UTF-8, un format de transformation pour Unicode et ISO 10646

Network Working Group
Request For Comments : 2044
Catégorie : Informatif

Auteur : F. Yergeau – Alis Technologies – Octobre 1996
Traducteur : N. Schaefer – sN Informatique – Août 2005

Statut de ce document

Ce document apporte des informations à la communauté Internet. En aucun cas, ce document ne spécifie pas un standard. La diffusion de ce document est libre.

Résumé

Les standards Unicode, version 1.1 et ISO/IEC 10646-1 :1993 définissent tout deux une table de caractères sur 16 bits qui convient à la plupart des systèmes d'écriture au monde. Les caractères codés sur 16 bits ne sont toutefois pas compatibles avec de nombreux protocoles et applications. Cette situation a conduit au développement de plusieurs formats de transformation UCS (UCS Transformation Format, UTF), chacun possédant ses propres caractéristiques. UTF-8 a la caractéristique de préserver la table ASCII américaine : les caractères ASCII américains sont codés sur un octet en utilisant la valeur ASCII traditionnelle, et chaque octet qui possède une telle valeur ne peut être qu'un caractère ASCII américain. Cela apporte une compatibilité entre les systèmes de fichiers, les analyseurs et les autres applications qui se basent sur les valeurs ASCII américaines mais qui ne considèrent pas les autres valeurs.

1. Introduction

Les standards Unicode, version 1.1 [UNICODE], et ISO/IEC 10646-1 :1993 [ISO-10646] définissent tout deux une table de caractères sur 16 bits, UCS-2, qui convient à la plupart des systèmes d'écriture du monde. L'ISO-10646 défini également une table de caractères sur 31 bits, UCS-4, qui n'assigne aucune valeur au-delà de la région couverte par UCS-2 (the Basic Multilingual Plane, BMP). Cependant, les codages UCS-2 et UCS-4 sont difficiles à utiliser dans de nombreuses applications et protocoles qui travaillent avec des caractères codés sur 8 ou même 7 bits. Même certains systèmes plus récents capables de travailler avec des caractères sur 16 bits ne peuvent pas traiter les données codées en UCS-4. Cette situation a conduit au développement de formats de transformation UCS (UCS Transformation Formats, UTF), chacun possédant ses propres caractéristiques.

UTF-1 a uniquement un intérêt historique, et a été retiré de l'ISO 10646. UTF-7 a l'avantage de coder l'ensemble de la table Unicode en utilisant uniquement des octets dont le bit de poids fort est supprimé (valeurs ASCII américaines codées sur 7 bits [US-ASCII]), ce qui assure un codage sûr des emails ([RFC1642]). UTF-8 utilise tous les bits d'un octet mais a l'avantage de préserver le table ASCII américaine : les caractères ASCII américains sont codés sur un octet qui possède la valeur ASCII normale, et chaque octet de cette valeur peut uniquement être interprété comme un caractère ASCII et rien d'autre.

UFT-16 est une méthode permettant de transformer un sous-ensemble de la table UCS-4 en une paire de valeurs UCS-2 d'une plage réservée. UTF-16 impacte donc UTF-8 car les valeurs UCS-2 de la plage réservée doivent être traitées de manière particulière dans la transformation UTF-8.

UTF-8 code les caractères UCS-2 ou UCS-4 sur un nombre variable d'octets, où le nombre d'octets, et la valeur de chacun de ces octets, dépendent de la valeur entière assignée au caractère avec ISO 10646. Ce format de transformation possède les caractéristiques suivantes (toutes les valeurs sont en hexadécimal) :

- Les caractères de valeurs allant de 0000 0000 à 0000 007F (valeurs ASCII américaines) correspondent aux octets 00 à 7F (valeurs ASCII américaines sur 7 bits).
- Dans tous les autres cas, les valeurs ASCII américaines n'apparaissent pas dans un flux de caractères codés en UTF-8. Cela apporte une compatibilité entre les systèmes de fichiers ou les autres applications qui se basent sur les valeurs ASCII américaines mais qui ne considèrent pas les autres valeurs.
- La conversion dans les deux sens entre UTF-8 et UCS-4, UCS-2 ou Unicode est facile.
- Le premier octet d'une séquence d'octets indique le nombre d'octets dans la séquence.
- Les délimitations des caractères sont facilement repérables dans un flux d'octets.
- Le tri lexicographique des chaînes de caractères UCS-4 est conservé. Bien sûr, cela présente un intérêt limité dans la mesure où le tri n'est, dans aucun des deux cas, culturellement valide.
- Les octets de valeurs FE et FF n'apparaissent jamais.

A l'origine, UTF-8 était un projet du XOJIG (X/Open Joint Internationalization Group) dont l'objectif était de réaliser un système de fichiers sécurisé basé sur UTF (File System Safe UCS Transformation Format, FSS-UTF) compatible avec les systèmes UNIX, et supportant les textes multilingues avec le même codage. Les premiers auteurs sont Gary Miller, Greger Leijonhufvud et John Entenmann. Plus tard, Ken Thompson et Rob Pike réalisèrent un travail significatif sur UTF-8.

Une description peut également être consultée dans le rapport technique Unicode #4 (Unicode Technical Report #4) [UNICODE]. La référence définitive, contenant les dispositions pour les données UTF-16 avec UTF-8, est l'annexe de l'ISO/IEC 10646-1 [ISO-10646].

2. Définition d'UTF-8

Avec UTF-8, les caractères sont codés en utilisant des séquences contenant jusqu'à 6 octets. L'octet d'une séquence composée d'un seul octet a son premier bit à 0, les 7 bits restants sont utilisés pour coder le caractère. Dans une séquence de n octets (n>1), les n premiers octets sont placés à 1 et suivis d'un bit à 0. Les bits restants contiennent une partie de la valeur du caractère à coder. Tous les octets suivants ont leur premier bit à 1 et le deuxième à 0, laissant les 6 derniers bits pour la valeur du caractère à coder.

Le tableau suivant résume le format des différents types d'octets. La lettre x indique les bits disponibles pour le codage des bits du caractère UCS-4.

Plages UCS-4 (hexadécimal)	**octets UTF-8** (binaire)
0000 0000 – 0000 007F	0xxxxxxx
0000 0080 – 0000 07FF	110xxxxx 10xxxxxx
0000 0800 – 0000 FFFF	1110xxxx 10xxxxxx 10xxxxxx
0001 0000 – 001F FFFF	11110xxx 10xxxxxx 10xxxxxx 10xxxxxx
0020 0000 – 03FF FFFF	111110xx 10xxxxxx 10xxxxxx 10xxxxxx 10xxxxxx
0400 0000 – 7FFF FFFF	1111110x 10xxxxxx 10xxxxxx 10xxxxxx 10xxxxxx 10xxxxxx

Pour coder en UTF-8 à partir de caractères UCS-4, procédez de la manière suivante :

1) Déterminez le nombre d'octets nécessaires à partir de la valeur du caractère à coder et de la première colonne du tableau précédent.

2) Préparez les bits de poids fort à partir de la deuxième colonne du tableau.

3) Remplissez les bits x à partir des bits du caractère, en commençant par les bits de poids faible du caractère et en les mettant en premier dans les derniers octets de la séquence, et ainsi de suite jusqu'à ce que les bits x soient remplis.

L'algorithme pour coder en UTF-8 à partir de caractères UCS-2 (ou Unicode) peut être déduit de la méthode précédente, en étendant chaque caractère UCS-2 avec deux octets nuls. Cependant, les valeurs UCS-2 entre D800 et DFFF, qui sont actuellement des caractères UCS-4 transformés à l'aide d'UTF-16, nécessitent un traitement particulier : il faut annuler la transformation UTF-16 afin d'obtenir un caractère UCS-4 qui peut être transformé à l'aide de la méthode précédente.

Pour décoder de l'UTF-8 en UCS-4, procédez de la manière suivante :

1) Initialisez les 4 octets du caractère UCS-4 en plaçant tous les bits à 0.

2) Déterminez les bits qui codent la valeur du caractère à partir du nombre d'octets dans la séquence et la deuxième colonne du tableau (les bits x).

3) Placez les bits de la séquence dans le caractère UCS-4, en premier les bits de poids faible du dernier octet de la séquence et en avançant vers la gauche jusqu'à ce qu'il ne reste plus de bits x.

Si la séquence UTF-8 ne fait pas plus de 3 octets de long, le décodage peut fournir directement un caractère UCS-2 (ou l'équivalent en Unicode).

Un algorithme et une formule plus détaillés peuvent être consultés dans [FSS-UTF], [UNICODE] ou l'annexe R de [ISO-10646].

3. Exemples

La séquence Unicode « A<NOT IDENTICAL TO><ALPHA>. » (0041, 2262, 0391, 002E) sera codée : 41 E2 89 A2 CE 91 2E

La séquence Unicode « Hi Mom <WHITE SMILLING FACE>! » (0048, 0069, 0020, 004D, 006F, 006D, 0020, 263A, 0021) sera codée : 48 69 20 4D 6F 6D 20 E2 98 BA 21

La séquence Unicode représentant les caractères Han pour le mot japonais « nihongo » (65E5, 672C, 8A9E) sera codée : E6 97 A5 E6 9C AC E8 AA 9E

Enregistrements MIME

Ce document est fait pour servir de base pour l'enregistrement d'un codage de caractères MIME selon la RFC 1521 ([RFC1521]). La valeur du paramètre proposée est « UTF-8 ». Cette chaîne de caractères pourrait étiqueter les données contenant du texte composé de caractères issus du format ISO 10646-1 et codés en utilisant la méthode détaillée plus haut.

Considérations relatives à la sécurité

Ce document ne comporte aucune considération relative à la sécurité.

Remerciements

Les personnes suivantes ont participé à la rédaction et aux discussions concernant ce document :

James E. Agenbroad	Andries Brouwer
Martin J. Dürst	David Goldsmith
Edwin F. Hart	Kent Karlsson
Markus Kuhn	Michael Kung
Alain LaBonte	Murray Sargent
Keld Simonsen	Arnold Winkler

Bibliographie

[FSS-UTF] X/Open CAE Specification C501 ISBN 1-85912-082-2 28cm. 22p. pbk. 172g. 4/95, X/Open Company Ltd., "File System Safe UCS Transformation Format (FSS_UTF)", X/Open Preleminary Specification, Document Number P316. Also published in Unicode Technical Report #4.

[ISO-10646] ISO/IEC 10646-1:1993. International Standard -- Information technology -- Universal Multiple-Octet Coded Character Set (UCS) -- Part 1: Architecture and Basic Multilingual Plane. UTF-8 is described in Annex R, adopted but not yet published. UTF-16 is described in Annex Q, adopted but not yet published.

[RFC1521] Borenstein, N., and N. Freed, "MIME (Multipurpose Internet Mail Extensions) Part One: Mechanisms for Specifying and Describing the Format of Internet Message Bodies", RFC 1521, Bellcore, Innosoft, September 1993.

[RFC1641] Goldsmith, D., and M. Davis, "Using Unicode with MIME", RFC 1641, Taligent inc., July 1994.

[RFC1642] Goldsmith, D., and M. Davis, "UTF-7: A Mail-safe Transformation Format of Unicode", RFC 1642, Taligent, Inc., July 1994.

[UNICODE] The Unicode Consortium, "The Unicode Standard -- Worldwide Character Encoding -- Version 1.0", Addison-Wesley, Volume 1, 1991, Volume 2, 1992. UTF-8 is described in Unicode Technical Report #4.

[US-ASCII] Coded Character Set -- 7-bit American Standard Code for Information Interchange, ANSI X3.4-1986.

Coordonnées de l’auteur

Francois Yergeau
Alis Technologies
100, boul. Alexis-Nihon
Suite 600
Montreal QC H4M 2P2
Canada

Téléphone : +1 (514) 747-2547
Fax : +1 (514) 747-2561
Adresse email : fyergeau@alis.com

Coordonnées du traducteur

Nils Schaefer
sN Informatique (www.sninformatique.net)
16 rue George Sand
51420 Witry-lès-Reims
France

Téléphone : +33 (03) 26 04 58 65
Adresse email : contact@sninformatique.net