

15-411 Compiler Design, Fall 2021

Lab 5++: Extra Optimization

Seth and co.

Sample C0 Programs Due: Tuesday, November 23, 2021 at 11:59 PM

Compiler Due: Tuesday, December 7, 2021 at 11:59 PM

Term Paper Due: Tuesday, December 14, 2021 at 11:59 PM

1 Introduction

The main goal of this project is to explore an advanced optimization technique. Your team will have the opportunity to implement a novel loop optimization, global optimization, or some other optimization not covered in class.

2 Requirements

You are required to hand in three separate items:

- The working compiler that implement your proposed optimization on top of Lab 5--.
- C0 programs that test your optimization and demonstrate its usefulness.
- A term paper that extends the Lab 5-- report requirements by describing and critically evaluating your project.

2.1 Possibilities

From writing the project proposals you submitted a week ago, you should have a good idea of the optimization you want to implement. Be sure to read and address feedback from your proposal before starting, including meeting with the course staff if necessary. Hopefully, you have had time to think through the implementation and consider the feasibility of your ideas in the last few days! If (after thorough consideration) you decide you'd like to significantly modify your proposed extra optimization, please alert the course staff as soon as possible.

2.2 Tests and Measurement Tools

You need to demonstrate that your compiler performs your extra optimization as intended. How you do this will obviously heavily depend on the optimization you choose, so you will need to submit sample C0 program(s) ahead of time that show how your selected optimizations significantly improves the performance of the executable your compiler produces. Unlike previous labs, you don't have to provide a suite of 10-20 test cases – instead, write one (or a few) real-world, interesting

programs that your optimization affects. Before proposing an optimization, you should give some thought to your testing approach. If there will be no way to check whether your compiler meets the goals of your project, we will probably not be satisfied with your project!

2.3 Term Paper

Your paper should include an analysis of all the optimizations you wrote for Lab 5--, including tables and charts where useful (see the Lab 5-- handout for these requirements). In addition, your report should include the following sections detailing your extra optimizations:

1. **Project and Implementation.** Give a specification for your project, including reference materials and an implementation sketch with relevant data structures and algorithms.
2. **Testing Methodology.** Describe the characteristics of your sample C0 program(s) that expose the usefulness of your extra optimization. Include information such as the criteria you used as you selected or designed your tests and how the optimized output improves runtime performance.
3. **Analysis.** Critically evaluate your compiler, focusing on the work you put into your extra optimization, and outline future improvements one might make to your current implementation as well as related optimizations to further improve performance.

Since this report will encompass all of your work over the final five weeks of the semester, we expect at least 5–6 pages of thorough, thoughtful discussion and a much greater level of detail than teams who only complete Lab 5--. It will be graded at the end of the semester.

3 Deliverables and Deadlines

All your code should be placed in subdirectories of the `lab5` directory as before. Be sure to create a branch or tag named `lab5extra` for the course staff to read when you're done. We will grade you based on the code and `README` file(s) you have checked in at the deadline.

3.1 Compiler Files

As for all labs, the files comprising the compiler itself should be collected in the `lab5/` directory which should contain a `Makefile`. **Important:** You should also update the `README` file and insert a roadmap to your code. This will be a helpful guide for the grader. In particular, since there are likely to be many different projects undertaken, do introduce your extra optimization at the very top of the `README`.

Issuing the shell command

```
% make
```

should generate the appropriate files so that

```
% bin/c0c --exe <args>
```

will run your compiler and produce an executable.

After running `make`, issuing the shell command

```
% make test
```

should run your own tests and print out informative output. The command

```
% make clean
```

should remove all binaries, heaps, and other generated files.

3.2 Tests and Measurement Tools

In a directory called `lab5/bench/`, include the sample C0 program(s) that you wrote for the purpose of testing your extra optimization.

3.3 Term Paper

Submit your term paper in PDF form via Gradescope before the stated deadline. Early submissions are much appreciated since it lessens the grading load of the course staff near the end of the semester.

You may not use any late days on the term paper!

4 Notes and Hints

- Discuss your ideas with the course staff to get feedback on feasibility and scope of the project before embarking on it.
- Try to identify some intermediate goals in case your overall project turns out to be too ambitious. You should have some intermediate goals to draw on from your proposal.
- Apply regression testing. It is very easy to get caught up in new features. Please make sure that the L4 portion of your compiler continues to work correctly!