

Corrupt debug info when using lld with gcc 9.1 #42401

Edit

New issue

✓ Closed

MoritzS opened this issue on 26 Jun 2019 · 9 comments



MoritzS commented on 26 Jun 2019

Bugzilla Link	42401
Resolution	DUPLICATE
Resolved on	Jun 28, 2019 05:21
Version	unspecified
OS	Linux
CC	@dwblaikie, @MaskRay, @pogo59, @smithp35

Extended Description

I have the following program:

foo.cpp:

Assignees

No one—assign yourself

Labels

bugzilla duplicate lld

Projects

None yet

Milestone

No milestone

Linked pull requests

Successfully merging a pull request may close this issue.

None yet

```
int foo() {  
    return 1;  
}
```

main.cpp:

```
int main() {  
    return 0;  
}
```

If I compile it with gcc 9.1 like this:

```
gcc -g -fuse-ld=lld -o main foo.cpp main.cpp
```

I get corrupted debug info. gdb says:

Reading symbols from main...



Dwarf Error: wrong version in compilation unit header (is 1024, should be 2, 3, 4 or 5) [in module <...>/main]

(No debugging symbols found in main)

This is the output of "readelf --debug-dump=info main":

Contents of the .debug_info section:

0 participants

 Pin issue 

Compilation Unit @ offset 0x0:

Length: 0x53 (32-bit)

Version: 4

Abbrev Offset: 0x0

Pointer Size: 8

<0>: Abbrev Number: 1 (DW_TAG_compile_unit)

DW_AT_producer : (indirect string, offset: 0x31): GNU C++14 9.1.0 -mtune=generic -
march=x86-64 -g -fuse-ld=lld

<10> DW_AT_language : 4 (C++)

<11> DW_AT_name : (indirect string, offset: 0x6e): foo.cpp

<15> DW_AT_comp_dir : (indirect string, offset: 0x0): <...>

<19> DW_AT_low_pc : 0x10f9

<21> DW_AT_high_pc : 0xb

<29> DW_AT_stmt_list : 0x0

<1><2d>: Abbrev Number: 2 (DW_TAG_subprogram)

<2e> DW_AT_external : 1

<2e> DW_AT_name : foo

<32> DW_AT_decl_file : 1

<33> DW_AT_decl_line : 1

<34> DW_AT_decl_column : 5

<35> DW_AT_linkage_name: (indirect string, offset: 0x24): _Z3foov

<39> DW_AT_type : <0x4f>

<3d> DW_AT_low_pc : 0x10f9

<45> DW_AT_high_pc : 0xb

<4d> DW_AT_frame_base : 1 byte block: 9c (DW_OP_call_frame_cfa)

<4f> DW_AT_GNU_all_call_sites: 1

<1><4f>: Abbrev Number: 3 (DW_TAG_base_type)

<50> DW_AT_byte_size : 4

<51> DW_AT_encoding : 5 (signed)

<52> DW_AT_name : int

<1><56>: Abbrev Number: 0

readelf: Warning: Invalid pointer size (59) in compunit header, using 4 instead

Compilation Unit @ offset 0x57:

Length: 0x4f00 (32-bit)

Version: 1024

Abbrev Offset: 0x8000000

Pointer Size: 4

readelf: Warning: Debug info is corrupted, .debug_info header at 0x57 has length 4f00

This problem does not occur when using ld or gold. Also, it doesn't happen when using gcc < 9.1 or clang.



dwblaikie commented on 26 Jun 2019

Might be helpful to attach assembly for the file(s) (if, using that assembly, this can be reproduced without gcc - eg: does assembling and linking them with clang still produce the failure?)



MaskRay commented on 28 Jun 2019

I cannot reproduce the gdb issue with core/gcc 9.1.0-2 on Arch Linux.

As David suggested, it would be good to have gcc -g -S output.

You may also use -Wl,--reproduce=/tmp/rep.tar to get a reproduce tarball with all input files.

Also, make sure you are using a sufficiently newer lld: 8.0 or trunk. 7.0 may have some bugs that have been fixed.

gcc -g -fuse-lld=lld uses the lld found in PATH. If you have a locally built lld, that command may not use it. (Tip: append '-###' to check which linker gcc uses)



MoritzS commented on 28 Jun 2019

Author

I'm also using 9.1.0-2 on Arch Linux:

```
$ gcc --version  
gcc (GCC) 9.1.0
```

```
$ clang --version  
clang version 8.0.0 (tags/RELEASE_800/final)  
Target: x86_64-pc-linux-gnu  
Thread model: posix  
InstalledDir: /usr/bin
```

```
$ ld.lld --version  
LLD 8.0.0 (compatible with GNU linkers)
```

When I assemble the files with clang but link them with gcc, the bug does not happen. I attached the assembly for both files generated by gcc and by clang.



MoritzS commented on 28 Jun 2019

Author

[foo.c assembled with clang](#)



MoritzS commented on 28 Jun 2019

Author

[foo.c assembled with gcc](#)



MoritzS commented on 28 Jun 2019

Author

[main.c assembled with clang](#)



MoritzS commented on 28 Jun 2019

Author

[main.c assembled with gcc](#)



MaskRay commented on 28 Jun 2019

```
clang -fuse-ld=lld foo-gcc.s foo-main.s -o main
gdb main
```

I reproduced the

Reading symbols from a.arch...Dwarf Error: wrong version in compilation unit header (is 1024, should be 2, 3, 4 or 5) [in module /tmp/rep/a.arch]

bug with Arch Linux extra/lld 8.0.0-1 and the prebuilt 8.0.0 Ubuntu 18.04 archive on <http://releases.llvm.org/download.html#8.0.0>

lld trunk is fine. I checked out the 8.x branch <https://github.com/llvm/llvm-project/tree/release/8.x>, it is also fine.

So this is a bug (I cannot narrow down to the revision that fixed the issue) in the 8.0.0 release that will not go into the 8.0.1 release.

Moritz, you may just wait for the 8.0.1 release :)



MoritzS commented on 28 Jun 2019

Author

Thanks for the hint! I found this was fixed in Bug 40482

*** This bug has been marked as a duplicate of bug [#40482](#) ***



This issue was **closed**.
