普通高等教育“十一五”国家级规划教材

21世纪计算机科学与技术实践型教程

孙悦红 编著

编译原理及实现
（第2版）

丛书主编 陈明

本书第1版荣获
“北京高等教育精品教材”奖项

清华大学出版社

普通高等教育“十一五”国家级规划教材

孙悦红 编著

编译原理及实现
（第2版）

清华大学出版社
北京

21世纪计算机科学与技术实践型教程
丛书主编 陈明

内 容 简 介

本书以通俗易懂的语言介绍编译原理的理论和常用的方法与技术，并着重介绍各种编译方法的实现途径。全书共分10章，包括形式语言基础、词法分析、语法分析、语义分析及代码生成、符号表管理、运行时的存储分配，以及代码优化等。考虑目前学计算机专业的学生对C语言比较了解，本书以C语言为雏形设计了一种TEST语言，并在介绍全书内容时，用TEST语言进行分析与实现，使编译原理的抽象性通过TEST语言编译器的实现而具体化，从而使读者轻松掌握编译原理。

本书理论与实践并重，内容深入浅出，便于自学。每章后都提供了适量的习题。

本书可作为高等学校计算机专业的教材，也可供从事计算机应用和开发的人员使用。本书还配有教学辅助课件及书中所有程序示例，需要者可与作者(sun_yh@tom.com)联系。

图书在版编目(CIP)数据

编译原理及实现/孙悦红编著．--2版.--北京：清华大学出版社，2011.11
(21世纪计算机科学与技术实践型教程)
ISBN 978-7-302-26584-9

Ⅰ.①编… Ⅱ.①孙… Ⅲ.①编译程序－程序设计－高等学校－教材 Ⅳ.①TP314

中国版本图书馆CIP数据核字(2011)第175407号

责任编辑：谢　琛　战晓雷
责任校对：白　蕾
责任印制：何　芊
出版发行：清华大学出版社　　**地　　址**：北京清华大学学研大厦A座
http://www.tup.com.cn　　**邮　　编**：100084
社　总　机：010-62770175　　**邮　购**：010-62786544
投稿与读者服务：010-62795954，jsjjc@tup.tsinghua.edu.cn
质　量　反　馈：010-62772015，zhiliang@tup.tsinghua.edu.cn
印 装 者：北京鑫海金澳胶印有限公司
经　　销：全国新华书店
开　　本：185×260　**印　张**：17　**字　数**：402千字
版　　次：2011年11月第2版　　**印　　次**：2011年11月第1次印刷
印　　数：1～3000
定　　价：28.00元

产品编号：043044-01

21世纪计算机科学与技术实践型教程

编辑委员会

21世纪计算机科学与技术实践型教程

序

21世纪影响世界的三大关键技术：以计算机和网络为代表的信息技术；以基因工程为代表的生命科学和生物技术；以纳米技术为代表的新型材料技术。信息技术居三大关键技术之首。国民经济的发展采取信息化带动现代化的方针，要求在所有领域中迅速推广信息技术，导致需要大量的计算机科学与技术领域的优秀人才。

计算机科学与技术的广泛应用是计算机学科发展的原动力，计算机科学是一门应用科学。因此，计算机学科的优秀人才不仅应具有坚实的科学理论基础，而且更重要的是能将理论与实践相结合，并具有解决实际问题的能力。培养计算机科学与技术的优秀人才是社会的需要、国民经济发展的需要。

制定科学的教学计划对于培养计算机科学与技术人才十分重要，而教材的选择是实施教学计划的一个重要组成部分，《21世纪计算机科学与技术实践型教程》主要考虑了下述两方面。

一方面，高等学校的计算机科学与技术专业的学生，在学习了基本的必修课和部分选修课程之后，立刻进行计算机应用系统的软件和硬件开发与应用尚存在一些困难，而《21世纪计算机科学与技术实践型教程》就是为了填补这部分空白。将理论与实际联系起来，使学生不仅学会了计算机科学理论，而且也学会应用这些理论解决实际问题。

另一方面，计算机科学与技术专业的课程内容需要经过实践练习，才能深刻理解和掌握。因此，本套教材增强了实践性、应用性和可理解性，并在体例上做了改进——使用案例说明。

实践型教学占有重要的位置，不仅体现了理论和实践紧密结合的学科特征，而且对于提高学生的综合素质，培养学生的创新精神与实践能力有特殊的作用。因此，研究和撰写实践型教材是必需的，也是十分重要的任务。优秀的教材是保证高水平教学的重要因素，选择水平高、内容新、实践性强的教材可以促进课堂教学质量的快速提升。在教学中，应用实践型教材可以增强学生的认知能力、创新能力、实践能力以及团队协作和交流表达能力。

实践型教材应由教学经验丰富、实际应用经验丰富的教师撰写。此系列教材的作者不但从事多年的计算机教学，而且参加并完成了多项计算机类的科研项目，他们把积累的经验、知识、智慧、素质融合于教材中，奉献给计算机科学与技术的教学。

我们在组织本系列教材过程中，虽然经过了详细的思考和讨论，但毕竟是初步的尝试，不完善甚至缺陷不可避免，敬请读者指正。

本系列教材主编　陈明

2005年1月于北京

前　言

编译原理是高等学校计算机专业的必修专业课之一，是一门理论与实践并重的课程。编译原理介绍程序设计语言翻译的原理、技术及实现，对引导学生进行科学思维、提高学生解决实际问题的能力有重要作用。

在我国高等教育逐步实现大众化后，越来越多的高等学校将会面向国民经济发展的第一线，为行业、企业培养各类高级应用型专门人才。而受我国传统历史文化思想的影响，重理论、轻实践的观念在高教界仍较普遍，使我们培养的很多人才不适应社会需求，造成毕业生的结构性就业困难，这也迫使很多高等学校走向应用型教育，培养应用型人才。目前国内的编译原理教材大多偏重于理论，对实现技术介绍得较少，使学习者感到抽象、难以理解；而且教材篇幅厚重，由于授课时数的限制，以及学生接受能力的差异，教科书的内容往往不能充分发挥作用。根据这种现状，我们编写了本书，目的在于加强对学生应用能力的培养，使学生不仅具备理论知识，更要具备应用能力，使所学能为所用，以适应新经济时代对人才的需要，满足就业要求。

本书以通俗易懂的语言介绍编译原理，包括词法分析、语法分析、语义分析及代码生成、符号表管理、运行时的存储分配、代码优化等，并着重介绍各种编译方法的实现途径。考虑到目前计算机专业的学生对 C 语言比较了解，书中以 C 语言为基础设计了一种 TEST 语言，建立该语言的词法、语法、语义文法规则，系统介绍编译过程的各个部分。包括词法分析、语法分析、语义分析及代码生成、符号表的建立及存储分配、错误处理都用具体的实例进行分析与实现。并针对 TEST 语言中的典型语句，深入讲解如何具体用 C 语言编程实现词法分析、语法分析以及语义分析和代码生成，摆脱以往编译教材的抽象性以及理论与实际的脱节，使编译原理的抽象性通过 TEST 语言的编译器实现而具体化，从而使学习者轻松掌握编译原理。

全书共分 10 章，大约需要 70 课时，其中包括 20 课时的上机。第 1 章对编译过程、编译程序的逻辑结构以及编译程序各组成部分的功能进行概述；第 2 章介绍文法和语言，它为后面各章的学习奠定了理论基础；第 3 章介绍词法分析程序的设计原理，包括适合手工设计和自动生成词法分析程序的方法，以及 TEST 语言的词法分析程序的具体编程实现；第 4 章、第 5 章分别介绍自顶向下和自底向上的语法分析方法，主要介绍递归下降分析法、LL(1)分析法以及 LR 分析法，同时介绍 TEST 语言的递归下降分析实现；第 6 章介绍语法制导翻译的概念以及属性翻译文法；第 7 章介绍符号表的组织与管理；第 8 章介绍存储组织与分配技术；第 9 章介绍语义分析及代码生成的概念和技术，并以 TEST 语

言为范例，实现语义分析并同时生成抽象机汇编目标代码；第 10 章主要介绍局部优化和循环优化常采用的方法。另外，附录中列出了 TEST 语言的文法规则、词法分析程序、语法分析程序、语义及代码生成程序以及 TEST 抽象机模拟器的完整程序。每章后都提供适量的习题，使学习者通过适量的练习掌握书中的内容。

本书是作者多年教学实践的汇集和提炼，同时也参考了许多国内外的参考书，第 2 版除了修改第 1 版的部分内容外，还增加了实例。在第 2 版的编写中，司慧琳、曹建、陈红倩参与了本书第 3、4、6、9 章的部分内容的编写和示例程序的设计与调试，陈谊参加了本书的内容编排和资料收集工作，并提出了许多宝贵意见。本书还配有相应的教学辅助课件，以及词法分析、语法分析和语义分析方法的演示程序（包括递归下降、LL(1)、LR 分析法和可在 DOS 环境下运行的 LEX 与 YACC），有需要者可与作者联系，E-mail 地址为：sun_yh@tom.com。

鉴于作者水平有限，书中难免有错误和不妥之处，恳请读者批评指正。

作　者

目　录

第1章 编译概述

程序设计语言之所以能由专用的机器语言发展到现今通用的多种高级语言，就是因为有了编译技术。编译技术是计算机专业人员必须具备的专业基础知识，它涉及程序设计语言、形式语言与自动机、计算机体系结构、数据结构、算法分析与设计、操作系统以及软件工程等各个方面。现在程序员大多数使用各种高级程序设计语言编写程序，而计算机只能识别用二进制数0和1表示的指令和数所构成的机器语言程序，用高级语言编写的程序不能直接在机器上运行，要想让它运行并得到预期的结果，必须将源程序转换成等价的目标程序，这个转换过程就是所谓的编译。

本章主要介绍程序设计语言编译程序的组成和结构以及编译程序的工作环境，以便读者对编译的基本概念和工作过程有所了解。

1.1 程序设计语言

在计算机发展的初期，人们直接使用机器语言编写程序。机器语言由二进制数字0和1表示的机器指令组成，很不直观，而且难写、难读、难记，容易出错，调试极不方便，程序员在程序设计及检查错误时需要花很大的精力；更由于不同类型的计算机使用不同的机器指令，程序员必须针对某种类型的机器编程，编写的程序不适于移植，因此限制了计算机的推广与使用。

为了便于记忆、阅读和检查，人们用较直观的符号来代替机器指令，进一步发展成为汇编语言。汇编语言采用比较直观且具有含义的指令助记符表示每条机器指令，同时为了方便编程，还提供了若干宏指令对应一组机器指令，从而完成一些特定的功能。但是汇编指令依赖于机器，对问题的描述处于低层次，没有高级语言中的条件、循环等控制结构，编程人员必须考虑寄存器和内存的分配，使用仍不方便，程序设计的效率依然很低。

为了解决这些问题，提高编程效率，人们又发展出了更接近自然语言的高级程序设计语言，如BASIC、C、Pascal语言等。这类语言完全摆脱了机器指令的约束，用它编写的程序接近自然语言和习惯上对算法的描述，故称为面向用户的语言或面向过程的语言。后来，又相继出现许多专门用于某个应用领域问题的专业语言。例如用于数据库操作的SQL(Structured Query Language，结构化查询语言)语言，这类语言称为面向问题的语言。

汇编语言和机器语言依赖于机器，称为低级语言；而面向用户的语言称为高级语言。高级语言与低级语言相比较具有以下优点：

（1）高级程序设计语言不依赖于具体的机器，对计算机了解较少的人也可以学习和使用，有良好的可移植性，在一种类型的机器上编写的程序不做很大改动就能在别的机器上运行；

（2）编写高级语言程序时，不用考虑具体的寄存器和内存的分配，不用知道如何实现将数据的外部形式转换成计算机内部形式，也不必了解机器的硬件；

（3）每条高级语言语句对应于多条汇编指令或机器指令，编程效率高；

（4）高级语言提供了丰富的数据结构和控制结构，提高了问题的表达能力，降低了程序的复杂性；

（5）高级语言接近于自然语言，编程更加容易，编写出的程序有良好的可读性，便于交流和维护。

尽管高级语言有这么多优点，但是使用高级语言编写的程序是不能立即在计算机上执行的，它必须经过翻译程序翻译成机器语言程序，计算机才能执行。这种翻译程序就称为编译程序。虽然汇编语言不是高级语言，但是也不能在计算机上直接执行，仍需要翻译程序将汇编语言编制的程序翻译成机器语言程序，这种翻译程序称为汇编程序。对高级语言来说，其编译程序再加上一些相应的支持用户程序运行的子程序就构成了该语言的编译系统。编译系统是计算机的重要组成部分。本书主要介绍构造编译系统的原理、技术及其实现方法。

1.2 翻译程序

翻译程序扫描所输入的源程序，然后将源程序转换成目标程序。翻译程序的源程序分高级语言源程序和汇编语言源程序两种。

（1）如果要翻译的源程序是汇编语言编写的，而目标语言是机器语言，则翻译程序称为汇编程序；

（2）如果要翻译的源程序是用高级语言编制的，其翻译后的目标程序是某种具体机器的机器语言或汇编语言，那么这种翻译程序称为编译程序，而实现源程序到目标程序的转换所花费的时间叫做编译时间。

在高级语言程序的编译和运行过程中，源程序和数据是在不同时间处理的。源程序的处理在编译阶段进行，而数据则在程序的运行阶段处理。有的编译程序的目标程序是机器语言，则源程序从编译到被执行的过程如图 1-1 所示。

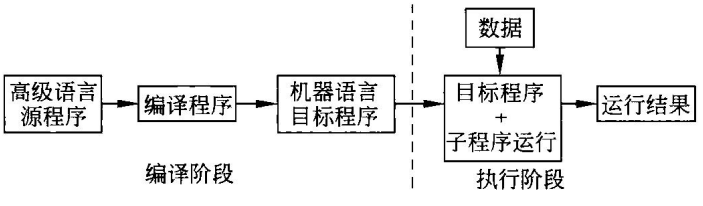

图 1-1 生成机器语言目标程序的编译方式

如果编译程序翻译得到的目标程序是汇编语言程序，则还要经过汇编程序翻译成机器语言程序，这种编译方式的源程序从编译到被执行的过程如图 1-2 所示。

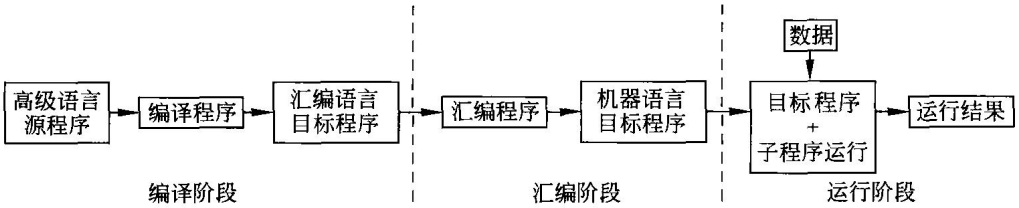

图 1-2 生成汇编语言目标程序的编译方式

还有一种高级语言翻译程序，从源程序的编译到执行只有一个阶段——解释执行阶段，它同时处理源程序和数据，按源程序中语句的动态顺序，逐句地进行分析解释，并立即予以执行，这种翻译程序称为解释程序，运行过程如图 1-3 所示。最常见的高级语言 BASIC 就是在解释环境下运行的。在解释方式下，最终并不生成目标程序，这是编译方式与解释方式的根本区别。解释方式很适合于程序调试，易于查错，在程序执行中可以修改程序，但与编译方式相比，执行效率太低。现在有些语言的集成开发环境提供了两种方式，如 Visual Basic，调试期间可解释执行源程序，而调好的程序可以编译生成目标程序。

图 1-3 高级语言的解释方式

本书虽然主要介绍编译技术及其实现，但同样的技术也适合于解释程序。掌握了编译技术，就不难设计和实现解释程序，而汇编程序的设计与实现和编译程序相比则更为简单。

1.3 编译程序的组成

为了将高级语言程序翻译成目标程序，编译程序首先必须对高级语言源程序进行分析，然后产生目标程序。因此，编译程序分成前后两个阶段：分析阶段和综合阶段。分析阶段根据源语言的定义，分析源程序的结构，检查源程序是否符合语言的规定，确定源程序所表示的对象和规定的操作，并以某种中间代码形式表示出来。词法分析、语法分析、语义分析和中间代码生成都属于分析阶段。综合阶段根据分析结果构造所要求的目标代码程序，包括代码优化和目标代码生成。为了记录分析过程中识别出的标识符及有关信息，还需要使用符号表。如果在编译过程中发现源程序有错误，不仅要报告错误，还要进行错误处理，使编译能继续进行。基本的编译程序模型见图 1-4。

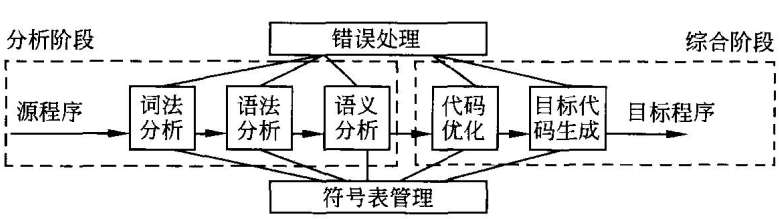

图 1-4 典型的编译程序模型