

はじめに

コンピュータを使えるようにするためには、まず自分の意思をコンピュータに知らせ、コンピュータで目的とする実行をさせることである。では、自分の意思をどうやってコンピュータに知らせるかが問題となる。コンピュータ黎明期においては機械語でプログラムを作成し、続いてアセンブラ言語でプログラムを作るようになった。しかしながら、人間が通常使う自然言語になるべく近いかたちでプログラムを作れたらということになり、高水準言語 COBOL や FORTRAN が生まれ、高水準言語を機械語に翻訳するコンパイラが開発された。その後、コンピュータ言語は PL/I, PASCAL, Ada などと発展し、それに伴い、コンパイラが各種開発されてきた。

このように、歴史的に各種のコンピュータ言語が開発されてきたが、オペレーティングシステム開発用に開発された C 言語が、簡単ながら、きめ細かい記述が可能なこともあり、一般的になってきた。また、Java がケータイ、スマホの開発に適したこともあり、Java もスマホプログラミングを中心に利用されてきている。本書では、主として C 言語をコンパイラの対象として説明し、Java についてはコンパイラの構成の観点から説明することとする。

本書は、15 週講義の教科書として使用することを想定しており、中間試験と最終のまとめの 2 週分を加味し、13 章構成となっている。各章のはじめにはその章のポイントやキーワードを示し、各章の内容を確認できるようにした。また、各章の終わりには演習問題をつけており、読者の理解度を確認できるようにしている。

本書の構成は以下の通りである。

まず、第 1 章でコンピュータアーキテクチャの簡単な紹介から、プログラムがどのように動作するかを述べる。続いて第 2 章では、コンパイラ設計に伴う準備として、コンピュータ言語の紹介から、コンパイラを開発するためにアルゴリズムを紹介し、言語が動作する仕組みを概説する。アルゴリズムが実際にコンパイラ上でどのように使用されるかについては、第 3 章以降で述べる。

第 3 章では、コンパイラの最初の処理である字句解析を紹介し、文字の並びからトークンへの並びへの解析について述べる。第 4 章では、プログラミング言語の持つべき文法について、その規定方法を述べ、コンパイラ処理に必要な構文木を中心に述べる。

第 5 章から第 9 章で、構文解析について述べる。まず、第 5 章では、トークンの列から、文法の一つである生成規則を逆に適用して、開始記号にさかのぼる道筋を求める下向構文解析につ

いて述べる。第 6 章では構文解析処理系作成ツールである yacc を用いて、構文解析処理の例を示す。第 7 章では、文法から終端記号の列を求める上向構文解析の概要と、SLR および LR と呼ばれる構文解析の概要を述べる。第 8 章では、yacc の仕組みを LALR 構文解析で述べる。第 9 章では、構文木に対して意味づけを加え、名前管理の方法を述べる。

第 10 章から第 13 章で、コンパイラから機械語が生成されるために、コンピュータとの関わりについて述べる。まず、第 10 章では機械語を生成するコード生成について説明する。そのために、仮定するコンピュータの機械語の命令を紹介し、コード生成、レジスタ割り当てを中心に述べる。第 11 章では、関数と手続きに関する処理方法を、関数呼び出し、引数、スコープなどについて述べる。第 12 章では、最適化について述べる。生成するコードは、実行速度などに多大に影響するため、機械独立レベル、機械依存レベル、機械語命令レベルで最適化が可能となる。第 13 章では、プログラムの実行方法として、インタプリタ型や仮想計算機型を紹介し、コンピュータ、オペレーティングシステムとの関わりと実現方式を述べる。

本書をまとめるにあたって大変なご協力をいただきました。未来へつなぐデジタルシリーズの編集委員長の白鳥則郎先生、編集委員の高橋修先生、岡田謙一先生、および編集協力委員の片岡信弘先生、松平和也先生、宗森純先生、村山優子先生、山田園裕先生、吉田幸二先生、ならびに共立出版編集制作部の島田誠氏、他の方々に深くお礼を申し上げます。

2014 年 2 月

執筆者 佐渡 一広
寺島 美昭
水野 忠則