清华大学计算机系列教材

编译原理

（第2版）

张素琴
吕映芝
蒋维杜
戴桂兰
编著

清华大学计算机系列教材

清华大学出版社

清华大学计算机系列教材

编译原理

（第2版）

张素琴　吕映芝　蒋维杜　戴桂兰　编著

清华大学出版社
北　京

内 容 简 介

本书介绍编译系统的一般构造原理、基本实现技术和一些自动构造工具。主要由语言基础知识、词法分析、语法分析、中间代码生成、代码优化、目标代码生成、符号表的构造和运行时存储空间的组织等部分组成。

书中在介绍编译程序构造基本原理的同时引入“PL/0 语言的编译程序”结构及文本，还引入 LEX、YACC 使用方法与实例。

本书是高等院校计算机科学与技术专业的本科生教材，也可作为教师、研究生或软件工程技术人员的参考书。

图书在版编目(CIP)数据

编译原理/张素琴等编著. 2 版. 北京:清华大学出版社,2005.2
(清华大学计算机系列教材)
ISBN 7-302-08979-5

Ⅰ. 编… Ⅱ. 张… Ⅲ. 编译程序－程序设计－高等学校－教材 Ⅳ. TP314

中国版本图书馆 CIP 数据核字(2004)第 064676 号

出 版 者：清华大学出版社　　**地　　址**：北京清华大学学研大厦
http://www.tup.com.cn　　**邮　　编**：100084
社 总 机：010-62770175　　**客户服务**：010-62776969
责任编辑：马瑛珊
印 刷 者：北京密云胶印厂
装 订 者：三河市李旗庄少明装订厂
发 行 者：新华书店总店北京发行所
开　　本：185×260　**印张**：29.5　**字数**：697 千字
版　　次：2005 年 2 月第 2 版　2005 年 9 月第4 次印刷
书　　号：ISBN 7-302-08979-5/TP·6351
印　　数：18001～23000
定　　价：35.00 元

作者简历

吕映芝 清华大学计算机系教授，1961年毕业于清华大学数学力学系计算数学专业。主要从事程序设计语言编译原理的教学工作、程序语言结构和自动生成工具的研究工作以及程序设计语言编译原理计算机辅助教学软件的研制和开发工作。

张素琴 清华大学计算机系教授。中国计算机学会系统软件专业委员会委员。1970年毕业于清华大学数学力学系。从事计算机专业基础课“编译原理”的教学和程序设计语言设计与实现、编译与编译自动生成技术等方面的研究工作。作为主要负责人和主要承担者，完成多项国家自然基金、“八五”、“九五”、“863”科技攻关任务。发表学术论文多篇，主编《程序设计语言C》和《编译原理》教材2部，译著6部。

蒋维杜 清华大学计算机系教授，长期从事语言编译、数据库、软件复用和面向对象软件工程的教学工作。负责并承担过国家“863”、“六五”到“九五”等高科技攻关项目，以及其他横向项目，并完成多项与国外公司的科技合作及软件出口项目。在交叉编译系统、数据库、信息系统及环境工具等方面取得多项成果，并获得中国科学院及电子部的科技进步奖。在长期培养研究生的教学及科研中，对面向对象方法及面向对象软件构造方面有较深入的研究。

戴桂兰 1972年生，博士，主要研究方向为程序语言，编译技术，软件测试技术，已发表论文二十余篇。目前在清华大学信息技术研究院Web与软件技术中心工作。

序

清华大学计算机系列教材已经出版发行了近30种，包括计算机专业的基础数学、专业技术基础和专业等课程的教材，覆盖了计算机专业大学本科和研究生的主要教学内容。这是一批至今发行数量很大并赢得广大读者赞誉的书籍，是近年来出版的大学计算机教材中影响比较大的一批精品。

本系列教材的作者都是我熟悉的教授与同事，他们长期在第一线担任相关课程的教学工作，是一批很受大学生和研究生欢迎的任课教师。编写高质量的大学（研究生）计算机教材，不仅需要作者具备丰富的教学经验和科研实践，还需要对相关领域科技发展前沿的正确把握和了解。正因为本系列教材的作者们具备了这些条件，才有了这批高质量优秀教材的出版。可以说，教材是他们长期辛勤工作的结晶。本系列教材出版发行以来，从其发行的数量、读者的反映、已经获得的许多国家级与省部级的奖励以及在各个高等院校教学中所发挥的作用上，都可以看出本系列教材所产生的社会影响与效益。

计算机科技发展异常迅速、内容更新很快。作为教材，一方面要反映本领域基础性、普遍性的知识，保持内容的相对稳定性；另一方面，又需要跟踪科技的发展，及时地调整和更新内容。本系列教材都能按照自身的需要及时地做到了这一点，如《计算机组成与结构》一书十年中共出版了三版，其他如《数据结构》等也都已出版了第二版，使教材既保持了稳定性，又达到了先进性的要求。本系列教材内容丰富、体系结构严谨、概念清晰、易学易懂，符合学生的认识规律，适合于教学与自学，深受广大读者的欢迎。系列教材中多数配有丰富的习题集和实验，有的还配备多媒体电子教案，便于学生理论联系实际地学习相关课程。

随着我国进一步的开放，我们需要扩大国际交流，加强学习国外的先进经验。在大学教材建设上，我们也应该注意学习和引进国外的先进教材。但是，计算机系列教材的出版发行实践以及它所取得的效果告诉我们，在当前形势下，编写符合国情的具有自主版权的高质量教材仍具有重大意义和价值。它与前者不仅不矛盾，而且是相辅相成的。本系列教材的出版还表明，针对某个学科培养的要求，在教育部等上级部门的指导下，有计划地组织任课教师编写系列教材，还能促进对该学科科学、合理的教学体系和内容的研究。

我希望今后有更多、更好的优秀教材出版。

清华大学计算机系教授，中科院院士

张钹

2002年6月28日

前　言

本书介绍程序设计语言编译程序构造的一般原理、基本设计方法、主要实现技术和一些自动构造工具。为计算机科学和技术专业的本科生提供教材。

尽管“编译程序”是特指将高级程序设计语言翻译成低级语言的软件，但编译程序构造的基本原理和技术也广泛应用于一般软件的设计和实现，因此本书也是从事系统软件和软件工具研究及开发者的参考书。

几年来的教学实践证明，本教材第一版的内容和架构都不错，受到广大读者的欢迎，且被一些院校选用，目前已出版 20 余万册。随着嵌入式系统的迅速发展和高性能体系结构的推陈出新，对支持多源语言多目标机的编译技术的研究显得尤为重要，同时，面向对象技术的兴起与广泛使用也对传统的编译技术提出了新的挑战和要求，这些发展变化应在教材中有所体现。本书在第一版的基础上，修改和添加了相应章节。首先从剖析一个简单的编译程序(PL/0)入手，对编译程序设计的基本理论，如有穷自动机、上下文无关文法等给予必要的介绍；对于广泛使用的语法分析方法和语义分析技术，如递归子程序法、算符优先分析、LR 分析及语法制导翻译等进行了详细的讲解；对编译程序的结构及其各部分功能、实现方法以及整体的设计考虑等给予了描述；还介绍了编译程序的构造技术，包括可重定向编译器的开发方法；此外，讨论了面向对象语言的编译技术以及利用面向对象方法构造编译程序的基本思想。并在附录中给出了 PL/0 的编译程序文本(包括 C 和 Pascal 两种语言文本)。

本书的第 1 章、第 3 章、第 4 章、第 8 章、第 10 章和第 11 章由张素琴编写，第 2 章、第 5 章至第 7 章由吕映之编写，第 9 章、第 14 章和第 15 章由蒋维杜编写，第 12 章和第 13 章由戴桂兰编写。另外，感谢蔡锐提供了 PL/0 编译程序的 C 版本。

书中若有不妥之处，请读者批评指正。

目　录

第1章 引 论

1.1 什么是编译程序

编译程序是现代计算机系统的基本组成部分之一，而且多数计算机系统都配有不止一个高级语言的编译程序，对有些高级语言甚至配置了几个不同性能的编译程序。从功能上看，一个编译程序就是一个语言翻译程序。语言翻译程序把一种语言（称作源语言）书写的程序翻译成另一种语言（称作目标语言）的等价程序。比如，汇编程序是一个翻译程序，它把汇编语言程序翻译成机器语言程序。如果源语言是像 FORTRAN、Pascal 或 C 那样的高级语言，目标语言是像汇编语言或机器语言那样的低级语言，则这种翻译程序称作编译程序。把编译程序看成一个“黑盒子”，它所执行的转换工作可以用图 1.1 来说明。

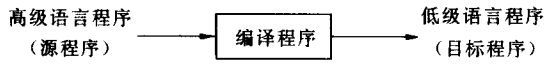

图 1.1 编译程序的功能

一个编译程序的重要性体现在它使得多数计算机用户不必考虑与机器有关的繁琐细节，使程序员和程序设计专家独立于机器，这对于当今机器的数量和种类持续不断地增长的年代尤为重要。

使用过计算机的人都知道，除了编译程序外，还需要一些其他的程序才能生成一个可在计算机上执行的目标程序。让我们分析一下一个程序设计语言程序的典型的处理过程（见图 1.2），可以从中进一步了解编译程序的作用。

一个源程序有时可能分成几个模块存放在不同的文件里，将这些源程序汇集在一起的任务，由一个叫做预处理程序的程序来完成，有些预处理程序也负责宏展开，像 C 语言的预处理程序要完成文件合并、宏展开等任务。图 1.2 中的编译程序生成的目标程序是汇编代码形式，需要经由汇编程序翻译成可再装配的机器代码，再经由装配/连接编辑程序与某些库程序连接成真正能在机器上运行的代码。也就是说，一个编译程序的输入可能要由一个或多个预处理程序来产生，另外，为得到能运行的机器代码，编译程序的输出可能仍需要进一步地处理。

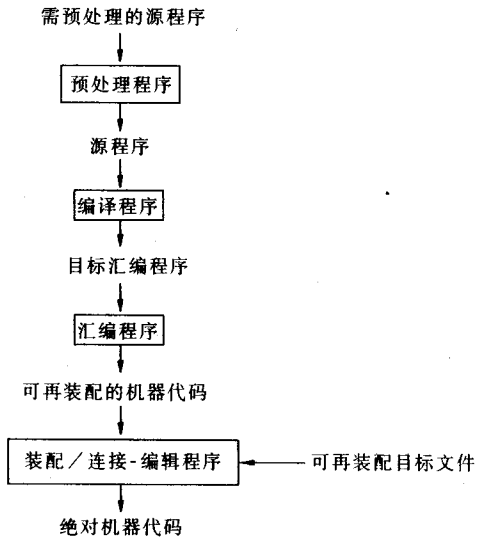

图 1.2 高级语言程序的处理过程

前面介绍过，编译程序的基本任务是将源语言程序翻译成等价的目标语言程序。我们

知道，源语言的种类成千上万，从常用的诸如 FORTRAN、Pascal 和 C 语言，到各种各样的计算机应用领域的专用语言，而目标语言也是种类繁多的，加上编译程序根据它们的构造不同，所执行的具体功能的差异又分成了各种类型，比如：一趟编译、多趟编译、具有调试或优化功能的等。尽管存在这些明显的复杂因素，但是任何编译程序所必须执行的主要任务基本是一样的，通过理解这些任务，使用同样的基本技术，我们可以为各种各样的源语言和目标语言设计和构造编译程序。

据说第一个编译程序的出现是在 20 世纪 50 年代早期，很难讲出确切的时间，因为当初大量的实验和实现工作是由不同的小组独立完成的，多数早期的编译工作是将算术公式翻译成机器代码。用现在的标准来衡量，当时的编译程序能完成的工作十分初步，如只允许简单的单目运算，数据元素的命名方式有很多限制。然而它们奠定了对高级语言编译程序的研究和开发的基础。20 世纪 50 年代中期出现了 FORTRAN 等一批高级语言，相应的一批编译系统开发成功。随着编译技术的发展和社会对编译程序需求的不断增长，20 世纪 50 年代末有人开始研究编译程序的自动生成工具，提出并研制编译程序的编译程序。它的功能是以任一语言的词法规则、语法规则和语义解释出发，自动产生该语言的编译程序。目前很多自动生成工具已广泛使用，如词法分析程序的生成系统 LEX，语法分析程序的生成系统 YACC 等。

1.2 编译过程和编译程序的结构

1.2.1 编译过程概述

编译程序完成从源程序到目标程序的翻译工作，是一个复杂的整体的过程。从概念上来讲，一个编译程序的整个工作过程是划分成阶段进行的，每个阶段将源程序的一种表示形式转换成另一种表示形式，各个阶段进行的操作在逻辑上是紧密连接在一起的，图 1.3 给出了一个编译过程的各个阶段，这是一种典型的划分方法。将编译过程划分成了词法分析、语法分析、语义分析、中间代码生成、代码优化和目标代码生成六个阶段。

我们通过源程序在不同阶段所被转换成的表示形式来介绍各个阶段的任务。

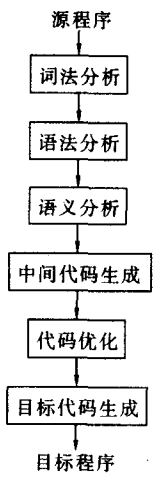

图 1.3 编译的各个阶段

词法分析 词法分析是编译过程的第一个阶段。这个阶段的任务是从左到右一个字符一个字符地读入源程序，对构成源程序的字符流进行扫描和分解，从而识别出一个个单词(也称单词符号或符号)。这里所谓的单词是指逻辑上紧密相连的一组字符，这些字符具有集体含义。比如标识符是由字母字符开头，后跟字母、数字字符的字符序列组成的一种单词。保留字(关键字或基本字)是一种单词，此外还有算符、界符等。例如某源程序片断如下：

begin var sum, first, count: real; sum := first + count * 10 end. 词法分析阶段将构成这段程序的字符组成了如下单词序列：

① 保留字	begin	② 保留字	var	③ 标识符	sum
④ 逗　号	,	⑤ 标识符	first	⑥ 逗　号	,
⑦ 标识符	count	⑧ 冒　号	:	⑨ 保留字	real
⑩ 分　号	;	⑪ 标识符	sum	⑫ 赋值号	:=
⑬ 标识符	first	⑭ 加　号	+	⑮ 标识符	count
⑯ 乘　号	*	⑰ 整　数	10	⑱ 保留字	end
⑲ 界　符	.				

可以看出,5 个字符即 b,e,g,i 和 n 构成了 1 个称为保留字的单词 begin,2 个字符即：和＝构成了表示赋值运算的符号：＝。这些单词间的空格在词法分析阶段都被滤掉了。

我们使用 id1,id2 和 id3 分别表示 sum、first 和 count 这 3 个标识符的内部形式,那么经过词法分析后上述程序片断中的赋值语句 sum ：＝first＋count * 10 则表示为 id1 ：＝id2＋id3 * 10。

语法分析　语法分析是编译过程的第二个阶段。语法分析的任务是在词法分析的基础上将单词序列分解成各类语法短语,如“程序”、“语句”、“表达式”等。一般这种语法短语,也称语法单位,可表示成语法树,比如上述程序段中的单词序列：

id1 ：＝id2＋id3 * 10　经语法分析得知其是 Pascal 语言的“赋值语句”,表示成如图 1.4 或图 1.5 所示的语法树。

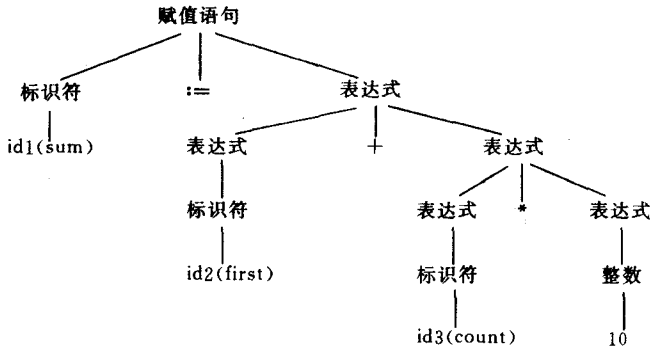

图 1.4　语句 id1 ：＝id2＋id3 * 10 的语法树

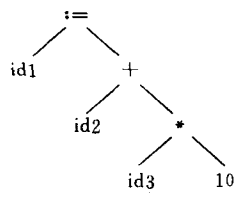

图 1.5　语句 id1 ：＝id2＋id3 * 10 的语法树的另一种形式

语法分析所依据的是语言的语法规则,即描述程序结构的规则。通过语法分析确定整个输入串是否构成一个语法上正确的程序。

程序的结构通常是由递归规则表示的,例如,我们可以用下面的规则来定义表达式：

(1) 任何标识符是表达式。

(2) 任何常数(整常数、实常数)是表达式。