

CENG513 Final Exam - Spring 2024

Due date: 12.06.2024 23:00

Q.1 (30 Points) Select three LLVM [transform passes](#) (optimizations). Give small, simple C code examples that can benefit from each pass. You can either write your own codes or use an existing code (from an open source repository/benchmark suite) by providing the reference. Make sure that they are not so complicated to analyze. Apply each pass for a target code by using appropriate LLVM commands, generate IR and build control flow graphs for both original and transformed version. Briefly explain the effect of each pass by pointing out differences. You don't need to generate the executable and execute the code, it is enough to examine the IR and/or CFG structures.

You need to have six versions as the following and make three comparisons:

- Orig-Code-1 vs Code-1-After-Pass-1
- Orig-Code-2 vs Code-2-After-Pass-2
- Orig-Code-3 vs Code-3-After-Pass-3

Q.2 (20 Points) Look at the [open issues with clang-codegen label](#) at llvm github repository. Select one that you can understand and attempt to solve by using the skills you gained at CENG513 this semester. You may probably not be able to solve it, but you must select an issue that is the closest to the concepts we covered in the lectures or in the paper presentations. Explain the issue, relate it to the concepts in our code generation lectures (or paper presentations) and suggest a solution about how to fix it (we do not expect you to solve, just a suggestion is enough).

Q.3 (30 Points) Watch the following talk: <https://www.youtube.com/watch?v=abodcRZMWgg>, the keynote at 2022 CGO conference. Answer the following questions based on the video content:

- What are the changes in the last few decades that might be useful but not used by the compiler technology? How does each can improve compiler structure?
- What are three key points for proposed next-generation compiler technology? Explain each one briefly according to the information given in the talk.
- Choose two papers mentioned in the talk and read their Abstract and Introduction sections. Compare the proposed works by relating the concepts we discussed in our lectures. You can also look at the other sections, but you don't need to understand the implementation and experimental results. It is enough to know about the proposed work and its high-level definition. It is better to select papers that you can compare each other and relate to the concepts in our lectures.
- What is the question you want to ask to the speaker? Why?

Q.4 (20 Points) Based on the paper presentations and reproducibility attempts by yourself and your friends, write a short paragraph about the status of the reproducible research in the compiler studies. You must discuss the following questions by providing specific examples from your and your friends' experiences: What are the characteristics of reproducible research? What are the good practices to make it easier? What are the difficulties that can be encountered in the reproducibility efforts?

What to Submit:

- ✓ Four pdf files including the answer for each question (q1_StudentID.pdf, q2_StudentID.pdf, q3_StudentID.pdf, q4_StudentID.pdf).
- ✓ You need to put everything in your pdf submission. You can optionally submit any supplementary files or materials like LLVM IR outputs or CFG files.

Important Note: You need to use **your own** words/ideas in the answer of the questions. If you have any reference (paper, video, web site, AI, friend discussion etc.) used in your answers, you must explicitly cite them in your submission.