

企画番号：2024-12

企画テーマ：0 から作るコンパイラ ～自作コンピュータ(ホンモノ)を作る～

代表者：桂田治輝 (Y220186)

メンバー：桂田治輝 (Y220186)

アドバイザー教員：芝 公仁

1 目的

自作 CPU を作成しようとしたときに、「コンピュータシステムの理論と実装」を読んだ後に CPU を作り始めると、その原理がよく理解できるという意見を見つけたため、自分もそれを読んでみようということである。ハードウェア、ソフトウェアのどちらであっても日頃活用しているコンピュータの内部を理解することは、コンピュータの最適化やパフォーマンスの向上を図るときに重要となる。今後ソフトウェアなどの分野を研究しようとする者が、その基礎的知識を持たないまま研究に取り組むことは遠回りになるのではないかと考えたためでもある。

コンピュータが動く仕組み

アプリ → OS → コンパイラ → VM変換器* → アセンブラ → ハードウェア

これを全部作りたい！

図 1-1 コンピュータが動く仕組み

2 計画

O'Reilly Japan より出版されている「コンピュータシステムの理論と実装^(7.1)」(以下、Nand2Tetris と呼ぶ)を読み、またその過程でアセンブラ、コンパイラなどを作成し、最終的に OS の完成を目的とする。この本の各章に設けられている課題に取り組むことで、システムソフトウェアの理解を深められると考えている。



図 2-1 コンピュータシステムの理論と実装

3 実施方法

Nand2Tetris は 13 章構成であり、第 5 章までの 16 ビット CPU などのハードウェアを制作するパートと、第 6 章からの CPU 上で動作するソフトウェアを制作するパートに分かれている。今回の企画では、第 6 章以降のソフトウェアのパート完成を目標とする。各章は、プログラム動作の説明の部分とそれを自らの手で実装する課題の部分に分かれている。各章を読み終えるごとに 1 つのプログラムが完成するようになっている。課題では設計手順が詳細に載っているわけではなく、あくまでその設計アイデアが記載されているのみであるため、各プログラムの詳細は読者自身が設計を考える必要がある。

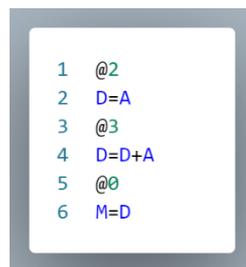
筆者は、すべてのプログラムを C++ で作成する。

4 活動経過

4.1 6 章 アセンブラ

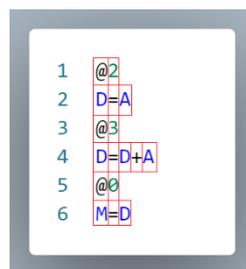
Nand2Tetris では、第 5 章までで制作した 16 ビット CPU に対する独自のアセンブリ言語 Hack を用いる。第 6 章では、この Hack から 16bit の 2 進数で構成される機械語に変換するアセンブラを制作する。

設計したアセンブラは、入力された Hack ソースコードをすべてトークン列に分割してから、各行ごとのトークンの組を 16bit の 2 進数に変換する。図 4.1-1、図 4.1-2、図 4.1-3 にそれぞれ “2+3” を計算し R0 レジスタに記録する Hack ソースコードと、それをトークンに分割したもの、その Hack ソースコードをアセンブルした機械語のソースコードを示す。



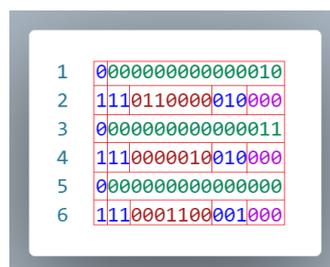
```
1 @2
2 D=A
3 @3
4 D=D+A
5 @0
6 M=D
```

図 4.1-1 “2+3”を計算し R0 レジスタに記録する Hack ソースコード



```
1 @2
2 D=A
3 @3
4 D=D+A
5 @0
6 M=D
```

図 4.1-2 図 4.1-1 のソースコードをトークン列に分割したもの



```
1 000000000000010
2 1110110000010000
3 000000000000011
4 1110000010010000
5 000000000000000
6 1110001100001000
```

図 4.1-3 図 4.1-1 のソースコードを機械語にアセンブルしたもの

特に、このアセンブラを作成する段階で重要となるのが、アドレスのラベルに対する処理である。アドレスのラベルについてはアドレステーブルで管理し、ラベルをアセンブルする都度、ラベルとアドレスの対応を参照する方法を取る。

4.2 7章 バーチャルマシン#1：スタック操作

Nand2Tetris では、Java 仮想マシン^(7.2)のようなスタックをベースとするバーチャルマシン (VM) を Hack プラットフォーム上で実装する。第7章では、このVMのスタック処理と、VM言語をHackへ変換するプログラムを実装する。具体的に、算術演算子やメモリアクセスに関する命令を実装する。メモリアクセスについて、仮想セグメントのベースアドレスを考慮しなくてよいことに注意しなければならない。

4.3 8章 バーチャルマシン#2：プログラム制御

第8章では、第7章で実装したVMのプログラムフローや関数呼び出しに関する具体的な処理を実装し、そのVM言語をHackへ変換するプログラムを実装する。関数呼び出しについて、アドレスに注意しなければならない。

4.4 9章 高水準言語

Nand2Tetris では、高水準言語として Jack と呼ばれる独自に定義した仕様の言語を用いる。第9章では、Jack を用いたプログラムを制作する過程で、その言語仕様を理解する。

4.5 10章 コンパイラ#1：構文解析

第10章では、Jack に対して構文解析を行い、解析結果をXMLファイルで生成するプログラムを実装する。Jack は、文脈自由文法、特に LL(1)文法である。これは、シンプルでエレガントな再帰下降構文解析によって簡単に扱うことができる。LL(1)文法と再帰下降構文解析の簡単な例を図4.5-1に示す。図4.5-1に示すのは、 $(2 + 3) \times 4 \div (1 + 4)$ という式を構文木で表したものである。

構文木とは、構文解析の結果を木構造であらわしたものである。下の葉から順に計算すると、元の式と同等であることが確認できる。

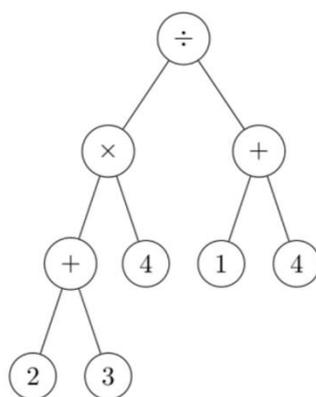


図 4.5-1 LL(1)文法と再帰下降構文解析の簡単な例

4.6 11章 コンパイラ#2：コード生成

第11章では、第10章で生成した解析結果をもとに、Jack をVM言語に変換するコンパイラを実装する。コンパイラを設計するときに作成したクラス図の一部を図4.6-1に示す。

4.7 12章 オペレーティングシステム

第12章では、Jack を用いてオペレーティングシステム命令(OS 命令)を実装する。ここでいう OS 命令とは、一般的なプログラミング言語における標準ライブラリに相当するものである。実装する命令には、乗除演算や画面出力、キーボード入力、文字列操作などがある。

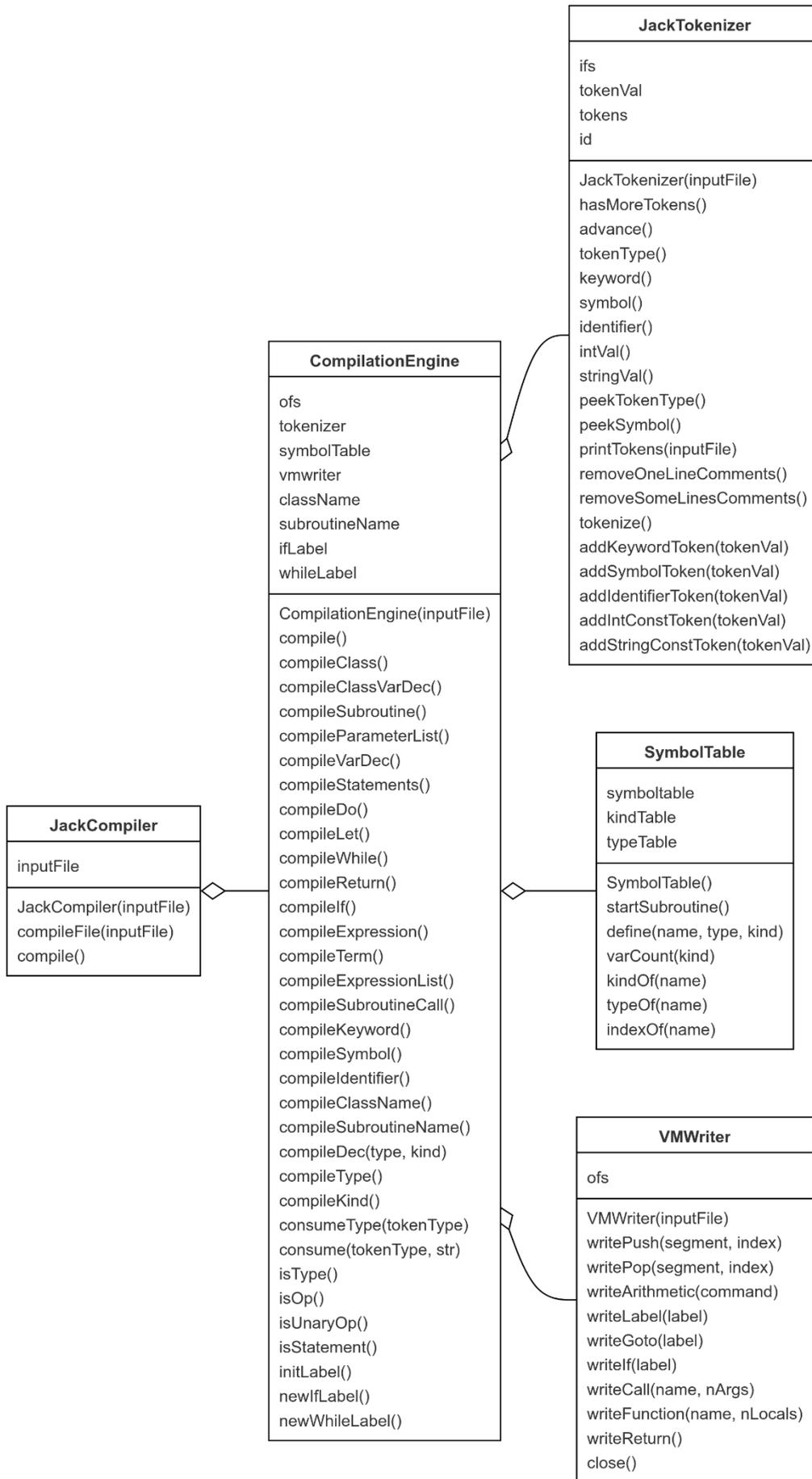


図 4.6-1 設計したコンパイラのクラス図

5 成果・結果等

この企画の成果はすべて筆者の GitHub のレポジトリ^(7.4)にある。まず `projects` ディレクトリ内の 0~13 までのディレクトリに、各章で配布されたプログラム^(7.3)などの雛形をもとにそのうえで筆者が実装したものと、それぞれのプログラムのテストを行うプログラムが格納されている。また、`assembler`、`vmtranslator`、`jackcompiler` ディレクトリに各プログラムの C++ソースコードと Linux の実行可能ファイルが格納されている。さらに `Assembler.md`、`JackCompiler.md`、`VMtranslator.md` には各プログラムを設計するときに作成したクラス図を記載している。このクラス図は、Mermaid^(7.5)により作成した。

また、この企画を実施するにあたり、X アカウントにおいて、感想ポスト^(7.6)を複数投稿している

6 今後の展望

今回の企画では、仮想のハードウェア上に独自の高水準言語 Jack を実装することができた。今後、Jack を実在するハードウェア上に実装したいと考えている。具体的には、Jack を ARM アーキテクチャの命令に変換するプログラムを C++で書き、Raspberry Pi 上に Jack を実装したいと考えている。

7 参考文献：

- 7.1 Noam Nisan, Shimon Schocken 著, 斎藤 康毅 訳. コンピュータシステムの理論と実装. O'Reilly Japan. 2015.
- 7.2 ウィキペディアの執筆者. “Java 仮想マシン”. ウィキペディア日本語版. 2023-04-20.
<https://ja.wikipedia.org/w/index.php?title=Java%E4%BB%AE%E6%83%B3%E3%83%9E%E3%82%B7%E3%83%B3&oldid=94862137>, (参照 2024-11-30).
- 7.3 nand2tetris. “From Nand to Tetris ~Building a Modern Computer From First Principles~”.
<https://www.nand2tetris.org/>, (参照 2024-11-30) .
- 7.4 nao109. “Nand2Tetris”. 2024-03-27. GitHub.
<https://github.com/nao109/nand2tetris>, (参照 2024-11-30) .
- 7.5 Mermaid. “Mermaid Diagramming and charting tool”.
<https://mermaid.js.org/>, (参照 2024-11-30) .
- 7.6 nao109(@Nao191020). “コンピュータシステムの理論と実装をやり始めた”. X. 2024-03-27.
<https://x.com/Nao191020/status/1772828798393921940>, (参照 2024-11-30) .
- 7.7 Scott M. King (smking). “Nand2Tetris”. 2012-11-12. GitHub.
<https://github.com/havivha/Nand2Tetris>, (参照 2024-11-30) .
- 7.8 Rui Ueyama. “低レイヤを知りたい人のための C コンパイラ作成入門”.
<https://www.sigbus.info/compilerbook>, (参照 2024-11-30) .

8. 補足

コンピュータシステムの理論と実装は、2024 年 12 月 2 日に 9 年ぶりの第 2 版(<https://www.oreilly.co.jp/books/9784814400874/>)が出版されるので、この企画を読んで同じように勉強したいと考えた方は、そちらを参考にされたい。