

Preface

This text has been collecting dust for more than a year now. I am only publishing it, because it was pretty much *completed* when I abandoned it. Computer science has lost its appeal to me. It was a long and painful process. Let's not get into the details.

Computing in the 1980's, when I got started, was a magical thing. Programming a machine to do things automatically, all by itself, imagine that! The first programs I wrote were video games, because I could not afford to buy any commercial ones. Soon I discovered that the thing that interpreted my programs was itself a program. How cool was that! I read a lot of books and was a guest student at a university for several years. There was no Internet, so you had to go outside to collect the information you wanted.

As time passed, the magic faded away, though. Hackers turned into entrepreneurs, magic became business, fun became pressure, hobbies turned into projects, and everyone started to take themselves so terribly seriously. And then the complexity at every corner. Oh, the horror!

My first computer had 1024 *bytes* of RAM. My first compiler ran in 40K bytes of RAM and *compiled from RAM to RAM*. Yes, it kept its own code and both the source code *and* the object code in those 40K bytes of memory.

```
$ ls -lh /usr/bin/cc  
-r-xr-xr-x 95,329,088 Nov 30 2022 /usr/bin/cc
```

How times have changed.

Sure, complexity increases over time. Sure, this compiler can do things that my little toy compiler back in the days could not do. But **91** megabytes? At what point did things get completely out of control? This is *more than four times the size* of my first Unix system, *including* its C compiler.

Anyway, I digress.

In the beginning of 2023 some straw broke the camel's back. I do not even remember what it was, but one day I decided that I was done with all that. Never touch a computer again, except maybe for email and other mundane stuff. It was a good idea.

Some months later I watched myself writing another compiler book. Without any pomposity, without any complexity, without any pressure. I worked on it for maybe an hour per day. It was fun. It was simple. It was not this book. ¹

Really, you should read that book instead of this one. This one is a book that you put in your shelf and pretend that you have read it. Maybe skim over it, look at the figures. It is not much fun, although it describes the foundation of a fun idea: a compiler that everyone (with a little interest in computers) can understand. A compiler that is even simpler than the one I wrote some 40 years ago.

This book is its own antithesis. It takes a simple idea and describes it in the most incomprehensible terms. If you are into mathematics, computer science, and formalisms, you might even enjoy it. The formulae certainly have some strange aesthetics to them.

Really, I have no idea why I wrote this book, but since it is already finished, I might as well publish it. Maybe there is someone out there who will find a use for it, maybe even like it.

Nils M Holm, April 2024

¹ It was "Write Your Own Retro Compiler", you can find it on my homepage, <http://t3x.org>