

<<编译原理及实现>>

图书基本信息

书名：<<编译原理及实现>>

13位ISBN编号：9787302265849

10位ISBN编号：7302265844

出版时间：2011-11

出版时间：清华大学出版社

作者：孙悦红

页数：256

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>

<<编译原理及实现>>

内容概要

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。

<<编译原理及实现>>

书籍目录

第1章 编译概述

- 1.1 程序设计语言
- 1.2 翻译程序
- 1.3 编译程序的组成
- 1.4 编译程序的结构
- 1.5 编译程序的前后处理器
- 1.6 test语言与编译器

习题

第2章 文法和语言

- 2.1 字母表和符号串
- 2.2 文法
- 2.3 推导
- 2.4 句型和句子
- 2.5 语言
- 2.6 递归规则与递归文法
- 2.7 短语、简单短语和句柄
- 2.8 语法树
- 2.9 子树与短语
- 2.10 由树构造推导过程
- 2.11 文法的二义性
- 2.12 有关文法的实用限制
- 2.13 文法和语言分类

习题

第3章 词法分析

- 3.1 词法分析的功能
- 3.2 程序语言的单词符号种类及词法分析输出
- 3.3 正则文法及状态图
- 3.4 词法分析程序的设计与实现
- 3.5 正则表达式
- 3.6 有穷自动机
- 3.7 词法分析程序的自动生成器lex

习题

第4章 语法分析——自顶向下分析

- 4.1 自顶向下分析方法
- 4.2 first集合和follow集合
- 4.3 递归下降分析
- 4.4 LL(1)分析方法

习题

第5章 语法分析——自底向上分析

- 5.1 规范推导、规范句型和规范归约
- 5.2 自底向上分析方法的一般过程
- 5.3 LR分析方法
- 5.4 LR(0)分析器
- 5.5 SLR(1)分析器
- 5.6 LR(1)分析器

<<编译原理及实现>>

5.7 lalr(1)分析器

5.8 语法分析程序的自动生成工具——yacc

习题

第6章 语法制导翻译技术

6.1 翻译文法

6.2 语法制导翻译

6.3 自顶向下语法制导翻译

6.4 属性翻译文法

6.5 属性文法的自顶向下翻译

6.6 自底向上语法制导翻译

习题

第7章 符号表管理技术

7.1 何时建立和访问符号表

7.2 符号表的组织和内容

7.3 符号表上的操作

7.4 非块程序结构语言的符号表结构

7.5 块程序结构语言的符号表组织

习题

第8章 程序运行时的存储组织及管理

8.1 程序运行时的存储组织

8.2 静态存储分配

8.3 栈式动态存储分配

8.4 堆式动态存储分配

习题

第9章 语义分析和代码生成

9.1 语义分析的概念

9.2 中间代码

9.3 声明的处理

9.4 表达式语句

9.5 if语句

9.6 while语句

9.7 for循环语句

9.8 write_语句

9.9 read_语句

9.10 过程调用和返回

9.11 语义分析及代码生成实现

9.12 错误处理

习题

第10章 代码优化

10.1 局部优化

10.2 循环内的优化

习题

附录a test语言文法规则

a.1 test语言词法规则

a.2 test的语法规则

a.3 test的语义和代码生成

附录b 词法分析程序

<<编译原理及实现>>

- b.1 词法分析程序
- b.2 主程序
- 附录c 语法分析程序
 - c.1 语法分析程序
 - c.2 主程序
- 附录d 语义及代码生成程序
 - d.1 语法、语义及代码生成程序
 - d.2 主程序
- 附录 etest抽象机模拟器完整程序
 - e.1 testmachine函数
 - e.2 主程序

<<编译原理及实现>>

章节摘录

版权页：插图：处理过程调用和返回时，主要涉及下列两个基本问题：（1）实现程序中过程调用和返回的控制逻辑；（2）处理实在参数和形式参数之间的数据传递问题。

在过程调用时，要将实在参数传递给形式参数。

在执行过程或从过程返回时，要将过程的处理结果返回给相应的主调过程。

在过程调用时不同语言、不同性质的形式参数将采用不同的数据传递方式，而不同的数据传递方法又将对过程和过程调用的语义分析和代码生成产生影响。

9.10.1 参数的基本传递形式目前，常见的参数传递有值传递、引用传递（地址传递）、值结果传递和名字传递。

这里，只介绍两个最常用的参数传递机制，即值传递和引用传递，对其他两个传递机制有兴趣的读者可查阅有关书籍。

1. 值传递 值传递又称为值调用，它是最简单的数据传递方式。

在这种参数传递机制下，调用程序段要把实在参数的值计算出来，并存放在操作数栈、寄存器或被调过程能够取得的数据单元中；而在被调用的子程序或函数中，首先要将实在参数的值送进相应的形式参数的数据单元中。

编译程序对过程体中形式参数的处理就像处理一般实在参数标识符那样，生成目标代码。

在这种形式参数与实在参数的结合方式下，数据传递是单方向进行的，即调用段将实在参数的值传递到被调用段相应的形式参数的数据单元中；而在执行被调用段的过程中，不可能将赋值形参的数据单元中的内容再传回调用段的相应实在参数单元中去。

C语言就采用这种参数传递机制。

2. 引用传递 所谓引用传递就是传递地址，又叫引用调用。

在这种参数传递机制下，调用程序段要将实在参数的地址传递给相应的形式参数。

对于实际参数的各种情况，处理方法如下。

（1）若实在参数是简单变量，则编译程序要生成将它们的地址保存在操作数栈、寄存器或某个被调过程能够取得的数据单元的代码；（2）若实在参数是数组元素，则除上述代码外，在调用段中还应先有计算该数组元素地址的代码；（3）若实在参数是表达式或常量，则编译程序应先分配一个临时数据单元，在调用程序中先要有计算表达式的值送进临时数据单元中的代码，还要有将该临时数据单元的地址保存在操作数栈、寄存器或某个被调过程能够取得的数据单元的目标代码。

而在被调程序段，对形参的处理方法是：首先将实在参数的地址抄进自己相应的形式参数的数据单元中。

过程体中对形式参数的引用或赋值都将被处理成对形式参数数据单元的间接访问。

<<编译原理及实现>>

编辑推荐

《编译原理及实现(第2版)》教材以通俗易懂的语言讲解编译原理，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，并注重介绍各种编译方法的应用实现。

本教材考虑到目前学计算机专业的学生对C语言比较了解，以C语言为雏形设计了一种TEST语言，并建立该语言的词法、语法、语义文法规则，系统介绍编译过程的各个部分，摆脱了以往编译教材的抽象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，目前多数已出版的编译原理教材普遍偏重于理论，对实现技术讲解的极少或根本没有，而且教材内容过多，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利用，而《编译原理及实现(第2版)》以通俗易懂的语言讲解编译原理，注重理论与实践相结合，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，从而使学习者不再畏惧编译原理的学习，提高学习并动手编写编译器的兴趣？

<<编译原理及实现>>

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>