

The NoBeard Project

Peter Bauer

March 8, 2015

1 Introduction

1.1 Overview

The NoBeard Project aims to provide a field of experimentation in the area of assembler programming, compiler construction, debugging, memory management and further system related topics. Its main target group are students of computer science. Analogously to Tanenbaum's approach when creating MINIX as a play ground for programming of operating systems [6] the NoBeard project wants to provide such a play ground for different aspects of assembler programming and compiler construction.

1.2 Components

The following components are part of the project. We list all parts and give a very brief description of their intention.

- *NoBeard Machine*: a virtual machine with a small instruction set which is sufficient to write general purpose programs.
- *NoBeard Assembler*: a translator which translates NoBeard assembler programs into machine programs which can be executed on the NoBeard Machine.
- *NoBeard Compiler*: a translator which translates NoBeard programs (a procedural programming language) into NoBeard machine language.
- *NooBeard Compiler*: a translator which translates NooBeard programs (an object oriented programming language) into NoBeard machine language.
- *NoBeard Debugger*: a tool to debug NoBeard Assembler, NoBeard, and NooBeard programs.
- *NoBeard IDE*: An integrated development environment for developing programs for the NoBeard platform.

1.3 The Name

According to [4] the popularity of programming languages is strongly related to the fact whether its inventor(s) is/are are bearded m[ae]n or not. Well, the main aim of the programming language NoBeard is not to be popular, moreover it should give the reader a clear insight how the main principles of compiler construction are.

1.4 Initial Situation

There exist already first versions of some parts of the NoBeard project. Primarily an implementation of the NoBeard machine already exists which supports an instruction set for basic memory instructions like loading and storing values from and to memory and for basic integer instructions, e.g., addition, subtraction, etc.

Additionally a basic version of the NoBeard Compiler is available which supports the evaluation of arithmetic expressions, declaration of int, char and array of char variables and basic input and output operations.

Both programs are documented in the NoBeard report [3] which gives a formal framework of how NoBeard components should be documented. The terminology and notions used are close to the ones used in [13] and [1].

As far as things are implemented they were developed in a strictly test driven manner. Up to now it was aimed to reach a coverage level of at least 80 %.

2 Diploma Theses

2.1 Introduction

The list of possible diploma theses given here is provided as an enumeration of different releases of the NoBeard project. The release descriptions provide a set of features and goals which give sufficient material for a diploma thesis to be written by two diploma students. Since many of the features of releases stated later in this list depend on features planned in releases given earlier, it only makes sense to study the first thesis in detail. All further theses are listed here to give the student a broader overview in which context his/her work can be seen.

As mentioned in section 1.4 the implementation so far was done purely test driven. This development method is mandatory for all further releases. We strive for a coverage of 80 – 85 %.

The documentation of the theses has to be done in L^AT_EX. The basics of the formal requirements concerning the documentation are laid in the template handed over to the diploma students at the beginning of their writing.

The names of the releases follow the names of persons who made significant contributions to the field of compiler construction. According to the NoBeard naming principle we did not choose the most famous contributors to compiler construction but people who made very important input but did not gain this kind of popularity like Ritchie, Stroustrup, Gosling, etc. Furthermore, we looked for a balanced appearance of male and

female contributors. Each subsection starts with one sentence introducing this person and giving a brief statement which contribution (s)he did.

2.2 Liskov

Barbara Liskov (born November 7, 1939) is an American computer scientist. She led the design and implementation of CLU; Argus, the first high-level language to support abstract data types and implementation of distributed programs [10]. In the release Liskov the NoBeard project should have the following features:

The currently existing code base of the NoBeard Machine shall be migrated to jdk 8. The migration includes a refactoring of the code base towards Java 8 features like lambda expressions, streaming interface, etc. wherever possible.

A proper concept of a graphical user interface shall be developed. The concept has to focus didactic aims to enable the user to explore the execution cycle of an assembler instruction, the execution of programs on assembler level, the monitoring of stack frames, the expression stack, etc. this concept has to be implemented in Java using Java FX as GUI framework.

An assembler for the Machine shall be implemented. It shall support simple mnemonics, symbolic addresses, symbolic jump labels, etc.

The code base of the existing NoBeard Compiler shall be ported to Java 8. The port includes a refactoring of the code base towards Java 8 features like lambda expressions, streaming interface, etc. wherever possible.

The compiler has to be extended to fully support the following features

- Selections: if, else, else if
- Switch statements: supporting strings and range matching
- Loops: while, do while, and simple for loop
- Arrays: to be fully implemented, strings to be implemented as arrays of chars
- Functions: arbitrary many parameters, Input, Input Output, and Output parameters

Furthermore the concept shall be extended towards a more flexible and modern typing concept, where data types of variables shall be determined by the compiler based on their initial assignment (similar to Swift [2])..

The error reporting shall support a proper re-synchronization after errors.

2.3 Ichbiah

Jean David Ichbiah (March 25, 194 – January 26, 2007) was a French computer scientist and the initial chief designer (from 1977 – 1983) of Ada [8]. The release Ichbiah shall achieve the following goals:

- Debugger and IDE

- Multifile Programs
- Dynamic Memory Allocation

In particular in the debugger and IDE part the following goals and tasks have to be achieved:

- Analysis of usual debugging formats
- Implement debugger engine
- IDE with editor with syntax high lighting
- Integration of debugger engine

NoBeard has to support

- Functions: Complex return values
- Separately compilable units with type checking across unit boundaries
- A possibility to declare names as public and / or private
- Dynamically allocatable memory

2.4 Goldberg

Adele Goldberg (born July 7, 1945) is a computer scientist who participated in the development of the programming language Smalltalk-80 and various concepts related to object-oriented programming [9]. The release Goldberg shall achieve the following goals:

The compiler has to be extended to fully support the following features

- Enums: as value type
- Structs: as value type
- Class: as reference type
- Inheritance
- Basic collections

The runtime system and IDE shall support

- Intellisense
- Garbage Collector

2.5 Steele

Guy Lewis Steele Jr. (born October 2, 1954) is an American computer scientist who has, together with Gerald Sussman, defined and implemented Scheme the more puristic and minimalistic Lisp dialect compared to Common Lisp [12]. As described in [5] and [7] NoBeard and NooBeard grammars shall be modelled such that both language parsers can be generated using Coco. Furthermore the back end of the compiler shall be given, such that a full compiler is generated.

2.6 Hopper

Grace Murray Hopper (December 9, 1906 – January 1, 1992) was an American computer scientist and United States Navy rear admiral. She was one of the first programmers of the Harvard Mark I computer in 1944, and invented the first compiler for a computer programming language. She popularized the idea of machine-independent programming languages, which led to the development of COBOL [11]. In this release NoBeard goes Arduino.

- Port of the NoBeard Machine to the Arduino platform
- Enriched instruction set supporting access to Arduino's IO

References

- [1] Alfred V. Aho, Monica S. Lahm, Ravi Sethi, and Jeffrey D. Ullmann. *Compilers: Principles, Techniques, and Tools*. Addison Wesley, 2nd edition, 2006.
- [2] Apple. Swift - Apple Developer. URL: <https://developer.apple.com/swift/>.
- [3] Peter Bauer. The NoBeard Report. Technical report, HTBLA Leonding, Leonding, 2015.
- [4] Tamir Khason. Computer languages and facial hair – take two, 2008. URL: <http://khason.net/blog/computer-languages-and-facial-hair-%e2%80%93take-two/>.
- [5] Peter Rechenberg and Hanspeter Mössenböck. *Ein Compiler-Generator für Mikrocomputer*. Hanser, München, 2nd edition, 1988.
- [6] Andrew S. Tanenbaum. *Modern Operating Systems*. Pearson International Edition, 3rd edition, 2009.
- [7] Pat Terry. *Compiling With C# And Java*. Addison-Wesley Educational Publishers Inc, Harlow, England; New York, 2004.
- [8] Wikipedia. Jean Ichbiah, December 2014. Page Version ID: 639475683. URL: http://en.wikipedia.org/w/index.php?title=Jean_Ichbiah&oldid=639475683.

- [9] Wikipedia. Adele Goldberg (computer scientist), January 2015. Page Version ID: 644121205. URL: [http://en.wikipedia.org/w/index.php?title=Adele_Goldberg_\(computer_scientist\)&oldid=644121205](http://en.wikipedia.org/w/index.php?title=Adele_Goldberg_(computer_scientist)&oldid=644121205).
- [10] Wikipedia. Barbara Liskov, February 2015. Page Version ID: 646590588. URL: http://en.wikipedia.org/w/index.php?title=Barbara_Liskov&oldid=646590588.
- [11] Wikipedia. Grace Hopper, March 2015. Page Version ID: 650412004. URL: http://en.wikipedia.org/w/index.php?title=Grace_Hopper&oldid=650412004.
- [12] Wikipedia. Guy L. Steele Jr., March 2015. Page Version ID: 649890638. URL: http://en.wikipedia.org/w/index.php?title=Guy_L._Steele_Jr.&oldid=649890638.
- [13] Niklaus Wirth. *Compiler Construction*. Addison Wesley, Reading, Mass., June 1996.