise par les informaticiens de considérer que 1 kilo = 1024, les préfixes « binaires » suivants sont définis :

Nom unité	Symbole	Valeur	Nombre d'octets
Un kibioctet	Kio	2^{10} octets	1.024 octets
Un mébioctet	Mio	2^{20} octets	1.048.576 octets
Un gibioctet	Gio	2^{30} octets	1.073.741.824 octets
Un tébioctet	Tio	2^{40} octets	1.099.511.627.776 octets
Un pébioctet	Pio	2^{50} octets	1.125.899.906.842.624 octets
Un exbioctet	Eio	2^{60} octets	1.152.921.504.606.846.976 octets
Un zebioctet	Zio	2^{70} octets	1.180.591.620.717.411.303.424 octets
Un yobioctet	Yio	2^{80} octets	1.208.925.819.614.629.174.706.176 octets

L'usage traditionnel reste largement en vigueur chez les professionnels comme le grand public, même si c'est en contradiction avec les recommandations qui définissent clairement d'autres préfixes. L'usage des préfixes binaires

TD séance n° 10

Multimédia : Texte

reste assez confidentiel et ne se répand presque pas dans le langage courant, alors que les valeurs représentées par ces unités en puissance de 2 sont très utilisées dans les applications, notamment les systèmes d'exploitation.

Cette distinction (entre préfixes binaires et décimaux) est nécessaire, car la confusion entre les deux séries de coefficients est utilisée depuis longtemps par les fabricants de matériels (de disques durs par exemple) pour afficher commercialement des valeurs supérieures à celles données par les puissances de 2 et ainsi introduire une erreur d'appréciation (concernant la taille réelle d'un disque dur par exemple). Bien que les constructeurs indiquent sur leurs composants des valeurs décimales : « 32GB » représentant $32 \cdot 10^9$ octets, ces logiciels vont indiquer, soit correctement « 29.8GiB » soit incorrectement « 29.8GB ». Ces confusions ont été à l'origine de plusieurs actions en justice, notamment aux USA¹.

3.2 Base

En arithmétique, un même nombre peut être représenté dans différentes bases. Pour rappel, une base désigne la valeur dont les puissances successives interviennent dans l'écriture des nombres, ces puissances définissant l'ordre de grandeur de chacune des positions occupées par les chiffres composant tout nombre.

Vous manipulez couramment les nombres en base 10 (décimale). Cela signifie que vous avez 10 symboles (les chiffres de 0 à 9) pour représenter un nombre (10, 123, 9876, ...). Dans le cas d'une notation binaire, on a que deux symboles pour coder l'information : 0 et 1 comme on l'a vu précédemment. Donc le nombre 11 (en décimal) se codera en binaire : 00001011 ($2^3 + 2^1 + 2^0 = 8 + 2 + 1$).

Mais nous pouvons aussi utiliser d'autres bases que la base 10 ou la base 2 pour représenter les nombres. Par exemple, en informatique, on utilise souvent la base 16 (système hexadécimal, facilitant les conversions en base 2 en regroupant des chiffres binaires, 16 étant une puissance de 2). Pour représenter les chiffres au-delà de 9, on utilisera les lettres. Par exemple, pour la base 16, les symboles utilisés sont 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F.

Vous pouvez utiliser le tableau en annexe (4 premières colonnes) pour prendre quelques exemples de conversions entre nombre décimaux, hexadécimaux ou binaires.

3.3 Représentation des nombres en binaire

On peut donc représenter grâce aux informations binaires (0 et 1) des nombres comme dans l'exemple ci-dessus. Mais combien de bits sont nécessaires pour coder des nombres entiers naturels ?

Comme nous avons pu le voir précédemment, sur un octet, donc à l'aide de 8 bits, nous pouvons représenter les entiers naturels entre 0 et 255, soit 2^8 valeurs possibles. Donc d'une manière générale, un codage sur n bits permettra de représenter des nombres entre 0 et $2^n - 1$.

Mais peut-on coder des entiers relatifs (un nombre relatif étant un entier pouvant être négatif) ? Eh bien oui ! Nous ne rentrerons pas ici dans les détails, mais sachez que l'on utilise un codage binaire que l'on appelle le complément à deux. Ainsi :

- Sur 8 bits (1 octet), l'intervalle de codage est $[-128, 127]$.
- Sur 16 bits (2 octets), l'intervalle de codage est $[-32768, 32767]$.
- Sur 32 bits (4 octets), l'intervalle de codage est $[-2147483648, 2147483647]$.

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

Et on peut bien sûr aussi encoder des nombres réels (sinon, comment fonctionnerait même votre simple calculatrice). Mais revenons au texte qui nous intéresse aujourd'hui.

¹ <https://fr.wikipedia.org/wiki/Octet>

TD séance n° 10

Multimédia : Texte

3.4 Et le codage du texte dans tout cela...

Nous avons vu que nous pouvions coder toute sorte de nombres dans un ordinateur, mais les caractères pour écrire un texte ? En fait, dans un ordinateur, chaque lettre est codée à l'aide d'un nombre (et nous avons vu qu'un nombre peut être codé en binaire). Il faut « juste » définir à quel caractère correspond quelle valeur.

Parmi un grand nombre de codages possibles, le plus répandu historiquement est le code ASCII étendu (voir le tableau en annexe). Par exemple, le « A » aura pour valeur 65, le « Z » aura pour valeur 90, etc. Mais vous pouvez constater que le nombre de caractères est limité à 256 valeurs possibles. Donc chaque caractère est codé sur 1 octet.

Dans le monde, bien d'autres symboles existent dans les différentes langues : allemand, espagnol, grec, arabe, chinois, coréen, japonais, etc. On a donc besoin de bien plus de 255 caractères. On a donc d'autres encodages qui ont été normalisés et dit « UNICODE » comme UTF-8, UTF-16, UTF-32 qui utilisent respectivement 8, 16 et 32 bits pour exprimer les valeurs représentant les caractères.

Pour bien comprendre cela, vous pouvez ouvrir des fichiers, dans leur format numérique, c'est-à-dire avant que les valeurs des octets du fichier soient converties en caractères. Vous pouvez aussi retrouver des problèmes de codage en ouvrant des fichiers texte créés sous Windows en ASCII/ANSI avec des éditeurs sous Linux. Ça sera dans ce cas les caractères de saut de ligne qui ne sont pas les mêmes...

Heureusement la plupart des éditeurs gèrent plusieurs encodages en les identifiant, comme vous allez le voir dans les exercices suivants.

3.5 Théorie de l'information

Et tout cela ne se résume pas simplement à des chiffres et à des conversions. Un des piliers de l'informatique repose sur la théorie de l'information de Shannon² qui est une théorie probabiliste permettant de quantifier le contenu moyen en information d'un ensemble de messages, dont le codage informatique satisfait une distribution statistique précise. Cette théorie, communément appelée « théorie de l'information » ou « théorie de la communication », est fondatrice du codage de l'information, de la mesure quantitative de redondance dans un texte, de la compression de données et de la cryptographie.

² Shannon C. E., "A Mathematical Theory of Communications", The Bell System Technical Journal, Vol. 27, 1948.

TD séance n° 10 Multimédia : Texte

Exercices

3.6 Objets multimédia composites

Exercice n°1:

Voyez-vous d'autres exemples d'objets multimédias composites que ceux cités dans le cours ?

3.7 Chaînes de traitements multimédia

Exercice n°2:

Pouvez-vous donner un exemple d'une chaîne de traitement multimédia qui permet d'acquérir du son et de restituer du texte ?

Où vient alors s'insérer le traitement spécifique qui se rajoute à la chaîne de traitement multimédia entre le Microphone et l'Écran par exemple ? Quel est-il ?

Exercice n°3:

Pouvez-vous donner un exemple d'une chaîne de traitement multimédia qui permet d'acquérir de l'image et de restituer du texte ?

Où vient alors s'insérer le traitement spécifique qui se rajoute à la chaîne de traitement multimédia entre le Scanner et l'Écran par exemple ? Quel est-il ?

3.8 Bases

Si aucune notation de base n'est précisée, c'est que le nombre est exprimée dans la base décimale.

Exercice n°4:

De la base 10 vers la base 2. Donner l'écriture en base 2 des nombres suivants :

$$A = 10 \quad B = 31 \quad C = 32$$

Exercice n°5:

De la base 2 vers la base 10. Convertir en base 10 les nombres suivants :

$$D = 101001_{(2)} \quad E = 10110011_{(2)} \quad F = 1100101_{(2)} \quad G = 100010111_{(2)}$$

Exercice n°6:

Quel est le nombre le plus grand pour chacun de ces couples de nombres ?

$$H : 10000_{(2)} \text{ et } 64 \quad I : 10000001_{(2)} \text{ et } 101 \quad J : 10000000_{(2)} \text{ et } 128 \quad K : 11111111_{(2)} \text{ et } 256$$

3.9 Codage du texte

Nous allons voir que de simples fichiers texte, sans gérer de mise en forme particulière à l'exception du retour à la ligne, peuvent déjà être codés différemment. Pour les exercices suivants, nous allons utiliser des programmes disponibles sous Windows. Vous démarrerez donc la machine virtuelle Windows grâce à la commande `vmWindows` que vous lancerez depuis un terminal.

Exercice n°7:

Créez un fichier `texteANSI.txt` sous Windows avec Notepad++ que vous aurez préalablement installé depuis <http://notepad-plus-plus.org/fr/>. Dans ce fichier, mettez le texte « Bonjour ! » et enregistrez-le.

TD séance n° 10

Multimédia : Texte

Exercice n°8:

Créez alors deux nouveaux fichiers textes `texteUTF8.txt` et `texteUTF16-BE.txt` avec le même texte « Bonjour ! » mais en modifiant le format de codage (cf. menu Encodage, successivement en UTF-8 puis UTF16-BE).

Exercice n°9:

Commencez par démarrer un terminal sous Windows (Démarrer / et taper `cmd` dans la zone pour rechercher les programmes). Pour afficher successivement le contenu des deux fichiers dans le terminal utilisez la commande `type` (équivalent de `cat` sous Unix) :

```
type nom_de_fichier (avec successivement les fichiers texteANSI.txt, texteUTF8.txt puis texteUTF16-BE.txt)
```

Que remarquez-vous ?

Exercice n°10:

En comparant les fichiers `texteANSI.txt`, `texteUTF8.txt` et `texteUTF16-BE.txt`, à l'aide de Notepad++, que constatez-vous ?

En ouvrant les deux fichiers avec le logiciel Bloc-Notes que remarquez-vous ? Comment le logiciel peut connaître l'encodage utilisé par le fichier ?

Exercice n°11:

Voici par exemple une série de caractère codés en ASCII (ANSI) tels que stockés dans un fichier texte.

6C	65	73	20	6E	65	75	6E	65	75	73	20	66	6F	6E	74	20	71	75	65
20	63	6C	69	71	75	65	72												

Utilisez le tableau figurant en annexe pour trouver le texte correspondant.

Exercice n°12:

En UTF16-BE, le codage sera le suivant :

FE	FF	00	6C	00	65	00	73	00	20	00	6E	00	65	00	75	00	6E	00	65
00	75	00	73	00	20	00	66	00	6F	00	6E	00	74	00	20	00	71	00	75
00	65	00	20	00	63	00	6C	00	69	00	97	00	75	00	65	00	72		

Si l'éditeur de texte le traite comme un fichier avec des caractères codés en ASCII/ANSI, quel sera le texte affiché ?

Cela vous rappelle-t-il quelque chose ?

Exercice n°13:

A l'aide du logiciel EditHexa (<http://stephane.lavirotte.com/teach/cours/envinfor/EditHexa.zip>), inspectez le contenu des fichiers `texteANSI.txt`, `texteUTF8.txt` et `texteUTF16-BE.txt`. Vous devez retrouver les valeur d'encodage des caractères dans ces différents formats.

TD séance n° 10 Multimédia : Texte

Exercices complémentaires

Chaînes de traitements multimédia

Exercice n°1:

Décliner un exemple de chaîne de traitement d'objets multimédias, en vous inspirant en particulier du tableau sur les différents types de fichiers multimédia. Si vous connaissez des logiciels qui permettent de travailler sur une ou plusieurs étapes de la chaîne, sous Linux, sous Windows, mentionnez-les.

Exercice n°2:

Vers des domaines inexplorés... ou presque ... Clavier vers Haut-parleurs ? Image vers Vidéo ? ...

Maintenant que vous avez compris, décrivez une chaîne de traitement multimédia de votre invention. Quel traitement spécifique est nécessaire dans cette chaîne de traitement ?

Si l'on conçoit un objet image comme des cases grises (de 0% de noir à 100% de noir) posée sur un damier, et un objet son composé d'une séquence de couples (note, durée), décrivez un algorithme (et oui, c'est ça !) qui permettrait de transformer l'objet image vers un objet son.

Exercice n°3:

Illustrez cette transformation en traitant deux exemples simples :

- Une image toute grise (que des cases à 50% de noir)
- Une image « damier » alternant 2 cases noires et 2 cases blanches

Bases

Dans ces exercices, on passera directement d'une base à l'autre sans passer par la base 10. Pour vous faciliter la conversion, penser à regrouper les bits par 4... et oui, $15_{(10)} = 1111_{(2)} = F_{(16)}$!

Exercice n°4: Conversion du binaire vers hexadécimal

Donner l'écriture en base 16 des nombres suivants :

$$V = 101101_{(2)} \quad W = 101101011110_{(2)} \quad X = 100111001110111_{(2)}$$

Exercice n°5: Conversion de l'hexadécimal vers le binaire

Donner l'écriture en base 2 des nombres suivants :

$$Y = 24D_{(16)} \quad Z = FE_{(16)}$$

TD séance n° 10

Multimédia : Texte

Annexe

Décimal	Octal	Hex	Binaire	Caractère
0	0	0	00000000	NUL
1	1	1	00000001	SOH
2	2	2	00000010	STX
3	3	3	00000011	ETX
4	4	4	00000100	EOT
5	5	5	00000101	ENQ
6	6	6	00000110	ACK
7	7	7	00000111	BEL
8	10	8	00001000	BS
9	11	9	00001001	HT
10	12	A	00001010	LF
11	13	B	00001011	VT
12	14	C	00001100	FF
13	15	D	00001101	CR
14	16	E	00001110	SO
15	17	F	00001111	SI
16	20	10	00010000	DLE
17	21	11	00010001	DC1
18	22	12	00010010	DC2
19	23	13	00010011	DC3
20	24	14	00010100	DC4
21	25	15	00010101	NAK
22	26	16	00010110	SYN
23	27	17	00010111	ETB
24	30	18	00011000	CAN
25	31	19	00011001	EM
26	32	1A	00011010	SUB
27	33	1B	00011011	ESC
28	34	1C	00011100	FS
29	35	1D	00011101	GS
30	36	1E	00011110	RS
31	37	1F	00011111	US
32	40	20	00100000	SP
33	41	21	00100001	!
34	42	22	00100010	"
35	43	23	00100011	#
36	44	24	00100100	\$
37	45	25	00100101	%
38	46	26	00100110	&
39	47	27	00100111	'
40	50	28	00101000	(

Décimal	Octal	Hex	Binaire	Caractère
41	51	29	00101001)
42	52	2A	00101010	*
43	53	2B	00101011	+
44	54	2C	00101100	,
45	55	2D	00101101	-
46	56	2E	00101110	.
47	57	2F	00101111	/
48	60	30	00110000	0
49	61	31	00110001	1
50	62	32	00110010	2
51	63	33	00110011	3
52	64	34	00110100	4
53	65	35	00110101	5
54	66	36	00110110	6
55	67	37	00110111	7
56	70	38	00111000	8
57	71	39	00111001	9
58	72	3A	00111010	:
59	73	3B	00111011	;
60	74	3C	00111100	<
61	75	3D	00111101	=
62	76	3E	00111110	>
63	77	3F	00111111	?
64	100	40	01000000	@
65	101	41	01000001	A
66	102	42	01000010	B
67	103	43	01000011	C
68	104	44	01000100	D
69	105	45	01000101	E
70	106	46	01000110	F
71	107	47	01000111	G
72	110	48	01001000	H
73	111	49	01001001	I
74	112	4A	01001010	J
75	113	4B	01001011	K
76	114	4C	01001100	L
77	115	4D	01001101	M
78	116	4E	01001110	N
79	117	4F	01001111	O
80	120	50	01010000	P
81	121	51	01010001	Q

TD séance n° 10

Multimédia : Texte

Décimal	Octal	Hex	Binaire	Caractère
82	122	52	01010010	R
83	123	53	01010011	S
84	124	54	01010100	T
85	125	55	01010101	U
86	126	56	01010110	V
87	127	57	01010111	W
88	130	58	01011000	X
89	131	59	01011001	Y
90	132	5A	01011010	Z
91	133	5B	01011011	[
92	134	5C	01011100	
93	135	5D	01011101]
94	136	5E	01011110	^
95	137	5F	01011111	_
96	140	60	01100000	`
97	141	61	01100001	a
98	142	62	01100010	b
99	143	63	01100011	c
100	144	64	01100100	d
101	145	65	01100101	e
102	146	66	01100110	f
103	147	67	01100111	g
104	150	68	01101000	h
105	151	69	01101001	i
106	152	6A	01101010	j
107	153	6B	01101011	k
108	154	6C	01101100	l
109	155	6D	01101101	m
110	156	6E	01101110	n
111	157	6F	01101111	o
112	160	70	01110000	p
113	161	71	01110001	q
114	162	72	01110010	r
115	163	73	01110011	s
116	164	74	01110100	t
117	165	75	01110101	u
118	166	76	01110110	v
119	167	77	01110111	w
120	170	78	01111000	x
121	171	79	01111001	y
122	172	7A	01111010	z
123	173	7B	01111011	{

Décimal	Octal	Hex	Binaire	Caractère
124	174	7C	01111100	
125	175	7D	01111101	}
126	176	7E	01111110	~
127	177	7F	01111111	DEL