Nouvelle version d'Unicode, la 6.0

Stéphane Bortzmeyer
<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 16 octobre 2010

https://www.bortzmeyer.org/unicode-6-0.html

Le 11 octobre a vu la sortie d'une nouvelle version du jeu de caractères Unicode, la 6.0. On peut trouver une description des principaux changements en <http://www.unicode.org/versions/Unicode6.0.0/> mais voici ceux qui m'ont intéressé particulièrement.

Pour explorer plus facilement la grande base Unicode, j'utilise un programme qui la convertit en SQL <https://www.bortzmeyer.org/unicode-to-sql.html> et permet ensuite des faire des analyses variées. Comme la version 6 vient de sortir, vous n'avez probablement pas ses données dans votre `/usr/share/unicode` ou équivalent. Il faut donc d'abord télécharger les deux fichiers de la nouvelle version :

```
% wget http://www.unicode.org/Public/zipped/6.0.0/UCD.zip
% wget http://www.unicode.org/Public/zipped/6.0.0/Unihan.zip
```

et les dézipper dans le répertoire qu'utilise par défaut le programme, `./unicode-data/`. Ensuite, plus qu'à taper `make`, prendre un café, et on se retrouve avec une base de données relationnelle intégrant les nouveaux caractères. Faisons quelques requêtes SQL :

```
ucd=> SELECT count(*) AS Total FROM Characters;
 total
--------
 109449
```

Unicode aproche donc désormais les 110 000 caractères. Conclusion de cet élargissement, la majorité des caractères, pour la première fois dans l'histoire d'Unicode, est désormais en dehors du Plan Multilingue de Base :

```
ucd=> SELECT count(*) FROM Characters where codepoint > 65535;
 count
-------
 54854

ucd=> SELECT count(*) FROM Characters where codepoint <= 65535;
 count
-------
 54595
```

Cette nouvelle version en a apporté combien de caractères ?

```
ucd=> SELECT version,count(version) FROM Characters
            GROUP BY version ORDER BY version;
 version | count
---------+-------
...
 5.0     |  1369
 5.1     |  1624
 5.2     |  6648
 6.0     |  2088
```

Donc, une version moyenne, avec 2 088 caractères nouveaux.

Que sont ces nouveaux caractères ? Ils viennent des « nouvelles » écritures comme le br[Caractère Unicode non montré [1]]hm[Caractère Unicode non montré], des caractères longtemps discutés comme la roupie indienne, [Caractère Unicode non montré] (U+20B9), mais surtout beaucoup des très contestés "*emoji*", très utilisés au Japon dans les téléphones portables :

```
ucd=> SELECT To_U(codepoint) AS Codepoint, name FROM Characters
     WHERE version = '6.0';
...
 U+1F4A3   | BOMB
 U+1F300   | CYCLONE
 U+1F302   | CLOSED UMBRELLA
 U+1F303   | NIGHT WITH STARS
 U+1F304   | SUNRISE OVER MOUNTAINS
 U+1F305   | SUNRISE
 U+1F306   | CITYSCAPE AT DUSK
```

(Vous pouvez aussi regarder à quoi ils ressemblent <http://surf-style.us/manual3.htm>.) Pour une idée des discussions suscitées, voir l'excellent article de Roozbeh Pournader <http://www.advogato.org/person/roozbeh/diary/163.html>, l'expert Unicode de l'écriture arabe, qui est allé jusqu'à se pencher sur la mise en œuvre d'Unicode pour un musulman fervent et les problèmes qu'elle peut poser. (Je vous laisse imaginer ce que peut être le caractère U+1F37A contesté...)

Il n'y a pas que des ajouts, il y a aussi des changements. Unicode a des règles de stabilité qui empêchent, par exemple, le retrait d'un caractère et la réaffectation de son point de code. Mais ces règles n'empêchent pas des changements comme l'attribution d'un caractère à une nouvelle catégorie car l'ancienne classification était erronée. C'est ainsi que deux caractères Kannada, [Caractère Unicode non montré] (U+0CF1) et [Caractère Unicode non montré] (U+0CF2) sont passés de la catégorie So

1. Car trop difficile à faire afficher par LaTeX

(symboles divers) à Lo (lettres diverses). Cela a des conséquences pour d'autres normes qui utilisent Unicode comme les IDN. Ainsi, ce changement de catégorie fait passer ces deux caractères de Interdit à Autorisé dans les noms de domaine (cf. RFC 5892 [2]). Bien plus génant est le cas d'un caractère Tai Lue, [Caractère Unicode non montré] (U+19DA) qui avait été classé par erreur Nd (nombres décimaux) alors qu'il aurait dû être No (autres nombres) ;

```
ucd=> SELECT To_u(codepoint), name, category FROM Characters
                WHERE codepoint=x'19DA'::Integer;
  to_u  |            name             | category
--------+-----------------------------+----------
 U+19DA | NEW TAI LUE THAM DIGIT ONE | No
```

Résultat, il voyage en sens inverse, des Autorisés vers les Interdits. Cela a suscité un débat vif dans la liste idnabis <http://www.alvestrand.no/mailman/listinfo/idna-update>. Fallait-il suivre Unicode et accepter que des noms de domaines légaux deviennent illégaux ? Cela remettrait en cause la stabilité des IDN. Ou bien fallait-il le mettre dans les exceptions prévues par la section 2.7 du RFC 5892 ? (Cette liste d'exceptions est actuellement vide.) La question a été tranchée dans le premier sens : on n'ajoute pas d'exceptions et les noms qui auraient utilisé U+19DA deviennnent donc illégaux (la décision a été publiée et documentée dans le RFC 6452).

2. Pour voir le RFC de numéro NNN, https://www.ietf.org/rfc/rfcNNN.txt, par exemple https://www.ietf.org/rfc/rfc5892.txt