Backport200 - Backport #8966
SEGV when a refined method is undefined by undef
09/29/2013 10:05 PM - shugo (Shugo Maeda)

Status: Closed
Priority: Normal
Assignee: nagachika (Tomoyuki Chikanaga)

Description
=begin
The following code causes SEGV:

module Foo
refine Object do
  def foo
    puts "foo"
  end
end

using Foo

class Object
  undef foo
end

foo

I guess undef is not aware of refinements.
=end

Associated revisions
Revision 3cfa0074 - 10/04/2013 05:13 PM - nagachika (Tomoyuki Chikanaga)
merge revision(s) 43090,43091: [Backport #8966]

* vm_method.c (rb_undef): raise a NameError if the original method of a refined method is not defined.
* vm_insnhelper.c (rb_method_entry_eq): added NULL check to avoid SEGV.
* test/ruby/test_refinement.rb: related test.

Revision 43143 - 10/04/2013 05:13 PM - nagachika (Tomoyuki Chikanaga)
merge revision(s) 43090,43091: [Backport #8966]

* vm_method.c (rb_undef): raise a NameError if the original method of a refined method is not defined.
* vm_insnhelper.c (rb_method_entry_eq): added NULL check to avoid SEGV.
* test/ruby/test_refinement.rb: related test.

History
#1 - 09/29/2013 10:38 PM - nobu (Nobuyoshi Nakada)
- Description updated

=begin
What should happen in that case?
Printing `(("foo"))`, or raising a `((NameError))`?
=end
What should happen in that case? Printing `"foo"`, or raising a `NameError`?

`undef foo` should raise a NameError because foo is not defined in the original class.

committed at r43090.
And it should be backported to ruby_2_0_0.

Ah, I forgot to add the bug number to the commit log of r43090. Thanks.

This issue was solved with changeset r43143.
Shugo, thank you for reporting this issue.
Your contribution to Ruby is greatly appreciated.
May Ruby be with you.

* vm_method.c (rb_undef): raise a NameError if the original method of a refined method is not defined.

* vm_insnhelper.c (rb_method_entry_eq): added NULL check to avoid SEGV.

* test/ruby/test_refinement.rb: related test of a refined method is not defined.