普通高等教育“十一五”国家级规划教材

21世纪大学本科计算机专业系列教材

康慕宁 任国霞 唐晶磊 编著

编译原理

http://www.tup.com.cn

- 根据教育部“高等学校计算机科学与技术专业规范”组织编写
- 与美国 ACM 和 IEEE *Computing Curricula 2005* 同步

清华大学出版社

普通高等教育“十一五”国家级规划教材

21世纪大学本科计算机专业系列教材

编译原理

康慕宁 任国霞 唐晶磊 编著

清华大学出版社

北京

内 容 简 介

本书系统地阐述了编译系统的结构、工作流程、设计原理和实现技术。主要内容包括程序设计语言基本知识、词法分析、语法分析、语义分析与属性文法、语法制导的代码生成、运行时存储空间组织、代码生成与优化等。通过本书的学习，使学生掌握编译理论和方法的基本知识，具有设计实现、分析和维护编译程序方面的初步能力，提高学生科学思维能力和综合运用专业知识的能力与解决实际问题的能力。书中每章开始有本章内容简介，每章后面都有与内容紧密相关、难度适宜的习题，可以使学生更好地掌握本章所学的知识内容。

本书适合作为高等学校计算机及相关专业的教材，也可以作为考研学生的一本参考书。

图书在版编目（CIP）数据

编译原理 / 康慕宁，任国霞，唐晶磊编著．—北京：清华大学出版社，2009.7
（21 世纪大学本科计算机专业系列教材）
ISBN 978-7-302-19705-8

Ⅰ．编…　Ⅱ．①康…②任…③唐…　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(2009)第 036852 号

责任编辑：张瑞庆　顾　冰
责任校对：梁　毅
责任印制：孟凡玉

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 装 者：北京国马印刷厂
经　　销：全国新华书店
开　　本：185×230　　**印　　张**：14　　**字　　数**：280 千字
版　　次：2009 年 7 月第 1 版　　**印　　次**：2009 年 7 月第 1 次印刷
印　　数：1～4000
定　　价：19.80 元

21世纪大学本科计算机专业系列教材编委会

本书责任编委：宋方敏

序言

PREFACE

21世纪是知识经济的时代，是人才竞争的时代。随着21世纪的到来，人类已步入信息社会，信息产业正成为全球经济的主导产业。计算机科学与技术在信息产业中占据了最重要的地位，这就对培养21世纪高素质创新型计算机专业人才提出了迫切的要求。

为了培养高素质创新型人才，必须建立高水平的教学计划和课程体系。在20多年跟踪分析ACM和IEEE计算机课程体系的基础上，紧跟计算机科学与技术的发展潮流，及时制定并修正教学计划和课程体系是尤其重要的。计算机科学与技术的发展对高水平人才的要求，需要我们从总体上优化课程结构，精炼教学内容，拓宽专业基础，加强教学实践，特别注重综合素质的培养，形成“基础课程精深，专业课程宽新”的格局。

为了适应计算机科学与技术学科发展和计算机教学计划的需要，要采取多种措施鼓励长期从事计算机教学和科技前沿研究的专家教授积极参与计算机专业教材的编著和更新，在教材中及时反映学科前沿的研究成果与发展趋势，以高水平的科研促进教材建设。同时适当引进国外先进的原版教材。

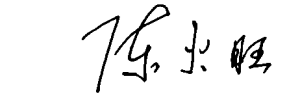
为了提高教学质量，需要不断改革教学方法与手段，倡导因材施教，强调知识的总结、梳理、推演和挖掘，通过加快教案的不断更新，使学生掌握教材中未及时反映的学科发展新动向，进一步拓宽视野。教学与科研相结合是培养学生实践能力的有效途径。高水平的科研可以为教学提供最先进的高新技术平台和创造性的工作环境，使学生得以接触最先进的计算机理论、技术和环境。高水平的科研还可以为高水平人才的素质教育提供良好的物质基础。学生在课题研究中不但能了解科学研究的艰辛和科研工作者的奉献精神，而且能熏陶和培养良好的科研作风，锻炼和培养攻关能力和协作精神。

进入21世纪，我国高等教育进入了前所未有的大发展时期，时代的进步与发展对高等教育质量提出了更高、更新的要求。2001年8月，教育部颁发了《关于加强高等学校本科教学工作，提高教学质量的若干意见》。文件指出，本科教育是高等教育的主体

和基础，抓好本科教学是提高整个高等教育质量的重点和关键。随着高等教育的普及和高等学校的扩招，在校大学本科计算机专业学生的人数将大量上升，对适合21世纪大学本科计算机科学与技术学科课程体系要求的，并且适合中国学生学习的计算机专业教材的需求量也将急剧增加。为此，中国计算机学会和清华大学出版社共同规划了面向全国高等院校计算机专业本科生的**"21世纪大学本科计算机专业系列教材"**。本系列教材借鉴美国ACM和IEEE最新制定的*Computing Curricula 2005*（简称CC2005）课程体系，反映当代计算机科学与技术学科水平和计算机科学技术的新发展、新技术，并且结合中国计算机教育改革成果和中国国情。

中国计算机学会教育专业委员会和全国高等学校计算机教育研究会，在清华大学出版社的大力支持下，跟踪分析CC2001，并结合中国计算机科学与技术学科的发展现状和计算机教育的改革成果，研究出了《中国计算机科学与技术学科教程2002》（China Computing Curricula 2002，简称CCC2002），该项研究成果对中国高等学校计算机科学与技术学科教育的改革和发展具有重要的参考价值和积极的推动作用。

"21世纪大学本科计算机专业系列教材"正是借鉴美国ACM和IEEE CC2005课程体系，依据CCC2002基本要求组织编写的计算机专业教材。相信通过这套教材的编写和出版，能够在内容和形式上显著地提高我国计算机专业教材的整体水平，继而提高我国大学本科计算机专业的教学质量，培养出符合时代发展要求的具有较强国际竞争力的高素质创新型计算机人才。



中国工程院院士

国防科学技术大学教授

21世纪大学本科计算机专业系列教材编委会名誉主任

前言

FOREWORD

编译原理是计算机专业中一门重要的专业课程。设置本课程的目的，在于系统地向学生讲述编译系统的结构、工作流程、设计原理和实现技术。通过本书的学习，使学生掌握在编译理论和方法方面的基本知识，也具有设计实现、分析和维护编译程序方面的初步能力，也可以提高学生科学思维能力和综合运用专业知识与解决实际问题的能力，所以这门课程在计算机专业体系中占有十分重要的地位。

编译原理是一门理论性和实践性都比较强的课程。编者在编写过程中，力图尽可能使本教材达到知识内容组织合理，由浅入深，循序渐进的目的。书中详细而系统地介绍了程序设计语言翻译的基本原理与方法，同时本书的特色主要体现在每章开始都有本章内容简介，每章后面都有与讲解内容紧密相关的习题，可以使学生更好地掌握所学的知识内容，也可以成为考研学生很好的一本参考书籍。本书共分为 11 章，第 1～6 章由西北工业大学计算机学院的康慕宁教授编写，第 7～9 章由西北农林科技大学信息工程学院的唐晶磊编写，第 10、11 章由西北农林科技大学信息工程学院的任国霞副教授编写。

我们特别感谢北京工业大学蒋宗礼教授，蒋教授对本书进行了仔细的审阅，提出了许多宝贵的意见及指导性的建议。这里也特别感谢清华大学出版社的大力支持，使本书得以顺利出版。

由于我们的能力有限，书中难免有一些错误和不足，敬请读者批评指正。

作　者

2009 年 1 月

目 录

CONTENTS

第 1 章 编译程序理论概述

1.1 导　　言

编译程序设计是少数几个由抽象的理论改变人们编写程序方法的计算机科学领域之一。最早的编译程序采用传统的“对号入座”的程序设计技术，直到“语法制导”分析程序的出现才得到改变。一个好的编译程序的任何部分都或多或少地与语法相关，因为语法理论可以驱动编译程序的设计，并使编译程序的设计清晰、易实现。

1.2 语言及其翻译

计算机语言像自然语言一样，为了沟通信息而定义了用单词构造句子的方法。自然语言可以表达、传递人们心灵的感觉、客观的世界真相、人们对这些感觉和真相提出的问题以及要求他人听从的指令等等，但计算机语言则被限制用于描述要求接受该语言的机器听从的指令。自然语言限制了允许说的句子的形式，例如，人们可以说：“约翰吃香蕉”，但不能说：“约翰香蕉吃”。第一句语法正确，而第二句则不符合中文的语法。又如，英文句子 John eats an apple 是正确的句子，而 eats John apple an 却不符合英语语法。会英、德两种语言的人马上就知道德语句子 John ißt einen Apfel 与 John eats an apple 说的是一回事，但对只熟悉其中一种语言的人来说，就不知道两句话同义了。因为，单词 ißt 在英语中无任何意义，同样 eats 也不是德语中的正确单词。

当一个不懂德语的英国人想与不懂英语的德国人相互交流时，就需要有一个翻译。在自然语言中，翻译是将他所接收的一种语言的信息表述为另一种语言的人。他将所读的英语句子 John eats an apple 写成德语句子 John ißt einen Apfel。对于错误的句子 eats John apple an，他将无法翻译。

计算机程序设计语言的编译程序就像会翻译两种语言的人一样，它从计算机语言中读取句子，若该句子在计算机语言中是有意义的，它就将其翻译为具有相同含义的另一种计算机语言的句子。当然，计算机语言中定义了使句子有意义的规则，编译程序可利用这些规则来判定它的输入是否有意义，并确保其输出也有相同意义。一个计算机语言程序就是该语言的句子的序列。编译程序的任务就是将一种语言（称为源语言(source language)）程序翻译为另一种语言（称为目标语言(target language)）的程序。

目前，世界上已有很多种计算机语言及其编译程序。最简单的翻译程序可读取简单的计算机语言单词，并直接将其翻译成计算机指令代码，这种翻译程序叫做汇编程序。汇编程序的源语言是汇编语言，目标语言是机器指令。大多数机器指令由若干部分组成，汇编语言使用单独的单词或数字来描述每个部分，由汇编程序将其组合成数字形式的机器代码。汇编程序可以由一个查表程序组成，源程序的每个单词可在该表中找到等价的数字形式的机器代码，这些数字代码将作为目标代码程序的一部分被输出。汇编语言通常使程序员可精确地直接使用计算机硬件的各种性能，但是，用汇编语言编写出正确的程序比使用其他大多数高级计算机语言困难得多。

术语“编译”通常用于更复杂的语言，因为复杂语言程序中的“单词”与目标语言之间不存在简单的或直接的关系。计算机语言翻译程序的目标就是将源程序变换成目标语言程序。多数编译程序将源程序翻译为汇编语言程序，然后再让汇编程序将编译程序所输出的汇编语言程序翻译成机器语言程序。我们把需要使用编译程序进行翻译的程序设计语言称为高级语言(High-Level Language，HLL)。HLL 描述问题求解更接近自然语言，而不是像机器指令那样只是简单的加、减、移位等操作序列。因此，高级语言更容易为程序员所接受和使用。例如，在商业应用中，COBOL(COmmon Business Oriented Language)语言使用了会计和中层管理人员更易懂的术语；科学问题常被描述为公式(formula)，为此，FORTRAN(FORmula TRANslator)语言更适合于它。现在有些程序员喜欢更抽象的 HLL 结构，希望程序设计语言既能像其他高级语言那样描述算法，又能具备像汇编程序所提供的一些“低级”控制能力，C 语言的产生解决了这一难题。

解释程序是 HLL 的翻译程序模式之一，它就像口译翻译人员边听、边理解、边翻译一样，一边阅读源程序，一边直接将其翻译（解释）为目标程序并执行之。解释程序一般不生成目标程序，即在解释执行完源程序后，不产生任何目标程序。而编译程序则是将源程序翻译成目标程序后，待以后需要时再执行。显然，若需经常执行的程序，使用编译程序将其翻译为目标代码后再执行，效率更高些。当然，解释程序与编译程序有许多共同之处。

1.3 语法的功能

在学习自然语言(如英语、法语)时,必须学习的一部分内容就是该语言的语法(grammar,也称为文法)。一个语言的语法定义了其句子在语法上的正确形式。例如,英语中可以有如下的语法规则:

```
<句子>      →<名词短语><动词><名词短语>
<动词>      →eats|hits
<名词短语>  →<冠词><名词>|<特有名字>
<冠词>      →a|the
<名词>      →ball|banana
<特有名字>  →Peter|John
```

该语法说明,句子是由两个名词短语及夹在其间的动词构成的。它把抽象的概念"句子"用一个符号<句子>表示出来。在该语法中,<句子>可被重写为(be rewritten as 或称替换为 be replaced as)表示三个抽象概念的符号序列:<名词短语><动词><名词短语>。<名词短语>又可以被替换为<特有名字>(又继之被替换为 John),或者被替换为<名词>(再被替换为带有冠词 a 的名词 banana);<动词>可被替换为 eats。最后,句子可被替换为 John eats a banana。根据定义规则可知,这是一个合法的英语句子。通过对语法定义规则的合理使用,还可以得到其他的句子,如 Peter hits John,a banana eats a ball,a ball eats Peter 等。上述规则中的箭头"→"或读作"可重写为"或"定义为"或"可替换为",其含义是箭头左部的称号可用箭头右部的符号序列取而代之,从而得到新的符号序列。当可替换的序列多于一个时,我们用特殊符号"|"(读作"或")分隔之,它表示箭头左部的符号可由产生式右部中由"|"分隔开的若干个符号序列中的任一个符号序列替换之。这样,上述替换规则可以将符号<句子>最终替换为 72 种不同的不能再进一步替换的符号序列,被称之为该语法所定义的语言中的"句子",即该语言有 72 个不同的句子。

事实上,一个程序设计语言通常被定义为两类语法,一类用于定义该语言的所有"词汇",另一类用于规定如何将这些词汇联合在一起构成"句子"。

从形式语言学的意义上来说,上面例子中的语法,我们称为前后文无关(亦称为上下文无关)文法,它所定义的语言称为前后文无关语言。目前人们所使用的大多数高级程序设计语言,其主要的结构大都可用前后文无关文法描述。由于前后文无关文法还不能完全描述程序设计语言的语义,因此,从严格意义上来说,一种被称为前后文有关文法的语法决定了编译程序的核心——语法分析器(parser)。语法定义了如何将计算

机语言的词汇组合成语法合理的程序。语法分析工作就是依据语法对构成程序的单词序列进行结构分析，判断其是否为合法的程序(即是否是该语言的句子)。

另一类语法常被用于描述程序设计语言的单词的“拼写”规则，因此也被称为词法(lexical grammar)。编译程序中进行词法分析的部分(或模块)被称为词法分析器或扫描器(scanner)。例如，我们可定义英语单词<word>的词法如下：

```
<word>   →<word><letter>|<letter>
<letter> →a|b|c|…|z
```

虽然，单词<word>的文法很简单且可以产生许多无意义的单词，但它告诉我们，一个很简单的文法可以产生一个非常大的语言。事实上，上述文法可以产生一个无限集：只要重复地使用第一个规则，即在尾部添加一个字母(letter)，这个单词的长度就可以任意地增长。

在实际的程序设计语言中，除使用文法进行定义外，还要通过在文法中附加属性及其属性值计算功能(因此也称为属性文法)来描述语义，即用属性概念来告诉编译系统应该如何根据程序的语义进行翻译。

1.4 程序设计语言的发展

在计算机诞生的初期，人们只能通过以0、1代码的指令序列形式告诉计算机应当如何工作。即用最“低级”的机器语言进行编程。显然，这样的编程模式所编制的程序难于阅读和理解，特别是在程序出现错误时，几乎无法定位出错地点。为此，人们使用一些便于记忆和理解的单词对程序中的所用的指令和量进行命名，这就是汇编语言程序。可以说，汇编语言是计算机机器语言的符号形式。世界上第一个高级程序设计语言是五十年代中期出现的公式翻译语言FORTRAN，它取名于FORmula TRANslation。FORTRAN语言使计算数学家们方便地把数学算法改写(翻译)为高级语言程序，而不必用0、1代码形式来编写复杂的计算机程序。从应用的角度上来讲，FORTRAN的出现具有划时代的意义。但是，FORTRAN语言的定义无法用形式语法来描述，而是用自然语言以说明的方式进行定义，从而使其编译程序的设计非常复杂，甚至它的词法分析程序(扫描器)的设计都相当困难。例如，在FORTRAN中规定空格是无意义的符号，可随意删除，扫描器在识别下面的两个FORTRAN语句中的单词时，需向后扫描许多符号才能“断词”成功：

```
DO 10 I=1,5
DO 10 I=1.5
```