

The goal of this assignment is to familiarize you with the LLVM compiler by implementing a simple LLVM pass that outputs a bitcode file with C-level source line numbers added as comments.

1 Your Assignment

1. Read the documentation about “Writing an LLVM Pass,” which can be found here:

<http://llvm.org/releases/3.5.1/docs/WritingAnLLVMPass.html>

In particular, read the sections “Introduction - What is a Pass?” and “Quick Start - Writing hello world,” which provides an example of how you can write a pass that prints all the function names of a C file.

However, do *not* build your pass in the LLVM source tree. Instead use the skeleton project in the virtual machine (and separately on the website). It uses a simple `Makefile` to build a pass without having to integrate with the LLVM build system.

2. Implement a new pass that writes the source line number that corresponds to each LLVM instruction. More specifically, given a C source file `hello.c`, use the following command to compile it down to LLVM bitcode file `hello.bc` with debugging information included:

```
clang hello.c -c -emit-llvm -O0 -g
```

Now, given the bitcode file `hello.bc`, your pass should write to stdout the bitcode in textual human-readable format, with the source line number that corresponds to each LLVM instruction appended at the end of the line as comments.

The skeleton project has code to set things up. After you complete your pass and run `make`, it will produce a dynamic library named `annotate.so`. You can load it using `opt`, and you can test your pass as follows:

```
opt -load ./annotate.so -disable-output -annotate hello.bc
```

To understand how to get the source line number of an LLVM instruction, consult the document “Source Level Debugging with LLVM”:

<http://llvm.org/docs/SourceLevelDebugging.html>

In particular, you will want to read Subsection “C/C++ source file information” of the Section called “C/C++ front-end specific debug information.”

To add comments to an LLVM instruction, you will want to override the `AssemblyAnnotationWriter` class:

http://llvm.org/doxygen/classllvm_1_1AssemblyAnnotationWriter.html

3. Use Canvas to submit a single `tar.gz` or `tar.bz2` file that contains your source code in a directory with a `Makefile` that will build it. For this project please name the directory `assignment1`. Make sure that your code builds correctly on the provided virtual machine and does not depend on any files outside the code you submit.

The build system is not the subject of this class so feel free to help each other with it or post useful variants of the `Makefile`.

2 Due Date

This assignment is due on Monday February 2nd before class.