Ruby master - Bug #18967
Segmentation fault in stackprof with Ruby 2.7.6
08/19/2022 03:42 PM - RubyBugs (A Nonymous)

| Status: | Third Party's Issue |
| Priority: | Normal |
| Assignee: | |
| Target version: | |
| Backport: | 2.7: UNKNOWN, 3.0: UNKNOWN, 3.1: UNKNOWN |

ruby -v:
ruby 2.7.6p219 (2022-04-12 revision c9c2245c0a) [x86_64-linux]

Description
Ruby 2.7.6 appears to have broken the stackprof gem, crashing on every run with a segmentation fault.

Please see the following issues reported on stackprof:
https://github.com/tmm1/stackprof/issues/185
https://github.com/tmm1/stackprof/issues/182

History
#1 - 08/19/2022 05:23 PM - mame (Yusuke Endoh)
- Status changed from Open to Feedback

Thank you for your report. At least we need the full trace, especially "C level backtrace information" section. Can you provide it?

#2 - 08/21/2022 06:54 PM - RubyBugs (A Nonymous)
- File stackprof_crash_ruby_2_7_6.txt.bz2 added

mame (Yusuke Endoh) wrote in #note-1:

Thank you for your report. At least we need the full trace, especially "C level backtrace information" section. Can you provide it?

Hello! Thank you. I have captured the full trace, and attached it to this reply, compressed with bzip2. Please let me know if you are able to view it?

#3 - 08/21/2022 07:23 PM - RubyBugs (A Nonymous)
Per @Eregon (Benoit Daloze) on https://github.com/tmm1/stackprof/issues/182#issuecomment-1221274946 -
It appears that the stackprof gem may have been segfaulting in CI for some time: https://github.com/tmm1/stackprof/actions/workflows/ci.yml

#4 - 08/22/2022 02:03 AM - mame (Yusuke Endoh)
- Status changed from Feedback to Open

Thank you for providing the full stack trace. In conclusion, I couldn't find the cause, sorry.
Ruby 2.7 is under the security maintenance phase (maybe EOL next March), so I'd recommend to use Ruby 3.0 or later as soon as possible.

Maybe relevant stack trace fragment:

```
/ruby/bin/../lib/libruby.so.2.7(sigsegv+0x4b) [0x7f8f1e64c0cb] signal.c:946
/lib/x86_64-linux-gnu/libc.so.6[0x7f8f1e0f0f10] [0x7f8f1e0f0f10]
/ruby/bin/../lib/libruby.so.2.7{imemo_type=0x0} [0x7f8f1e6a9b39] vm_insnhelper.c:588
/ruby/bin/../lib/libruby.so.2.7(check_method_entry) vm_insnhelper.c:594
/ruby/bin/../lib/libruby.so.2.7{rb_vm_frame_method_entry} vm_insnhelper.c:618
/ruby/bin/../lib/libruby.so.2.7{rb_profile_frames=0x78} [0x7f8f1e6c8380] vm_backtrace.c:1323
/usr/packages/ruby-2.7.6/gems/stackprof-0.2.20/lib/stackprof/stackprof.so(stackprof_buffer_sample+0x68) [0x7f8f0dedce678] stackprof.c:615
/usr/packages/ruby-2.7.6/gems/stackprof-0.2.20/lib/stackprof/stackprof.so(stackprof_buffer_sample) {null}=0
/usr/packages/ruby-2.7.6/gems/stackprof-0.2.20/lib/stackprof/stackprof.so(stackprof_signal_handler+0x0) [0x7f8f0dedce8ed] stackprof.c:740
/lib/x86_64-linux-gnu/libc.so.6[0x7f8f1e0f0f10] [0x7f8f1e0f0f10]
/lib/x86_64-linux-gnu/libpthread.so.0{pthread_cond_timedwait+0x289} [0x7f8f1dc83fb9]
```

Indeed the segfault occurs in the hook of stackprof. I looked at the code around this, but couldn't find any significant difference between 2.7 and 3.0. I think there may be a garbage VALUE in the VM stack, but I don't recall such a problem. Does anyone have any ideas?
#5 - 08/22/2022 10:54 AM - Eregon (Benoit Daloze)
- Status changed from Open to Third Party's Issue

I found the bug: https://github.com/tmm1/stackprof/pull/180/files#r951294711

#6 - 08/22/2022 10:57 AM - Eregon (Benoit Daloze)

i.e., rb_profile_frames is called at a random place and that's not supported on < 3.0.
(TBH even on >= 3.0 I wonder if it's truly supported, it seems pretty dangerous to call rb_profile_frames()/anything not-async-signal-safe from a signal handler)

#7 - 08/22/2022 11:18 AM - ivoanjo (Ivo Anjo)

As pointed by @Eregon (Benoit Daloze) I planned experimenting with this and raising it at some point -- the change in https://github.com/ruby/ruby/commit/0e276dc458f94d9d79a0f7c7669bde84abe80f21 did reorder things as far as the C source goes, but as far as I see it there really doesn't seem to be anything guaranteeing that the compiler won't reorder the write to ec->cfp with the actual initialization of the structure.

So... yeah this doesn't seem particularly safe at this moment.

(But it would be great if rb_profile_frames could indeed be made async-safe!)

#8 - 08/22/2022 11:29 AM - byroot (Jean Boussier)

The fix is here: https://github.com/tmm1/stackprof/pull/186 I'll try to get a release soon.

@RubyBugs (A Nonymous) in the meantime I suggest sticking to 0.2.19.

#9 - 08/22/2022 05:14 PM - tenderlovemaking (Aaron Patterson)

byroot (Jean Boussier) wrote in #note-8:

The fix is here: https://github.com/tmm1/stackprof/pull/186 I'll try to get a release soon.

@RubyBugs (A Nonymous) in the meantime I suggest sticking to 0.2.19.

I merged it and shipped 0.2.21. Thanks!

Files

| stackprof_crash_ruby_2_7_6.txt.bz2 | 69.1 KB | 08/21/2022 | RubyBugs (A Nonymous) |