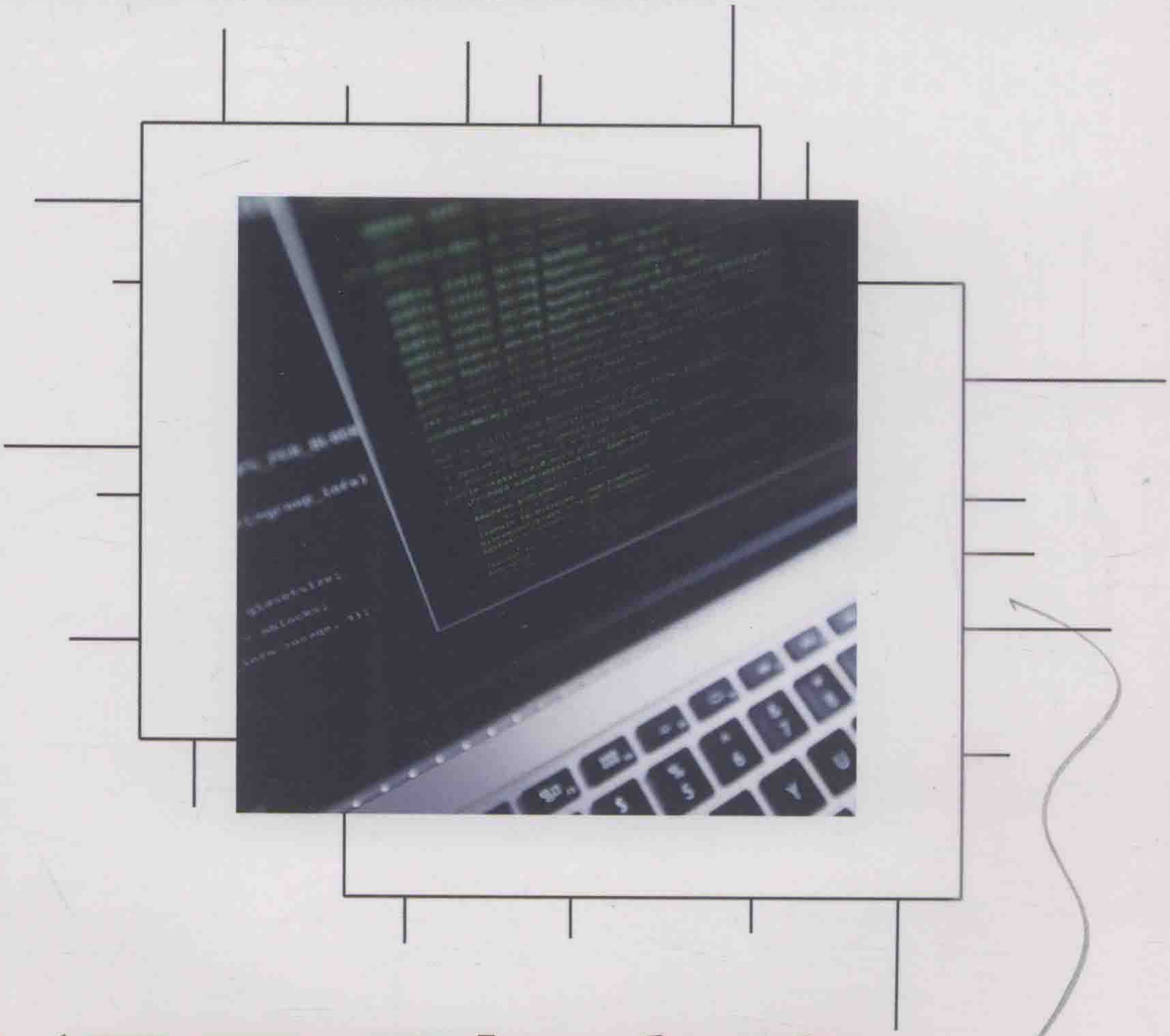
国家新闻出版改革发展项目库入库项目
普通高等教育“十三五”规划教材
全国高等院校计算机基础教育研究会重点立项项目



编译原理与实践

主　编　鲁　斌
副主编　李继荣　黄建才　闫　蕾　岳　燕

“互联网+”创新型教材
北邮智信

北京邮电大学出版社
www.buptpress.com

策 划 人：张珊珊
责任编辑：张珊珊
封面设计：七星博纳

BIANYI YUANLI YU SHIJIAN

提供免费教学资源：包括5个演示动画、65张教学图片、精美教学PPT等。

配套云资源的使用说明

扫一扫，下载安装“北邮智信”APP

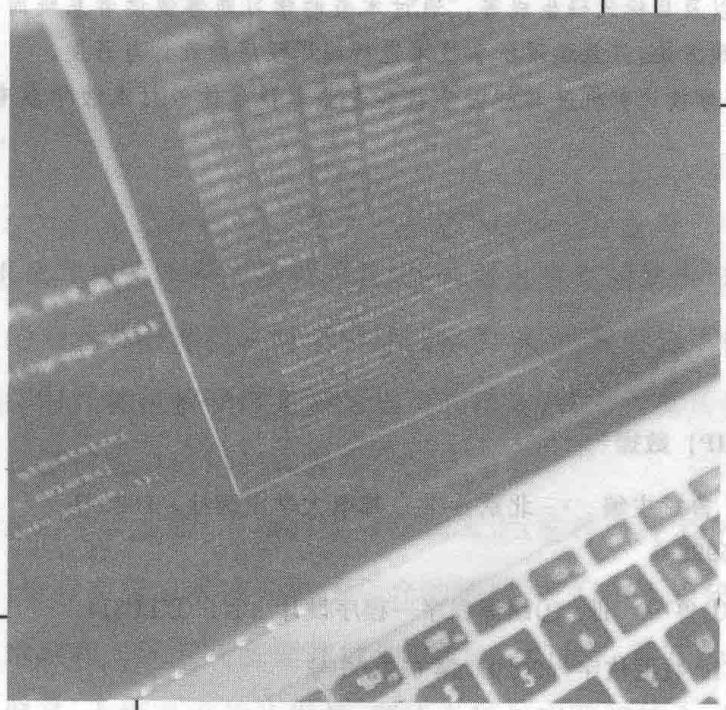
刮开涂层，在“北邮智信”APP中验证正版教材，加载免费资源

策划中心
电　话：010－62282796
E-mail：8956096@qq.com

ISBN 978-7-5635-5993-0

定价：36.00元

国家新闻出版改革发展项目库入库项目
普通高等教育“十三五”规划教材
全国高等院校计算机基础教育研究会重点立项项目



编译原理与实践

主　编　鲁　斌
副主编　李继荣　黄建才　闫　蕾　岳　燕

北京邮电大学出版社
www.buptpress.com

内容简介

编译的精髓在于做到原理、技术与实践方法的融会贯通，本书正是这样一部综合、全面、实用的编译技术教材。本着知识与能力相结合、理论与实用并重的指导思想，以贯穿全书的样本语言编译器的开发为例，在简要介绍编译技术所涉及的基本知识和高级语言的语法描述方法之后，按照编译程序的工作过程逐步介绍编译各个阶段的主要内容，具体包括：词法分析、语法分析、语义分析与中间代码生成、符号表与运行时存储空间组织、代码优化以及目标代码生成等。通过本书的学习能够使读者系统而全面地掌握编译各个阶段的基本原理、技术和实践方法，并且运用所学技术进行编译程序的设计与开发。

本书可用作高等学校计算机及其相关学科各专业本科生的教材或教学参考书，也可供其他技术开发人员参考。

图书在版编目（CIP）数据

编译原理与实践 / 鲁斌主编. -- 北京 : 北京邮电大学出版社，2020.1
ISBN 978-7-5635-5993-0

Ⅰ. ①编… Ⅱ. ①鲁… Ⅲ. ①编译程序—程序设计 Ⅳ. ①TP314

中国版本图书馆 CIP 数据核字（2020）第 012405 号

书　　名：编译原理与实践
责任编辑：张珊珊
出版发行：北京邮电大学出版社
社　　址：北京市海淀区西土城路 10 号（邮编：100876）
发 行 部：电话：010-62282185　传真：010-62283578
E-mail：publish@bupt.edu.cn
经　　销：各地新华书店
印　　刷：保定市中画美凯印刷有限公司
开　　本：787 mm×1 092 mm　1/16
印　　张：14.5
字　　数：376 千字
版　　次：2020 年 1 月第 1 版　2020 年 1 月第 1 次印刷

ISBN 978-7-5635-5993-0　　定　价：36.00 元

前　言

编译技术作为计算机及其相关学科的基础课程之一，经过多年的发展，现已成为一门理论体系完备、实践技术可行的专业核心课程。对于编译技术的学习，其价值不仅仅在于掌握了一门课程知识，更有助于加深对计算机软硬件系统的认识和理解，树立大型系统软件开发的思路和体系，有助于学生基本计算机素养的培养和提高。

当前，已出版的编译教材不在少数，有些以原理介绍为主，有些以技术介绍为主，还有一些则以编程实践为主，当然，也存在着二者或三者相结合的情况，它们中不乏佼佼者。严格来说，把编译划分为原理型、技术型和实践型其实很难做到，因为它们是一种互为依托、相辅相成的关系，离开谁都不全面。但是，大家共有的认识是：理论必须与实践相结合，使学生既懂基本的原理，又能够掌握主要的技术，更重要的是能够将所学应用于实践当中。如何在有限的学时中达到上述目的，重新组织和安排内容体系结构，按照认知的基本规律，有条理、有逻辑地逐步分析编译内涵，是本书撰写的主要目的。

本书是一部综合、全面、实用的编译技术教材，反映了作者多年来的教学思路和经验，经过反复推敲、几易其稿后得以成文。与已出版的同类书籍相比，具有贴近教学、层次分明、体系完整等特点和教学参考价值，具体体现在以下三个方面。

(1) 贴近教学。编译的精髓在于做到原理、技术与实践方法的融会贯通，而非只知其一。若想做到这一点，就有限的学时而言，确实是一件很困难的事情。这就需要在充分理解编译内涵的基础上，对众多的编译技术知识重新梳理，归纳总结，详略得当、重点分明、通俗易懂地安排教学内容。这正是本书撰写的整体思路。

(2) 层次分明。本书是对作者长期教学与实践经验的良好总结，在知识点的逻辑结构上按照层次化的教学理念，围绕横纵两条主线将编译系统逐步解剖开来。横向上，按照编译程序的工作过程组织章节，纵向上按照先原理，再技术，然后实践方法的顺序安排内容，从而使读者学习思路清晰，不仅知其然，而且知其所以然，由浅入深，循序渐进，以适合教与学。

(3) 体系完整。本书特别强调知识与能力结合，理论与实用并重。在讲授基本原理和技

术的同时，也讲授相关的实践知识，力图培养读者扎实的动手实践能力，使读者能够深入地运用所学技术进行编译程序的设计与开发。

本书力求达到如下目标。

(1) 使读者系统而全面地掌握编译各个阶段的基本原理、技术和实践方法。

(2) 层次化的章节组织、通俗易懂的文字描述，使学习过程不再枯燥难懂，而变得简单易学。

(3) 贯穿全书的样本语言编译器的设计和实现过程使读者运用编译技术时能够得心应手。

本书在内容的安排上，首先简洁明了地介绍了编译技术所涉及的基本知识和高级语言的语法描述方法，然后按照编译程序的工作过程逐步介绍编译各个阶段的主要内容，具体包括：词法分析、语法分析、语义分析与中间代码生成、符号表与运行时存储空间组织、代码优化以及目标代码生成等。上述安排能够有效地帮助读者系统地理解和掌握编译的主要知识和技术。

本书由鲁斌主编，并负责前两章的撰写工作；第 3 章由岳燕编写；第 4、5 章由李继荣编写；第 6、7 章由黄建才编写；最后两章由闫蕾编写；鲁斌负责统稿。

由于编者水平有限，错误之处在所难免，恳请广大读者批评指正。

编　者

“北邮智信”App 使用说明

目　录

第1章　绪　论

高级语言编译程序是计算机最为重要的系统软件之一。任何一门高级语言若想真正在计算机上被执行，都必须通过相应的翻译程序将其翻译为机器语言才行。翻译程序的典型代表是编译程序，开发编译程序所涉及的有关原理、方法和技术具有普适性的特点，广泛应用于与计算机相关的各个领域之中，因此，学习编译技术具有十分重要的理论意义和应用价值。本章首先介绍编译程序的基本概念和应用，然后介绍编译的过程与结构，最后对编译程序的构造方法进行简单说明。

1.1　编译程序简介

世界上最早的编译程序是20世纪50年代中期研制成功的FORTRAN语言编译程序。那时，人们普遍认为设计和实现编译程序是一件十分困难的事情。后来，为了改善这种状况，人们花费了大量时间来研究编译程序的自动构造技术，开发出了用于词法和语法分析的自动生成工具Lex与Yacc。在20世纪70年代后期和80年代早期，许多项目都致力于编译程序其他部分的自动生成，其中也包括了代码生成的自动化。

为什么需要编译程序

经过半个多世纪的不懈努力，编译理论与技术得到迅速发展，现在已形成了一套较为成熟的、系统化的理论和方法，并且开发出了为数不少的、优秀的编译程序的实现语言、环境与工具，在此基础上设计并实现一个编译程序不再是高不可攀的事情。

目前，编译程序的发展与复杂的程序设计语言的发展密切结合，编译程序也已成为交互式开发环境(IDE)的一个核心组成部分。随着多处理机和并行技术、并行语言的发展，将串行程序转换成并行程序的自动并行编译技术正在深入研究之中。另外，嵌入式应用的快速增长也极大地推动了编译技术的飞速发展。同时，对系统芯片设计方法和关键EDA技术的研究也带动了VHDL等专用语言及其编译技术的不断深化。

1.1.1　编译概述

计算机系统中的语言可以分为3个层次：机器语言(Machine Language)，汇编语言(Assembly Language)以及高级语言(High-level Language)。计算机刚出现时，人们编写程序使用的是机器语言。用机器语言编程费时、费力、难度大，因此，很快就被汇编语言取而代之。汇编语言以助记符的形式表示指令和地址，与机器语言相比尽管程序开发的难度有所降低，但仍与人类的思维方式相差甚远，不易阅读和理解，并且严格依赖于机器，在一种机器上编写的代码应用于另一种机器时必须完全重写。高级语言的出现拉近了人类思维和计算机语

程序设计语言分类

言间的距离，使得编写高级语言程序就如同书写自然语言一般，并且与机器无关。

然而，只有机器语言程序才能直接在计算机上执行，因此，必须寻找一种方法，能够将高级语言和汇编语言程序转换为对应的机器语言程序，这种方法就是翻译。通常所说的**翻译程序**(Translator)是指能够把某一种语言(称为**源语言**，Source Language)书写的程序转换成另一种语言(称为**目标语言**，Target Language)书写的程序，而后者与前者在逻辑上是等价的。如果源语言是诸如FORTRAN、Pascal、C或Java这样的“高级语言”，而目标语言是诸如汇编语言或机器语言之类的“低级语言”，则称这样的翻译程序为**编译程序**(Compiler)或**编译器**。

编译器的作用

根据用途和功能上的不同，编译程序可以分为以下几种类别。专门用于帮助程序开发和调试的编译程序称为**诊断编译程序**(Diagnosic Compiler)。着重于提高目标代码运行效率的编译程序叫作**优化编译程序**(Optimizing Compiler)。运行编译程序的计算机称为**宿主机**，而运行编译程序所产生目标代码的计算机则称为**目标机**。如果一个编译程序产生不同于其宿主机的目标代码，则称其为**交叉编译程序**(Cross Compiler)。如果不需要重写编译程序中与机器无关的部分就能改变目标机，则称该编译程序为**可变目标编译程序**(Retargetable Compiler)。目前，很多编译程序具备多种功能，往往是上述不同类型编译程序的集合体，用户只需简单地通过选项设置即可实现不同功能。

动画：编译程序的作用

高级语言易编程、易调试、效率高的特点使得程序员开发程序成为一件相对轻松的事情，然而高级语言源程序要想在计算机上最终被执行，除了需要编译程序的处理外，还需要预处理程序、汇编程序以及装配连接程序等的相互配合才能达到目的。一个典型的高级语言程序的处理过程如图1-1所示。

高级语言程序的处理过程

其中，**预处理程序**(Preprocessor)是指既能够将位于不同文件中的源程序模块汇集在一起，又能够将源程序中的宏语句展开为原始语句并加入源程序中的程序。**汇编程序**(Assembler)是把汇编语言程序转换为机器语言程序的程序。**装配连接程序**(Loader-linker)是指将可装配的机器代码进行装配连接得到绝对机器代码的程序。预处理结束后，源程序经过编译过程生成目标程序，对于图1-1所示的汇编语言目标程序，需要经过汇编过程转换为可装配的机器代码，再经装配连接过程才能得到真正可执行的绝对机器代码。

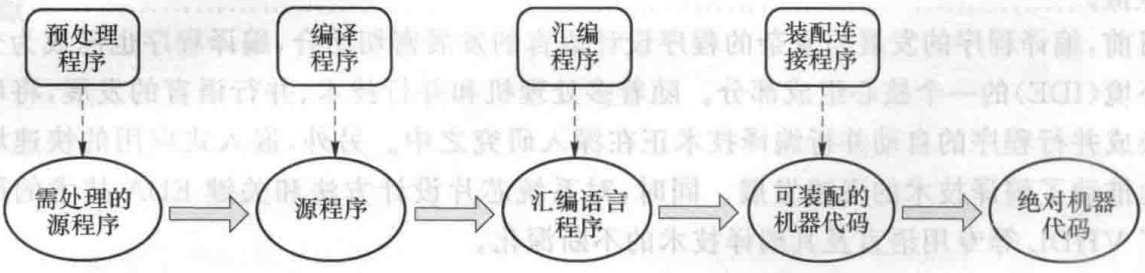


图1-1　高级语言程序的处理过程

高级语言除了可以“编译”执行外，也可以“解释”执行。**解释程序**(Interpreter)是指按高级语言程序的语句执行顺序边翻译边执行相应功能的程序。编译程序和解释程序的主要区别如下：

编译 VS 解释

(1) 编译程序将整个源程序翻译为目标程序之后再执行,而且目标程序可以反复执行。

(2) 解释程序对源程序逐句地翻译执行,不产生目标程序。若需再次执行,则必须重新解释源程序。

本书重点介绍编译程序的基本原理、方法和技术,至于解释程序则不做专门探讨。但是,书中有关编译程序的构造与实现技术同样也适用于解释程序。

编译型语言执行流程

混合型语言执行流程

解释性语言执行流程

1.1.2 编译技术的重要性

高级语言的不断向前发展促使编译程序日益完善起来,现已形成了一套系统化的理论和方法。学习编译技术对于提高计算机软硬件素养均具有举足轻重的作用,主要体现在如下几个方面:

(1) 有助于深刻理解和正确使用程序设计语言。学习编译技术之前,读者在开发程序时仅仅能够按照语法规则来编写代码,缺乏对语言机理的深入理解,只知其"形",不知其"神"。例如,全局变量和局部变量的本质区别是什么?函数的递归调用又是如何实现的?诸如此类的问题,通过学习编译技术都可以得到很好的解答,从而有助于读者深刻理解和正确使用程序设计语言。

ACM 图灵奖在程序语言及编译方面的获奖历程

(2) 有助于加深对整个计算机系统的理解。在目标代码的生成过程中,编译程序的内容涉及计算机内部的组织结构和指令系统,也涉及计算机的动态存储管理,还有可能涉及操作系统方面的知识。可以形象地说,编译程序如同一根红线,它把程序语言、算法与数据结构、计算机组成、指令系统以及操作系统等各方面的知识串联起来,使读者通过学习编译技术能够加深对整个计算机系统的理解。

(3) 编译程序的设计与开发技术具有普适性。本书在讲述编译技术时采用的是"自顶向下""逐步求精"的结构化程序设计思想,将整个编译程序划分为几个相对独立的功能模块分别予以介绍,从而将复杂的问题简单化。这样做既便于读者掌握各模块的构建方法和技术,又能够有效地保证程序的正确性、可靠性和可维护性。这种程序设计技术同样可用于其他软件的设计与开发之中。

(4) 编译技术的应用越来越广泛。编译技术不仅仅应用于编译器的开发当中,还广泛应用于文本编辑、程序调试、语言转换以及排版系统等领域内。近年来,随着微处理器技术的飞速发展,处理器性能的好坏在很大程度上取决于编译器的质量。因此,编译技术当之无愧地成了计算机的核心技术,地位变得越来越重要。

1.1.3 编译技术的应用

编译技术的应用领域很广,最为典型的当属程序设计语言方面。为了提高编程的效率,缩短调试的时间,确保程序的可靠性,软件工作者研发了不少用于程序语言处理方面的工具。开发这些工具不同程度地应用了编译程序各个部分的方法和技术,具体表现在以下 4 个方面。

(1) 结构化的程序编辑器。这种编辑器除了具备基本的文本编辑功能外,还能够像编译程序那样对程序文本进行分析,从而使程序的书写规范化。代表性的软件工具有 Editplus 和

Ultraedit 等。

(2) 程序调试工具。结构化编辑器只能解决语法错误的问题,对于一个已经通过编译的程序而言,需要进一步了解程序的执行过程是否满足算法的设计要求,是否实现了预期的功能以及程序的运行结果是否正确等,而这些工作都可以利用调试工具来完成。调试工具的开发主要涉及程序的语法和语义分析技术,调试功能越强大,实现起来就越复杂。

(3) 程序测试工具。一般来说,程序的测试工具包括两种,一种是静态分析器,一种是动态测试器。其中,静态分析器用来对程序进行静态的分析,主要工作包括对程序进行语法分析并查填相关表格,检查变量定值与引用的关系等。例如,检查变量是否未定值就引用,或定值后未被引用,或源代码冗余等一些编译程序的语法分析所发现不了的错误。动态测试器通过在程序的合适位置插入一些信息,结合测试用例来记录程序的实际执行路径,并将运行结果与期望的结果进行比较分析,以此来帮助程序开发人员查找所存在问题。C 语言的测试工具就属于这种类型。

(4) 语言转换工具。计算机硬件的逐步更新换代推动着程序设计语言顺应时代潮流,与时俱进,不断向着更新、更好的方向迈进,同时,程序设计语言的推陈出新也为提高计算机的使用效率提供了良好的条件。然而,如何把一些常用软件或重要软件在无须重新编程的前提下就能在新的机器和语言环境下使用起来,成为一个十分关键的问题。为了避免重新编制程序所带来的人力和时间耗费,一种可行的解决途径就是将一种高级语言程序自动转换成另一种高级语言程序。当然,这种方法也适用于汇编语言程序向高级语言程序的转换。转换工具的开发要用到程序的词法和语法分析技术,最终生成的目标语言是另一种高级语言,而非编译程序所生成的汇编语言或机器语言。

1.2 编译程序的结构及编译过程

编译程序的工作,从输入源程序开始到输出目标代码为止的整个过程,是非常复杂的。图 1-2 给出了一个典型的编译程序的结构,从图中可以看出,一个完整的编译程序主要包含 7 个功能模块,分别是:词法分析器、语法分析器、语义分析与中间代码生成器、代码优化器、目标代码生成器,以及表格管理模块和错误处理模块。

编译器的组成

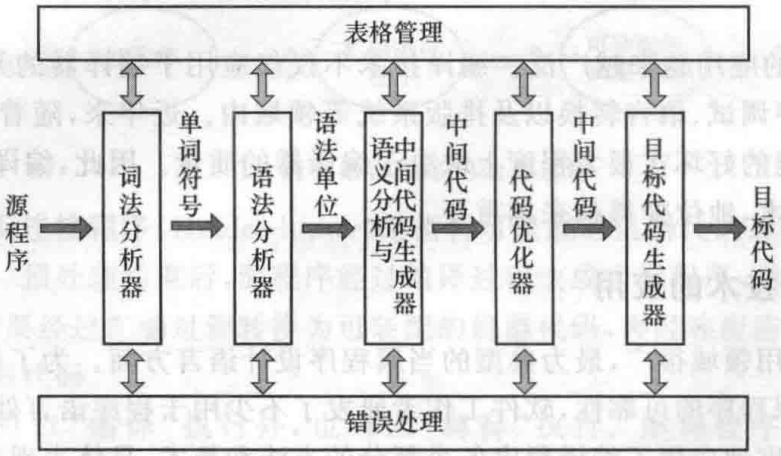


图 1-2 编译程序的结构

从工作过程角度分析，可将编译程序的工作过程划分为5个阶段：词法分析阶段、语法分析阶段、语义分析与中间代码生成阶段、代码优化阶段以及目标代码生成阶段。每个阶段分别与除表格管理和错误处理之外的编译模块相对应，完成不同的任务，并且各个阶段所做的操作在逻辑上是紧密相关的。编译过程尽管复杂，但与人们进行自然语言之间的翻译有许多相似之处，可以将二者进行比较，进而加深对编译过程的理解和认识。例如，把英文句子翻译为中文时，其过程与编译过程的相似情况如表1-1所示。


编译 VS 英译


动画：编译程序的处理过程

表1-1 英文翻译与编译过程的比较

序号	英文翻译	编译过程
1	识别句子中的单词	词法分析
2	分析句子的语法结构	语法分析
3	初步翻译句子的含义	语义分析与中间代码生成
4	译文修饰	代码优化
5	写出最终译文	目标代码生成

词法分析是实现编译器的基础，语法分析是实现编译器的关键，只有语法结构正确，才能进行正确的翻译。然而，上述编译过程的5个阶段只是一种典型的分法，实际上，并非所有的编译程序都分成这5个阶段。有些编译程序不提供优化功能，那么优化阶段就可以省去；有时为了提高编译速度，中间代码产生阶段也可以省去；甚至有些编译程序简化到在语法分析的同时直接产生目标代码。尽管如此，多数实用的编译程序其工作过程大都与上面所说的那5个阶段相一致，本书将按照这个顺序来讲述编译程序各个阶段涉及的基本理论、方法和技术。


编译过程

1.2.1 词法分析器

词法分析器，又称为扫描器(Scanner)，功能是输入源程序，进行词法分析，输出单词符号。

每种高级语言都规定了允许使用的字符集，常见的字符有大写字母A～Z、小写字母a～z、数字0～9、＋、－、＊、/等。高级语言的单词符号(简称为单词)都由定义在字符集上的符号构成，它们都是语言中有意义的最小单位。单词一般分为5种，分别是：保留字、标识符、常数、运算符和界符。

例如，对于如下的Pascal程序片段：

```
var x, y, z : real;
x := y + z * 60;
```

词法分析器将从左到右扫描输入的字符流，识别出一个个单词符号，并输出其内部表示形式，如表1-2所示。