Quick start: LLVM compiler framework

Stefano Cherubin

Politecnico di Milano

13-04-2016
Contents

1 Introduction

2 LLVM framework quick start
LLVM is not a compiler.
LLVM is not a compiler.

LLVM is a collection of components which is useful to build a compiler.
What LLVM is made of

- C++ libraries
  - src/include/llvm/...
  - src/lib/...

- small application (tools)
  - src/tools/...
  - src/utils/...

You can find binaries of them in the installation directory under root/bin/...
**clang**

- **clang** is a compiler based on LLVM.

- It compiles all major C-like languages.

- It can be added as a tool in the LLVM framework but must be manually cloned in the tool directory:
  1. `cd src/tools`
  2. `git clone http://llvm.org/git/clang`

- You can easily see on a production quality compiler the impact of changes you made on your local copy of LLVM.
Contents

1 Introduction

2 LLVM framework quick start
Quick start: LLVM compiler framework

Commands

**llvm-as**  LLVM assembler
**llvm-dis**  LLVM disassembler
**opt**  LLVM optimizer
**llc**  LLVM static compiler
**lli**  directly execute programs from LLVM bitcode
**llvm-link**  LLVM bitcode linker
**llvm-lib**  LLVM lib.exe compatible library tool
**llvm-nm**  list LLVM bitcode and object file’s symbol table
**llvm-config**  Print LLVM compilation options
**llvm-stress**  generate random .ll files
**llvm-dwarfdump**  print contents of DWARF sections

For a complete reference, see LLVM command guide

Quick start: LLVM compiler framework

- c source
  - gcc / clang
  - .bc / .ll
    - llvm-link
      - libWhatever.a
    - .bc / .ll
      - opt
        - .bc / .ll
          - llc
            - .s
              - llvm-mc / as
                - .o
                  - dynLibWhatever.o
                    - ld
                      - executable

Stefano Cherubin <stefano.cherubin@polimi.it>
Writing a LLVM pass

There are a lot of tutorials available:

- Official developer guide
  llvm.org/docs/WritingAnLLVMPass

- Out-of-source pass
  github.com/quarkslab/llvm-dev-meeting-tutorial-2015

We will follow the first one, with a few adjustments.
LLVM has an internal testing infrastructure. ² Please use it.

**llvm-lit** LLVM Integrated Tester

1. Forge a proper LLVM-IR input file (.ll) for your test case
2. Instrument it with **lit** script comments
3. Run **lit** on your test
   
   - `llvm-lit /llvm/test/myTests/singleTest.ll`
     run a single test
   
   - `llvm-lit /llvm/test/myTests`
     run the test suite (folder)
4. Run **lit** on the LLVM test suite (regression testing)

To submit a bug report to LLVM developers you will be asked to write a **lit** test case that highlights the bug.

---