

# Unicode demystified

Stéphane Bortzmeyer

<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 27 décembre 2002

<https://www.bortzmeyer.org/unicode-demystified.html>

Auteur(s) : Richard Gillam

ISBN n°

Éditeur : Addison-Wesley

Publié en

---

Pour tous ceux qui sont désormais :-) convaincus de l'utilité d'Unicode (parce qu'ils travaillent dans un environnement non-anglo-saxon ou simplement parce qu'ils sont sensibles à la diversité culturelle du monde), voici un livre qui évitera de se cogner le texte de la norme. Celle-ci n'est pas incompréhensible (un peu plus austère qu'un RFC mais **beaucoup** moins que les normes ITU/ISO) mais c'est une norme, elle n'a pas de vocation pédagogique.

Le livre de Gillam, au contraire, se veut une introduction pratique et claire à Unicode, et notamment aux problèmes pratiques que pose son déploiement. Il cible les programmeurs d'abord, les ingénieurs système et réseaux ensuite.

Il explique très bien les concepts de base d'Unicode (si différents des autres jeux de caractère que beaucoup de bêtises ont été écrites par des informaticiens n'ayant rien compris à Unicode) et il a notamment une utilisation originale du modèle en couches appliqué aux jeux de caractères (Unicode a cinq couches).

Le chapitre historique est passionnant, avec plein de détails sur les techniques d'encodage (un encodage n'est pas juste un choix arbitraire de bits pour représenter une information, il a des avantages et des inconvénients).

Les sujets très difficiles de la canonicalisation ("masse" est-il l'équivalent de "ma[Caractère Unicode non montré <sup>1</sup> je"?) ou de la bidirectionnalité ("Avram said 'Mazel Tov', and smiled", avec la phrase du milieu en hébreu) sont très bien traités, très clairs.

---

1. Car trop difficile à faire afficher par L<sup>A</sup>T<sub>E</sub>X

Une grande (peut-être trop grande, mais c'est la taille d'Unicode qui est en cause) partie est consacrée à l'examen de beaucoup d'écritures, avec leurs particularités (il n'existe pas une écriture sans au moins une idiosyncrasie gênante à implémenter, pas une langue qui ne rajoute ses propres variations; vous connaissiez le "French accent sorting"?). Un voyage passionnant, et plein de jolis dessins.

Les derniers chapitres, consacrés à l'implémentation, raviront ceux qui se souviennent avec émotion du cours de Structures de Données à la fac. Plein de solutions rigolotes et détaillées à des problèmes spécifiques à Unicode (la plupart des opérations nécessitent une table, contrairement à ASCII, et il n'est pas évident de gérer une table de  $2^{21}$  entrées). On y retrouve même le « trie » des routeurs IP, qui sert aussi, par exemple, aux conversions de l'alphabet latin tapé au clavier en cyrillique (car il y a la meme règle du "longest match").

La plupart des codes sont en Java mais aucun gadget objet n'est utilisé et ces exemples sont facilement compréhensibles.

Quelques critiques : le chapitre 1 est bâclé (confondant Unicode et UTF-16) et contredit le texte ultérieur (qui explique bien la différence). Il y a des erreurs manifestes concernant le monde Internet qu'il connaît évidemment peu (le livre d'Andries <<https://www.bortzmeyer.org/unicode-en-pratique.html>> est bien meilleur sur ce point). Et la dernière partie sur le support d'Unicode dans les langages de programmation et dans les systèmes d'exploitation n'est qu'une synthèse d'informations officielles, dans un domaine où tout système prétend être Unicode et où le diable est dans les détails (quel niveau de support Unicode, exactement?)