GMOペパボ

How to Begin Developing Ruby Core

Introduction for language testing

Hiroshi SHIBATA / GMO PEPABO inc.

2016.06.21 RedDotRubyConf 2016

self.introduce

```
=>
{
  name: “SHIBATA Hiroshi”,
  nickname: “hsbt”,
  title: “Chief engineer at GMO Pepabo, Inc.”,
  commit_bits: [“ruby”, “rake”, “rubygems”, “rdoc”, “psych”, “syck”, “ruby-build”, “railsgirls”, “railsgirls-jp”, …],
  sites: [“hsbt.org”, “ruby-lang.org”, “rubyci.org”, “railsgirls.com”, “railsgirls.jp”],
}
```



How to contribute to OSS

Contributing of OSS

People say:
“Contributing to OSS is easy! Please write some documentation and submit a patch!”

You say:
“Okay! I will contribute new documentation for Ruby!”



Documentation is hard

But documentation is hard, I think

- No-one knows the true behavior of OSS
  - Only author knows that.
- Documentation is tedious work :bow:

Because documentation is valuable work.



Testing and Running are easy

On the other hand:

- Ruby has a lot of test ecosystem and libraries.
- Bundler and Docker provide an encapsulated environment.

If you get test failures, you can submit issue ticket to tracker like github. It helps OSS author.



Let’s try
to
contribute ruby core

Start to test Ruby language

You can invoke language tests with the following instructions:

```
$ git clone https://github.com/ruby/ruby
$ cd ruby
$ autoconf
$ ./configure —disable-install-doc
$ make -j
$ make check
```



```
% make check TESTS='-j4'
(snip)

PASS all 1010 tests
exec ./miniruby -I./lib -I. -I.ext/common  ./tool/runruby.rb --extout=.ext  -- --disable-gems "./bootstraptest/runner.rb" --ruby="ruby --disable-gems"   ./
KNOWNBUGS.rb
2016-06-18 15:42:02 +0900
Driver is ruby 2.4.0dev (2016-06-18 trunk 55440) [x86_64-darwin15]
Target is ruby 2.4.0dev (2016-06-18 trunk 55440) [x86_64-darwin15]
last_commit=* test/rubygems/test_gem_installer.rb: Fixed broken test with extension   build. https://github.com/rubygems/rubygems/pull/1645

KNOWNBUGS.rb  PASS 0
No tests, no problem

test succeeded
Run options: "--ruby=./miniruby -I./lib -I. -I.ext/common  ./tool/runruby.rb --extout=.ext  -- --disable-gems"

# Running tests:

Finished tests in 2.513254s, 87.9338 tests/s, 177.4592 assertions/s.
221 tests, 446 assertions, 0 failures, 0 errors, 0 skips

ruby -v: ruby 2.4.0dev (2016-06-18 trunk 55440) [x86_64-darwin15]
Run options: "--ruby=./miniruby -I./lib -I. -I.ext/common  ./tool/runruby.rb --extout=.ext  -- --disable-gems" --excludes-dir=./test/excludes --name=!/
memory_leak/ --exclude=/testunit/ --exclude=/minitest/ -j4

# Running tests:

[ 89/803] 4625=test_bigdecimal 4626=test_table 4627=test_encodings 4628=test_bug_reporter
```



common.mk

- check/check-ruby
- btest/btest-ruby
- test-basic/test-knownbug
- test-testframework
- test
- test-all
- test-almost
- test-ruby
- test-rubyspec



make test

- btest-ruby
  - snip for next slides
- test-basic
  - invoke `basictest/runner.rb` with target ruby
- test-knownbug
  - invoke `KNOWNBUGS.rb`
  - It’s empty a lot of the time.



make btest-ruby

```
% ls bootstraptest
pending.rb  runner.rb*  test_attr.rb  test_autoload.rb  test_block.rb  test_class.rb  test_eval.rb  test_exception.rb  test_finalizer.rb  test_flip.rb
test_flow.rb  test_fork.rb  test_gc.rb  test_io.rb  test_jump.rb  test_literal.rb  test_literal_suffix.rb  test_load.rb  test_marshal.rb  test_massign.rb
test_method.rb  test_objectspace.rb  test_proc.rb  test_string.rb  test_struct.rb  test_syntax.rb  test_thread.rb
```

What's `bootstraptest/runner.rb` ?

- load test files and invoke them
- define assertion methods like `assert_equal` etc.



`cat bootstraptest/test_class.rb`

```
assert_equal 'true',    %q( class C; end
                            Object.const_defined?(:C) )
assert_equal 'Class',   %q( class C; end
                            C.class )

(snip)

assert_equal 'Class',   %q( class A; end
                            class C < A; end
                            C.class )

(snip)

assert_equal 'M',       %q( module M; end
                            M.name )

(snip)

assert_equal 'A::B',    %q( class A; end
                            class A::B; end
                            A::B )
```



test-basic

```
a, = nil; test_ok(a == nil)
a, = 1; test_ok(a == 1)
a, = []; test_ok(a == nil)

(snip)

def r; return *[]; end; a = r(); test_ok(a == [])
def r; return *[1]; end; a = r(); test_ok(a == [1])
def r; return *[nil]; end; a = r(); test_ok(a == [nil])

(snip)

f = lambda { |a, b=42, *c| [a,b,c] }
test_ok(f.call(1      ) == [1,42,[  ]] )
test_ok(f.call(1,43   ) == [1,43,[  ]] )
test_ok(f.call(1,43,44) == [1,43,[44]] )

(snip)
```



These are for Developer of VM/GC/Syntax/etc

make test-all

test-all invokes test files under the `test` directory. test-all has some options for testing:

- make test-all TESTS=“logger”
  - test only files under `test/logger`

- make test-all TESTS=“-j4”
  - it make parallel execution with 4 processes.



cat `test/ruby/test_array.rb`

```
% cat test/ruby/test_array.rb
# coding: US-ASCII
require 'test/unit'

class TestArray < Test::Unit::TestCase

(snip)

  def test_percent_i
    assert_equal([:foo, :bar], %i[foo bar])
    assert_equal([:"\"foo"], %i["foo])
  end

  def test_0_literal
    assert_equal([1, 2, 3, 4], [1, 2] + [3, 4])
    assert_equal([1, 2, 1, 2], [1, 2] * 2)
    assert_equal("1:2", [1, 2] * ":")

(snip)
```



cat `test/logger/test_logger.rb`

```
% cat test/logger/test_logger.rb
# coding: US-ASCII
require 'test/unit'
require 'logger'
require 'tempfile'

class TestLogger < Test::Unit::TestCase

(snip)

 def test_add
    logger = Logger.new(nil)
    logger.progname = "my_progname"
    assert(logger.add(INFO))
    log = log_add(logger, nil, "msg")
    assert_equal("ANY", log.severity)
    assert_equal("my_progname", log.progname)
(snip)
```



cat `test/-ext-/array/test_resize.rb`

```
% cat ext/-test-/array/resize/resize.c
#include "ruby/ruby.h"

static VALUE
ary_resize(VALUE ary, VALUE len)
{
    rb_ary_resize(ary, NUM2LONG(len));
    return ary;
}

void
Init_resize(void)
{
    rb_define_method(rb_cArray, "__resize__", ary_resize, 1);
}
```

```
require 'test/unit'
require '-test-/array/resize'

class TestArray < Test::Unit::TestCase
  class TestResize < Test::Unit::TestCase
    def test_expand
      feature = '[ruby-dev:42912]'
      ary = [*1..10]
      ary.__resize__(10)
      assert_equal(10, ary.size, feature)
      assert_equal([*1..10], ary, feature)
      ary.__resize__(100)
      assert_equal(100, ary.size, feature)
(snip)
```



make check

make check depends on the following definitions:

- main
  - build encodings and extensions.
- test
- test-testframework
  - run tests for `testunit` and `minitest`
- test-almost
  - run tests under `test` excluding `testunit` and `minitest`



These are for “you”

make test-rubyspec

CRuby has `make update-rubyspec` and `make test-rubyspec` tasks.

- `make update-rubyspec`
  - pulls ruby/rubyspec and ruby/mspec into the spec directory.

- `make test-rubyspec`
  - invokes mspec with the ruby binary and the latest rubyspecs.



cat spec/rubyspec/core/string/append_spec.rb

```
% cat spec/rubyspec/core/string/concat_spec.rb
require File.expand_path('../../../spec_helper', __FILE__)
require File.expand_path('../fixtures/classes', __FILE__)
require File.expand_path('../shared/concat', __FILE__)

describe "String#<<" do
  it_behaves_like :string_concat, :<<
  it_behaves_like :string_concat_encoding, :<<
end
```

```
% cat spec/rubyspec/core/string/shared/concat.rb
describe :string_concat, shared: true do
  it "concatenates the given argument to self and returns self" do
    str = 'hello '
    str.send(@method, 'world').should equal(str)
    str.should == "hello world"
  end

(snip)
```

Please check details: https://github.com/ruby/spec



These are for developer
who
interest compatibility

Ruby 2.4.0-preview1 Released

Posted by naruse on 20 Jun 2016

We are pleased to announce the release of Ruby 2.4.0-preview1.

Ruby 2.4.0-preview1 is the first preview of Ruby 2.4.0. This preview1 is released earlier than usual because it includes so many new features and improvements. Feel free to send feedback since you can still change the features.

Unify Fixnum and Bignum into Integer

Though ISO/IEC 30170:2012 doesn't specify details of the Integer class, CRuby has two visible Integer classes: Fixnum and Bignum. Ruby 2.4 unifies them into Integer.

String supports Unicode case mappings

`String/Symbol#upcase/downcase/swapcase/capitalize(!)` now handle Unicode case mappings instead of only ASCII case mappings.

Performance improvements

Ruby 2.4 also contains the following performance improvements including

Recent News

Ruby 2.4.0-preview1 Released

ConFoo Vancouver 2016 is looking for Ruby speakers

Ruby 2.3.1 Released

Ruby 2.2.5 Released

Ruby 2.1.10 Released

Syndicate

Recent News (RSS)

Please contribute tests to ruby

You can invoke CRuby tests:

```
$ git clone https://github.com/ruby/ruby
$ cd ruby
$ autoconf
$ ./configure —disable-install-doc
$ make -j
$ make check
```

You can invoke focused tests with coverage:

```
$ COVERAGE=1 make test-all TESTS=“logger”
$ open coverage/index.html
```

You can code new tests or update existing tests. and submit patches to our tracker or github.com/ruby/ruby.



Ruby (language) testing is so easy