redhat.

Janitor of Sanity

Stephan Bergmann

LibreOffice Conference, September 2019

The wonderful and frightening world of building with -fsanitize=...

redhat.

-fsanitize=undefined

- Adds lots of checks to the code being compiled
- At runtime warns about various cases of C/C++ undefined behavior:
  - From benign invalid-null-argument

    memcpy(dest, src, size);

    → if (size != 0) memcpy(dest, src, size);
    - (And remember that compilers happily exploit *all* kinds of UB.)
  - To dramatic (signed) integer overflow
    - e.g., involving Writer object positioning near

      #define FAR_AWAY LONG_MAX – 20000

      → #define FAR_AWAY SAL_MAX_INT32 – 20000

redhat.

-fsanitize=address

- Tracks memory areas:
  - Out-of-bounds array access
  - Heap use-after-free
  - Stack use-after-return
  - …
- Similar to Valgrind
  - Less runtime overhead
  - No detection of uninitialized memory

redhat.

-fsanitize=... more

- -fsanitize=memory
  - Detects use of uninitialized variables
  - But would need all of the software stack (incl. libc) be recompiled
  - Sometimes -fsanitize=undefined can step in (“load of value 160, which is not a valid value for type ‘bool’”)
- -fsanitize=thread
  - Detects data races
  - Detects lots of data races...
- -fsanitize=address with ASAN_OPTIONS=detect_leaks=1
  - Detects lots of leaks...

http://clang.llvm.org/docs/UsersManual.html#controlling-code-generation

redhat.

Performance

- UBSan just generates more code
- ASan needs shadow memory at runtime
- Runtime slowdown (but much less than Valgrind)
- `make -j8 check` works well with 16GB RAM
- Core files are disabled (but gdb works fine)
- Runtime speed feels OK for light use
- But generating slides sucks
  - Slow to edit text
  - Crashes

redhat.

Application

- Tinderbox running ASan+UBSan `make check screenshot`
- OSS-Fuzz (sanitizers are a good oracle to decide whether a given input causes bad behavior)
- Passing our bug document corpus through `soffice --convert-to ...`
- Developer dog food

redhat.

redhat.

Stumbling Blocks

- Available for Clang and GCC
  - But I only use it with Clang
- Requires Clang 9
  - UBSan needs RTTI symbols for many types
  - Itanium ABI and Clang compare RTTI pointers; GCC compares strings
  - Hack to use -fvisibility=ms-compat; no longer needed with Clang 9
- Poor documentation on our end

redhat.

Recipes

- autogen.input:

  CC=clang -fsanitize=address,undefined
  CXX=clang++ -fsanitize=address,undefined

  - Causes --disable-runtime-optimizations
  - Causes -Wl,-z,undefs
    - Because libraries expect __asn/__ubsan symbols in the executable
    - Which thus always needs to be built with -fsanitize, too
    - Leading to some LIBO_TUNNEL_LIBRARY_PATH hacks
- --enable-optimized works now

redhat.

Recipes

- Environment variables:

  ASAN_OPTIONS=external_symbolizer_path=/... /llvm-symbolizer
  UBSAN_OPTIONS=suppressions=/.../solenv/sanitizers/ubsan-suppressions

  (float-divide-by-zero to NaN is wanted in Calc)

redhat.

Lunacy's Back

–T. Rex