### Description

I got a strange NoMethodError in Time.now.

```ruby
Integer.prepend(Module.new)
module Foo
  refine Integer do
    define_method(:+) { }
  end
end
Time.now
```

% ~/tmp/ruby-trunk/bin/ruby -v a.rb
ruby 2.4.0dev (2016-11-10 trunk 56692) [x86_64-darwin15]
a.rb:7:in `initialize': undefined method `+' for 1478773005000000000:Integer (NoMethodError)
  from a.rb:7:in `now'
  from a.rb:7:in `<main>'
```

### Related issues:

Related to Ruby master - Bug #12876: Calling new hangs Ruby when class prepended...

### Associated revisions

**Revision 9996945a - 11/10/2016 02:18 PM - nobu (Nobuyoshi Nakada)**

vm_eval.c: fix refined method when prepended

- vm_eval.c (vm_call0_body): refined module should not be skipped as prepended. [ruby-core:78073] [Bug #12920]

**Revision 56694 - 11/10/2016 02:18 PM - nobu (Nobuyoshi Nakada)**

vm_eval.c: fix refined method when prepended

- vm_eval.c (vm_call0_body): refined module should not be skipped as prepended. [ruby-core:78073] [Bug #12920]

**Revision 56694 - 11/10/2016 02:18 PM - nobu (Nobuyoshi Nakada)**

vm_eval.c: fix refined method when prepended

- vm_eval.c (vm_call0_body): refined module should not be skipped as prepended. [ruby-core:78073] [Bug #12920]

**Revision 56694 - 11/10/2016 02:18 PM - nobu (Nobuyoshi Nakada)**

vm_eval.c: fix refined method when prepended

- vm_eval.c (vm_call0_body): refined module should not be skipped as prepended. [ruby-core:78073] [Bug #12920]

**Revision def18ca5 - 11/11/2016 12:08 AM - nobu (Nobuyoshi Nakada)**

vm_eval.c: reorder
vm_eval.c (vm_call0_body): reorder the conditions. [ruby-core:78073] [Bug #12920]

Revision 56698 - 11/11/2016 12:08 AM - nobu (Nobuyoshi Nakada)
vm_eval.c: reorder
  • vm_eval.c (vm_call0_body): reorder the conditions. [ruby-core:78073] [Bug #12920]

Revision 56698 - 11/11/2016 12:08 AM - nobu (Nobuyoshi Nakada)
vm_eval.c: reorder
  • vm_eval.c (vm_call0_body): reorder the conditions. [ruby-core:78073] [Bug #12920]

Revision 56698 - 11/11/2016 12:08 AM - nobu (Nobuyoshi Nakada)
vm_eval.c: reorder
  • vm_eval.c (vm_call0_body): reorder the conditions. [ruby-core:78073] [Bug #12920]

Revision 56698 - 11/11/2016 12:08 AM - nobu (Nobuyoshi Nakada)
vm_eval.c: reorder
  • vm_eval.c (vm_call0_body): reorder the conditions. [ruby-core:78073] [Bug #12920]

Revision 56698 - 11/11/2016 12:08 AM - nobu (Nobuyoshi Nakada)
vm_eval.c: reorder
  • vm_eval.c (vm_call0_body): reorder the conditions. [ruby-core:78073] [Bug #12920]

Revision 9b5c2baa - 11/16/2016 02:41 PM - nagachika (Tomoyuki Chikanaga)
merge revision(s) 56694,56698: [Backport #12920]
  • vm_eval.c (vm_call0_body): refined module should not be skipped as prepended.  [ruby-core:78073] [Bug #12920]

Revision 56816 - 11/16/2016 02:41 PM - nagachika (Tomoyuki Chikanaga)
merge revision(s) 56694,56698: [Backport #12920]
  * vm_eval.c (vm_call0_body): refined module should not be skipped as prepended.  [ruby-core:78073] [Bug #12920]

History

#1 - 11/10/2016 11:36 AM - akr (Akira Tanaka)
The problem is not reproduced with ruby-2.4.0-preview2.
The problem is reproduced with ruby-2.4.0-preview3.
It seems something changed between ruby-2.4.0-preview2 and ruby-2.4.0-preview3.

#2 - 11/10/2016 02:18 PM - nobu (Nobuyoshi Nakada)
  - Status changed from Open to Closed

Applied in changeset r56694.

vm_eval.c: fix refined method when prepended
  • vm_eval.c (vm_call0_body): refined module should not be skipped as prepended.  [ruby-core:78073] [Bug #12920]

#3 - 11/15/2016 01:41 PM - kyanagi (Kouhei Yanagita)
This problem happens with Ruby 2.3.2.

% cat a.rb
Fixnum.prepend(Module.new)
Module.new do
  refine Fixnum do
    define_method(:+) {}
  end
end

Time.now

% ruby -v a.rb
ruby 2.3.2p217 (2016-11-15 revision 56796) [x86_64-darwin16]
a.rb:8:in `initialize': undefined method `+' for 1479217045000000000:Fixnum (NoMethodError)
  Did you mean? +@
  from a.rb:8:in `now'
  from a.rb:8:in `<main>'

#4 - 11/16/2016 10:33 AM - usa (Usaku NAKAMURA)
  - Backport changed from 2.1: UNKNOWN, 2.2: UNKNOWN, 2.3: UNKNOWN to 2.1: UNKNOWN, 2.2: UNKNOWN, 2.3: REQUIRED

#5 - 11/16/2016 02:30 PM - nagachika (Tomoyuki Chikanaga)
OMG I've confirmed with the script in the description (prepend and using with Integer) and it cannot reproduce NoMethodError on 2.3.2. But it's because Time.now doesn't use Integer#+ but Fixnum#+.
I've confirmed https://bugs.ruby-lang.org/issues/12920#note-3 was reproducible.
I'm going to backport this and try to understand the impact of this issue.

#6 - 11/16/2016 02:40 PM - usa (Usaku NAKAMURA)
  - Related to Bug #12876: Calling new hangs Ruby when class prepends an empty module and makes initialize method public added

#7 - 11/16/2016 02:41 PM - nagachika (Tomoyuki Chikanaga)
  - Backport changed from 2.1: UNKNOWN, 2.2: UNKNOWN, 2.3: REQUIRED to 2.1: UNKNOWN, 2.2: UNKNOWN, 2.3: DONE

ruby_2_3 r56816 merged revision(s) 56694,56698.