

Date: 2018/12/12 (Thu)
Time: 13:30-17:30 (JST)
Place: pixiv (Tokyo, Japan)
<https://bugs.ruby-lang.org/issues/>

Logs

Next dev-meeting

- 2019/01/10 (Thu) 13:00-17:00
Cookpad (Ebisu, Tokyo, Japan)
- after this meeting, let's party without Matz

About 2.6 timeframe

- RC1 12/6
- RC2 12/12~18?(for downgrade Bundler for Heroku breakage)

Check security tickets (confidential)

Carry-over from previous meeting(s)

- It seems no carry-over issue was there.

From Attendees

- [Bug [#15303](#)] Return tracepoint doesn't fire when tailcall optimization is applied (ko1)
 - tailcall skips return event and it breaks a debugger's logic. there are several ideas, and we need to introduce a solution for ruby 2.6.
 - (1) Introduce `tailcall?`
 - (2) Introduce `return` event before tailcall
 - (3) Cancel tailcall optimization
 - (4) Don't use tailcall opt on forwardable.rb
 - (5) Delete tailcall feature
 - -> (4)
- [Feature [#15287](#)] New TracePoint events to support loading features (ko1)
 - Ruby 2.6
 - I already introduced `script_compiled` TracePoint event. We need to decide related methods.
 - Matz: remove "compiled_" prefix.
- Should hash separation be prohibited? (mame)
 - Ruby 2.6

```
def foo(h = {}, **kw)
  p [h, kw]
end
foo("str" => 1, :sym => 2)
```

- Was prohibited in 2.6. Is it really okay?
 - Akr: open3.rb has problem

```
% ./ruby -ropen3 -e 'p Open3.capture2("echo foo", :in => IO::NULL, 3 => IO::NULL)'
```

 Traceback (most recent call last):
 -e:1:in `': non-symbol key in keyword arguments: 3 (ArgumentError)
 - Akr: open3.rb will be changed to avoid **. r66352
- Obsolete taint flags (naruse)
 - Rails doesn't use it
 - We should (re-)announce taint should not be considered as security feature. By now it is only left as fail-proof mechanism.
 - Akr: It has always been a problem with taint for library developers that there is no way to tell what could be safe or not for the caller. For example, open-uri fails when \$SAFE is on and it tries to handle redirection because the header values are always tainted. Users tend to ask me to internally untaint it because open-uri is almost useless without doing it, but that could risk other users depending on taint/\$SAFE as security feature.

Non-attendees

- [Feature [#15230](#)] RubyVM.resolve_feature_path
 - 2.6 issue
 - Matz: no fix on 2.6. Consider on next version.
- [Feature [#13581](#)] Syntax sugar for method reference
 - 2.7 or later
 - knu: Introducing “...” (as in Lua) would allow for writing this way: ary.each { puts(...) }
 - Matz: Allowing omission of “self” sounds like a bad idea because that makes each(&:puts) and each(&.:puts) look very similar but act so differently.
- [Feature [#10771](#)] An easy way to get the source location of a constant
 - 2.7 or later
 - Matz: OK, but wait for 2.7.
- [Feature [#15373](#)] Proposal: Enable refinements to #method and #instance_method
 - 2.7 or later
 - Matz: ok to introduce.
- [Feature [#15374](#)] Proposal: Enable refinements to #method_missing
 - 2.7 or later
 - Matz: let me think about it for a while.
- [Feature [#15007](#)] Proposal: Introduce support for cold function attributes (clang and GCC)
 - reduced scope PR <https://github.com/ruby/ruby/pull/2005> (rb_memerror, rb_bug, rb_warn)

- Already merged.
- [Bug [#15394](#)] Ruby adds unexpected HTTP header value when using symbol key
 - This is a small change I think, but could make codes less buggy
 - Already solved.
- [Feature [#15066](#)] Documentation and providing better API for accessing Complex numbers functions in C extensions

Carry over:

-