Ruby master - Bug #11928

Segmentation fault in did_you_mean extension

12/30/2015 03:27 PM - amw (Adam Wróbel)

Status:	Closed		
Priority:	Normal		
Assignee:			
Target version:			
ruby -v:	ruby 2.3.0p0 (2015-12-25 revision 53290) [x86_64-darwin15]	**Backport:**	2.1: DONTNEED, 2.2: DONTNEED, 2.3: DONE

Description

I'm getting a segmentation fault in my Rails application test suite if I attempt to use an undefined method/object.

I was not able to reproduce this outside of the complex environment of my application, but hopefully you might be able to figure out what is going on just using the crash logs. I had to filter out lines that contained file paths of my application - fortunately they were only in the "Loaded features" section and not anywhere in C or Ruby backtraces.

Related issues:

Has duplicate Ruby master - Bug #12000: Crash report for 2.3.0	**Closed**
Has duplicate Ruby master - Bug #12107: Segmentation fault at 0x000000000000...	**Closed**
Has duplicate Ruby master - Bug #12078: Segmentation fault in did_you_mean wi...	**Closed**
Has duplicate Ruby master - Bug #12265: did_you_mean spell checker crash	**Closed**
Has duplicate Ruby master - Bug #12796: Segmentation fault at ~/.rvm/gems/rub...	**Closed**

Associated revisions

Revision 1b39a6e5 - 01/12/2016 03:17 PM - nobu (Nobuyoshi Nakada)

iseq.c: mark parents of wrapped iseq

- iseq.c (iseqw_mark): as wrapped iseq is isolated from the call stack, it needs to take care of its parent and ancestors, so that they do not become orphans. [ruby-core:72620] [Bug #11928]

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@53514 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision 53514 - 01/12/2016 03:17 PM - nobu (Nobuyoshi Nakada)

iseq.c: mark parents of wrapped iseq

- iseq.c (iseqw_mark): as wrapped iseq is isolated from the call stack, it needs to take care of its parent and ancestors, so that they do not become orphans. [ruby-core:72620] [Bug #11928]

Revision 53514 - 01/12/2016 03:17 PM - nobu (Nobuyoshi Nakada)

iseq.c: mark parents of wrapped iseq

- iseq.c (iseqw_mark): as wrapped iseq is isolated from the call stack, it needs to take care of its parent and ancestors, so that they do not become orphans. [ruby-core:72620] [Bug #11928]

Revision 53514 - 01/12/2016 03:17 PM - nobu (Nobuyoshi Nakada)

iseq.c: mark parents of wrapped iseq

- iseq.c (iseqw_mark): as wrapped iseq is isolated from the call stack, it needs to take care of its parent and ancestors, so that they do not become orphans. [ruby-core:72620] [Bug #11928]

Revision 53514 - 01/12/2016 03:17 PM - nobu (Nobuyoshi Nakada)

iseq.c: mark parents of wrapped iseq

- iseq.c (iseqw_mark): as wrapped iseq is isolated from the call stack, it needs to take care of its parent and ancestors, so that they do not become orphans. [ruby-core:72620] [Bug #11928]

Revision 22121544 - 01/13/2016 07:56 AM - nobu (Nobuyoshi Nakada)

iseq.c: mark parent iseq

- iseq.c (rb_iseq_mark): mark parent iseq to prevent dynamically generated iseq by eval from GC. [ruby-core:72620] [Bug #11928]

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@53524 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision 53524 - 01/13/2016 07:56 AM - nobu (Nobuyoshi Nakada)

iseq.c: mark parent iseq

- iseq.c (rb_iseq_mark): mark parent iseq to prevent dynamically generated iseq by eval from GC. [ruby-core:72620] [Bug #11928]

Revision 53524 - 01/13/2016 07:56 AM - nobu (Nobuyoshi Nakada)

iseq.c: mark parent iseq

- iseq.c (rb_iseq_mark): mark parent iseq to prevent dynamically generated iseq by eval from GC. [ruby-core:72620] [Bug #11928]

Revision 53524 - 01/13/2016 07:56 AM - nobu (Nobuyoshi Nakada)

iseq.c: mark parent iseq

- iseq.c (rb_iseq_mark): mark parent iseq to prevent dynamically generated iseq by eval from GC. [ruby-core:72620] [Bug #11928]

Revision 53524 - 01/13/2016 07:56 AM - nobu (Nobuyoshi Nakada)

iseq.c: mark parent iseq

- iseq.c (rb_iseq_mark): mark parent iseq to prevent dynamically generated iseq by eval from GC. [ruby-core:72620] [Bug #11928]

Revision d3c05ae7 - 03/29/2016 02:15 PM - naruse (Yui NARUSE)

merge revision(s) 53514,53524: [Backport #11928]

```
    * iseq.c (iseqw_mark): as wrapped iseq is isolated from the call
      stack, it needs to take care of its parent and ancestors, so
      that they do not become orphans.  [ruby-core:72620] [Bug #11928]
```

```
    * iseq.c (rb_iseq_mark): mark parent iseq to prevent dynamically
      generated iseq by eval from GC.  [ruby-core:72620] [Bug #11928]
```

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/branches/ruby_2_3@54405 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision 54405 - 03/29/2016 02:15 PM - naruse (Yui NARUSE)

merge revision(s) 53514,53524: [Backport #11928]

```
* iseq.c (iseqw_mark): as wrapped iseq is isolated from the call
  stack, it needs to take care of its parent and ancestors, so
  that they do not become orphans.  [ruby-core:72620] [Bug #11928]
```

```
* iseq.c (rb_iseq_mark): mark parent iseq to prevent dynamically
  generated iseq by eval from GC.  [ruby-core:72620] [Bug #11928]
```

History

#1 - 12/30/2015 08:17 PM - nobu (Nobuyoshi Nakada)

- Status changed from Open to Feedback

What exception will raise with --disable=did_you_mean command line option?

#2 - 01/05/2016 02:39 PM - hirokiraj (Jakub Jatczak)

I think i encountered similiar or even the same issue, managed to replicate in quite simple env with roda
https://github.com/hirokiraj/roda-2.3.0-bug

Control frame and backtrace -> https://gist.github.com/hirokiraj/83d917de90e0f60253ba
Crash diagnostic report -> https://gist.github.com/hirokiraj/4ef37abec72bec2f6ada

Will investigate what happens with --disable=did_you_mean

#3 - 01/05/2016 04:00 PM - yuki24 (Yuki Nishijima)

I was able to replicate it without did_you_mean.

1. Save this file as loader.rb:

```
-> { require_relative 'segfault' }.call
```

2. Save this file as segfault.rb:

```
class Segfault
  at_exit { Segfault.new.segfault }

  define_method 'segfault' do
    while true do
      (foo rescue $!).local_variables
    end
  end
end
```

3. Then run:

```
$ ruby --disable-gems loader.rb
```

#4 - 01/11/2016 03:06 PM - wanabe (_ wanabe)

- Status changed from Feedback to Open

- Assignee deleted (yuki24 (Yuki Nishijima))

With the just experimental patch, I didn't encounter SEGV.
I suspect about GC mark matter of T_IMEMO iseq, referenced by NameError#local_variables.

```
diff --git a/gc.c b/gc.c
index 874cb98..e34da14 100644
--- a/gc.c
+++ b/gc.c
@@ -2064,6 +2064,10 @@ obj_free(rb_objspace_t *objspace, VALUE obj)
 #endif
 #endif

+    if (BUILTIN_TYPE(obj) == T_IMEMO && imemo_type(obj) == imemo_iseq) {
+       return 1;
+    }
+
     switch (BUILTIN_TYPE(obj)) {
       case T_OBJECT:
        if (!(RANY(obj)->as.basic.flags & ROBJECT_EMBED) &&
```

I changed the status and the assignee because this seem not to be a did_you_mean matter.

#5 - 01/12/2016 03:16 PM - nobu (Nobuyoshi Nakada)

- Status changed from Open to Closed

Applied in changeset r53514.

---

iseq.c: mark parents of wrapped iseq

- iseq.c (iseqw_mark): as wrapped iseq is isolated from the call stack, it needs to take care of its parent and ancestors, so that they do not become orphans. [ruby-core:72620] [Bug #11928]

#6 - 01/13/2016 04:44 AM - wanabe (_ wanabe)

This script still causes SEGV.
Confirmed with ruby 2.4.0dev (2016-01-13 trunk 53518) [x86_64-darwin15] and ruby 2.4.0dev (2016-01-13 trunk 53518) [x86_64-linux].

```
->{
  ->{
    ->{
      eval <<-EOS
        class Segfault
          at_exit { Segfault.new.segfault }

          define_method :segfault do
            GC.disable
            0.step do |n|
              n.times do
```

```
                (foo rescue $!).local_variables
              end
              GC.start
            end
          end
        end
      EOS
    }.call
  }.call
}.call
```

#7 - 01/18/2016 12:53 AM - wanabe (_ wanabe)

_ wanabe wrote:

> This script still causes SEGV.

Fixed at r53524. Thanks!

#8 - 01/19/2016 03:16 AM - nobu (Nobuyoshi Nakada)

- Has duplicate Bug #12000: Crash report for 2.3.0 added

#9 - 02/17/2016 02:06 AM - wanabe (_ wanabe)

ruby-2.3 seems to have the bug as pointed by #12078.
I guess the issue should be set "2.3: REQUIRED" to make r53514 and r53524 backported, shouldn't this?

#10 - 02/17/2016 02:09 PM - amw (Adam Wróbel)

- Backport changed from 2.3: UNKNOWN to 2.3: REQUIRED

I've changed the backport flag as requested, but can't change ticket status. Hope a responsible party will be notified.

#11 - 02/18/2016 06:05 AM - yuki24 (Yuki Nishijima)

- Status changed from Closed to Open

- Backport changed from 2.3: REQUIRED to 2.1: DONTNEED, 2.2: DONTNEED, 2.3: REQUIRED

I'm re-opening this ticket as the fix needs to be backported to 2.3.0 as well.

#12 - 02/18/2016 06:22 AM - usa (Usaku NAKAMURA)

- Status changed from Open to Closed

Never change the status!
"Open" means that it's not fixed in trunk and then not be able to be backported yet.
"Closed" means "It's already fixed! Now it can be backported!"

#13 - 02/24/2016 02:51 PM - nobu (Nobuyoshi Nakada)

- Has duplicate Bug #12107: Segmentation fault at 0x000000000000b8 - did_you_mean added

#14 - 02/24/2016 02:59 PM - nobu (Nobuyoshi Nakada)

- Has duplicate Bug #12078: Segmentation fault in did_you_mean with ruby revision 53608 added

#15 - 03/29/2016 02:15 PM - naruse (Yui NARUSE)

- Backport changed from 2.1: DONTNEED, 2.2: DONTNEED, 2.3: REQUIRED to 2.1: DONTNEED, 2.2: DONTNEED, 2.3: DONE

ruby_2_3 r54405 merged revision(s) 53514,53524.

#16 - 04/11/2016 03:09 AM - nobu (Nobuyoshi Nakada)

- Has duplicate Bug #12265: did_you_mean spell checker crash added

#17 - 09/28/2016 05:25 AM - nobu (Nobuyoshi Nakada)

- Has duplicate Bug #12796: Segmentation fault at ~/.rvm/gems/ruby-2.3.0@global/gems/did_you_mean-1.0.0/lib/did_you_mean/spell_checkers/name_error_checkers/variable_name_checker.rb:10: [BUG] added

Files

did_you_mean_clean.log	300 KB	12/30/2015	amw (Adam Wróbel)
ruby_2015-12-27-133453_iMac.crash	26.4 KB	12/30/2015	amw (Adam Wróbel)