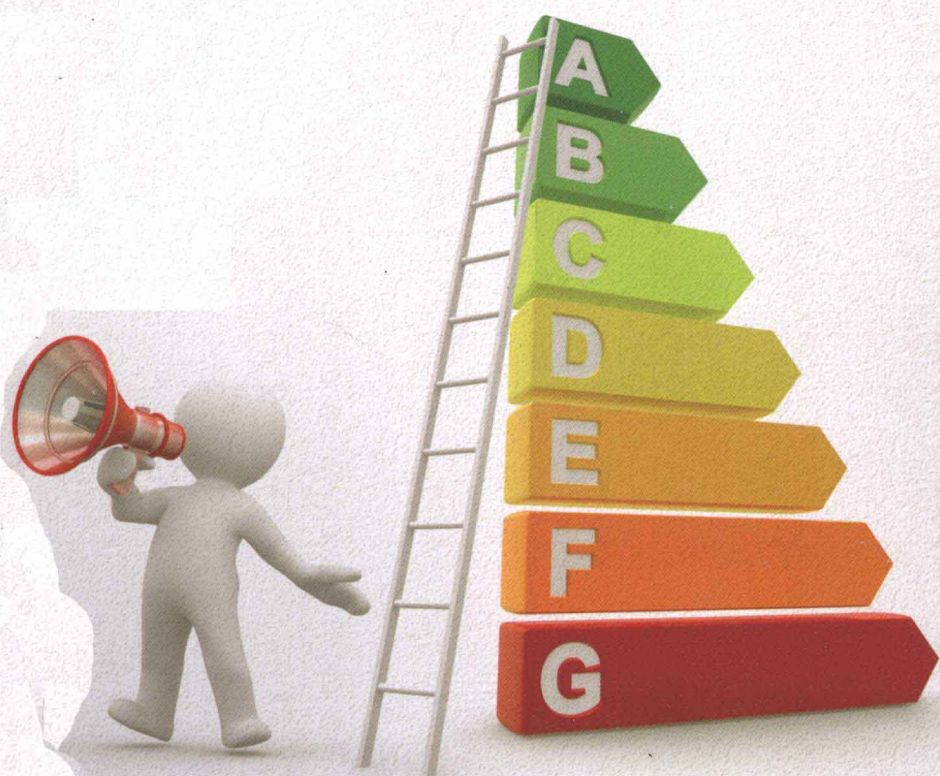


高等学校计算机类课程应用型人才培养规划教材

编译原理基础与应用

Fundamentals of Compiler Principle and its Application

游晓明 刘 升 编著



中国铁道出版社
CHINA RAILWAY PUBLISHING HOUSE

高等学校计算机类课程应用型人才培养规划教材

编译原理基础与应用

游晓明 刘 升 编著

中国铁道出版社
CHINA RAILWAY PUBLISHING HOUSE

内 容 简 介

本书系统、全面地介绍了计算机高级语言编译程序的基本原理和技术,对当前最新的嵌入式系统编译技术做了较详细的讨论。主要内容包括词法分析、语法分析、语法制导翻译的语义分析与中间代码生成、符号表与运行时存储空间的组织与代码优化、面向对象语言编译方法及并行编译基础、嵌入式系统编译器及其工具的应用开发。本书理论与实践并重,每章之前都有导读,最后均附有小结和习题,便于读者学习。

本书适合作为普通高等学校计算机学科及相关专业的本科教材,也可供科技工作者及软件研发人员学习和参考。

图书在版编目(CIP)数据

编译原理基础与应用 / 游晓明, 刘升编著. —北京:
中国铁道出版社, 2011.9
高等学校计算机类课程应用型人才培养规划教材
ISBN 978-7-113-13005-3

I. ①编… II. ①游… ②刘… III. ①编译程序—程
序设计—高等学校—教材 IV. ①TP314

中国版本图书馆 CIP 数据核字 (2011) 第 126262 号

书 名: 编译原理基础与应用
作 者: 游晓明 刘 升 编著

策划编辑: 严晓舟
责任编辑: 周海燕
助理编辑: 王 惠
责任印制: 李 佳

读者热线: 400-668-0820
封面设计: 付 巍
封面制作: 白 雪

出版发行: 中国铁道出版社(北京市宣武区右安门西街8号 邮政编码: 100054)
印 刷: 北京东海印刷有限公司
版 次: 2011年9月第1版 2011年9月第1次印刷
开 本: 787mm×1092mm 1/16 印张: 17.25 字数: 412千
印 数: 1~3000册
书 号: ISBN 978-7-113-13005-3
定 价: 28.00元

版权所有 侵权必究

凡购买铁道版图书, 如有印制质量问题, 请与本社教材研究开发中心批销部联系调换。

编 审 委 员 会

主 任：何新贵 教授、中国工程院院士 (北京大学)

副主任：(按姓氏笔画排序)

刘瑞挺 教授	(南开大学)
罗晓沛 教授	(中国科学院研究生院)
岳丽华 教授	(中国科学技术大学)
周广声 教授	(上海海事大学)
徐洁磐 教授	(南京大学)

委 员：(按姓氏笔画排序)

王 浩 教授	(合肥工业大学)
王晓峰 教授	(上海海事大学)
史九林 教授	(南京大学)
白晓颖 教授	(清华大学)
刘 强 副教授	(清华大学)
许 勇 教授	(安徽师范大学)
孙志挥 教授	(东南大学)
李龙澍 教授	(安徽大学)
李银胜 副教授	(复旦大学)
李盛恩 教授	(山东建筑大学)
李敬兆 教授	(安徽理工大学)
杨吉江 教授	(清华大学)
何守才 教授	(上海第二工业大学)
余 粟 副教授	(上海工程技术大学)
张 莉 教授	(北京航空航天大学)
张 瑜 教授	(上海工程技术大学)
张燕平 教授	(安徽大学)
陈世福 教授	(南京大学)
陈涵生 研究员	(上海华东计算技术研究所)
迟洪钦 副教授	(上海师范大学)
林钧海 教授	(南京航空航天大学)
金志权 教授	(南京大学)
周鸣争 教授	(安徽工程科技学院)
周根林 教授级高级工程师	(中电集团)
郑人杰 教授	(清华大学)
胡学钢 教授	(合肥工业大学)
姜同强 副教授	(北京工商大学)
徐永森 教授	(南京大学)
殷人昆 教授	(清华大学)
郭学俊 教授	(河海大学)
黄上腾 教授	(上海交通大学)
董继润 教授	(山东大学)
蔡瑞英 教授	(南京工业大学)

当前,世界格局深刻变化,科技进步日新月异,人才竞争日趋激烈。我国经济建设、政治建设、文化建设、社会建设以及生态文明建设全面推进,工业化、信息化、城镇化和国际化深入发展,人口、资源、环境压力日益加大,调整经济结构、转变发展方式的要求更加迫切。国际金融危机进一步凸显了提高国民素质、培养创新人才的重要性和紧迫性。我国未来发展关键靠人才,根本在教育。

高等教育承担着培养高级专门人才、发展科学技术与文化、促进现代化建设的重大任务。近年来,我国的高等教育获得了前所未有的发展,大学数量从1950年的220余所已上升到2008年的2200余所。但目前诸如学生适应社会以及就业和创业能力不强,创新型、实用型、复合型人才紧缺等高等教育与社会经济发展不相适应的问题越来越凸显。2010年7月发布的《国家中长期教育改革和发展规划纲要(2010—2020年)》提出了高等教育要“建立动态调整机制,不断优化高等教育结构,重点扩大应用型、复合型、技能型人才培养规模”的要求。因此,新一轮高等教育类型结构调整成为必然,许多高校特别是地方本科院校面临转型和准确定位的问题。这些高校立足于自身发展和社会需要,选择了应用型发展道路。应用型本科教育虽早已存在,但近几年才开始大力发展,并根据社会对人才的需求,扩充了新的教育理念,现已成为我国高等教育的一支重要力量。发展应用型本科教育,也已成为中国高等教育改革与发展的重要方向。

应用型本科教育既不同于传统的研究型本科教育,又区别于高职高专教育。研究型本科培养的人才将承担国家基础型、原创型和前瞻型的科学研究,它应培养理论型、学术型和创新型的研究人才。高职高专教育培养的是面向具体行业岗位的高素质、技能型人才,通俗地说,就是高级技术“蓝领”。而应用型本科培养的是面向生产第一线的本科层次的应用型人才。由于长期受“精英”教育理念的支配,脱离实际、盲目攀比,高等教育普遍存在重视理论型和学术型人才培养的偏向,忽视或轻视应用型、实践型人才的培养。在教学内容和教学方法上过多地强调理论教育、学术教育而忽视实践能力的培养,造成我国“学术型”人才相对过剩,而应用型人才严重不足的被动局面。

应用型本科教育不是低层次的高等教育,而是高等教育大众化阶段的一种新型教育层次。计算机应用型本科的培养目标是:面向现代社会,培养掌握计算机学科领域的软硬件专业知识和专业技术,在生产、建设、管理、生活服务等第一线岗位,直接从事计算机应用系统的分析、设计、开发和维护等实际工作,维持生产、生活正常运转的应用型本科人才。计算机应用型本科人才有较强的技术思维能力和技术应用能力,是现代计算机软、硬件技术的应用者、实施者、实现者和组织者。应用型本科教育强调理论知识和实践知识并重,相应地,其教材更强调“用、新、精、适”。所谓“用”,是指教材的“可用性”、“实用性”和“易用性”,即教材内容要反映本学科基本原理、思想、技术和方法在相关现实领域的典型应用,介绍应用的具体环境、条件、方法和效果,培养学生根据现实问题选择合适的科学思想、理论、技术和方法去分析、解决实际问题的能力。所谓“新”,是指教材内容应及时反映本学科的最新发展和最新技术成就,以及这些新知识和新成就在行业、生产、管理、服务等方面的最新应用,从而有效地保证学生“学以致用”。所谓“精”,不是一般意义的“少而精”。事实常常告诉我们“少”与“精”是有矛盾的,数量的减少并不能直接促使质量的提高。而且,“精”又是对“宽与厚”的直接“背叛”。

因此,教材要做到“精”,教材的编写者要在“用”和“新”的基础上对教材的内容进行去伪存真的精练工作,精选学生终身受益的基础知识和基本技能,力求把含金量最高的知识传承给学生。“精”是最难掌握的原则,是对编写者能力和智慧的考验。所谓“适”,是指各部分内容的知识深度、难度和知识量要适合应用型本科的教育层次,适合培养目标的既定方向,适合应用型本科学生的理解程度和接受能力。教材文字叙述应贯彻启发式、深入浅出、理论联系实际、适合教学实践,使学生能够形成对专业知识的整体认识。以上 4 个方面不是孤立的,而是相互依存的,并具有某种优先顺序。“用”是教材建设的唯一目的和出发点,“用”是“新”、“精”、“适”的最后归宿。“精”是“用”和“新”的进一步升华。“适”是教材与计算机应用型本科培养目标符合度的检验,是教材与计算机应用型本科人才培养规格适应度的检验。

中国铁道出版社同高等学校计算机类课程应用型人才培养规划教材编审委员会经过近两年的前期调研,专门为应用型本科计算机专业学生策划出版了理论深入、内容充实、材料新颖、范围较广、叙述简洁、条理清晰的系列教材。本系列教材在以往教材的基础上大胆创新,在内容编排上努力将理论与实践相结合,尽可能反映计算机专业的最新发展;在内容表达上力求由浅入深、通俗易懂;编写的内容主要包括计算机专业基础课和计算机专业课;在内容和形式体例上力求科学、合理、严密和完整,具有较强的系统性和实用性。

本系列教材是针对应用型本科层次的计算机专业编写的,是作者在教学层次上采纳了众多教学理论和实践的经验及总结,不但适合计算机等专业本科生使用,也可供从事 IT 行业或有关科学研究工作的人员参考,适合对该新领域感兴趣的读者阅读。

本系列教材出版过程中得到了计算机界很多院士和专家的支持和指导,中国铁道出版社多位编辑为本系列教材的出版做出了很大贡献,在此表示感谢。本系列教材的完成不但依靠了全体作者的共同努力,同时也参考了许多中外有关研究者的文献和著作,在此一并致谢。

应用型本科是一个日新月异的领域,许多问题尚在发展和探讨之中,观点的不同、体系的差异在所难免,本系列教材如有不当之处,恳请专家及读者批评指正。

“高等学校计算机类课程应用型人才培养规划教材”编审委员会

2011 年 1 月

编译程序是计算机系统的核心系统软件之一，是高级程序设计语言的支撑环境。编译原理和技术为人们理解计算机程序语言、构建优秀软件提供了理论基础，拓宽了视野，加深了对高级语言的理解程度。随着各种高级语言的相继出现，支撑这些语言的编译方法和技术也在不断发展。编译的原理和技术也可以应用在诸如软件建模语言、硬件描述语言、脚本语言等的翻译上。在当今的信息化社会里，嵌入式系统的应用开发越来越重要，而其编译技术的研究和应用也成为计算机科学技术的一个重要方面。编译理论的研究有力地推动了计算机科学、计算机工程、软件开发以及人机工程等领域的研究和发展。

本书主要介绍计算机程序语言编译程序的基本原理、设计方法和实现技术，其内容主要针对应用型本科高等院校计算机专业的学生，亦可作为其他高等学校计算机学科及相关专业的教学参考书或教材。

编者在本书的编写过程中，力求保持较强的系统性，基本概念的阐述力求严谨和清晰，叙述力求通俗易懂，增强教材的可读性和易读性。本书的总体结构和内容具有以下特点：

(1) 在内容上增加了嵌入式系统编译技术的内容，介绍了嵌入式系统编译器及其工具的基本概念和应用开发，以利于读者了解编译原理和技术的发展动态和新的应用领域。编者还通过实例描述了嵌入式系统 C 编译器开发工具的工作过程，以及嵌入式系统 Java 编译器及开发工具的应用。

(2) 简要介绍了面向对象语言一些特性的处理方法，探讨了并行编译理论及实践。针对面向对象语言的语法结构及其语义的特征，着重从类及类的继承结构上讨论了相关的编译技术；用面向对象技术构造编译程序的思路，便于读者学习和理解面向对象的软件体系结构。

(3) 本书注重理论与实践相结合，每章都提供了适量的习题，便于读者理解基本概念和原理，掌握编译的基本算法和实现技术。本书还配备了教学辅助课件及书中所有程序示例软件，读者可到中国铁道出版社教材研究开发中心网站下载（网址：<http://www.edusources.net>）。

全书共分 10 章：

第 1 章概述了编译程序的功能、结构、工作过程、组织方式等方面的基本知识，其他各章按照编译程序的构成和编译过程的顺序，依次介绍编译的基本原理、设计方法和构造技术。

第 2 章介绍形式语言识别的理论基础、文法和语言的一些基本概念。

第 3 章首先介绍词法分析的设计和词法分析程序的手工构造，然后讲述自动生成词法分析程序的理论基础——有限状态自动机的理论。

第 4 章介绍语法分析的方法，包括自底向上和自顶向下语法分析方法。

第 5 章对语义描述技术、属性文法以及以语法制导的语义分析进行了详尽的阐述，讨论了基于语法制导技术的中间代码翻译。

第 6 章讨论了符号表的组织和访问符号表的各种方法、运行时的存储组织和运行时的分配策略。

第 7 章介绍代码优化技术，包括中间代码的局部优化和目标代码的优化方法。

第 8 章论述了目标代码生成时涉及的知识和技术。

第 9 章论述面向对象语言编译及并行编译基础。

第 10 章介绍嵌入式系统编译器及其工具的应用开发。

使用本书要求读者已经学习和掌握了高级程序设计语言，如 C、Java，最好具备离散数学、数据结构、计算机组成原理和汇编语言的基本知识。

本书的第 6~8 章由刘升编写；其余各章由游晓明编写，并负责全书的统稿工作。

本书由陈涵生教授主审，他给本书提出了许多宝贵的修改意见，使本书更能体现应用型本科教育“一个目标”（培养计算机应用一线工作的本科人才）、“四个能力”（自我学习能力、创新能力、专业应用能力、社会适应能力）、“三种知识”（专业基础知识、专业技能知识、通用职业知识）的基本特征。周广声教授和周海燕编辑为本书的编审付出了辛勤的劳动，在此表示衷心的感谢。在本书的编写过程中，还得到了王裕明老师和张瑜老师的大力支持，并引用了一些学者的论著和研究成果，以及部分公司的产品介绍。对此，编者向所有关心、支持本书的老师和社会同仁表示诚挚的感谢。

由于编者水平所限，书中难免有疏漏或不足之处，恳请广大读者提出宝贵意见。

编 者

2011 年 8 月

第 1 章 编译概述	1
1.1 程序设计语言	2
1.2 翻译程序	3
1.3 编译程序的逻辑结构	5
1.3.1 词法分析	5
1.3.2 语法分析	6
1.3.3 语义分析及中间代码生成	6
1.3.4 代码优化	7
1.3.5 目标代码生成	7
1.3.6 符号表管理程序	8
1.3.7 错误检查和处理程序	8
1.4 编译程序的组织方式	9
1.4.1 单遍编译程序	9
1.4.2 多遍编译程序	9
1.5 编译程序的前后处理器	10
1.6 编译程序的生成	11
小结	12
习题一	12
第 2 章 文法和语言	13
2.1 程序设计语言的定义	14
2.1.1 语法	14
2.1.2 语义	14
2.2 基本概念和术语	15
2.2.1 字母表	15
2.2.2 符号串	15
2.2.3 符号串及其集合的运算	15
2.3 文法形式定义	17
2.4 语言和语法树	19
2.4.1 直接推导定义	19
2.4.2 推导定义	19
2.4.3 规范推导	20
2.4.4 句型、句子和语言	20
2.4.5 语法树和二义性	23
2.5 短语、简单短语和句柄	25
2.6 文法的化简与改造	27

2 编译原理基础与应用

2.6.1 无用符号和无用产生式的删除	27
2.6.2 ε 产生式的消除	28
2.6.3 单产生式的消除	31
2.7 文法和语言的乔姆斯基分类	31
2.8 PL/0 编译程序概述	34
2.8.1 PL/0 语言的功能	34
2.8.2 PL/0 程序示例	35
2.8.3 PL/0 语言的语法描述	35
2.8.4 PL/0 编译程序的结构	38
小结	39
习题二	39
第3章 词法分析	41
3.1 词法分析器的功能	42
3.2 单词符号	42
3.3 正规文法及状态转换图	44
3.3.1 正规文法	44
3.3.2 由正规文法构造状态转换图	49
3.4 有限状态自动机	52
3.4.1 确定的有限状态自动机	52
3.4.2 不确定的有限状态自动机	56
3.4.3 NFA 与 DFA 的转化	57
3.4.4 正规表达式与有限状态自动机的等价性	59
3.4.5 确定的有限状态自动机的化简	61
3.5 词法分析程序的设计与实现	63
3.5.1 词法分析程序的手工编写	63
3.5.2 词法分析程序的自动生成	66
3.6 PL/0 编译程序的词法分析	70
小结	72
习题三	72
第4章 语法分析	74
4.1 语法分析器的功能	75
4.2 自顶向下分析	75
4.2.1 自顶向下分析面临的问题	75
4.2.2 消除文法的左递归	78
4.2.3 消除左递归算法	79
4.2.4 回溯的消除及 LL(1) 文法	79
4.2.5 LL(1) 文法的判断条件	80
4.2.6 FIRST 集合和 FOLLOW 集合及构造方法	81
4.2.7 LL(1) 分析的基本方法	83
4.2.8 预测分析表的构造方法	83

4.2.9 预测分析控制程序的实现技术.....	84
4.2.10 递归下降分析.....	86
4.3 自底向上分析.....	92
4.3.1 规范推导、规范句型和规范归约.....	92
4.3.2 自底向上分析方法的一般过程.....	94
4.3.3 算符优先分析.....	95
4.3.4 LR 分析方法.....	106
4.3.5 LR 分析表的构造.....	110
4.3.6 SLR(1) 分析器.....	117
4.3.7 LR(1) 分析器.....	121
4.3.8 LALR(1) 分析器.....	122
4.4 语法分析程序的自动生成.....	124
4.4.1 语法分析程序的自动生成工具 YACC.....	124
4.4.2 PL/0 编译程序的语法分析.....	130
小结.....	132
习题四.....	132
第 5 章 语法制导翻译和中间代码生成.....	135
5.1 语法制导翻译的基本思想.....	136
5.2 属性文法综述.....	139
5.2.1 文法的属性.....	139
5.2.2 属性文法.....	139
5.3 语法制导翻译的实现方法.....	141
5.4 中间代码的形式.....	142
5.4.1 树形表示法.....	142
5.4.2 逆波兰表示法.....	143
5.4.3 三元式.....	144
5.4.4 四元式.....	145
5.5 语句的翻译.....	146
5.5.1 说明语句(简单变量)的翻译.....	146
5.5.2 赋值语句的翻译.....	147
5.5.3 布尔表达式的翻译.....	151
5.5.4 标号和无条件转移语句的翻译.....	153
5.5.5 控制语句的翻译.....	155
5.5.6 复合语句的翻译.....	156
5.5.7 多分支控制语句 case 的翻译.....	158
5.5.8 数组元素的翻译.....	158
5.5.9 过程或函数调用语句的翻译.....	159
5.5.10 数组说明的翻译.....	160
小结.....	160
习题五.....	160

4 编译原理基础与应用

第 6 章 符号表与运行环境	162
6.1 符号表的组织和作用	162
6.1.1 符号表的作用和地位	163
6.1.2 符号表的内容	163
6.1.3 符号表的组织方式	163
6.2 存储组织	167
6.2.1 目标程序运行时的组织	167
6.2.2 运行时存储器的划分	167
6.2.3 活动记录	168
6.3 运行时的分配策略	169
6.3.1 静态存储分配	170
6.3.2 栈式分配	173
6.3.3 堆式动态存储分配	180
小结	182
习题六	182
第 7 章 代码优化	183
7.1 优化概述	183
7.2 局部优化	185
7.2.1 基本块的划分方法	186
7.2.2 利用 DAG 进行基本块的优化	188
7.3 循环优化	192
7.3.1 程序流图与循环	192
7.3.2 循环的查找	196
7.3.3 循环优化的实现	197
小结	207
习题七	207
第 8 章 目标代码生成	209
8.1 代码生成器	209
8.2 一个简单的目标代码生成器	211
8.2.1 待用信息与活跃信息	212
8.2.2 源程序到目标代码生成示例	217
8.3 代码生成器的开发方法	219
小结	220
习题八	220
第 9 章 面向对象语言编译及并行编译基础	222
9.1 面向对象语言的概念	223
9.2 方法的编译	224
9.3 继承的编译方案	226
9.3.1 单一继承的编译方案	226

9.3.2 多重继承的编译方案	227
9.4 并行编译系统	228
9.4.1 并行编译系统的结构	229
9.4.2 并行编译方法	231
9.4.3 向量计算机	231
9.4.4 共享存储器多处理机	232
9.4.5 分布存储器大规模并行计算机	233
9.5 并行编译基础	233
9.5.1 向量与向量的次序	233
9.5.2 循环模型与索引空间	234
9.5.3 输入与输出集合	234
9.5.4 语句的执行顺序	234
9.5.5 依赖关系定义	235
9.6 循环的向量化与并行化	236
9.7 循环变换技术	236
小结	237
习题九	237
第 10 章 嵌入式系统编译器及其工具的应用开发	239
10.1 嵌入式系统概述	240
10.2 嵌入式系统常用芯片介绍	242
10.3 嵌入式系统结构、开发工具和编译器	244
10.3.1 嵌入式系统的体系结构	244
10.3.2 嵌入式系统开发工具和编译器	248
10.4 嵌入式系统 C 编译器开发和相应开发工具的应用	249
10.5 嵌入式系统 Java 编译器开发、JDK 结构及开发工具的应用	252
10.5.1 JIT 编译方法	253
10.5.2 AOT 编译方法	256
10.5.3 JDK 结构及开发工具的应用	258
小结	260
习题十	261
参考文献	262

第1章 编译概述

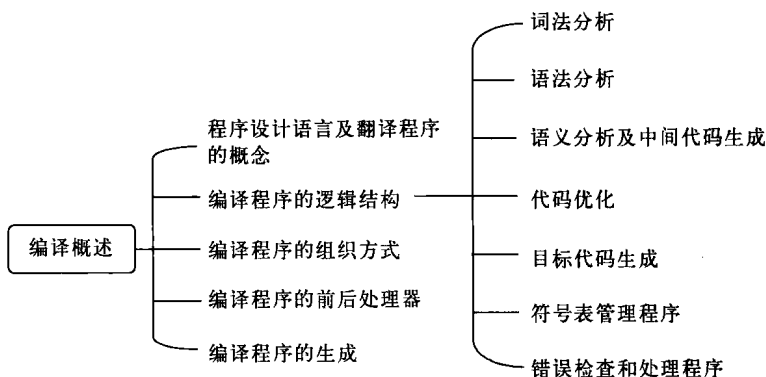
本章导读

本章作为学习编译原理的基础，给出编译程序中所涉及的一些基本概念，阐述编译过程和编译程序的基本结构，最后介绍编译程序的生成过程和构造工具。

本章内容要点：

- 程序设计语言及翻译程序的概念；
- 编译程序的逻辑结构；
- 编译程序的组织方式；
- 编译程序的生成过程和构造工具。

内容结构



学习目标

本章内容一般只要求作认识性了解，通过学习，学生应该能够：

- 正确了解什么是编译程序，了解编译程序工作的基本过程和基本任务，熟悉编译程序的总体框架。
- 为今后的深入学习和研究打下基础。

1.1 程序设计语言

计算机无法直接执行高级语言，必须为计算机构造编译程序，将高级语言翻译成计算机能够识别的机器语言，然后由计算机去执行。因此，编译技术是计算机专业人员必须具备的专业基础知识，它涉及程序设计语言、形式语言与自动机、计算机体系结构、数据结构、算法分析与设计、操作系统及软件工程等各个方面。

在计算机的应用领域，程序设计语言充当了人与计算机之间的通信工具。程序设计语言是一套系统化的记法，用于描述计算或非计算过程。通常可分为两大类，即低级语言和高级语言。

低级语言可分为机器语言和汇编语言。在计算机发展的初期，人们直接使用机器语言编写程序。机器语言由二进制数字 0 和 1 表示的机器指令组成，很不直观，而且难写、难读、难记，容易出错，调试极不方便，程序员在程序设计及检查错误方面要花很大的精力；由于不同类型的计算机使用不同的机器指令，程序员必须针对某种类型的机器编程，编写的程序不适于移植，因此限制了计算机的推广与使用。

为了便于记忆、阅读和检查，人们用较直观的符号来代替机器指令，进一步发展成为汇编语言。汇编语言采用比较直观且具有含义的指令助记符表示每条机器指令，同时为了方便编程，还提供了若干宏指令对应一组机器指令，从而完成一些特定的功能。但是，汇编指令依赖于机器，对问题的描述处于低层次，没有高级语言中的条件、循环等控制结构，编程人员必须考虑寄存器和内存的分配，使用仍然不方便，程序设计的效率依然很低。

高级语言可分为过程式语言、函数式语言、逻辑式语言和对象式语言等。这类语言完全摆脱了机器指令的约束，用它编写的程序接近自然语言和习惯上对算法的描述，故称为面向用户的语言或面向过程的语言。后来，又相继出现许多专门用于某个应用领域问题的专业语言。例如，用于数据库操作的 SQL (Structured Query Language, 结构化查询语言)，这类语言称为面向问题的语言。

高级语言与低级语言相比具有以下优点：

(1) 高级程序设计语言不依赖于具体的机器，在一种类型的机器上编写的程序不做很大改动就能在其他机器上运行，故移植性好。

(2) 编写高级语言程序时，不用考虑具体的寄存器和内存的分配，也不必了解机器的硬件，故易于使用和维护。

(3) 每条高级语言语句对应于多条汇编指令或机器指令，故编程效率高。

(4) 高级语言提供了丰富的数据结构和控制结构，提高了问题的表达能力，因此降低了程序的复杂性。

(5) 高级语言接近于自然语言，编程更加容易，编写出的程序有良好的可读性，因此便于交流和维护。

尽管高级语言有这么多优点，然而，就目前的情况而言，计算机硬件只能识别机器语言，而不能直接执行用高级语言或汇编语言编写的程序。因此，除机器语言外，用其他语言编写的程序都必须经过翻译才能被计算机识别，这一过程由翻译程序来完成。

1.2 翻译程序

所谓翻译程序是指能把某一种语言程序（称为源语言程序）翻译成与之等价的另一种语言程序（称为目标语言程序）的程序；如果源语言是诸如 FORTRAN、Pascal、C、Ada、Smalltalk 或 Java 这样的“高级语言”，而目标语言是诸如汇编语言或机器语言之类的“低级语言”，这样的—个翻译程序就称为编译程序。

翻译程序的功能如图 1-1 所示。

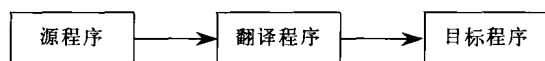


图 1-1 翻译程序的功能

翻译程序的源程序分高级语言源程序和汇编语言源程序两种。

(1) 如果要翻译的源程序是用汇编语言编写的，而目标语言是机器语言，则翻译程序称为汇编程序。

(2) 如果要翻译的源程序是用高级语言编写的，其翻译后的目标程序是某种具体机器的机器语言或汇编语言，那么这种翻译程序称为编译程序。

根据不同的用途和侧重点，编译程序还可进一步分类。专门用于帮助程序开发和调试的编译程序称为诊断编译程序 (diagnostic compiler)，着重于提高目标代码效率的编译程序称为优化编译程序 (optimizing compiler)。现在很多编译程序同时提供了调试、优化等多种功能，用户可以通过“开关”进行选择。运行编译程序的计算机称宿主机，运行编译程序所产生目标代码的计算机称目标机。如果一个编译程序产生不同于其宿主机的机器代码，则称它为交叉编译程序 (cross compiler)。如果不需重写编译程序中与机器无关的部分就能改变目标机，则称该编译程序为可变目标编译程序 (retargetable compiler)。

世界上第一个编译程序——FORTRAN 编译程序是 20 世纪 50 年代中期研制成功的。当时，人们普遍认为设计和实现编译程序是一件十分困难、令人生畏的事情。经过 50 多年的努力，编译理论与技术得到迅速发展，现在已形成了一套比较成熟的、系统化的理论与方法，并且开发出了一些好的编译程序的实现语言、环境与工具，在此基础上设计并实现一个编译程序不再是高不可攀的事情。

若编译程序的目标程序是机器语言，则源程序从编译到被执行的过程如图 1-2 所示。

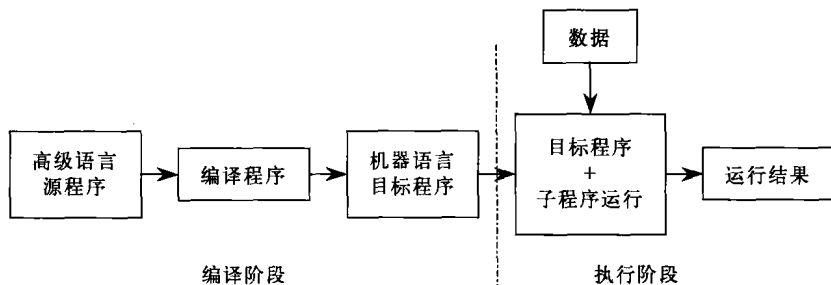


图 1-2 生成机器语言目标程序的编译方式

4 编译原理基础与应用

如果编译程序翻译得到的目标程序是汇编语言程序,则还要经过汇编程序翻译成机器语言程序,这种编译方式的源程序从编译到被执行的过程如图 1-3 所示。

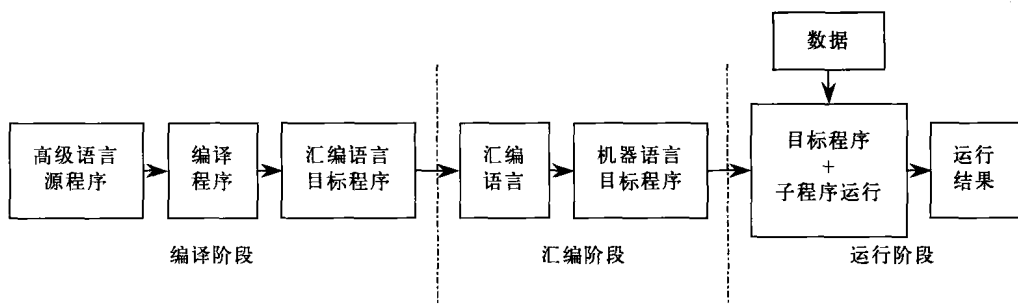


图 1-3 生成汇编语言目标程序的编译方式

总之,编译方式是一种分阶段进行的方式,一般需经过两个阶段:第一阶段称为编译阶段,其任务是由编译程序将源程序编译为目标程序;第二阶段称为运行阶段,其任务是在目标计算机上执行编译阶段所得到的目标程序。在执行目标程序时,一般还应有一些子程序配合进行工作,如数据空间分配子程序、标准函数子程序等,这些子程序组成一个子程序库,称为运行系统。通常所说的编译系统实际上包含编译程序和相应的运行系统,在执行时,程序的初始数据作为目标程序和运行系统的输入,处理后输出相应的运行结果。

还有一种高级语言翻译程序,从源程序的编译到执行只有一个阶段——解释执行阶段,它同时处理源程序和数据,按源程序中语句的动态顺序逐句翻译成可执行代码,一旦具备执行条件(如获得必要的初始数据等),则立即执行这一段代码得到部分结果。这种翻译程序称为解释程序。

解释执行的过程如图 1-4 所示。

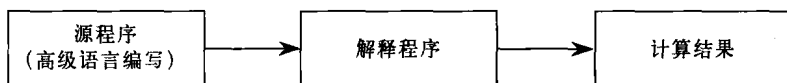


图 1-4 程序的解释执行

解释执行的工作效率很低,但由于解释程序的结构比编译程序简单,占用内存较少,在执行过程中也易于对源程序进行修改,因此一些规模较小的语言常采用解释执行方式,例如 BASIC 语言。然而就目前的情况来看,纯粹的解释程序并不多见,通常是将编译程序和解释程序结合起来。例如 Visual Basic,调试期间可解释执行源程序,而调试好的程序可以编译生成目标程序。

在解释执行方式下并不生成目标代码,而是直接执行源程序本身,这是编译方式与解释方式的根本区别。解释过程类似于自然语言中的口译,随时进行翻译。而编译过程类似于自然语言中的笔译,一次翻译可多次阅读。

对高级语言来说,其编译程序再加上一些相应的支持用户程序运行的子程序就构成了该语言的编译系统。编译系统是计算机的重要组成部分。本课程的目的向读者介绍构造编译系统的原理、技术及其实现方法。