Introduction à l'informatique

Les textes

Jean-Christophe Dubacq

IUT de Villetaneuse

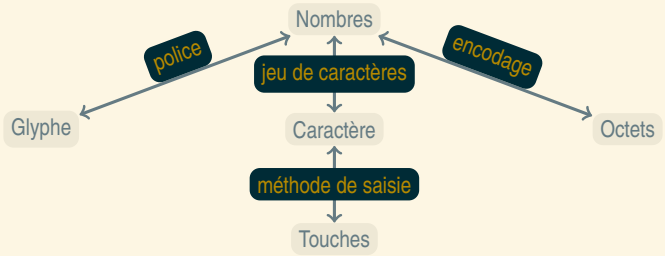
S1 2016

Les textes

De l'écrit au binaire
Jeux de caractères et codages
Les chaînes de caractères

Du texte au(x) glyphe(s)

- Les écrits sous forme d'images ne sont pas exploitables ;
- L'écriture est donc simplifiée pour ne retenir que les *caractères* les uns à la suite des autres (≠ *lettres*) ;



- Les glyphes sont les dessins des lettres, différents selon les polices

Du caractère au glyphe : la police

- Les polices supportent souvent plusieurs jeux de caractères. Le dessin n'y est stocké qu'une fois.
- ⛔ Une même police peut comporter plusieurs glyphes pour le même caractère (formes décoratives)
- Une police comporte une partie programme pour sélectionner le dessin le mieux adapté

Différence de glyphes

La lettre $\mathcal{A}$ et A représentent le même caractère mais pas le même que A.
De même le *a* de *Abba*, de Abba ou Abba ou Abba sont les mêmes caractères.

Ligatures esthétiques ou linguistiques

La lettre Œ (ligature linguistique) est différente de OE. La lettre fi représente deux caractères, avec affichage fi (ligature esthétique pour éviter le fi).
En arabe ou sanskrit, la ligature est obligatoire mais esthétique : تونس contre ت و ن س.
La ligature esthétique apparaît au niveau des polices, la ligature linguistique au niveau des caractères.

Qu'est-ce qu'un caractère ?

- Au début : lettres, chiffres, ponctuation simplifiée.

Correspondait grossièrement à une touche de machine à écrire (+Majuscule/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

C	Usuel	Français	Anglais
#	dièse	croisillon, octothorpe	hash, number sign
&	et	esperluette, et commercial	ampersand, and
|	ou, *païpe*	barre verticale	pipe
/	slash	barre oblique	slash
@	*arobasse*	arobase	at, at sign
\	backslash	contre-oblique	backslash
_	underscore	(blanc) souligné	underscore
[ ]	crochets	crochets	(square) brackets
{ }	accolades	accolades	(curly) braces

Jeux de caractères

- Plusieurs jeux de caractères primitifs sur 7 ou 8 bits par caractère.
- Un seul a vraiment survécu : ASCII
- Création de jeux de caractères nationaux
- Normes ISO-8859-* : caractères 0 à 127 = ASCII ; caractères 128 à 255 = caractères locaux
- Autres méthodes : KOI-8R (russe), JIS (Japonais), BIG5 (Chinois)... collections de caractères
- Universalisation : Unicode : plus de 100 000 caractères.

Certains caractères sont dupliqués pour des raisons historiques

Exercices

La table ASCII

Trouvez dans la table ASCII :

1. Le caractère de code 0x41
2. Le caractère de code 0x30
3. Le caractère *a* et *A*. Comparez l'écriture binaire des codes numériques correspondants.
4. Le caractère de code 0x20. Quel est-il ?
5. Le caractère *retour chariot* (son nom est NEWLINE ou NL).

Comment passe-t-on d'une lettre à la suivante ? D'une majuscule à une minuscule ?

Au début était le *byte*

- Premiers codages : un caractère = un *byte* = 6 à 8 bits
- Rapidement *byte* = octet = 8 bits. 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.

Le Mojibake

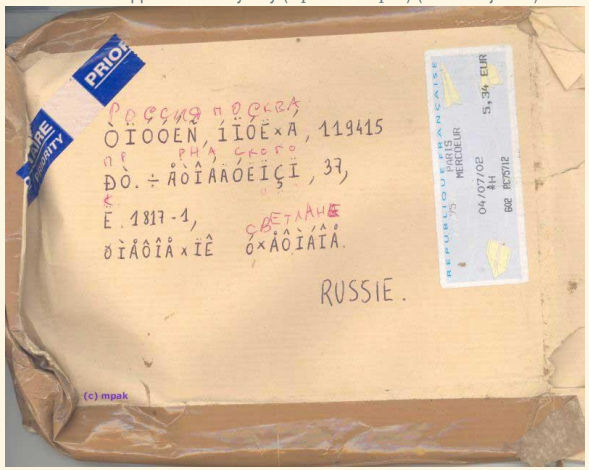
L'enveloppe était envoyée à un étudiant russe par une amie française qui a recopié son adresse reçue par e-mail. Le logiciel ne savait pas lire les caractères cyrilliques (page de code KOI8-R) et les a remplacés par les caractères du code ISO-8859-1.

En KOI8-R :
Россия Москва, 119415
пр.Вернадского, 37,
к.1817-1,
Плетневой Светлане

En ISO-8859-1 :
òÏÓÓÉÑ ÍÏÓ×Á, 119415
ÐÒ.÷ÅÒÎÁÄÓËÉÊ, 37,
Ë.1817-1,
ðÌÅÔÎÅ×ÏÊ ó×ÅÔÌÁÎÅ

Le postier a réussi à faire la transformation inverse ! (en rouge)

Une enveloppe en krakozyabry (кракозябры) (aussi Mojibake).



Exercices

Codage nationaux et Mojibake

Soit le texte : Coefficient marée trop fort pour livraison tomates cœur-de-bœuf

1. Identifiez dans ce texte les ligatures linguistiques et les ligatures esthétiques
2. Est-il possible de représenter ce texte dans le jeu de caractères ASCII ?
3. Dans le jeu de caractère ISO-8859-15 (dit *latin-9*), il est possible de coder ce texte. Chaque caractère est alors codé par un octet unique. Quelle est la taille du fichier qui contient uniquement ce texte ?
4. Un polonais lit sur son vieil ordinateur le texte précédent. Il voit qu'une des lettres a été remplacée par ˝ (c'est un double accent aigu, comme dans Erdős, et pas un tréma comme dans Gwenaël). Laquelle et pourquoi ? S'il renvoie le texte tel quel a son correspondant français du début, que verra le français et pourquoi ?

Unicode et UTF-8

- Unicode est une collection de plus de 100 000 caractères qui ne spécifie pas la façon de le représenter par une séquence d'octets. La taille maximale est de 17×2^{16} et le code maximal 0x10FFFF
- UTF-8 est une façon de transformer un numéro en une séquence d'octets

Valeurs	Écriture binaire	Codage UTF-8 (binaire)	octets
0x0–0x7F	abc defg	0abc defg	1
0x80–0x7FF	abc defg hijk	110a bcde 10fg hijk	2
0x800–0xFFFF	abcd efgh ijkl mnop	1110 abcd 10ef ghij 10kl mnop	3
0x10000–0x1FFFFF	a bcde fghi jklm nopq rstu	1111 0abc 10de fghi 10jk lmno 10pq rstu	4

- UCS-2 est un codage partiel sur 2 octets par caractères (représente les 2^{16} premiers caractères)
- UTF-16 est un codage plus simple qu'UTF-8 utilisant 2 ou 4 octets par caractères : 2 pour les premiers, 4 pour les autres (10 octets par paire de 2 octets).

Avantage de l'UTF-8 : économe en place pour l'ASCII (1 octet par caractère)

Inconvénient de l'UTF-8 : impossible de dire facilement à quel octet est le n° caractère.

Exercices

UTF8

1. Le caractère de numéro 0x0041 (A) est codé par quel(s) octet(s) en UTF-8 ?
2. Le caractère de numéro 0x00E9 (é) est codé par quel(s) octet(s) en UTF-8 ?
3. Le caractère de numéro 0x0F03 () est codé par quel(s) octet(s) en UTF-8 ?
4. Le caractère de numéro 0x12084 () est codé par quel(s) octet(s) en UTF-8 ?
5. Dans un fichier codé en UTF-8, on trouve les six octets suivants. Combien de caractères sont réellements codés dans ce texte ?

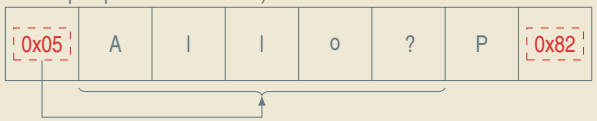
   `0xE6 0x9D 0x8c 0xDE 0xBC 0x43`
6. L'anglais n'utilise que des caractères dont le numéro est dans la première ligne, et est codé traditionnellement en ISO-8859-1 (1 caractère = 1 octet). Le français utilise 5% de caractères de la deuxième ligne (le reste de la première), et est codé pareil (1 caractère = 1 octet). L'arabe (le russe, l'hébreu, le grec) sont aussi codés traditionnellement par 1 caractère = 1 octet, et comportent 95% de caractères de la deuxième ligne (le reste de la première ligne). Le chinois, en revanche est traditionnellement codé en BIG5 (1 caractère = 2 octets). Les textes chinois sont à 99% des caractères de la troisième ligne (le reste de la première ligne).
   Pour un texte de 1000 caractères codé en UTF-8, combien d'octets seront utilisés en moyenne pour un texte anglais, français, russe et chinois ?
7. Quel est en chinois l'augmentation de la taille du texte par rapport au codage traditionnel ?

Les chaînes avec longueur spécifiée

Les chaînes de caractères sont des listes ordonnées de caractères.
Lorsqu'une chaîne de caractères est stockée en mémoire, elle occupe plusieurs positions consécutives dans la mémoire. On désigne souvent la chaîne par la première position occupée.
Certains langages résolvent le problème de savoir où la chaîne s'arrête en stockant aussi la longueur.
Problème avec certains codages/jeux de caractères pour trouver le n^{e} élément d'une chaîne (et en particulier, la longueur en nombre de caractères).
Avantage : le calcul de la place mémoire occupée est instantané.

Exemple

On stocke ici la chaîne « Allo ? » (le P et la valeur 0x82 sont des éléments qui sont dans la mémoire mais ne font pas partie de la chaîne).



Est-ce que la longueur est en caractères ou en octets ?

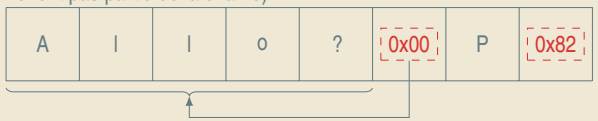
En **octets**, le plus souvent, ou les deux. le plus important est de savoir trouver la fin de la chaîne (pour pouvoir la copier).

Les chaînes avec marqueurs de fin

Une autre possibilité est de marquer la fin de la chaîne avec un octet particulier ou une séquence d'octets particulière. C'est le cas du langage C (et de beaucoup d'autres langages dérivés) qui utilise le caractère nul.

Exemple

On stocke ici la chaîne « Allo ? » (le P et la valeur 0x82 sont des éléments qui sont dans la mémoire mais ne font pas partie de 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.

Le problème de l'échappement

La fin de chaîne

Quand la longueur n'est pas spécifiée à côté d'une chaîne, la fin de la chaîne est forcément indiquée par une séquence spécifique de bits.

- S'il existe une séquence spécifique invalide dans le codage pour la représentation de caractères, alors on peut la choisir comme représentant la fin de chaîne.
- Sinon, il faut choisir un caractère qui va coder la fin de la chaîne

Comment coder une chaîne qui comporte ce caractère ?

Les séquences signifiantes

Parfois, on veut pouvoir utiliser dans des chaînes des séquences qui ont un sens spécial. Par exemple, on pourrait vouloir que `0x0F03` représente le caractère 3e qu'on ne peut pas rentrer facilement au clavier. Mais dans ce cas, comment écrire la chaîne `0x0F03` (comme par exemple pour la phrase « Si on met `0x0F03` dans une chaîne on obtient le caractère 3e » ?

Il faut donc utiliser une procédure d'échappement !

Les séquences d'échappement

- On utilise une séquence (parfois codante) d'échappement qui permet de modifier le sens des caractères qui suivent
- Si la séquence d'échappement est codante, on doit prévoir au moins une combinaison qui permet de redonner le caractère d'échappement
- Avoir des chaînes interprétables complique énormément les opérations élémentaires, comme calculer le nombre de caractères dans la chaîne, ou savoir si un caractère est présent dans la chaîne.

On se retrouve souvent à « empiler » les modes d'échappement identiques ou différents.

Exemple

En langage C et dérivés, le caractère \ est utilisé pour introduire des séquences d'échappement.
`\0` est le caractère nul.
`\n` est le caractère 10 (NEWLINE).
`\t` est le caractère 9 (TAB).
`\xxx` est le caractère de numéro octal xxx.
`\Uxxxx` est le caractère unicode de numéro hexadécimal xxxx. Ce caractère unicode peut représenter plusieurs octets. Le codage choisi dépend du compilateur et du type de la chaîne.

Exercices

Les échappements en C

Dessinez quelle est la structure en mémoire des chaînes C suivantes ? Comment sont elles affichés ?

1. `"Toto"`
2. `"Bonjour le monde\n"`
3. `"Acheter:\n\tponey\n\tporte-avions\n"`
4. `"\303\251\n"`
5. `"\U20AC"` (symbole euro)
6. `"\0"`

Une bizarrerie historique du C/C++ fait que certaines séquences sont remplacées avant compilation par d'autres caractères :

Trigraphe	??(	??)	??<	??>	??=	??/	??'	??!	??-
Remplacement	[	]	{	}	#	\	^	|	~

7. `"Hello??!"`
8. `"Bye??/n"`

La table ASCII

- 32 caractères de « contrôle », 96 « affichables » ;
- Unicode, ISO-8859 compatibles avec ASCII.

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

Retour