# Practical Testing of Ruby Core

## Details for language testing

GMOペパボ株式会社



# self.introduce

=>

{

name: “SHIBATA Hiroshi”,

nickname: “hsbt”,

title: “Chief engineer at GMO Pepabo, Inc.”,

commit_bits: [“ruby”, “rake”, “rubygems”, “rdoc”, “psych”, “syck”, “ruby-build”, “tdiary”, “railsgirls”, “railsgirls-jp”, …],

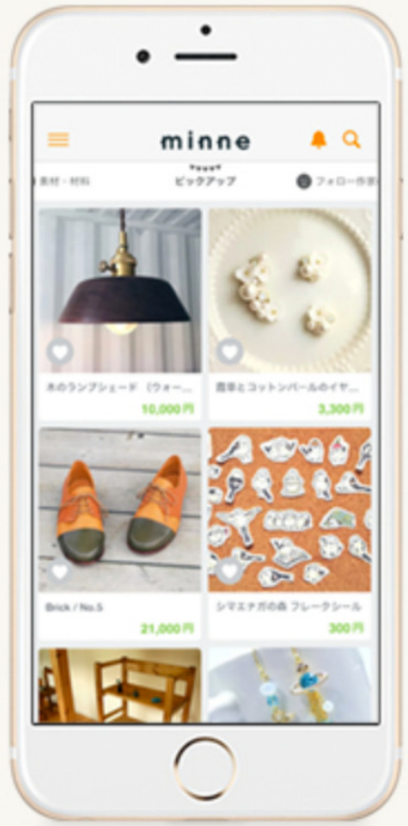
sites: [“hsbt.org”, “ruby-lang.org”, “rubyci.org”, “railsgirls.com”, “railsgirls.jp”],

}








テレビCM
公開中!

最新ニュース　セミナー・イベント情報　IRニュース

2015.12.02 お知らせ GMOペパボ

「J-Stock」銘柄選定に関するお知らせ

グローバルサイン認証サイト
SSL secured クリックして確認
GlobalSign by GMO

株価情報

pepabo.com

もっと

おもしろくできる

# Our challenges on GMO Pepabo, Inc.

## Infrastructure as a Code

- https://github.com/yaocloud
- https://github.com/matsumoto-r/mod_mruby
- https://github.com/matsumoto-r/ngx_mruby
- https://github.com/matsumoto-r/rcon

## Productive Development with OSS

- https://github.com/pepabo/capistrano-stretcher
- https://github.com/pepabo/mackerel-rb
- https://github.com/pyama86/malsh
- https://github.com/linyows/capistrano-github-releases



# self.introduce

=>

{

name: “SHIBATA Hiroshi”,

nickname: “hsbt”,

title: “Chief engineer at GMO Pepabo, Inc.”,

commit_bits: [“ruby”, “rake”, “rubygems”, “rdoc”, “psych”, “syck”, “ruby-build”, “tdiary”, “railsgirls”, “railsgirls-jp”, …],

sites: [“hsbt.org”, “ruby-lang.org”, “rubyci.org”, “railsgirls.com”, “railsgirls.jp”],

}

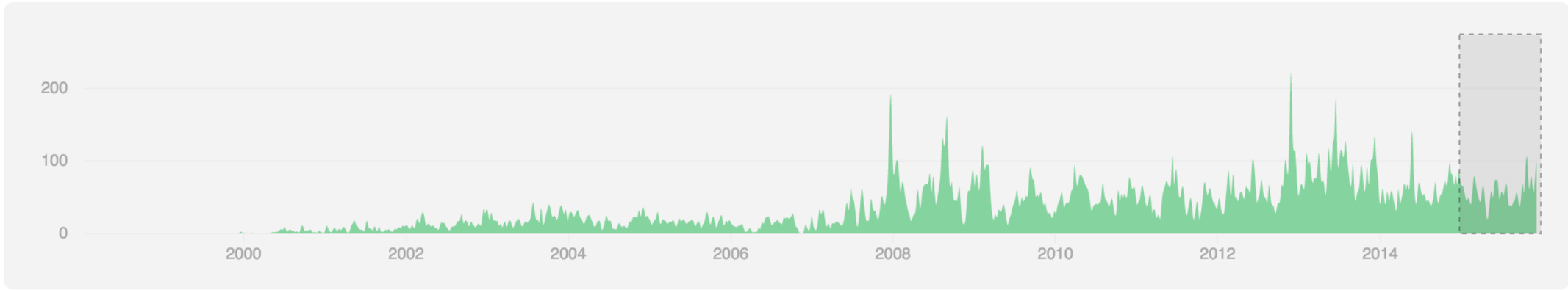




# Dec 25, 2014 – Dec 25, 2015

Contributions to trunk, excluding merge commits

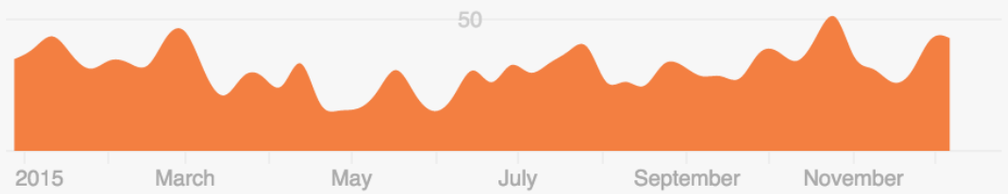
Contributions: **Commits**






**nobu** #1

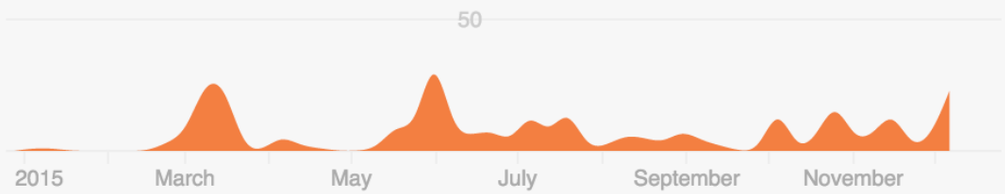
1,573 commits / 35,698 ++ / 16,940 --






**ko1** #2

325 commits / 20,009 ++ / 10,463 --






**hsbt** #3

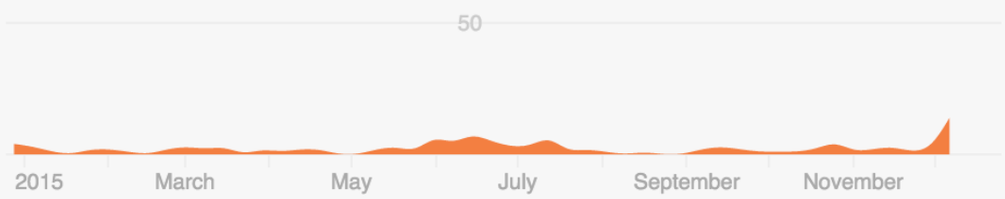
212 commits / 24,735 ++ / 27,998 --






**znz** #4

118 commits / 321 ++ / 205 --






**unak** #5

114 commits / 1,162 ++ / 377 --






**nurse** #6

113 commits / 1,844 ++ / 1,120 --



# 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 the Ruby language.
  - Only Matz knows that.
  - English is hard (for Japanese)
  - Documentation is boring work :bow:

Because documentation is valuable work.



# Testing and Running are easy

On the other hand, testing and running code is easy.

- 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 our tracker. It helps ruby committer.

If test coverage is missing for some ruby code, you can also write new tests and submit a patch to upstream.



# Code reading tips

## I always start code reading with the following commands

```
$ git clone https://github.com/some/gems
$ cd gems
$ less .travis.yml
```

## I pick out `before_script` and `script` code from .travis.yml and invoke it. For example:

```
$ bundle install
$ rake spec
$ rake spec:plugins
```



# In the case of ruby

```
$ git clone https://github.com/ruby/ruby
$ cd ruby
$ less .travis.yml
```

## You will get…



# .travis.yml in ruby/ruby

```
before_script:
  - "if [[ $TRAVIS_OS_NAME = 'osx' ]]; then rm -f ~/Library/Logs/DiagnosticReports/ruby_*.crash; fi"
  - "uname -a"
  - "uname -r"
  - "rm -fr .ext autom4te.cache"
  - "make -f common.mk BASERUBY=ruby MAKEDIRS='mkdir -p' srcdir=. update-config_files"
  - "autoconf"
  - "mkdir config_1st config_2nd"
  - "./configure -C --disable-install-doc --with-gcc=$CC $CONFIG_FLAG"
  - "cp -pr config.status .ext/include config_1st"
  - "make reconfig"
  - "cp -pr config.status .ext/include config_2nd"
  - "diff -ru config_1st config_2nd"
  - "make after-update BASERUBY=ruby"
  - "make -s $JOBS encs"
  - "make -s $JOBS exts"
  - "make update-rubyspec"
  - "if [[ $TRAVIS_OS_NAME = 'osx' ]]; then echo 'exclude :test_deadlock_by_signal_at_forking, \"under
investigation\"' >> test/excludes/TestProcess.rb; fi"
script:
  - "make test"
  - "make test-all TESTOPTS='-q -j2'"
  - "make test-rubyspec"
```



# Dive into testing of ruby language

# Tips for testing ruby

## 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
(snip)

test succeeded
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
2015-12-04 11:53:44 +0900
Driver is ruby 2.3.0dev (2015-12-03 trunk 52872) [x86_64-darwin15]
Target is ruby 2.3.0dev (2015-12-03 trunk 52872) [x86_64-darwin15]

KNOWNBUGS.rb  PASS 0
No tests, no problem
Run options: "--ruby=./miniruby -I./lib -I. -I.ext/common  ./tool/runruby.rb --extout=.ext  -- --disable-gems"

# Running tests:

Finished tests in 2.009208s, 109.4959 tests/s, 219.9872 assertions/s.
220 tests, 442 assertions, 0 failures, 0 errors, 0 skips

ruby -v: ruby 2.3.0dev (2015-12-03 trunk 52872) [x86_64-darwin15]
Run options: "--ruby=./miniruby -I./lib -I. -I.ext/common  ./tool/runruby.rb --extout=.ext  -- --disable-gems" --
excludes=./test/excludes -x /memory_leak/ -x /testunit/ -x /minitest/

# Running tests:

[  365/15731] Fiddle::TestFunc#test_qsort1
```

# common.mk

common.mk defines test tasks for the make command

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



# make test

make test depends on the following tests

- test-sample
  - invoke `tool/rubytest.rb` with target ruby
  - rubytest.rb run `sample/test.rb` !!1

- btest-ruby
  - snip for next slides

- test-knownbug
  - invoke `KNOWNBUGS.rb`
  - It’s empty a lot of the time.



# `cat sample/test.rb`

```
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)
```



# make btest-ruby

## btest-ruby invokes test files with the ruby binary and `bootstraptest/runner.rb` under the `bootstraptest` directory.

```
% 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 )
```



# make test-all

test-all invokes test files under the `test` directory.

These test files contain core libraries like String and Array and stdlib like Webrick and Logger. This task is a good one for a typical contributor.

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/-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)
```



# 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)
```



# make check

make check depends on the following definitions:

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

make check runs all test tasks in CRuby



# make testframework-test/test-almost

I separated test files testunit and minitest from test-all.

Why does CRuby have test files for testunit and minitest?

- CRuby Forked test-unit and minitest
- CRuby added parallel execution function to test-unit

We need to invoke to test to test-framework before CAPI, core library and standard library.

test-almost invokes tests under `test` without test-unit and minitest.



# test-unit/minitest

# Why separated the test framework?

**The following libraries uses minitest directly in Ruby 2.3:**

- **rubygems**
- **rdoc**
- **net-smtp (It seems unnecessary)**

**Other libraries uses test-unit. rubygems and rdoc are developed at github.com/rubygems/rubygems and github.com/rdoc/rdoc. We need to support  these libraries and their tests.**



# How to merge upstream from others

## I merged upstream into ruby/ruby periodically using following instructions.

```
$ git clone https://github.com/ruby/ruby
$ git clone https://github.com/rubygems/rubygems
$ cd ruby
$ rm -rf lib/rubygems test/rubygtems lib/rubygems.rb
$ cp -rf ../../rubygems/rubygems/lib/rubygems ./lib
$ cp -rf ../../rubygems/rubygems/lib/rubygems.rb ./lib
$ cp -rf ../../rubygems/rubygems/test/rubygems ./test
$ git checkout lib/rubygems/LICENSE.txt
```

## ruby and rubygems guarantee to work to test and code each other. it's the same situation for ruby and rdoc



# backporting is hard

## rubygems and rdoc still support Ruby 1.8.

```
% g show a34fb569e41cd87866e644d92a9df4be89b3cad2 test/rubygems/test_gem_package.rb
commit a34fb569e41cd87866e644d92a9df4be89b3cad2
Author: Eric Hodel <drbrain@segment7.net>
Date:   Tue Jul 8 16:53:50 2014 -0700

    Fix tests on ruby 1.8

diff --git test/rubygems/test_gem_package.rb test/rubygems/test_gem_package.rb
index f07c083..128dcdb 100644
--- test/rubygems/test_gem_package.rb
+++ test/rubygems/test_gem_package.rb
@@ -638,7 +638,7 @@ class TestGemPackage < Gem::Package::TarTestCase
                   e.message
       io
     end
-    tf.close!
+    tf.close! if tf.respond_to? :close!
   end

   def test_verify_empty
```



# Forked code for Test::Unit

- test/lib/envutil.rb
  - some assertion and function for language testing
- leakchecker.rb
  - checker for memory and fd leak.
- test/lib/test/lib/parallel.rb
  - helper library parallel execution for test-case



# rubyspec

# RubySpec

Q. What’s rubyspec?
A. RubySpec is an executable specification for the Ruby programming language.

“Matz's Ruby Developers Don't Use RubySpec and It's Hurting Ruby”

http://rubini.us/2014/12/31/matz-s-ruby-developers-don-t-use-rubyspec/

rubyspec is not a “specification”. It’s actually a set of “test”. The only real ruby specification is inside of Matz :)



# 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

```
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)
```



# rubyspec and mspec

We approved new or updated examples at github.com/ruby/rubyspec.

@headius wrote: “So nice to see RubySpec getting a steady stream of Ruby 2.3 specs.”
https://twitter.com/headius/status/667793518098673664

A lot of contributors submitted new specs for Ruby 2.3 features.



# rubyci

# rubyci and chkbuild

Ruby CI is a CI results collector for alternative platforms:

https://github.com/nurse/rubyci

ruby ci uses chkbuild built by akr:

https://github.com/akr/chkbuild



| Server | Datetime | Branch | Option | Revision | test | test-all | rubyspec | Summary | Diff |
|---|---|---|---|---|---|---|---|---|---|
| Debian 7.5 i686 | 12-04 18:04 | trunk | | 52882 | | | | 410W success | no diff |
| Debian 7.9 x86_64 | 12-04 12:00 | trunk | | 52877 | | 1F0E | | 420W 1F0E | diff:src,test- |
| Debian 8.2 x86_64 | 12-04 15:30 | trunk | | 52880 | | | | 414W success | diff:src,make |
| Debian 9.0(testing) x86_64 | 12-04 15:30 | trunk | | 52880 | | | | 409W success | diff:src,proc |
| Ubuntu 10.04 32-bit | 12-04 17:43 | trunk | | 52882 | | | | 408W success | diff:src,minir |
| Ubuntu 10.04 64-bit | 12-04 17:03 | trunk | | 52882 | | | | 408W success | diff:src,minir |
| Ubuntu 14.04 x86_64 | 12-04 17:00 | trunk | | 52882 | | | | 404W success | diff:src,make |
| Ubuntu 14.04 x86_64n | 12-04 18:14 | trunk | | 52882 | | | | 399W success | diff:test-all |
| Ubuntu armv7l eabihf | 12-04 16:17 | trunk | | 52880 | | | | 407W success | diff:src,make |
| Ubuntu 14.04 aarch64 | 12-04 13:54 | trunk | | 52878 | | 1F0E | 1F0E | 407W 1F0E rubyspec:1F0E | diff:src,proc |
| Ubuntu 15.10 x86_64 | 12-04 15:30 | trunk | | 52880 | | | | 414W success | diff:src,proc |
| Ubuntu 16.04(dev) x86_64 | 12-04 15:30 | trunk | | 52880 | | 2F0E | | 400W 2F0E | diff:src,proc |
| Gentoo | 12-04 15:30 | trunk | | 52880 | | | | 405W success | diff:src,make |
| CentOS 5 i386 | 12-04 17:03 | trunk | | 52882 | | | | 419W success | diff:src,minir |
| CentOS 5 x86_64 | 12-04 16:33 | trunk | | 52881 | | | | 432W success | diff:src,make |
| CentOS 6 64bit | 12-04 18:02 | trunk | | 52882 | | | | 429W success | diff:src,minir |
| CentOS 7.1 x86_64 | 12-04 14:00 | trunk | | 52878 | | | | 415W success | diff:src,test- |
| Fedora 19 ppc64 | 12-04 18:03 | trunk | | 52882 | | | | 413W success | diff:src,make |
| Fedora 21 x86_64 | 12-04 15:15 | trunk | | 52879 | | | | 407W success | diff:src,test- |
| Fedora 22 x86_64 | 12-04 17:15 | trunk | | 52882 | | | | 416W success | diff:src,make |
| Amazon Linux 2015.09 x64 | 12-04 15:30 | trunk | | 52880 | | | | 408W success | diff:src,proc |
| Arch Linux | 12-04 15:30 | trunk | | 52880 | | | | 420W success | diff:src,proc |
| openSUSE 13.2 | 12-04 15:30 | trunk | | 52880 | | | | 408W success | diff:src,make |
| RHEL 7.1 s390x | 12-04 09:33 | trunk | | 52872 | | 1F0E | | 414W 1F0E | diff:test-all |
| FreeBSD 10.1 x86 | 12-04 18:03 | trunk | | 52882 | | | | 408W success | diff:src,btest |
| FreeBSD 10.1 x64 | 12-04 18:03 | trunk | | 52882 | | | | 407W success | diff:src,test- |
| OS X Yosemite | 12-04 16:45 | trunk | | 52881 | | | | 407W success | diff:src,make |
| icc16-x64 | 12-04 12:00 | trunk | | 52877 | | 90F0E | 7F0E | 3795W 90F0E rubyspec:7F0E | diff:src,confi |
| Solaris 10x | 12-04 16:48 | trunk | | 52882 | | failed | 5F0E | 586W 2[BUG] 2[SEGV] failed(test-all) fail… | no diff |
| Solaris 11x | 12-04 16:11 | trunk | | 52880 | | | | 470W success | diff:src,pflag |
| Solaris 10s | 12-04 11:03 | trunk | | 52872 | | failed | | 597313W failed(test-all CommandTimeo… | no diff |
| Solaris 11s | 12-04 16:25 | trunk | | 52880 | | | | 472W success | diff:src,pflag |
| vc10-x64 | 12-04 17:15 | trunk | | 52882 | | | | 310W success | no diff |

# How to add a new server

You can add your server to rubyci.org

Requirements:

- not yet supported platforms.
  - ex. linux with ARM, *BSD, icc with OSX, Windows
- periodically running every day
- It must be possible to access to AWS S3

You should check the following commands on your server

```
$ git clone https://github.com/akr/chkbuild
$ cd chkbuild
$ ruby start-build
```



# appendix

# make run/bisect

`make run` invokes the `test.rb` file on ruby source directory. ko1 said this task helped with YARV development.

`make bisect` invokes `make run` with git-bisect. It helps detect commits containing defects.

but it’s only useful for a single ruby file. we need to invoke git-bisect under the bundler environment. that’s very difficult.



# test coverage

## I added a coverage task using simplecov

```
% make update-coverage
updating simplecov ...
remote: Counting objects: 90, done.
(snip)
updating simplecov-html ...
updating doclie …
% COVERAGE=1 make test-all TESTS=webrick
```

**You can get coverage results for `webrick` under the coverage directory.**



```
25.     # The ERB file is evaluated with the local variables +servlet_request+ and
26.     # +servlet_response+ which are a WEBrick::HTTPRequest and
27.     # WEBrick::HTTPResponse respectively.
28.     #
29.     # Example .rhtml file:
30.     #
31.     #   Request to <%= servlet_request.request_uri %>
32.     #
33.     #   Query params <%= servlet_request.query.inspect %>
34.
35.     class ERBHandler < AbstractServlet
36.
37.       ##
38.       # Creates a new ERBHandler on +server+ that will evaluate and serve the
39.       # ERB file +name+
40.
41.       def initialize(server, name)
42.         super(server, name)
43.         @script_filename = name
44.       end
45.
46.       ##
47.       # Handles GET requests
48.
49.       def do_GET(req, res)
50.         unless defined?(ERB)
51.           @logger.warn "#{self.class}: ERB not defined."
52.           raise HTTPStatus::Forbidden, "ERBHandler cannot work."
53.         end
54.         begin
55.           data = open(@script_filename){|io| io.read }
56.           res.body = evaluate(ERB.new(data), req, res)
57.           res['content-type'] ||=
58.             HTTPUtils::mime_type(@script_filename, @config[:MimeTypes])
59.         rescue StandardError
60.           raise
61.         rescue Exception => ex
62.           @logger.error(ex)
63.           raise HTTPStatus::InternalServerError, ex.message
64.         end
65.       end
66.
67.       ##
68.       # Handles POST requests
69.
70.       alias do_POST do_GET
71.
72.       private
73.
74.       ##
75.       # Evaluates +erb+ providing +servlet_request+ and +servlet_response+ as
76.       # local variables.
```

# Limitation of test coverage

## We have some defects related to the “Coverage” library.

```
% COVERAGE=1 make test-all
       CC = clang
(snip)
Run options: "--ruby=./miniruby -I./lib -I. -I.ext/common  ./tool/runruby.rb --extout=.ext
-- --disable-gems" --excludes=./test/excludes -x /memory_leak/

# Running tests:

[ 3491/15951] TestCoverage#test_big_code = 0.17 s
  1) Failure:
TestCoverage#test_big_code [/path/to/ruby/test/lib/tracepointchecker.rb:18]:
The number of active trace events was changed.
<[[#<RubyVM:0x000001017c3588>, 1]]> expected but was
<[[#<RubyVM:0x000001017c3588>, 0]]>.

/path/to/ruby/test/lib/leakchecker.rb:116: [BUG] Segmentation fault at
0x00000000000000
ruby 2.3.0dev (2015-12-03 trunk 52872) [x86_64-darwin15]
```



# Conclusion

# Plan for Ruby 2.4/3.0

- Restructured test directories and files
  - Separated test focus
  - Removed duplicate tests
  - Simplify test tasks
- stdlib tests more friendly with JRuby
- Increase coverage
- Integrate rubyspec and ruby tests



# Please contribute tests to ruby

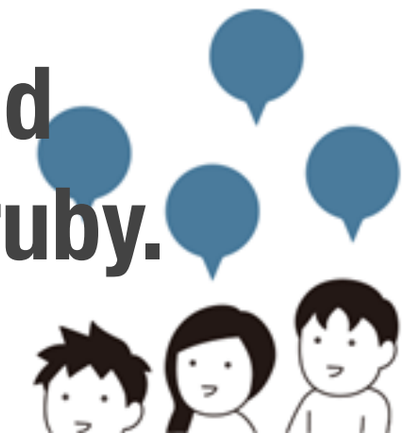
## You can invoke CRuby tests:

```
$ 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 testing is so easy