❄ Chapitre 9 ❄

# Représentation d'un texte en machine

## I. Encodage ?

Lors des premières transmissions d'information (télégraphe optique de Chappe en 1794, télégraphe électrique de Cooke et Wheatstone en 1938, ...), il a été nécessaire de définir un code pour représenter les différents symboles utilisés dans le langage courant (lettres, chiffres, ponctuations, ...). Ce code peut être visuel (drapeaux, tour Chappe), codé à l'aide de plusieurs impulsions (code Morse), ... C'est ce que l'on appelle **l'encodage**.

Un ordinateur utilisant un système binaire pour stocker, transmettre et utiliser les données, il a fallu choisir un moyen de représenter les différentes lettres et symboles à l'aide de 0 et de 1. Un des premiers systèmes d'encodage binaire (sur 5 bits) est du au français Emile Baudot en 1874.

## II. Quelle représentation choisir ?

Au début de l'automatisation des recensements (en 1890, machines fonctionnant à l'aide de cartes perforées) chaque machine/constructeur possédait son propre système d'encodage. Certains étaient liés à la structure matérielle du lecteur de cartes perforées par exemple (lecture purement mécanique) et il y avait incompatibilité entre les deux plus grands constructeurs de l'époque (Bull et IBM).

Cette prolifération d'encodages possède plusieurs inconvénients :

- transfert de données difficile entre utilisateurs,
- changement d'appareil compliqué,
- obligé de redévelopper les programmes à chaque changement de gamme,
- ...

Le besoin pour un encodage standardisé se fait ressentir.

### 1. A.S.C.I.I.

En 1963, après un travail du Department of Defense des Etats-Unis d'Amérique, une première norme apparaît, c'est la naissance de l'encodage A.S.C.I.I. (American Standard Code for Information Interchange).
Elle définit un standard pour coder les caractères en binaire.

**Nombres de bits utilisés :** 7 soit $2^7 = 128$ symboles différents.

**Symboles codables :** chiffres de 0 à 9 (indo-arabes), alphabet latin (majuscules et minuscules mais sans caractères spéciaux ou accentués), la ponctuation et caractères non imprimables (par exemple retour à la ligne).

**Avantages :** prend peu de place en machine (7 bits par caractère) et est un standard supporté par quasiment toutes les machines/logiciels.

**Inconvénients :**

- N'est pas adapté aux langues autres que l'anglais.
- Seulement $2^7$ possibilités.

### 2. ISO 8859-15

La norme ISO-8859−15 (aussi appelée Latin-9) est une norme européenne qui apparaît en 1998. Elle concerne les pays suivants : Canada, Allemagne, Royaume-Uni, Danemark, Espagne, Finlande, France, Italie, Pays-Bas, Norvège, Portugal et Suède (parmi d'autres pays européens).

**Nombres de bits utilisés :** 8 soit $2^8 = 256$ symboles différents.

**Symboles codables :** tous ceux de la norme ASCII (en ajoutant un 0 devant les 7 bits du codage en ASCII) + tous les caractères spéciaux propres aux pays concernés (notamment les accents, le ç, ...) et le symbole €.

Il existe d'autres normes ISO-8859−? selon les pays concernés (8859−9 : turque, 8859−10 : pays scandinaves, 8859−8 : hébreu, ...).

**Avantages :**

- bien adapté aux pays concernés,
- standard qui est compatible avec la norme ASCII.

**Inconvénients :**

- besoin d'autres normes pour d'autres pays (Japon, Chine, pays Arabes, ...),
- pas forcément adapté aux moyens de communications modernes (smileys, ...).

## 3. UTF-8

L'encodage UTF-8 (Unicode Transformation Format) apparaît avec la norme ISO/CEI 10646 de 1993. Il est depuis devenu entièrement compatible avec la norme Unicode (qui a pour but depuis 1991 d'offrir un encodage commun au monde entier, pour cela la norme Unicode nécessite $2^{21}$ nombres).

**Nombres de bits utilisés :** entre 1 et 4 octets mais « uniquement » $2^{21} = 2\,097\,152$ symboles différents.

**Symboles codables :** à peu près tous ceux dont on a eu besoin un jour!

**Avantages :**

- universel : contient les symboles nécessaires pour la plupart des langues existantes,
- compatible avec la norme ASCII,
- compatible avec les symboles autres que les lettres : smileys, notes de musique, ...

**Inconvénients :** prend plus de place que le code ASCII (logique pour coder davantage de choses) et est plus difficile à encoder/décoder (nombre de bits variable).

C'est maintenant l'encodage le plus répandu sur internet (utilisé par environ 95% des sites internets en 2019 : W3techs).

**Sources et liens :** Unicode, Wikipédia et ISO

**Le défi du cours :** Rechercher le principe du codage UTF-8 sur internet puis décoder le texte ci-dessous.

| | | | | | | |
|---|---|---|---|---|---|---|
| 0101 0110 | 0110 1111 | 0110 1001 | 0110 1100 | 1100 0011 | 1010 0000 | 0010 0000 |
| 0011 0001 | 1110 0010 | 1000 0010 | 1010 1100 | 0010 1100 | 0010 0000 | 0111 0101 |
| 0110 1110 | 0010 0000 | 1111 0000 | 1001 1111 | 1001 1000 | 1000 0100 | 0010 0000 |
| 0110 0101 | 0111 0100 | 0010 0000 | 0110 1101 | 1100 0011 | 1010 1010 | 0110 1101 |
| 0110 0101 | 0010 0000 | 1111 0000 | 1001 1101 | 1000 0100 | 1001 1110 | 0010 0001 |