(clang-)Tidying up includes in systemd

Why do (unused) includes matter?

- Before:
  - Compilation (1572 times):
    - Parsing (frontend): 322.4 s
    - Codegen & opts (backend): 51.1 s
- After:
  - Compilation (1722 times):
    - Parsing (frontend): 240.1 s
    - Codegen & opts (backend): 49.6 s
- Expensive headers:
  - **27991 ms**: ../src/basic/alloc-util.h (included 1424 times, avg 19 ms)
- Various CI jobs => 2x speedup
- > 3000 includes removed

How to reduce the number of includes?

1. Remove unused includes
2. Minimize transient includes (x.c => y.h => z.h)
    - Forward declarations
    - Move macro implementation details to source file
    - Minimize inline functions

Preparing by removing all circular includes

- Various tools I tried seemed to really dislike circular includes
- Not easily automated, best done by hand
- **misc-header-include-cycle** clang-tidy check is amazing!
- clangd also seemed to stop crashing after I removed all circular includes

Integrating clang-tidy in the build system

- Attempt #1: meson's builtin clang-tidy target
  - Can't add dependencies on generated sources
  - Runs on all source files regardless of whether they're built or not
  - Doesn't allow passing extra arguments
- Attempt #2: LLVM's run-clang-tidy script
  - Does not run on header files
  - Independent scheduler (meson already has a working scheduler)
- Attempt #3: meson tests
  - Each clang-tidy invocation is a separate meson test
  - All clang-tidy tests are grouped together in the "clang-tidy" test suite

Attempt #1: IWYU (include-what-you-use)

- Independent tool using the clang C++ API
- clang C++ API does not offer any backwards compat
  - => Tied to a single major LLVM version :/
- Various bugs and missing features, fixed a few:
  - Add support for the cleanup attribute
  - Make sure C arrays result in includes and not a forward declaration
  - …

Attempt #2: clang-include-cleaner

- Less opinionated than IWYU
  - Can only report unused includes, without complaining about missing includes
- Ships with LLVM
- Also various bugs and missing features:
  - include-cleaner: Report function decls from __cleanup__ as used
  - [clang-include-cleaner] Make cleanup attr report expr location #140233
  - [clang-tidy] Add UnusedIncludes/MissingIncludes options to misc-include-cleaner
- **Integrated into clang-tidy! (misc-include-cleaner)**

Managing forward declarations

- Repeating forward declarations in every header file doesn’t scale
- Solution? **Forward declaration headers!**
  - basic-forward.h
  - sd-forward.h
  - shared-forward.h
  - core-forward.h
  - …
- basic-forward.h also transitively includes most commonly used headers
  - stdint.h
  - errno.h
  - …

Dealing with conditional compilation

- Almost all library dependencies in systemd are optional
- Lots of conditional compilation based on which libraries are available
  - => Unused includes reported when libraries are disabled!
- Solution: Only guarantee clean clang-tidy runs in a restricted environment where all dependencies are enabled

Problems with system headers

- Can’t add IWYU pragmas to system headers
  - <getopt.h> is always marked unused in favor of internal implementation headers included by <getopt.h>
- clang-include-cleaner doesn’t allow specifying pragmas in a configuration file

Other useful clang-tidy checks

- bugprone-argument-comment
- readability-inconsistent-declaration-parameter-name
- bugprone-sizeof-expression

The final boss: clang-format

- systemd has its own formatting style
  - Distinct from all the others shipped in clang-format
- systemd maintainers refuse to change style to accommodate tooling
- Slowly trying to add missing features to clang-format
  - [clang-format] Add IgnoreExtension to SortIncludes
  - [clang-format] Allow custom pointer/ref alignment in return types
- Also quite a few bugs in clang-format that still need to be fixed:
  - Designated initializer formatting
  - Table formatting

Questions?