L'encodage des caractères en informatique

Site Internet : **www.gecif.net**	Type de document : **Cours**	Intercalaire :	Date :

En informatique un ordinateur manipule seulement des 0 et des 1. Afin de faire le lien entre un caractère (symbole graphique affiché à l'écran) et son code (nombre binaire) il faut utiliser un **jeu de caractères**. Il n'existe en fait que 3 types de jeux de caractères : la table ASCII de base, les tables ASCII étendues, et l'Unicode.

I – La table ASCII de base

Le premier encodage historique est l'**ASCII**, soit l'***A**merican **S**tandard **C**ode for **I**nformation **I**nterchange* (en français, le *code américain normalisé pour l'échange d'informations*). C'est une norme américaine, inventée en 1961, qui avait pour but d'organiser le bazar informatique à l'échelle nationale. Ce n'est pas le premier encodage utilisé mais on peut oublier les précédents.

La norme **ASCII** établit une correspondance entre une représentation binaire des caractères de l'alphabet latin et les symboles, les signes, qui constituent cet alphabet. Par exemple, le caractère "a" est associé à "01100001" et "A" à "01000001".La norme **ASCII** permet ainsi à toutes sortes de machines de stocker, analyser et communiquer de l'information textuelle. La quasi totalité des ordinateurs et des stations de travail utilisent l'encodage **ASCII**.

La table ASCII de base (il n'y en a qu'une, appelé aussi US-ASCII), ne contient que 128 caractères non accentués, sans symboles graphiques et avec peu de caractères spéciaux, mais reste compatible avec tous les systèmes, tous les logiciels, tous les pays du monde, et toutes les époques. Elle a été conçue à l'origine pour échanger du texte en anglais. L'ensemble des langages de programmation utilisent encore aujourd'hui seulement des caractères de la table ASCII de base pour écrire le code source d'un programme (syntaxe et mots clé), tant qu'on ne manipule pas des chaînes de caractères accentuées. Sa limite est l'absence de caractères spéciaux ou accentués.

II – Les tables ASCII étendues

Le code ASCII a été mis au point à l'origine pour la langue anglaise, il ne contient donc pas de caractères accentués, ni de caractères spécifiques à une langue. Pour coder ce type de caractère il faut recourir à un autre code. Dans les années 1980 le code ASCII a donc été étendu à 8 bits (un octet) pour pouvoir coder plus de caractères (on parle d'ailleurs de code ASCII étendu…). Ce code attribue les valeurs 0 à 255 (donc codées sur 8 bits, soit 1 octet) aux lettres majuscules et minuscules, aux chiffres, aux marques de ponctuation et aux autres symboles (caractères accentués dans le cas du code *iso-latin1*).

Les pages de code à 256 caractères (il y en a plusieurs, appelées aussi tables ASCII étendues), ont été définies à l'origine localement pour être utiliser seulement dans une zone géographique précise en fonction des caractères particuliers à afficher dans la langue locale (lettres ou ponctuation). Les pages de code à 256 caractères sont parfois appelées à tort "table ASCII étendue", ce qui donne l'impression d'une normalisation, alors que justement il n'y a rien de normalisé et qu'il existe plusieurs dizaines de pages de code différentes sur l'ensemble de la planète. Exemple de pages de code pour l'Europe occidentale : la page de code 850 de MS-DOS, la norme ISO 8859-1, la page de code Windows-1252 de Microsoft, la page de code MacRoman d'Apple, etc. Le point commun de toutes ces pages de code est que les 128 premiers caractères (de 0 à 127) sont ceux de la table ASCII de base. Une page de code est donc la table US-ASCII à qui on a ajouté 128 nouveaux caractères, codés entre 128 et 255.

Les deux jeux de caractères ASCII étendus les plus couramment utilisés sont :

- Le code ASCII étendu OEM, c'est-à-dire celui qui équipait les premières machines de type IBM PC
- Le code ASCII étendu ANSI (appelée **ISO 8859-1**), utilisé par les systèmes d'exploitation récents

La norme **ISO 8859-1** est également appelée ASCII étendu **ANSI**, **Latin-1** ou **Europe occidentale**, code 191 caractères de l'alphabet latin, et est utilisée par Linux, Windows, et de nombreux documents et pages web.

En 1998 la norme **ISO 8859-15** (***latin-9*** ou **« *Occidental (euro)* »**) introduit le signe de l'euro (€) et complète le support de quelques langues dont le français (avec Œ) en abandonnant des symboles peu utilisés (dont le mystérieux ¤ signifiant « monnaie »). Elle est néanmoins peu utilisée par rapport à la norme **ISO 8859-1** et à l'arrivée de l'Unicode.

Les langues latines s'en sont plutôt bien sorties finalement avec ces tables ASCII étendues à 256 caractères. Elles ont réussi à ne pas dépasser la limite fatidique de l'octet, ce qui restait quand même le plus pratique pour les traitements (et la consommation mémoire). Mais les langues asiatiques comme le japonais, le coréen ou le chinois disposent de bien trop de caractères pour que tout tienne sur 8 bits. Les encodages mis au point en Asie de l'Est ont donc franchi le saut du multi-octet. Certains utilisaient 2 octets, ce qui permet 65 536 (2^{16}) codes différents.

La limite des pages de code est qu'elles ne sont pas adaptées à l'échange mondial d'informations dans différentes langues car le code ASCII étendu n'est pas unique et dépend fortement de la plateforme utilisée.

III – L'Unicode

Le code *Unicode* est un système de codage des caractères à l'origine sur 16 bits mis au point en 1991. Le système Unicode permet de représenter n'importe quel caractère par un code sur 16 bits ou plus, indépendamment de tout système d'exploitation ou langage de programmation. Il regroupe ainsi la quasi-totalité des alphabets existants et connus à ce jour (arabe, arménien, cyrillique, grec, hébreu, latin, ...) et est compatible avec le code ASCII.

L'Unicode est un jeu de caractères contenant plus de 100 000 de caractères (et pouvant à terme en coder plus d'un million) possédant chacun un nom et un code numérique unique (appelé **point de code**), définitif et partagé par l'ensemble des systèmes, des logiciels, des pays du monde entier et désormais pour toujours. Unicode constitue la 3ème et dernière méthode d'encodage des caractères (après la table ASCII de base et après les différentes pages de code qui ont montré leurs limites), et se veut **universel et définitif** : il n'y aura jamais un autre système d'encodage des caractères qui le remplacera et Unicode a été pensé afin de solutionner tous les problèmes rencontrés avec les systèmes d'encodage précédents. Son avantage : possibilité d'échanger un texte dans le monde entier sans problème d'interprétation à l'arrivée, possibilité d'écrire à la fois en Grec, en Chinois, en Japonais et en Russe dans le même fichier, et possibilité d'accéder à plusieurs centaines de nouveaux symboles graphiques ou mathématiques non existant dans les pages de code à 256 caractères. UTF-8 est l'algorithme d'encodage permettant de sauvegarder ou de transférer un caractère Unicode sur 1, 2, 3 ou 4 octets selon le caractère. La particularité d'UTF-8 est que les 128 caractères de base de la table ASCII de base se codent toujours sur 1 seul octet, de même valeur, en ASCII comme en UTF-8. Un fichier texte basique, codé seulement en US-ASCII (code source d'un programme ou fichier HTML par exemple), reste donc parfaitement lisible avec un éditeur ne lisant que de l'UTF-8. Les caractères accentués quant à eux se codent sur 2 octets en UTF-8, mais cette fois le codage est absolu, définitif, partagé par tous, et ne dépend pas d'une page de code locale. A noter que dans une page web la séquence HTML **&#nnn;** permet d'envoyer au navigateur client le caractère Unicode numéro **nnn** sans forcément utiliser l'algorithme d'encodage UTF-8 et sans forcément encoder le fichier HTML en UTF-8 (il peut rester en ISO-8859-1 et même en US-ASCII). L'utilisation du jeu de caractères Unicode sur une page web n'est donc pas systématiquement liée à l'encodage UTF-8.

L'UTF-8 est un des algorithmes permettant d'encoder (sur 1 à 4 octets) dans un fichier texte n'importe quel caractère du jeu de caractères Unicode contenant plus de 100 000 symboles. La table Unicode est identique à l'ISO-8859-1 pour les caractères 0x80 à 0xFF sauf qu'ils sont codés sur 2 octets.

Unicode, dans la théorie c'est très bien, mais dans la pratique c'est une autre paire de manches : généralement en Unicode, un caractère prend **2 octets**. Autrement dit, le moindre texte prend **deux fois plus de place qu'en ASCII** : c'est du gaspillage. De plus, si on prend un texte en français, la grande majorité des caractères utilisent seulement le code ASCII (sur 1 octet). Seuls quelques rares caractères nécessitent l'Unicode (sur 2 octet). On a donc trouvé une astuce : l'**UTF-8**. Un texte en UTF-8 est simple: il est partout en ASCII, et dès qu'on a besoin d'un caractère appartenant à l'Unicode, on utilise un caractère spécial signalant "*attention, le caractère suivant est en Unicode*".

Par exemple, pour le texte "**Bienvenue chez Sébastien !**", seul le "**é**" ne fait pas partie de la table ASCII de base. On écrit donc en UTF-8 : `Bienvenue chez SÃ©bastien !`

Pour être rigoureux, on indique quand même au début du fichier que c'est un fichier en UTF-8 à l'aide de caractères spéciaux : `ï»¿Bienvenue chez SÃ©bastien !`

L'UTF-8 rassemble le meilleur de deux mondes : l'efficacité de l'ASCII et l'étendue de l'Unicode. D'ailleurs l'UTF-8 a été adopté comme norme pour l'encodage des fichiers XML. La plupart des navigateurs récents supportent également l'UTF-8 et le détectent automatiquement dans les pages HTML.

IV – Conclusion

En résumé, la table ASCII de base et l'Unicode sont parfaitement clairs, bien définis, normalisés, non ambigus et compris par tout le monde. En revanche les pages de code (ou tables ASCII étendues), qui étaient adaptées dans les années 1980 et 1990 lorsque l'information restait locale sans se déplacer, est à l'origine des problèmes d'encodage et de décodage de caractères accentués et spéciaux en informatique, car chacun utilise sa propre page de code : chaque pays, chaque système, chaque logiciel. L'arrivée de nouveaux symboles comme le symbole monétaire de l'Euro a accentué le problème et l'incompatibilité entre les fichiers texte puisque chacun a rajouté dans la précipitation le symbole Euro dans sa page de code, mais sans se concerter : le symbole Euro avait un code différent selon la table utilisée. Les pages de code ne sont donc plus adaptées à l'usage que l'on fait de l'information aujourd'hui, avec un partage mondial grâce au réseau de communication et d'échange d'informations Internet : l'avenir c'est l'Unicode !