CS6383: Assignment #0
An Introduction to the LLVM Infrastructure, IR and Compiler Options
Due Wednesday August 28th, 2019 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. You will also use it for the next couple of assignments.

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

- **Download LLVM:** Obtain source code of LLVM and Clang. Official releases are available at `http://llvm.org/releases/download.html`, however please download the latest version(trunk) using git/svn `http://llvm.org/docs/GettingStarted.html#checkout-llvm-from-git`. Build LLVM and Clang from source. You can refer `http://llvm.org/docs/GettingStarted.html#compiling-the-llvm-suite-source-code`.
- **LLVM and Clang:** Understand the directory hierarchy of LLVM and Clang. Read the documentation *"Getting Started with LLVM"* at `http://llvm.org/docs/GettingStarted.html#directory-layout`. Use `cscope` or `tags` facility of linux and vim to navigate across source files. Alternatively, you can use online repo `http://llvm.org/doxygen/index.html` to *browse* the source as well as see class hierarchy.
- **Error messages:** Do a study of how the error messages are handled in LLVM source code, primarily the assert mechanism of LLVM. Report on the handling of some specific error messages. You need to compile LLVM with flag -DLLVM_ENABLE_ASSERTIONS=On to enable assertions.
- **LLVM-IR:** Do a study of the LLVM-IR. Print the IR for five non-trivial programs and study them (ex. factorial, loop constructs). Submit a report on your study along with the generated *.ll* files.
- **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.
- **Compiler toolchain and options:** For a set of programs, go all the way; print the LLVM-IR, print the assembly code. Play with the various compiler frontend/middle-end/optimizer options. Report your findings.

Submission You will be evaluated on a *technical report* on your study. Also submit code samples for both C/C++ and *.ll* files. The report has to focus on the Frontend, LLVM-IR, the LLVM compiler options, and finally the ASM code generated. Additional material on LLVM projects and tools (ex. openmp, polly) will be considered for bonus points. The report should be a PDF, preferably generated from a lyx/latex file. Your submission should be a bzip2 archive with name Asgn0_ROLLNO.bzip2 containing all the files.

Evaluation Your report should show a *good* understanding on each of the points asked. The usual rules of plagiarism apply to your report. In particular **do not** directly copy the 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.