The next dev meeting

Date: 2022/02/17 13:00-17:00 (JST)

- 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 2022/02/14. 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 master - Misc #14770: [META] DevelopersMeeting Open
- Prevents incompatible binaries from being loaded, which can cause bugs, crashes, or other incorrect behavior.
- Summary of motivation here: https://bugs.ruby-lang.org/issues/18249#note-15

#4 - 02/02/2022 01:43 AM - hsbt (Hiroshi SHIBATA)

- [Feature #18513] Hide patchlevel from ruby -v
  - There is no reason to display it now.

#5 - 02/02/2022 08:32 AM - byroot (Jean Boussier)

- [Feature #18498] Introduce a public WeakKeysMap that compares by equality (byroot)
  - Ruby's current weak reference constructs are not very usable.
  - The most useful weak collection that is offered by most platform is WeakKeysMap
  - The keys would be weak references, but the values would be strong references
  - It would use equality semantic, just like a regular Hash

#6 - 02/03/2022 04:24 PM - headius (Charles Nutter)

- [Feature #18568] Explore lazy RubyGems boot to reduce need for --disable-gems
  - With more and more default gems, RubyGems has become a major part of Ruby's startup time.
  - On CRuby, the base boot time is reduced by 80% using --disable-gems.
  - Several ruby-core members would like to remove --disable-gems but others and some users need it for fast command line tools.
  - It seems like it is time to eliminate the overhead of booting RubyGems and default gemspecs.

#7 - 02/04/2022 11:46 AM - hsbt (Hiroshi SHIBATA)

- [Feature #18571] Removed the bundled sources from release package after Ruby 3.2
  - (dummy comment)

#8 - 02/07/2022 08:45 AM - byroot (Jean Boussier)

[Feature #18566] Merge io-wait gem into core IO class. (byroot)

- It's very small and any non-trivial IO code will require it.
- Merge io-wait into io.c for Ruby 3.2
- Remove io-wait as a dependency of all gems maintained by ruby-core (e.g. net-protocol).
- Publish a new io-wait version that is simply an empty gem.
- Add a lib/io/wait.rb stub, with eventually a deprecation warning.
- @normalperson (Eric Wong) suggest to do the same with io-nonblock.

#9 - 02/07/2022 06:22 PM - normalperson (Eric Wong)

"byroot (Jean Boussier)" noreply@ruby-lang.org wrote:

Issue #18557 has been updated by byroot (Jean Boussier).

[Feature #18566] Merge io-wait gem into core IO class. (byroot)

- It's very small and any non-trivial IO code will require it.
- Merge io-wait into io.c for Ruby 3.2
- Remove io-wait as a dependency of all gems maintained by ruby-core (e.g. net-protocol).
- Publish a new io-wait version that is simply an empty gem.
- Add a lib/io/wait.rb stub, with eventually a deprecation warning.

Please just keep lib/io/wait.rb stub empty indefinitely. No need for deprecation warning, just a no-op is fine and don't cause pain for existing users. I don't want to hurt:

1. users who aren't coders
2. users who lack permissions to change code
3. codebases which need to support both old/ancient Rubies + new ones

I'm looking at system-wide implications from distro-installed Rubies; not from a webapp+bundler perspective.

- @normalperson (Eric Wong) suggest to do the same with io-nonblock.

Actually, Benoit suggested io-nonblock. I suggested going beyond that :)

08/28/2022
#10 - 02/07/2022 10:31 PM - jeremyevans0 (Jeremy Evans)

- **[Bug #18487]** Kernel#binding behaves differently depending on implementation language of items on the stack (jeremyevans0)
  - Is it OK to limit Kernel#binding to only look at the preceding frame, instead of the closest Ruby-level frame?
  - If so, is the patch acceptable?
- **[Bug #18663]** Add block or filtered forms of Kernel#caller to allow early bail-out (jeremyevans0)
  - Since the last developer meeting, a prototype has been created.
  - Is the behavior of the prototype acceptable?
  - Incorrect usage with to_enum raises StopIteration instead of crashing. Is that acceptable?
- **[Bug #14103]** Regexp absence operator has no chance to ^C (jeremyevans0)
  - One possible way around this is checking for interrupts and potentially yield to another thread every N times during backtracking.
  - I submitted a pull request to do that (N = 1000), and it fixes the example given.
  - Is such an approach acceptable? If so, is the patch acceptable?
- One issue not addressed by the patch is the modification of the string being searched by another thread during regexp evaluation (potentially causing a crash), but I'm not sure whether that is possible.

#11 - 02/07/2022 10:52 PM - greggzst (Grzegorz Jakubiak)

- **[Feature #16989]** Sets: need ♥️
  - matz has already accepted adding Sets as default. However, there hasn't been any updates on the progress

#12 - 02/08/2022 06:57 PM - byroot (Jean Boussier)

- **[Feature #18339]** GVL instrumentation API
  - The goal is to be able to measure GVL contention and tell users if they use too many threads for their workload.
  - It would allow to instrument the GVL which is a key metric for threaded environments, and to tune concurrency in applications.
  - I implemented a (posix only for now) draft: [github.com/ruby/ruby/pull/5590](https://github.com/ruby/ruby/pull/5590).
  - The overhead when no hook is registered is just a single unprotected boolean check, so close to zero.
  - rb_gvl_hook_t * rb_gvl_event_new(rb_gvl_callback callback, rb_event_flag_t event) to register a hook.
  - bool rb_gvl_event_delete(rb_gvl_hook_t * hook) to unregister it.
  - Is the general idea acceptable?
  - Any API change before I try to implement the Windows version?

#13 - 02/09/2022 09:54 PM - tenderlovemaking (Aaron Patterson)

- **[Feature #12962]** Adding "friend" method visibility
  - Allow libraries with Object Oriented internals to call "not public" methods
  - Useful when "protected" is too restricted, but the method shouldn't be "public"
  - Refinements can't help because they are lexically scoped
  - It's easier to add method visibility than to move all code around

#14 - 02/10/2022 07:38 AM - zverok (Victor Shepelev)

- **[Bug #18580]** Range#include? inconsistency for beginless String ranges (zverok)
  - (..’z’).include?('wv') works for strings, due to historical reasons
  - there is no semantical justification for it, and it is confusing for users

#15 - 02/14/2022 02:08 PM - ktsj (Kazuki Tsujimoto)

- **[Feature #18585]** Promote find pattern to official feature (ktsj)
  - Is it OK to to promote find pattern to official feature?

#16 - 02/14/2022 02:15 PM - nobu (Nobuyoshi Nakada)

- **[Misc #18362]** mswin builds & vs2022
  - What can we do for the bug?

#17 - 02/14/2022 03:24 PM - byroot (Jean Boussier)

- **[Feature #18576]** Rename ASCII-8BIT encoding to BINARY
  - The ASCII-8BIT name is very confusing for beginners.
  - Even for people who know what it is, it can easily be misread as US-ASCII
  - It has been aliased as BINARY for a very long time, and in the vast majority of the case it's a better description.
  - I expect the backward compatibility problems to be very minimal (limited to Encoding#name).

#18 - 02/16/2022 05:36 PM - kddeisz (Kevin Newton)
- [Feature #18589] Finer-grained constant invalidation (Kevin Newton)
  - We want to trade a very small memory increase for much better constant cache invalidation.
  - This will allow relying on more stable constant caches for better code specialization in JITs.
  - This can also help speed up initial boot times since caches are currently invalidated each time a class is created.

#19 - 02/18/2022 02:30 AM - mame (Yusuke Endoh)
- Description updated
- Status changed from Open to Closed