

Ruby - Bug #14702

On Ruby 2.5.1, tracepoint isn't working on the file that is loaded by load_iseq

04/20/2018 10:03 AM - makoto_tajitsu@hotmail.co.jp (Makoto Tajitsu)

Status:	Closed	
Priority:	Normal	
Assignee:	ko1 (Koichi Sasada)	
Target version:		
ruby -v:	ruby 2.5.1p57 (2018-03-29 revision 63029) [x86_64-darwin16]	Backport: 2.3: UNKNOWN, 2.4: UNKNOWN, 2.5: DONE

Description

On Ruby 2.5.1, when loading file with RubyVM::InstructionSequence.load_from_binary, TracePoint callback event is not working on loaded file.

Steps to reproduce

Ruby version.

```
$ ruby -v
# => ruby 2.5.1p57 (2018-03-29 revision 63029) [x86_64-darwin16]
```

The example with yomikomu gem which is simply used load_iseq and load_from_binary method to load file quickly is following.

```
$ YOMIKOMU_AUTO_COMPILE=true bundle exec irb

> TracePoint.trace(:line) { |tp| puts tp.lineno if tp.path == '/path/to/somefile.rb' }
# => #<TracePoint:enabled>

> load '/path/to/somefile.rb'
# 3
# 4
# ...

> require 'yomikomu'

> load '/path/to/somefile.rb'
# no tracepoint callback for line event
```

Result of reproduce process

tracepoint callback isn't called on loaded file by using load_iseq and load_from_binary.

Expected result and the reason why you expect

tracepoint callback should be called for any loaded file to debug with debugger(e.g. byebug).

On Ruby 2.4.3, this example is working correctly, so tracepoint callback is called and output lineno to stdout when loaded file after require 'yomikomu'.

Associated revisions

Revision b85b10c11f9846908a319e72c92e779e46c2217f - 08/23/2018 04:12 AM - ko1 (Koichi Sasada)

check trace flags at loading [Bug #14702]

- iseq.c (iseq_init_trace): at ISeq loading time, we need to check ruby_vm_event_enabled_flags to turn on trace instructions. Separate this checking code from finish_iseq_build() and make new function. iseq_ibf_load() calls this function after loading.
- test/ruby/test_iseq.rb: add a test for this fix.

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@64514 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision b85b10c11f9846908a319e72c92e779e46c2217f - 08/23/2018 04:12 AM - ko1 (Koichi Sasada)

check trace flags at loading [Bug #14702]

- iseq.c (iseq_init_trace): at ISeq loading time, we need to check ruby_vm_event_enabled_flags to turn on trace instructions. Seprate this checking code from finish_iseq_build() and make new function. iseq_ibf_load() calls this funcation after loading.
- test/ruby/test_iseq.rb: add a test for this fix.

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@64514 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision b85b10c1 - 08/23/2018 04:12 AM - ko1 (Koichi Sasada)

check trace flags at loading [Bug #14702]

- iseq.c (iseq_init_trace): at ISeq loading time, we need to check ruby_vm_event_enabled_flags to turn on trace instructions. Seprate this checking code from finish_iseq_build() and make new function. iseq_ibf_load() calls this funcation after loading.
- test/ruby/test_iseq.rb: add a test for this fix.

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@64514 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision c1034574aa7e6f47843b5533411c75d8196b29d8 - 10/11/2018 02:20 PM - nagachika (Tomoyuki Chikanaga)

merge revision(s) 64514: [Backport #14702]

```
check trace flags at loading [Bug #14702]
```

```
* iseq.c (iseq_init_trace): at ISeq loading time, we need to check
`ruby_vm_event_enabled_flags` to turn on trace instructions.
Seprate this checking code from `finish_iseq_build()` and make
new function. `iseq_ibf_load()` calls this fuction after loading.
```

```
* test/ruby/test_iseq.rb: add a test for this fix.
```

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/branches/ruby_2_5@64997 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision c1034574aa7e6f47843b5533411c75d8196b29d8 - 10/11/2018 02:20 PM - nagachika (Tomoyuki Chikanaga)

merge revision(s) 64514: [Backport #14702]

```
check trace flags at loading [Bug #14702]
```

```
* iseq.c (iseq_init_trace): at ISeq loading time, we need to check
`ruby_vm_event_enabled_flags` to turn on trace instructions.
Seprate this checking code from `finish_iseq_build()` and make
new function. `iseq_ibf_load()` calls this fuction after loading.
```

```
* test/ruby/test_iseq.rb: add a test for this fix.
```

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/branches/ruby_2_5@64997 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision c1034574 - 10/11/2018 02:20 PM - nagachika (Tomoyuki Chikanaga)

merge revision(s) 64514: [Backport #14702]

```
check trace flags at loading [Bug #14702]
```

```
* iseq.c (iseq_init_trace): at ISeq loading time, we need to check
`ruby_vm_event_enabled_flags` to turn on trace instructions.
Seprate this checking code from `finish_iseq_build()` and make
new function. `iseq_ibf_load()` calls this fuction after loading.
```

```
* test/ruby/test_iseq.rb: add a test for this fix.
```

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/branches/ruby_2_5@64997 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

History

#1 - 04/20/2018 10:05 AM - makoto_tajitsu@hotmail.co.jp (Makoto Tajitsu)

- Description updated

#2 - 04/20/2018 10:06 AM - makoto_tajitsu@hotmail.co.jp (Makoto Tajitsu)

- Description updated

#3 - 04/20/2018 10:16 AM - makoto_tajitsu@hotmail.co.jp (Makoto Tajitsu)

- Description updated

#4 - 04/30/2018 04:53 PM - makoto_tajitsu@hotmail.co.jp (Makoto Tajitsu)

- Description updated

How is the issue?

In addition to above, I confirm the issue in following example that does not use 3rd party gem.

```
#!/usr/bin/env ruby

# Write ruby file
File.write('./loaded_file.rb', 'puts :hello')

# load file and write iseq file
iseq = RubyVM::InstructionSequence.compile_file('./loaded_file.rb')
File.write('./loaded_file.yarb', iseq.to_binary)

# define load_iseq
class RubyVM::InstructionSequence
  def self.load_iseq(fname)
    b = File.read('./loaded_file.yarb')
    RubyVM::InstructionSequence.load_from_binary(b)
  end
end

TracePoint.trace(:line) do |tp|
  puts "file: #{tp.path}, line: #{tp.lineno}" if tp.path =~ /loaded_file/
end

load './loaded_file.rb' # => on ruby 2.5.0 and 2.5.1, no tracepoint callback is.
```

#5 - 08/22/2018 02:45 AM - ko1 (Koichi Sasada)

- Assignee set to ko1 (Koichi Sasada)

#6 - 08/23/2018 04:13 AM - ko1 (Koichi Sasada)

- Status changed from Open to Closed

Applied in changeset trunk|r64514.

check trace flags at loading [Bug [#14702](#)]

- iseq.c (iseq_init_trace): at ISeq loading time, we need to check ruby_vm_event_enabled_flags to turn on trace instructions. Separate this checking code from finish_iseq_build() and make new function. iseq_ibf_load() calls this function after loading.
- test/ruby/test_iseq.rb: add a test for this fix.

#7 - 08/23/2018 11:00 AM - nagachika (Tomoyuki Chikanaga)

- Backport changed from 2.3: UNKNOWN, 2.4: UNKNOWN, 2.5: UNKNOWN to 2.3: UNKNOWN, 2.4: UNKNOWN, 2.5: REQUIRED

#8 - 10/11/2018 02:20 PM - nagachika (Tomoyuki Chikanaga)

- Backport changed from 2.3: UNKNOWN, 2.4: UNKNOWN, 2.5: REQUIRED to 2.3: UNKNOWN, 2.4: UNKNOWN, 2.5: DONE

ruby_2_5 r64997 merged revision(s) 64514.

#9 - 10/28/2018 06:22 PM - alanwu (Alan Wu)

- *File 0001-Fix-TracePoint-for-nested-iseq-loaded-from-binary.patch added*

Hello. I don't think this issue is fully fixed.

I have put together a patch that has a fix and tests that fail on both trunk and ruby_2_5 for your consideration:

I've opened up a ticket for this patch: [#15270](#)

```
Fix TracePoint for nested iseq loaded from binary [Bug #14702]

When loading iseq from binary while a TracePoint is on, we need to
recompile instructions to their "trace_" variant. Before this commit
we only recompiled instructions in the top level iseq, which meant
that TracePoint was malfunctioning for code inside module/class/method
definitions.

* compile.c: Move rb_iseq_init_trace to rb_ibf_load_iseq_complete.
  It is called on all iseqs during loading.

* test_iseq.rb: Test that tracepoints fire within children iseq when
  using load_from_binary.
```

#10 - 10/30/2018 02:49 PM - alanwu (Alan Wu)

- *File deleted (0001-Fix-TracePoint-for-nested-iseq-loaded-from-binary.patch)*

#11 - 03/04/2019 11:58 PM - fxn (Xavier Noria)

Hi!

Rails 6 applications have bootsnap and Zeitwerk enabled by default. Right now, that needs Ruby 2.6 because Zeitwerk sets a trace point on the :class event to support what I call "explicit namespaces".

Just wanted to let you know about this situation, in case you'd like to take it into account for the timeline to ship 2.5.4.

#12 - 03/05/2019 08:21 AM - fxn (Xavier Noria)

For reference: <https://github.com/rails/rails/issues/35475>.