Ruby master - Bug #3555
segvs since r28570
07/10/2010 07:12 AM - tenderlovemaking (Aaron Patterson)

Status: Closed
Priority: Normal
Assignee: authorNari (Narihiro Nakamura)
Target version: 2.0.0
Backport:
ruby -v:
ruby 1.9.3dev (2010-07-09 trunk 28592) [x86_64-darwin10.3.1]

Description
begin
I'm seeing segv's when running the rails tests since r28570. r28569 does not segv.
I've attached the trace left from Ruby as well as a gdb backtrace. I will attempt to reduce the problem in the mean time.
r28570 seems to be a pure ruby change, so ruby should not segv. I hope we can fix the root cause rather than revert r28570.
=end

History
#1 - 07/10/2010 07:24 AM - tenderlovemaking (Aaron Patterson)

begin
On Sat, Jul 10, 2010 at 07:12:22AM +0900, Aaron Patterson wrote:

Bug #3555: segvs since r28570
http://redmine.ruby-lang.org/issues/show/3555

Author: Aaron Patterson
Status: Open, Priority: Normal
ruby -v: ruby 1.9.3dev (2010-07-09 trunk 28592) [x86_64-darwin10.3.1]
I'm seeing segv's when running the rails tests since r28570. r28569 does not segv.
I've attached the trace left from Ruby as well as a gdb backtrace. I will attempt to reduce the problem in the mean time.
r28570 seems to be a pure ruby change, so ruby should not segv. I hope we can fix the root cause rather than revert r28570.

I forgot:

Steps to reproduce:

$ git clone http://github.com/rails/rails.git
$ cd rails
$ gem install bundler --pre
$ bundle install
$ cd activerecord
$ rake test_sqlite3

or

$ ruby -w -I"lib:test/test/connections/native_sqlite3" test/cases/adapter_test.rb

--
Aaron Patterson
http://tenderlovemaking.com/

Attachment: (unnamed)
=end

#2 - 07/12/2010 11:29 PM - mame (Yusuke Endoh)

begin
Hi,

09/15/2021
On Sat, Jul 10, 2010 at 07:12:22AM +0900, Aaron Patterson wrote:

Bug #3555: segvs since r28570
http://redmine.ruby-lang.org/issues/show/3555

Author: Aaron Patterson
Status: Open, Priority: Normal
ruby -v: ruby 1.9.3dev (2010-07-09 trunk 28592) [x86_64-darwin10.3.1]

I'm seeing segv's when running the rails tests since r28570. ?r28569 does not segv.

I've attached the trace left from Ruby as well as a gdb backtrace. ?I will attempt to reduce the problem in the mean time.

r28570 seems to be a pure ruby change, so ruby should not segv. ?I hope we can fix the root cause rather than revert r28570.

I forgot:

Steps to reproduce:

$ git clone http://github.com/rails/rails.git
$ cd rails
$ gem install bundler --pre
$ bundle install
$ cd activerecord
$ rake test_sqlite3

Unfortunately, works for me with r28620 [i686-linux].

http://www.atdot.net/sp/view/oh7q5l

--

Yusuke Endoh mame@tsg.ne.jp
=end

#3 - 07/13/2010 02:25 AM - tenderlovemaking (Aaron Patterson)

=begin
On Mon, Jul 12, 2010 at 11:29:32PM +0900, Yusuke Endoh wrote:

Issue #3555 has been updated by Yusuke Endoh.

Hi,

2010/7/10 Aaron Patterson aaron@tenderlovemaking.com:

On Sat, Jul 10, 2010 at 07:12:22AM +0900, Aaron Patterson wrote:

Bug #3555: segvs since r28570
http://redmine.ruby-lang.org/issues/show/3555

Author: Aaron Patterson
Status: Open, Priority: Normal
ruby -v: ruby 1.9.3dev (2010-07-09 trunk 28592) [x86_64-darwin10.3.1]

I'm seeing segv's when running the rails tests since r28570. ?r28569 does not segv.

I've attached the trace left from Ruby as well as a gdb backtrace. ?I will attempt to reduce the problem in the mean time.

r28570 seems to be a pure ruby change, so ruby should not segv. ?I hope we can fix the root cause rather than revert r28570.

I forgot:

Steps to reproduce:

$ git clone http://github.com/rails/rails.git
$ cd rails
$ gem install bundler --pre
$ bundle install
$ cd activerecord
$ rake test_sqlite3
Unfortunately, works for me with r28620 [i686-linux].

http://www.atdot.net/sp/view/oh7q5l

I've been able to reduce the problem to this:

ruby -e 'GC.stress = true; $-w = true; require "rubygems"; require "yaml"

http://atdot.net/sp/view/w6gg5l

I will continue to reduce further.

--
Aaron Patterson
http://tenderlovemaking.com/

Attachment: (unnamed)
=end

#4 - 07/13/2010 04:11 AM - runpaint (Run Paint Run Run)

=begin
Just to confirm:

runt@paint:~ → ruby -e 'GC.stress = true; $-w = true; require "rubygems"; require "yaml"

/usr/local/lib/ruby/1.9.1/rubygems.rb:1062: warning: method redefined; discarding old gem
internal:gem_prelude:15: warning: previous definition of gem was here
/usr/local/lib/ruby/1.9.1/rubygems/source_index.rb:360: warning: assigned but unused variable - name
/usr/local/lib/ruby/1.9.1/rubygems/specification.rb:674: warning: assigned but unused variable - default_value
/usr/local/lib/ruby/1.9.1/rubygems/specification.rb:706: warning: assigned but unused variable - yaml
/usr/local/lib/ruby/1.9.1/rubygems/custom_require.rb:7: warning: loading in progress, circular require considered harmful - 
/usr/local/lib/ruby/1.9.1/rubygems.rb
from -e:1:in <main>
from -e:1:in require'
from /usr/local/lib/ruby/1.9.1/rubygems.rb:1119:in <top (required)>::1:in require'
from /usr/local/lib/ruby/1.9.1/rubygems/custom_require.rb:7:in require'
from /usr/local/lib/ruby/1.9.1/rubygems/custom_require.rb:7:in <require

*** glibc detected *** ruby: malloc(): smallbin double linked list corrupted: 0x09165618 ***
======== Backtrace: ========
/lib/tls/i686/cmov/libcrypt-2.11.1.so.6(+0x6b591)[0x3c3591]
/lib/tls/i686/cmov/libcrypt-2.11.1.so.6(+0x6e710)[0x3c6710]
/lib/tls/i686/cmov/libcrypt-2.11.1.so.6(libc_malloc+0x5c)[0x3c7f9c]
ruby[0x8197b4e]
ruby[0x8198352]
ruby[0x81982ff]
ruby[0x8198757]
ruby[0x81a4745]
ruby[0x8159c11]
ruby[0x815b347]
ruby[0x815f996]
ruby(rb_yield+0x50)[0x8160d90]
ruby[0x817920a]
ruby[0x8159c11]
ruby[0x815bfb8]
ruby[0x815f996]
ruby(rb_seq_eval+0x126)[0x815f6c6]
ruby[0x81a455f]
ruby(rb_require_safe+0x54a)[0x81a5bda]
ruby[0x8159c11]
ruby[0x815bfb8]
ruby[0x815f996]
ruby(rb_seq_eval_main+0x1c7)[0x815fd67]
ruby[0x805e3e2]
ruby(rubyrun_node+0x32)[0x805f02]
ruby[main+0x5a][0x805d1a]
/lib/tls/i686/cmov/libcrypt-2.11.1.so.6(libc_start_main+0xe6)[0x36ebd6]
ruby[0x805721]

======== Memory map: ========
00270000-00279000 r-xp 00000000 08:01 276341 /lib/tls/i686/cmov/libcrypt-2.11.1.so
00279000-0027a7000 r--p 00008000 08:01 276341 /lib/tls/i686/cmov/libcrypt-2.11.1.so
0027a000-0027b9000 rw-p 00009000 08:01 276341 /lib/tls/i686/cmov/libcrypt-2.11.1.so
0027b000-0027c2000 rw-p 00000000 00:00 0

09/15/2021
Aborted
run@paint:~ → ruby -v
ruby 1.9.3dev (2010-07-12 trunk 28620) [i686-linux]
=end

#5 - 07/13/2010 07:14 AM - nobu (Nobuyoshi Nakada)

=begin
Hi,
At Tue, 13 Jul 2010 04:11:32 +0900, Run Paint Run Run Run wrote in [ruby-core:31221]:

  run@paint:~ → ruby -e'GC.stress = true; $-w = true; require "rubygems"; require "yaml"'

I could reproduce it on trunk, but not on 1.9.2 with or without backporting r28570. So I suspect it is related to lazy-sweep now.

Also,
At Thu, 8 Jul 2010 02:57:59 +0900, Roger Pack wrote in [ruby-core:31108]:

C:/installs/ruby_trunk_installed/lib/ruby/gems/1.9.1/specifications/rdoc-data-2.5.1.gemspec:18:

I could reproduce it on trunk, but not on 1.9.2 with or without backporting r28570. So I suspect it is related to lazy-sweep now.

Also,
This might be caused by the same bug. 0x230bd19 is a Fixnum in normal, so it can't be This seems heap_slot is broken/discarded.

--
Nobu Nakada
=end

#6 - 07/13/2010 02:36 PM - authorNari (Narihiro Nakamura)
=begin
2010-07-13 07:14 Nobuyoshi Nakada nobu@ruby-lang.org:

Hi,
2010年7月13日7:14 Nobuyoshi Nakada nobu@ruby-lang.org:

At Tue, 13 Jul 2010 04:11:32 +0900,
Run Paint Run Run wrote in [ruby-core:31221]:

  run@paint:~ → ruby -e'GC.stress = true; $-w = true; require "rubygems"; require "yaml"

I could reproduce it on trunk, but not on 1.9.2 with or without backporting r28570. So I suspect it is related to lazy-sweep now.

I'll check the lazy-sweep on trunk.

Also,

At Thu, 8 Jul 2010 02:57:59 +0900,
Roger Pack wrote in [ruby-core:31108]:

  C:/installs/ruby_trunk_installed/lib/ruby/gems/1.9.1/specifications/rdoc-data-2.5.1.gemspec:18:
  [BUG] gc_sweep(): unknown data type 0x0(0x230bd19)

This might be caused by the same bug. 0x230bd19 is a Fixnum in normal, so it can't be This seems heap_slot is broken/discarded.

--
Nobu Nakada

--
Narihiro Nakamura (nari)
=end

#7 - 07/13/2010 11:19 PM - mame (Yusuke Endoh)
=begin
I agree with Nobu. I set the target to 1.9.x. If anyone succeeds to reproduce this with 1.9.2, please notify us.

--
Yusuke Endoh mame@tsg.ne.jp
=end

#8 - 07/20/2010 01:47 PM - authorNari (Narihiro Nakamura)
=begin
I found memory error on trunk:

$ cat /tmp/t.rb

09/15/2021
module Kernel
  def require
  end
end
GC.start
GC.stress = true
10.times{ "a".dup }
end

$ cat /tmp/f.rb
require "/tmp/t"

$ valgrind ./ruby /tmp/f.rb

==11321== Memcheck, a memory error detector
==11321== Copyright (C) 2002-2009, and GNU GPL'd, by Julian Seward et al.
==11321== Using Valgrind-3.6.0.SVN-Debian and LibVEX; rerun with -h for copyright info
==11321== Command: ./ruby /tmp/f.rb
==11321==
==11321== Invalid write of size 1
==11321==    at 0x814AF5D: rb_thread_mark (vm.c:1651)
==11321==    by 0x8065905: gc_mark_children (gc.c:1752)
==11321==    by 0x80654A2: gc_mark (gc.c:1546)
==11321==    by 0x80654DC: rb_gc_mark (gc.c:1552)
==11321==    by 0x814AB63: vm_mark_each_thread_func (vm.c:1475)
==11321==    by 0x80F0F2DD: st_foreach (st.c:747)
==11321==    by 0x814ABD0: rb_vm_mark (vm.c:1498)
==11321==    by 0x8065905: gc_mark_children (gc.c:1752)
==11321==    by 0x80654A2: gc_mark (gc.c:1546)
==11321==    by 0x80654DC: rb_gc_mark (gc.c:1552)
==11321==    by 0x806682D: gc_lazy_sweep (gc.c:2067)
==11321==    by 0x814AB64: Address 0x43928e4 is 4 bytes inside a block of size 20 free'd
==11321==    at 0x4024B3A: free (vg_replace_malloc.c:366)
==11321==    by 0x8064203: vm_xfree (gc.c:752)
==11321==    by 0x806435A: ruby_xfree (gc.c:800)
==11321==    by 0x8142E70: rb_free_method_entry (vm_method.c:185)
==11321==    by 0x8142D5A: rb_sweep_method_entry (vm_method.c:151)
==11321==    by 0x8066304: after_gc_sweep (gc.c:2020)
==11321==    by 0x806662C: gc_lazy_sweep (gc.c:2065)
==11321==    by 0x80649A6: rb_newobj_from_heap (gc.c:1044)
==11321==    by 0x8064A5F: rb_newobj (gc.c:1128)
==11321==    by 0x80F1EF6: str_alloc (string.c:378)
==11321==    by 0x80F3C3F2: rb_str_resurrect (string.c:901)
==11321==    by 0x813D7C1: vm_exec_core (insns.def:379)
==11321==
==11321== HEAP SUMMARY:
==11321==     in use at exit: 384,939 bytes in 14,320 blocks
==11321== total heap usage: 16,121 allocs, 1,801 frees, 1,528,828 bytes allocated
==11321==
==11321== LEAK SUMMARY:
==11321== definitely lost: 46,603 bytes in 1,639 blocks
==11321== indirectly lost: 198,711 bytes in 7,706 blocks
==11321== possibly lost: 164 bytes in 7 blocks
==11321== still reachable: 139,461 bytes in 4,968 blocks
==11321== suppressed: 0 bytes in 0 blocks
==11321== Run with --leak-check=full to see details of leaked memory
==11321== For counts of detected and suppressed errors, rerun with: -v
==11321== ERROR SUMMARY: 2 errors from 1 contexts (suppressed: 29 from 10)

I will fix it.

=end

#9 - 07/20/2010 03:17 PM - Anonymous
  - Status changed from Assigned to Closed
  - % Done changed from 0 to 100

=begin
This issue was solved with changeset r28689.

09/15/2021
Aaron, thank you for reporting this issue. Your contribution to Ruby is greatly appreciated. May Ruby be with you.

=end

#10 - 07/21/2010 12:43 AM - tenderlovemaking (Aaron Patterson)
=begin
On Tue, Jul 20, 2010 at 01:47:09PM +0900, Narihiro Nakamura wrote:

Issue #3555 has been updated by Narihiro Nakamura.

I found memory error on trunk:

$ cat /tmp/t.rb
require "/tmp/f.rb"

module Kernel
def require
end
GC.start
GC.stress = true
10.times{ "a".dup }
end

$ cat /tmp/f.rb
require "/tmp/t"

$ valgrind ./ruby /tmp/f.rb

[snip]

I will fix it.

Thanks Nakamura!

--
Aaron Patterson
http://tenderlovemaking.com/

Attachment: (unnamed)
=end

Files

<table>
<thead>
<tr>
<th>File</th>
<th>Size</th>
<th>Date</th>
<th>Author</th>
</tr>
</thead>
<tbody>
<tr>
<td>rubybt.txt</td>
<td>1.65 KB</td>
<td>07/10/2010</td>
<td>tenderlovemaking (Aaron Patterson)</td>
</tr>
<tr>
<td>gdbbt.txt</td>
<td>8.78 KB</td>
<td>07/10/2010</td>
<td>tenderlovemaking (Aaron Patterson)</td>
</tr>
</tbody>
</table>