

Independent Study Proposal: Assembler Construction

Introduction

UNLV's Bachelor's of Science in Computer Science provides deep insight into the fundamentals of computation. However, one gap still exists between two foundational classes. That is, CS 218 and CS 219. The former provides students with insight to assembly and systems programming, supplying instruction for the lowest level of programming, while the latter shows the hardware organization at a high-level of computer engineering. However, the link between these two levels of abstraction is talked about relatively briefly, being simply a tool for completing assignments in CS 218 with only surface-level knowledge on it being provided. This tool is the assembler, a fundamental component in building modern-day programs for execution.

Outline

This project will be a two-pass assembler for a subset of x86 assembly language. More precisely, it will be only for the directives covered in CS 218, so as to not be too complex in scope yet still retain the insight that this project aims to gain. It will be written primarily in C++, with some files written in other languages to work with other libraries as deemed necessary. A two-pass assembler first identifies the labels in the source file and places such labels in a symbol table in the first pass. Then, the instructions are read on the second pass. As such, the first step in completing an assembler is scanning or lexing. This includes macros, which must be handled specially. On the second pass, the assembler will translate each directive into corresponding machine code, outputting object file(s) as necessary. Then, a linking loader will be used to link the relocatable object files to create one, cohesive executable program from the produced object files.

This development process will require information covered in CS 218, 219, 302, 326, 370, and 460. It necessitates applications of such concepts as data structures, scanning and parsing, low-level systems understanding, file system understanding and management, and much more in order to be achieved.

Conclusion

This project is large in scope and complexity at the lowest level of programming. In tackling a project of this propensity, I would expand my knowledge on the relationship between programming and computer organization, and processing as a whole. I will document my progress, drafting findings and lessons learned as I go along. Such findings can easily be shared within UNLV among professors and curious students alike. This project will particularly be beneficial to understanding the link between the lowest programmatic layer, being assembly language, with the highest organizational layer, being CPU instructions provided in bit-like format. As such, professors teaching CS 218 and CS 219 will gain insight to most effectively teaching such information to novice undergraduate students. This can help students bridge the gap in understanding between the two courses.