

<<编译原理-编译程序构造实践教程>>

图书基本信息

书名：<<编译原理-编译程序构造实践教程>>

13位ISBN编号：9787115215123

10位ISBN编号：711521512X

出版时间：2010-4

出版时间：人民邮电出版社

作者：张幸儿，戴新宇 编著

页数：282

版权说明：本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问：<http://www.tushu007.com>

## &lt;&lt;编译原理-编译程序构造实践教程&gt;&gt;

## 前言

编译原理是计算机专业的一门重要课程，本课程开设的目的是讲解编译程序的构造原理。由于编译程序是把高级程序设计语言源程序翻译成等价的低级语言目标程序，其处理对象是符号序列，因此可以说编译程序是符号处理的工具，只要是与符号处理相关的领域，都将可能需要应用编译原理中的基本原理。

例如，编译原理中所讨论的正则表达式与状态转换图就应用在很多领域。

编译原理中的代码优化，尽管是在源程序的内部中间表示一级上进行的，但优化的思想可以为软件开发人员，特别是计算机程序设计爱好者所借鉴。

编译原理课程的特点是理论性强，因为它与形式语言理论紧密相关，可以说是在形式语言理论基础上讨论编译原理，这使得编译原理成为计算机专业最为困难的课程之一。

其实，实践性应是编译原理课程更为关注的问题，也就是说，掌握编译程序构造原理的目的是实践编译程序构造。

本书着力降低理论难度，围绕编译程序的编制开展讨论，以加强读者的实践能力。

本书有如下一些特点。

(1) 以C语言为背景。

紧密结合C语言，尽可能处处以C语言相关内容为例子进行讨论，使得讲解更有针对性，读者也更易于理解和接受。

(2) 突出实践性。

对编译过程的每个方面进行论述时，都有计算机实现的讨论，例如文法的存储表示、推导和语法分析树的生成、词法分析、识别程序句型分析，以及翻译方案的实现，都有相当的篇幅。

最后还讨论了符号模拟执行虚拟机目标代码的解释程序的编制，给出可运行的解释程序。

(3) 明确C语言语法成分目标代码结构与其他语言的区别。

例如函数调用中，实际参数的计算次序与赋值语句左部变量地址的计算次序等，全书有多个C语言程序作为例子，包括应用指针与结构，以及建立结点链的C程序编写等，使读者对C语言有更深入的认识。

(4) 加强讲解编程知识。

本书中，除了作为例子的一些经典算法，如排序程序与计算多项式的霍纳方案外，不乏编程的方法和技巧。

例如，文法的存储表示、LR分析表的计算机存储表示与特征变量的引进以及置初值等，以启发读者应用C语言编程的能力。

学习是吸收新知识的过程，更重要的是培养和提高分析问题和解决问题能力的过程。

一个常见的思维方法是：以简单的实例，通过对它的观察分析，得到解决问题的思路。

另一种惯用的思维方法是：从已给的输入（Input）和要得到的输出（Output）出发，设法进行处理（Process），以得到解决方案，这就是IPO。

读者可以体会本书中各种方法和技巧的应用。

总的来说，本书力图以简洁易懂的文字阐述主要的基本概念，用朴实的实例展示实用的方法，按直观的思维方法启发寻找问题的解答。

## <<编译原理-编译程序构造实践教程>>

### 内容概要

本书系统而简洁地介绍编译程序的构造原理，内容主要包括：概论、编译程序构造的基础知识、词法分析、语法分析、语义分析与目标代码生成、中间表示代码与代码优化、程序错误的检查与校正、目标代码的运行，以及虚拟机目标程序的解释程序的编制。

各章开始于本章导读，各章末有本章小结、复习思考题以及习题。

本书突出实践性，在编译程序构造各个环节中，提供了具体可行的实现方法和技巧，供读者参考。

本书可作为计算机及相关专业的编译原理课程教材，也可作为计算机软件技术人员、研究生及广大计算机爱好者的参考用书。

## 作者简介

张幸儿，南京大学计算机科学与技术系教授，长期从事教学和科研工作，参与多个编译程序和软件项目的研制，长期从事编译原理、程序设计等课程的教学。

有丰富的教学经验和计算机软件开发经验。

多次获得省部级科研与优秀教材奖，其中《计算机编译理论》和《计算机编译原理》分别获得教育部普通高校优秀教材二等奖。

戴新宇，南京大学计算机科学与技术系副教授，博士，研究方向是自然语言处理，多次讲授计算机编译原理课程。

## &lt;&lt;编译原理-编译程序构造实践教程&gt;&gt;

## 书籍目录

第1章 概论 1.1 编译程序概况 1.2 编译程序的构造 1.3 编译程序的实现 本章小结 复习思考题  
第2章 编译程序构造的基础知识 2.1 符号串与符号串集合 2.2 文法与语言 2.3 句型分析 2.4 语法分析树的计算机生成 本章小结 复习思考题 习题  
第3章 词法分析 3.1 概况 3.2 词法分析程序的手工实现 3.3 词法分析程序的自动生成 本章小结 复习思考题 习题  
第4章 语法分析——自顶向下分析技术 4.1 自顶向下分析技术概况 4.2 无回溯的自顶向下分析技术 4.3 预测识别程序句型分析的计算机实现 本章小结 复习思考题 习题  
第5章 语法分析——自底向上分析技术 5.1 自底向上分析技术概况 5.2 LR(1)分析技术 5.3 其他的自底向上分析技术 本章小结 复习思考题 习题  
第6章 语义分析与目标代码生成 6.1 概况 6.2 说明部分的翻译 6.3 类型检查 6.4 目标代码的生成 6.5 翻译方案的实现 本章小结 复习思考题 习题  
第7章 中间表示代码与代码优化 7.1 概况 7.2 源程序的中间表示代码 7.3 基本块的代码优化 7.4 与循环有关的优化 7.5 全局优化的实现思想 7.6 窥孔优化 本章小结 复习思考题 习题  
第8章 程序错误的检查与校正 第9章 目标代码的运行 第10章 虚拟机目标程序的解释程序的研制 参考文献

## 章节摘录

语法分析。

词法分析时识别开各个符号之后，由语法分析部分根据程序设计语言的语法规则，识别出各个语法成分，最终识别出完整的程序。

在识别各类语法成分的同时，也就检查了语法的正确性。

当识别出是语法上正确的程序时，生成相应的内部中间表示（通常是语法分析树或其他内部中间表示），如果存在错误，则给出相应的报错信息。

完成语法分析的部分称为语法分析程序，或称为识别程序。

概括起来，语法分析程序的功能是识别出各个语法成分，生成相应的内部中间表示，同时进行语法正确性的检查。

语义分析。

编译程序继语法分析之后进行语义分析，即基于语法分析时输出的内部中间表示，依据各个语法成分的含义进行语义分析。

由于一个程序通常由数据结构和控制结构两部分组成，必然对这两部分进行语义分析。

对于数据结构，语义分析部分进行的语义分析工作是确定类型和类型检查，确切地说，检查标识符是否有定义，确定标识符所对应数据对象的数据类型等属性，检查运算的合法性及运算分量数据类型的一致性；对于控制结构，根据程序设计语言所规定的语义，对它们进行相应的语义处理，这时可以生成相应的目标代码。

例如，对于一个加法运算，当检查了两个运算分量都有定义，它们都能进行加法运算（运算是合法的），且两个运算分量的类型一致（相容）时，可以生成进行加法的目标代码。

不言而喻，执行语义分析的同时，还进行一些语义检查，当然这只是静态语义检查，即在编译时刻所能进行的语义检查，例如，检查是否从循环外通过控制转移语句把控制转入循环体。

在运行时刻才能进行的语义检查称为动态语义检查，如检查数组元素下标是否越界，以及指针变量是否有初值等，自然不在语义分析时刻进行。

为了改进目标程序质量，语义分析时可能不生成目标代码，而是生成另外一种内部中间表示，或称中间表示代码。

代码优化阶段就是基于这种中间表示代码进行优化，然后再从优化了的中间表示代码生成目标代码。

语义分析工作通常由语义子程序完成。

完成语义分析的部分称为语义分析程序。

概括起来，语义分析程序的功能是确定类型、类型检查、识别含义与相应语义处理，以及其他一些静态语义检查等。

代码优化。

代码优化指的是为改进目标程序质量而在编译时刻进行的各项优化工作。

代码优化通常基于语义分析部分生成的中间表示代码进行，把它变换成功能相同、但功效更高的优化了的中间表示代码。

本书的中间表示代码主要是四元式序列。

代码优化有与机器无关的优化和与机器有关的优化之分。

本书重点讨论与机器无关的优化。

除了对中间表示代码进行优化外，也可以对目标代码进行优化，这种优化通常称为窥孔优化。

窥孔优化是机器语言级上仅在一个很小的范围内进行的、不太复杂因而代价不是很高的一类优化。

当前的很多高级程序设计语言的编译程序都进行代码优化，c语言编译程序是突出的优化典型。

完成代码优化的部分称为代码优化程序。

概括起来，代码优化程序的功能是：在编译时刻基于中间表示代码进行目标程序质量的改进。

目标程序生成。

如前所述，语义分析时可以直接生成目标程序，这里的目标程序生成指的是基于优化了的中间表示代码（也即四元式序列）生成目标程序。

<<编译原理-编译程序构造实践教程>>

目标程序的生成与运行目标程序的计算机密切相关。

为了学习的目的，本书避免把大量的时间花在对具体计算机细节的了解上，将基于一种所谓的虚拟机来讨论目标程序的生成。

## <<编译原理-编译程序构造实践教程>>

### 编辑推荐

牢牢掌握编译程序构造的基本原理 深入理解C程序设计语言的特性 实践中领悟、实践中提升 编译原理课程讨论的对象是编译程序。  
编译程序是符号处理的工具，只要是与符号处理相关的领域，都将可能需要应用编译程序构造的基本原理。

仅讨论编译原理最基本的内容，突出从源程序读入到目标程序生成这一主线。

紧密结合C语言。

以C语言相关内容为例进行讨论，更具针对性。

突出实践性。

对编译过程的每个关键部分都有计算机实现的讨论，提供具体而切实可行的实现方法和技巧，读者将对概念与计算机实现之间的差异有深刻的体会，对C语言的应用能力将有较大的提升，并能积累程序编写和软件开发的经验。

《编译原理：编译程序构造与实践教程》不是单纯地给出问题的解答，注重对重点问题进行分析，阐述思路。

结合各种方法以实例说明，使学生在理解的基础上掌握问题的求解方法。

概括地说，《编译原理：编译程序构造与实践教程》力图以简洁易懂的文字阐述主要的基本概念，用朴实的实例展示实用的方法，按直观的思维方法启发寻找问题的解答。



版权说明

本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问:<http://www.tushu007.com>