TNML302A&B

Encodage des caractères & itérateurs en shell

Patrick Paroubek

LIMSI-CNRS
Dépt. CHM - Groupe ILES
Rue John von Neumann, Campus Universitaire d'Orsay
Bât 508, 91405 Orsay cedex
pap@limsi.fr

mercredi 05 octobre 2016 / TNML302A&B - Cours 3

Produire la liste des bigrammes .

```
cat mon_corpus.txt | tr '\040' '\012' > mots.txt
echo "<BOF>" > /tmp/c1.txt
cat mon_corpus_mots.txt >> /tmp/c1.txt
cat mon_corpus_mots.txt > /tmp/c2.txt
echo "<EOF>" >> /tmp/c2.txt
paste /tmp/c1.txt /tmp/c2.txt > bigrammes.txt
```

- Dans un ordinateur, toute information est représenté en binaire avec un alphabet à 2 lettres (0 et 1) et les règles de l'arithmétique binaire (0+1=1+0=1, 0+0=0, 1+1=10).
- Les lettres sont appelées des bits,et sont groupés par 8, groupe appelé un octet (*byte*), ensuite on parle de mots mémoire de 16, 32, 64 et parfois 128 bits.
- Le sens de lecture (gauche/droite ou l'inverse varie d'un ordinateur à l'autre, ce sens est appelé "endianness" (2 valeurs : "big"=octet/byte de poids le plus fort d'abord, "little"=octet/byte de poids le plus faible d'abord)

Les unités de mesure de l'information sont :

- le bit,
- l'octet,
- le kilo octet Ko/Kb (1024 bits),
- le mega octet Mo/Mb (1024^2),
- le giga octet Go/Gb (1024^3),
- le tera octet To/Tb (1024^4),
- le peta octet Po/Pb (1024^5).

ASCII

- L'ASCII (American Standard Code for Information Interchange) a été crée en 1963 aux États-Unis, publié par l'USASI (United States of America Standards Institute) en 1968, avant d'être adopté ensuite comme norme interantionale (ISO-646).
- C'est un codage des caractères sur 7 bits, qui permet donc de coder 128 caractères différents.
- C'est le seul codage «universel», car il est inclus dans toutes les normes de codage utilisées en informatique.

Standards

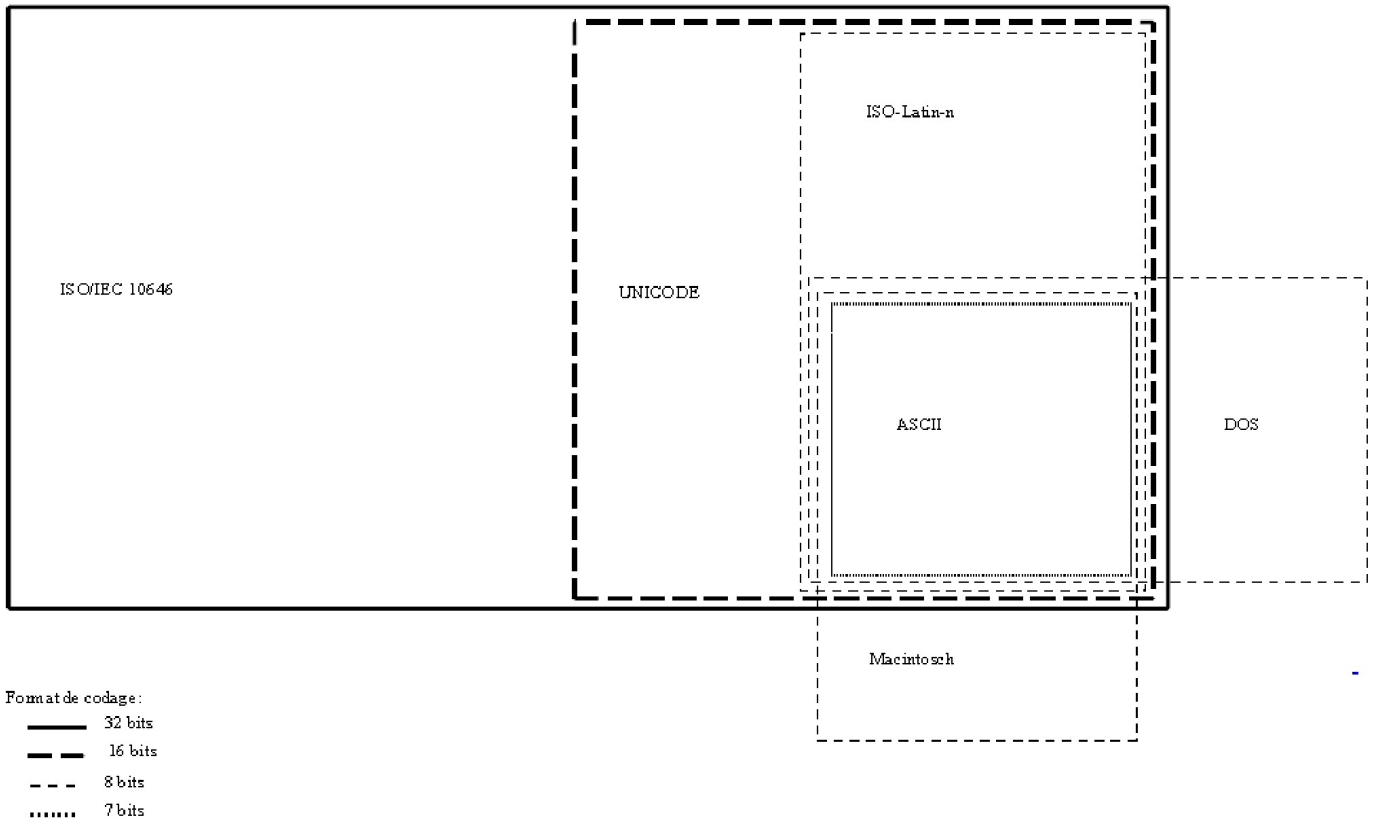

Fig. *Anthony Seiha Lim (TNML3 09-10)*

Avec la commande shell `man ascii` :

```
The following table contains the 128 ASCII characters.

C program '\X' escapes are noted.

Oct   Dec   Hex   Char                        Oct   Dec   Hex   Char
--------------------------------------------------------------------
000   0     00    NUL '\0'                    100   64    40    @
001   1     01    SOH (start of heading)      101   65    41    A
002   2     02    STX (start of text)         102   66    42    B
003   3     03    ETX (end of text)           103   67    43    C
004   4     04    EOT (end of transmission)   104   68    44    D
005   5     05    ENQ (enquiry)               105   69    45    E
006   6     06    ACK (acknowledge)           106   70    46    F
007   7     07    BEL '\a' (bell)             107   71    47    G
010   8     08    BS  '\b' (backspace)        110   72    48    H
011   9     09    HT  '\t' (horizontal tab)   111   73    49    I
012   10    0A    LF  '\n' (new line)         112   74    4A    J
013   11    0B    VT  '\v' (vertical tab)     113   75    4B    K
014   12    0C    FF  '\f' (form feed)        114   76    4C    L
```

015	13	0D	CR	'\r' (carriage ret)	115	77	4D	M	
016	14	0E	SO	(shift out)	116	78	4E	N	
017	15	0F	SI	(shift in)	117	79	4F	O	
020	16	10	DLE	(data link escape)	120	80	50	P	
021	17	11	DC1	(device control 1)	121	81	51	Q	
022	18	12	DC2	(device control 2)	122	82	52	R	
023	19	13	DC3	(device control 3)	123	83	53	S	
024	20	14	DC4	(device control 4)	124	84	54	T	
025	21	15	NAK	(negative ack.)	125	85	55	U	
026	22	16	SYN	(synchronous idle)	126	86	56	V	
027	23	17	ETB	(end of trans. blk)	127	87	57	W	
030	24	18	CAN	(cancel)	130	88	58	X	
031	25	19	EM	(end of medium)	131	89	59	Y	
032	26	1A	SUB	(substitute)	132	90	5A	Z	
033	27	1B	ESC	(escape)	133	91	5B	[	
034	28	1C	FS	(file separator)	134	92	5C	\	'\\'
035	29	1D	GS	(group separator)	135	93	5D	]	

036	30	1E	RS (record separator)	136	94	5E	^
037	31	1F	US (unit separator)	137	95	5F	_
040	32	20	SPACE	140	96	60	`
041	33	21	!	141	97	61	a
042	34	22	"	142	98	62	b
043	35	23	#	143	99	63	c
044	36	24	$	144	100	64	d
045	37	25	%	145	101	65	e
046	38	26	&	146	102	66	f
047	39	27	'	147	103	67	g
050	40	28	(	150	104	68	h
051	41	29	)	151	105	69	i
052	42	2A		152	106	6A	j
053	43	2B	+	153	107	6B	k
054	44	2C	,	154	108	6C	l
055	45	2D	-	155	109	6D	m
056	46	2E	.	156	110	6E	n

057	47	2F	/
060	48	30	0
061	49	31	1
062	50	32	2
063	51	33	3
064	52	34	4
065	53	35	5
066	54	36	6
067	55	37	7
070	56	38	8
071	57	39	9
072	58	3A	:
073	59	3B	;
074	60	3C	<
075	61	3D	=
076	62	3E	>
077	63	3F	?

157	111	6F	o
160	112	70	p
161	113	71	q
162	114	72	r
163	115	73	s
164	116	74	t
165	117	75	u
166	118	76	v
167	119	77	w
170	120	78	x
171	121	79	y
172	122	7A	z
173	123	7B	{
174	124	7C	|
175	125	7D	}
176	126	7E	~
177	127	7F	DEL

ISO Latin-1

Vinrent ensuite, des codes sur 8 bits (256 caractères) pour coder les lettres accentuées de beaucoup de langues de l'Europe de l'Ouest.

- ISO/CEI 8859-1 ou ISO Latin-1, qui contient 191 caractères (sauf les ligatures Œ, œ et Ÿ).
- ISO-8859-1 (attention au tiret supplémentaire), validé par l'Internet Assigned Numbers Authority, est un sur-ensemble de l'ISO/CEI 8859-1, attribue des caractères de contrôle aux valeurs 00 à 1F, 7F, et 80 à 9F, et dispose donc d'un code sur 256 positions.
- ISO-8859-15, plus récent, prend en charge le caractère euro et est mieux adapté au français.

UNICODE/ISO 10646

- Le Consortium Unicode (organisation à but non-lucratif), qui regroupe des grandes companies informatiques comme Adobe, Apple, IBM, Microsoft, Sun et Xerox, a pour but de définir un codage non-ambigu sur 16 bits (< v3.2),puis sur 20 bits (> v4), sans séquences de contrôle, ni de méthode de compactage.
- Les travaux d'UNICODE et ISO 10646 sont synchronisés.
- Le sous-ensemble sur 16 bits de UCS s'appelle le BMP (Basic Multilingual Plan). La norme le définissant a été publiée en 1993 sous le nom de ISO 10646-1. UCS assigne à chaque caractère un code et un nom.

UNICODE/ISO 10646

- Le standard international ISO 10646 définit le jeu de caractères international, Universal Character Set (UCS). C'est un super-ensemble de tous les autres jeux de caractères standard (reversibilité sans perte d'information).
- Se décline en plusieurs formats d'encodage
  - UTF8 (de taille variable sur 1, 2 3, ou 4 octets),
  - UTF16 (sur 2 octets, ou 4 octets),
  - UTF32 (sur 4 octets).

UNICODE/ISO 10646

Avantage : permet de coder quasiment n'importe quelle écriture/langue (principe de "localisation", c.a.d paramétrage culturel).

Inconvénient : nécessite des bibliothèques adaptées au type de base des objets manipulés par les primitives d'entrée sortie et les chaînes de caractères, car le type de base n'est plus l'octet.

Langage de programmation le plus adapté actuellement : JAVA (UTF16).

Télécharger une page WEB en français et la sauvegarder au format html. Construire ensuite le vecteur distribution de caractères de la page avec la suite de commandes :

```
cat f.htm | sed -e 's/./&\n/g' | sort | uniq -c | sort -k 1,1 -n
```

Faire de même avec un page dans un autre langue possédant des caractères latin et comparer la lettre qui apparaît le plus fréquemment entre les deux langues (en français c'est le 'e').

Nous allons observer les différences de représentation informatique d'un caractère accentué, par exemple le « é ».

1. Créer avec un éditeur de texte (e.g. `gedit`, `kate` etc.) un fichier (`ex1.txt`) encodé en UTF-8 contenant uniquement les 3 caractères : « aéa ».
2. Observer la taille du fichier en octets avec la commande shell : `wc -c ex1.txt` le résultat retourné est : `5 ex1.txt`.
3. Vérifier le contenu du fichier avec `cat ex1.txt`

4. Transcoder le fichier avec par ex. `recode` ou `iconv` de UTF-8 vers ISO-8859-1 (latin-1).
   - `recode utf8..latin1 ex1.txt` , ATTENTION cette commande produit un « effet de bord », elle change le contenu du fichier, il faut faire attention à ne pas l'appliquer 2 fois de suite.
   - ou autre possibilité :
     ```
     iconv -f UTF-8 -t ISO-8859-1 ex1.txt  > ex1_lat1.txt
     ```
     notez qu'`iconv` ne produit pas d'effet de bord.
5. Vérifier le contenu du fichier avec `cat ex1.txt` (si conversion avec recode) ou `cat ex1_lat1.txt` (si conversion avec iconv), le contenu a changé, dans le terminal on voit comme résultat : aña
6. et la command `wc -c` produit maintenant : `4 ex1_lat1.txt`.