

<<编译原理>>

图书基本信息

书名：<<编译原理>>

13位ISBN编号：9787111288183

10位ISBN编号：7111288181

出版时间：2010-3

出版时间：机械工业出版社

作者：何炎祥

页数：246

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>>

前言

编译程序(Compiler)是计算机的重要系统软件,是高级程序设计语言的支撑基础。

本书主要介绍设计和构造编译程序的基本原理和方法。

本书共分12章。

第1章讲述编译程序的功能、结构、工作过程、组织方式、编译程序与高级语言的关系以及编译自动化方面的基本知识。

第2章介绍形式语言理论,我们仅仅给出了便于理解、有助于研究各种分析方法和设计构造编译程序的形式语言理论,并着重介绍上下文无关文法。

有穷自动机是描述词法的有效工具,也是进行词法分析的主要理论基础。

因此,第3章专门介绍有穷状态自动机,它与正规文法、正规表达式之间的对应关系以及它的确定化和最小化方面的知识。

第4章讨论词法分析程序的设计方法。

上下文无关文法可用于描述现今大多数高级程序设计语言的语法,也是语法分析的主要理论支柱。

为此,在接下来的几章里,主要讨论与上下文无关文法相关的各类语法分析方法。

第5章介绍自顶向下分析方法,包括LL(k)文法、LL(1)分析方法和应用十分广泛的递归下降分析方法。

第6章讨论自底向上分析方法的一般原理和优先分析方法,包括简单优先分析技术和算符优先分析方法。

第7章专门讨论自底向上的LR(k)分析方法,包括LR(0)、SLR(1)、规范LR(1)以及LALR(1)分析表的构造算法。

第8章介绍语法制导翻译方法(SDTS),主要讨论SDTS的基本原理、属性翻译文法及其在中间代码生成中的应用。

第9章讨论运行时的存储组织与管理,其中考虑了一些重要的语言特征,如过程调用、参数传递、数组和记录的存取方式以及多种存储分配技术。

第10章讨论符号表的组织和存取符号表的各种方法。

第11章介绍常用的优化方法。

第12章主要通过具体机器无关的PL/0语言的目标代码生成过程,讨论了代码生成的原理。

考虑到完整性,我们在附录中给出了PL/0编译程序源程序。

“编译原理”这门课程是计算机专业的主干课和必修课,也是计算机专业工程硕士课程中较难学习的一门课程,其先导课程是汇编语言程序设计、计算机组成原理、数据结构、高级语言程序设计和离散数学等。

本课程的参考学时数72,教师可根据具体情况对教材内容进行取舍,例如,工科院校的学生可略过第7、8章并可精减第2、3章的内容,从而使授课学时数减至54。

教师在讲授本课程的同时应加强并引导学生做好该课程实践方面的规划和检验。

本书特别注重设计和构造编译程序的理论与实践、原理与方法的互通,而且基本概念清晰,深入浅出,循序渐进,系统性强。

每章之后还附有难度不一的习题供学生复习、思考和探索。

本书既可作为高等院校计算机专业工程硕士及相关层次学生的教材,也可供有关师生和科技工作者及软件开发人员学习和参考。

本书的第9、10、12章和附录由伍春香编写,第11章由王汉飞编写,其余各章由何炎祥编写。

何炎祥和王汉飞最后统稿。

本书是在我们已经出版的几本编译原理教材的基础上经过适当修订而成。

在修订过程中得到了原合作者的全力支持。

此外,书中引用了一些专家学者的论著和研究成果,以及一些公司的产品介绍,在此一并表示真诚的感谢。

限于水平,书中错误难免,敬请读者指正。

<<编译原理>>

内容概要

本书主要介绍设计和构造编译程序的基本原理和方法，内容包括适合于计算机高级程序设计语言翻译的形式语言和自动机理论、常用的词法分析方法、各种经典的语法分析方法、语法制导翻译方法、存储器的组织与管理方法、符号表的组织与造查表方法、代码优化和代码生成方法、并行编译程序及编译自动化技术等。

本书特别注重理论与实践、原理与方法的互通，而且基本概念清晰，深入浅出，循序渐进，系统性强。各章之后还附有难度不一的习题供复习、思考和探索之用。

本书既可作为高等院校计算机专业的教材，也可供科技工作者及软件研发人员学习和参考。

<<编译原理>>

作者简介

何炎祥，教授、博士生导师，武汉大学计算机学院院长。

主要研究方向为：分布并行处理（含移动计算与Web服务）、可信软件、数据挖掘和软件工程等。

主持和主要承担的科研项目包括863高技术计划项目、国家自然科学基金重大研究计划项目、国家自然科学基金面上项目、省市科技攻关

<<编译原理>>

书籍目录

出版者的话 序言 前言 教学建议 第1章 引论 1.1 程序设计语言的发展 1.1.1 程序设计语言 1.1.2 翻译程序 1.2 为什么需要编译程序 1.3 编译程序的工作过程 1.3.1 分析部分 1.3.2 综合部分 1.4 编译程序的结构 1.4.1 编译程序的典型结构 1.4.2 编译程序的前端和后端 1.4.3 编译程序的分遍 1.4.4 源程序中的错误及出错处理 1.5 编译程序的组织方式 1.6 编译程序的其他技术 1.6.1 编译程序的自展技术 1.6.2 编译程序的移植技术 1.6.3 编译程序的自动化技术 1.6.4 程序的可再入性 1.7 翻译程序的编写系统 1.8 并行编译程序 1.9 小结 习题第2章 形式语言概论 2.1 语言成分 2.2 文法和语言 2.2.1 产生式文法 2.2.2 上下文无关文法 2.2.3 推导与直接推导 2.3 文法的分类 2.3.1 文法分类 2.3.2 文法分类的意义 2.3.3 文法举例 2.4 语言和语法 2.4.1 句型、句子和语言 2.4.2 语法树 2.4.3 产生式树和产生式图 2.5 文法和语言的一些特性 2.5.1 无用非终结符号 2.5.2 不可达文法符号 2.5.3 可空非终结符 2.5.4 最左推导、最右推导和规范推导 2.5.5 二义性 2.6 分析方法简介 2.6.1 自顶向下分析方法 2.6.2 确定的自顶向下分析方法 2.6.3 自底向上分析方法 2.6.4 文法在内存中的表示 2.7 小结 习题第3章 有穷自动机第4章 词法分析第5章 自顶向下语法分析第6章 自底向上分析和优先分析方法第7章 自底向上的LR(k)分析方法第8章 语法制导翻译法第9章 运行时的存储组织与管理第10章 符号表第11章 优化第12章 代码生成附录 PL/0编译程序源程序参考文献

<<编译原理>>

章节摘录

版权页：插图：一个高级语言的编译程序就是这个高级语言的翻译程序。

每个计算机程序都表示了一种行为，都可以完成某些特定任务。

对于编译程序的主要要求在于源程序与目标程序的行为应该是完全相同的。

也就是说，不管由一个高级语言程序（如一个C或Java程序）转换得来的机器语言程序的细节如何，这个高级语言程序对于相同的输入都应该产生相同的结果，而且执行相应机器语言程序的结果与执行原来高级语言程序的结果也是完全一致的。

这样，编译程序的存在就可以使高级语言独立于机器。

程序员在使用高级语言编写程序时不需要考虑那些与机器有关的繁琐细节，而可以把和机器有关的问题交给编译程序去处理。

众所周知，我们不可能通过替换几个单词就能简单地把一种语言翻译成另一种语言。

而必须首先分析源程序，弄清楚它的基本含义和结构。

这个过程称为“分析”（Parsing）。

然后在此基础上进行一次或多次转换，最终构造出相应的目标程序。

从图1.1可知，在编译方式中源程序的编译和目标程序的运行是分成两个阶段完成的。

编译所得的目标程序，计算机暂不能直接执行，还必须由连接装配程序（linker-loader）将目标程序和系统子程序等连接成一个可执行程序，这个可执行程序则可直接被计算机执行。

由于编译方式具有上述这些特征，因此，很多高级语言，如FORTRAN、ALGOL、PASCAL、COBOL、C、C++语言等，均采用这种编译方式。

<<编译原理>>

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>