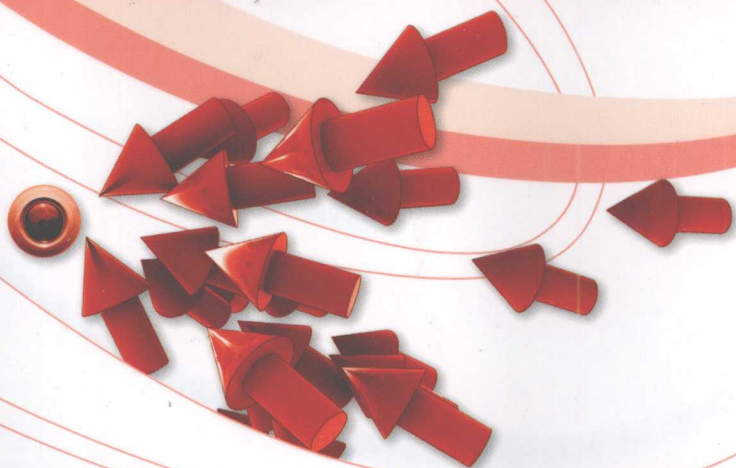


提供实验源代码、自测题及习题参考答案、电子课件



高等学校计算机规划教材

编译原理(第3版)



NLIC 2970661697

■ 刘 铭 徐兰芳 骆 婷 编著



电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY

<http://www.phei.com.cn>

高等学校计算机规划教材

编译原理

(第3版)

刘 铭 徐兰芳 骆 婷 编著



NLIC 2970661697

电子工业出版社

Publishing House of Electronics Industry

北京·BEIJING

内 容 简 介

本书根据高等学校“编译原理”课程教学基本要求编写。全书系统介绍了编译程序的一般构造原理、基本设计方法和主要实现技术。内容包括:文法和语言基本知识、词法分析程序的设计原理与构造方法、各种语法分析技术、语法制导翻译技术与中间代码生成、符号表的组织和管理、代码优化、运行时存储空间的组织与管理、目标代码生成、并行编译技术基本常识等。

本书系统性强,概念清晰,内容简明通俗,每章配有本章学习导读、本章小结、自测练习题和习题。附录给出了自测练习题与习题参考答案及编译程序实验,本书还免费提供电子课件和实验源代码。

本书可作为高等学校计算机专业本科生教材,也可作为成人教育本科和专升本学生的教材,对相关工程技术人员也有参考价值。

未经许可,不得以任何方式复制或抄袭本书之部分或全部内容。
版权所有,侵权必究。

图书在版编目(CIP)数据

编译原理 / 刘铭, 徐兰芳, 骆婷编著. —3版. —北京: 电子工业出版社, 2011.3
高等学校计算机规划教材
ISBN 978-7-121-12938-4

I. 编… II. ①刘… ②徐… ③骆… 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 024619 号

策划编辑: 童占梅

责任编辑: 童占梅

印 刷: 北京京师印务有限公司

装 订:

出版发行: 电子工业出版社

北京市海淀区万寿路 173 信箱 邮编 100036

开 本: 787×1092 1/16 印张: 16.5 字数: 420 千字

印 次: 2011 年 3 月第 1 次印刷

印 数: 5000 册 定价: 29.00 元

凡所购买电子工业出版社图书有缺损问题, 请向购买书店调换。若书店售缺, 请与本社发行部联系, 联系及邮购电话: (010)88254888。

质量投诉请发邮件至 zllts@phei.com.cn, 盗版侵权举报请发邮件至 dbqq@phei.com.cn。

服务热线: (010)88258888。

前 言

21 世纪是信息时代,计算机技术已经渗透到人类活动的方方面面,因此各行各业对计算机应用人才的需求越来越大,为了适应计算机教育迅猛发展的需要,编者根据多年来讲授“编译原理”课程的教学经验,编写了适合高校计算机专业教学的编译原理教材。在广大读者及电子工业出版社的大力支持下,本书第 2 版自出版至今已印刷 12 次发行近 6 万册,使得本书有机会修订再版。

“编译原理”是一门研究设计和构造编译程序原理和方法的课程,是计算机各专业的一个重要专业基础课。编译原理这门课程实际蕴含着计算机学科中解决问题的思路、形式化问题和解决问题的方法,这些思路和方法除了对应用软件和系统软件的设计与开发有一定的启发和指导作用外,在提倡学科交叉不断创新的时代背景下,仍然具有非常重要的意义。编译程序构造的原理和技术在软件工程、逆向工程、软件再工程、语言转换及其他领域中都有着广泛的应用。本书主要介绍设计和构造编译程序的一般原理、基本方法和主要实现技术。通过本课程的学习,使学生掌握编译系统的结构、工作流程及编译程序各组成部分的设计原理和常用的编译技术和方法,为今后从事应用软件和系统软件的开发打下一定的理论和实践基础。

在本书编写过程中,充分考虑本课程的特点,力求将基本概念、基本原理和实现方法的思路阐述清楚,条理清晰,通俗易懂,便于自学。为了帮助学生掌握每章的重点和难点,每章附有本章学习导读、本章小结、扩展阅读、自测练习题和习题,书末附录中提供自测练习题和习题参考答案。

本教材参考学时数为 60~70 学时,书中主要算法、例题、习题均以 C 语言为背景,其主要内容包括:编译程序结构及各部分功能、文法和语言的基本概念和表示、词法分析、语法分析、属性文法与语法制导翻译技术、符号表、运行时存储空间的组织、代码优化与目标代码生成、并行编译技术概述等。

“编译原理”是一门实践性较强的课程,为了理论联系实际,本书在附录 C 中拟定了实验内容、要求及实验参考算法,并给出了相应的编译实验程序的 C 语言框架。

为了方便教学,本书提供电子课件及实验源代码,欢迎任课老师登录华信教育资源网 <http://www.hxedu.com.cn> 免费注册下载。

这次修订,除保持第 2 版简明、实用的风格并在文字上作进一步润饰之外,我们作了下面的修改:

- (1) 对教学过程中发现的印刷错误、文字错误、个别内容上的错误作出了相应的修改。
- (2) 在 1~9 章的后面增加了扩展阅读。扩充了知识面;使学生能全面了解编译原理发展、研究的情况,以开阔眼界。
- (3) 增加了 4.5.7 节(LR 语法分析中的错误恢复技术)及 5.5.7 节(过程和函数调用语句的翻译)等内容,使体系更丰富、实用。
- (4) 为切实做到理论联系实际,便于读者更深刻地理解编译程序的实现过程,对附录 C 的词法分析实验部分做了进一步的完善。另外增加了实验 C.6(正规式转换成自动机的图形表

示),体现编译原理技术的实际用途。

以学习本课程总学时为 64 学时(52 学时上课+12 学时上机)为例,建议各章的学时分配如下表所示。

章次	内容	讲课学时	习题课学时	上机学时
第 1 章	编译概述	2		
第 2 章	文法和语言的基本知识	6	1	
第 3 章	词法分析与有穷自动机	7	1	4
第 4 章	语法分析	15	1	4
第 5 章	语法制导翻译技术和中间代码生成	7	1	4
第 6 章	符号表的组织与管理	1		
第 7 章	代码优化	4		
第 8 章	运行时的存储组织与管理	2		
第 9 章	目标代码生成	2		
第 10 章	并行编译技术基本常识	2		
合计		48	4	12

受第 2 版作者的委托,参加本次修订工作的有刘铭、徐兰芳和骆婷。在本书再版过程中,特别感谢胡伦骏老师的支持;华中科技大学计算机学院卢炎生教授仔细审阅了书稿;还得到了华中科技大学计算机学院领导的关心和支持,感谢徐丽萍、周时阳、胡福林、祝建华等老师的宝贵意见;电子工业出版社的领导和有关同志为本书的编辑再版付出了辛勤的劳动,在此对他们表示衷心的感谢。在成书过程中,编者参考了书末所列出的有关文献,在此,也向这些书籍的作者一并表示感谢。

由于编者水平有限,书中难免存在疏忽和错误,恳请读者批评指正。

编者

于华中科技大学计算机学院

目 录

第 1 章 编译概述	1
1.1 翻译程序与编译程序	1
1.2 编译过程和编译程序的基本结构	2
1.3 编译程序的生成方法	5
1.4 编译技术在软件开发中的应用	6
本章小结	6
扩展阅读	6
自测练习题 1	7
习题 1	7
第 2 章 文法和语言的基本知识	8
2.1 概述	8
2.2 字母表和符号串的基本概念	8
2.2.1 字母表和符号串	8
2.2.2 符号串的运算	9
2.3 文法和语言的形式定义	10
2.3.1 形式语言	10
2.3.2 文法的形式定义	11
2.3.3 语言的形式定义	14
2.3.4 规范推导和规范归约	16
2.3.5 递归规则与文法的递归性	17
2.4 短语、直接短语和句柄	18
2.4.1 短语和直接短语	18
2.4.2 句柄	19
2.5 语法树与文法的二义性	20
2.5.1 推导和语法树	20
2.5.2 文法的二义性	22
2.5.3 文法二义性的消除	22
2.6 文法和语言的分类	24
2.7 有关文法的实用限制和变换	26
本章小结	27
扩展阅读	27
自测练习题 2	28
习题 2	30
第 3 章 词法分析与有穷自动机	32
3.1 词法分析程序的功能	32
3.2 单词符号及输出单词的形式	32
3.2.1 语言的单词符号	33
3.2.2 词法分析程序输出单词的形式	33

3.3 语言单词符号的两种定义方式	34
3.3.1 正规式与正规集	34
3.3.2 正规文法与正规式	35
3.4 正规式与有穷自动机	38
3.4.1 确定有穷自动机(DFA)	38
3.4.2 非确定有穷自动机(NFA)	39
3.4.3 由正规表达式 R 构造 NFA	40
3.4.4 NFA 确定化为 DFA 的方法	41
3.4.5 DFA 的化简	44
3.4.6 有穷自动机到正规式的转换	46
3.5 正规文法与有穷自动机	47
3.5.1 右线性正规文法到有穷自动机的转换方法	47
3.5.2 左线性正规文法到有穷自动机的转换方法	48
3.5.3 有穷自动机到正规文法的转换方法	48
3.6 词法分析程序的编写方法	49
本章小结	54
扩展阅读	55
自测练习题 3	55
习题 3	57
第 4 章 语法分析	59
4.1 语法分析程序的功能	59
4.2 自上而下分析法	60
4.2.1 非确定的自上而下分析法的思想	60
4.2.2 文法的左递归性和回溯的消除	61
4.2.3 某些非 LL(1)文法到 LL(1)文法的改写	64
4.2.4 递归下降分析法	66
4.2.5 预测分析法与预测分析表的构造	68
4.3 自下而上分析法的一般原理	70
4.4 算符优先分析法	71
4.4.1 方法概述	71
4.4.2 算符优先文法的定义	72
4.4.3 算符优先关系表的构造	73
4.4.4 算符优先分析算法的设计	74
4.4.5 优先函数的构造	77
4.4.6 算符优先分析法的局限性	79
4.5 LR 分析法	79
4.5.1 LR 分析器的工作原理和过程	79
4.5.2 LR(0)分析法	82
4.5.3 SLR(1)分析法	86
4.5.4 LR(1)分析法	90
4.5.5 LALR(1)分析法	93
4.5.6 LR 分析法对二义性文法的应用	96
4.5.7 LR 语法分析中的错误恢复技术	97
本章小结	100

扩展阅读	102
自测练习题 4	102
习题 4	104
第 5 章 语法制导翻译技术和中间代码生成	107
5.1 概述	107
5.2 属性文法	107
5.3 语法制导翻译概述	109
5.4 中间语言	111
5.4.1 逆波兰式	111
5.4.2 三元式和树形表示	112
5.4.3 四元式和三地址代码	114
5.5 自下而上语法制导翻译	114
5.5.1 简单算术表达式和赋值语句的翻译	114
5.5.2 布尔表达式的翻译	116
5.5.3 控制语句的翻译	121
5.5.4 循环语句的翻译	124
5.5.5 简单说明语句的翻译	126
5.5.6 含数组元素的赋值语句的翻译	126
5.5.7 过程和函数调用语句的翻译	130
5.6 递归下降语法制导的翻译	131
本章小结	133
扩展阅读	133
自测练习题 5	133
习题 5	134
第 6 章 符号表的组织与管理	136
6.1 符号表的作用	136
6.2 符号表的组织	138
6.3 符号表的建立和查找	141
本章小结	143
扩展阅读	144
自测练习题 6	144
习题 6	144
第 7 章 代码优化	145
7.1 优化概述	145
7.2 局部优化	149
7.2.1 划分基本块的方法	149
7.2.2 基本块的 DAG 表示	149
7.2.3 利用 DAG 进行基本块的优化处理	153
7.3 循环优化	154
7.3.1 程序流图与循环	155
7.3.2 循环查找	156
7.3.3 循环优化	158
7.4 窥孔优化	162
本章小结	164

扩展阅读	164
自测练习题7	165
习题 7	165
第 8 章 运行时的存储组织与管理	167
8.1 概述	167
8.2 静态存储分配	168
8.3 栈式存储分配	169
8.3.1 简单栈式存储分配	169
8.3.2 嵌套过程的栈式存储分配	170
8.4 堆式存储分配	172
8.5 临时变量的存储分配	173
本章小结	173
扩展阅读	174
自测练习题8	174
习题 8	174
第 9 章 目标代码生成	175
9.1 概述	175
9.2 假想的计算机模型	175
9.3 简单代码生成器	176
9.3.1 待用信息与活跃信息	176
9.3.2 代码生成算法	177
9.3.3 寄存器的分配	179
9.4 代码生成器的自动生成技术	179
本章小结	179
扩展阅读	180
自测练习题9	180
习题 9	180
第 10 章 并行编译技术基本常识	181
10.1 并行编译技术的引入	181
10.2 并行编译系统的功能和结构	182
10.2.1 并行编译系统的功能	182
10.2.2 并行编译系统的结构	182
10.3 向量语言编译技术	183
10.3.1 向量语法处理	183
10.3.2 向量结构优化	184
10.4 共享存储器并行机并行编译技术	184
10.4.1 预编译	184
10.4.2 可再入的目标代码	184
本章小结	185
习题 10	185
附录 A 词法分析程序生成器 LEX	186
A.1 词法分析程序生成器 LEX 简介	186
A.2 LEX 输入文件的格式	186

A.3	正规表达式的 LEX 约定	187
A.4	LEX 源程序中的规则部分	188
A.5	FLEX 的命令选项	189
A.6	LEX 程序示例	190
附录 B	语法分析程序生成器 YACC	191
B.1	语法分析程序 YACC 简介	191
B.2	YACC 输入文件的格式	191
B.3	YACC 各部分的书写格式	192
B.3.1	定义部分	192
B.3.2	规则部分	194
B.3.3	辅助程序部分	196
B.4	YACC 的内置名称和定义机制	196
B.5	YACC 源程序示例	197
附录 C	编译程序实验	199
C.1	词法分析	199
C.1.1	实验目的	199
C.1.2	实验要求	199
C.1.3	词法分析程序的算法思想	200
C.1.4	词法分析程序的 C 语言程序框架	200
C.2	语法分析	206
C.2.1	实验目的	206
C.2.2	实验要求	206
C.2.3	语法分析程序的算法思想	206
C.2.4	语法分析程序的 C 语言程序框架	208
C.3	语义分析	209
C.3.1	实验目的	209
C.3.2	实验要求	209
C.3.3	语义分析程序的 C 语言程序框架	210
C.4	算符优先分析法	212
C.5	实验实例	213
C.6	正规式转换成自动机的图形表示	230
C.6.1	实验目的	230
C.6.2	实验要求	230
C.6.3	参考设计思路	230
C.6.4	参考算法	231
附录 D	自测练习题与习题参考答案	234
参考文献	254

第 1 章 编译概述



本章学习导读

编译程序是计算机系统中重要的系统软件,是高级语言的支撑基础。本章主要介绍编译程序的基本知识。具体包括下面 4 个方面的内容:

- (1) 编译程序
- (2) 编译过程
- (3) 编译程序的结构
- (4) 编译程序的生成方法

1.1 翻译程序与编译程序

语言是人与人之间传递信息的媒介和手段。世界上存在着多种语言,人们为了通信方便,建立了各种语言之间的翻译。人与计算机之间的信息交流,同样需要翻译。我们知道,每种计算机只懂得自己独特的指令系统,即它只能直接执行用机器语言编写的程序,这对人们来说很不方便,其原因是机器语言对计算机依赖性强、直观性差、编写程序工作量大、程序的结构也欠清晰。因此使用过现代计算机的人们多数都是用接近自然语言的高级程序设计语言来编写程序,但是计算机不能够直接接受和执行用高级语言编写的程序,需要通过一个翻译程序将它翻译成等价的机器语言程序才能执行。

所谓翻译程序是指这样一个程序,它把一种语言(称做源语言)所写的程序(源程序)翻译成与之等价的另一种语言(称做目标语言)的程序(目标程序),其功能如图 1.1 所示。

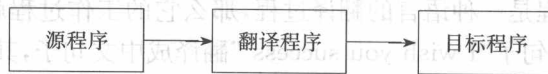


图 1.1 翻译程序功能

如果源语言是高级语言,如 Pascal, C, Ada, Java 语言等,目标语言是诸如汇编语言或机器语言之类的低级语言,那么称这样的翻译程序为编译程序,它所执行的转换工作如图 1.2 所示。

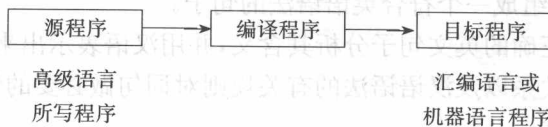


图 1.2 编译程序功能

可见,编译程序是一种翻译程序,它将高级语言所写的源程序翻译成等价的机器语言或汇编语言的目标程序。

采用编译方式在计算机上执行用高级语言编写的程序,需分阶段进行,一般分为两大阶

段,即编译阶段和运行阶段,其过程如图 1.3 所示。

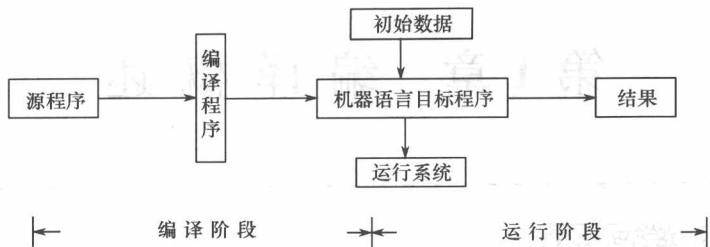


图 1.3 源程序的编译和运行阶段

如果编译阶段生成的目标程序不是机器语言程序,而是汇编语言程序,则程序的执行需分 3 个阶段,即编译阶段、汇编阶段和运行阶段,如图 1.4 所示。

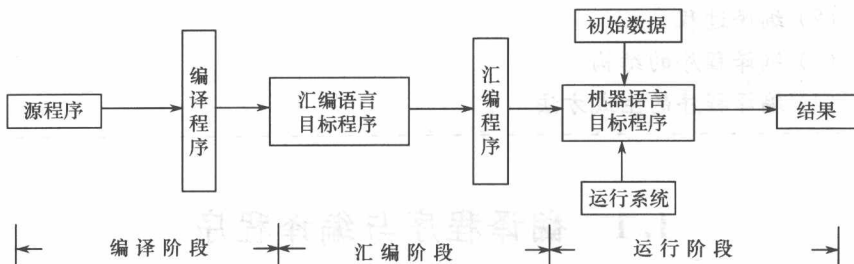


图 1.4 源程序的编译、汇编和运行阶段

用高级语言编写的程序也可通过解释程序来执行。解释程序也是一种翻译程序,它将源程序作为输入并执行之,即边解释边执行。它与编译程序的主要区别是在解释程序的执行过程中不产生目标程序,而是按照源语言的定义解释执行源程序本身。

本书主要介绍设计和构造编译程序的基本原理和方法。

1.2 编译过程和编译程序的基本结构

编译程序的功能是把用高级语言编写的源程序翻译成等价的机器语言或汇编语言表示的目标程序。既然编译过程是一种语言的翻译过程,那么它的工作过程就类似于外文的翻译过程。例如,我们要将英文句子“*I wish you success*”翻译成中文句子,其翻译的大致过程是:

(1) 词法分析。根据英语的词法规则,从由字母、空格字符和各种标点符号所组成的字符串中识别出一个一个的英文单词。

(2) 语法分析。根据英语的语法规则,对词法分析后的单词串进行分析、识别,并做语法正确性的检查,看其是否组成一个符合英语语法的句子。

(3) 语义分析。对正确的英文句子分析其含义,并用汉语表示出来。

(4) 根据上下文的关系以及汉语语法的有关规则对词句做必要的修饰工作。

(5) 最后翻译成中文。

类似地,编译程序是将一种语言形式翻译成另一种语言形式,因此其工作过程一般可划分为下列 5 个阶段:词法分析、语法分析、语义分析及中间代码生成、代码优化和目标代码生成。

下面,我们以一个简单的程序段为例,分别介绍这 5 个阶段所完成的任务。

例如,计算圆柱体表面积的程序段如下:

float r,h,s;

s = 2 * 3.1416 * r * (h+r)

第1阶段 词法分析

词法分析阶段的任务是对构成源程序的字符串从左到右进行扫描和分解,根据语言的词法规则,识别出一个一个具有独立意义的单词(也称单词符号,简称符号)。

词法规则是单词符号的形成规则,它规定了哪些字符串构成一个单词符号。上述源程序通过词法分析识别出如下单词符号:

基本字 float

标识符 r,h,s

常数 3.1416,2

运算符 *,+

界符 () ; , =

第2阶段 语法分析

语法分析的任务是在词法分析的基础上,根据语言的语法规则从单词符号串中识别出各种语法单位(如表达式、说明、语句等)并进行语法检查,即检查各种语法单位在语法结构上的正确性。

语言的语法规则规定了如何从单词符号形成语法单位,换言之,语法规则是语法单位的形成规则。

上述源程序,通过语法分解,根据语言的语法规则识别单词符号串 $s=2 * 3.1416 * r * (h+r)$,其中“s”是〈变量〉,单词符号串“ $2 * 3.1416 * r * (h+r)$ ”组合成〈表达式〉这样的语法单位,则由〈变量〉=〈表达式〉构成〈赋值语句〉这样的语法单位。在识别各类语法单位的同时进行语法检查,可以看到,上述源程序是一个语法上正确的程序。

第3阶段 语义分析及中间代码生成

定义一种语言除要求定义语法外,还要求定义其语义,即对语言的各种语法单位赋予具体的意义。语义分析的任务是首先对每种语法单位进行静态的语义审查,然后分析其含义,并用另一种语言形式(比源语言更接近于目标语言的一种中间代码或直接用目标语言)来描述这种语义。例如,上述源程序中,赋值语句的语义为:计算赋值号右边表达式的值,并把它送到赋值号左边的变量所确定的内存单元中。语义分析时,先检查赋值号右边表达式和左边变量的类型是否一致,然后再根据赋值语句的语义,对它进行翻译可得到如下形式的四元式中间代码:

(1) (*, 2, 3.1416, T_1)

(2) (*, T_1 , r, T_2)

(3) (+, h, r, T_3)

(4) (*, T_2 , T_3 , T_4)

(5) (=, T_4 , —, s)

其中, T_1, T_2, T_3, T_4 是编译程序引进的临时变量,存放每条指令的运算结果。上述每一个四元式所表示的语义为:

$2 * 3.1416 \Rightarrow T_1$

$T_1 * r \Rightarrow T_2$

$h + r \Rightarrow T_3$

$T_2 * T_3 \Rightarrow T_4$

$$T_4 \Rightarrow s$$

这样,我们将源语言形式的赋值语句翻译为四元式表示的另一种语言形式,这两种语言在结构形式上是不同的,但在语义上是等价的。

第4阶段 代码优化

代码优化的任务是对前阶段产生的中间代码进行等价变换或改造,以期获得更为高效的,即省时间和空间的目标代码。优化主要包括局部优化和循环优化等,例如上述四元式经局部优化后得:

- (1) $(*, 6.28, r, T_2)$
- (2) $(+, h, r, T_3)$
- (3) $(*, T_2, T_3, T_4)$
- (4) $(=, T_4, -, s)$

其中,2和3.1416两个运算对象都是编译时的已知量,在编译时就可计算出它的值6.28,而不必等到程序运行时再计算,即不必生成 $(*, 2, 3.1416, T_1)$ 的运算指令。

第5阶段 目标代码生成

目标代码生成的任务是将中间代码变换成特定机器上的绝对指令代码或可重定位的指令代码或汇编指令代码。

在编译程序的各个阶段中,都要涉及表格管理和错误处理。

编译程序的重要功能之一,是记录源程序中所使用的变量的名字,并且收集与名字属性相关的各种信息。名字属性包括一个名字的存储分配、类型、作用域等信息。如果名字是一个函数名,还会包括其参数数量、类型、参数的传递方式以及返回类型等信息。符号表数据结构可以为变量名字创建记录条目,来登记源程序中所提供的或在编译过程中所产生的这些信息,编译程序在工作过程的各个阶段需要构造、查找、修改或存取有关表格中的信息,因此在编译程序中必须有一组管理各种表格的程序。

如果编译程序只处理正确的程序,那么它的设计和实现将会大大简化。但是程序设计人员还期望编译程序能够帮助定位和跟踪错误。无论程序员如何努力,程序中难免总会有错误出现。虽然错误很常见,但很少有语言在设计的时候就考虑到错误处理问题。大部分程序设计语言的规范没有规定编译程序应该如何处理错误;错误处理方法由编译程序的设计者决定。因此,从一开始就计划好如何进行错误处理,不仅可以简化编译程序的结构,还可以改进错误处理方法。一个好的编译程序在编译过程中,应具有广泛的程序查错能力,并能准确地报告错误的种类及出错位置,以使用户查找和纠正,因此在编译程序中还必须有一个出错处理程序。

编译过程的这5个阶段的任务分别由5个程序完成,这5个程序分别称为词法分析程序、语法分析程序、语义分析及中间代码生成程序、代码优化程序和目标代码生成程序,另外再加上表格管理程序和出错处理程序。这些程序便是编译程序的主要组成部分,一个典型的编译程序结构如图1.5所示。

需要注意的是,图中所给出的各个阶段之间的关系是指它们之间的逻辑关系,不一定是执行时间上的先后关系。实际上,可按不同的执行流程来组织上述各阶段的工作,这在很大程度上依赖于编译过程中对源程序扫描的遍数以及如何划分各遍扫描所进行的工作。此处所说的“遍”,是指对源程序或其等价的中间语言程序从头到尾扫描一遍,并完成规定加工处理工作的过程。例如,可以将前述5个阶段的工作结合在一起,对源程序从头到尾扫描一遍来完成编译的各项工作,这种编译程序称为一遍扫描的编译程序。对于某些程序设计语言,用一遍扫描的

编译程序去实现比较困难,可采用多遍扫描的编译程序结构,每遍可完成上述某个阶段的一部分、全部或多个阶段的工作,且每一遍的工作是从前一遍获得的工作结果开始的,最后一遍的工作结果是目标语言程序,第一遍的输入则是用户书写的源程序。

多遍扫描的编译程序较一遍扫描的编译程序少占存储空间,遍数多一些,可使各遍所要完成的功能独立而单纯,其编译程序逻辑结构清晰,但遍数多势必增加输入输出开销,这将降低编译效率。一个编译程序究竟应分成几遍和它所面临的源语言的特征、机器规模、设计的目标等因素有关,很难统一划定。一般在主存可能的前提下,还是遍数少一点为好。

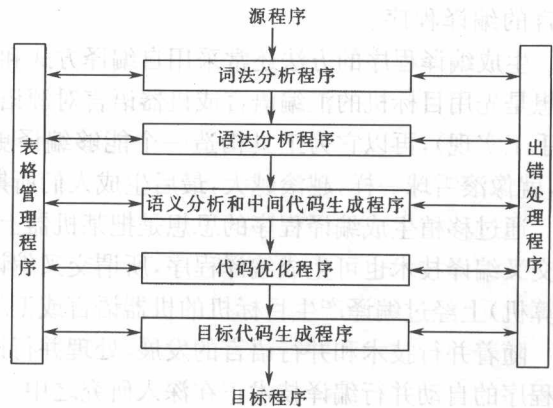


图 1.5 编译程序结构框图

1.3 编译程序的生成方法

编译程序是一个复杂的系统程序,要生成一个编译程序,一般要考虑下面几个方面。

1. 对源语言和目标语言认真分析

编译程序的功能是把某语言的源程序翻译成某台计算机上的目标程序。因此,我们首先要熟悉源语言(如 C 语言),要正确理解它的语法和语义;其次要搞清楚目标语言和目标机的性质,在此基础上,确定编译程序的结构和所采用的具体策略。

2. 设计编译算法

设计编译算法是构造编译程序过程中最关键的一步。把一种语言程序翻译成另一种语言程序的方法很多,但在算法设计中要着重考虑如何使编译程序具有易读性、易改性和易扩充性。

3. 选择语言编制程序

根据所设计的算法选用某种语言(如机器语言、汇编语言、Pascal 或 C 语言等)编写出编译程序。早期人们使用机器语言或汇编语言并用手工方式编写编译程序,虽然目标程序效率高但可靠性差,不便于阅读、修改和移植。从 20 世纪 80 年代开始,几乎所有编译程序都用高级语言来编写,这样可以提高开发效率,而且构造出来的编译程序增加了易读性、易修改性和可移植性。

4. 调试编译程序

通过大量实例对编写好的编译程序进行调试,调试过程中不断修改、完善编译程序。

5. 提交相关文档资料

为方便用户,需提交一份有关编译程序的文档资料,内容包括源语言的语法、目标机指令系统、编译程序结构和所采用的具体策略、错误信息表及使用说明等。

我们希望能有一个自动生成编译程序的软件工具,只要把源程序的定义以及机器语言的描述输入到这个软件工具中去,就能自动生成该语言的编译程序。

随着编译技术和自动机理论的发展,近年来已研制出了一些编译程序的自动生成系统。如目前已广泛使用的词法分析程序自动生成系统 LEX 和语法分析程序自动生成系统 YACC 等,此外,还有可用来自动产生整个编译程序的软件工具——编译程序产生器,

它的功能是将任一语言的词法规则、语法规则和语义解释的描述作为输入,自动生成该语言的编译程序。

生成编译程序的方法还常采用自编译方式和移植方式。采用自编译方式生成编译程序的思想是先用目标机的汇编语言或机器语言对源语言的核心部分构造一个小小的编译程序(可用手工实现),再以它为工具构造一个能够编译更多语言成分的较大编译程序。如此扩展下去,就像滚雪球一样,越滚越大,最后生成人们所期望的整个源语言的编译程序。

通过移植生成编译程序的思想是把某机器上已有的编译程序移植到另一台机器上去。使用交叉编译技术也可生成编译程序,所谓交叉编译是指一个源语言在宿主机(运行编译程序的计算机)上经过编译产生目标机的机器语言或汇编语言代码。

随着并行技术和并行语言的发展,处理并行语言的并行编译技术和将串行程序转换成并行程序的自动并行编译技术正在深入研究之中。

1.4 编译技术在软件开发中的应用

虽然只有少数人从事构造或维护程序语言编译程序的工作,但是大部分系统软件和应用软件的开发,通常要用到编译原理和技术。例如,设计词法分析器的串匹配技术已用于正文编辑器、信息检索系统和模式识别程序,上下文无关文法和语法制导定义已用于创建诸如排版、绘图系统和语言结构化编辑器,代码优化技术已用于程序验证器和从非结构化的程序产生结构化程序的编程之中。通常,在软件开发过程中,我们需要将某种语言开发的程序转换成另一种语言程序,这种转换过程和编译程序的工作过程是类似的,需要对被转换的语言进行词法分析和语法分析,只不过生成的目标语言不一定是可执行的机器语言或汇编语言。

由于编译原理和技术在软件工程的许多领域中被广泛地应用,因此编译原理和技术是一切从事计算机软件开发和研究的科学家和工程技术人员必须具备的专业基础知识。

本章小结

本章重点介绍了什么是编译程序及编译程序的结构。

编译程序是一种翻译程序,它将高级语言所写的源程序翻译成等价的机器语言或汇编语言的目标程序。

整个编译过程可以划分为5个阶段:词法分析、语法分析、语义分析及中间代码生成、代码优化和目标代码生成。

编译程序的结构按上述5个阶段的任务分模块进行设计。一个典型编译程序的结构框图如图1.5所示。

扩展阅读

学习实现编译器技术的一个好方法,是阅读现有编译器的程序代码。但是,编译器的代码通常是不公开的。Randell和Russell(1964年)给出了一个复杂的早期Algol编译器的代码。编译器的代码也可以从McKeenman、Horning和Wortman(1970年)的研究资料中找到。

Barron(1981年)写过关于Pascal语言实现的论文集,包括Pascal P编译器的实现注释、代码生成细节、Pascal S的实现代码。Wirth(1981年)为学生使用而设计了Pascal的子集。Knuth(1985年)非常清晰地描述了TeX翻译器。有兴趣的读者可以搜索相关资料。

自测练习题 1

1.1 选择题(从下列各题4个备选答案中选出一个或多个正确答案写在题干中的横线上)

(1) 若源程序是高级语言编写的程序,目标程序是_____ ,则称它为编译程序。

- A. 汇编语言程序或高级语言程序 B. 高级语言程序或机器语言程序
C. 汇编语言程序或机器语言程序 D. 连接程序或运行程序

(2) 编译程序是对_____ 程序进行翻译。

- A. 高级语言 B. 机器语言 C. 自然语言 D. 汇编语言

(3) 如果编译程序生成的目标程序是机器代码程序,则源程序的执行分为两大阶段:_____

- A. 编译阶段 B. 汇编阶段 C. 运行阶段 D. 置初值阶段

(4) 编译程序的工作过程一般可划分为下列5个基本阶段:词法分析、_____

代码优化和目标代码生成。

- A. 出错处理 B. 语义分析和中间代码生成
C. 语法分析 D. 表格管理

(5) 编译过程中,词法分析阶段的任务是_____ 。

- A. 识别表达式 B. 识别语言单词 C. 识别语句 D. 识别程序

1.2 判断题(对下列叙述中正确的说法,在题后括号内打“√”,错误的打“×”)

- (1) 编译程序是一种常用的应用软件。 ()
(2) C语言的编译程序可以用C语言来编写。 ()
(3) 编译方式与解释方式的根本区别在于是否生成目标代码。 ()
(4) 编译程序与具体的语言无关。 ()
(5) 编译程序与具体的机器有关。 ()
(6) 对编译程序而言,代码优化是不可缺少的一部分。 ()
(7) 对编译程序而言,中间代码生成是不可缺少的一部分。 ()
(8) 编译程序生成的目标程序一定是可执行的程序。 ()
(9) 含有优化部分的编译程序的执行效率高。 ()

习 题 1

1.1 什么是编译程序?

1.2 编译过程的5个阶段是什么?

1.3 请给出编译程序的结构框图。