Ruby master - Bug #17052

Ruby with LTO enabled has issues with SIGSEGV handler

07/27/2020 12:27 PM - vo.x (Vit Ondruch)

Status:	Third Party's Issue		
Priority:	Normal		
Assignee:			
Target version:			
ruby -v:	ruby 2.7.1p83 (2020-03-31 revision a0c7c23c9c) [powerpc64le-linux]	**Backport:**	2.5: UNKNOWN, 2.6: UNKNOWN, 2.7: UNKNOWN

Description

Fedora aims to enable link time optimization (LTO) of packages in next release. The specific changes in configuration options are available here 1. Since that time, I observe following errors 2 at least on {aarch64,ppc64le} (and possibly also other architectures):

```
  1) Failure:
TestBugReporter#test_bug_reporter_add [/builddir/build/BUILD/ruby-2.7.1/test/-ext-/bug_reporter/te
st_bug_reporter.rb:22]:
pid 32395 killed by SIGSEGV (signal 11) (core dumped)
| -:1: [BUG] Segmentation fault at 0x000003e800007e8b
| ruby 2.7.1p83 (2020-03-31 revision a0c7c23c9c) [powerpc64le-linux]
|
| -- Control frame information -----------------------------------------------
| c:0003 p:---- s:0012 e:000011 CFUNC  :kill
| c:0002 p:0021 s:0006 e:000005 EVAL   -:1 [FINISH]
| c:0001 p:0000 s:0003 E:000f80 (none) [FINISH]
|
| -- Ruby level backtrace information ----------------------------------------
| -:1:in `<main>'
| -:1:in `kill'
|
| -- C level backtrace information -------------------------------------------
.
1. [2/2] Assertion for "stderr"
   | Expected /Sample bug reporter: 12345/
   | to match
   |   "-- Control frame information -----------------------------------------------\n"+
   |   "c:0003 p:---- s:0012 e:000011 CFUNC  :kill\n"+
   |   "c:0002 p:0021 s:0006 e:000005 EVAL   -:1 [FINISH]\n"+
   |   "c:0001 p:0000 s:0003 E:000f80 (none) [FINISH]\n\n"+
   |   "-- Ruby level backtrace information ----------------------------------------\n"+
   |   "-:1:in `<main>'\n"+
   |   "-:1:in `kill'\n\n"+
   |   "-- C level backtrace information -------------------------------------------\n"
   | after 4 patterns with 120 characters.
  2) Failure:
TestRubyOptions#test_segv_loaded_features [/builddir/build/BUILD/ruby-2.7.1/test/ruby/test_rubyopt
ions.rb:735]:
pid 38444 killed by SIGSEGV (signal 11) (core dumped)
| -e:1: [BUG] Segmentation fault at 0x000003e80000962c
| ruby 2.7.1p83 (2020-03-31 revision a0c7c23c9c) [powerpc64le-linux]
|
| -- Control frame information -----------------------------------------------
| c:0003 p:---- s:0012 e:000011 CFUNC  :kill
| c:0002 p:0016 s:0006 e:000005 BLOCK  -e:1 [FINISH]
| c:0001 p:0000 s:0003 E:002460 (none) [FINISH]
|
| -- Ruby level backtrace information ----------------------------------------
| -e:1:in `block in <main>'
| -e:1:in `kill'
|
| -- C level backtrace information -------------------------------------------
.
```

```
1. [2/2] Assertion for "stderr"
   | <""> expected but was
   | <"-- C level backtrace information -------------------------------------------\n">.
  3) Failure:
TestRubyOptions#test_segv_setproctitle [/builddir/build/BUILD/ruby-2.7.1/test/ruby/test_rubyoption
s.rb:749]:
pid 38451 killed by SIGSEGV (signal 11) (core dumped)
| -e:1: [BUG] Segmentation fault at 0x000003e800009633
| ruby 2.7.1p83 (2020-03-31 revision a0c7c23c9c) [powerpc64le-linux]
|
| -- Control frame information -----------------------------------------------
| c:0003 p:---- s:0012 e:000011 CFUNC  :kill
| c:0002 p:0029 s:0006 e:000005 EVAL   -e:1 [FINISH]
| c:0001 p:0000 s:0003 E:000480 (none) [FINISH]
|
| -- Ruby level backtrace information ----------------------------------------
| -e:1:in `<main>'
| -e:1:in `kill'
|
| -- C level backtrace information -------------------------------------------
.
1. [2/2] Assertion for "stderr"
   | <""> expected but was
   | <"-- C level backtrace information -------------------------------------------\n">.
  4) Failure:
TestRubyOptions#test_segv_test [/builddir/build/BUILD/ruby-2.7.1/test/ruby/test_rubyoptions.rb:729
]:
pid 38460 killed by SIGSEGV (signal 11) (core dumped)
| -e:1: [BUG] Segmentation fault at 0x000003e80000963c
| ruby 2.7.1p83 (2020-03-31 revision a0c7c23c9c) [powerpc64le-linux]
|
| -- Control frame information -----------------------------------------------
| c:0003 p:---- s:0012 e:000011 CFUNC  :kill
| c:0002 p:0015 s:0006 e:000005 EVAL   -e:1 [FINISH]
| c:0001 p:0000 s:0003 E:0006a0 (none) [FINISH]
|
| -- Ruby level backtrace information ----------------------------------------
| -e:1:in `<main>'
| -e:1:in `kill'
|
| -- C level backtrace information -------------------------------------------
.
1. [2/2] Assertion for "stderr"
   | <""> expected but was
   | <"-- C level backtrace information -------------------------------------------\n">.
Finished tests in 840.600443s, 25.0047 tests/s, 3238.9681 assertions/s.
21019 tests, 2722678 assertions, 4 failures, 0 errors, 70 skips
ruby -v: ruby 2.7.1p83 (2020-03-31 revision a0c7c23c9c) [powerpc64le-linux]
```

When I raised the issue on fedora-devel ML 3, there was suggestion that it might happen when signal handler modifies any global variable. Now I am not sure if that is the case. Can somebody confirm? Or investigate/fix this, please?

History

#1 - 07/27/2020 03:02 PM - vo.x (Vit Ondruch)

- Subject changed from Ruby with LTO enabled on {aarch64,ppc64le} architectures. to Ruby with LTO enabled has issues with SIGSEGV handler

Actually, the platform might not matter. It might fail just randomly, because now I have caught the same issue also on x86_64.

#2 - 08/20/2020 03:52 AM - shyouhei (Shyouhei Urabe)

- Status changed from Open to Third Party's Issue

Yes I can reproduce this on my machine.

```
% LC_ALL=C gdb --args ./miniruby -e'Process.kill("SIGSEGV",$$)'
GNU gdb (Ubuntu 8.2-0ubuntu1~18.04) 8.2
```

```
Copyright (C) 2018 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.
Type "show copying" and "show warranty" for details.
This GDB was configured as "x86_64-linux-gnu".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<http://www.gnu.org/software/gdb/bugs/>.
Find the GDB manual and other documentation resources online at:
    <http://www.gnu.org/software/gdb/documentation/>.

For help, type "help".
Type "apropos word" to search for commands related to "word"...
Reading symbols from ./miniruby...done.
(gdb) run
Starting program: ./miniruby -eProcess.kill\(\"SIGSEGV\",\$\$\)
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/lib/x86_64-linux-gnu/libthread_db.so.1".

Program received signal SIGSEGV, Segmentation fault.
0x00007ffff7a23237 in kill () at ../sysdeps/unix/syscall-template.S:78
78      ../sysdeps/unix/syscall-template.S: No such file or directory.
(gdb) bt
#0  0x00007ffff7a23237 in kill () at ../sysdeps/unix/syscall-template.S:78
#1  0x000055555571ec9b in rb_f_kill (argc=2, argv=0x7ffff7ecb048) at signal.c:480
#2  0x0000555555789c44 in vm_call_cfunc_with_frame (ec=0x555555add590, reg_cfp=0x7ffff7fcafa0, calling=<optimi
zed out>, cd=<optimized out>) at vm_insnhelper.c:2601
#3  0x000055555578d4d5 in vm_sendish (ec=0x555555add590, reg_cfp=0x7ffff7fcafa0, cd=0x555555bfaf30, block_hand
ler=<optimized out>, method_explorer=<optimized out>) at vm_callinfo.h:337
#4  0x000055555578ef98 in vm_exec_core (ec=0x555555add590, initial=11) at insns.def:799
#5  0x00005555557a4ee4 in rb_vm_exec (ec=0x555555add590, mjit_enable_p=1) at vm.c:1953
#6  0x00005555555ff1d0 in rb_ec_exec_node (ec=ec@entry=0x555555add590, n=n@entry=0x555555ae9958) at eval.c:296
#7  0x0000555555603533 in ruby_run_node (n=0x555555ae9958) at eval.c:354
#8  0x000055555557ee0f in main (argc=<optimized out>, argv=<optimized out>) at main.c:50
(gdb) c
Continuing.
-e:1: [BUG] Segmentation fault at 0x0000286b000022c1
ruby 2.8.0dev (2020-08-20T02:48:16Z flto 71753b7f6d) [x86_64-linux]

-- Control frame information -----------------------------------------------
c:0003 p:---- s:0012 e:000011 CFUNC  :kill
c:0002 p:0015 s:0006 e:000005 EVAL   -e:1 [FINISH]
c:0001 p:0000 s:0003 E:000a10 (none) [FINISH]

-- Ruby level backtrace information ----------------------------------------
-e:1:in `<main>'
-e:1:in `kill'

-- Machine register context ------------------------------------------------
 RIP: 0x00007ffff7a23237 RBP: 0x0000000000000002 RSP: 0x00007fffffffc2b8
 RAX: 0x0000000000000000 RBX: 0x0000000000000001 RCX: 0x00007ffff7a23237
 RDX: 0x0000555555adca01 RDI: 0x00000000000022c1 RSI: 0x000000000000000b
  R8: 0x0000555555b2a610  R9: 0x0000555555bed890 R10: 0x0000555555bed890
 R11: 0x0000000000000206 R12: 0x00007ffff7ecb048 R13: 0x00000000000022c1
 R14: 0x00000000000022c1 R15: 0x000000000000000b EFL: 0x0000000000000206

-- C level backtrace information -------------------------------------------

Program received signal SIGSEGV, Segmentation fault.
0x00005555557a9a90 in uleb128 (p=0x555555bd5428) at addr2line.c:1274
1274        uint64_t abbrev_number = uleb128(&reader->p);
(gdb) bt
#0  0x00005555557a9a90 in uleb128 (p=0x555555bd5428) at addr2line.c:1274
#1  di_read_die (reader=0x555555bd53f0, die=0x555555bd5310) at addr2line.c:1274
#2  0x00005555557bb65d in read_abstract_origin (line=0x555555bd5380, abstract_origin=<optimized out>, reader=0
x555555bd53f0) at addr2line.c:1486
#3  debug_info_read (traces=<optimized out>, offset=<optimized out>, lines=<optimized out>, num_traces=<optimi
zed out>, reader=<optimized out>) at addr2line.c:1550
#4  fill_lines (num_traces=num_traces@entry=16, check_debuglink=check_debuglink@entry=1, objp=objp@entry=0x555
555bd5d18, lines=lines@entry=0x555555c15480, offset=0, offset@entry=-1, traces=<optimized out>) at addr2line.c
:1763
#5  0x00005555557bc069 in rb_dump_backtrace_with_lines.constprop.0 (num_traces=16, traces=<optimized out>) at
addr2line.c:2150
#6  0x00005555557a8810 in rb_print_backtrace () at vm_dump.c:759
```

```
#7  0x000055555557acf21 in rb_vm_bugreport (ctx=<optimized out>) at vm_dump.c:955
#8  0x000055555555f59a4 in rb_bug_for_fatal_signal (default_sighandler=0x0, sig=11, ctx=0x555555bd6180, fmt=0x5
555557ff3fa "Segmentation fault at %p") at error.c:675
#9  0x000055555571ac29 in sigsegv (sig=11, info=0x555555bd62b0, ctx=0x555555bd6180) at signal.c:959
#10 <signal handler called>
#11 0x00007ffff7a23237 in kill () at ../sysdeps/unix/syscall-template.S:78
#12 0x000055555571ec9b in rb_f_kill (argc=2, argv=0x7ffff7ecb048) at signal.c:480
#13 0x0000555555789c44 in vm_call_cfunc_with_frame (ec=0x555555add590, reg_cfp=0x7ffff7fcafa0, calling=<optimi
zed out>, cd=<optimized out>) at vm_insnhelper.c:2601
#14 0x000055555578d4d5 in vm_sendish (ec=0x555555add590, reg_cfp=0x7ffff7fcafa0, cd=0x555555bfaf30, block_hand
ler=<optimized out>, method_explorer=<optimized out>) at vm_callinfo.h:337
#15 0x000055555578ef98 in vm_exec_core (ec=0x555555add590, initial=11) at insns.def:799
#16 0x000055555557a4ee4 in rb_vm_exec (ec=0x555555add590, mjit_enable_p=1) at vm.c:1953
#17 0x000055555555ff1d0 in rb_ec_exec_node (ec=ec@entry=0x555555add590, n=n@entry=0x555555ae9958) at eval.c:296
#18 0x0000555555603533 in ruby_run_node (n=0x555555ae9958) at eval.c:354
#19 0x000055555557ee0f in main (argc=<optimized out>, argv=<optimized out>) at main.c:50
(gdb)
```

It seems the generated DWARF section is broken. For instance addr2line(1) also fails to understand it.

```
% nm ./miniruby | fgrep -w rb_f_kill | LC_ALL=C addr2line -e ./miniruby
addr2line: Dwarf Error: Could not find abbrev number 64.
??:?
:?
```

When you kill LTO option the above one liner must show "signal.c:423" or something.

vo.x (Vit Ondruch) is it possible for you to ask this to linker people instead? As addr2line(1) is also affected, it is hard for me to think we are the ones who is doing something wrong.

#3 - 08/20/2020 03:53 AM - shyouhei (Shyouhei Urabe)

Meanwhile it can not be a bad idea for us to avoid SEGV even when DWARF is broken.

#4 - 08/20/2020 07:43 AM - vo.x (Vit Ondruch)

Thank you for the investigation. There is a whole thread on Fedora devel mailing list:

https://lists.fedoraproject.org/archives/list/devel@lists.fedoraproject.org/thread/OAD6BEN6UMVIXAEI5EDS5COUYFSPNLZN/

I have posted results of your investigations there and I think Jeff Law will be back soon with some advice/fix.

#5 - 08/20/2020 07:44 AM - vo.x (Vit Ondruch)

vo.x (Vit Ondruch) wrote in #note-4:

> Thank you for the investigation. There is a whole thread on Fedora devel mailing list:
>
> https://lists.fedoraproject.org/archives/list/devel@lists.fedoraproject.org/thread/OAD6BEN6UMVIXAEI5EDS5COUYFSPNLZN/
>
> I have posted results of your investigations there

This is the link:

https://lists.fedoraproject.org/archives/list/devel@lists.fedoraproject.org/message/NLQBH6N6GK53T4CC5CM2SWQEPYGS2CMR/

> and I think Jeff Law will be back soon with some advice/fix.