Ruby - Misc #19684

DevMeeting-2023-06-08

05/22/2023 03:33 AM - mame (Yusuke Endoh)

Status:	Closed	
Priority:	Normal	
Assignee:		

Description

The next dev meeting

Date: 2023/06/08 13:00-17:00 (JST)
Log: https://github.com/ruby/dev-meeting-log/blob/master/2023/DevMeeting-2023-06-08.md

- 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 record of the discussion in the 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 of the meeting 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.
```

- It is recommended to add a comment by 2023/06/05. We hold a preparatory meeting to create an agenda a few days before the dev-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. We appreciate it if you could write a short summary and update from a previous discussion.

Related issues:

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

History

#1 - 05/22/2023 03:33 AM - mame (Yusuke Endoh)

- Related to Misc #14770: [META] DevelopersMeeting added

#2 - 05/22/2023 07:28 AM - byroot (Jean Boussier)

- [Bug #19681] The final classpath of partially named modules is sometimes inconsistent once permanently named (byroot)
  - Please see the code snippet in the ticket description.
  - The current implementation iterate over RCLASS_CONST_TBL and assign the first name it find.
  - But that table is unordered, so if the module was aliased, the result can be surprising.

#3 - 05/30/2023 10:55 PM - jeremyevans0 (Jeremy Evans)

- [Bug #11704] Refinements only get "used" once in loop (jeremyevans0)
  - Is it expected that if a refinement is activated for a block, it remains activated if the block is re-entered?
  - This is how CRuby works, but may be confusing to users expecting more dynamic behavior.

    - This would seem difficult to change in CRuby, though JRuby (and maybe TruffleRuby) already operates the more dynamic way.
- [Bug #18622] const_get still looks in Object, while lexical constant lookup no longer does (jeremyevans0)
    - Should we change Module#const_get to raise NameError instead of looking into Object, for modules and subclasses of Object?
    - If so, should we deprecate the behavior in Ruby 3.3 and change the behavior in Ruby 3.4?
- [Bug #15428] Refactor Proc#>> and #<< (jeremyevans0)
    - Last discussed at August 2021 developer meeting, no decision made.
    - Calling to_proc if the passed object does not respond to call should be backwards compatible.
    - Do we want to change the behavior of Proc#>> and Proc#<< to encourage their usage with symbols?
    - Alternatively, should Proc#>> and Proc#<< raise error if called with argument not supporting #call (to avoid NoMethodError later when you call the resulting Proc)?

#4 - 05/31/2023 09:11 AM - hsbt (Hiroshi SHIBATA)

- [Bug #19702] Promote racc as bundled gems
    - Does anyone have objection this?

#5 - 06/01/2023 12:55 AM - ioquatix (Samuel Williams)

- [Feature #19057] Hide implementation of rb_io_t.
    - ~~So far the agreed approach is to hide the entire implementation and fix dependencies.~~ We tried it and it caused some downstream issues.
    - Are we okay to deprecate the public struct to start with? Removal by what version?
    - PR: https://github.com/ruby/ruby/pull/7916
- [Bug #19717] ConditionVariable#signal is not fair when the wakeup is consistently spurious.
    - Can we discuss whether we should prevent spurious wakeup?
    - No PR yet.. can we discuss best approach?
- [Bug #18818] Thread waitq does not retain referenced objects, can lead to use after free.
    - Should we mark waitq and similar structures?
    - PR: https://github.com/ruby/ruby/pull/5979
- [Feature #19520] Support for Module.new(name) and Class.new(superclass, name).
    - Any thoughts on moving forward with this proposal?
    - PR: https://github.com/ruby/ruby/pull/7376

#6 - 06/08/2023 02:16 AM - yui-knk (Kaneko Yuichiro)

[Feature #19719] Universal Parser

- Want to discuss the direction of development (Design) and release management (Release management).

#7 - 06/09/2023 02:04 AM - hsbt (Hiroshi SHIBATA)

- Status changed from Open to Closed

#8 - 06/09/2023 08:36 AM - mame (Yusuke Endoh)

- Description updated