



# **BoF: DWARF5/DWARF64**



# Discussion topics

- Can we default to DWARF5 for GCC11?
  - I actually tried and have some patches
- Beyond DWARF5 (don't have patches yet)
  - DWARF64
  - (Single File) Split DWARF
  - debug-types

# DWARF5 Why and What

- Why
  - Smaller, less relocations, mainly because of more efficient `.debug_rnglists` and `.debug_loclists`
  - Easier to reason about than the various (now standardized) GNU extensions.
- What
  - Make it the default, look what breaks
  - GCC, elfutils, binutils and GDB test suites

# DWARF5 results

- elfutils/binutils don't like location-views
- GCC needs some test tweaks for new FORMs
- GDB has issues with static data members as DW\_TAG\_variable, but looks good otherwise
- binutils, various small fixes needed (see also .debug\_line discussion)
- Valgrind, RPM/debugedit, DWZ need patches

# binutils/gas .debug\_line v5

- -gdwarf-5 supported by binutils
  - But buggy .debug\_line (have patches)
    - Can they be backported?
  - When fixed want to add -gdwarf-<version> to ASM\_SPEC



# BONUS Discussions

- I had wanted to have patches for these topics, but don't yet... Lets see if people think there should be patches



# DWARF64

- (cross) section references > 32bit
  - Supported by almost any consumer already
    - If not, easy to add
- -fdwarf64/-fdwarf32
- But... needs all CUs to agree  
(or some CUs cannot references some parts)

# Single File Split DWARF

- Put `.debug_*.dwo` sections in same object file
  - No separate `.dwo` file (allowed by standard)
  - `-gsplit-dwarf=single`
  - `SHF_EXCLUDE` (ignored by link editor)
    - sadly not standardized, but looks to also be used by `.gnu_lto*` sections



# -fdebug-types

- Negatives...
  - Somewhat large overhead
  - Can only reference full type directly
  - Not universally supported (yet)
- Positives...
  - Makes deduplication easier, can be done earlier (dwz is better, but runs after link phase)
  - Should naturally fit in early debug scheme so would be somewhat natural for LTO