



DoctorWkt /
acwj



<> Code

Issues 19

Pull requests 2

Actions

Projects

Security

Insights

acwj / Readme.md



Warren Toomey Added a link to the new section.

3 years ago



94 lines (85 loc) · 5.24 KB

Preview

Code

Blame

Raw



A Compiler Writing Journey

In this Github repository, I'm documenting my journey to write a self-compiling compiler for a subset of the C language. I'm also writing out the details so that, if you want to follow along, there will be an explanation of what I did, why, and with some references back to the theory of compilers.

But not too much theory, I want this to be a practical journey.

Here are the steps I've taken so far:

- [Part 0](#): Introduction to the Journey
- [Part 1](#): Introduction to Lexical Scanning
- [Part 2](#): Introduction to Parsing
- [Part 3](#): Operator Precedence
- [Part 4](#): An Actual Compiler
- [Part 5](#): Statements
- [Part 6](#): Variables
- [Part 7](#): Comparison Operators
- [Part 8](#): If Statements
- [Part 9](#): While Loops
- [Part 10](#): For Loops
- [Part 11](#): Functions, part 1
- [Part 12](#): Types, part 1

- [Part 13](#): Functions, part 2
- [Part 14](#): Generating ARM Assembly Code
- [Part 15](#): Pointers, part 1
- [Part 16](#): Declaring Global Variables Properly
- [Part 17](#): Better Type Checking and Pointer Offsets
- [Part 18](#): Lvalues and Rvalues Revisited
- [Part 19](#): Arrays, part 1
- [Part 20](#): Character and String Literals
- [Part 21](#): More Operators
- [Part 22](#): Design Ideas for Local Variables and Function Calls
- [Part 23](#): Local Variables
- [Part 24](#): Function Parameters
- [Part 25](#): Function Calls and Arguments
- [Part 26](#): Function Prototypes
- [Part 27](#): Regression Testing and a Nice Surprise
- [Part 28](#): Adding More Run-time Flags
- [Part 29](#): A Bit of Refactoring
- [Part 30](#): Designing Structs, Unions and Enums
- [Part 31](#): Implementing Structs, Part 1
- [Part 32](#): Accessing Members in a Struct
- [Part 33](#): Implementing Unions and Member Access
- [Part 34](#): Enums and Typedefs
- [Part 35](#): The C Pre-Processor
- [Part 36](#): `break` and `continue`
- [Part 37](#): Switch Statements
- [Part 38](#): Dangling Else and More
- [Part 39](#): Variable Initialisation, part 1
- [Part 40](#): Global Variable Initialisation
- [Part 41](#): Local Variable Initialisation
- [Part 42](#): Type Casting and NULL
- [Part 43](#): Bugfixes and More Operators
- [Part 44](#): Constant Folding
- [Part 45](#): Global Variable Declarations, revisited
- [Part 46](#): Void Function Parameters and Scanning Changes
- [Part 47](#): A Subset of `sizeof`

- [Part 48](#): A Subset of `static`
- [Part 49](#): The Ternary Operator
- [Part 50](#): Mopping Up, part 1
- [Part 51](#): Arrays, part 2
- [Part 52](#): Pointers, part 2
- [Part 53](#): Mopping Up, part 2
- [Part 54](#): Spilling Registers
- [Part 55](#): Lazy Evaluation
- [Part 56](#): Local Arrays
- [Part 57](#): Mopping Up, part 3
- [Part 58](#): Fixing Pointer Increments/Decrements
- [Part 59](#): Why Doesn't It Work, part 1
- [Part 60](#): Passing the Triple Test
- [Part 61](#): What's Next?
- [Part 62](#): Code Cleanup
- [Part 63](#): A New Backend using QBE

There isn't a schedule or timeline for the future parts, so just keep checking back here to see if I've written any more.

Copyrights

I have borrowed some of the code, and lots of ideas, from the [SubC](#) compiler written by Nils M Holm. His code is in the public domain. I think that my code is substantially different enough that I can apply a different license to my code.

Unless otherwise noted,

- all source code and scripts are (c) Warren Toomey under the GPL3 license.
- all non-source code documents (e.g. English documents, image files) are (c) Warren Toomey under the Creative Commons BY-NC-SA 4.0 license.