Architecture des ordinateurs

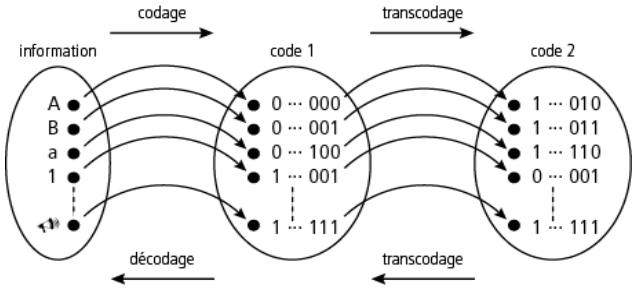
4 - Représentation de l'information en machine
Définitions de base
Les codes alphanumériques

Philippe Darche
IUT Paris Descartes

Le problème

- Pour écrire l'anglais, 26 lettres, 10 chiffres et quelques caractères de ponctuation sont suffisants
- Pour les langue européennes comme le français, il y a en plus d'autres caractères ou variations sur les caractères
- Il existe d'autres alphabets ou certains pays utilisent des idéogrammes
- Comment coder ces symboles en machine ?

La notion de code

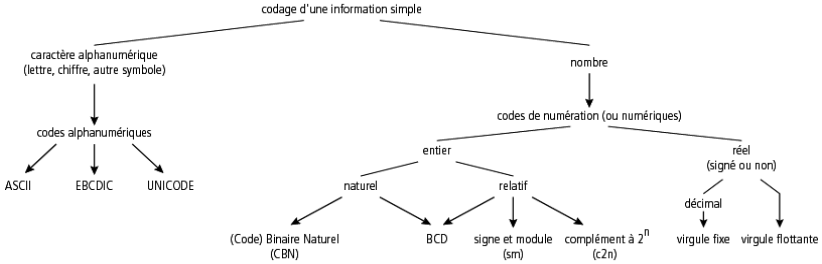


Définitions

- Code = {mot de code au format n}
  - nombre de mots-code possible = 2^n
  - rendement $\eta = \frac{Card_C}{2^n}$
- Codage = loi de correspondance arbitraire
- Opération inverse : le décodage
- Types de code :
  - alphanumérique
  - de numération ou numérique
  - instruction
  - de canal
  - détecteur (et correcteur) d'erreur(s)
  - de chiffrement
  - etc.

Représentation de l'information en machine

- Codage des deux principales familles avec codes usités

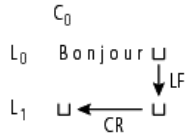


Code alphanumérique

- Caractères alphanumériques
  - contraction des mots *alphabétique* et *numérique*
  - ⇒ caractère alphabétique, numérique ou, éventuellement, caractère spécial (ponctuation, technique (monnaie, etc.) ou caractère espace (*space*)
    (définition de la norme ISO/IEC 2382-5)
- Codage des caractères alphanumériques (*graphic character*)
  - En anglais, *character set*
  - loi de correspondance biunivoque
  - et éventuellement des codes de contrôle de périphérique (*control function*)
    - retour chariot (*Carriage Return* ou CR), saut à la ligne (*Line Feed* ou LF), etc.

Utilisation d'un code de contrôle

- Pour gérer un périphérique
  - écran + clavier = un terminal
  - imprimante
- Exemple : le retour à la ligne
  - retour chariot puis saut de ligne ou inversement



Remarque importante

- Les codes de contrôle ne s'affichent pas.
- Ils sont interprétés par le périphérique.

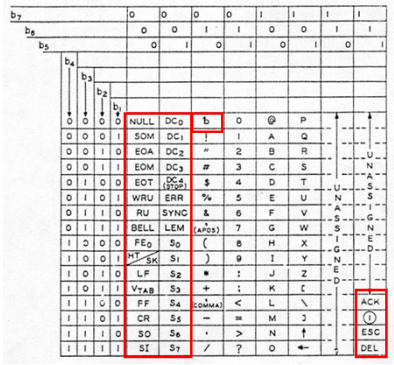
Ne pas confondre

- Caractère graphique
  - information à coder
  - exemple : lettre capitale latine a
- Symbole graphique = forme graphique
  - représentation visuelle d'un caractère graphique ou d'une fonction de contrôle
  - autre appellation : glyphe
- Police de caractères = {symboles graphiques}

Le code ASCII d'origine

- *American Standard Code for Information Interchange*
- Inventé par l'américain Bob Bemer [Bemer 61]
- Code au format n = 7 bits
  - si octet, un bit de parité pour le contrôle d'erreur
- Pas de minuscule
- Codage des fonctions de contrôle

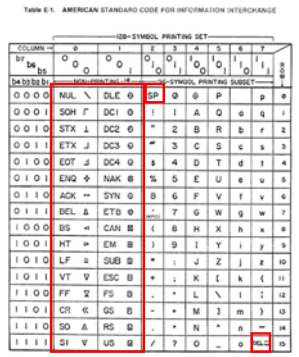
La table originale standard ASA X3.4-1963



contrôle

Évolution de la table originale ASCII

- Prise en compte des minuscules



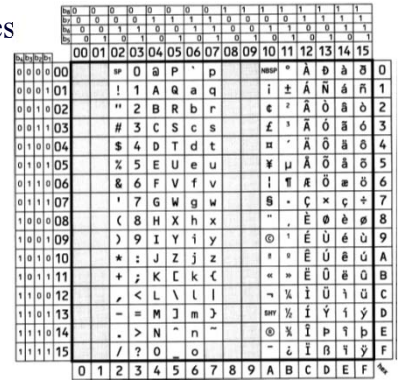
contrôle

La norme ISO 8859-x

- Le code des micro-ordinateurs
- Code ASCII complété pour les valeurs 128 à 255
  - format n = 8 bits
  - prise en compte des besoins des langues européennes
  - ⇒ nécessité de passer au format octet
  - normalisé par l'*International Organization for Standardization* (ISO)
  - ensemble latin 1 : ISO 8859-1:1998
    - prise en compte des caractères accentués d'europe occidentale
  - version avec signe euro : ISO 8859-15

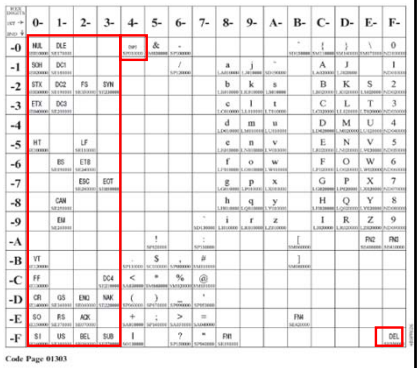
La norme ISO/CEI 8859-1

- 191 caractères graphiques codés
- Version avec codes de contrôle
  - ISO-8859-1



L'EBCDIC

- Origine : IBM
- format natif n = 8 bits



contrôle

Limitation des codes alphanumériques classiques

- Format limité à n = 7 ou 8 bits
  - nombre de caractères codables limité (2^n)
  - ⇒ impossibilité de prise en compte de toutes les lettres, signes de ponctuation et autres symboles d'une langue
- Si prise en compte de quelques langues
  - ⇒ problème pour l'échange mondiale d'informations non résolu
  - ⇒ transcodage nécessaire mais souvent impossible

La réponse aux limitations

- UNICODE pour *Universal Code*
- Codage de tous les caractères de toutes les langues
- Association pour chaque caractère abstrait d'un numéro de code (valeur scalaire, *code point*) et d'un nom standard
  - comme pour les codes précédents
  - n = 16 bits à l'origine, 21 bits actuellement
  - identique à la norme ISO/CEI 10646-1:1993
- Et représentation par un format d'encodage (*Unicode Transformation Format*)
  - UTF-32 (32 bits), UTF-16 (16 bits) et UTF-8 (8 bits)
    - UTF-8 le plus efficace en terme de stockage

Vocabulaire d'UNICODE

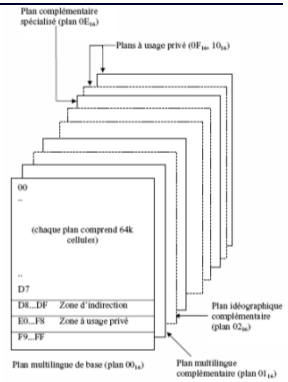
- Caractère : forme abstraite d'un composant atomique d'un langage
  - lettre, signe de ponctuation, etc.
  - exemple : U+0041 *latin capital letter a*
- Glyphe : forme graphique d'un caractère
  - police : ensemble de glyphes
  - de l'exemple précédent : A, 𝐀, 𝖠, etc.

UNICODE

- Conformité avec le standard ISO/IEC 10646 pour l'affectation du numéro de code et du nom standard
  - UCS-2 et UCS-4 (*Universal Character Set*)

Plans et zones de codage

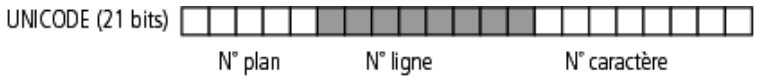
- Plans de 64 K cellules
  - plan de base (PMB pour *Basic Multilingual Plane*)
    - caractères usuels alphabétiques



[Andries 02]

Espace d'adressage

- 1 114 112 caractères possibles (*code point*)



Schémas d'encodage

- UTF-x (x = 8, 16 et 32)
  - *Unicode or UCS Transformation Format*
- Objectifs : adapter la valeur du code au format courant des données (format n = 8, 16 ou 32 bits) en optimisant l'encombrement

Schéma d'encodage UTF-32

- Correspondance directe (remplissage avec 0)

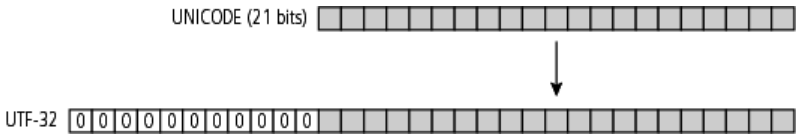


Schéma d'encodage UTF-16

- Valeur tronquée à 16 bits pour les caractères ASCII sinon séquence de deux mots de 16 bits
- Optimisé pour les caractères du plan multilangue de base (BMP)
- Tableau de codage dans le TD

Schéma d'encodage UTF-16

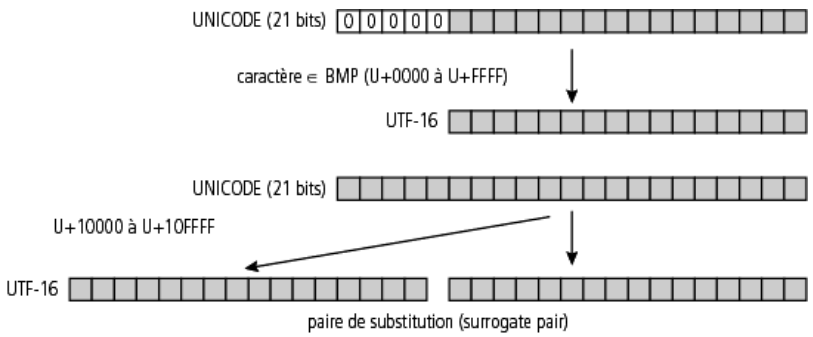
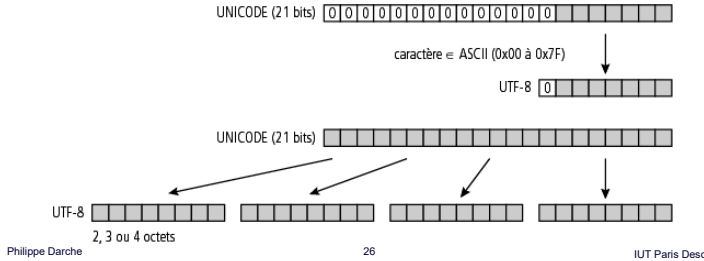


Schéma d'encodage UTF-8

- Facilité d'usage avec les systèmes basés sur l'ASCII
- Format variable
- Tableau de codage dans le TD



Schémas d'encodage

- Attention, ce n'est pas qu'un simple découpage

Table 3-5. UTF-16 Bit Distribution

Scalar Value	UTF-16
xxxxxxxxxxxxxxxx	xxxxxxxxxxxxxxxx
000uuuuuxxxxxxxxxxxxxxxx	110110wwwwxxxxxx 110111xxxxxxxxxx

Note: wwww = uuuuu - 1

Table 3-6. UTF-8 Bit Distribution

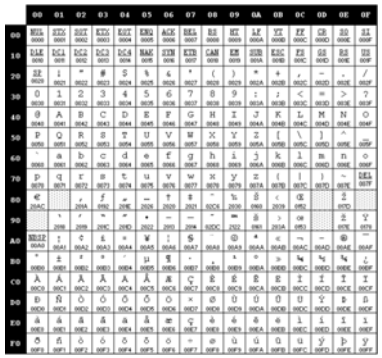
Scalar Value	First Byte	Second Byte	Third Byte	Fourth Byte
00000000 0xxxxxxx	0xxxxxxx			
00000yyy yyxxxxxx	110yyyyy	10xxxxxx		
zzzzyyyy yyxxxxxx	1110zzzz	10yyyyyy	10xxxxxx	
000uuuuu zzzzyyyy yyxxxxxx	11110uuu	10uuzzzz	10yyyyyy	10xxxxxx

Schémas de sérialisation

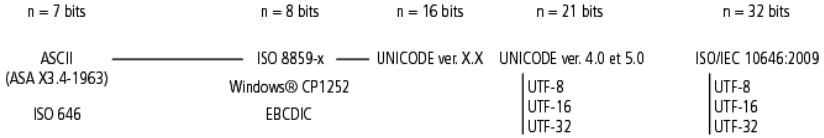
- Pour UTF-32 et UTF-16 :
  - forme *Little-Endian* (LE) : UTF-16LE et UTF-32LE
  - ⇒ LSB en premier
  - forme *Big-Endian* (BE) : UTF-16BE et UTF-32BE
  - ⇒ MSB en premier
- Version *endian-ness*
  - *Byte Order Mark* (BOM)

D'autres codes alphanumériques

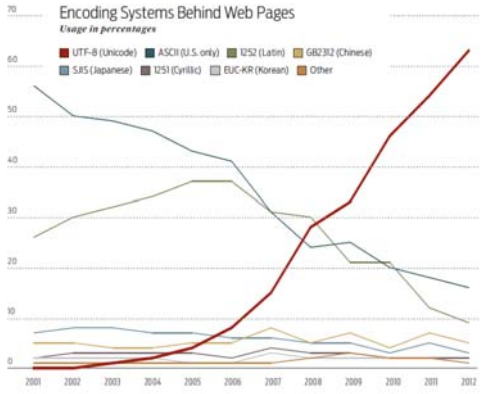
- Code Page 850 de l'IBM PC (latin 1, Europe occidentale)
  - codes 128 à 255 en complément de l'ASCII
- Windows®-1252 → (latin 1, Europe occidentale)
- Le code Hollerith
  - origine : IBM
  - pour la carte perforée
- Le jeu de caractères vidéotex
- Etc.



Résumé des codes courants



And the winner is …



[King 12]

Conclusion : le chemin des données (*datapath*)