**Ruby master - Misc #16454**

**DevelopersMeeting20200116Japan**

12/26/2019 07:45 AM - mame (Yusuke Endoh)

<table>
<thead>
<tr>
<th>Status:</th>
<th>Closed</th>
</tr>
</thead>
<tbody>
<tr>
<td>Priority:</td>
<td>Normal</td>
</tr>
<tr>
<td>Assignee:</td>
<td></td>
</tr>
</tbody>
</table>

**Description**

**The next dev meeting**

Date: 2020/01/16 13:00-17:00

Place/Sign-up/Agenda/Log: [https://docs.google.com/document/d/1NxNMc7tt-IoPRyRQDhA1ZWw30Atr6a-rfQxQ9BRXY](https://docs.google.com/document/d/1NxNMc7tt-IoPRyRQDhA1ZWw30Atr6a-rfQxQ9BRXY)

- Dev meeting IS NOT a decision-making place. All decisions should be done at the bug tracker.
- Dev meeting is a place we can ask Matz, nobu, nurse and other developers directly.
- Matz is a very busy person. Take this opportunity to ask him. If you can not attend, other attendees can ask instead of you (if attendees can understand your issue).
- We will write a log about the discussion to a file or to each ticket in English.
- All activities are best-effort (keep in mind that most of us are volunteer developers).
- The date, time and place are scheduled according to when/where we can reserve Matz's time.
- **DO NOT** discuss then on this ticket, please.

**Call for agenda items**

If you have a ticket that you want matz and committers to discuss, please post it into this ticket in the following format:

* [Ticket ref] Ticket title (your name)
  * Comment (A summary of the ticket, why you put this ticket here, what point should be discussed , etc.)

Example:

* [Feature #14609] `Kernel#p` without args shows the receiver (ko1)
  * I feel this feature is very useful and some people say :+1: so let discuss this feature.

- Comment deadline: 2020/01/09 (one week before the meeting)
- The format is strict. We'll use this script to automatically create an markdown-style agenda. We may ignore a comment that does not follow the format.
- Your comment is mandatory. We cannot read all discussion of the ticket in a limited time.

**Related issues:**

Related to Ruby master - Misc #14770: [META] DevelopersMeeting

**History**

#1 - 12/26/2019 07:45 AM - mame (Yusuke Endoh)
- Related to Misc #14770: [META] DevelopersMeeting added

#2 - 12/26/2019 07:47 AM - mame (Yusuke Endoh)
- Description updated

#3 - 12/26/2019 07:50 AM - mame (Yusuke Endoh)

I'd like to set the comment deadline for the agenda. Please add a ticket number by one week ago about the meeting date. (I'll go through the tickets before the meeting.)

#4 - 12/26/2019 07:52 AM - mame (Yusuke Endoh)

Carry-over:

- [Bug #8841] Module#included_modules and prepended modules (mame)
  * Module#include? and Module#included_modules regard prepended modules as included (not well documented); Module#included is not
called when the module is prepended. Is this right?
  
  * IMO, changing the behavior is no longer acceptable (without any actual trouble). How about just changing the document?

- [Feature #8026] Need Module#prepended_modules (mame)
  
  * It is accepted six years ago, but not implemented yet. I've never heard any actual trouble, but should we still add the feature?

- [Bug #9815] attr_reader doesn't warn on a uninitialized instance variable (mame)
  
  * A reader method defined by attr_reader :foo is not warned as "instance variable @foo not initialized". Is it intentional?

- [Bug #10388] Operator precedence problem in multiple assignment (massign) (mame)
  
  * "a, b = c = 1, 2 is currently taken as a, b = (c = 1), 2; I'd expect it to be taken as a, b = (c = 1, 2)." Jeremy gave a try to implement but seemed difficult due to the limitation of LALR(1) parser. Let's give up.

- [Bug #10475] Array#flatten should not accept a nil argument (mame)
  
  * Should we add a document that Array#flatten accepts nil? Negative argument too?

- [Bug #10929] NilClass#to_proc and & don't mix? (mame)
  
  * I think it is not worth adding.

- [Bug #11014] String#partition doesn't return correct result on zero-width match (mame)
  
  * I'd like to confirm if the current behavior is intended.

- [Feature #16432] Using _1 inside binding.irm will cause unintended behavior (osyo)
  
  * Calling binding.irm in a block that uses _1 and using _1 in irb will cause unintended behavior.
  * Should it be a runtime error?

#5 - 12/26/2019 08:21 AM - duerst (Martin Dürst)
  
  Description updated

#6 - 12/26/2019 08:23 AM - duerst (Martin Dürst)
  
  mame (Yusuke Endoh) wrote:

  Please add a ticket number by one week ago about the meeting date.

Do you mean "by one week before the meeting date"? In English, "ago" can only be used for the past, not for the future.

#7 - 12/26/2019 09:23 AM - zverok (Victor Shepelev)

(I honestly promised to stop doing this...)

- [Feature #16441] Enumerable#take_while_after (zverok)
  
  * Just like take_while, but also returns the matched element; practical examples are shown

- [Feature #16435] Array#to_proc (zverok)
  
  * Make map{[:,foo]} it just a shortcut for map { [hash] hash[:foo] }. Ambitious, but justified. Two alternative approaches proposed (dig and just [])

#8 - 12/26/2019 10:33 AM - mame (Yusuke Endoh)
  
  duerst (Martin Dürst) wrote:

  mame (Yusuke Endoh) wrote:

  Please add a ticket number by one week ago about the meeting date.

Do you mean "by one week before the meeting date"? In English, "ago" can only be used for the past, not for the future.
Yes, I meant 2020/01/09. Thanks :-)

#9 - 01/04/2020 04:26 AM - jeremyevans0 (Jeremy Evans)
- [Bug #16383] TracePoint does not report calls to attribute reader methods (jeremyevans0)
  - Do we want to support this?
- [Feature #5391] Introducing Numeric#exact? and Numeric#inexact? (jeremyevans0)
  - Can we add Numeric#exact? ? This is necessary to fix Bug #5179.
- [Bug #16790] Zlib::GzipReader only decompresed the first of concatenated files (jeremyevans0)
  - Can we implement Zlib::GzipReader.each_file using the patch?

#10 - 01/06/2020 12:48 AM - mame (Yusuke Endoh)
- Description updated

#11 - 01/06/2020 11:27 PM - mame (Yusuke Endoh)
- [Bug #16468] Hash.ruby2_keywords?(hash) and Hash.ruby2_keywords!(hash) (mame)
  - It turned out that we sometimes need to deal with ruby2_keywords flag explicitly. I'd like to add the feature as class methods. (naruse [Yui NARUSE] says that backport to 2.7.1 is okay)

#12 - 01/08/2020 12:59 PM - Eregon (Benoit Daloze)
- [Feature #16463] Fixing *args-delegation in Ruby 2.7: ruby2_keyswords semantics by default in 2.7.1 (eregon)
  - This would be a way to make transition to Ruby 2.7 a lot simpler. Advantages are: easier migration (no explicit ruby2_keyswords, no confusing warnings, useful delegation warnings when migrating), better debugging, more compatible.
  - Right now it's confusing warnings, forcing most gems to use ruby2_keyswords which is just a workaround and the code will need to change again.
  - What do you think?
    - I think if we don't do it, it will delay Ruby 2.7 adoption very significantly. I'd also think Rubyists will get a very negative opinion about the keyword arguments change due to ruby2_keyswords and delegation issues, when there is no need and we can make it easier.
    - The only drawback (sometimes treating kwargs as kwargs instead of positional, very rare) can easily be worked around by using foo(1, {a: 1}) instead of foo(1, a: 1).

#13 - 01/08/2020 10:47 PM - alanwu (Alan Wu)
- [Misc #16467] Potential for SIMD usage in ruby-core (alanwu)
  - It would be great if we can come up with a stance on this and document it. It doesn't have to be permanent; we can revise it later, but I think the policy can be very helpful to new contributors.
  - Related, I think we can put something in the Github pull request template to help guide new contributors. I think Sorbet's repo uses this to good effects.
- [Feature #16461] Proc#using (alanwu)
  - This looks like a serious change worth discussing.

#14 - 01/08/2020 11:20 PM - mame (Yusuke Endoh)

alanwu (Alan Wu) wrote:

- [Misc #16487] Potential for SIMD usage in ruby-core (alanwu)
  - It would be great if we can come up with a stance on this and document it. It doesn't have to be permanent; we can revise it later, but I think the policy can be very helpful to new contributors.
  - Related, I think we can put something in the Github pull request template to help guide new contributors. I think Sorbet's repo uses this to good effects.

I'm unsure what topic should be discussed. Do you want us to discuss not the issue itself but a policy for feature request? I think it would be good to create another ticket for that; and a draft would be very helpful.

#15 - 01/09/2020 10:14 AM - zverok (Victor Shepelev)

- [Feature #16494] Allow hash unpacking in non-lambda Proc
  - allow map { [foo, bar] ... } just like array unpacking to map { [foo, bar] ... } is allowed. The ticket contains pragmatic examples.

#16 - 01/09/2020 01:38 PM - Eregon (Benoit Daloze)

mame (Yusuke Endoh) Here is an updated version of the script which also works if there is no (author) at the end of a comment:
https://gist.github.com/eregon/b4aa78dca42e4204e3a276c92205ade

This seems a common mistake, both zverok (in comment 15) and I made it.
Could you adopt it?

02/23/2021
Okay updated.

mame (Yusuke Endoh) wrote:

alanwu (Alan Wu) wrote:

- [Misc #16487] Potential for SIMD usage in ruby-core (alanwu)
  - It would be great if we can come up with a stance on this and document it. It doesn’t have to be permanent; we can revise it later, but I think the policy can be very helpful to new contributors.
  - Related, I think we can put something in the Github pull request template to help guide new contributors. I think Sorbet's repo uses this to good effects.

I'm unsure what topic should be discussed. Do you want us to discuss not the issue itself but a policy for feature request? I think it would be good to create another ticket for that; and a draft would be very helpful.

I would like to see discussion about [Misc #16487] itself. My notes are a bit of an aside, but I think the outcome of #16487 will set a precedent at least. I can make tickets about additional documentations later if appropriate.

I see. I'd appreciate if you write your (non-meta) opinion about that ticket.

Note that we have only about a few minutes to discuss each ticket. (The meeting is about 200 minutes long, and we have about 20 tickets on the agenda.) We can discuss only one topic for each ticket, so I'd like you to make it clear the point that you want us to discuss. And usually, I look over all tickets in advance of the meeting and explain them to the attendees; your short summary is very useful for me to understand each ticket.

Thank you for your cooperation.

[Feature #16495] Inconsistant Quotes in Error Messages (mame)
- This is the third time that we receive a ticket about the backtick and quote in error messages: #12321 for inconvenience in Slack. #13589 for weird appearance and inconvenience in syntax highlighting, this ticket for weird appearance.
- I know that it is an old-school custom and think that it is hard to change, but I'd like just to hear matz's feelings.

[Feature #8709] Dir.glob should return sorted file list (eregon)
- It causes non-determinism on e.g., Linux, which causes complex bugs (which are not worth investigating)
- Many other languages seem to sort by default (C's glob(3), bash, perl, gmake, ...)
- TruffleRuby already sorts Dir.glob to avoid this issue
- Quite a few people supporting this idea; https://twitter.com/eregontp/status/1217036685059461120

[Feature #16484] Remove xmlrpc and net-telnet from bundled gems
- Does anyone have an objection?

[Feature #15973] Let Kernel#lambda always return a lambda (ko1)
- How about to prohibit passing Proc for lambda to solve this issue completely?
- Or how about to obsolete lambda because we already have ->?

So the following issues are optional (candidates for carry over):

- [Feature #16499] define_method(non_lambda) should not the semantics of the given Proc (ko1)
I have no strong opinion, but we need to check as related issue of [#15973].

#24 - 01/15/2020 05:54 AM - mame (Yusuke Endoh)
I'm sorry but let me insert my two proposals since they should be determined before 2.7.1.

* [Feature #16501] Support marshaling of ruby2_keywords flag
  * Marshal.dump and load do not support ruby2_keywords flag, so serializing and desalinizing a flagged Hash drops the flag.
  * This is not critical because AFAIK there is no real application that requires this feature. It would be helpful to make drb support the separation, but if [#16486] is accepted, drb can work around the issue.
  * But if we introduce it, it should be into 2.7.1.

* [Feature #16466] *args -> *args delegation should be warned when the last hash has a ruby2_keywords flag
  * In 2.7.0, no warnings are displayed at second and later stages of multi-stage delegation. It looks a bug.
  * I think it is not critical, but in principle, it should be warned. I'd like to determine it before 2.7.1.

#25 - 01/16/2020 04:43 AM - Dan0042 (Daniel DeLorme)

* [Feature #16511] Subclass of Hash for keyword arguments
  * I'm too late (the dev meeting has started!) but it would be nice for the discussion on [#16463] and [#16494] to be aware this alternative exists

#26 - 01/17/2020 05:35 AM - sawa (Tsuyoshi Sawada)
The linked log [https://docs.google.com/document/d/1NxNMe7Itt-o1PRyRQDhAIZWw30Atr6a-rfQxQ9BRXY](https://docs.google.com/document/d/1NxNMe7Itt-o1PRyRQDhAIZWw30Atr6a-rfQxQ9BRXY) is not updated. Can I ask for it to be updated?

#27 - 01/17/2020 05:51 AM - mame (Yusuke Endoh)
- Status changed from Open to Closed

Published. Closing.