概要

波ダッシュ問題とは、Shift_JIS や EUC-JP のような JIS X 0208 ベースのコードから Unicode に変換する際に、一部の記号の対応する Unicode 符号位置がおかしいコード変換の実装がある問題を指します。記号の文字化けの原因となっています。この問題に現れる代表的な記号が波ダッシュであることから、波ダッシュ問題といいます。

一覧

以下に、問題の対象となる文字の一覧を記します。表の中の「文字名」は、JIS X 0208 にて規定されている文字名であり、Unicode (ISO/IEC 10646 UCS) の文字名に対応します。「誤り」の変換先に移す実装は、主に Microsoft Windows に見られます。

日本語通用名称	区点	SJIS	文字名	正しい Unicode 符号位置	誤り
波ダッシュ	1-33	8160	WAVE DASH	U+301C	U+FF5E
双柱	1-34	8161	DOUBLE VERTICAL LINE	U+2016	U+2225
負符号	1-61	817C	MINUS SIGN	U+2212	U+FF0D
セント記号	1-81	8191	CENT SIGN	U+00A2	U+FFE0
ポンド記号	1-82	8192	POUND SIGN	U+00A3	U+FFE1
否定記号	2-44	81CA	NOT SIGN	U+00AC	U+FFE2
ダッシュ (全角)	1-29	815C	EM DASH	U+2014	U+2015

文字化け

上記の「正しい Unicode 符号位置」を用いる変換だけを使っている分には問題を生じませんが、誤った変換が混入すると、変換したコードを元のコードに戻そうとした際に対応先がなくて「?」に化けるなどします。

例えば、波ダッシュ (WAVE DASH) SJIS 0x8160 を Unicode に変換する際に、上記の「誤り」の符号位置 U+FF5E (FULLWIDTH TILDE, 全角チルダ) にしてしまうと、変換後のデータを別のプログラムで再び SJIS に戻そうとした時に、U+FF5E に対応するコードが SJIS に存在しないために「?」などに化けてしまいます。

こうしたことが生じるケースは、例えば、Unicode に変換する時に CP932 (MS932, Windows-31J など) のような上記の誤った符号位置に移す変換器を用いてしまい、一方、逆方向の変換には Shift_JIS のような正しい変換器を用いた場合です。

対策

この問題を引き起こさないためには、上記「正しい符号位置」への変換を行うコード変換を常に用いることです。

この観点から最も推奨されるのは、Shift_JIS-2004 (Shift_JISX0213) の変換を用いることです。例えば iconv コマンドでは下記のようなオプション指定にてシフト JIS から UTF-8 へ変換します。

```
iconv -f SHIFT_JISX0213 -t UTF-8 < sjis.txt > utf8.txt
```

これにより、正しい Unicode 符号位置へ変換されます。この指定は、波ダッシュ問題を避けるだけでなく、丸つき数字やローマ数字等の救済にも効果があり、また第 3 第 4 水準漢字やアクセントつきラテン文字などといった日本で使用されている各種の文字への対応という点からも望まれます。

UTF-8 から SJIS へ変換するには、上の -f と -t を逆にします。

```
iconv -f UTF-8 -t SHIFT_JISX0213 < utf8.txt > sjis.txt
```

iconv 以外で同様の指定をするには、文字コードを指定する場面で例えば Java では "x-SJIS_0213", Python では "shift_jis-2004" とすると、Shift_JIS-2004 を指定できます。

一方、"CP932" や "Windows-31J" のような Windows のベンダ定義外字の指定を用いると、誤った変換が行われる可能性が高くなるので、用いるべきではありません。(CP932 と指定しても、上記の正しい変換を用いる実装もあるので、常に間違った変換が行われるとは限りません)

参考

- JIS X 0213 と Unicode の対応表 - 機械可読形式のコード変換表。正しい変換が得られます。
- プログラマのための文字コード技術入門 第 8 章