Build your own tools

Shugo Maeda

Network Applied Communication Laboratory Ltd.

2018-06-02

Self introduction

- Shugo Maeda

- Ruby committer

- Secretary General at Ruby Association

- Director at Network Applied Communication Laboratory Ltd. (NaCl)

Ruby Association

- Foundation dedicated to Ruby
  - Maintenance of the stable version of Ruby
  - Grant
  - Conferences and Seminars
  - Case studies at www.ruby.or.jp
  - Examination

[ANN] Grant report

- Date: 2018-07-07

- Venue: Shinagawa, Tokyo

- Speakers: Matz, Naotoshi Seo, Uchio Kondo, Takashi Kokubun

- https://rubyassociation.doorkeeper.jp/events/74355

[ANN] RubyWorld Conference

- Date: 2018-11-01 〜 2018-11-02

- Venue: Matsue, Shimane

- Keynote Speakers: Matz, Chad Fowler

- http://www.rubyworld-conf.org

NaCl

- Commissioned development of Web (and other) applications

- Foucused on FLOSS

- Matz belongs to NaCl since 1997

- Head office in Shimane / Branch office in Tokyo

FLOSS

- Free/Libre and Open Source Software

Free software

- ~~Free as in beer~~
- Free as in speech
- 自由ソフトウェア
- Social movement

Open source software

- Open Source Definition
  - Published by the Open Source Initiative
  - Based on Debian Free Software Guidelines

- Development methodology

Microsoft

- Joined the Linux Foundation
  - https://www.linuxfoundation.org/press-release/microsoft-fortifies-commitment-to-open-source-becomes-linux-foundation-platinum-member/

- Joined the Open Source Initiative
  - https://opensource.org/node/901

Did FLOSS win?

Web applications

- Everyone use them

- FLOSS used often
  - Linux, GNU toolchain, Apache, Ruby, Rails…

I hate Web applications

I hate

- Gmail
- Google Docs, Sheets, and Slides
- Cloud9

These are OK

- tDiary

- Redmine

- Amazon (shopping site)

- Netflix

What is common to these?

SaaSS

- ≠ SaaS (Software as a Service)

- **S**ervice **a**s **a** **S**oftware **S**ubstitute

  - https://www.gnu.org/philosophy/who-does-that-server-really-serve.html

- Services for your computing

Non SaaSS

- FLOSS
  - tDiary
  - Redmine

- Not for your computing
  - Amazon
  - Netflix

Take back control over your computing

Build your own tools

Why?

- You can't always get what you want
- Serve yourself
- For fun

Why Ruby?

- Availability
- Productivity
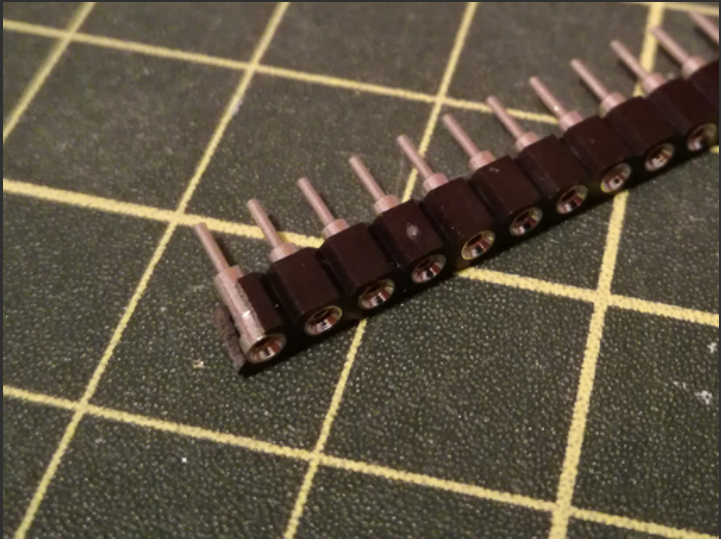- Text handling power
- Ecosystem
- Fun

My own tools

Keyboard

Viterbi

- Split Ortholinear keyboard kit

- Available at https://keeb.io

Hard to fix hardware

Free keyboard

QMK Firmware

- Open source firmware for AVR and ARM based keyboards

- Lincensed under GPL

T-Code Keyboard

- T-Code
  - Direct Kanji Input
  - No Kana-to-Kanji conversion
  - Kanji composition
    - e.g. 五 + 口 = 吾

- Implemented on QMK Firmware

No Ruby Inside

- Max firmware size: 28KB

- Even mruby/c is too large

- Written in C

- http://nacl-ltd.github.io/2018/01/23/japanese-input-by-qmk.html

Table generation

```ruby
require_relative "tc_tbl"

puts "/* Generated by gentbl.rb; DO NOT EDIT! */"
puts "const uint16_t PROGMEM tcode_table[] = {\n"
TC_TBL.flat_map { |row|
  row.gsub(" ", "").each_char.map { |ch|
    "0x%04X" % ch.ord
  }
}.each_slice(8) do |cs|
  puts "  " + cs.join(", ") + ","
end
puts "};\n"
```

Generated table

```
 * Generated by gentbl.rb; DO NOT EDIT! */
const uint16_t PROGMEM tcode_table[] = {
  0x25A0, 0x25A0, 0x25A0, 0x25A0, 0x25A0, 0x25A0, 0x25A0, 0x25A0,
  0x25A0, 0x25A0, 0x30EE, 0x30F0, 0x30F1, 0x30F5, 0x30F6, 0x8ACB,
  0x5883, 0x7CFB, 0x63A2, 0x8C61, 0x308E, 0x3090, 0x3091, 0x25A0,
  0x25A0, 0x76DB, 0x9769, 0x7A81, 0x6E29, 0x6355, 0x25A0, 0x25A0,
  0x25A0, 0x25A0, 0x25A0, 0x4F9D, 0x7E4A, 0x501F, 0x9808, 0x8A33,
  0x25A0, 0x25A0, 0x25A0, 0x25A0, 0x25A0, 0x25A0, 0x25A0, 0x25A0,
  0x25A0, 0x25A0, 0x4E11, 0x81FC, 0x5BB4, 0x7E01, 0x66F3, 0x5C1A,
  0x8CC0, 0x5CB8, 0x8CAC, 0x6F01, 0x65BC, 0x6C5A, 0x4E59, 0x7A4F,
  0x25A0, 0x76CA, 0x63F4, 0x5468, 0x57DF, 0x8352, 0x25A0, 0x25A0,
  0x25A0, 0x25A0, 0x25A0, 0x7E54, 0x7236, 0x679A, 0x4E71, 0x9999,
  0x25A0, 0x25A0, 0x25A0, 0x25A0, 0x25A0, 0x25A0, 0x25A0, 0x25A0,
  0x25A0, 0x25A0, 0x9B3C, 0x865A, 0x72ED, 0x8105, 0x9A5A, 0x820E,
  0x559C, 0x5E79, 0x4E18, 0x7CD6, 0x5947, 0x65E2, 0x83CA, 0x5374,
  0x4EAB, 0x5EB7, 0x5F92, 0x666F, 0x51E6, 0x305C, 0x25A0, 0x25A0,
  0x25A0, 0x25A0, 0x25A0, 0x8B72, 0x30D8, 0x6A21, 0x964D, 0x8D70,
  0x25A0, 0x25A0, 0x25A0, 0x25A0, 0x25A0, 0x25A0, 0x25A0, 0x25A0,
```

Small scripts

ftpup

- Upload files via FTP

- FTP was used to upload HTML files in the past

- Using Net::FTP

imaparchive

- Move or Delete old mails on IMAP servers

- Using Net::IMAP

rd2sxi

- Convert RD to OpenOffice Impress slides

rd2latex

- Convert RD to LaTeX files

Programming languages

demi

- Original language for JVM
- Lisp-1
- No class definition
- `for` expression
- Closures
- REPL (Read-Eval-Print Loop)

Example

```
frame = new Frame("Test");
frame.setLayout(new FlowLayout());
hello = new Button("Hello");
hello.addActionListener([ e:
    println("Hello, World!");
]);
frame.add(hello);
```

babel

- Sather Compiler for .NET

Sather

- Statically typed object oriented language

- Separation of Subtyping and Code Inclusion

- Iterators

- Closures

Example

```
abstract class $PERSON is
   say;
end;

class HELLO_MIXIN is
   say is
      #OUT + "hello.\n";
   end
end;

class PERSON < $PERSON is
   include HELLO_MIXIN;
end
```

Servers

rskkserv

- Dictionary server for SKK

- SKK: Japanese Input Method

- Binary search

ximapd

- eXperimental IMAP server

- Use mailbox names as search queries

TUI applications

TUI

- **T**ext-based **U**ser **I**nterface
- Used on terminals
- More rich than CUI

Textbringer

- Emacs-like text editor

- Less parentheses
  - Matz Lisp a.k.a. Ruby

- Cool name

Mournmail

- MUA (**M**essage **U**ser **A**gent)

- Textbringer plugin

- Gmail like search interface
  - Powered by Groonga/Rroonga

- Multi account support

- Unstable specification

Groonga/Rroonga

- Groonga
  - Full-text search engine

- Rroonga
  - Ruby interface for Groonga

Good support

Search multiple columns

```ruby
# query = "jemalloc"
# query = "from:@matz subject:@ruby"
messages = Groonga["Messages"].select { |record|
  record.match(query) { |match_record|
    match_record.subject | match_record.body
  }
}.paginate([["date", :desc]], page: page, size: 100)
```

Demo

How to build tools

Materials

- Motivation

- Time

- Skills

Feel good about yourself

- Many geniuses in the world

- I'm not…

- But it's OK

Find your favorite theme

- Text editors?

- Programming languages?

- Graphical tools?

- Machine learning?

Start with something easy

- One liners
- Small scripts

Examples

- Mail archivers over Message user agents

- Slide generators over Presentation tools

Phisical LOC

imaparchive	99
ftpup	294
rskkserv	316
Mournmail	2236
ximapd	5101
Textbringer	6526
babel	11822
demi	12205

- exluding test code

Cut corners

- Create a Minimum Viable Product before getting bored

- Reinvent the wheel only when you want to

- Inconvenience vs Implementation cost

Examples in programming languages

- Use existing virtual machines

- Use existing libraries

Examples in text editors

- ASCII support only at first

- Text User Interface

- Redisplay by curses

Postpone test writing

- You don't know what you want at first

- Use it everyday instead

- Test code should help changes

Specification bugs

- What's wrong in the following code?

```ruby
def move_mail(src_mailbox, uid, dst_mailbox)
  src_path = Mournmail.mail_cache_path(src_mailbox, uid)
  dst_path = Mournmail.mail_cache_path(dst_mailbox, uid)
  begin
    File.rename(src_path, dst_path)
  rescue Errno::ENOENT
  end
```

UID is mailbox local

- UID is changed after COPY

- Change cache paths
  - Old: mailbox/uid
  - New: 00-ff/hash

Take risks

- eval

- monkey patching

- refinements

- callcc

Protect your data

- Leave mails on the IMAP server

- Dump unsaved files on crash

- Backup / Use VCS for important data

Make it extensible

- Small core with extensions

- Emacs = **E**ditor **MAC**ro**S**

Emacs vs vi

- Key bindings don't matter

- Extensible or not?

- Vim is an Emacs-type editor

Extension languages

	Host language	Extension language
GNU Emacs	C	Emacs Lisp
Vim	C	Vim script
Textbringer	Ruby	Ruby

Ruby

- Suitable for extensions

- Dynamic

- Brief notation

Customization

```ruby
def load_user_config
  config_file = File.expand_path("~/.textbringer.rb")
  begin
    load(config_file)
  rescue LoadError
  end
end
```

~/.textbringer.rb

```ruby
CONFIG[:east_asian_ambiguous_width] = 2
Buffer.detect_encoding_proc = Buffer::NKF_DETECT_ENCODING

Face.define :mode_line, foreground: "#ffffff", background: "#75507b"

add_hook :c_mode_hook, -> {
  buf = Buffer.current
  if buf.file_name &&
      File.dirname(buf.file_name).end_with?("/ruby")
    buf[:indent_level] = 4
    buf[:tab_width] = 8
    buf[:indent_tabs_mode] = true
    buf[:c_continued_statement_offset] = 4
    buf[:c_case_label_offset] = -2
    buf[:c_label_offset] = -2
  end
}
```

Plugins

- Load files in ~/.textbringer/ plugins

- Packaging system
  - RubyGems

Gem.find_latest_files

```ruby
def self.load_plugins
  files = Gem.find_latest_files("textbringer_plugin.rb", false) +
    Dir.glob(File.join(directory, "*/**/textbringer_plugin.rb"))
  files.each do |file|
    begin
      load(file)
    rescue Exception => e
      show_exception(e)
    end
  end
end
```

Interactive programming

- Change behavior of object at runtime

- load, eval, open classes

- REPL

- Examples in Textbringer
  - eval_buffer, eval_region, eval_expression

Unreloadable code

```
module Mournmail
  @current_mailbox = nil
```

Keep states on reload

```ruby
module Mournmail
  def self.define_variable(name, initial_value: nil,
                                 attr: nil)
    var_name = "@" + name.to_s
    if !instance_variable_defined?(var_name)
      instance_variable_set(var_name, initial_value)
    end
    ...
  end

  define_variable :current_mailbox, attr: :accessor
```

Local monkey patching

```ruby
module FillExtension
  refine Buffer do
    def fill_region(s, e)
      ...
    end
  end
end

module Commands
  using FillExtension
  define_command(:fill_region) do
    |s = Buffer.current.point, e = Buffer.current.mark|
    Buffer.current.fill_region(s, e)
  end
end
```

Internal DSLs

- Declarative

- Too much DSLs decrease readability as **Ruby code**

Parser combinator

```ruby
using RaddDjur::DSL

grammar = RaddDjur::Grammar.new(:additive) {
  # additive ← multitive '+' additive / multitive
  define :additive do
    :multitive.bind { |x|
      "+".bind {
        :additive.bind { |y|
          ret x + y
        }
      }
    } / :multitive
  end
```

Bootstrapping

- Self-compiling compiler
  - C → Language X → Language X

- Self-editing text editor
  - Vim → Textbringer → Textbringer

After finished

Osusowake

- Japanese word meaning sharing
  - Share meals when cooked too much

- Software never decrease

- Share your software as FLOSS

Summary

- Take back control over your computing

- Build your own tools by Ruby

- Share your tools as FLOSS