Line number information

Line number information maps between machine code instructions and the source level location.

Encoding

The line number information is stored in the `.debug_line` section for ELF and `__debug_line` section of the `__DWARF` segment for Mach-O object files. The line number information contains a header followed by the line program. The line program is a program for a virtual machine with instructions like set line number for the current machine code instruction and advance the current machine code instruction.

Tips

You need to set either `DW_AT_low_pc` and `DW_AT_high_pc` **or** `DW_AT_ranges` of a `DW_TAG_compilation_unit` to the range of addresses in the compilation unit. After that you need to set `DW_AT_stmt_list` to the `.debug_line` section offset of the line program. Otherwise a debugger won't find the line number information. On macOS the debuginfo relocations **must** be section relative and not symbol relative. See #303 (comment) for more information.

Function debuginfo

Tips

`DW_TAG_subprogram` requires `DW_AT_name`, `DW_AT_low_pc` and `DW_AT_high_pc` **or** `DW_AT_ranges`. Otherwise gdb will silently skip it. When `DW_AT_high_pc` is a length instead of an address, the DWARF version must be at least 4.

IRC log of #gdb on irc.freenode.org at 2020-04-23

```
(13:46:11) bjorn3: i am writing a backend for a compiler that uses DWARF for debuginfo. for
(13:47:49) bjorn3: this is the output of llvm-dwarfdump: https://gist.github.com/bjorn3/8a34
(13:47:50) osa1: luispm: why is that problem not exists in 'commands'? (the target vs. host)
(13:52:16) luispm: osa1, commands is a bit more high level. It executes isolated commands.
(13:52:36) luispm: osa1, Oh, i see your point now. Commands is only executed on the host.
(13:53:18) luispm: osa1, The commands are not tied to the execution context of the debugged
(13:55:00) luispm: bjorn3, Likely something GDB thinks is wrong. Does enabling "set debug dw
(13:56:01) bjorn3: luispm: no
(13:56:12) bjorn3: for more context: https://github.com/bjorn3/rustc_codegen_cranelift/pull/
(13:58:16) osa1 verliet de ruimte (quit: Quit: osa1).
(13:58:28) bjorn3: luispm: wait, for b m<TAB> it shows nothing, but when stepping into a new
(13:58:45) bjorn3: it still doesn't show anything for `info args` though
(13:58:50) bjorn3: No symbol table info available.
(14:00:50) luispm: bjorn3, Is that expected given the nature of the binary?
```

```
(14:01:17) bjorn3: b main<TAB> may show nothing as I only set DW_AT_linkage_name and not DW_
(14:01:24) bjorn3: info args should work though
(14:03:26) luispm: Sorry, I'm not sure what's up. There may be a genuine bug there.
(14:03:41) luispm: tromey (not currently in the channel, but maybe later today) may have mor
(14:04:08) bjorn3: okay, thanks luispm!
(14:04:27) luispm: In the worst case, reporting a bug may prompt someone to look into that a
(14:04:48) luispm: Or send an e-mail to the gdb@sourceware.org mailing list.
(14:05:11) bjorn3: I don't know if it is a bug in gdb, or just me producing (slightly) wrong
(14:39:40) irker749: gdb: tom binutils-gdb.git:master * 740480b88af / gdb/ChangeLog gdb/darw
(15:22:45) irker749: gdb: tromey binutils-gdb.git:master * ecc6c6066b5 / gdb/ChangeLog gdb/c
(15:23:13) bjorn3: tromey: ping
(15:23:29) tromey: bjorn3: hey
(15:24:16) bjorn3: I am writing a backend for a compiler which uses DWARF for debuginfo. I u
(15:25:13) bjorn3: it just says: No symbol table info available.
(15:25:21) bjorn3: any idea what it could be?
(15:25:34) bjorn3: dwarfdump output: https://gist.github.com/bjorn3/8a34e333c80f13cb048381e9
(15:26:48) bjorn3: more context: https://github.com/bjorn3/rustc_codegen_cranelift/pull/978
(15:28:05) tromey: offhand I don't know, but if you can send me an executable I can look
(15:28:17) bjorn3: how should I send it?
(15:29:26) tromey: good question
(15:29:41) tromey: you could try emailing it to tromey at adacore.com
(15:29:47) tromey: dunno if that will work or not
(15:30:26) bjorn3: i will try
(15:37:27) bjorn3: tromey: i sent an email with the subject "gdb args not showing"
(15:38:29) tromey: will check now
(15:38:40) bjorn3: thanks!
(15:42:51) irker749: gdb: tdevries binutils-gdb.git:master * de82891ce5b / gdb/ChangeLog gdb
(15:42:52) irker749: gdb: tdevries binutils-gdb.git:master * 70bc38f5138 / gdb/ChangeLog gdb
(15:43:36) tromey: bjorn3: sorry, got distracted.  I have the file now
(15:45:35) tromey: my first thing when investigating was to enable complaints
(15:45:37) tromey: so I did
(15:45:40) tromey: set complaints 1000
(15:45:42) tromey: then
(15:45:51) tromey: file -readnow mini_core_hello_world
(15:46:00) tromey: gdb printed just one style of complaint
(15:46:07) tromey: During symbol reading: missing name for subprogram DIE at 0x3f7
(15:46:18) tromey: (which is really pretty good, most compilers manage to generate a bunch)
(15:46:29) tromey: and then the gdb DWARF reader says
(15:46:34) tromey:   /* Ignore functions with missing or empty names.  These are actually
(15:46:34) tromey:      illegal according to the DWARF standard.  */
(15:46:34) tromey:   if (name == NULL)
(15:46:34) tromey:     {
(15:46:37) tromey:       complaint (_("missing name for subprogram DIE at %s"),
(15:46:40) tromey:       sect_offset_str (die->sect_off));
(15:46:47) tromey: I wonder if that comment is correct though
(15:47:34) tromey: I guess pedantically maybe it is, DWARF 5 3.3.1 says
```

```
(15:47:43) tromey: The subroutine or entry point entry has a DW_AT_name attribute whose valu
(15:47:43) tromey: a null-terminated string containing the subroutine or entry point name.
(15:48:14) bjorn3: i tried set complaints, but it returned complaints for system files. i di
(15:48:21) tromey: cool
(15:48:26) bjorn3: i will try adding DW_AT_name
(15:48:45) tromey: without readnow unfortunately you get less stuff, because for whatever re
(15:49:02) tromey: sort of anyway
(15:49:43) tromey: this seems kind of pedantic of gdb, like if there's a linkage name but no
(15:50:01) tromey: also what about anonymous functions
(15:50:17) tromey: but anyway this explains the current situation and if you don't mind addi
(15:51:47) bjorn3: i added DW_AT_name.
(15:51:54) bjorn3: now it says cannot get low and high bounds for subprogram DIE at ...
(15:52:01) tromey: ugh
(15:52:10) bjorn3: i will add DW_AT_low_pc and DW_AT_high_pc
(15:52:15) tromey:   /* Ignore functions with missing or invalid low and high pc attributes.
(15:52:37) tromey: you can also use DW_AT_ranges
(15:52:55) tromey: if you'd prefer
(15:53:08) bjorn3: already using DW_AT_ranges for DW_TAG_compilation_unit
(15:53:19) bjorn3: for individual functions, there are no gaps
(15:57:07) bjorn3: still the same error with DW_AT_low_pc and DW_AT_high_pc
(15:57:24) bjorn3: tromey: ^
(15:58:08) tromey: hmmm
(15:58:30) bjorn3: should i send the new executable?
(15:58:31) tromey: send me another executable & I will debug
(15:58:33) tromey: yep
(15:59:23) bjorn3: sent as repy of the previous mail
(16:03:23) tromey: the low PC has DW_FORM_addr, but the high PC has DW_FORM_udata, which see
(16:03:50) mjw: no
(16:03:54) tromey: no?
(16:04:00) mjw: I suggested that for the DWARF standard...
(16:04:05) mjw: sorry
(16:04:58) mjw: The idea was that instead of two relocations and two address wide fields, yo
(16:05:05) tromey: ahh, I see the code now
(16:05:07) tromey: I forgot about this
(16:05:18) tromey:    if (cu->header.version >= 4 && attr_high->form_is_constant ())
(16:05:18) tromey:      high += low;
(16:05:36) mjw: that second offset doesn't need a relocation and can often be packed in some
(16:05:51) mjw: using udata might not be ideal though, but is allowed
(16:05:51) tromey: bjorn3: the problem is that this CU claims to be DWARF 3 but is using a D
(16:05:58) mjw: aha
(16:05:59) bjorn3: which one?
(16:06:03) ryoshu: hi
(16:06:08) tromey:             high_pc             (udata) 107 (+0x00000000000011b0 <_ZN21
(16:06:08) tromey:
(16:06:12) ryoshu: just soft ping, I have a queue of patches :)
(16:06:22) tromey: using this as a length requires DWARF 4
```

```
(16:06:36) tromey: for gdb at least it's fine to always emit DWARF 4
(16:06:44) bjorn3: trying dwarf 4 now
(16:06:48) tromey: I think there are some DWARF 5 features still in the works but DWARF 4 sh
(16:07:03) tromey: fini
(16:07:08) tromey: lol wrong window
(16:07:56) mjw: Maybe you can accept it for DWARF < 4. But if I remember correctly it might
(16:08:13) tromey: yeah, I vaguely recall this as well, though I'd expect there to be a comm
(16:08:21) mjw: Cannot really remember why it needed version >= 4. Maybe there was no good r
(16:08:32) bjorn3: tromey: it works!!!! thanks for all the help!
(16:08:41) tromey: my pleasure bjorn3
```