

CMPSC 401
Compiler Development
Spring 2019
Janyl Jumadinova

Course Project
Given: March 26, 2019

This course has a project component. For your project, you can work individually or in groups of two (preferably). If you decide to work in a group, each member of the group will be evaluated separately based on his or her contributions to the project. This evaluation will be determined largely from the feedback of the group members.

Your project has to be related to compilers in some way (duh!), although I do not expect you to build another complete compiler for the course project. However, your project can implement an algorithm/technique used in compilers for another application or concentrate on other areas related to the compiler (better optimization, better type system, etc.). I am giving you a lot of flexibility with what you can do with your project, use it wisely! Of course, you can not claim anything you have previously done as a contribution for this project, but you can certainly use your previous work, knowledge and experience as a backbone for this project.

Hard Requirements

1. Your project must be approved by me before you start working on it.
2. Your project must solve a problem and/or use or develop a software tool related to compilers. You need to research the problem you select to get an idea of what has been already done. You must include references to existing work in your final report.
3. Your project must have a significant implementation part. You may write your code from scratch or reuse and extend some existing code. Obviously, anything you use that is not yours must be documented. You may program in any programming language that you like.
4. Your project must be extensive enough to qualify as a project (think of work for at least 2 one-week lab assignments), but not too extensive so that you can not finish it by the end of the semester.

Timeline: Deliverables

1. **Proposal** (minimum 500 words) **Deadline: Tuesday, April 2, 2019 by midnight:**
 - 1.1. Develop an idea for your project. Write a half page to one page technical description of what you propose to do for your project and share it through the project's GitHub repository. Your proposal (description of your project) does not need to be very detailed at this point. It should describe what you want to do (the problem you will tackle) and at least couple of references to show me that you have done some research about the problem. But you do not need to include any specifications on how you will solve your proposed problem.

1.2. I will provide the feedback with an approval or a suggested modification to your project proposal within a week.

2. **Progress report** (minimum 1000 words) **Deadline: April 16, 2019 by midnight:**
Describe everything you have done so far in your progress report, even if it is incomplete. By this point, you should have made tremendous amount of progress towards implementing the solution to your proposed project. Were there any unexpected challenges?
3. **Presentation April 23, in class and during the lab:** By the presentation session, you should have made a good progress towards implementation, and have something to showcase. In the presentation, you should describe the project description and the motivation behind it, approaches, and any outcomes you have so far. Explain how your chosen project fits the scope of the class (its relation to compilers). Use diagrams and a few bullet points rather than long sentences and screenshots of programs. The goal of the presentation is to convey the important high-level ideas and give intuition rather than be a formal specification of everything you did. Prepare for ~ 10 minute presentation, use visual slides, which include a slide with the title of your project and group members' names. Every member of the group needs to contribute to the presentation talk. At the end of the presentation give a demonstration of your project if appropriate.
4. **Final report and Source Code Deadline: May 3, 2019 by midnight:** Incorporate any feedback from the progress report and the presentation session. Your final report should be clearly and well written, this includes no typos or grammatical errors. Your report should be written in a professional and technical manner. The writing will be graded more harshly here. Your report should include:
 - The motivation for your project. Why is the problem you decided to solve important/useful?
 - Background for the proposed problem. What have others done for it already? Include references.
 - Detailed description of the work you completed for this project. Without giving a snapshot of the code you wrote, describe what you implemented and how you implemented it.
 - Description of your results. Make graphs, tables, snapshot of output, or anything else that can help me understand your results.
 - Conclusion. Give a short overview of your project and its results. Describe what you learned, what were the biggest challenges and the biggest rewards.

Grading rubric

10 points **Proposal**

20 points **Progress report and current implementation**

25 points **Presentation**

45 points Final report and final project implementation

For each deliverable the required items must be submitted via your project's GitHub repository. Your commit log will be used to assess the progress and timely completion of each project deliverable. When your project has been completed, your project repository must contain:

- README that describes the tools and languages used and outlines the steps for successful run of your software.
- `/writing` directory containing your proposal, progress report and final report documents, written in Markdown.
- `/src` directory containing all source programs.
- `/test` directory containing all test programs.
- `/output` directory containing sample output.

Since you have a lot of flexibility in this project, please review the Honor code, and make sure to follow it.