Ruby master - Misc #15874
DevelopersMeeting20190613Japan
05/24/2019 03:58 PM - 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

Please comment your favorite ticket numbers you want to ask to discuss with your SHORT comment or summary. (your summary/comment will help us because we don't need to read all of ticket comments)

DO NOT discuss then on this ticket, please.

Date: 2019/06/13 (Thu)
Time: 14:00-17:30 (JST)
Place and Sign-up: [https://ruby.connpass.com/event/132888/](https://ruby.connpass.com/event/132888/)
log: [https://docs.google.com/document/d/1XP_e-3uXICaxZLJrov6c426ZNaMPmx3eh9MASrzkl/edit#](https://docs.google.com/document/d/1XP_e-3uXICaxZLJrov6c426ZNaMPmx3eh9MASrzkl/edit#)

NOTES

- 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 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 is scheduled according to when/where we can reserve Matz's time.

Agenda

Next dev-meeting

About 2.7 timeframe

Check security tickets

Carry-over from previous meeting(s)

None.

From Attendees

From non-attendees

Please comment your favorite ticket we need to discuss with the following format.

* [Ticket ref] Ticket title (your name)
  * your comment why you want to put this ticket here if you want to add.

Your comment is very important if you are no attendee because we can not ask why you want to discuss about it.

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 about this feature.

We don't guarantee to put tickets in agenda if the comment violate the format (because it is hard to copy&paste).
Related issues:
Related to Ruby master - Misc #14770: [META] DevelopersMeeting

Open

History

#1 - 05/24/2019 04:03 PM - mame (Yusuke Endoh)
- Description updated

#2 - 05/24/2019 04:04 PM - mame (Yusuke Endoh)
- Related to Misc #14770: [META] DevelopersMeeting added

#3 - 05/24/2019 08:39 PM - alanwu (Alan Wu)
- [Bug #15792] GC can leave strings used as hash keys in a corrupted state / bad fstring + shared string interaction can lead to use-after-free
  - I have a PR https://github.com/ruby/ruby/pull/2183 for a case of this with String#b that I think we should also fix.
  - Could someone mark the ticket for backporting to all supported versions? It affects all supported versions as wanabe-san showed in the thread.

#4 - 05/25/2019 12:13 AM - nobu (Nobuyoshi Nakada)
- [Feature #14111] ArgumentErrorが発生した時メソッドのプロトタイプをメッセージに含む
  - Include the method prototype in the message when ArgumentError occurred
- [Feature #14145] Proposal: Better Method#inspect
- [Feature #15799] pipeline operator

#5 - 05/27/2019 01:00 AM - sikachu (Prem Sichanugrist)
- [Feature #15725] Add Array#reverse_sort, #revert_sort!, #reverse_sort_by, and #reverse_sort_by! I would like to propose these methods as the preferred way for developers to create reverse sorted array, which is to call sort(_by).reverse! internally. By introduce these methods, we can discourage user to use them instead of do array.sort.reverse (without bang) or array.sort { |a, b| b <=> a } which would create an unnecessary Array (after `.sort`).

#6 - 05/27/2019 09:53 AM - tagomoris (Satoshi TAGOMORI)
- [Bug #15733] Inconsistent __FILE__ and Kernel#__dir__

#7 - 05/28/2019 09:54 AM - byroot (Jean Boussier)
- [Feature #15777] Module#autoload?(cname, inherit=true)
  - By default autoload? also check in the ancestors chain, such option would be consistent with Module#const_defined? and totally backward compatible.
  - For more context this would be needed in Zeitwerk: https://github.com/fxn/zeitwerk/pull/17
- [Proposal #15836] Make Module#name and / or Symbol#to_s return their internal fstring.
  - I realize this is unlikely to be accepted because it has a very significant backward compatibility impact, but I'll still make a case for it.
  - It's surprising for both of these method to allocate new strings when intuitively we're accessing some static metadata.
  - These two method are extensively used in metaprogramming code and other DSL, they are responsible for a significant amount of allocations and duplications.
  - Ideally to address the backward compatibility issues, the behavior would only change if frozen_string_literal: true is set, but I'm unsure this is actually possible to implement.

#8 - 05/30/2019 10:31 PM - Eregon (Benoit Daloze)
- [Misc #15723] Reconsider numbered parameters.
  - To matz: Could you agree on Jeremy's single-argument patch?
#9 - 06/01/2019 01:36 AM - nobu (Nobuyoshi Nakada)

- [Feature #15160] ArgumentError: year too big to marshal

#10 - 06/06/2019 01:27 AM - nobu (Nobuyoshi Nakada)

- [Bug #15210] UTF-8 BOM should be removed from String in internal representation
  
  A method to set the encoding on an IO by BOM?

#11 - 06/06/2019 09:27 AM - Eregon (Benoit Daloze)

- [Feature #15903] Move RubyVM.resolve_feature_path to Kernel.resolve_feature_path
  
  matz: is it OK to add Kernel.resolve_feature_path (only class method)? I think RubyVM is not a good place for resolve_feature_path which is not MRI-specific. Moving the method is necessary for other Ruby implementations to implement it, and keep the clean separation that RubyVM is only defined on MRI.

#12 - 06/08/2019 12:46 PM - nobu (Nobuyoshi Nakada)

- [Bug #15908] Detecting BOM with non-UTF encoding

#13 - 06/08/2019 12:58 PM - pitr.ch (Petr Chalupa)

- [Feature #14912] Introduce pattern matching syntax (pitr.ch)
  
  Could the pattern matching be made available as a first-class citizen to be used as a filter when searching in data structures and to be able to implement Actor receive method
  
  Consider contractions log_messages.find in [:error, message] { puts message }
  
  Non-symbol matching of Hashes would be very desirable
  
  Elaborated in https://bugs.ruby-lang.org/issues/14912#note-22

#14 - 06/11/2019 12:10 AM - mame (Yusuke Endoh)

- Description updated

#15 - 06/11/2019 05:45 PM - jeremyevans0 (Jeremy Evans)

- [Feature #15797] Use realpath(3) instead of custom realpath implementation if available (jeremyevans0)
  
  Is this OK to commit? It may cause regressions on less common Unix not tested by Travis (e.g. FreeBSD, NetBSD, AIX, Solaris), though we could just fallback to current implementation in that case.
  
  Do we want to add workarounds for Mac OS <=10.5 (10.5 was released in October 2007)?

#16 - 06/12/2019 06:22 AM - mame (Yusuke Endoh)

- [Feature #15897] it as a default block parameter
  
  A keyword it has never been seriously considered because adding a new keyword brings a big incompatibility.
  
  However, I'd like to show that it is possible to add it as a keyword with much smaller impact than we thought.

#17 - 06/12/2019 01:20 PM - k0kubun (Takashi Kokubun)

- [Feature #15563] #dig that throws an exception if an key doesn't exist
  
  Previously Matz said #dig is a bad name and asked a real-world use case.
  
  How about #deep_fetch for the name? See 15563#note-6 for the real-world use case.
  
  I often use Ruby for configuration management, and I've seen the real-world use case at least 2 years ago, 5 months ago, and today.

#18 - 06/17/2019 03:15 AM - mame (Yusuke Endoh)

- Status changed from Open to Closed