Ruby master - Bug #13612

Segmentation Fault

05/30/2017 10:33 AM - rovf (Ronald Fischer)

Status:	Closed		
Priority:	Normal		
Assignee:			
Target version:			
ruby -v:	ruby 2.3.3p222 (2016-11-21 revision 56859) [x86_64-cygwin]	**Backport:**	2.2: UNKNOWN, 2.3: UNKNOWN, 2.4: UNKNOWN

Description

I got a segmentation fault, fllowed by the message "You may have encountered a bug in the Ruby interpreter or extension libraries.". See attachment.

The exception occured in this piece of code:

```
  class EXPERT

    FIND_CHAIN_FOR_POS_FILTER_PIPELINE =
      [
        :no_filter,
        :filter_by_etkz_e,
        :filter_chain_in_chain,
      ].each

    def find_chain_for_pos(sw_lines_for_module, wsc_sw_lines)

      FIND_CHAIN_FOR_POS_FILTER_PIPELINE.rewind

      begin

        while NVP.multiple_grpids?(sw_lines_for_module)

          filter_method_symb = FIND_CHAIN_FOR_POS_FILTER_PIPELINE.next
# <----- This is line 30, where the exception reportedly occured.

          sw_lines_for_module = NVP.send(filter_method_symb, br, sw_lines_for_module)

        end

      rescue StopIteration

        LOG.trace "Could not find unique GRPID"

      end

    end
  end
```

The segmentation fault occurs only after this method has been executed repeatedly several hundred times.

History

#1 - 05/30/2017 11:41 AM - nobu (Nobuyoshi Nakada)

- Status changed from Open to Feedback

- Description updated

Please show the whole code to reproduce, not a part.
And 2.3.3 has been outdated, try 2.3.4, 2.4.1, or trunk.

#2 - 05/30/2017 01:34 PM - rovf (Ronald Fischer)

nobu (Nobuyoshi Nakada) wrote:

> Please show the whole code to reproduce, not a part.
> And 2.3.3 has been outdated, try 2.3.4, 2.4.1, or trunk.

Aside from the fact that my company unfortunately does not allow me to send you the whole application, as this is a strategic product, I think you wouldn't be too happy to debug with it anyway, as it is roughly 20000 lines of Ruby code (not counting the gems), which needs to run for 10 minutes or so until the bug appears.

I found meanwhile a more compact example, which moreover shows erroneous behaviour reproducibly and does not contain any , but before I do this, I will follow your advice and verify, whether the error still occurs with a newer version.

It's strange that so many new versions are available already. I made a fresh install less than one months ago.....

#3 - 05/30/2017 01:44 PM - nobu (Nobuyoshi Nakada)

rovf (Ronald Fischer) wrote:

> Aside from the fact that my company unfortunately does not allow me to send you the whole application, as this is a strategic product, I think you wouldn't be too happy to debug with it anyway, as it is roughly 20000 lines of Ruby code (not counting the gems), which needs to run for 10 minutes or so until the bug appears.

Of course, we don't want the whole **application**.
What we need is **runnable** code fragment.

> I found meanwhile a more compact example, which moreover shows erroneous behaviour reproducibly and does not contain any , but before I do this, I will follow your advice and verify, whether the error still occurs with a newer version.

Thank you so much in advance.

> It's strange that so many new versions are available already. I made a fresh install less than one months ago.....

2.3.4 was released at 2017-03-30.
2.4.1 was released at 2017-03-22.

#4 - 05/30/2017 01:49 PM - rovf (Ronald Fischer)

- File segf_test.rb added

nobu (Nobuyoshi Nakada) wrote:

> And 2.3.3 has been outdated, try 2.3.4, 2.4.1, or trunk.

I just verified: There is no newer port available for Cygwin, at least not on those mirrors, which Cygwin offers.

However, I have included a highly simplified version of the original code, which crashes. The crash looks different, in that we don't get a backtrace (like in the original case), but like in the original version, it happens only after the program runs for a while, and with my tests, it terminates always at the same point. If you try it, please make sure that you install the 64-bit version of Cygwin, not the 32-bit. I'm running it on Windows 7. If possible, try to execute it on the same platform.

The program is supposed to do a counting loop, for instance when invoked by

```
ruby segf_test.rb 5
```

it counts from 0 to 4.

If I run the program with

```
ruby segf_test.rb 99999
```

the program terminates already at count 12685.

One notable thing is that the behaviour is very sensitive to what's in the source code. For example, you will see that the code contains a class named Cnt, which is never used; but when I remove the (unnecessary) class definition, the bug does not occur. It counts correctly up to 99998.

Also, I found that if I just add another "puts" at the very end of the code, the bug also disappears. This suggests that the bug has something to do with how the compiler arranges the code.

#5 - 05/30/2017 02:19 PM - rovf (Ronald Fischer)

- File segf_test_improved.rb added

I just made a minor change, which shows better where the error occurs (segf_test_improved.rb). The program needs to be operated in the same way as segf_test.rb which I described before.

Two things are notable:

1. Like in the backtrace which I provided initially, the program crashes at the statement
   filter_method_symb = FIND_CHAIN_FOR_POS_FILTER_PIPELINE.next
   (the "puts 'before'" before the line is shown, the "puts 'after'" is not shown.

2. When I run this improved test script, it aborts already (reproduceably) in the iteration 4937.

What puzzles me most, is how sensitive this bug is related to program layout. For example, if I insert just one empty line before the first 'class' definition, the bug disappears! This is insane!!!!

#6 - 05/30/2017 02:55 PM - rovf (Ronald Fischer)

There is one more observation I would like to add; I don't know whether or not this is important:

When the segmentation fault occured the first time in our original application with Cygwin MRI Ruby, I reexecuted the whole application with JRuby, because I suspected that I might run out of memory due to a memory leak, and while JRuby certainly does not have the the same memory footprint as MRI Ruby, given the huge amount of input data, a leak caused by my own programming would likely show up in JRuby too.

Too my surprise, the JRuby execution already crashed after a couple of seconds, not after many minutes, and it was (of course) not a Segmentation Fault, but a NoMethodError: Due to a programming error, a variable v was nil, and when I applied v.times {....}, it crashed of course. Now what was really surprising, that it **should** have crashed under MRI Ruby too immediately. Same program, same data, and an obvious bug! Anyway, I fixed the bug, and executed it again on MRI Ruby - and no segmentation fault.

After some time, I run the program again with a different set of input data. Again, MRI Ruby got a segmentation fault. Running it with JRuby showed, like before, a similar programming error, just in a completely different module. Here too, the culprit was an object which was nil, and where I forgot to catch this fact. And, like before, after fixing this bug, running on MRI Ruby did not crash anymore!

Now I had two cases where MRI Ruby - at least the Cygwin port - did not manage to catch the fact that I am mistakingly was using a nil object. I have no idea, what Ruby did **instead**. It's supposed to throw an exception in such a case, and **usually** it DOES throw an exception. So far, it misbehaved only in this two cases.

#7 - 05/31/2017 04:44 AM - duerst (Martin Dürst)

rovf (Ronald Fischer) wrote:

> nobu (Nobuyoshi Nakada) wrote:
>
> > And 2.3.3 has been outdated, try 2.3.4, 2.4.1, or trunk.
>
> I just verified: There is no newer port available for Cygwin, at least not on those mirrors, which Cygwin offers.

Various distributions have various delays because they want to make sure they only ship widely used code, or because they don't have the time to follow every new release. Cygwin doesn't spend that much time on Ruby packaging, I guess.

But you can easily compile Ruby trunk on cygwin; I do that almost every morning. Just don't try to run the whole test suite, because there are some tests that don't work on Cygwin :-(.

#8 - 08/25/2019 06:13 PM - jeremyevans0 (Jeremy Evans)

- Status changed from Feedback to Closed

Files

ruby_segf.txt	18.5 KB	05/30/2017	rovf (Ronald Fischer)
segf_test.rb	1.49 KB	05/30/2017	rovf (Ronald Fischer)
segf_test_improved.rb	1.54 KB	05/30/2017	rovf (Ronald Fischer)