Ruby - Bug #14334

Segmentation fault after running rspec (ruby/2.5.0/erb.rb:885 / simplecov/source_file.rb:85)

01/08/2018 02:24 AM - jesselatham (Jesse Latham)

Status:	Closed		
Priority:	Normal		
Assignee:			
Target version:			
ruby -v:	ruby 2.5.0p0 (2017-12-25 revision 61468) [x86_64-darwin16]	**Backport:**	2.3: UNKNOWN, 2.4: UNKNOWN, 2.5: UNKNOWN

Description

Segmentation fault after running rspec (ruby/2.5.0/erb.rb:885 and simplecov-0.15.1/lib/simplecov/source_file.rb:85)

Intermittently receiving a segmentation fault as rspec test suite is finishing up run of about 400 tests on a Rails application. No good way to effectively reproduce this as I can also get through complete rspec test runs without issues.

Attached console output and log file from DiagnosticReports folder.

Never experienced the issue when working with ruby 2.4.2p198 (2017-09-14 revision 59899) [x86_64-darwin16]

Please let me know what else you might need to diagnose.

Related issues:

Related to Ruby - Bug #14561: Consistent 2.5.0 seg fault in GC, related to ac...	**Closed**

History

#1 - 01/08/2018 02:46 AM - jesselatham (Jesse Latham)

- Subject changed from Segmentation fault after running rspec (ruby/2.5.0/erb.rb:885 /) to Segmentation fault after running rspec (ruby/2.5.0/erb.rb:885 / simplecov/source_file.rb:85)

#2 - 01/31/2018 03:10 PM - chrismanderson (Chris Anderson)

- File ruby_2018-01-31-095536_Blackbriar.crash added

FWIW I am getting virtually the same exception in simplecov running minitest. Exception is attached and happy to provide any other info as well.

#3 - 01/31/2018 04:04 PM - PragTob (Tobias Pfeiffer)

Hello! Here is the corresponding simplecov issue: https://github.com/colszowka/simplecov/issues/652 (maintainer here) - I'm stumped as to what happens. Would happy to find the source but also to find a work around that I could publish :D

#4 - 02/01/2018 12:06 AM - hsbt (Hiroshi SHIBATA)

- Status changed from Open to Third Party's Issue

#5 - 02/02/2018 04:01 PM - PragTob (Tobias Pfeiffer)

Hi there!

I'm not quite sure exactly what the status "Third Party's issue" means, but I'm relatively certain that it's a bug in Ruby we managed to reproduce it without any simplecov involved purely using the coverage library: https://github.com/colszowka/simplecov/issues/652#issuecomment-362436552

Moreover, simplecov doesn't have any C-extensions - it's pure ruby only using the Coverage library. So simplecov shouldn't be able to produce a segfault at any rate.

Thanks!
Tobi

#6 - 02/02/2018 07:41 PM - normalperson (Eric Wong)

pragtob@gmail.com wrote:

> issue" means, but I'm relatively certain that it's a bug in
> Ruby we managed to reproduce it without any simplecov involved
> purely using the coverage library:

> https://github.com/colszowka/simplecov/issues/652#issuecomment-362436552

I couldn't reproduce it (using Coverage directly), but I suspect
the following patch is a fix for more aggressive compilers:

```
diff --git a/thread.c b/thread.c
index 23957eba09..3c8d77ddd8 100644
--- a/thread.c
+++ b/thread.c
@@ -5188,6 +5188,7 @@ rb_reset_coverages(void)
{
VALUE coverages = rb_get_coverages();
st_foreach(rb_hash_tbl_raw(coverages), reset_coverage_i, 0);
+    RB_GC_GUARD(coverages);
GET_VM()->coverages = Qfalse;
rb_remove_event_hook((rb_event_hook_func_t) update_line_coverage);
if (GET_VM()->coverage_mode & COVERAGE_TARGET_BRANCHES) {
```

Can you give it a shot? Thanks

#7 - 02/14/2018 02:11 AM - normalperson (Eric Wong)

Eric Wong wrote:

> pragtob@gmail.com wrote:
>
> > issue" means, but I'm relatively certain that it's a bug in
> > Ruby we managed to reproduce it without any simplecov involved
> > purely using the coverage library:
> > https://github.com/colszowka/simplecov/issues/652#issuecomment-362436552
>
> I couldn't reproduce it (using Coverage directly), but I suspect
> the following patch is a fix for more aggressive compilers:

Sorry, I think my patch was bogus (wrote it while barely awake that day :x)

However, I have a suspicion that this is related to
https://bugs.ruby-lang.org/issues/14357 and fixed by r62396 in
trunk since Coverage uses unoptimized objects as hash keys which
require rb_funcall (which allow triggering thread switches).

#8 - 02/23/2018 12:59 AM - dazuma (Daniel Azuma)

- Status changed from Third Party's Issue to Open

Some additional information. We're encountering the same segmentation fault, with the same C backtrace from inside the GC. (
https://github.com/GoogleCloudPlatform/google-cloud-ruby/issues/1979). We can reproduce it even when we remove simplecov from our bundle completely, and we are also not using the coverage library, so those libraries don't appear to be the direct culprit.

I also applied patch r62396 and can still reproduce this segfault, so this in fact appears to be a distinct problem from
https://bugs.ruby-lang.org/issues/14357

The top of the C backtrace is:

```
-- C level backtrace information -------------------------------------------
0   ruby                          0x00000001025aeee7 rb_vm_bugreport + 135
1   ruby                          0x00000001024346b8 rb_bug_context + 472
2   ruby                          0x00000001025234a1 sigsegv + 81
3   libsystem_platform.dylib          0x00007fff6d0fef5a _sigtramp + 26
4   ruby                          0x000000010244dba3 rb_gc_mark_machine_stack + 99
5   ruby                          0x000000010259df49 rb_execution_context_mark + 137
6   ruby                          0x000000010241c3bb cont_mark + 27
7   ruby                          0x0000000102458a12 gc_marks_rest + 146
8   ruby                          0x00000001024573d0 gc_start + 2816
9   ruby                          0x000000010245674f newobj_slowpath + 1055
10  ruby                           0x0000000102456304 newobj_slowpath_wb_protected + 20
11  ruby                           0x0000000102544716 rb_sym_to_s + 38
12  ruby                           0x00000001025a9480 vm_call0_body + 560
13  ruby                           0x00000001025aa208 rb_call0 + 152
14  ruby                           0x0000000102597aae rb_funcall_with_block + 62
15  ruby                           0x0000000102597fbe rb_yield + 158
(etc...)
```

#9 - 03/01/2018 02:04 AM - dazuma (Daniel Azuma)

I filed a separate issue https://bugs.ruby-lang.org/issues/14561 with a small repro case involving Enumerator and threads, which seems to yield the same C backtrace as the ones reported here.

#10 - 08/26/2019 04:46 PM - jeremyevans0 (Jeremy Evans)

- Related to Bug #14561: Consistent 2.5.0 seg fault in GC, related to accessing an enumerator in a thread added

#11 - 08/26/2019 04:46 PM - jeremyevans0 (Jeremy Evans)

- Status changed from Open to Closed

Files

ruby_2018-01-07-175630_Australia.crash	43.3 KB	01/08/2018	jesselatham (Jesse Latham)
segfault simplecov.rb console-output-2.txt	423 KB	01/08/2018	jesselatham (Jesse Latham)
segfault erb.rb console-output-1.txt	423 KB	01/08/2018	jesselatham (Jesse Latham)
ruby_2018-01-31-095536_Blackbriar.crash	45.1 KB	01/31/2018	chrismanderson (Chris Anderson)