

Ruby trunk - Bug #14267

Lazy proc allocation introduced in #14045 creates regression

01/02/2018 06:47 AM - myronmarston (Myron Marston)

Status: Open	
Priority: Normal	
Assignee:	
Target version:	
ruby -v: ruby 2.5.0p0 (2017-12-25 revision 61468) [x86_64-darwin15]	Backport: 2.3: UNKNOWN, 2.4: UNKNOWN, 2.5: UNKNOWN
Description	
<p>The following script consistently prints Proc equality: true on versions of Ruby before 2.5, but prints Proc equality: false on Ruby 2.5:</p>	
<pre># regression.rb def return_proc(&block) block end def return_procs(&block) block.inspect if ENV['INSPECT_BLOCK'] proc_1 = return_proc(&block) proc_2 = return_proc(&block) return proc_1, proc_2 end proc_1, proc_2 = return_procs { } puts RUBY_VERSION puts "Proc equality: #{proc_1 == proc_2}"</pre>	
<p>Here's the output on Ruby 2.4 and 2.5:</p>	
<pre>\$ chruby 2.4 \$ ruby regression.rb 2.4.2 Proc equality: true \$ chruby 2.5 \$ ruby regression.rb 2.5.0 Proc equality: false</pre>	
<p>As the output shows, the two procs were equal on 2.4 but are no longer equal on 2.5. I believe this is due to the lazy proc allocation introduced in #14045. Note that if I call a method on the proc (such as inspect) it defeats the lazy allocation and "fixes" the regression:</p>	
<pre>\$ chruby 2.5 \$ INSPECT_BLOCK=1 ruby regression.rb 2.5.0 Proc equality: true</pre>	
<p>This caused a bug in RSpec, which I've worked around for now by calling <code>__id__</code> on the proc.</p>	
<p>Is there a way to keep the lazy proc allocation while fixing this regression?</p>	
Related issues:	
Related to Ruby trunk - Feature #14045: Lazy Proc allocation for block parame...	Closed

History

#1 - 01/02/2018 11:33 AM - duerst (Martin Dürst)

- Related to Feature #14045: Lazy Proc allocation for block parameters added