

Backport200 - Backport #8489

Tracepoint API: B_RETURN_EVENT not triggered when "next" used

06/05/2013 12:27 AM - deivid (David Rodríguez)

Status:	Closed
Priority:	Normal
Assignee:	nagachika (Tomoyuki Chikanaga)
Description	
<p>I'm using the Tracepoint API in my gem and I notice that when the keyword <code>next</code> is used inside a block, a block return event (<code>B_RETURN_EVENT</code>) is not triggered. For example, I would expect the same number of <code>B_CALL_EVENT</code> and <code>B_RETURN_EVENT</code> in the piece of code:</p> <pre>traceBlocks = TracePoint.new(:b_call, :b_return) do tp p "#{tp.event == :b_call ? 'Block called' : 'Block returned'}" end.enable do 3.times do next end end</pre> <p>but instead there are 4 <code>B_CALL_EVENT</code> and 1 <code>B_RETURN_EVENT</code>. As I understand it, "next" causes the block to exit immediately, returning control to the iterator, which may invoke the block again, so I would say a block return event should be triggered.</p> <p>Thanks a lot.</p>	

Associated revisions

Revision f41b284f - 06/14/2013 09:54 AM - ko1 (Koichi Sasada)

- `compile.c (rb_iseq_compile_node)`: fix location of a ``trace'` instruction (`b_return` event). [ruby-core:55305] [ruby-trunk - Bug #8489](#)
- `test/ruby/test_settracefunc.rb`: add a test.

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@41300 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision 41300 - 06/14/2013 09:54 AM - ko1 (Koichi Sasada)

- `compile.c (rb_iseq_compile_node)`: fix location of a ``trace'` instruction (`b_return` event). [ruby-core:55305] [ruby-trunk - Bug #8489](#)
- `test/ruby/test_settracefunc.rb`: add a test.

Revision 41300 - 06/14/2013 09:54 AM - ko1 (Koichi Sasada)

- `compile.c (rb_iseq_compile_node)`: fix location of a ``trace'` instruction (`b_return` event). [ruby-core:55305] [ruby-trunk - Bug #8489](#)
- `test/ruby/test_settracefunc.rb`: add a test.

Revision 41300 - 06/14/2013 09:54 AM - ko1 (Koichi Sasada)

- `compile.c (rb_iseq_compile_node)`: fix location of a ``trace'` instruction (`b_return` event). [ruby-core:55305] [ruby-trunk - Bug #8489](#)
- `test/ruby/test_settracefunc.rb`: add a test.

Revision 41300 - 06/14/2013 09:54 AM - ko1 (Koichi Sasada)

- `compile.c (rb_iseq_compile_node)`: fix location of a ``trace'` instruction (`b_return` event). [ruby-core:55305] [ruby-trunk - Bug #8489](#)
- `test/ruby/test_settracefunc.rb`: add a test.

Revision 41300 - 06/14/2013 09:54 AM - ko1 (Koichi Sasada)

- `compile.c (rb_iseq_compile_node)`: fix location of a ``trace'` instruction (`b_return` event). [ruby-core:55305] [ruby-trunk - Bug #8489](#)
- `test/ruby/test_settracefunc.rb`: add a test.

Revision 41300 - 06/14/2013 09:54 AM - ko1 (Koichi Sasada)

- `compile.c (rb_iseq_compile_node): fix location of a `trace' instruction (b_return event).` [ruby-core:55305] [ruby-trunk - Bug #8489](#)
- `test/ruby/test_settracefunc.rb: add a test.`

Revision 113d3599 - 06/15/2013 04:14 PM - nagachika (Tomoyuki Chikanaga)

merge revision(s) 41300: [Backport #8489]

```
* compile.c (rb_iseq_compile_node): fix location of a `trace'
  instruction (b_return event).
  [ruby-core:55305] [ruby-trunk - Bug #8489]
  (need a backport to 2.0.0?)

* test/ruby/test_settracefunc.rb: add a test.
```

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/branches/ruby_2_0_0@41319 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision 41319 - 06/15/2013 04:14 PM - nagachika (Tomoyuki Chikanaga)

merge revision(s) 41300: [Backport #8489]

```
* compile.c (rb_iseq_compile_node): fix location of a `trace'
  instruction (b_return event).
  [ruby-core:55305] [ruby-trunk - Bug #8489]
  (need a backport to 2.0.0?)

* test/ruby/test_settracefunc.rb: add a test.
```

History

#1 - 06/14/2013 05:55 PM - deivid (David Rodríguez)

This is the "no replies after 10 days reminder".

If this is not a priority right now, I can try have a look at this myself (not sure how it will go though...).

Thanks again.

#2 - 06/14/2013 06:55 PM - ko1 (Koichi Sasada)

- Status changed from Open to Closed

- % Done changed from 0 to 100

This issue was solved with changeset r41300.

David, thank you for reporting this issue.

Your contribution to Ruby is greatly appreciated.

May Ruby be with you.

-
- `compile.c (rb_iseq_compile_node): fix location of a `trace' instruction (b_return event).` [ruby-core:55305] [ruby-trunk - Bug #8489](#)
 - `test/ruby/test_settracefunc.rb: add a test.`

#3 - 06/14/2013 06:59 PM - ko1 (Koichi Sasada)

Thank you for your report.

I fixed it. Could you check it?

(2013/06/14 17:55), deivid (David Rodríguez) wrote:

Issue [#8489](#) has been updated by deivid (David Rodríguez).

This is the "no replies after 10 days reminder".

If this is not a priority right now, I can try have a look at this myself (not sure how it will go though...).

Thanks again.

Bug [#8489](#): Tracepoint API: B_RETURN_EVENT not triggered when "next" used
<https://bugs.ruby-lang.org/issues/8489#change-39925>

Author: deivid (David Rodríguez)

Status: Open

Priority: Normal

Assignee:

Category:

Target version:
ruby -v: ruby 2.0.0p201 (2013-06-04) [i686-linux]
Backport: 1.9.3: UNKNOWN, 2.0.0: UNKNOWN

I'm using the Tracepoint API in my gem and I notice that when the keyword next is used inside a block, a block return event (B_RETURN_EVENT) is not triggered. For example, I would expect the same number of B_CALL_EVENT and B_RETURN_EVENT in the piece of code:

```
traceBlocks = TracePoint.new(:b_call, :b_return) do |tp|
  p "#{tp.event} == :b_call ? 'Block called' : 'Block returned'"
end.enable do
  3.times do
    next
  end
end
```

but instead there are 4 B_CALL_EVENT and 1 B_RETURN_EVENT. As I understand it, "next" causes the block to exit immediately, returning control to the iterator, which may invoke the block again, so I would say a block return event should be triggered.

Thanks a lot.

--
// SASADA Koichi at atdot dot net

#4 - 06/14/2013 09:54 PM - nagachika (Tomoyuki Chikanaga)

- Backport changed from 1.9.3: UNKNOWN, 2.0.0: UNKNOWN to 1.9.3: UNKNOWN, 2.0.0: REQUIRED

#5 - 06/15/2013 12:41 AM - deivid (David Rodríguez)

Just tried it [ko1 \(Koichi Sasada\)](#), it works perfectly!

[nagachika \(Tomoyuki Chikanaga\)](#), forgive my ignorance but "Backport to 2.0.0" means the patch will be released with the next patchlevel release?

Thanks a lot!!

#6 - 06/15/2013 09:53 AM - zzak (Zachary Scott)

On Sat, Jun 15, 2013 at 12:41 AM, deivid (David Rodríguez) deivid.rodriguez@gmail.com wrote:

[nagachika \(Tomoyuki Chikanaga\)](#), forgive my ignorance but "Backport to 2.0.0" means the patch will be released with the next patchlevel release?

Yes, nagachika-san is the 2.0.0 release manager, he will backport this for the next patchlevel release of ruby 2.0.0 series.

#7 - 06/15/2013 10:31 AM - deivid (David Rodríguez)

Sweet, thanks a lot guys!

#8 - 06/16/2013 01:14 AM - nagachika (Tomoyuki Chikanaga)

- Tracker changed from Bug to Backport
- Project changed from Ruby master to Backport200
- Status changed from Closed to Assigned
- Assignee set to nagachika (Tomoyuki Chikanaga)

Hello, deivid.
Never mind. It's my work to check if a changeset need to backport :)

And I'm going to backport r41300.

#9 - 06/16/2013 01:14 AM - nagachika (Tomoyuki Chikanaga)

- Status changed from Assigned to Closed

This issue was solved with changeset [r41319](#).
David, thank you for reporting this issue.
Your contribution to Ruby is greatly appreciated.

merge revision(s) 41300: [Backport [#8489](#)]

```
* compile.c (rb_iseq_compile_node): fix location of a `trace`  
  instruction (b_return event).  
  [ruby-core:55305] [ruby-trunk - Bug #8489]  
  (need a backport to 2.0.0?)
```

```
* test/ruby/test_settracefunc.rb: add a test.
```