高等学校规划教材　计算机系列（工程应用型）

总主编　胡学钢

编译原理

BIANYI YUANLI

主编　王一宾　陈义仁

北京师范大学出版集团
BEIJING NORMAL UNIVERSITY PUBLISHING GROUP
安徽大学出版社

编译原理

总 主 编　胡学钢

主　　编　王一宾　陈义仁

编写人员　（以姓氏笔画为序）

王一宾　安庆师范学院

刘义红　淮南师范学院

李　立　安庆广播电视大学

陈义仁　安庆师范学院

张春梅　铜陵学院

北京师范大学出版集团
BEIJING NORMAL UNIVERSITY PUBLISHING GROUP
安徽大学出版社

图书在版编目(CIP)数据

编译原理/王一宾，陈义仁主编. —合肥：安徽大学出版社，2013.12
高等学校规划教材. 计算机系列. 工程应用型 / 胡学钢总主编
ISBN 978-7-5664-0616-3

Ⅰ.①编… Ⅱ.①王… ②陈… Ⅲ. ①编译程序－程序设计－高等职业教育－教材
Ⅳ. ①TP314

中国版本图书馆 CIP 数据核字(2013)第 272337 号

编译原理

胡学钢 总主编
王一宾 陈义仁 主 编

出版发行：北京师范大学出版集团
安 徽 大 学 出 版 社
（安徽省合肥市肥西路 3 号 邮编 230039）
www. bnupg. com. cn
www. ahupress. com. cn
印 刷：安徽省人民印刷有限公司
经 销：全国新华书店
开 本：184mm×260mm
印 张：15.75
字 数：383 千字
版 次：2014 年 1 月第 1 版
印 次：2014 年 1 月第 1 次印刷
定 价：32.00 元
ISBN 978-7-5664-0616-3

策划编辑：李 梅 蒋 芳
责任编辑：蒋 芳
责任校对：程中业
装帧设计：李 军
美术编辑：李 军
责任印制：赵明炎

编写说明

计算机科学与技术的迅速发展，促进了许多相关学科领域以及应用分支的发展，同时也带动了各种技术和方法、系统与环境、产品以及思维方式等的发展，由此而进一步激发了对各种不同类型人才的的需求。按照教育部计算机科学与技术专业教学指导委员会的研究报告来分，学校培养的人才类型可以分为科学型、工程型和应用型三类，其中科学型人才重在基础理论、技术和方法等的创新；工程型人才以开发实现预定功能要求的系统为主要目标；应用型人才以系统集成为主要途径实现特定功能的需求。

虽然这些不同类型人才的培养有许多共同之处，但是因不同类型人才的就业岗位所需要的责任意识、专业知识能力与素质、人文素养、治学态度、国际化程度等方面存在一定的差异，因而培养目标、培养模式等方面也存在不同。对大多数高校来说，很难兼顾各类人才的培养。因此，合理定位培养目标是确保教学目标和人才培养质量的关键。

由于当前社会领域从事工程开发和应用的岗位数量远远超过从事科学人才的数量，结合当前绝大多数高校的办学现状，安徽省高等学校计算机教育研究会在和多所高校专业负责人以及来自企业的专家反复研究和论证的基础上，确定了以培养工程应用型人才为主的安徽省高等学校计算机类专业的培养目标，并组织研讨组共同探索相关问题，共同建设相关教学资源，共享研究和建设成果，为全面推动安徽省高等学校计算机教育教学水平做出积极的贡献。北京师范大学出版集团安徽大学出版社积极支持安徽省高等学校计算机教育研究会的工作，成立了编委会，组织策划并出版了全套工程应用型计算机系列教材。

为了做好教材的出版工作，编委会在许多方面都采取了积极的措施：

编委会组成的多元化：编委会不仅有来自高校的教育领域的资深教师和专家，而且还有从事工程开发、应用技术的资深专家，从而为教材内容的重组提供更为有力的支持。

教学资源建设的针对性：教材以及教学资源建设的目标就是要突出体现“学以致用”的原则，减少“学不好，用不上”的空泛内容，增加其应用案例，尤其是增设涵盖更多知识点和应用能力的系统性、综合性的案例，以培养学生系统解决问题的能力，进而激发其学习兴趣。

建设过程的规范性：编委会对整体的框架建设、对每本教材和资源的建设都采取汇报、交流和研讨的方式，以听取多方意见和建议；每本书的编写组也都进行反复的讨论和修订，努力提高教材和教学资源的质量。

如果我们的工作能对安徽省高等学校计算机类专业人才的培养做出贡献，那将是我们的荣幸。真诚欢迎有共同志向的高校、企业专家提出宝贵意见和建议，更期待你们参与我们的工作。

胡学钢

2013 年 8 月 10 日于合肥

编委会名单

前　言

“编译原理”是计算机专业的一门核心课程，在计算机本科教学中占有十分重要的地位。编译原理课程具有很强的理论性与实践性，读者学习起来普遍感到内容抽象、不易理解。为此，本书采取由浅入深、循序渐进的方式介绍编译原理的基本概念和实现方法。在内容的组织上，本书将编译的基本理论与具体的实现技术有机地结合起来，既注重理论的完整性，化繁为简，又将理论融于具体的实例中，化难为易，以达到准确、清晰地阐述相关概念和原理的目的。除了各章节对理论阐述的条理性之外，书中给出的例子也具有实用性与连贯性，使读者对编译的各个阶段能有一个全面、直观的认识，从而透彻地领悟编译原理的精髓。本书采用的算法全部用C语言描述。

本书共分9章：第1章简要介绍了编译程序的基本概念、工作过程和逻辑结构。第2章介绍了文法和语言的形式化定义以及Chomsky文法分类等相关内容。第3章主要介绍了词法分析器的设计原理，以及正规表达式与有限自动机的相关内容。第4章主要介绍自上而下的语法分析方法，首先分析自上而下分析方法的一般思想及其所遇到的问题，然后介绍这些问题的解决方案，最后介绍两种不带回溯的自上而下语法分析算法：递归下降分析法和LL(1)分析法。第5章主要介绍自下而上的语法分析方法，首先介绍自下而上语法分析方法的一般思想及其核心问题，然后根据其核心问题的不同解决方案，介绍两种自下而上语法分析算法：算符优先分析法和LR分析法。第6章介绍了语法制导翻译与语义分析及中间代码生成的有关内容，给出了如何在语法分析的同时进行语义加工并产生出中间代码的方法。第7章主要介绍符号表的作用与内容、符号表的组织与管理等相关内容。第8章介绍了代码优化的有关内容，主要涉及基本块内的局部优化和循环优化。第9章讨论目标代码生成的有关内容，讲述了如何由中间代码产生出最终的目标代码。

为了便于读者正确理解有关概念，各章还配有一定数量的习题。这些习题大多选自本科生和研究生的考试试题，也包括作者结合多年教学实践经验设计出来的典型范例，力求使读者抓住重点、突破难点，进一步全面、深入地巩固所学知识。

本书第1章和第8章由王一宾编写，第2章和第3章由李立编写，第4章、第5章、第7章和第9章由陈义仁和张春梅编写，第6章由刘义红编写，最后由王一宾统稿。由于作者水平有限，书中难免存在一些缺点和错误，敬请读者批评指正。

编者

2013年5月

内容简介

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。教材主要内容包括：编译程序概论、文法和语言、词法分析与有限自动机、自上而下语法分析方法、自下而上语法分析方法、属性文法与语法制导翻译、语义分析与中间代码产生、符号表、代码优化、目标代码生成等。本书主要特色是突出基础知识和基本理论，强调工程实践与应用。书中配有丰富的例题和习题，相关章节后配有相应的实验项目。通过本教程的学习和相关实验的操作，能够培养和提高掌握编译程序的基本理论和设计原理，以及应用相关算法解决实际问题的能力。

本书可作为高等院校计算机等相关专业的本科或高职高专以及各类培训班的教材，也可作为教师、研究生、软件工程技术人员的参考书。

目　录

第 1 章　编译程序概论

本章目标

- 解释翻译程序和编译程序的概念
- 说明编译程序的工作过程
- 阐述编译程序的逻辑结构，给出其总体框架图
- 介绍编译技术的相关应用

计算机的诞生是科学发展史上的一个里程碑。经过近八十年的发展，计算机已经改变了人类生活、工作的各个方面，成为人类不可缺少的工具。计算机之所以能够如此广泛地被应用，应当归功于高级程序设计语言。没有高级语言，计算机的推广应用是难以实现的；而没有编译程序，高级语言就无法使用。计算机语言之所以能由最初单一的机器语言发展到现今数千种高级语言，就是因为有了编译程序。编译理论与技术也是计算机科学中发展得最迅速、最成熟的一个分支，它集中体现了计算机发展的成果与精华。本章主要介绍编译程序的概念、工作过程和逻辑结构，并介绍编译技术和理论的相关应用。

1.1　编程语言与翻译系统

1.1.1　程序设计语言

众所周知，自然语言是人类传递信息、交流思想和感情的工具，程序设计语言则是人与计算机联系的工具。人正是通过程序设计语言指挥计算机按照人的意志进行运算和操作、显示信息和输出运算结果的。

程序设计语言的发展至今已经历了 4 代，3 个阶段，如图 1-1 所示。

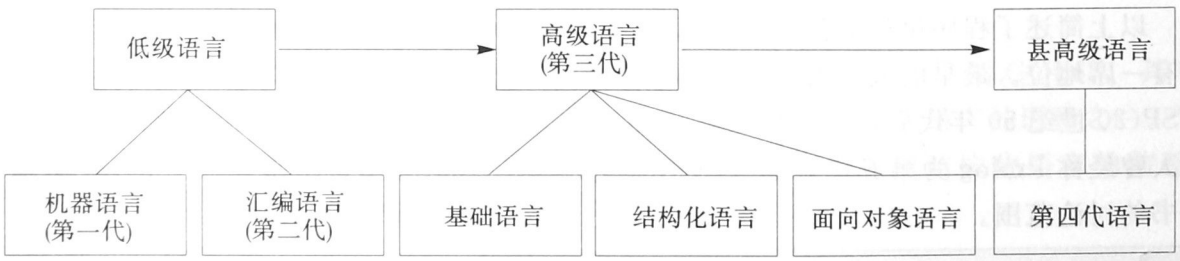

图 1-1　编程语言的发展和分类

1. 低级语言

低级语言包括第一代机器语言和第二代汇编语言。这两代语言依赖于机器的结构，其指令系统随机器而异，难学难用。该类语言，不仅编码效率低，容易出错，而且维护困难。它们在系统软件开发和计算机控制智能处理等方面使用较为广泛。

2. 高级语言

1956 年，第一个高级语言 FORTRAN 在美国诞生，成为最早的第三代程序设计语言(3GL)。有人统计，在同等条件下，一人一天生产的高级语言程序行数大致等于汇编语言程序的行数，但对于同一个问题，用高级语言书写的程序可比用汇编写出的程序缩短 3～7 倍，所以两者的生产率也要相差数倍。20 世纪 60 年代末期，高级语言编码在科学计算领域已达到 98%，但对于效率要求较高的实时系统，其比例还不到 10%。在系统软件领域，当时还是汇编语言的独家天下。随着现代语言特别是 C 语言和 Ada 语言的出现，汇编语言在上述两个领域的传统优势受到了严重的挑战。时至今日，除去使用高级语言确实不能满足软件需要的个别情况外，已很少使用汇编语言编码。

从 FORTRAN 到 Ada，大多数常用的第三代语言都是面向过程的。随着面向对象程序设计的推广，在 20 世纪八九十年代，一批著名的常用语言扩展了面向对象的功能，出现了 C＋＋、Object Pascal、Ada95 等面向对象的第三代语言，以及全新的面向对象高级语言，如 Java、Eiffel 等。它们配合 OO 软件的开发，使编码语言有了更多的选择。

3. 高级语言的发展方向

50 多年来，高级语言的面貌发生了巨大变化，反映了人们对程序设计的认识从浅到深的过程。但是从根本上说，上述的通用语言仍都是“过程化语言”。编码的时候，要详细描述问题求解的过程，告诉计算机每一步应该“怎样做”。为了把程序员从繁重的编码中解放出来，还须寻求进一步提高编码效率的新语言，这就是甚高级语言(VHLL)或第四代语言(4GL)产生的背景。

对于 4GL 语言，迄今仍没有统一的定义。一种意见认为，3GL 是过程化的语言，目的在于高效地实现各种算法；4GL 则是非过程化的语言，目的在于直接地实现各类应用系统。前者面向过程，需要描述“怎样做”；后者面向应用，只需说明“做什么”。曾多年担任 IFIP 数据库专家组主席的 G. M. Nijssen 教授则主张，语言的划代应该以数据结构为标准。他认为，前三代语言的基础着重算法的描述，一次仅处理一个记录或数据元素，不支持对大量共享数据的处理。第四代和第五代语言应该以数据或知识为基础，以对集合的处理代替对于单个记录或元素的处理，能支持对大型数据库进行高效处理的机制。Nijssen 还认为，前三代语言是工业时代的产物，4GL 则标志了信息时代的开始。

以上简述了程序设计语言的发展，下面再补充一点。在高级语言的应用中，人工智能也占有一席地位。最早的人工智能语言可追溯到 20 世纪 50 年代的 IPL 语言。随后出现的 LISP(20 世纪 50 年代后期)和 Prolog(20 世纪 70 年代前期)等语言，都具有很高的知名度。有人曾经称 Prolog 为第五代语言，但它主要是为新一代人工智能计算机设计的，故不属于本书的讨论范围。

关键概念：程序程序设计语言的发展经历了“低级语言——高级语言——甚高级语言”这样的 3 个阶段。

1.1.2 常用的高级语言

高级语言种类繁多，总数已不下千种。一般情况下，可以把它们分为基础语言、结构化语言和面向对象语言 3 大类。

1. 基础语言

FORTRAN、COBOL 和 BASIC 是这类语言的代表。之所以称它们为基础语言，是因为它们都有较长的使用历史，在国内外流传甚广，有大量已开发的软件，今天仍拥有广大的用户。这些语言创始于 20 世纪五六十年代，部分性能已趋老化，但随着版本的几次重大改进，除旧更新，至今仍被广泛使用。

(1)FORTRAN

FORTRAN 是使用最早的高级语言。从 1956 年到现在，经过 40 多年的实践检验，始终保持着科学计算重要语言的地位。在我国，20 世纪 70 年代流行较广的有 FORTRAN－II 和 FORTRAN66 等版本，以后又引入了 FORTRAN77、FORTRAN90 等新版本。从 FORTRAN77 起，就在支持结构化控制结构与字符串处理等方面做了较大的改进，但数据类型仍欠丰富，对复杂的数据结构也缺乏支持。

(2)COBOL

COBOL 是商业数据处理中应用甚广的高级语言。它发明于 20 世纪 50 年代末期，后来又发表了 1972、1978 等国际标准文本。它广泛支持与事务数据处理有关的各种过程技术，使用近于自然语言，虽然程序不够紧凑，但易于理解，从而受到企业、事业从业人员的欢迎。在美、日等国，它与 FORTRAN 并列为两大基础语言，长期拥有大量的用户。其主要不足是计算功能弱，编译速度也不够快。

(3)BASIC

BASIC 原是 20 世纪 60 年代初期为适应分时系统而研制的一种交互式语言，可用于一般数值计算与事务处理。由于它简单易懂，具有交互功能，成为许多初学者学习程序设计的入门语言，对计算机语言的普及起过巨大作用。它的早期版本不支持结构程序设计，不区分数据类型，加上解释执行的速度过慢，不适用于较大软件等原因，使它在 20 世纪 70 年代曾一度衰落。随着微型计算机的兴起，BASIC 又成为微机上配置最广的高级语言，出现了上百种不同的版本。这些版本各行其是，难于移植，一些被称为“street BASIC”，认为不能登大雅之堂。1985 年，BASIC 语言的创始人在美国国家 BASIC 标准(1984)的基础上，研制公布了取名“True BASIC”的新版本。“True BASIC”保留了简单易学的特点，完全支持结构程序设计的概念，加强了绘图、窗口、矩阵运算等功能，给这一大众的语言注入了新的活力，使之在适应各种较大的应用课题上更进一步。

属于这类的常用语言还有 ALGOL 语言，包括 ALGOL60 与 ALGOL68。其中 ALGOL60 是我国引进最早的高级语言，也是 20 世纪 70 年代国内流行最广的语种之一。它对 20 世纪 70 年代出现的 Pascal 语言有强烈的影响，被认为是现代结构化语言的前驱。ALGOL68 由于过于庞大，在公布后不久就夭折了。

2. 结构化语言

20 世纪 70 年代以来，在结构化程序设计影响下，先后出现了一批常用的结构化语言，Pascal、C 等语言就是其中著名的代表。

(1)Pascal

Pascal 是第一个系统地体现结构化程序概念的现代高级语言，是在 1970 由 Wirth 首先开发的。最初的目标，是把它用作结构化程序设计的教学工具。由于它模块清晰，控制结构完备，有丰富的数据类型和数据结构，加上语言表达力强，容易移植，不仅被国内许多高等学

校采用为教学语言，而且在科学计算、数据处理以及系统软件开发中都有较广的应用。1983年，美国正式公布了ANSI－Pascal标准。随后便在从微型到大型的各类计算机上实现，并出现了一些有特色的微机Pascal语言(例如Turbo－Pascal)。但由于Pascal不是为支持大型软件开发设计的，现在软件开发中已很少使用。

(2)C语言

C语言是1973年由美国Bell实验室的Ritchie研制成功的。它起初是作为UNIX操作系统的主要语言开发的，现已成功地移植到多种微型与小型计算机上，成为独立于UNIX操作系统的通用程序设计语言。它除了具有结构化语言的公共特征，如表达简洁，控制结构与数据结构完备，有丰富的运算符和数据类型外，尤以移植能力强、编译质量高等特点，吸引了人们的注意。用C编译程序产生的目标程序，其质量可以与汇编语言产生的机器程序媲美。因为它既具有汇编语言的高效率，又不像汇编语言那样只能束缚在某种处理机上运行，所以有人称之为“可移植的汇编语言”。C语言的这些特点，使它不仅能写出效率高的应用软件，也适用于编写操作系统、编译程序等系统软件。著名的UNIX操作系统，就有90%以上的代码是用C语言编写的。

(3)Ada语言

Ada语言是迄今为止最完善的面向过程的现代语言，也是集FORTRAN以来各种语言之大成的语言。其开发计划始于1975年，由美国国防部直接领导。1983年公布了Ada的美国国家标准和军用标准。1985至1986年，法、英等国和我国都相继宣布Ada为军用的通用语言。

Ada的目标，是适用于一切“嵌入式(Embed)计算机系统”。它具有许多经典的语言特征，也包含许多新的特征。为了满足实时应用，它支持并发处理与过程间的通信，支持在异常处理中实施中断，并能支持通常只能由汇编语言实现的低级操作，如位操作、字节操作等。在语言的表达与结构上，它又具有高级语言的特点，远比汇编语言容易开发和维护。

Ada还是一个充分体现软件工程思想的语言。它既是编码语言，又可用作设计表达工具。Ada提供的多种程序单元(包括子程序、程序包、任务与类属)、与实现相分离的规格说明，以及分别编译等措施，支持对大型软件的开发，也为采用现代开发技术开发提供了便利。Ada还明确规定了对支持环境的要求，能为软件的开发与维护提供全生存周期的支持。

3. 面向对象语言

(1)C++语言

C++是从C语言进化而来，是C语言的超集。1980年，为了满足管理程序的复杂性的需要，贝尔实验室的Bjarne Stroustrup开始对C进行改进和扩充，1983年正式取名为C++。经过3次修订后，于1994年制定了ANSI C++标准的草案。以后又经过不断完善，成为目前的C++。C++在程序结构的本质上与C语言是一致的，都是用函数驱动机制实现。因此C++既可以进行过程化程序设计，也可以进行面向对象程序设计。

(2)Java语言

Java语言是当今流行的新兴网络编程语言，它的面向对象、跨平台、分布应用等特点给编程人员带来了一种崭新的概念，使WWW从最初的单纯提供静态信息发展到现在的提供各种各样的动态服务。Java不仅能够编写小应用程序实现嵌入网页的声音和动画功能，而且能够应用于独立的大中型应用程序，其强大的网络功能能够把整个Internet作为一个统

一的运行平台，极大地拓展了传统单机或 Client/Server 模式下应用程序的内涵和外延。自从 1995 年正式问世以来，Java 已经逐步从一种单纯的计算机高级编程语言发展为一种重要的 Internet 平台，并进而引发、带动了 Java 产业的发展和壮大，成为当今计算机业界不可忽视的力量和重要的发展方向。

近 20 年来，国外不少软件公司在 4GL 的影响下，推出了一些快速开发的编程工具，其中流行较广的有 Delphi、Power Builder、Visual Basic、Visual Foxpro 和 Javascript 等。

关键概念：常用的高级语言可以分为基础语言、结构化语言和面向对象语言 3 大类。

1.1.3　编译程序的概念

使用过现代计算机的人都知道，多数用户是应用高级语言来实现他们所需要的功能的。现代计算机系统一般都含有不止一个的高级语言编译程序，对有些高级语言甚至配置了几个不同性能的编译程序，供用户按不同需要进行选择。高级语言编译程序是计算机系统软件最重要的组成部分之一，也是用户最直接关心的工具之一。

在计算机上执行一个高级语言程序一般要分为两步：第一步，用一个编译程序把高级语言翻译成机器语言程序；第二步，运行所得的机器语言程序求得计算结果。

通常所说的翻译程序是指这样的一个程序，它能够把某一种语言程序（称为“源语言程序”）转换成另一种语言程序（称为“目标语言程序”），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 、Ada 、Smalltalk 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”，这样的一个翻译程序就称为编译程序。

高级语言程序除了像上面所说的先编译后执行外，有时也可“解释”执行。一个源语言的解释程序是这样的程序，它以该语言写的源程序作为输入，但不产生目标程序，而是边解释边执行源程序本身。本书将不对解释程序作专门的讨论。实际上，许多编译程序的构造与实现技术同样适用于解释程序。

关键概念：编译程序实际上是一个源语言为高级语言而目标语言是低级语言的翻译程序。在计算机上执行高级语言编写的程序通常有编译和解释两种方式，它们的区别为是否生成目标代码。

1.2　编译程序的工作过程

编译程序的工作，从输入源程序开始到输出目标程序为止，整个过程是非常复杂的。但就其过程而言，它与人们进行自然语言之间的翻译有许多相近之处。当我们把一种文字翻译为另一种文字，例如把一段英文翻译为中文时，通常需经过下列步骤：

①识别出句子中的一个个单词；

②分析句子的语法结构；

③根据句子的含义进行初步翻译；