情報系教科書シリーズ

コンパイラ

湯淺 太一 [著]

Ohmsha

まえがき

コンパイラを学ぶことには，いろいろな意義がある．これからコンパイラを作成しようとする人や，将来コンパイラを開発する可能性のある職業に就く人には，コンパイラの知識はもちろん不可欠である．しかし何らかの形でソフトウェア開発に関係する人なら，普段使用しているコンパイラの内部を知っておくことは重要である．これによって，プログラムの実行をより深く理解することができるし，プログラミング言語自体の理解も深まるからである．

プログラムを書く者にとって，コンパイラは身近なソフトウェアの一つである．規模の大きいソフトウェアであり，高水準のプログラミング言語を使用する抽象化された計算の世界と，現実の計算機のハードウェアやオペレーティングシステムとの橋渡しを行う．さまざまな理論やアルゴリズム，ソフトウェア開発手法，ノウハウが，コンパイラ開発には凝集されており，他のソフトウェア開発の参考となることが多い．大学の情報系カリキュラムでコンパイラが重視され，しかも比較的低学年を対象に講義が行われているのは，これらの理由によるところが大きい．

筆者は長い間，大学でコンパイラの講義を担当してきたが，コンパイラの研究者ではない．プログラミング言語の研究を行うために，必要にせまられて自らコンパイラを開発してきた．本書ではコンパイラの基礎的な理論も紹介しているが，理論そのものよりも，それらがコンパイラ開発にどう活かされるかに重点を置くことにした．雑多な知識を羅列するよりも，コンパイラを構築する上で必要不可欠な知識を選択し，それらが実際のコンパイラで活かされている様子をイメージできるように工夫した．

本書の内容は，さまざまなプログラミング言語や計算機アーキテクチャを対象としたコンパイラに応用可能であるが，具体例を示すために，C 言語と Pentium プロセッサを主たる対象とした．コンパイラを学ぼうとする人は，C 言語 (あるいは C++) の知識は持ち合わせているだろう．Pentium プロセッ

サについては，その詳細は知らないでも，最近の PC の急速な普及にともなって，多くの人にとって身近な存在となっている．その上，実際のコンパイラが生成するアセンブリコードが，Pentium プロセッサ用の場合は比較的読みやすい．実際，本書に掲載したアセンブリコードの生成例は，筆者の手元にある Pentium プロセッサ搭載の計算機上で，C コンパイラに生成させたコードに基づいている．

コンパイラを構成するさまざまな処理に対して，紙面の許す限り，処理プログラムを掲載して具体性を持たせるようにした．これらは基本的に C 言語で記述しており，部分的に抽象化してはいるが，具体化すれば実際に動作する処理プログラムである．そのほとんどは，実際の C 言語プログラムを作成してテストし，本書に掲載するために適宜抽象化したものである．

本書を理解するための前提知識は，あまり多くない．C 言語の初歩的な知識とプログラム作成経験 (コンパイラの使用経験は必須)，集合や論理演算の基礎知識があれば十分であろう．これらは大学の情報系学科などで 1 年半ほど勉強すれば，習得できているはずである．アセンブリ言語や機械語の知識は，期待するほうが無理なので，本書でかなりの紙面を割いて解説することにした．

本書は，コンパイラの処理順序にあわせて，字句解析，構文解析，意味解析，コード生成の順に章を構成している．基本的な事項を重視したので，最適化については概説にとどめた．本格的な最適化コンパイラでは，中間言語が重要な役割を果たすが，初心者にはその有用性を実感するのが難しいので，本書では言及しなかった．

本書の技術的内容について，筑波大学まで押しかけた筆者を親切に指導してくださった中田育男先生，執拗な訪問と電話で本書を完成に導いていただいた昭晃堂編集部の小林孝雄氏，本書の草稿を使って勉強し，質問やエラーを指摘してくれた京都大学の学生諸君，そして，家族に感謝します．

2001 年 3 月　　京都岩倉にて

湯淺太一

目　　次

1　コンパイラの概要

2　字句解析

3 文 法

4 構文解析

7 最適化