the build of ruby 2.3.3 fails on NetBSD/macppc (which is a powerpc architecture) with

... linking miniruby
generating encdb.h
./tool/generic_erb.rb:3: [BUG] Segmentation fault
ruby 2.3.3p222 (2016-11-21 revision 56859) [powerpc-netbsd]

-- Control frame information -----------------------------------------------
c:0003 p:---- s:0025 e:000024 CFUNC :require
c:0002 p:0017 s:0021 E:fffffffffffffe32c EVAL ./tool/generic_erb.rb:3 [FINISH]
c:0001 p:0000 s:0002 E:fffffffffffffd9f8 (none) [FINISH]

-- Ruby level backtrace information ----------------------------------------
./tool/generic_erb.rb:3:in `<main>'
./tool/generic_erb.rb:3:in `require'

-- Other runtime information -----------------------------------------------
* Loaded script: ./tool/generic_erb.rb

* Loaded features:
  0 enumerator.so
  1 thread.rb
  2 rational.so
  3 complex.so
  4 /usr/pkgsrc/lang/ruby23-base/work/ruby-2.3.3/lib/cgi/util.rb
  5 Jun

[NOTE]
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

*** Error code 1 (ignored)
*** Signal 6

Stop.

Compiling miniruby with -g and running the same command reveals where the SEGV is triggered:

{62} gmake -n encdb.h
echo generating encdb.h
./miniruby -I./lib -I. -I.ext/common ./tool/generic_erb.rb -c -o encdb.h ./template/encdb.h.tmpl
./enc enc
{63} gdb miniruby
GNU gdb (GDB) 7.3.1
Program received signal SIGSEGV, Segmentation fault.
(Switching to LWP 1)
iseq_compile_each (iseq=0xfda11cb0, ret=0xffffc420, node=0xf, poped=0)
  at compile.c:3713
    line = (int)nd_line(node);
(gdb) where
#0  iseq_compile_each (iseq=0xfda11cb0, ret=0xffffc420, node=0xf, poped=0)
  at compile.c:3713
#1  0x01973fcc in iseq_compile_each (iseq=0xfda11cb0, ret=0xffffc420, node=0xfdcd7b70, poped=0) at compile.c:3738
#2  0x0197ba18 in rb_iseq_compile_node (iseq=0xfda11cb0, node=0xfdcd18a0)
  at compile.c:590
#3  0x019821c4 in rb_iseq_new_with_opt (node=0xfdcd18a0, name=<optimized out>,
    path=<optimized out>, absolute_path=<optimized out>,
    first_lineno=<optimized out>, parent=<optimized out>,
    type=<optimized out>, option=<optimized out>) at iseq.c:474
#4  0x0196bd98 in new_child_iseq (iseq=0xfdcd12490, node=<optimized out>,
    name=<optimized out>, parent=<optimized out>, line_no=<optimized out>)
  at compile.c:1100
#5  0x01977388 in iseq_compile_each (iseq=0xfdcd12490, ret=0xffffc780,
    node=0xfdcd17f8, poped=1) at compile.c:5612
#6  0x0197ba18 in rb_iseq_compile_node (iseq=0xfdcd12490, node=0xfda12928)
  at compile.c:590
#7  0x019821c4 in rb_iseq_new_with_opt (node=0xfda12928, name=<optimized out>,
    path=<optimized out>, absolute_path=<optimized out>,
    first_lineno=<optimized out>, parent=<optimized out>,
    type=<optimized out>, option=<optimized out>) at iseq.c:474
#8  0x0196bd98 in new_child_iseq (iseq=0xfda12568, node=<optimized out>,
    name=<optimized out>, parent=<optimized out>, line_no=<optimized out>)
  at compile.c:1100
#9  0x01977388 in iseq_compile_each (iseq=0xfda12568, ret=0xffffc780,
    node=0xfda12760, poped=1) at compile.c:5612
#10 0x019773694 in iseq_compile_each (iseq=0xfda12760, ret=0xffffc780,
    node=0xfda122820, poped=0) at compile.c:3733
#11 0x019773694 in iseq_compile_each (iseq=0xfda12760, ret=0xffffc780,
    node=0xfda12748, poped=0) at compile.c:6002
#12 0x0197b6f4 in rb_iseq_compile_node (iseq=0xfda12568, node=0xfda12760)
  at compile.c:602
#13 0x019821c4 in rb_iseq_new_with_opt (node=0xfda12760, name=<optimized out>,
    path=<optimized out>, absolute_path=<optimized out>,
    first_lineno=<optimized out>, parent=<optimized out>,
    type=<optimized out>, option=<optimized out>) at iseq.c:474
#14 0x0198256c in rb_iseq_new_top (node=<optimized out>, name=<optimized out>,
    path=<optimized out>, absolute_path=<optimized out>,
    parent=<optimized out>) at iseq.c:436
#15 0x0185ef40 in rb_load_internal0 (th=0xfdc09800, fname=4257932088,
    wrap=<optimized out>) at load.c:616
#16 0x018610e8 in rb_require_internal (fname=4257932400, safe=0) at load.c:998
#17 0x01861458 in rb_require_safe (fname=4257932424, safe=<optimized out>)
  at load.c:1043
#18 0x01861588 in rb_f_require (obj=<optimized out>, fname=4257932424)
at load.c:824

#20 0x01985894 in call_cfunc_1 (func=<optimized out>, recv=<optimized out>,
    argc=<optimized out>, argv=<optimized out>) at vm_insnhelper.c:1475

#21 0x0198c070 in vm_call_cfunc_with_frame (calling=<optimized out>,
    reg_cfp=0xfdc96fc0, th=0xfdc09800, ci=<optimized out>, cc=<optimized out>)
    at vm_insnhelper.c:1642

#22 vm_call_cfunc (th=0xfdc09800, reg_cfp=0xfdc96fc0, calling=<optimized out>,
    ci=<optimized out>, cc=<optimized out>) at vm_insnhelper.c:1737

#23 0x019a33b4 in vm_call_method_each_type (th=0xfdc09800, cfp=0xfdc96fc0,
    calling=0xffffffffd464, ci=<optimized out>, cc=<optimized out>)
    at vm_insnhelper.c:2026

#24 0x019a3f58 in vm_call_method (cc=0xfdc0d820, ci=0xfdb7418c,
    calling=0xffffffffd464, cfp=0xfdc96fc0, th=0xfdc09800) at vm_insnhelper.c:2179

#25 vm_call_general (th=0xfdc09800, reg_cfp=0xfdc96fc0, calling=0xffffffffd464,
    ci=0xfdb7418c, cc=0xfdc0d820) at vm_insnhelper.c:2193

#26 0x01994928 in vm_exec_core (th=<optimized out>, initial=<optimized out>)
    at insns.def:994

#27 0x01998f98 in vm_exec (th=0xfdc09800) at vm.c:1650

#28 0x01859e5c in ruby_exec_internal (n=0xfdcb06f0) at eval.c:245

#29 0x0185b570 in ruby_exec_node (n=value has been optimized out)
    at eval.c:302

#30 0x0185deac in ruby_run_node (n=0xfdcb06f0) at eval.c:302

#31 0x01812c10 in main (argc=11, argv=0xffffdb34) at main.c:36

Suggestions for what to dig out next would be appreciated, ruby is
unknown ground for me.

The only slightly "odd" thing with NetBSD/powerpc is that it's big-endian.

Regards,

Håvard

History

#1 - 07/25/2019 11:42 PM - jeremyevans0 (Jeremy Evans)
- Status changed from Open to Closed