Fedora recently switched from DWARF4 to DWARF5 and since that time, I observe test suite errors on ppc64le:

1) Failure: TestBugReporter#test_bug_reporter_add [/builddir/build/BUILD/ruby-3.0.0/test/-ext-/bug_reporter/test_bug_reporter.rb:24]:
   pid 1691449 killed by SIGSEGV (signal 11) (core dumped)
   | -:1: [BUG] Segmentation fault at 0x000003e80019cf39
   | ruby 3.0.0p0 (2020-12-25 revision 95aff21468) [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:000480 (none) [FINISH]
   | -- Ruby level backtrace information ----------------------------------------
   | -:1:in `main'
   | -:1:in `kill'
   | -- C level backtrace information -----------------------------------------

2) Failure: TestRubyOptions#test_segv_loaded_features [/builddir/build/BUILD/ruby-3.0.0/test/ruby/test_rubyoptions.rb:748]:
   pid 1721938 killed by SIGSEGV (signal 11) (core dumped)
   | -e:1: [BUG] Segmentation fault at 0x000003e8001a4652
   | ruby 3.0.0p0 (2020-12-25 revision 95aff21468) [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:000480 (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 /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:000480 (none) [FINISH]\n   "-- Ruby level backtrace information ----------------------------------------\n   "-:1:in `main'\n   "-:1:in `kill'\n   "-- C level backtrace information -----------------------------------------\n
after 4 patterns with 119 characters.
1) Failure:
TestBugReporter#test_bug_reporter_add [/build/bui
/d/BUILD/ruby-3.0.0/test/-ext/-bug_reporter/test
bug_reporter.rb:24]);
pid 722 killed by SIGSEGV (signal 11) (core dumped)
+ e:1: [BUG] Segmentation fault at 0x0000002d2
ruby 3.0.0p0 (2020-12-25 revision 95aff21468) [armv7hl-linux]
| -- Control frame information -----------------------------------------------
| c:0003 p:---- s:0002 e:0000003 CFUNC :kill
| c:0002 p:0000 s:00000 e:0000005 EVAL -e:1 [FINISH]
| c:0001 p:0000 s:0003 E:0000c98 (none) [FINISH]
| -- Ruby level backtrace information ----------------------------------------
| -e:1:in `<main>'
| -e:1:in `kill'
| -- Machine register context ----------------------------------------------

And similar issues on armv7hl:

1) Failure:
TestBugReporter#test_bug_reporter_add [/build/bui
/d/BUILD/ruby-3.0.0/test/-ext/-bug_reporter/test
bug_reporter.rb:24]);
pid 722 killed by SIGSEGV (signal 11) (core dumped)
+ e:1: [BUG] Segmentation fault at 0x0000002d2
ruby 3.0.0p0 (2020-12-25 revision 95aff21468) [armv7hl-linux]
| -- Control frame information -----------------------------------------------
| c:0003 p:---- s:0002 e:0000003 CFUNC :kill
| c:0002 p:0000 s:00000 e:0000005 EVAL -e:1 [FINISH]
| c:0001 p:0000 s:0003 E:0000c98 (none) [FINISH]
| -- Ruby level backtrace information ----------------------------------------
| -e:1:in `<main>'
| -e:1:in `kill'
| -- Machine register context ----------------------------------------------

Finished tests in 1111.35507s, 18.8689 tests/s, 2396.5689 assertions/s.
20970 tests, 266349 assertions, 4 failures, 0 errors, 60 skips
ruby -v: ruby 3.0.0p0 (2020-12-25 revision 95aff21468) [powerpc64le-linux]
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:000c98 (none) [FINISH]\n   "-- Ruby level backtrace information ----------------------------------------\n   | -:1:in 'block in <main>'\n   | -:1:in 'kill'
   "-- Machine register context -----------------------------------------------\n   "r0: 0x00000000 "r1: 0x0000000b "r2: 0x00000001 "r3: 0x00000001 "r4: 0x0000002d\n   "r5: 0xb689a028 "r6: 0x0000000b "r7: 0x00000025 "r8: 0x00000002 "r9: 0x0000002d\n   "r1: 0x00000001 "sp: 0xb476d1c "fa: 0x00000000
   "-- C level backtrace information ------------------------------------------

2) Failure:
TestRubyOptions#test_segv_loaded_features [/builddir/build/BUILD/ruby-3.0.0/test/ruby/test_rubyoptions.rb:748]:
pid 8085 killed by SIGSEGV (signal 11) (core dumped)
   -:1: [BUG] Segmentation fault at 0x00001f95
   ruby 3.0.0p0 (2020-12-25 revision 95aff21468) [armv7hl-linux]
   "-- Control frame information -----------------------------------------------\n   | c:0003 p:---- s:0012 e:000011 CFUNC :kill\n   | c:0002 p:0016 s:0006 e:000005 BLOCK -:1 [FINISH]\n   | c:0001 p:0000 s:0003 E:000660 (none) [FINISH]\n   "-- Ruby level backtrace information ----------------------------------------\n   | -:1:in `block in <main>'\n   | -:1:in 'kill'
   "-- Machine register context -----------------------------------------------\n   "r0: 0x00000000 "r1: 0x0000000b "r2: 0x00000001 "r3: 0x00000001 "r4: 0x000000f9\n   "r5: 0xb68e3028 "r6: 0x0000000b "r7: 0x00000025 "r8: 0x00000002 "r9: 0x000001f9\n   "r1: 0x00000001 "sp: 0xb7e8c5ac4 "fa: 0x00000000
   "-- C level backtrace information ------------------------------------------

3) Failure:
TestRubyOptions#test_segv_setproctitle [/builddir/build/BUILD/ruby-3.0.0/test/ruby/test_rubyoptions.rb:762]:
pid 8092 killed by SIGSEGV (signal 11) (core dumped)
   -:1: [BUG] Segmentation fault at 0x00001f9c
   ruby 3.0.0p0 (2020-12-25 revision 95aff21468) [armv7hl-linux]
   "-- Control frame information -----------------------------------------------\n   | c:0003 p:---- s:0012 e:000011 CFUNC :kill\n   | c:0002 p:0029 s:0006 e:000005 EVAL -:1 [FINISH]\n   | c:0001 p:0000 s:0003 E:000550 (none) [FINISH]\n   "-- Ruby level backtrace information ----------------------------------------\n   | -:1:in `block in <main>'\n   | -:1:in 'kill'
   "-- Machine register context -----------------------------------------------\n   "r0: 0x00000000 "r1: 0x0000000b "r2: 0x00000001 "r3: 0x00000001 "r4: 0x000001f9\n   "r5: 0xb68e3028 "r6: 0x0000000b "r7: 0x00000025 "r8: 0x00000002 "r9: 0x000001f9\n   "r1: 0x00000001 "sp: 0xb7e8c5ac4 "fa: 0x00000000
   "-- C level backtrace information ------------------------------------------
Failed tests in 1428.305544s, 14.6810 tests/s, 1866.1553 assertions/s.
20969 tests, 2665440 assertions, 4 failures, 0 errors, 56 skips

ruby v: ruby 3.0.0p0 (2020-12-25 revision 95aff21468) [armv7hl-linux]

So is it really due to DWARF5? Why is that just on two platforms?

Associated revisions
Revision 9e5105ca - 03/15/2021 06:22 AM - mame (Yusuke Endoh)
Support GCC's DWARF 5 [Bug #17585]
Co-Authored-By: xtkoba (Tee KOBAYASHI) xtkoba+ruby@gmail.com

History
#1 - 01/26/2021 05:12 PM - vo.x (Vit Ondruch)
Just FTR, I have reported this initially against GCC:
https://bugzilla.redhat.com/show_bug.cgi?id=1920533

#2 - 01/26/2021 09:02 PM - kou (Kouhei Sutou)
- Subject changed from DWAR5 support? to DWARF5 support?

#3 - 02/03/2021 05:50 AM - xtkoba (Tee KOBAYASHI)
"A segfault in the segfault handler" occurs also on my x86_64-linux and aarch64-linux when gcc-10.2.0 -gdwarf-5 is used. A backtrace with GDB:

```
(gdb) run -e 'Process.kill :SEGV, $$'
Starting program: /var/tmp/ruby.build/ruby-3.0.0-x86_64-gcc.dwarf5/miniruby -e 'Process.kill :SEGV, $$'
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/lib64/libthread_db.so.1".
```

Program received signal SIGSEGV, Segmentation fault.
0x00007ffff7befdf7 in kill () from /lib64/libc.so.6
(gdb) cont
Continuing.
e:e:1: [BUG] Segmentation fault at 0x000003e800000a68
ruby 3.0.0p0 (2020-12-25 revision 95aff21468) [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:000ac0 (none) [FINISH]

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

-- Machine register context ---------------------------------------------------------------
RIP: 0x00007ffff7befdf7
RBP: 0x00007fffffffc6d0
RSP: 0x00007fffffffc678
RAX: 0x0000000000000000
RBX: 0x000055555599e0c0
RCX: 0x0000000000000000
RDI: 0x0000000000000000
RSI: 0x0000000000000000
R8: 0x00005555557192c4
R9: 0x000055555599e0c0
R10: 0x0000000000000000
R11: 0x0000000000000000
R12: 0x000055555557d2c0
R13: 0x0000000000000000
R14: 0x0000000000000000
R15: 0x00007ffff7788fa0
EFL: 0x0000000000000206

-- C level backtrace information ----------------------------------------------------------
Program received signal SIGSEGV, Segmentation fault.
uleb128 (p=0x555555a965f8) at addr2line.c:198
198     unsigned char b = *(*(unsigned char *)p)+; (gdb)
#0  uleb128 (p=0x555555a965f8) at addr2line.c:198
#1 0x0000000000000000 in di_read_die (reader=0x555555a965a0, die=0x555555a96390) at addr2line.c:1327
#2 0x0000000000000000 in debug_info_read (reader=0x555555a965a0, num_traces=23, traces=0x555555998900 <trace>, lines=0x5555555ad7c0, offset=0) at addr2line.c:1557
#3 0x0000000000000000 in fill_lines (num_traces=23, traces=0x555555998900 <trace>, check_debuglink=1, objp=0x555555a966a0, lines=0x555555a967c0, offset=0) at addr2line.c:1813
#4 0x0000000000000000 in rb_dump_backtrace_with_lines (num_traces=23, traces=0x555555998900 <trace>) at addr2line.c:2207
#5 0x0000000000000000 in rb_print_backtrace () at vm_dump.c:760
#6 0x0000000000000000 in rb_vm_bugreport (ctx=0x555555a97240) at vm_dump.c:998
#7 0x0000000000000000 in rb_bug_for_fatal_signal (default_sighandler=0x0, sig=11, ctx=0x555555a97240, fmt=0x5555557718b "Segmentation fault at %p") at error.c:786
#8 0x0000000000000000 in sigsegv (sig=11, info=0x555555a97370, ctx=0x555555a97240) at signal.c:960
#9 <signal handler called>
#10 0x0000000000000000 in __do_signal hooked (inpcb=0x0, action=1) from /lib64/libc.so.6
#11 0x0000000000000000 in rb_f_kill (argc=2, argv=0x7fffffff7689048) at signal.c:481
     (...snip...)
     (gdb) p *p
$1 = 0x401 <error: Cannot access memory at address 0x401>

I propose the following change as a workaround:

--- a/addr2line.c
+++ b/addr2line.c
@@ -1471,7 +1471,7 @@
    reader->cu_end = reader->p + unit_length;
    version = read_uint16(&reader->p);
-   if (version > 5) {  
+   if (version > 4) {
       return -1;
   }
   else if (version == 5) {  

#4 - 02/03/2021 06:00 AM - mame (Yusuke Endoh)
xtkoba (Tee KOBAYASHI) Beautiful!

vo.x (Vit Ondruch) Can you check the patch on your Fedora machines?

#5 - 02/05/2021 05:13 AM - xtkoba (Tee KOBAYASHI)
- File ruby-addr2line-dwarf5.patch added

#6 - 02/10/2021 04:39 PM - xtkoba (Tee KOBAYASHI)
- File ruby-dwarf5-debug_line.patch added

03/17/2021
It seems that GCC emits version 3 .debug_line sections even when -gdwarf-5 is provided, in contrast to Clang/LLVM emitting version 5, which addr2line.c simply skips currently.

Not causing crashes with DWARF-5 is one thing, and supporting DWARF-5 is another. It will require a fair amount of work to fully support version 5 .debug_line (or other) sections.

I tried making a minimal modification to addr2line.c so that it prints file:line information with Clang/LLVM's DWARF-5 sections. A patch is attached for that. It works with my x86_64 Linux + Clang 11.0.1, but probably will not with others.

#7 - 02/16/2021 12:00 PM - vo.x (Vit Ondruch)
xtkoba (Tee KOBAYASHI) wrote in #note-5:

A patch is attached to avoid segfaults with DWARF5.

I can't see any change with this patch. It fails on ppc64le and armv7hl. Will try the other one.

#8 - 02/16/2021 03:38 PM - vo.x (Vit Ondruch)

With the later patch, it crashes even on my x86_64 :(

#9 - 02/17/2021 08:32 AM - xtkoba (Tee KOBAYASHI)

Excuse me, but the patch in #note-6 is just a scratch and is not expected to work as it is.

And it is highly probable that there exist new attributes from DWARF-5 that are not covered by the patch in #note-5.

Once I thought it would be a workaround to add -gdwarf-4 to debugflags, but it will be insufficient and will still cause a crash when a shared library linked from ruby (e.g. libc.so) uses DWARF-5.

Now might be the time to consider moving from Ruby's own addr2line to Ian Lance Taylor's libbacktrace:
https://github.com/ianlancetaylor/libbacktrace

#10 - 02/17/2021 03:53 PM - xtkoba (Tee KOBAYASHI)

For the libbacktrace library, I made a feature request (Feature #17638).

What I feel strange is that the failures were reported to happen only on two platforms, contrary to my experiment in which a crash occurred with x86_64-linux + gcc-10.2.0 -gdwarf-5 -ggdb3.

It might not be improbable that in the reported situations crashes happened within backtrace(3). I know that it can cause a crash on QEMU-user-emulated ppc64le Linux. It is true that if so the version of DWARF would be irrelevant, but it might be worth investigating using GDB or so exactly where a crash happens.

#11 - 02/17/2021 06:14 PM - vo.x (Vit Ondruch)

Just FTR, there are long time outstanding platform specific issues with SIGSEV handler such as #16492 so this might be just coincidence. Not sure though

#12 - 02/18/2021 12:43 PM - xtkoba (Tee KOBAYASHI)

With the patch in #note-5 applied and compiled using gcc-10.2.0 -O0 -gdwarf-5 -ggdb3, addr2line.c appears to print incorrect information on some Linux environments.

Version:
ruby 3.1.0dev (2021-02-18 master 07ab172ebe) (patched)

Command:
./miniruby -e 'Process.kill :SEGV, $$'

The top 3 lines from C level backtrace information for each environment:

```
[aarch64-linux]
./miniruby(rb_print_backtrace+0x18) [0x55002dc6b4] vm_dump.c:822
./miniruby(rb_vm_bugreport+0x124) [0x55002d5cc6c] vm_dump.c:1074
./miniruby(rb_bug_for_fatal_signal+0xf8) [0x55000b8504] error.c:801

[armv7a-linux-eabi]
./miniruby(rb_print_backtrace+0x20) [0x40348d80] vm_dump.c:824
./miniruby(rb_vm_bugreport+0x13c) [0x4034928c] vm_dump.c:1083
./miniruby(rb_bug_for_fatal_signal+0x8b0) [0x400c02b4] error.c:848
```

03/17/2021
In the above outputs, the line numbers seem to be correct on riscv64, s390x and x86_64, but not on aarch64, armv7a or powerpc64le. In the latter three architectures, it also happens that some function names are not printed (in another case of segfault):

In summary, there are three changes:

1. correct the interpretation of DW_LNS_advance_pc statements when the minimum instruction length is not equal to 1 (which is the case for aarch64, arm and powerpc64le),
2. skip DW_FORM_implicit_const attributes properly, and
3. read and use the .debug_rnglists section if available.

When applied together with the patch for Bug #17656, this patch will make addr2line.c correctly extract function+offset and file:line information from GCC's DWARF 5 sections, except that some "inlined by" information is not shown.

Note that in the above changes, (3) is minimal and only DW_RLE_base_address and DW_RLE_offset_pair are interpreted, because other range list entries seem unnecessary for now. It is highly probable that other entries become necessary in the future.

I hope that the workaround in #note-3 would be no longer needed.

#note-5 is deleted to avoid confusion.
A minor mod to the patch in #note-13 (due to an issue similar to Bug #17645):

```c
--- a/addr2line.c
+++ b/addr2line.c
@@ -1463,7 +1463,7 @@
       }
       break;
     case DW_RLE_base_address:
-       base_address = read_dw_form_addr(reader, &p);
+       base_address = (uintptr_t)read_dw_form_addr(reader, &p);
       break;
     case DW_RLE_start_end:
       read_dw_form_addr(reader, &p);
```

xtkoba (Tee KOBAYASHI) wrote in #note-13:

A (revised) patch is attached to avoid segfaults with GCC's DWARF 5.

I have tested this patch (without any others you have referenced) and after 10 build in Fedora build system (that is 60 builds on various platforms supported by Fedora) and it seems to fix the original issue. I think I'll keep it applied in Fedora, unless you foresee some further issues.

Thx a lot for working on this!

xtkoba (Tee KOBAYASHI) wrote in #note-13:

A (revised) patch is attached to avoid segfaults with GCC's DWARF 5.

I have tested this patch (without any others you have referenced) and after 10 build in Fedora build system (that is 60 builds on various platforms supported by Fedora) and it seems to fix the original issue. I think I'll keep it applied in Fedora, unless you foresee some further issues.

Thx a lot for working on this!

xtkoba (Tee KOBAYASHI) wrote in #note-13:

A (revised) patch is attached to avoid segfaults with GCC's DWARF 5.

I have tested this patch (without any others you have referenced) and after 10 build in Fedora build system (that is 60 builds on various platforms supported by Fedora) and it seems to fix the original issue. I think I'll keep it applied in Fedora, unless you foresee some further issues.

Thx a lot for working on this!

xtkoba (Tee KOBAYASHI) wrote in #note-13:

A (revised) patch is attached to avoid segfaults with GCC's DWARF 5.

I have tested this patch (without any others you have referenced) and after 10 build in Fedora build system (that is 60 builds on various platforms supported by Fedora) and it seems to fix the original issue. I think I'll keep it applied in Fedora, unless you foresee some further issues.

Thx a lot for working on this!

xtkoba (Tee KOBAYASHI) wrote in #note-13:

A (revised) patch is attached to avoid segfaults with GCC's DWARF 5.

I have tested this patch (without any others you have referenced) and after 10 build in Fedora build system (that is 60 builds on various platforms supported by Fedora) and it seems to fix the original issue. I think I'll keep it applied in Fedora, unless you foresee some further issues.

Thx a lot for working on this!

xtkoba (Tee KOBAYASHI) wrote in #note-13:

A (revised) patch is attached to avoid segfaults with GCC's DWARF 5.

I have tested this patch (without any others you have referenced) and after 10 build in Fedora build system (that is 60 builds on various platforms supported by Fedora) and it seems to fix the original issue. I think I'll keep it applied in Fedora, unless you foresee some further issues.

Thx a lot for working on this!

xtkoba (Tee KOBAYASHI) wrote in #note-13:

A (revised) patch is attached to avoid segfaults with GCC's DWARF 5.

I have tested this patch (without any others you have referenced) and after 10 build in Fedora build system (that is 60 builds on various platforms supported by Fedora) and it seems to fix the original issue. I think I'll keep it applied in Fedora, unless you foresee some further issues.

Thx a lot for working on this!

xtkoba (Tee KOBAYASHI) wrote in #note-13:

A (revised) patch is attached to avoid segfaults with GCC's DWARF 5.

I have tested this patch (without any others you have referenced) and after 10 build in Fedora build system (that is 60 builds on various platforms supported by Fedora) and it seems to fix the original issue. I think I'll keep it applied in Fedora, unless you foresee some further issues.

Thx a lot for working on this!

xtkoba (Tee KOBAYASHI) wrote in #note-13:

A (revised) patch is attached to avoid segfaults with GCC's DWARF 5.

I have tested this patch (without any others you have referenced) and after 10 build in Fedora build system (that is 60 builds on various platforms supported by Fedora) and it seems to fix the original issue. I think I'll keep it applied in Fedora, unless you foresee some further issues.

Thx a lot for working on this!

xtkoba (Tee KOBAYASHI) wrote in #note-13:

A (revised) patch is attached to avoid segfaults with GCC's DWARF 5.

I have tested this patch (without any others you have referenced) and after 10 build in Fedora build system (that is 60 builds on various platforms supported by Fedora) and it seems to fix the original issue. I think I'll keep it applied in Fedora, unless you foresee some further issues.

Thx a lot for working on this!

xtkoba (Tee KOBAYASHI) wrote in #note-13:

A (revised) patch is attached to avoid segfaults with GCC's DWARF 5.

I have tested this patch (without any others you have referenced) and after 10 build in Fedora build system (that is 60 builds on various platforms supported by Fedora) and it seems to fix the original issue. I think I'll keep it applied in Fedora, unless you foresee some further issues.

Thx a lot for working on this!

xtkoba (Tee KOBAYASHI) wrote in #note-13:

A (revised) patch is attached to avoid segfaults with GCC's DWARF 5.

I have tested this patch (without any others you have referenced) and after 10 build in Fedora build system (that is 60 builds on various platforms supported by Fedora) and it seems to fix the original issue. I think I'll keep it applied in Fedora, unless you foresee some further issues.

Thx a lot for working on this!

xtkoba (Tee KOBAYASHI) wrote in #note-13:

A (revised) patch is attached to avoid segfaults with GCC's DWARF 5.

I have tested this patch (without any others you have referenced) and after 10 build in Fedora build system (that is 60 builds on various platforms supported by Fedora) and it seems to fix the original issue. I think I'll keep it applied in Fedora, unless you foresee some further issues.

Thx a lot for working on this!

xtkoba (Tee KOBAYASHI) wrote in #note-13:

A (revised) patch is attached to avoid segfaults with GCC's DWARF 5.

I have tested this patch (without any others you have referenced) and after 10 build in Fedora build system (that is 60 builds on various platforms supported by Fedora) and it seems to fix the original issue. I think I'll keep it applied in Fedora, unless you foresee some further issues.

Thx a lot for working on this!

xtkoba (Tee KOBAYASHI) wrote in #note-13:

A (revised) patch is attached to avoid segfaults with GCC's DWARF 5.

I have tested this patch (without any others you have referenced) and after 10 build in Fedora build system (that is 60 builds on various platforms supported by Fedora) and it seems to fix the original issue. I think I'll keep it applied in Fedora, unless you foresee some further issues.

Thx a lot for working on this!

xtkoba (Tee KOBAYASHI) wrote in #note-13:

A (revised) patch is attached to avoid segfaults with GCC's DWARF 5.

I have tested this patch (without any others you have referenced) and after 10 build in Fedora build system (that is 60 builds on various platforms supported by Fedora) and it seems to fix the original issue. I think I'll keep it applied in Fedora, unless you foresee some further issues.

Thx a lot for working on this!

xtkoba (Tee KOBAYASHI) wrote in #note-13:

A (revised) patch is attached to avoid segfaults with GCC's DWARF 5.

I have tested this patch (without any others you have referenced) and after 10 build in Fedora build system (that is 60 builds on various platforms supported by Fedora) and it seems to fix the original issue. I think I'll keep it applied in Fedora, unless you foresee some further issues.

Thx a lot for working on this!