Ruby - Bug #10449

[TracePoint API] Duplicated line events when using parenthesis

10/29/2014 03:08 PM - deivid (David Rodríguez)

Status:	Closed		
Priority:	Normal		
Assignee:			
Target version:			
ruby -v:	ruby 2.1.4p265 (2014-10-27 revision 48166) [i686-linux]	**Backport:**	2.0.0: UNKNOWN, 2.1: DONE, 2.2: DONTNEED

Description

Sometimes, when using parenthesis, I get duplicated lines events from the TracePoint API. See the failing test case, it generates 2 line events where I think it should generate just one.

```
def test_no_duplicate_line_events
  events = []
  TracePoint.new(:line) { |tp| events << tp.event }.enable { a = (1) * 5 }

  assert_equal [:line], events
end
```

Thanks!!

Associated revisions

Revision ea290804891b259cce4a156790ba024c1d52ad27 - 11/27/2014 11:06 AM - ko1 (Koichi Sasada)

- compile.c (iseq_compile_each): remove duplicated line event.
  [Bug #10449]
- test/ruby/test_settracefunc.rb: add and fix tests.

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@48609 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision ea290804 - 11/27/2014 11:06 AM - ko1 (Koichi Sasada)

- compile.c (iseq_compile_each): remove duplicated line event.
  [Bug #10449]
- test/ruby/test_settracefunc.rb: add and fix tests.

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@48609 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision 9f8d00ff9aa85d097def654494b16c2ce09d8e92 - 11/18/2015 11:24 AM - U.Nakamura

merge revision(s) 48609: [Backport #10449] [Backport #11651]

```
    * compile.c (iseq_compile_each): remove duplicated line event.
      [Bug #10449]

    * test/ruby/test_settracefunc.rb: add and fix tests.
```

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/branches/ruby_2_1@52640 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision 9f8d00ff - 11/18/2015 11:24 AM - U.Nakamura

merge revision(s) 48609: [Backport #10449] [Backport #11651]

```
    * compile.c (iseq_compile_each): remove duplicated line event.
      [Bug #10449]

    * test/ruby/test_settracefunc.rb: add and fix tests.
```

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/branches/ruby_2_1@52640 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

History

#1 - 11/27/2014 11:07 AM - ko1 (Koichi Sasada)

- Status changed from Open to Closed

- % Done changed from 0 to 100

Applied in changeset r48609.

- compile.c (iseq_compile_each): remove duplicated line event.
  [Bug #10449]
- test/ruby/test_settracefunc.rb: add and fix tests.

#2 - 11/28/2014 01:28 PM - deivid (David Rodríguez)

Hi Koichi, I've looked at the patch and it seems to me that *every* duplicated event has been removed. So if I do something like

`object.method1.method2.method3`

only one line event will be generated. Is this the case?

If that's the case I actually preferred the previous behaviour so it is possible to inspect intermediate results. byebug is aware of this duplication and even provides an option that ignores duplicated events in case the user wants to always change line when stepping through the code.

I just reported this specific case cause I couldn't recognize the "separate line events" but in general I think the behaviour was fine.

#3 - 11/30/2014 02:03 AM - deivid (David Rodríguez)

Confirmed, this commit breaks some tests in Byebug, I think it should be reverted.

#4 - 11/30/2014 11:15 AM - deivid (David Rodríguez)

Actually I've given this a second thought and I'm fine with the change. The funcionality loss is minimal (and easy to do it in other ways), the patch corrects issues like this one and the code using the API will be simpler as it will not need to handle the duplication.

So basically, I'm fine with anything. :)

#5 - 12/01/2014 05:45 AM - ko1 (Koichi Sasada)

> Hi Koichi, I've looked at the patch and it seems to me that *every* duplicated event has been removed. So if I do something like
>
> `object.method1.method2.method3`
>
> only one line event will be generated. Is this the case?

Before this fix, only 1 line event (for the method chain line) is generated, isn't?

```
     1  obj = Object.new
     2  def obj.method_missing *args
     3    p args
     4    self
     5  end
     6
     7  TracePoint.new(:line){|tp|
     8    p [tp.event, tp.lineno]
     9  }.enable{
    10    obj.method1.method2.method3 # method chain line
    11  }
```

```
#=>
ruby 2.1.4p261 (2014-10-19 revision 48004) [x86_64-linux]
[:line, 10] #<-- this line
[:line, 3]
[:method1]
[:line, 4]
[:line, 3]
[:method2]
[:line, 4]
[:line, 3]
[:method3]
[:line, 4]
```

#6 - 12/01/2014 01:35 PM - deivid (David Rodríguez)

Oops, I need to start thinking before talking... :) The change in the behaviour actually happened in this example so something like the following used to generate 2 line events before the change.

```
TracePoint.new(:line){|tp|
  p [tp.event, tp.lineno]
}.enable{
  num = 1 ; num += 1
}
```

Using semicolon is the only case I found when this happened (other that this issue), and it's not a very popular style anyways.

So definitely up for keeping this fix. Could we get this backported to 2.0 and 2.1?

THANKS!

#7 - 11/03/2015 01:16 PM - usa (Usaku NAKAMURA)

- Backport changed from 2.0.0: UNKNOWN, 2.1: UNKNOWN to 2.0.0: UNKNOWN, 2.1: REQUIRED, 2.2: DONTNEED

#9 - 11/18/2015 11:24 AM - usa (Usaku NAKAMURA)

- Backport changed from 2.0.0: UNKNOWN, 2.1: REQUIRED, 2.2: DONTNEED to 2.0.0: UNKNOWN, 2.1: DONE, 2.2: DONTNEED

ruby_2_1 r52640 merged revision(s) 48609.