

APL/J が着目されている？

横山 暁*

2009年5月30日

1 はじめに

JAPLA 研究会で発表するのは 2005 年 8 月の夏合宿以来である。2005 年に博士課程に進学した直後は、研究上でも J を使っていたのだが、その後、今行っている研究¹の既存研究の分析用のプログラムが C であったため、J から離れてしまっていた²。

今回、久しぶりに発表をしようと思い立ったのは、先日、竹内研の先輩と飲み会があり、その席で APL や J 言語の話が出たことが発端である。

2 APL/J と Rubyist

「J 言語基礎講座」³という Web Page がある。この Web Page は竹内研の OB の masa 氏によってかなり前に開設されたのだが、本人曰く「最近アクセスが増えていたり、リンクが張られていたりする」そうである。どうやら何者かが APL や J を使っているのである。

実際に Google で「APL 言語」と検索をしてみると、

『未来の言語は「APL」？ Ruby のまつもと氏が講演』

<http://www.atmarkit.co.jp/news/200902/13/matz.html>

や

『Rubyist のための他言語探訪【第 12 回】APL と J』

<http://jp.rubyist.net/magazine/?0019-Legwork>

というページがヒットした。どちらも Ruby 関連である。

ここで Ruby について簡単に説明しておく。

Ruby (ルビー) は、まつもとゆきひろ (通称 Matz) により開発されたオブジェクト指向スクリプト言語であり、従来 Perl などのスクリプト言語が用いられてきた領域

* 慶應義塾大学理工学部管理工学科 訪問研究員 (satoru_y@ae.keio.ac.jp)

¹ 主にクラスター分析法の一手法である重複クラスター分析法についての研究を行っている。

² 実際には、博士論文執筆時に指導教授から過去の研究の見直しを指示され、その時点で 3 年程前に書いた J のプログラムを再度解読して使っている。

³ J 言語基礎講座 : <http://masa.vis.ne.jp/Jkiso/>

でのオブジェクト指向プログラミングを実現する。Ruby は当初 1993 年 2 月 24 日に生まれ、1995 年 12 月に fj 上で発表された。名称の Ruby は、プログラミング言語 Perl が 6 月の誕生石である Pearl (真珠) と同じ発音をすることから、まつもとの同僚の誕生石 (7 月) のルビーを取って名付けられた。(Wikipedia より)

また、Ruby の愛好者のことを Rubyist と呼ぶ。

『未来の言語は「APL」？ Ruby のまつもと氏が講演』の中で、

次の世代のプログラミング言語が取り入れる可能性があるパラダイムとは何か？次に来るロスト・テクノロジーとは何か？

という質問に対し、まつもと氏が

「まだメインストリームになっていない言語で、パラダイムシフトがある言語、これから来る言語は以下の言語の延長線にあるのではないか」と述べて、Ruby、Haskell、Erlang、APL の 4 つを挙げた。

と答えている。ちなみにこの講演会の中で、まつもと氏は APL で作成したライフゲームのソースコードを紹介している。また、『Rubyist のための他言語探訪【第 12 回】APL と J』は、まつもとゆきひろ氏が 2007 年 5 月 18 日に書いている。つまり、少なくとも 2007 年頃には、まつもと氏は APL や J に何らかの形で興味をもっていたと考えられ、さらに Rubyist の間でも APL や J に着目している人が出ているようである。現に Rubyist の一人によって <http://rubyist.g.hatena.ne.jp/hatecha/20071029/p3> などに J の記事が書かれている⁴。

3 J で FizzBuzz

3.1 2007 年の秋に

FizzBuzz と言葉遊びをご存じだろうか。これは、言葉遊びの一つで、日本では「世界のナベアツ」が「3 の倍数と 3 のつく数字の時に馬鹿になる」というネタで世の中を笑わせている言葉遊びである。FizzBuzz は、1 から順に数えて行き、3 の倍数の時に Fizz、5 の倍数の時に Buzz と言わなければならないもので、必然的に 15 の倍数の時には FizzBuzz となる。

この FizzBuzz をプログラミング言語にによって実現する FizzBuzz 問題というのがあり、<http://recompile.net/2007/05/jfizzbuzz.html> で高井直人氏⁵が

酔っぱらった勢いで、J 言語で FizzBuzz を書くと言ってしまった

ようで、2007 年 5 月 21 日の Blog で実現を試みている。(ちなみにこの記事では「大学 2 年の頃に

⁴ 実際は「J 言語基礎講座」を引用しているだけの部分も多い。

⁵ ソフトウェアエンジニア。1977 年横浜生まれ。Java EE や Ruby といったオープンソース技術を活用したウェブサイトの企画、構築、運用などの分野で活躍中。(日本 Ruby 会議 2007 より)
現在は伊藤忠テクノソリューションズ株式会社所属のようである。

やった」と書いてあるので、慶應の理工かなと思ったりもしたのですが、どうも違うようだ。別の大学でも J を教えている人が居るんですね・・・)

ここでは、

```
A = . 1 + i. 100
```

と定義しておき、

```
0 E. 15 | A
```

によって 15 の倍数の時にフラグを立て、

```
(0 E. 15 | A) # A
```

によってフラグが立っているところを抽出しています。

その上で、X m } Y を用いて抽出された部分を置き換えるということを

```
'Fizz' (((0 E. 15 | A) # A) - 1) } A
```

としているのですが、

```
|domain error  
| 'Fizz' (((0 E.15|A)#A)-1)}A
```

となってしまって、

うう……。数値と文字の変換あたりでエラーがでてしまうようです。うまく文字列と数値を置換してあげる方法については、時間切れでした……。心より恥じる。どなたか J 言語ハッカーの方、お助けを！！

と困っています。

この記事があることを、2007 年 9 月 28 日に masa 氏から紹介され、

この問題解決してあげてくださいね。

という一言で、プログラムを書いてみることにしました。

高井氏も自覚しているように、文字列と数値の置換がうまくいっていないわけです。

そこで

```
(<'Fizz') (<:(0 E. 3 | A) # A) } <"0 A
```

と A を 1 つずつボックスで囲み、また 'Fizz' もボックスで囲み、ボックスを置換する事にしました。最終的には、数値と文字列が混合するボックスを外すために、強引に行の取り出しや置換を使って

```
' ' (((' E. B) # i.#B) } B=.}.):1{"(<'FizzBuzz') (<:(0 E. 15 | A) # A) }  
(<'Buzz') (<:(0 E. 5 | A) # A) } (<'Fizz') (<:(0 E. 3 | A) # A) } <"0 A=>:i.100
```

(実際は 1 行) という形で実現をしました。また, $i \cdot 100$ を y . (J601 以降は y とすれば良いらしい) として, 動詞として定義して使う形のものも作成しました。もっとうまいやり方があるような気もしたのですが, J を思い出しながら作成したので, この程度が限界で, 2007 年 10 月 12 日に Blog のコメント欄に, このプログラムを貼り付けて (その後コメント欄でのやり取りはあったものの), その時は終了しました。

3.2 FizzBuzz 再び

竹内研の先輩と飲み会の後, 気になったので, Google で「J 言語 FizzBuzz」と検索したところ, 前述した高井氏の Blog ではないところで, J 言語で FizzBuzz のプログラムを作成している人を発見した。niming538 氏(ハンドルネーム)は, はてなダイアリー(<http://d.hatena.ne.jp/niming538/>)にてで APL/J 言語についていくつかの記事(日記)を書いている。

その中で何日かにわたって FizzBuzz 問題にチャレンジしている。関係するページは以下の 4 つである。

```
http://d.hatena.ne.jp/niming538/20090309
http://d.hatena.ne.jp/niming538/20090310
http://d.hatena.ne.jp/niming538/20090312
http://d.hatena.ne.jp/niming538/20090313
```

niming538 氏は 2009 年 3 月 9 日の記事で,

```
a=:<"0 (1+i.100)
boxall=:a,'Fizz';'Buzz';'FizzBuzz'
```

と変換用のテーブルを用意しておいて,

```
a=:1+i.100
fizz=:101 (((0 E. 3|a) # a) - 1) } a
buzz=:102 (((0 E. 5|a) # a) - 1) } fizz
fizzbuzz=:103 (((0 E. 15|a) #a) - 1) } buzz
10 10$result=: (fizzbuzz - 1) { boxall
```

と半ば力づくで解いています。(翌日 3 月 10 日の記事で自己解説をしています。)

しかし 3 月 12 日の記事では, 前述の高井氏の Blog を発見したらしく, 私がコメント欄に貼り付けたプログラムを引用して, 解説をしています。

3 月 13 日の記事で, niming538 氏の完成品として,

```
FizzBuzz=: 3 : 0
(<'FizzBuzz') (<:(0 E. 15 | A) # A) } (<'Buzz') (<:(0 E. 5 | A) # A) }
(<'Fizz') (<:(0 E. 3 | A) # A) } <"0 A=.:i.y
)
```

というプログラムが記述されています (2 行目と 3 行目は実際は一行)。

これは J602 用で、

ボックスから出す必要はないと思うのでその分短くなっています。結果の表現は
10 × 10 の表にします。

とあるように、私のプログラムをベースに(ほぼそのままとも言う)プログラムを作成しています。

4 最後に

今回の発表では、インターネット上で Rubyist によって APL/J が着目されている事について、また、FizzBuzz 問題についての報告を行った。

何故、まつもとゆきひろ氏や、高井直人氏を中心に、Rubyist が APL/J に着目されているのか不明であるし、なぜ Rubyist なのかも不明である。彼らにとって APL/J の何が役に立つのか分からないが、APL/J のもつ独特さが単純に彼らの興味を惹いているようにも思われる。

Ruby は日本発であり、オブジェクト指向のスクリプト言語ということで、今後も発展していく言語であろう。APL/J の発展のためには、彼らの興味が続いているうちに、Web マニュアルを充実するなどして APL/J を紹介していくと良いのではないだろうか。