普通高等教育“十一五”国家级规划教材 计算机系列教材

编译原理

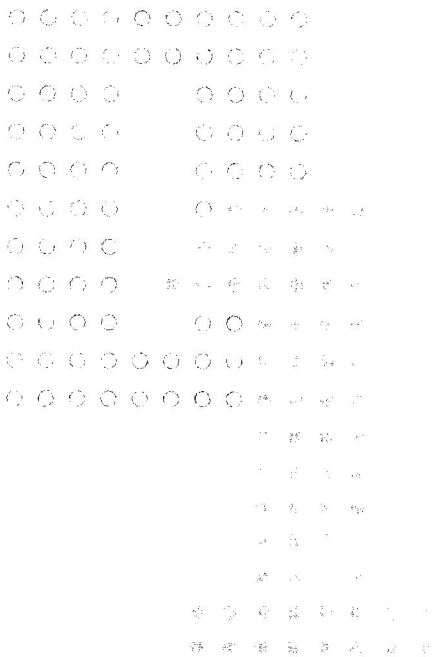
陈英 陈朔鹰 主编
计卫星 主审

清华大学出版社

普通高等教育“十一五”国家级规划教材　计算机系列教材

陈英　王贵珍　李侃　计卫星　陈朔鹰　编著

编译原理



清华大学出版社
北京

内 容 简 介

本书系统全面地介绍经典、广泛应用的高级程序设计语言编译程序的构造原理、实现技术、方法和工具。本书包含了现代编译程序设计的基础理论和技术，并在语义分析、代码优化，面向对象语言的编译及高级优化技术等方面反映了20世纪90年代后的一些重要研究成果，特别兼顾近年来编译原理及技术的发展和发生的一些重要变化，专辟“编译技术高级专题”予以介绍。本书的组织注重提炼精华、循序渐进、深入浅出，每章开头提炼了该章涉及的主要内容、要点和关键概念，全书精编、精选了近300道各种类型的习题和思考题，还提供了编译程序实现的具体实例，能够辅助读者更好地学习和掌握编译原理。

本书可以作为计算机学科类专业及相关专业本科和研究生编译原理的教科书，也可以作为软件技术人员的参考用书。

图书在版编目（CIP）数据

编译原理/陈英等编著. —北京：清华大学出版社，2009.4
（计算机系列教材）
ISBN 978-7-302-19744-7

Ⅰ. 编… Ⅱ. 陈… Ⅲ. 编译程序－程序设计 Ⅳ. TP314

中国版本图书馆CIP数据核字(2009)第039084号

责任编辑：谢 琛 薛 阳
责任校对：焦丽丽
责任印制：王秀菊

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：010-62776969，c-service@tup.tsinghua.edu.cn
质 量 反 馈：010-62772015，zhiliang@tup.tsinghua.edu.cn
印 刷 者：北京国马印刷厂
装 订 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**：22　　**字　　数**：536千字
版　　次：2009年4月第1版　　**印　　次**：2009年4月第1次印刷
印　　数：1～4000
定　　价：32.00元

《编译原理》前言

编译原理作为计算机学科的一门重要专业基础课，列入国际 ACM 教程和 IEEE 计算机学科的正式主干课程，并提高该课程内容的课时比重，这充分体现了其在计算机科学中的地位和作用。

编译程序是计算机系统软件的主要组成部分，是计算机科学中发展迅速、系统、成熟的一个分支，其基本原理和技术也适用于一般软件的设计和实现，而且在语言处理、软件工程、软件自动化、逆向软件工程、再造软件工程等诸多领域有着广泛的应用。

本书旨在介绍编译程序设计的基本原理、实现技术、方法和工具。本书的前驱版本系陈英教授主编，获得北京市 2008 年高校精品教材。在此基础上，规划、整合为“编译原理”课程的系列丛书，包括作为教材的本书及后续即将推出的《编译原理学习指导与习题解析》和《编译原理课程设计》。全书分为 11 章，第 1 章作为全书的开场白，介绍了编译程序有关的概念，编译过程、编译程序的结构与组织等要点。第 2 章作为后续各章的基础知识，也是学习编译原理应起码具备的理论基础，对形式语言与自动机理论作了基本的介绍。第 3 章以正则文法、正规式、有限自动机为工具，讨论了词法分析器的设计与实现。第 4 章和第 5 章对常规的语法分析方法，即自上而下和自下而上分析中的几种经典方法展开了较深入的讨论，并结合流行、实用、高效的 LR 分析方法，介绍了二义文法的分析应用，编译程序的出错处理。第 3 章和第 5 章还讨论了流行的词法分析和语法分析自动生成工具 Lex 及 Flex，YACC 及 Bison 的构造原理与应用，并以 ANSI_C 语言为例，给出了其 Lex 和 YACC 的描述。第 6 章涉及语义分析方法和属性翻译文法，中间语言，符号表及类型检查技术，流行的高级程序设计语言中典型语句的翻译。第 7 章介绍编译程序运行时环境的有关概念和存储组织与分配技术。第 8 章介绍中间代码级上的优化，展开讨论了优化的基本概念，优化涉及的控制流分析和数据流分析技术，以及中间代码上的局部优化和循环优化技术及实现。第 9 章简单介绍了代码生成的有关知识点及目标代码级可实施的窥孔优化技术。第 10 章以 PL/0 语言为源语言，提供了一个短小、精悍的编译程序实现的范例，以弥补编译程序从原理到工程实现的鸿沟。第 11 章反映了 20 世纪 90 年代后本领域的一些重要研究成果，如面向对象语言的翻译、GLR 分析。另外，高性能体系结构的发展与技术对编译技术提出新的挑战。本章针对主流并行处理器体系结构及与之相关的编译优化技术进行简要介绍，如并行优化技术、存储层次及其优化技术等。

纵观本书的组织，注重循序渐进，深入浅出，每章开头提炼了该章涉及的主要内容提要和关键概念，全书精编、精选了近 300 道各种类型的习题和思考题，还提供了编译程序实现的具体实例，辅助读者更好地学习和掌握编译原理。

本书是作者多年教学实践和科研工作的汇集、提炼和整理，特别是北京理工大学“编译原理”课程组老师们奉献了他们教学实践的汇集和积累。本书完成的责任编著和辅助编著的直接承担者是：陈英第1,3,4,5,6,9和11章；王贵珍第2,10,4和11章；李侃第6,7,11和2章；计卫星第8,9,11,1和5章；陈朔鹰第3,7和8章。本书参考了国内外一些专著、论文和资料，参考、借鉴了一些专家学者的研究成果，对所有这些前辈和同行的引导和帮助表示衷心的感谢。另外，本书过去多个不同版本通过数届学生和读者的使用，亦得到了他们许多宝贵的反馈意见和建议。本书完成过程中，得到了清华大学出版社的鼎力协助，尤其是编审谢琛高效的工作和非常专业的指导，作者在此一并深为致谢。

鉴于作者水平有限，本书稿虽经审慎校阅，仍难免存在疏误，敬请读者不吝赐教。

编　者

2009年1月于北京

FOREWORD

《编 译 原 理》 目录

第1章 编译引论

【本章导读提要】

本章内容作为了解、学习和掌握编译程序原理与技术的基础，主要涉及的内容与要点是：

- 什么是编译程序；编译程序的分类与表示。
- 源程序的运行及编译过程。
- 编译程序的组成结构和工作过程。
- 典型的编译程序的逻辑阶段划分，各阶段的主要功能和各阶段之间的逻辑关系。
- 编译程序的构造要素。
- 编译程序的实现方式。

【关键概念】

编译程序　解释程序　源程序的编译与执行　源语言　源程序　目标语言　目标程序　宿主机　宿主语言　遍　编译执行　解释执行

1.1 程序设计语言与编译程序

1.1.1 编译程序鸟瞰

学习编译程序的构造原理、方法和技术，需搞清编译程序的由来及定义，即何为编译程序，这亦是本书的研究对象。

众所皆知，一个计算机程序总是基于某种程序设计语言。半个多世纪以来，程序设计语言经历了由低级向高级的发展，从最初的机器语言、汇编语言，发展到较高级的程序设计语言，直至今天的第四代、第五代高级语言。高级程序设计语言的以人为本，面向自然语言表达，易学、易用、易理解、易修改等优势加速了程序设计语言的发展。程序设计语言的发展和应用，促进了计算机的普及使用，也大大提高了计算机的效率，增强了其功能，这在计算机科学发展史上是一个重要的里程碑。计算机的深入发展和应用普及除了计算机硬件本身发展迅速的因素外，与之相适应的更为重要的因素是计算机软件的飞速发展，多数计算机用户是通过应用程序设计语言这种更直接的方式来实现使用计算机的意图和目的。

但是就目前而言，计算机硬件自身根本不懂 BASIC，Pascal，C，C++，Ada 和 Java 等高级语言，用高级语言编写的程序计算机不能直接执行，因为计算机仅能识别的是机器语言。高级程序设计语言只是人和计算机交互的媒介。那么，如何使一个高级语言编写的程序能够在只认得机器语言的计算机上执行呢？这就需要像人们为了通信、交流的方便，建立各种语言的翻译一样，由从事计算机软件工作的人员搭一座桥梁，作为沟通计算机硬件与用户之间的渠道，这座桥梁即为“编译程序”，亦称“语言处理程序”。通过这样的程序翻译处理工

作，计算机才能执行高级语言编写的程序。编译程序所起的桥梁作用，可类比为两个不同语言的人借助翻译进行交流，不同之处在于编译程序是一个单向的翻译，编译程序的功能如图 1-1 所示。

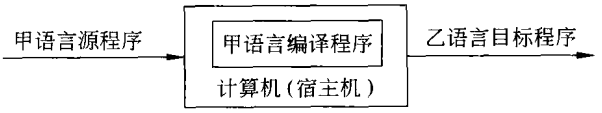


图 1-1　编译程序的功能

确切地讲，把用某一种程序设计语言写的源程序翻译成等价的另一种语言程序（目标程序）的程序，称之为编译程序（编译器 compiler）或翻译程序（翻译器 translator）。简单地说，编译程序是一个翻译程序，它是程序设计语言的支持工具或环境。术语“编译”的内涵是实现从源语言表示的算法向目标语言表示的算法的等价变换。

被编译的程序称为源程序（source program）。源语言是用来编写源程序的语言，一般是汇编语言或高级程序设计语言。源程序经过编译程序翻译后生成的程序称之为目标程序（target program）。目标程序可以用机器语言、汇编语言甚至高级语言或用户自定义的某类中间语言来描述。通常称运行编译程序的计算机为宿主机或目标机。编译程序实现的语言称为实现语言（或宿主语言）。例如，若将 A 语言的源程序翻译成 B 语言的程序，翻译的实现语言是 Y 语言，则称 A 语言是翻译的源语言，B 语言是目标语言，Y 语言是宿主语言。

1.1.2　源程序的执行

一个源程序编写后要投入运行，需要编译程序支持的执行过程分为两个阶段：编译阶段和运行阶段，如图 1-2 所示。编译阶段对整个源程序进行分析，翻译成等价的目标程序，然后在运行子程序的支持下在目标机上运行。运行子程序是为了支持目标的运行而开发的程序，例如有系统提供的标准函数及其他目标程序所调用的程序等。

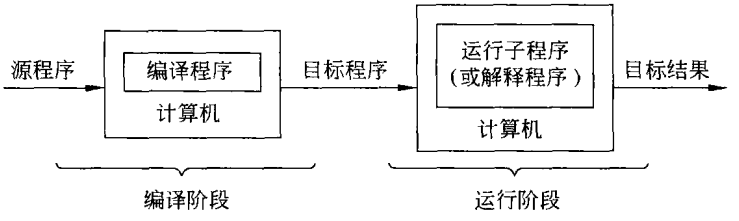


图 1-2　源程序的编译与运行

1.2　编译程序的表示与分类

1.2.1　T 型图

一个编译程序可以用三种语言来刻画，即源语言、目标语言和宿主语言（编译程序的实

现语言)，用T型图可以方便地对其进行表示。其中，T型图的左上角表示源语言，右上角表示目标语言，而其底部表示实现语言，如图1-3所示。

例如，对一个用Z语言实现的，从源语言X到目标语言Y的编译程序，可用如图1-4所示的T型图表示，此编译程序也可记做 C_Z^{XY}。

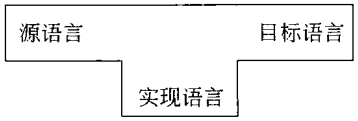


图1-3 一个编译程序的T型图表示

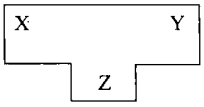


图1-4 编译程序 C_Z^{XY} 的T型图

用T型图表示交叉编译和编译程序的移植非常方便、清晰。所谓交叉编译是指，由于目标机B的指令系统与宿主机A的指令系统不同，要用运行在宿主机A上的某编译程序为另一台机器B生成目标代码。

例如，设机器B上已有源语言S的编译器 C_B^{SB}，现在要利用已经实现的语言S为另一个源语言L编写一个交叉编译器，并生成机器A的目标代码，即创建 C_S^{LA}。若 C_S^{LA} 通过 C_B^{SB} 来运行得到编译器 C_B^{LA}，则这就是一个运行于机器B上将源语言L翻译成机器A的目标代码的编译器。编译器 C_B^{LA} 的T型图可以用如图1-5所示的叠放在一起的T型图表示。

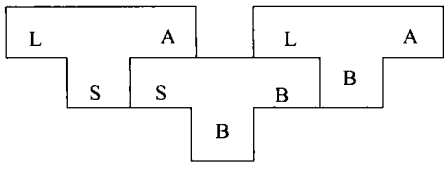


图1-5 编译器 C_B^{LA} 的T型图

叠放的T型图的结合规则需要满足以下条件：即中间那个T型图的两臂上的语言分别与左右两个T型图底部的语言相同，且对于左右两个T型图而言，其两个左端的语言必须相同，两个右端的语言必须相同。

1.2.2 编译程序的分类

如同各种事物的分类，基于不同的角度可以对编译程序进行不同的分类。

(1) 编译程序从源语言类型或实现机制不同角度一般可以分为汇编程序、解释程序和编译程序，甚至可以把宏汇编器、预处理器等也认为是编译程序的一类。

汇编程序(assembler)：汇编语言是计算机语言的符号形式。若源程序用汇编语言编写，经翻译生成的是机器语言表示的目标程序，该翻译程序称为“汇编程序”。通常，编译程序会生成汇编语言程序作为其目标语言，然后再由汇编程序将它翻译成目标代码。

编译程序：若源程序用高级语言编写，经翻译加工生成某种形式的目标程序，该翻译程序称为“编译程序”。

解释程序(interpreter)：接收某语言的源程序(或经翻译生成的中间代码程序)直接在机器上解释执行的一类翻译程序。

(2) 编译程序从对源程序执行途径的角度不同，可分为解释执行和编译执行的翻译程序。

解释执行：借助于解释程序完成，即按源程序语句运行时的动态结构，直接逐句地边分析、边翻译并执行。像自然语言翻译中的口译，随时进行翻译。解释执行的过程如

图 1-6 所示。解释执行的优点是易于查错，在程序执行过程中可以修改程序。

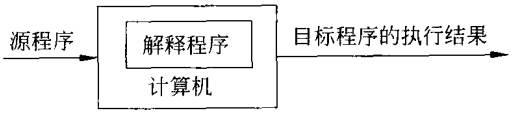


图 1-6 源程序的解释执行

编译执行：将源程序先翻译成一个等价的目标程序，然后再运行此目标程序，故编译执行分为编译阶段和运行阶段，如图 1-2 所示。

要注意源程序两种执行方式的区别，编译执行是由编译程序生成一个与源程序等价的目标程序，它可以完全取代源程序，目标程序可运行任意多次，不必依赖编译程序。正像自然语言翻译中的笔译，一次翻译可多次阅读。而解释执行不生成目标程序，对源程序的每次执行都伴随着重新翻译的工作，而且不能摆脱翻译程序。有些编译程序既支持解释执行又支持编译执行，在程序的开发和调试阶段用解释执行，一旦程序调试完毕，便采用编译执行。

(3) 从编译程序的用途、实现技术等侧重面，编译程序可以分为如下几类。

并行编译器：并行编译系统已成为现代高性能计算机系统中一个重要组成部分。它能够处理并行程序设计语言，可以针对并行体系结构进行程序优化；可以对向量机的向量语言处理和并行多处理机的并行语言处理，使串行程序向量化；可以对流水线、超长指令字、指令延迟槽等硬件结构的指令调度优化，针对分布存储器多处理机进行通信优化等。

优化型编译器：这类编译器产生高效率的机器代码，提供多级别、多层次的优化供用户选择。优化型编译器的代价是增加了编译程序的复杂性和编译时间。

交叉型编译程序(交叉编译器)：当一个编译程序在某种型号的目标机上运行，而生成在另一种型号目标机上运行的目标程序时，称该编译程序为交叉编译器。

诊断型编译器：此类编译器是为了帮助用户开发和调试程序，它能检查、发现源程序中的错误，并能在一定程度上校正一些错误，它适合于在程序开发的初始阶段使用。

可重定向型编译器：通常的编译程序都是为某个特定的程序设计语言或特定的目标机设计的，编译生成的目标程序只能在特定的目标机上运行。当使用同一种程序设计语言为另一种不同型号的目标机配置编译程序时，原有的编译程序不再适用而需重新设计。当然仅重写与机器有关的部分而与机器无关部分不必重写。可重定向型编译程序不用重写此编译程序中与机器无关部分就可以改变目标机的编译程序，它的可移植性好。

1.3 编译程序的结构与编译过程

1.3.1 编译程序的结构与编译过程

编译程序是比较复杂、庞大的系统软件。它所涉及的处理对象——源语言程序，从通用语言到计算机应用的各个领域的专用语言有成百上千种，它所涉及的处理结果——目标程序，其形式既可以是另一种程序设计语言或特定目标表示，又可以是从微型计算机到超大型计算机的某种机器语言，可见不同源语言需要不同的编译器。现在比较流行，使用比较广泛的一些编译器，如 Turbo 系列、Visual 系列等，已不仅仅是一个语言翻译工具，更是一个包

括编辑器、连接器、调试器等功能的庞大的集成开发环境。尽管编译程序的处理过程复杂，且不同的编译程序实现方法千差万别，构造原理各异，但任何编译程序要完成的基本任务都是类似的，其基本逻辑功能及必须完成的处理任务的分模块具有共同点。图 1-7 给出了编译程序总体结构的典型表示，也反映了编译程序的概貌与组成。

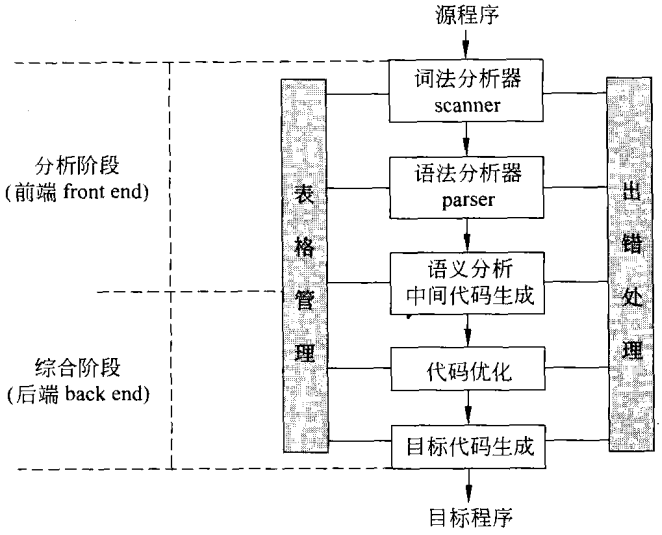


图 1-7　编译程序的总体结构

图 1-7 所示的编译程序总体结构图中，其中间位置的纵向 5 个矩形框表示编译程序工作过程的 5 个阶段或完成编译程序某阶段特定功能的模块，各模块间有密切的逻辑联系。图中两边的灰色矩形框是编译程序的辅助模块，可在编译的任何阶段被调用，辅助完成编译功能。

如图 1-7 所示，编译程序的工作过程是，从输入源程序开始到输出目标程序为止，经过词法分析、语法分析、语义分析与中间代码生成、代码优化及目标代码生成 5 个阶段，反映了一般编译器的动态编译过程。

1. 词法分析

词法分析(Lexical analysis)阶段的任务是对输入的符号串形式的源程序进行最初的加工处理。它依次扫描读入的源程序中的每个字符，识别出源程序中有独立意义的源语言单词，用某种特定的数据结构对它的属性予以表示和标注。词法分析实际上是一种线性分析，词法分析阶段工作依循的是源语言的词法规则。例如，有如下 C 语句：

```
a[index]=12 * 3;
```

经过词法分析识别出 9 个单词并输出每个单词的单词符号表示，如表 1-1 所示。

在表 1-1 中，“单词类型”和“单词值”表示单词符号，通常也称为属性字或记号(Token)。单词属性字的数据结构可据不同语言及编译程序实现方案来设计，但一般由单词类型标识及单词值两部分构成。通俗地讲，单词的属性字实际是单词机器内部表示的一种记号。