Ruby - Bug #20452

Ruby 3.3 on Alpine Linux results in a relatively shallow SystemStackError exception

04/24/2024 01:41 PM - Earlopain (Earlopain _)

Status:	Open		
Priority:	Normal		
Assignee:			
Target version:			
ruby -v:	3.3.1	**Backport:**	3.0: UNKNOWN, 3.1: UNKNOWN, 3.2: UNKNOWN, 3.3: UNKNOWN

Description

This is a redo of https://bugs.ruby-lang.org/issues/14387, reported against a non-eol version. The same issue still applies on recent rubies, though I personally only have tested 3.3 and 3.2:

```
n = 100000
res = {}
1.upto(n).to_a.inject(res) do |r, i|
  r[i] = {}
end

def f(x)
  x.each_value { |v| f(v) }
end

f(res)
```

The patch from https://bugs.ruby-lang.org/issues/14387#note-13 still works, though as per https://bugs.ruby-lang.org/issues/14387#note-16 it can't be accepted as is.

With the patch applied:

```
(irb):8:in `block in f': stack level too deep (SystemStackError)
        from (irb):8:in `each_value'
        from (irb):8:in `f'
        from (irb):8:in `block in f'
        from (irb):8:in `each_value'
        from (irb):8:in `f'
        from (irb):8:in `block in f'
        from (irb):8:in `each_value'
        from (irb):8:in `f'
         ... 11549 levels...
        from /usr/local/lib/ruby/gems/3.3.0/gems/irb-1.11.0/exe/irb:9:in `<top (required)>'
        from /usr/local/bin/irb:25:in `load'
        from /usr/local/bin/irb:25:in `<main>'
```

Without the patch:

```
(irb):8:in `each_value': stack level too deep (SystemStackError)
        from (irb):8:in `f'
        from (irb):8:in `block in f'
        from (irb):8:in `each_value'
        from (irb):8:in `f'
        from (irb):8:in `block in f'
        from (irb):8:in `each_value'
        from (irb):8:in `f'
        from (irb):8:in `block in f'
         ... 286 levels...
        from /usr/local/lib/ruby/gems/3.3.0/gems/irb-1.11.0/exe/irb:9:in `<top (required)>'
        from /usr/local/bin/irb:25:in `load'
        from /usr/local/bin/irb:25:in `<main>'
```

To reproduce, I have removed lines 102-105 from

History

#1 - 04/24/2024 02:07 PM - nobu (Nobuyoshi Nakada)

Does configure with ac_cv_func_pthread_get_stackaddr_np=no change something?

#2 - 04/24/2024 02:24 PM - Earlopain (Earlopain _)

nobu (Nobuyoshi Nakada) wrote in #note-1:

> Does configure with ac_cv_func_pthread_get_stackaddr_np=no change something?

Thank you for your reply. I modified export ac_cv_func_isnan=yes ac_cv_func_isinf=yes to export ac_cv_func_pthread_get_stackaddr_np=no ac_cv_func_isnan=yes ac_cv_func_isinf=yes and rebuild but the StackError occured at basically the same depth.

#3 - 04/24/2024 10:16 PM - hsbt (Hiroshi SHIBATA)

- Tags set to musl, alpine

- Status changed from Open to Feedback

Unfortunately, there is no active maintainer for musl or alpine platform.

I tagged them to musl. We welcome patch for them.

#4 - 04/25/2024 11:38 AM - nobu (Nobuyoshi Nakada)

make DEFS=-DMAINSTACKADDR_AVAILABLE=0 thread.o seems working.

#5 - 04/25/2024 12:05 PM - nobu (Nobuyoshi Nakada)

Does this help you?
https://github.com/nobu/ruby/tree/mainstackaddr

#6 - 04/25/2024 02:50 PM - Earlopain (Earlopain _)

nobu (Nobuyoshi Nakada) wrote in #note-5:

> Does this help you?
> https://github.com/nobu/ruby/tree/mainstackaddr

Thank you. I tried it out with your patch but something is missing still. Building with make DEFS=-DMAINSTACKADDR_AVAILABLE=0 from your comment above works wonderfully though.

I believe HAVE_PTHREAD_GETATTR_NP may be defined? In configure.log I see HAVE_PTHREAD_GETATTR_NP=1 a few times.
https://git.musl-libc.org/cgit/musl/tree/src/thread/pthread_getattr_np.c

#7 - 09/20/2024 12:47 PM - nobu (Nobuyoshi Nakada)

Sorry to be late.
I updated the PR to see HAVE_PTHREAD_GETATTR_NP instead of HAVE_PTHREAD_GET_STACKADDR_NP.

#8 - 09/21/2024 08:18 AM - Earlopain (Earlopain _)

nobu (Nobuyoshi Nakada) wrote in #note-7:

> Sorry to be late.
> I updated the PR to see HAVE_PTHREAD_GETATTR_NP instead of HAVE_PTHREAD_GET_STACKADDR_NP.

Hi nobu, thank you very much for getting back to this.

I tried out your revised PR patch and can confirm that this resolves the issue. I ran make test-all to make sure things are fine and it looks good.

#9 - 11/11/2024 10:01 AM - Earlopain (Earlopain _)

It would be nice if this could make it for 3.4. Would that be possible, or are you not yet confident in the patch? 3.5-dev would be fine too, just wondering.

#10 - 11/11/2024 05:45 PM - alanwu (Alan Wu)

- Status changed from Feedback to Open

#11 - 02/14/2025 09:47 AM - Earlopain (Earlopain _)

Could the PR be considered now? I would very much appreciate it, now that there is no release imminent.

#10 - 11/11/2024 05:45 PM - alanwu (Alan Wu)

- Status changed from Feedback to Open

#11 - 02/14/2025 09:47 AM - Earlopain (Earlopain _)

Could the PR be considered now? I would very much appreciate it, now that there is no release imminent.