CS6383: Mini-Assignment #1

Due Friday January 13th, 2023 at 11:59 PM

Introduction: This assignment is aimed at setting up and familiarizing you with the LLVM compiler infrastructure (*http://llvm.org*), which has lot of very interesting research modules with active development from many compiler communities across the world.

It aims to give you some familiarity with LLVM's C-family frontend Clang (*http://clang.llvm.org/*). More specifically, this assignment asks you to do a study of the IR and options of the LLVM compiler.

Download LLVM: Obtain source code of LLVM and Clang. Please clone the official GitHub repository of LLVM, corresponding to *release/14.x* branch.

Refer to *http://llvm.org/docs/GettingStarted.html#git-mirro*r for instructions (make sure to use appropriate flags to checkout the *release/14.x* branch). Build LLVM and Clang from source. (Refer: *http://llvm.org/docs/GettingStarted.html#compiling-the-llvm-suite-source-code*.)

1. **LLVM:** Understand the directory hierarchy of LLVM. Read the documentation "Getting Started with LLVM" at *http://llvm.org/docs/GettingStarted.html#directory-layout* for LLVM.

2. **LLVM-IR**: Do a study of the LLVM-IR. Print the IR for five non-trivial programs and study them. Submit a report on your study along with the generated .ll files.

3. **Assembly language**: Generate the assembly language codes of some simple C/C++ programs. Understand how the C/C++ compilers mangle the names of various entities. Report your findings.

4. **Compiler toolchain and options:**

   (a) For a set of programs, go all the way from the preprocessing stage to binary; print the LLVM-IR, print the assembly code. Play with the various compiler frontend/optimizer options. (Refer: *https://llvm.org/docs/CommandGuide/opt.html*)

   (b) Apply trivial optimizations like **constant propagation** and **dead code elimination**. Print their CFG and Dom Tree and observe how the IR gets transformed. (Refer: *https://llvm.org/docs/Passes.html*)

   (c) Explore the relevant tools in LLVM - llvm-as, llvm-dis, llc, lli, etc. with different examples. (Refer *https://llvm.org/docs/GettingStarted.html#llvm-tools*)

Report your findings. Support your observations with the inputs used and the obtained output.

Submission: You will be evaluated on a technical report of your study. Also submit code samples for both C/C++ and .ll files. The report has to focus on the Frontend, LLVM-IR, and the LLVM compiler options.

Your submission should be a gzip archive with name **Mini-Asgn-1_ROLLNO.tar.gz** containing the following files.

- The report in PDF format, preferably generated from a LyX/LaTeX
- The test cases and the outputs obtained (dot files, etc.)
- Single bash script file containing the necessary commands.

Note: It is expected that on running the script from its directory, we should be able to obtain all the outputs/observations that you have claimed in the report.

Evaluation: Your report should show a good understanding on each of the points asked. The rules of plagiarism would be strictly enforced - *https://cse.iith.ac.in/academics/plagiarism-policy.html*. In particular **do not** directly copy any material from manuals/websites. Your report should professionally refer the website(s) from where you downloaded the code and the manual(s) you referred to.