Encodage de caractères

Julien Baste

IUT de Lille

Séance 02

2025/2026

Langage écrit

- Une langue est *souvent* écrite à partir de mots composés de *lettres*
- Les lettres sont définies dans un *alphabet*
  - liste de représentations graphiques (*glyphe*, *dessin*)
  - plusieurs casses possibles (capitale ou minuscule)
- Les phrases sont organisées grâce à la *ponctuation*
- lettres, ponctuations et chiffres sont représentés par des dessins
- Écrire du texte :
  - utiliser un sens d'écriture
  - notion de lignes et de position
  - déplacer le *stylo*

Langage écrit

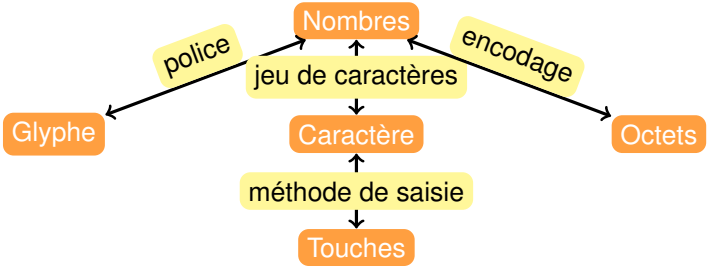
- Une langue est *souvent* écrite à partir de mots composés de *lettres*
- Les lettres sont définies dans un *alphabet*
  - liste de représentations graphiques (*glyphe*, *dessin*)
  - plusieurs casses possibles (capitale ou minuscule)
- Les phrases sont organisées grâce à la *ponctuation*
- lettres, ponctuations et chiffres sont représentés par des dessins
- Écrire du texte :
  - utiliser un sens d'écriture
  - notion de lignes et de position
  - déplacer le *stylo*

Écrire $\Leftrightarrow$ Dessiner

Manipuler du texte $\Rightarrow$ Coder ces symboles (caractères)

Du texte au(x) glyphe(s)

- Les écrits sous forme d'images ne sont pas exploitables ;
- On ne retient que les *caractères* les uns à la suite des autres ($\neq$ *lettres*) ;



- Les glyphes sont les dessins des lettres, différents selon les polices
- Les polices supportent souvent plusieurs jeux de caractères. Le dessin n'y est stocké qu'une fois.

Différence de glyphes

La lettre $\mathcal{A}$ et A représentent le même caractère, mais pas le même que *A* (α capitale).
De même le *a* de *tata*, de tɑtɑ, TATA ou tata sont les mêmes caractères.

Qu'est-ce qu'un caractère ?

- Au début : lettres, chiffres, ponctuation simplifiée.
- Correspondait grossièrement à une touche de machine à écrire (+Capitale/Minuscule)
- Au fur et à mesure, de très nombreux caractères ont été rajoutés.
- Jeu de caractères universel : Unicode.

Quelques caractères dont vous ne connaissez peut-être pas les noms

#	♯	&	|	/
mot-croisillon	dièse	et	ou, *païpe*	slash
@	\	_	[ ]	{ }
arobasse	backslash	underscore	crochets	accolades

Jeux de caractères

- Plusieurs jeux de caractères primitifs sur 7 ou 8 bits par caractère.
- Un seul a vraiment survécu : `US-ASCII`
  - 7 bits pour coder les caractères
  - 1 bit pour contrôler la parité
- Création de jeux de caractères nationaux
  - Normes ISO-8859-*
    - caractères 0 à 127 = US-ASCII
    - caractères 128 à 255 = caractères locaux
  - Autres méthodes :
    - KOI-8R (russe)
    - JIS (Japonais)
    - BIG5 (Chinois)
    - collections de caractères
- Universalisation : Unicode → plus de 1 000 000 caractères.

Le codage US-ASCII

American Standard Code for Information Interchange

- Code standardisé par l'ANSI

  American National Standards Institute

- Créé pour représenter du texte anglais
- Utilise 7 bits
- Inclut
  - lettres latines
  - chiffres arabes
  - caractères de ponctuation
  - caractères de contrôle

Le codage US-ASCII

- 32 caractères de *contrôle*,
- 96 caractères *affichables* ;
- Unicode et ISO-8859 sont compatibles avec ASCII.

00	NUL	10	DLE	20	SP	30	0	40	@	50	P	60	`	70	p
01	SOH	11	DC1	21	!	31	1	41	A	51	Q	61	a	71	q
02	STX	12	DC2	22	"	32	2	42	B	52	R	62	b	72	r
03	ETX	13	DC3	23	#	33	3	43	C	53	S	63	c	73	s
04	EOT	14	DC4	24	$	34	4	44	D	54	T	64	d	74	t
05	ENQ	15	NAK	25	%	35	5	45	E	55	U	65	e	75	u
06	ACK	16	SYN	26	&	36	6	46	F	56	V	66	f	76	v
07	BEL	17	ETB	27	'	37	7	47	G	57	W	67	g	77	w
08	BS	18	CAN	28	(	38	8	48	H	58	X	68	h	78	x
09	HT	19	EM	29	)	39	9	49	I	59	Y	69	i	79	y
0A	LF	1A	SUB	2A	*	3A	:	4A	J	5A	Z	6A	j	7A	z
0B	VT	1B	ESC	2B	+	3B	;	4B	K	5B	[	6B	k	7B	{
0C	NP	1C	FS	2C	,	3C	<	4C	L	5C	\	6C	l	7C	|
0D	CR	1D	GS	2D	-	3D	=	4D	M	5D	]	6D	m	7D	}
0E	SO	1E	RS	2E	.	3E	>	4E	N	5E	^	6E	n	7E	~
0F	SI	1F	US	2F	/	3F	?	4F	O	5F	_	6F	o	7F	DEL

Limite du codage sur un seul octet

Coder un caractère sur un seul octet :

- Un octet = 8 bits = 256 valeurs possibles
- ASCII sur 8 bits avait un bit *inutilisé*.
- Langues asiatiques : pas suffisant.
- Codage à décalage : certaines séquences (non rencontrées habituellement) permettent de changer de « zone » de caractères.
- Certaines séquences déclenchent du codage où 1 caractère est codé par 2 octets.
- Rupture de l'égalité 1 octet = 1 caractère
- Autres codages : BIG5 est un codage à 2 octets par caractères pour le chinois.

La table Unicode

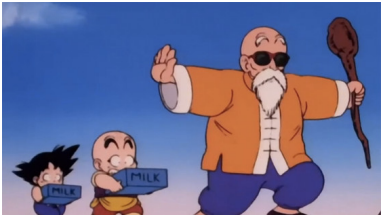
- Unicode est une collection de plus de 1 000 000 caractères (jusqu'à 17×2^{16}).
- Les positions Unicode s'écrivent de la forme
  - `U+XXXX` (avec 4 chiffres hexadécimaux) ou
  - `U+XXXXXX` (avec 6 chiffres hexadécimaux).
- Le premier élément est à la position `U+0000`.
- Le dernier élément est à la position `U+10FFFF`.
- On peut retrouver cette table par exemple ici :
  https://symbl.cc/fr/unicode-table/

Problème : La table Unicode est grande !

- Il ne fallait que 7 bits (donc 1 octet) pour un caractère ASCII
- Il faut 21 bits (donc 3 octets) pour encoder un caractère Unicode
  - On consomme trois fois plus de place pour stocker un même texte !

La table Unicode : Un exemple

Après un rude entraînement :



Vous avez le droit de porter le symbole de votre entraîneur :

Ce symbole est dans la table Unicode en position : U+4E80

L'encodage UTF-8

L'encodage UTF-8 permet de transformer une position Unicode en une séquence d'octets

Valeurs	Position Unicode en binaire (bits utiles)	Codage UTF-8 en binaire	octets
U+0000 à U+007F	abc defg (07)	0abc defg	1
U+0080 à U+ 07FF	abc defg hijk (11)	110a bcde 10fg hijk	2
U+0800 à U+FFFF	abcd efgh ijkl mnop (16)	1110 abcd 10ef ghij 10kl mnop	3
U+10000 à U+1FFFFF	a bcde fghi jklm nopq rstu (21)	1111 0abc 10de fghi 10jk lmno 10pq rstu	4

Pour encoder une position Unicode en UTF-8 :

1. Considérer la ligne correspondant à votre position Unicode
2. Convertir votre position Unicode en binaire
3. Garder les 7, 11, 16 ou 21 derniers bits (dépendant de la ligne utilisée)
4. Identifier chacun des bits conservés avec les lettres de la colonne <Position Unicode en binaire>
   (chaque lettre correspondant à exactement 1 bit)
5. Écrivez votre code UTF-8 sur 1, 2, 3, ou 4 octets (dépendant de la ligne utilisée)
   en suivant la forme donnée par la colonne <Codage UTF-8 en binaire>

Décodage de l'UTF-8

Valeurs	Position Unicode en binaire (nombre de bits)	Codage UTF-8 en binaire	octets
U+0000 à U+007F	abc defg (07)	0abc defg	1
U+0080 à U+ 07FF	abc defg hijk (11)	110a bcde 10fg hijk	2
U+0800 à U+FFFF	abcd efgh ijkl mnop (16)	1110 abcd 10ef ghij 10kl mnop	3
U+10000 à U+1FFFFF	a bcde fghi jklm nopq rstu (21)	1111 0abc 10de fghi 10jk lmno 10pq rstu	4

Pour obtenir une position Unicode à partir d'une séquence d'octets encodés en UTF-8 :

1. Écrire la séquence d'octets en binaire.
2. Regarder le nombre de `1` à la suite parmi les bits les plus à gauche

Nombre de `1`	0	2	3	4
Encodage sur	1 octet	2 octets	3 octets	4 octets

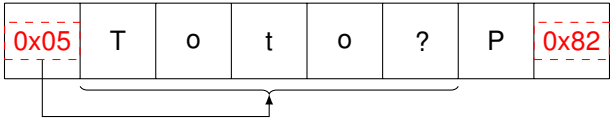
3. Considérer le nombre d'octets trouvé précédemment
4. Identifier chaque bit avec les chiffres ou les lettres de la colonne <Codage UTF-8 en binaire> (chaque lettre correspondant à exactement 1 bit, les chiffres doivent correspondre)
5. Écrivez votre position Unicode en binaire en suivant la forme donnée par la colonne <Position Unicode en binaire>
6. Convertissez en hexadécimal pour avoir une position Unicode de la forme `U+XXXX` ou `U+XXXXXX`.

Les chaînes avec longueur spécifiée

- Chaînes de caractères = listes ordonnées de caractères
- En mémoire elle occupe plusieurs positions consécutives
- On désigne souvent la chaîne par la première position occupée

- **Idée : stocker la longueur de la chaîne de caractères**

Exemple : Stockage de la chaîne « Toto ? »



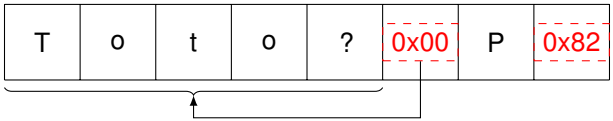
(Les éléments « P » et « 0x82 » sont dans la mémoire, mais n'appartiennent pas à la chaîne).

Les chaînes avec marqueurs de fin

Autre technique : Un caractère spécial en fin de chaîne.

- Le langage C utilise le caractère nul.

Exemple : Stockage de la chaîne « Toto ? »



(Les éléments « P » et « 0x82 » sont dans la mémoire, mais n'appartiennent pas à la chaîne).

Est-ce que le marqueur fait partie de la chaîne ?

En pratique, oui. Mais il ne fait pas partie du texte codé par la chaîne.
À l'intérieur d'un langage, il n'y a en général qu'une seule sorte de chaîne.

C’est tout pour aujourd’hui