Where art thou, LDC?

Kai Nacke
DConf 2017
Berlin, May 4
The LDC team

David (klickverbot)
Johan (JohanEngelen)
Nicholas (thewilsonator)

David (klickverbot)
Johan (JohanEngelen)
Nicholas (thewilsonator)

Rainer (rainers)
Joakim (joakim-noah)
Joseph (WebDrake)
Dan (smolt)

... and many more contributors!
What happened since last DConf

• ‘Usual’ updates
  – New D frontend, new LLVM version, bug fixes

• Lot of performance improvements
  – Profile guided optimization (PGO)
  – Link Time optimization (LTO)

• Support for non-Intel platforms matures
  – ARM and PPC improvements
  – Cross-compiling improvements
What happened since last DConf

• New features
  – Traits and attributes for optimization control
  – Support for aggressive math optimization
  – Experimental cross-module inlining

• More distribution packages

• Removed dependency on libconfig
The state of affairs with LDC

• Frontend up to date
  – 2.073.2 in beta stage (LDC 1.3)
  – 2.074.0 already integrated (LDC 1.4)

• Support for a wide range of platforms
  – Different CPU and OS
  – Includes runtime support

• Latest LLVM support
Ongoing development

• DCompute integration

• Cross-linking support with LLVM tools
  – LLD integration

• More platform support
  – AArch64, ARMv5, RISC-V, ...

• JIT-compiled functions
External projects

• People use/integrate LDC into other projects

• Tell us about your project!

• Can we distribute it with LDC?
Projects based on/using LDC

• Calypso: direct interfacing with C++
  – Enhanced LDC compiler

• Emscripten: An LLVM-to-JavaScript Compiler
  – Experimental ports to use LDC output

• Distribution packages
  – Snap, FreeBSD ports, Guix, ...
My idea #1: More D

• Replace C++ glue code with D
• Requires D bindings for LLVM C++ API

• Challenges
  – Different semantics between C++ and D
  – Name mangling issues

• Vision: Distribute D wrapper with LLVM
My idea #2: cent/ucent

• 128bit integer datatypes
• Specified but not implemented in DMD

• Useful for
  – crypto algorithms
  – large floating point support

• Partial hardware support (PowerPC)
My idea #2: cent/ucent

• Implementation based in LDC 1.0 available
  – Linux only
  – Concerns about memory footprint
  – Druntime/Phobos support already upstreamed

• Bootstrap requires cent/ucent support
  – How to do it with DMD?

• Requires changes of test cases
  – Not yet done
Questions?