21世纪高等院校计算机科学与技术系列教材

主编　李劲华　丁洁玉

编译原理与技术

BIANYI YUANLI YU JISHU

北京邮电大学出版社
www.buptpress.com

编译原理与技术

主　编　李劲华　丁洁玉

北京邮电大学出版社
·北京·

内 容 简 介

本书介绍了计算机高级语言编译程序的基本原理和技术，主要内容包括词法分析、语法分析、语法制导翻译的语义分析与中间代码生成、符号表与运行时存储空间的组织、代码优化以及目标代码的生成。本书着重描述了编译构造的一些基础理论，如形式语言、有限自动机和属性文法。从构造编译程序的技术角度，描述了编译程序的各类算法，以及编译程序的自动构造工具，如词法分析生成器 Lex 和语法分析生成器 YACC。

本书系统性较强，基本概念阐述清晰，通俗易懂，便于阅读，可作为普通高等院校计算机学科及相关专业的本科教材，也可供教师、研究生及有关专业人员学习和参考。

图书在版编目(CIP)数据

编译原理与技术/李劲华，丁洁玉主编. —北京：北京邮电大学出版社，2005

ISBN 7-5635-1071-0

Ⅰ.编... Ⅱ.①李...②丁... Ⅲ.编译程序—程序设计 Ⅳ.TP314

中国版本图书馆 CIP 数据核字(2005)第 099752 号

出 版 者：北京邮电大学出版社(北京市海淀区西土城路 10 号) 邮编：100876

发行部电话：(010)62282185 62283578(传真)

电子信箱：publish@bupt.edu.cn

经 销：各地新华书店

印 刷：北京通州皇家印刷厂

开 本：787 mm×1 092 mm 1/16

印 张：20.75

字 数：496 千字

印 数：1—3 000 册

版 次：2006 年 1 月第 1 版 2006 年 1 月第 1 次印刷

ISBN 7-5635-1071-0/TP·182

定价：32 元

21世纪高等院校计算机科学与技术系列教材

编 委 会

序

计算机科学技术是科学性与工程性并重的一门学科。它的迅猛发展除了源于微电子学等相关学科的发展外，更主要源于其应用需求的广泛性不断增长，它已渗透到人类社会的各个领域，成为经济发展的倍增器，科学文化与社会进步的催化剂。计算机与通信的融合和全球联网，更显示出它无可限量的发展前景。任何一个领域的发展都离不开计算机已成为无可否认的事实。应用是计算机科学技术发展的动力、源泉和归宿，而计算机科学技术又不断为应用提供先进的方法、设备与环境。

近年来，计算机科学技术的发展不仅极大地促进了整个科学技术的发展，而且明显地推进了经济信息化和社会信息化的进程。计算机科学技术对一个国家在政治、经济、科技、文化、国防等方面的催化作用和强化作用都具有难以估量的意义。计算机知识与能力已成为21世纪人才素质的基本要求之一，因此，计算机科学技术的教育在世界各国都备受重视，我国政府和教育部门对计算机科学技术的教育及人才培养也非常重视。为了适应社会发展对计算机科学技术人才的强烈要求，各高校均在着力培养基础扎实、知识面广、综合素质高、实践能力强、富有创新精神，且具有较强的科学技术运用、推广、转化能力的高层次人才。

由北京邮电大学出版社联合北京邮电大学、武汉大学、华中理工大学及山东、江苏等多所高校的计算机专业教学负责人组成的“21世纪高等院校计算机科学与技术系列教材编委会”按照《中国计算机科学与技术学科教程2002》的要求组织编写的系列教材，体现了近年计算机学科的新理论、新技术。内容涵盖计算机专业学生所应掌握的相关知识，并根据目前计算机科学技术的发展趋势与实际应用相结合，能够满足目前高校计算机专业教学的需要，也可作为计算机专业人员的自学参考材料。

本系列教材作者均为多年从事教学、科研的一线教师，有着丰富的教学和科研实践经验，所编写的这套教材具有结构严谨，内容丰富、理论与实际结合紧密的特点，是他们的教学经验和科研成果的结晶。

计算机科学技术日新月异，所以教材也要不断推陈出新，我希望本系列教材能为我国高校计算机专业教育做出新的贡献。

中国工程院院士 金怡濂

前　言

编译程序是计算机的核心系统软件之一，属于 ACM/IEEE Computing Curriculum 2004 的核心知识域，是掌握计算机理论和软件技术的关键知识。编译原理和技术为人们理解计算机程序语言、创造优秀的软件奠定了理论基础，拓宽了视野，开辟了捷径。不妨浏览一下计算机的历史，就会发现很多被誉为程序设计大师的人无一不是编译领域的泰斗：创建了 Pascal 语言及其系列的图灵奖获得者 Niklaus Wirth 教授，写出第一个微型机上运行的 Basic 语言的 Bill Gates，成为 Sun 公司副总裁的 Java 缔造者 James Gosling 博士，C++之父 Bjarne Stroustrup 教授，被誉为“世界上最优秀的程序员”的 Delphi 架构师和 C#的缔造者 Anders Hejlsberg 等。编译领域堪为计算机程序设计英才的沃土。

编译的原理和技术可以应用在其他诸如软件建模语言、硬件描述语言、脚本语言等的翻译方面；在集成化软件开发环境以及软件安全、软件工程和软件逆向/再向工程中一直有着广泛的应用。而且，编译理论的研究有力地推动了计算机科学、计算机工程、软件开发以及人机工程等领域的研究和发展。

本书介绍了计算机程序语言编译程序的基本原理、设计方法和主要实现技术，可以作为普通高等学校计算机科学与技术及相关专业的教材和参考书。

本书系统性较强，基本概念阐述清晰，文字通俗易懂，便于读者学习和掌握。和其他同类教材相比，本书的特点如下：

(1) 在内容上增加了对面向对象语言一些特性的处理，讨论了 C++、C#和 Java 语言的例子。

(2) 对多数编译教材中以集合为主要数据结构的抽象算法描述进行了改进，更加详尽、深入浅出地描述了编译中的主要算法，以便于读者的阅读理解和计算机的实现。例如在自底向上的 LR 分析器中，本书采用了图的深度优先策略，以增量的方式构造出识别活前缀的有限自动机。

本书在组织上也进行了新的尝试，力图保持知识的逻辑性和连贯性，同时减少读者的阅读和理解的难度。

在第 1 章中概述了编译，以后各章按照编译程序的构成和编译过程的顺序，

逐步介绍编译的基本原理、设计方法和构造技术，把读者的思路和精力保持在编译程序的构造上，强调对编译原理和技术的宏观理解和全局把握，按照需要和逻辑关系阐述和讲解抽象的基础概念和理论；第 2 章首先介绍词法分析的设计和词法分析程序的手工构造，然后讲述有限状态自动机的理论以及它在词法扫描器自动生成中的应用；第 3 章集中讲解描述计算机编程语言的形式化语言，包括上下文无关文法的基本概念和等价变换；第 4 章介绍自顶向下语法分析方法，包括递归下降分析和表驱动的 LL(1)分析；第 5 章讨论自底向上的算符优先分析方法、各种类型的 LR 分析方法及其语法分析的自动生成。为了便于理解语义分析和代码生成，本书在第 6 章介绍了编译程序符号表的组织与管理；第 7 章讨论编译构造所需要的程序运行时的环境，包括运行时的内存分配和手工与自动化的管理；在第 8 章里对语义描述技术、属性文法以及语法制导的语义分析进行了详尽的阐述；第 9 章讨论了基于语法制导技术的中间代码翻译；第 10 章论及了目标代码生成的原理和技术；最后，在第 11 章集中介绍代码优化的基本技术，主要包括中间代码的局部优化和目标代码的优化方法。

本书每章都附有各种类型的练习题，便于读者理解基本概念和原理，掌握编译的基本算法和实现技术。

编译知识理论抽象，算法丰富，建议学习时配合上机实践，在加深对编译原理理解的同时，掌握编译程序的基本技术。编译中有很多算法，比如 NFA 到 DFA 的转换、求首符集、计算 LR 项集等，都可以作为上机实践题。

与本书配套出版的还有光盘，内容包括不同学时的教学计划，以及部分课外阅读和学习材料。本书另有配套的习题解答和上机指导教材。

使用本教材要求读者学习和掌握高级程序设计语言，如 C、Pascal 或 Java，最好具备离散数学、数据结构、计算机组成和汇编语言的基本知识。

本书结合作者李劲华多年的科研工作和教学实践编著，主要参考了文献[1]和[2]以及国内外许多学者的研究成果，在此表示诚挚的感谢。丁洁玉参与了本书大纲的讨论，并撰写了第 3 章和第 4 章的内容。

由于编者时间紧迫、水平有限，书中难免存在一些疏误和缺点，恳请广大读者批评指正。

编者

2005 年 7 月

目　录

第 11 章　代码优化

第1章 概 论

1.1 为什么学习编译

编译程序构造的原理和技术一直属于最近公布的 ACM/IEEE Computing Curriculum 2004 的核心知识域，已成为计算机科学必备的专业基础知识。而且，编译程序的构造是计算机科学中一个非常成功的分支，也是最早获得成功的分支之一，它所建立的理论、技术和方法值得人们深入研究和学习。

首先，编译构造正确地建立了研究的问题领域和研究方式：分析输入内容、构造一个语义表示并合成输出。对于不同的源语言，可以用一个单一的语义表示产生出不同的目标语言，运行在不同的环境中。而且，编译构造可以划分成便于控制和管理的阶段，每个阶段的工作结果正好对应编译程序的子系统或模块。编译程序的这种分析－合成模式以及解释程序的解释模式已经成为软件开发领域最成功的设计模式和软件架构，在软件开发中获得了广泛的应用。

其次，针对编译程序构造的某些部分已经开发了标准的形式化技术，依据这些形式化技术所研制的编译程序生成工具，极大地减轻了编译程序的构造工作，使得编译程序成为计算机系统最可靠的基础软件之一。这些形式化技术包括有限自动机理论、上下文无关文法、正规表达式、属性文法、机器代码描述、数据流分析方程式等。编译程序自动生成技术具有的高效性、正确性、灵活性、易更改性和易扩展性等优点，也扩大到普通的软件开发领域，例如，数据库程序的自动产生器以及从图形化的 UML 自动生成高级语言的程序。

第三，编译程序包含许多普遍使用的数据结构和算法，例如散列法(哈希算法)、栈机制、堆机制、垃圾收集、集合算法、表驱动算法、图算法等。尽管其中的每一种都可以独立地学习，但是，在诸如编译程序这样一个有意义的环境中学习将更有教育意义。

第四，编译程序的许多构造技术已经得到了广泛的应用。许多应用程序非常接近于编译程序，已经采用了编译构造的部分技术，例如读入格式化的数据、文件转换问题等。如果数据有清晰的结构，就可以为它设计文法，使用分析程序生成器自动产生一个分析程序。从编译程序构造技术收益的文件转换系统的范例包括著名的 Tex/Latex 文本格式化程序(Knuth 开发的、应用广泛的、共享的文字处理软件的文件格式)，该程序把 Tex 文本转换成机器独立的 dvi 格式；还包括 PostScript(一种普遍应用的、与机器和软件无关的文件格式，主要用于文件的传输和打印)解释程序，它将 PostScript 文本转换为打印机的指令。这种技术还有利于迅速引进新的文件格式，例如，能够快速地创建 HTML、XML 和 UML 等文件的读入和分析程序。

最后，学习编译原理和技术还有助于理解程序设计语言，以编写出优秀的软件。浏览一下计算机的历史，就会发现很多被称为程序设计大师的人都是编译领域的泰斗：创建了 Pascal 语言及其系列的图灵奖获得者 Niklaus Wirth 教授，写出第一个微型机上运行的 Basic 语言的 Bill Gates，成为 Sun 公司副总裁的 Java 缔造者 James Gosling 博士，Delphi 的架构师与 C# 的缔造者 Anders Hejlsberg，C++ 之父 Bjarne Stroustrup 教授等。

1.2 什么叫编译程序

编译程序是计算机系统经典、核心的系统软件。自从 20 世纪 50 年代出现了以 Fortran 为代表的计算机高级编程语言以后，在抽象层次和使用方便等方面都超过机器代码和汇编程序的高级编程语言层出不穷，使用过的已经超过数千种。而且，随着计算机系统更加广泛的应用，研制和开发高级编程语言的工作远远没有停止的趋势。按照现在的计算机体系结构和组成原理以及软件开发的理论和实践，高级程序语言仍将是开发计算机应用系统的关键技术，与之不可分离的是高级程序语言的编译程序。

用高级编程语言，如 Fortran，Pascal，Ada，Smalltalk，C，C++，C# 和 Java，编写程序方便而且效率高，但是，计算机需要把高级编程语言的程序翻译成机器语言代码或汇编程序才能运行。翻译程序或翻译器是把一种语言（源语言）转换成等价的另外一种语言（目标语言）的程序。如果源语言是高级编程语言，目标语言是机器代码和汇编语言这样的低级语言，这类翻译程序就叫做编译程序或编译器。如果目标语言是汇编语言，则还需要汇编成机器代码之后才可以运行。由于汇编语言和机器代码十分接近，像其他编译教材一样，本书通常不区分目标程序是机器代码还是汇编程序。

源程序的运行实质上包含两个过程：首先是把源程序用编译程序翻译成机器可以执行的目标程序或目标代码，然后才能接受输入数据运行，图 1.1 显示的就是这种编译执行方式。

运行高级语言程序的另外一种方式是解释执行，它需要的翻译程序不是编译程序，而是解释程序。解释程序不产生源程序的目标代码，而是对源程序逐条语句进行分析，根据每个语句的含义执行产生结果，如图 1.2 所示。Basic 和多数脚本语言都是按照解释方式运行的。解释方式的主要优点是便于对源程序进行调试和修改，但是其加工过程降低了程序的运行效率。

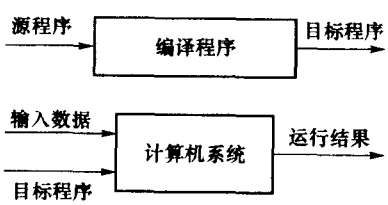

图 1.1 高级语言程序的编译执行方式

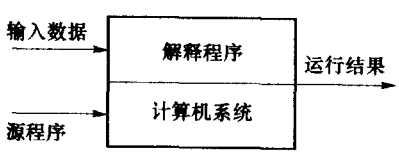

图 1.2 高级语言程序的解释执行方式

理论上,每一种语言都可以根据需要采用编译执行方式和解释执行方式,实践中每种语言都主要采用一种执行方式。Java语言同时具有编译执行方式和解释执行方式,分别应用于不同的计算机运行环境。由于编译程序和解释程序的原理基本一样,本书主要讲解编译程序的基本原理和技术,它们也可以适用于解释程序的构造与实现。

1.3 编译过程概述

把计算机高级编程语言翻译成计算机可以执行的代码的工作包括一系列的活动和任务,是一个复杂的完整过程。编译过程可以和把英语翻译成中文的过程相比较。在翻译一篇英文的时候,首先要确定英语的单词、标点符号等,把句子分解成单词以后去理解单词的基本含义;然后,分析英语句子的词组和语法结构,理解原句的含义,根据已知的中英文句型进行初步的翻译;接下来对译文进行修饰和润色,最后写出译文。

计算机程序的编译过程类似,一般划分为五个阶段:词法分析、语法分析、语义分析及中间代码生成、代码优化、目标代码生成。

1.3.1 词法分析

词法分析的任务是逐步地扫描和分解构成源程序的字符串,识别出一个一个的单词符号或符号。词法分析的工作主要包括:识别出程序中的单词符号,在编译程序符号表中查找并登记单词符号及其信息,譬如单词符号的类型、内部表示、数值等。计算机高级语言的单词符号通常包括:标识符、关键字或基本字、标点符号、常数、运算符、分隔符等。例如,对于C语言的while语句

```
while(i<100)sum+=i++;
```
(1.1)

词法分析的结果是识别出下列单词(见表1.1)。

表1.1 语句while(i<100)sum+=i++;的单词及其类型

符号	类型	符号	类型
while	关键字	(	分隔符
i	标识符	<	运算符
100	整常数	)	分隔符
sum	标识符	+=	复合赋值符
i	标识符	++	运算符
;	分隔符		

这些单词是C语言的基本单词符号,是构成C程序的基本组成成分,是人们理解和编写C程序的基本要素。在词法分析的过程中通常都把分隔符类型中的空格符号删掉。

编译程序的词法分析也叫词法扫描或线性扫描。本书的第2章将重点介绍词法分析的基本原理、技术和工具,如有限状态机、正规表达式及其之间的等价转换以及词法分析生成器Lex等。