J と日本語処理

Masato SHIMURA
JCD02773@nifty.ne.jp

10 Dec 2005
JAPLA2005

目次

はじめに

J の極初期のバージョンは日本語を受け付けた。次第に文字コードの取り扱いが厳格になり 2 バイトキャラクターを表示しなくなった。しかし、文字コード自体は変化しないので、ソートや take(`{m {. y.`) のような簡単な処理をして EXCEL や CSV HTML に出力すれば問題なく表示出来る。

一方 J のバージョン 5 から導入された JAVA 版の J は 2 バイト文字も FONT を選択すれば問題なく表示される。しかし JAVA 版は BOX の形が崩れるなど未整備な面が残っており、漢字の確認程度にしか使用していない。

最近公開された J のバージョン 6 は 4 バイトのユニコードの本格的なサポートを始めた。当座の J601 は GRID が中心で、WIN 版の J の画面へは J602 からであるのとアナウンスである。

1　キャラクタの取り扱い

キャラクタを扱う動詞に *s* : (*Symbol*) がある。s:を付けて定義した文字列は数字と似た配列操作が柔軟に出来る。

```
  cp=. s: ' green red black silver'
```

```
   cp
‘green ‘red ‘black ‘silver
```

```
   $ cp
4
```

s : では最初のワードの前に半角スペースが必要。無い場合は 1 文字が落ちる。

```
cp0=:s: 'green red black silver'
```

```
  1 s: cp0
‘reen red black silver
```

```
  $ 1 s: cp
23
```

2 s: cp greenredblacksilver	
3 s: cp green red black silver	(<3 s: cp),<a=. ?.4 4 $ 20 ┌─────┬──────┐ │green │ 6 15 19 12 │ │red │14 19 0 17 │ │black │ 0 14 6 18 │ │silver │13 18 11 12 │ └─────┴──────┘
5 s: cp +-----+---+-----+------+ |green|red|black|silver| +-----+---+-----+------+	

1.1 s: と 日本語

s:は日本語を取り扱うことが出来る。ここでは UNICODE を想定しない。file だエディタで書いて J に流し込む。主な用途としては、CSV や HTML の見出しの行や列である。

ca=: s: ’ 埼玉 栃木 福島 宮城 山形 秋田 岩手 青森 ’

これに対して 3 s: ca が可能であり、HTML に落とせる。

エディタで書いても 2 バイト文字はデリケートである。次のように書いた方が確実かも知れない。この場合でも、最初に半角スペースが必要である。J の Font は ISIJ でよい。

```
def_cal=: 3 : 0
c0=. ’ 埼玉’
c1=. ’ 栃木’
```

```
c2=.’福島’
c3=.’宮城’
c4=.’山形’
c5=. ’秋田’
c6=. ’岩手’
c7=.’青森’
CA2=: c0;c1;c2;c3;c4;c5;c6;c7
)
         CA2
+----+----+----+----+----+----+----+----+
|埼玉|栃木|福島|宮城|山形|秋田|岩手|青森|
+----+----+----+----+----+----+----+----+
```

列の見出しや CSV に用いるときはデータを <"_2 や {@>で角砂糖のようにピース化する。列 行の双方に見出しを付ける *round_cal* を作成した。

```
 CA2 round_cal ?.8 8 $ 100
+----+----+----+----+----+----+----+----+----+
|    |埼玉|栃木|福島|宮城|山形|秋田|岩手|青森|
+----+----+----+----+----+----+----+----+----+
|埼玉|46  |55  |79  |52  |54  |39  |60  |57  |
+----+----+----+----+----+----+----+----+----+
|栃木|60  |94  |46  |78  |13  |18  |51  |92  |
+----+----+----+----+----+----+----+----+----+
|福島|78  |60  |90  |62  |31  |16  |60  |64  |
+----+----+----+----+----+----+----+----+----+
|宮城|64  |71  |13  |3   |76  |26  |25  |77  |
+----+----+----+----+----+----+----+----+----+
|山形|68  |48  |42  |91  |99  |97  |99  |9   |
+----+----+----+----+----+----+----+----+----+
|秋田|81  |64  |13  |89  |57  |52  |4   |73  |
+----+----+----+----+----+----+----+----+----+
|岩手|57  |95  |93  |28  |33  |16  |5   |27  |
```

```
+----+----+----+----+----+----+----+----+----+
|青森|37  |45  |46  |43  |17  |9   |74  |47  |
+----+----+----+----+----+----+----+----+----+
```

1.2 UNICODE

1.2.1 概況

UNICODE にも何種類かあるが、ここで取り上げるのは最も使われている UTF-8 である。UNICODE は最初は 2 バイトに固執していたが、UTF-8 では 4 バイトを用い、ASCII 文字が 1 バイトで表され、ギリシア文字やロシア文字は 2 バイト，漢字や仮名などは 3 バイトに、さらに 4 バイトに割り当てられた文字もあり、100 万を超える文字や記号を扱うことが出来、漢字の差異も収まったようだ。

[*1]

数学記号や APL 文字も割り当てられている。

[*2]

1.2.2 J と UNICODE

[*3]

u: がユニコードを扱う関数である。

```
   u:  321+i. 26
ABCDEFGHIJKLMNOPQRSTUVWXYZ


 u: 97+i. 26
abcdefghijklmnopqrstuvwxyz
```

[*1] 割り当て CODE の詳細は
http://homepage1.nifty.com/nomenclator/unicode/data/index.htm
http://www.kishugiken.co.jp/cn/code06c.html

[*2] 三省堂の UNICODE の辞典も出版されている。

[*3] WIN2000 の環境で GRID を MS 明朝の font を用いてを試したが、J6 でも成功していない。

無理に日本語を出すよりも少し先のサポートを期待しよう。*TEX* の世界でも *unicode* と *CJK font* の混乱は収まっていないようだ。

2　EXCEL との OLE

データファイルの取り扱いには EXSEL との OLE、CSV ファイルの利用、データベースとの ODBC ドライバによるリンク、更にはソケットによる本格的なリモート操作などの方法がある。

WIN のマイクロソフトの ODBC ドライバは日本語 ACESS との相性があまり良くなく、CSV を介在させている。CSV は単調だが 20 万件程度のデータを ACESS から取り込んで J に渡したり、J の多量のデータを save して ACESS に格納している。 EXCEL の処理能力の範囲なら当然 EXCEL で扱える。

志村 · 竹内の getexcel.ijs は今秋竹内により *charin* などが追加された。EXCEL から正規に取り込んだデータは全て文字列であるが、*datain* で取り込めば、既に数値に変換されている。

2.1　CSV ファイル

CSV ファイルはあまり横長でなければ、20 万件程度のファイルは楽々と取り扱ってくれる。

csv.ijs のファイルをロードする。

```
load 'csv files'
```

CSV ファイルを readcsv foo.csv で読み込む。ボックスに入っているので開く。(".@ >)

```
TDATA0 =: ".@> readcsv 'user\odbc\test_03.csv'
```

書き込みは *writecsv* で行う。

```
  a1 writecsv 'temp\textcsv.csv'
439
```

3　Grid

J の Grid はビューアとして利用でき、アプリケーションとしても使える。J で当初から推奨されていた over,by より簡単に扱える。

最初に *load jwatch* や *require jwatch'* で *load* しておく。

'foo' conew 'jwatch' の'foo' は名詞の名前で任意で " で囲む。ここで計算式は受け付けないので、事前に計算しておく。

conew はオブジェクト用のコマンドで 'jwatch は jwatch の呼び出す。

```
'a1' conew 'jwatch'
```

```
┌──┐
│20│
└──┘
```

4　HTML

4.1　Boxed data

Oleg Kobchenck が HTML 出力の素晴らしいスクリプト okhtml2.zip を公表している。

'html' tag 'body' tag html y.

'html' tag 'body' tag　の部分はは先頭と最後に

<html><body></body></html> を付ける機能なので、省略して後で、書き込んでも良い。

html foo fwrite 'temp.html' としてファイルに書き込む。

```
(html a1) fwrite 'temp\texthtml.html'
2926
```

Box にすれば, 巨大なテーブルも HTML で出せる。

嬉しいことに, 日本語も通り、EXCEL や DB の日本語の情報を,J で壊さずに処理すれば, 日本語の部分のデータも戻る.

5 script

```
round_cal=: 4 : 0
((<'')),x.),.  x. ,{@> y.
)

cp=:  s: ' green red black silver'
cp0=: s: 'green red black silver'

NB. ca0=: u: ' 埼玉外環 埼玉 栃木 福島 宮城 山形 秋田 岩手 青森 '

def_cal=: 3 : 0
c0=. ' 埼玉'
c1=. ' 栃木'
c2=.' 福島'
c3=.' 宮城'
c4=.' 山形'
c5=. ' 秋田'
c6=. ' 岩手'
c7=.' 青森'
CA2=: c0;c1;c2;c3;c4;c5;c6;c7
)

ca=:  s: ' 埼玉 栃木 福島 宮城 山形 秋田 岩手 青森'
```