LLVM Project - InstCounter

Feb 7, 2018

1 Description

In this assignment, you will implement and test a compiler pass that **instruments** a program to report the number of intermediate-level executed instructions. The idea is to insert appropriate calls in the program(instrumentation). For your convenience, we provide a template compiler pass code that can report the number of static instructions via instruments during compile time.

2 Files

In the Github repository, there are 3 files you would use for this assignment(In cs255-llvm-pass directory).

lib/InstCounter.cpp

This file is the compiler pass template for this assignment. You **must** implement your compiler pass here. This file already provides the skeleton and a related example.

runtime/InstCounting.c

This file implements the runtime functions that you need for the instrumentation. *init()* is to initialize the counter to ZERO before counting, *increase()* is to increase the counter by 1 and *print_stmt_cnt()* is to print the value of the counter.

test/test.c

This is a simple program(reverse the input string) to test your pass. Add extra test is recommended.

Here is an sample output for string *csc255* and your program should output the same result:

```
Enter the string:cs255
Reverse string is:552sc
Total instructions executed: 77
```

3 Report

You are also required to write a report for this assignment. This report should include the title, your name, a brief description of your compiler pass implementation and your test result. The report should be in PDF format and **NO MORE THAN** 3 pages. (A good report will be considered for **EXTRA BONUS**)

4 Submission Guideline

This assignment is **DUE ON midnight of Feb.13(TUE), 2018**. For submission, you have to prepare 2 files: InstCounter.cpp and YOURNetID_report.pdf. Put these two files in a directory name YOURNetID_cs255-llvm and then archive it using the following command line, and submit on Blackboard.

tar -czvf YOURNetID-cs255-llvm.tar.gz YOURNetID-cs255-llvm

5 Late Submission Policy

Remember that you have a **OVERALL 2 DAY EXTENSIONS**. And after you run out of that, you will get **10%** off per day for late submission penalty.

6 Useful Documentations

1. **LLVM Programer's Guide:** instruction on how to write a hello-world LLVM pass step by step. http://llvm.org/docs/WritingAnLLVMPass.html

2. **LLVM Programers Manual:** highlights of some important classes and interfaces available in the LLVM source-base (For example, how to iterate over basic blocks, how to iterate over instructions inside a basic block or function). http://llvm.org/docs/ProgrammersManual.html

3. **LLVM Language Reference Manual:** reference manual for the LLVM assembly language. http://llvm.org/docs/LangRef.html

4. **LLVM class reference:** reference for the interfaces of the classes needed. (For example instructions, basic block, functions) http://llvm.org/doxygen/namespacellvm.html