

acwj

Trganda

May 28, 2021

Abstract

1 Part 0: Introduction

I've decided to go on a compiler writing journey. In the past I've written some **assemblers**, and I've written a **simple compiler** for a typeless language. But I've never written a compiler that can compile itself. So that's where I'm headed on this journey.

As part of the process, I'm going to write up my work so that others can follow along. This will also help me to clarify my thoughts and ideas. Hopefully you, and I, will find this useful!

1.1 Goals of the journey

Here are my goals, and non-goals, for the journey:

- To write a self-compiling compiler. I think that if the compiler can compile itself, it gets to call itself a real compiler.
- To target at least one real hardware platform. I've seen a few compilers that generate code for hypothetical machines. I want my compiler to work on real hardware. Also, if possible, I want to write the compiler so that it can support multiple backends for different hardware platforms.
- Practical before research. There's a whole lot of research in the area of compilers. I want to start from absolute zero on this journey, so I'll tend to go for a practical approach and not a theory-heavy approach. That said, there will be times when I'll need to introduce (and implement) some theory-based stuff.
- Follow the KISS principle: keep it simple, stupid! I'm definitely going to be using Ken Thompson's principle here: "When in doubt, use brute force."
- Take a lot of small steps to reach the final goal. I'll break the journey up into a lot of simple steps instead of taking large leaps. This will make each new addition to the compiler a bite-sized and easily digestible thing.

1.2 Target Language

The choice of a target language is difficult. If I choose a high-level language like **Python**, **Go** etc.

2 Introduction