Ruby master - Bug #13524

miniruby: [BUG] Segmentation fault at 0x0055e487e00230 ruby 2.4.1p111 (2017-03-22 revision 58053) [x86_64-linux]

04/28/2017 04:26 PM - rtacconi (Riccardo Tacconi)

<table>
<thead>
<tr>
<th>Status:</th>
<th>Third Party's Issue</th>
</tr>
</thead>
<tbody>
<tr>
<td>Priority:</td>
<td>Normal</td>
</tr>
<tr>
<td>Assignee:</td>
<td></td>
</tr>
<tr>
<td>Target version:</td>
<td>ruby -v: 2.4.1p111</td>
</tr>
<tr>
<td>Backport:</td>
<td>2.2: UNKNOWN, 2.3: UNKNOWN, 2.4: UNKNOWN</td>
</tr>
</tbody>
</table>

Description

Linux: Alpine Linux 3.5

./miniruby: [BUG] Segmentation fault at 0x0055e487e00230 ruby 2.4.1p111 (2017-03-22 revision 58053) [x86_64-linux]

-- Control frame information

```
c:0001 p:0000 s:0003 E:001c20 (none) [FINISH]
```

-- Machine register context

```
RIP: 0x00007f3c79126046 RBP: 0x000055e487e44d00 RSP: 0x00007ffd62f251e0
RAX: 0x00007f3c79608ae0 RBX: 0x000055e487e00000 RCX: 0x0000000000000000
RDX: 0x0000000000000001 RDI: 0x000055e487e44d00 RSI: 0x0000000000000037
R8: 0x000007f3c7916d680 R9: 0x0000000000000001 R10: 0x0000000000802005
R11: 0x0000000000000007a R12: 0x000007f3c78c0d000 R13: 0x000007f3c79608ae0
R14: 0x000007f3c7937c090 R15: 0x000007f3c7937c0a0 EFL: 0x0000000000010212
```

-- Other runtime information

- Loaded script: ./miniruby
- Loaded features:
  0 enumerator.so
  1 thread.rb
  2 rational.so
  3 complex.so
- Process memory map:

```
55e487930000-55e487c31000 r-xp 00000000 00:2e 18867 /root/ruby-2.4.1/miniruby
55e487e30000-55e487e36000 r-p 00300000 00:2e 18867 /root/ruby-2.4.1/miniruby
55e487e36000-55e487e37000 rw-p 00306000 00:2e 18867 /root/ruby-2.4.1/miniruby
55e487e37000-55e487e45000 rw-p 00000000 00:00 0
7f3c78a00000-7f3c79900000 rw-p 00000000 00:00 0
7f3c7911c000-7f3c79780000 r-xp 00000000 00:2e 12926 /usr/local/lib/libjemalloc.so.2
7f3c7978000-7f3c7979b000 r-p 0005c000 00:2e 12926 /usr/local/lib/libjemalloc.so.2
7f3c7979b000-7f3c797c000 rw-p 0005f000 00:2e 12926 /usr/local/lib/libjemalloc.so.2
7f3c7973c000-7f3c797d000 rw-p 00000000 00:00 0
7f3c7937d000-7f3c7945000 r-xp 00000000 00:2e 24 /lib/musl-x86_64.so.1
7f3c795ff000-7f3c79600000 ---p 00000000 00:00 0
7f3c7960000-7f3c79605000 rw-p 00000000 00:00 0
7f3c79605000-7f3c79606000 r-p 00088000 00:2e 24 /lib/musl-x86_64.so.1
7f3c79606000-7f3c79607000 rw-p 00089000 00:2e 24 /lib/musl-x86_64.so.1
7f3c79607000-7f3c7960a000 rw-p 00000000 00:00 0
7ffd6207000-7ffd6208000 rw-p 00000000 00:00 0 [stack]
7ffd6212000-7ffd6231000 r-p 00000000 00:00 0 [vvar]
7ffd6231000-7ffd6233000 r-xp 00000000 00:00 0 [vdata]
80000000-fffffffff601000 r-xp 00000000 00:00 0 [vsyscall]
```
You may have encountered a bug in the Ruby interpreter or extension libraries. Bug reports are welcome.

For details: [http://www.ruby-lang.org/bugreport.html](http://www.ruby-lang.org/bugreport.html)


---

**History**

#1 - 05/01/2017 01:06 AM - shyouhei (Shyouhei Urabe)
- Status changed from Open to Feedback

Is it possible for you to show us how to reproduce this?

#2 - 05/03/2017 10:24 AM - rtacconi (Riccardo Tacconi)

shyouhei (Shyouhei Urabe) wrote:

Is it possible for you to show us how to reproduce this?

If you try to build it with this Dockerfile it works: [https://github.com/docker-library/ruby/blob/master/2.4/alpine/Dockerfile](https://github.com/docker-library/ruby/blob/master/2.4/alpine/Dockerfile). However I modified that to use jemalloc, so I guess it has something to do with jemalloc. Here is some steps I am doing to compile jemalloc and ruby:

```
wget https://cache.ruby-lang.org/pub/ruby/2.4/ruby-2.4.1.tar.gz
tar xzfv ruby-2.4.1.tar.gz
apk update && apk add ca-certificates && update-ca-certificates && apk add openssl
JEMALLOC_VERSION=4.5.0
tar xzfv jemalloc.tar.gz && cd "jemalloc-$JEMALLOC_VERSION"
make && make install_bin install_include install_lib
cd ruby-2.4.1
./configure --with-jemalloc
```

So first you can create an Alpine image with that Dockerfile, than:

docker run image_id /bin/sh

and you run the commands above

#3 - 05/03/2017 01:33 PM - rtacconi (Riccardo Tacconi)

shyouhei (Shyouhei Urabe) wrote:

Is it possible for you to show us how to reproduce this?

Please, build it with this [[[https://github.com/docker-library/ruby/blob/master/2.4/alpine/Dockerfile]]]

#4 - 05/09/2017 04:08 PM - rtacconi (Riccardo Tacconi)

shyouhei (Shyouhei Urabe) wrote:

Is it possible for you to show us how to reproduce this?

Hi, any update?

#5 - 05/10/2017 03:21 PM - antoniobg (Antonio Borrero Granell)

I'm also having a segmentation fault using the official ruby image. In my case, the error only happens when executing `codeclimate-test-reporter`.

Here's the log: [https://gist.github.com/antoniobg/787521aee3d4e1019638a8cfe5b48fdd](https://gist.github.com/antoniobg/787521aee3d4e1019638a8cfe5b48fdd)

#6 - 05/11/2017 12:42 AM - wanabe (_ wanabe)

I think this is due to mixture of glibc and jemalloc. Here is a gdb output.

```
Thread 1 "miniruby" received signal SIGSEGV, Segmentation fault.
0x00007ff7bf03ca6 in je_arena_mapbitsp_read (mapbitsp=optimized out) at include/jemalloc/internal/arena.h:8 02:802 include/jemalloc/internal/arena.h: No such file or directory.
```

09/23/2021
(gdb) bt
#0 0x000007fff7b03ca6 in je_arena_mapbitsp_read (mapbitsp=<optimized out>) at include/jemalloc/internal/arena.h:802
#1 je_arena_mapbits_get (pageind=77, chunk=0x555555a00000) at include/jemalloc/internal/arena.h:809
#2 je_arena_mapbits_binind_get (pageind=77, chunk=0x555555a00000) at include/jemalloc/internal/arena.h:866
#3 je_arena_salloc (demote=false, ptr=0x555555a4dd00, tsdn=<optimized out>) at include/jemalloc/internal/arena.h:1387
#4 je_isalloc (demote=false, ptr=0x555555a4dd00, tsdn=<optimized out>)
   at include/jemalloc/internal/jemalloc_internal.h:1054
#5 ifree (tsd=<optimized out>, slow_path=false, tcache=0x7ffff760d000, ptr=0x555555a4dd00) at src/jemalloc.c:1891
#6 free (ptr=0x555555a4dd00) at src/jemalloc.c:2016
#7 0x000055555573426a in ruby_getcwd () at util.c:531

util.c:531  https://svn.ruby-lang.org/cgi-bin/viewvc.cgi/tags/v2_4_1/util.c?view=markup#l531
is call free(cwd) of jemalloc.
But cwd is a return value of getcwd(3) of glibc.

#7 - 05/22/2017 11:24 AM - shyouhei (Shyouhei Urabe)
- Status changed from Feedback to Third Party's Issue

Took me time for several reasons:

1. Had temporarily ran out of my docker hosts and had to prepare new one. This is my fault.
2. The ruby:2.4-alpine's Dockerfile you linked kindly deletes compiler and other toolchains from the image. This prevents the procedure you told us to work. Please provide us a reproducible bug report. I ended up writing my own Dockerfile for this.  
   https://github.com/shyouhei/docker-library/blob/master/%2313524/Dockerfile
3. I am able to reproduce the bug now, and agree with @wanbe. I think this is either Alpine's or jemalloc's fault (or both).

The tiny C source code that comes with my Dockerfile is a file to generate SEGV without the needs of ruby. So at least I can say this has nothing to do with ruby. Please report this to alpine, to check if they support jemalloc or not. If they think it's ruby who's wrong, please tell us so. I'll take a deeper look at it.

#8 - 04/09/2018 09:59 AM - thomasfedb (Thomas Drake-Brockman)
Alpine package jemalloc in their main repo, so it would seem that they support it to some extent:
https://pkgs.alpinelinux.org/package/v3.7/main/x86/jemalloc

#9 - 04/24/2018 07:35 PM - mperham (Mike Perham)
All, I modified Shyouhei's Dockerfile to use alpine3.7 and jemalloc 5.0.1 and it no longer segfaults:

FROM ruby:2.4.4-alpine3.7

Step 8/10 : ADD segv.c /tmp/segv.c
---> be86245150aa
Step 9/10 : RUN gcc -ljemalloc segv.c && ./a.out; true
   ---> Running in 2454384e2cfc
   Removing intermediate container 2454384e2cfc
   ---> ffb021513ffa
Step 10/10 : CMD gdb a.out core
   ---> Running in 039469e738f9
   Removing intermediate container 039469e738f9
   ---> 01fdbg4cf3190
   Successfully built 01fdbg4cf3190

I would suggest trying Alpine 3.7 and see if you still have problems.

#10 - 10/12/2018 08:49 AM - t_anjan (Anjan T)
I used this Docker file (https://github.com/t-anjan/ruby/blob/jemalloc/2.4/alpine3.7/Dockerfile) which uses Alpine 3.7 and jemalloc 5.0.1 from the alpine repositories.

It still threw the same error.

./miniruby: [BUG] Segmentation fault at 0x000000000001ec5b8
ruby 2.4.4p296 (2018-03-28 revision 63013) [x86_64-linux-musl]

-- Control frame information -----------------------------------------------
c:0001 p:0000 s:0000 E:00500 (none) [FINISH]

-- Machine register context -------------------------------
You may have encountered a bug in the Ruby interpreter or extension libraries. Bug reports are welcome.

For details: http://www.ruby-lang.org/bugreport.html
You may have encountered a bug in the Ruby interpreter or extension libraries. Bug reports are welcome.
For details: http://www.ruby-lang.org/bugreport.html

Any suggestions?