Secteur Tertiaire Informatique
Filière « Etude et développement »

Séquence « Développer des pages Web »

Introduction

Apprentissage

Mise en pratique

Evaluation

TABLE DES MATIERES

1. ENCODAGE, JEUX DE CARACTERES.

La table ASCII

Le processeur d'un ordinateur, c'est-à-dire le composant qui s'occupe de traiter les informations, exécuter les programmes, etc. …, ne comprend que les instructions formulées en **binaire** : il ne peut que lire une suite d'éléments pouvant être dans deux états : 0 ou 1.

Donc, tous les programmes informatiques que vous pourriez écrire, toutes les instructions SQL, tous les fichiers, sont **in fine** traduits en code machine, donc une longue suite de 0 et de 1, pour pouvoir être lus ou exécutés. Chacun de ces 0 ou 1 est un **bit**. Ces bits sont regroupés par **huit** pour former un **octet**.

L'ordinateur ne connaît donc pas la notion de caractères. Il ne sait pas ce qu'est un « A », ou un « é ». Il a donc fallu créer une **table de conversion** pour traduire les caractères de la langue courante en une série de bits. La table **ASCII** était née !

La table **ASCII** est donc une table de conversion, qui permet de traduire en **code binaire** 128 caractères, dont 33 caractères de contrôle (séparateur de fichier, saut de page, …) et 95 caractères affichables. Les caractères affichables sont les 26 lettres de l'alphabet, en majuscules et en minuscules, les 10 chiffres arabes et toute une série de caractères spéciaux courants (#, ;,), <, …).

De combien de bits avons-nous besoin pour stocker ces 128 caractères ?

Chaque **bit** peut prendre **deux valeurs différentes**. Donc, avec 1 bit, je peux représenter 2 caractères. Avec 2 bits, je représente 2^2, donc 4 caractères.

Et $128=2^7$. J'ai donc besoin de 7 bits pour pouvoir représenter toute ma table ASCII.

L'ordinateur travaillant sur des octets, on code donc ta table **ASCII** sur un octet, avec le 8^e bit à 0. « 00110100 » représente par exemple le caractère « 4 », « 01001101 » le caractère « M » (le premier bit étant celui de droite, le 8^e celui de gauche).

Jeux de caractères

On s'est ensuite rendu compte que ces 128 caractères n'étaient pas suffisants. En effet, ils ne comprennent pas les caractères accentués. Ils ne comprennent pas non plus les caractères cyrilliques, japonais, …

On a alors commencé à utiliser le huitième bit. Ce qui permettait de représenter 128 caractères supplémentaires (donc $2^8 = 256$ en tout). C'est l'UNICODE.

Mais avec 128 caractères supplémentaires, on n'a pas de quoi représenter tous les caractères qui n'existent pas dans la table **ASCII**. On a donc créé plusieurs jeux de caractères différents. En voici quelques exemples :

- **L'ISO 8859-1** (ou **latin1**) : qui permet de couvrir une bonne partie des langues d'Europe occidentale en ajoutant les lettres accentuées aux caractères ASCII de base.
- L'ISO 8859-7 : qui permet de représenter les lettres grecques.
- L'ISO 8859-11 : qui contient une bonne partie des glyphes de la langue thaï.
- L'ISO 8859-15 : qui est une révision de l'ISO 8859-1, et qui remplace quelques caractères peu utilisés par d'autres, plus nécessaires, comme l'euro (« € »)

Dès lors, lorsque l'on crée un fichier, un programme ou autre, il faut préciser quel jeu de caractère (ou encodage) est utilisé.

L'UTF-8

Il reste un petit problème : comment faire des documents (ou autres) qui doivent utiliser plusieurs jeux de caractères différents ? On a donc créé un nouveau type d'encodage, permettant de représenter les caractères avec **2 octets** au lieu d'**1**.

16 bits donnent $2^{16} = 65536$ possibilités. On peut donc, en utilisant 2 octets, représenter plus de 65000 caractères.

Cet encodage, s'appelle **l'UTF-8**. Le désavantage d'un tel encodage est bien entendu le **coût en mémoire**, deux octets prenant plus de place qu'un seul.

Cependant, tous les caractères ne sont pas codés sur deux octets. S'il s'agit d'un caractère de base de la table **ASCII**, le 8^e bit est à 0, ce qui indique qu'il n'est codé que sur un octet.

Par contre, lorsque le 8^e bit est à 1, cela indique qu'on a affaire à un caractère spécial et qu'il est codé sur deux octets. Donc, en **UTF-8**, un « é » prendra plus de place qu'un « e ».

CREDITS

ŒUVRE COLLECTIVE DE l'AFPA
Sous le pilotage de la DIIP et du centre d'ingénierie sectoriel Tertiaire-Services

Equipe de conception (IF, formateur, mediatiseur)
Formateur : Alexandre RESTOUEIX

Date de mise à jour : 29/01/19