Introducing a Neat Com-π-ler

Rushi Shah

25 February 2017

Right now I'm in a Computer Architecture class. It is a super cool class and our most recent project was implementing a compiler that takes code in a C like language (henceforth called the "pi" language) and generates x86_64 assembly code. Each individual spent two weeks implementing the initial compiler that worked with basic things like while loops, functions, if statements, local and global variables, etc. Then (and this is the crazy part) we split up into groups of 10-12 people and had a week-long code sprint to make coolest compiler we possibly could.

My team was called Hot Pi and had twelve students on it total including me. I was blown away with the amount we got done in a week, and the project is now open source!

1 Features

1.1 Error Reporting

1.2 Sound

1.3 Graphics

1.4 Structs

1.5 Scoped variables

1.6 Arrays

1.7 Pointers, Delays, and the Bell Keyword

1.8 For-loops

1.9 Types

1.10 Higher Order Functions

1.11 Comments and Random Numbers

1.12 User Defined Operators (Macros)

1.13 Switch Statements

2 Workflow

We ended the week with a single file that was 2,862 lines of C code. In retrospect we should have split that code up into multiple files, but you live and you learn. And even though each one of us poured our blood, sweat, and tears into that file, our git workflow was surprisingly smooth. We each worked on our own feature branches, and never pushed to master until we had resolved merge conflicts and all the tests passed. There were definitely head-aches resolving the merge conflicts, but that part of the process was just passed into the development process of a new feature rather than an after-thought. Other teams expressed their displeasure with the large team dynamic, but things turned out alright for us somehow because we followed good git practices. And boy did I get practice teaching beginners at git how to resolve their own merge conflicts.

3 Reflection

Overall this class has been very enjoyable. First of all it has introduced me to the incredible world of low-level programming. I don't expect to spend much time in my life doing low-level stuff, but it is very neat and I'm glad I at least have a foundation in it. Also, the class has yet again impressed upon me how incredible the Turing Scholars Honors Program

is at the University of Texas at Austin. I am not only impressed with the curriculum (which has been very informative and enjoyable) but consistently amazed at how innovative and hard-working my peers are. Hook 'em!