Ruby master - Feature #7035

defined? should return cached, frozen strings

09/18/2012 02:57 AM - headius (Charles Nutter)

<table>
<thead>
<tr>
<th>Status:</th>
<th>Closed</th>
</tr>
</thead>
<tbody>
<tr>
<td>Priority:</td>
<td>Normal</td>
</tr>
<tr>
<td>Assignee:</td>
<td>nobu (Nobuyoshi Nakada)</td>
</tr>
<tr>
<td>Target version:</td>
<td>2.0.0</td>
</tr>
</tbody>
</table>

Description

Yehuda and I have been looking into allocation rates in Rails under both MRI and JRuby, and one of the big standouts is defined? logic returning a new String every time (for success cases). We could think of no reason why defined? needs to return a new String, and neither of us know of any code in the wild that takes the resulting string and modifies it.

For systems that use defined? heavily, it would seem best to only ever return the same instance of a cached, frozen String, rather than a new String every time that is only used for its boolean-ness and thrown away. Eliminating these extra Strings would reduce allocation and GC burden on MRI, with only the tiniest behavioral change nobody will ever notice.

An alternative with a larger behavioral change would be to have defined? always return Symbol, rather than String. The additional danger here is if anyone is using the String result as a String for comparison purposes, which I have definitely seen in the wild.

In any case, I could come up with no justification for returning a new String every time, so I have implemented the cached, frozen logic in JRuby: https://github.com/jruby/jruby/commit/b03d0bc89aefca13deaff7a568e5d9118a9ca2a8

Associated revisions

Revision 2314b80d - 09/24/2012 08:36 AM - nobu (Nobuyoshi Nakada)

Feature #7035

- compile.c (defined_expr), insns.def (defined): share single frozen strings. [EXPERIMENTAL] [ruby-core:47558][Feature #7035]
- iseq.c (rb_iseq_defined_string): make expression strings.

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@37025 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision 37025 - 09/24/2012 08:36 AM - nobu (Nobuyoshi Nakada)

Feature #7035

- compile.c (defined_expr), insns.def (defined): share single frozen strings. [EXPERIMENTAL] [ruby-core:47558][Feature #7035]
- iseq.c (rb_iseq_defined_string): make expression strings.

Revision 37025 - 09/24/2012 08:36 AM - nobu (Nobuyoshi Nakada)

Feature #7035

- compile.c (defined_expr), insns.def (defined): share single frozen strings. [EXPERIMENTAL] [ruby-core:47558][Feature #7035]
- iseq.c (rb_iseq_defined_string): make expression strings.

Revision 37025 - 09/24/2012 08:36 AM - nobu (Nobuyoshi Nakada)

Feature #7035

- compile.c (defined_expr), insns.def (defined): share single frozen strings. [EXPERIMENTAL] [ruby-core:47558][Feature #7035]
- iseq.c (rb_iseq_defined_string): make expression strings.

Revision 37025 - 09/24/2012 08:36 AM - nobu (Nobuyoshi Nakada)

Feature #7035

- compile.c (defined_expr), insns.def (defined): share single frozen strings. [EXPERIMENTAL] [ruby-core:47558][Feature #7035]
- iseq.c (rb_iseq_defined_string): make expression strings.

Revision 37025 - 09/24/2012 08:36 AM - nobu (Nobuyoshi Nakada)

Feature #7035

- compile.c (defined_expr), insns.def (defined): share single frozen strings. [EXPERIMENTAL] [ruby-core:47558][Feature #7035]
- iseq.c (rb_iseq_defined_string): make expression strings.
Feature #7035

- compile.c (defined_expr), insns.def (defined): share single frozen strings. [EXPERIMENTAL] [ruby-core:47558][Feature #7035]
- iseq.c (rb_iseq_defined_string): make expression strings.

History

#1 - 09/18/2012 03:22 AM - sikachu (Prem Sichanugrist)
IMO, I think it should be fine for defined? to return a symbol in Ruby 2.0.0, if that really going to fix the allocation issue. Simple fix.

This is a major release, and I think it's fine to break backward compatibility just a little. I think what we gain will be more than what we lose. If it break people's application, they can just convert it back to string using #to_s.

#2 - 09/18/2012 04:00 AM - luislavena (Luis Lavena)
=begin
I like the idea.
Considering recent changes like (?!respond_to?)) and protected methods, I think this change is not introducing a big backward incompatible change. Most of the code out in the wild uses (?!defined?)) to check for definitions, not considering the result value at all.
=end

#3 - 09/18/2012 04:06 AM - kosaki (Motohiro KOSAKI)

I like the idea.

I like it too.

#4 - 09/24/2012 05:36 PM - nobu (Nobuyoshi Nakada)
- Status changed from Open to Closed
- % Done changed from 0 to 100

This issue was solved with changeset r37025.
Charles, thank you for reporting this issue.
Your contribution to Ruby is greatly appreciated.
May Ruby be with you.

Feature #7035

- compile.c (defined_expr), insns.def (defined): share single frozen strings. [EXPERIMENTAL] [ruby-core:47558][Feature #7035]
- iseq.c (rb_iseq_defined_string): make expression strings.

#5 - 09/24/2012 08:47 PM - luislavena (Luis Lavena)
- Status changed from Closed to Assigned
- Assignee set to nobu (Nobuyoshi Nakada)
- % Done changed from 100 to 50

=begin
Hello Nobu,
I'm seeing this failing on both i386-mingw32 and x86-mingw32:

2) Error:
test_defined_impl_specific(TestDefined):
ArgumentError: wrong number of arguments (1 for 0)
C:/Users/Worker/Jenkins/workspace/ruby-trunk-x64-build/test/ruby/test_defined.rb:91:in test_defined_impl_specific'
C:/Users/Worker/Jenkins/workspace/ruby-trunk-x64-build/lib/test/unit/testcase.rb:17:inrun'
C:/Users/Worker/Jenkins/workspace/ruby-trunk-x64-build/lib/test/unit.rb:650:in block in _run_suites'
C:/Users/Worker/Jenkins/workspace/ruby-trunk-x64-build/lib/test/unit.rb:648:ineach'
C:/Users/Worker/Jenkins/workspace/ruby-trunk-x64-build/lib/test/unit.rb:648:in _run_suites'
C:/Users/Worker/Jenkins/workspace/ruby-trunk-x64-build/lib/test/unit.rb:21:inrun'
C:/Users/Worker/Jenkins/workspace/ruby-trunk-x64-build/lib/test/unit.rb:767:in run'
C:/Users/Worker/Jenkins/workspace/ruby-trunk-x64-build/lib/test/unit.rb:820:inrun'
C:/Users/Worker/Jenkins/workspace/ruby-trunk-x64-build/lib/test/unit.rb:824:in run'
=end
Thank you Yui NARUSE, r37026 solved the error.

Closing this now.