Ruby trunk - Bug #4559

Proc#== does not match the documented behaviour

04/07/2011 05:07 AM - aprescott (Adam Prescott)

Status:	Closed		
Priority:	Normal		
Assignee:	ko1 (Koichi Sasada)		
Target version:	2.0.0		
ruby -v:	ruby 1.9.2p180 (2011-02-18 revision 30909) [i686-linux]	**Backport:**	

Description

=begin
To see a paste concisely showing the problem, see https://gist.github.com/899026/ba84d66b02727675320dc4f031637d753a41c209

Code is shown below for 1.9.2p180.

The documented behaviour (and my own expectation for (({#==}))) is: "Return true if prc is the same object as other_proc, or if they are both procs with the same body."

RUBY_DESCRIPTION #=> ruby 1.9.2p180 (2011-02-18 revision 30909) [i686-linux]

```
proc { 1 }     == proc { 1 }      # => true
proc { 1 + 1 } == proc { 1 + 1 }  # => false

proc { |x| x }     == proc { |x| x }      # => true
proc { |x| x.foo } == proc { |x| x.foo }  # => false

lambda { 1 }     == lambda { 1 }      # => true
lambda { 1 + 1 } == lambda { 1 + 1 }  # => false

lambda { |x| x }     == lambda { |x| x }      # => true
lambda { |x| x.foo } == lambda { |x| x.foo }  # => false

Proc.new { 1 }     == Proc.new { 1 }      # => true
Proc.new { 1 + 1 } == Proc.new { 1 + 1 }  # => false

Proc.new { |x| x }     == Proc.new { |x| x }      # => true
Proc.new { |x| x.foo } == Proc.new { |x| x.foo }  # => false
```

Similar also occurs on 1.8.7; on version 1.8.7 (2011-02-18 patchlevel 334), all returned values are (({false})). I also reproduced this behaviour on 1.9.2p136 (2010-12-25 revision 30365) [i686-linux].

Attached is a file which will (({puts})) the values.
=end

Related issues:

Related to Ruby trunk - Bug #4323: Proc#hash is Ill-Behaved	**Closed**	**01/26/2011**

Associated revisions

Revision fc57f2bf - 11/28/2012 08:01 AM - ko1 (Koichi Sasada)

add ticket number [Bug #4559]

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@37929 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision 37929 - 11/28/2012 08:01 AM - ko1 (Koichi Sasada)

add ticket number [Bug #4559]

Revision 37929 - 11/28/2012 08:01 AM - ko1 (Koichi Sasada)

add ticket number [Bug #4559]

Revision 37929 - 11/28/2012 08:01 AM - ko1 (Koichi Sasada)

add ticket number [Bug #4559]

Revision 37929 - 11/28/2012 08:01 AM - ko1 (Koichi Sasada)

add ticket number [Bug #4559]

Revision 37929 - 11/28/2012 08:01 AM - ko1 (Koichi Sasada)

add ticket number [Bug #4559]

Revision 37929 - 11/28/2012 08:01 AM - ko1 (Koichi Sasada)

add ticket number [Bug #4559]

History

#1 - 06/11/2011 05:05 PM - ko1 (Koichi Sasada)

- Category set to doc

- Status changed from Open to Assigned

- Assignee set to nobu (Nobuyoshi Nakada)

#2 - 06/13/2011 10:43 PM - ko1 (Koichi Sasada)

- Assignee changed from nobu (Nobuyoshi Nakada) to matz (Yukihiro Matsumoto)

- Priority changed from Normal to 3

- Target version set to 2.0.0

#3 - 06/14/2011 12:25 AM - matz (Yukihiro Matsumoto)

I am not satisfied with the current behavior. Two proc objects should
equal either

- when two proc objects are the same (identical) object (as 1.8 does).
- or when two proc objects share same body (as documented).

The 1.9 behavior is half-baked, and I think it should be fixed.
I don't say which way to go. It's up to the maintainer (ko1, you).

```
                        matz.
```

#4 - 06/14/2011 12:29 AM - usa (Usaku NAKAMURA)

- Category changed from doc to core

- Assignee changed from matz (Yukihiro Matsumoto) to ko1 (Koichi Sasada)

#5 - 07/14/2012 02:36 PM - ko1 (Koichi Sasada)

Matz (and akr) agreed that Proc#== only see the object_id. I'll change behavior and documents.

#6 - 09/22/2012 09:40 AM - ko1 (Koichi Sasada)

To solve this ticket, I only need to remove Proc#== from proc.c, isn't it?

#7 - 11/26/2012 06:54 PM - ko1 (Koichi Sasada)

def test_eq2
b1 = proc { }
b2 = b1.dup
assert(b1 == b2)
end

will be `false'. Is it okay?

#8 - 11/28/2012 04:10 PM - matz (Yukihiro Matsumoto)

ko1 (Koichi Sasada) You've chosen the latter option, so it is natural conclusion.

Matz.

#9 - 11/28/2012 04:23 PM - ko1 (Koichi Sasada)

(2012/11/28 16:10), matz (Yukihiro Matsumoto) wrote:

> ko1 (Koichi Sasada) You've chosen the latter option, so it is natural conclusion.

Okay. I'll delete Proc#==.

--
// SASADA Koichi at atdot dot net

#10 - 11/28/2012 05:01 PM - ko1 (Koichi Sasada)

- Status changed from Assigned to Closed

- % Done changed from 0 to 100

This issue was solved with changeset r37929.
Adam, thank you for reporting this issue.
Your contribution to Ruby is greatly appreciated.
May Ruby be with you.

---

add ticket number [Bug #4559]

#11 - 12/03/2012 03:29 PM - headius (Charles Nutter)

I believe this will be a spec change, albeit a small one. Can we
expect a test or rubyspec to be added for the altered behavior?

- Charlie

On Wed, Nov 28, 2012 at 1:10 AM, SASADA Koichi ko1@atdot.net wrote:

> (2012/11/28 16:10), matz (Yukihiro Matsumoto) wrote:
>
> > ko1 (Koichi Sasada) You've chosen the latter option, so it is natural conclusion.
>
> Okay. I'll delete Proc#==.
>
> --
> // SASADA Koichi at atdot dot net

#12 - 12/03/2012 04:29 PM - headius (Charles Nutter)

On Mon, Dec 3, 2012 at 1:15 AM, SASADA Koichi ko1@atdot.net wrote:

> (2012/12/03 15:26), Charles Oliver Nutter wrote:
>
> > I believe this will be a spec change, albeit a small one. Can we
> > expect a test or rubyspec to be added for the altered behavior?
>
> What is the best way to do with?
>
> I only removed one test from test/ruby/test_proc.rb it depends on
> previous behavior. (proc_obj == proc_obj.dup).

A similar change for RubySpec may be appropriate. In RubySpec, it
would be by putting a version guard around existing #== expectations
that are no longer valid on 2.0, something like this:

ruby_version_is "1.8.6...2.0.0" do
...

Note the exclusive range there. I think it's reasonable to assume that
if Object methods are not customized in Proc, there's no specified
behavior for them other than Object's (or else we'd have to specify
that every Object method in every subclass exhibits the same behavior
as Object).

> Making pull request to rubyspec?

That would be fine for now. We can get you commit rights as well.

> I need to learn how to write rubyspec to do it.

We'll happily walk you through that or answer any questions.

> (and how to make pull request!)

And this too :) Feel free to ping me on IRC whenever I'm around.

- Charlie

Files

proc-equality.rb	550 Bytes	04/07/2011	aprescott (Adam Prescott)