高等学校试用教材

编译方法

金成植　编

高等教育出版社

高等学校试用教材

编译方法

金成植 编

高等教育出版社

内 容 简 介

本书以 ALGOL60 语言为对象介绍了编译程序的构造原理，并讨论了相关的各个课题.全书共分十章，包括形式语言简介、词法分析、中间语言、目标生成、优化简介、存贮的组织与分配等内容．其基本概念清楚，重点突出，循序渐进，体现了本课程教学大纲的要求.可作为高等学校计算机有关专业的试用教材，也可供有关科技人员参考.

高等学校试用教材

编 译 方 法

金成植 编

*

高等教育出版社出版

新华书店北京发行所发行

河北省〇五 印刷厂印装

*

开本 850×1168 1/32 印张 13 字数 311,000

1984 年 5 月第 1 版 1984 年 11 月第 1 次印刷

印数 00,001—22,360

书号 13010·0995 定价 2.60 元

前　言

操作系统、编译系统等是计算机系统的基本组成部分，现代计算机都配有这种系统软件．编译程序的任务是把高级语言程序翻译成机器语言程序．计算机通常配有多个编译程序，如ALGOL编译程序、FORTRAN编译程序、PASCAL编译程序等．本教材以ALGOL60语言为对象，介绍编译程序的构造原理．掌握本教材的内容，就可以掌握编译程序的工作原理，并且具备了设计ALGOL60、FORTRAN、BASIC等语言的具体编译程序的基本能力．PASCAL语言需要更多的编译技术，但只要掌握ALGOL60语言的编译技术，就不难学习和掌握PASCAL语言的编译技术．

本教材需要70～80学时的讲课时间．

本书由清华大学郑人杰老师、武汉大学胡久清老师审阅，并提出了许多宝贵意见，特此感谢．

编　者

目　录

第一章　编译程序概述

1.1　什么叫编译程序

语言可分为两大类：**低级语言**和**高级语言**．低级语言又可分为两类：**机器语言**和**汇编语言**．机器语言是计算机的指令系统，而汇编语言是符号化的指令系统．这两种语言都依赖于计算机，使用起来既烦琐、又费时间，而且没有通用性，因此后来出现了很多高级语言．常见的高级语言有 ALGOL、FORTRAN、BASIC、COBOL 等．但计算机的硬件只懂自己的指令系统，因此想用汇编语言或高级语言算题，就必须有这样一种程序，它把用汇编语言写成的程序或用高级语言写成的程序转换成等价的机器语言程序．我们把这种转换程序称为**翻译程序**(Translator)．

通常把汇编语言的翻译程序称为**汇编程序** (Assembler)，把高级语言的翻译程序称为**编译程序** (Compiler)．编译程序的翻译对象称为**源程序**(Source program)，翻译出来的程序称为**目标程序**(Object program)．目标程序也可以是汇编语言的程序．编译程序并不运行源程序，它只是把源程序翻译成目标程序．

用高级语言算题的大致过程是：

1. 用高级语言编写程序并把它穿成输入纸带．
2. 用编译程序进行编译．
3. 根据编译程序所指出的语法、语义错误进行修改．
4. 重复步骤 2 和 3，直至没有错误为止．
5. 用一些简单数据执行目标程序，并检查是否和人算的一致．这一步工作称为**调试**．

6. 调试通过，就投入正式算题，即对正式数据执行目标程序。

高级语言的处理也可采用另一种方式，即它不是先把高级语言翻译成机器语言程序，而后再执行机器语言程序（这是把翻译和执行截然分开的一种方法），而是采用边翻译边执行的解释性方法。这种处理程序称为**解释程序**(Interpreter)。解释程序的加工结果是源程序的计算结果，而不是目标程序。解释程序的致命弱点是执行速度很慢。因此，较大题目的算题大部分都使用编译程序。

采用编译方式和解释方式的解题过程如图 1.1 所示。

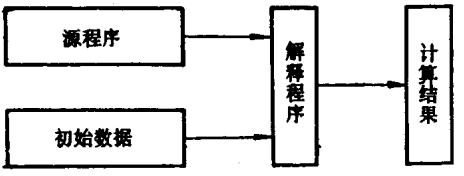


图 1.1(a)解释方式

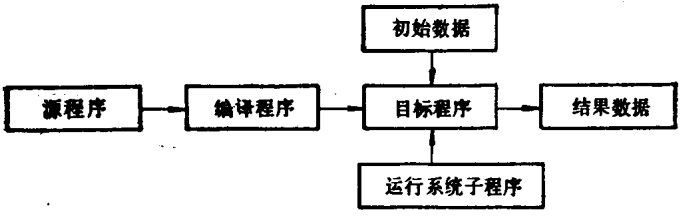


图 1.1(b)编译方式

在目标程序运行时还要用到编译系统所准备的现有子程序，如 sin、cos 等标准函数子程序，称这种程序为**运行系统子程序**。运行系统子程序里除了上述标准函数子程序外，还包含一些其它子程序，如计算下标变量地址的子程序、入口子程序、动态分配用子程序等。

1.2 编译程序的组成部分

不同的编译程序都有自己的组织和工作方式，因此我们无法给出一个标准结构，也不能说哪种结构好，哪种结构不好，它们各有长处和短处．但就编译程序所做的工作内容来说是基本相同的，特别是一些基本的工作．例如，不管是哪种编译程序，都做**词法分析**、**语法分析**和**代码生成**的工作，只是实现的具体方法不同．有的编译程序可能分开各部分来实现，而有的编译程序可能把一些部分合起来同时完成．

在衡量一个编译程序时，可以看以下几方面：

1．编译程序的算法(程序)结构是否良好．结构良好，就便于阅读和修改，从而能提高可靠性．

2．编译程序产生的目标程序的运行速度是否快．这是目标程序的质量问题．

3．编译程序是否有效地利用了存贮区．这是空间问题．

4．编译程序的编译速度是否快．

为了产生高质量的目标程序，往往在生成目标代码之前，把源程序转换成另一种特殊结构的内部形式，如三元式、四元式等．通常称这种内部表示为**中间语言**．

为了有效地利用存贮区，需要有一个组织和管理各种表格的**表处理部分**．一个好的编译程序能够把存贮区中的各种表组织得紧紧的，不留下任何废区．

编译程序在进行工作时首先检查源程序是否有语法和语义错误，并且在发现错误时要把错误信息打印给用户，同时要设法把工作进行下去，为此还要有一个**错误处理部分**．

综上所述，可以给出图 1.2 所示的一种典型的编译程序结构．

学习编译方法就是要学习图 1.2 中所示的各部分的内容，其

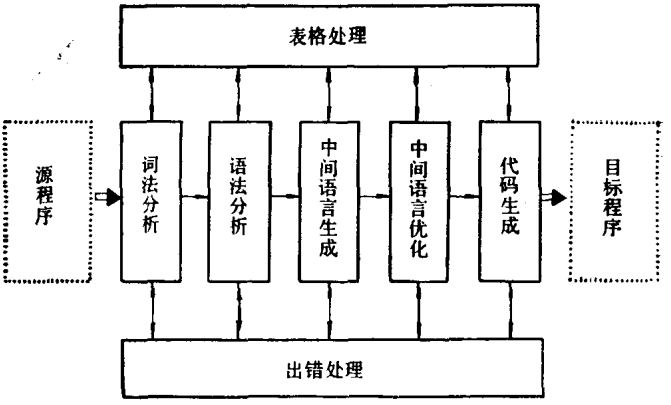


图 1.2　编译程序结构

中每一部分都有独立性，实现方法也有多种，但各部分之间有密切联系.

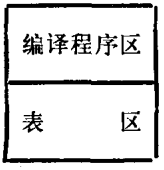
1.3　编译程序的分遍

编译程序按其扫描次数分为**一次扫描**和**多次扫描**. 一次扫描的意思是通过对源程序的一次扫描来直接生成目标程序，而多次扫描的意思是通过多次扫描来生成目标程序.

多次扫描的优点是算法清晰，便于分工，便于优化. 但难免要做一些重复性工作. 初学者开始最好是接触多次扫描的编译程序，特别是在小型机上搞编译程序时，最好采用多次扫描法. 国外有台计算机内存只有 4096 个单元，但在这台机器上实现了 4 万条指令的大型编译程序. 因为内存太小，把编译程序设计成 19 次扫描. 这样，虽然编译程序很大，但在内存中只要给定 2K 多单元就可以了. 但必须有好的外部设备，因为要把中间语言等大量信息频繁地传送(读入)到外部设备中.

在编译时，存贮区可分为两大区：一是编译程序区，二是表区.

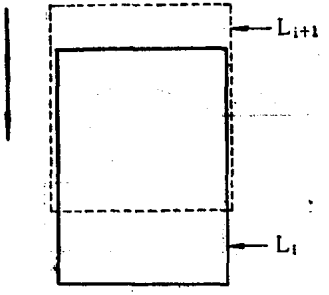
表区中包括很多表格，多时可达几十种。我们把源程序和中间语言等也看做表.



若用 $COMP_i$ 表示第 i 次扫描程序，用 L_i 表示 $COMP_i$ 的工作结果，则我们有

$$COMP_1(L_0) \Longrightarrow COMP_2(L_1) \Longrightarrow \cdots\cdots$$
$$\Longrightarrow COMP_n(L_{n-1}) = L_n$$

其中 L_0 是源程序，L_n 是目标程序. 也就是说，前一次的扫描结果是后一次扫描的加工对象，其中 $L_2, L_3, \cdots, L_{n-1}$ 是中间语言，很显然，我们不必保留所有中间语言，因此后一中间语言 L_{i+1} 可以复盖前一中间语言 L_i. 具体如下图所示:



在 L_{i+1} 和 L_i 区之间要留段间隔. 这个间隔只要能保证 L_{i+1} 的已被生成部分不复盖 L_i 的未被加工部分即可. 如果 L_i 的加工速度(指空间)不慢于 L_{i+1} 的生成速度，那么可把它们完全重迭起来.

假设 M_i 表示 $COMP_i$ 的长度，并且有

$$M = \max_i \{M_i\}$$

则编译程序区的长度只要大于等于M就可以了，通常就选为M.这时各次扫描程序在存贮区中占同一个区，即后次扫描程序将复盖前次的扫描程序．每个扫描程序 $COMP_i$ 按文件形式存放在外部设备中，而调度程序负责依次从外部设备读入 $COMP_i$ 并执行之．调度程序的工作流程如图 1.3 所示．其中 input($COMP_i$)表示将 $COMP_i$ 从外部设备读入到编译程序所指定的编译程序区中．

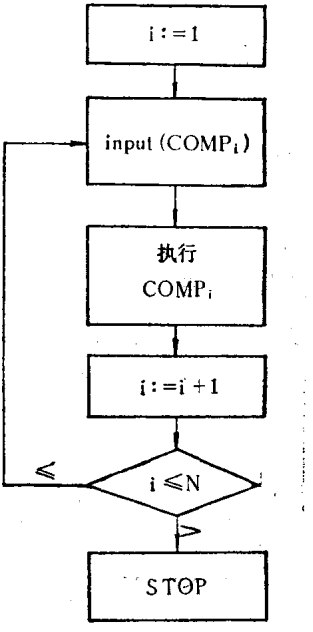


图 1.3

1.4 编译程序的开发

编译程序是一个很复杂的系统程序，通常有上万条指令，多至几万条．随着编译技术和编译理论的发展，编译程序的生成周期也逐渐地缩短．但目前还需要不少人年，而且工作相当艰巨．完成一个编译系统大致要经历下面几个阶段：

1. 分析阶段.

2. 算法设计阶段.

3. 编程序阶段.

4. 调试阶段.

5. 形成文本阶段

分析阶段主要分析和熟悉源语言、计算机以及支撑环境，确定一些原则并组织分工.

算法设计阶段是最重要而且是关键的一个阶段. 这个阶段将完成实质性的工作，在算法设计中头等重要的是使编译算法具有易读性和易改性，为此要使编译算法保持良好的结构. 编译算法的出错是难免的，因此在完成一个编译程序的过程中要不断修改编译算法. 如果算法结构不好，那么就要加重修改算法的难度，而且改错本身就容易产生新的错误. 结构良好，将来也便于扩充编译程序的功能. 另外要注意的是编译程序的速度问题.

编程序阶段是根据所设计的算法用机器语言或汇编语言或其它什么语言写出编译程序. 最终得到机器语言级的编译程序.

调试阶段是通过各种实际的程序例子去调试编译程序. 例如，为了检查编译程序，有意编出有错误的源程序并让编译程序去编译，然后查看是否所有错误都被发现. 这一阶段需要用大量例子来调试，而且在调试中要不断修改编译程序，甚至有时知道有错，但不知道究竟错在什么地方，因此调试阶段将占去很多时间.

形成文本阶段是根据前面的工作写出有关编译程序的一套资料，其中包括源语言文法、编译程序的使用说明、编译算法及其说明等.

我们的愿望是把编译程序的生成过程加以自动化，即让计算机来完成人所做的工作. 但现状还不能令人满意，绝大部分还是要人来完成. 最理想的是，人只要给出语言的文法和语义，计算机

就自动地生成编译程序．有关这方面的工作，已经研究了很多年，但没有得到令人满意的成果．词法分析和语法分析的自动化不是大问题，主要是语义方面的问题．目前的状况是用高级语言来编写编译程序，这样人们可以从琐碎的机器指令中解放出来，把主要精力放在逻辑方面．这就要求有一个适合于编写这种系统程序的高级语言，于是很多人研究了这方面的问题．这种适用于系统程序设计的语言，通常被称为**系统程序设计语言**．目前比较流行的似乎是PASCAL 语言．ALGOL 和 FORTRAN 等语言是科技计算用语言，它们只适合于描述数值计算过程，而不适合于描述非数值过程．当然，我们必须有系统程序设计语言的编译程序，否则用该语言写出来也没有用．除了有这种高级语言及其编译程序外，还要有一些工具性程序．

每个编译程序都是某一种计算机上的，因此编译程序没有通用性．但在为一种计算机设计编译程序时，我们不一定都要重新设计，可以采用“移植”的办法，即把某一机器上的编译程序移到另一机器上来．有不少人在研究一般的移植方法，但要搞通用的移植系统是很难办到的．

还有些人曾采用过所谓的“自展”方法，其主要思想是：首先确定一个非常简单的核心语言，并对此构造编译程序；再扩充语言并用已有语言编制扩充语言的编译程序；如此扩展下去，最后产生我们所希望的较大语言的编译程序．最初，核心语言的编译程序可用汇编语言来写．

第二章　形式语言简介

2.1 基 本 概 念

形式语言理论是编译理论的重要基础.因此,尽管实际编译程序没有完全用到形式语言理论,但有必要学习形式语言的一些基本理论,特别是有关语法分析的理论.这一节将简单介绍形式语言的最基本的概念,而在第四章集中介绍几种较典型的语法分析方法.

字母表(alphabet):它是一个非空有穷集合.例如 $\Sigma=\{a, b, c\}$.

符号(symbol):字母表的元素称为符号.例如,上例中的 a, b, c.

符号串 (string):符号的有穷序列称为符号串.例如 a, b, ab, cba, aaa 等都是上述字母表 Σ 上的符号串.符号串也称为**字**.特别用 ε 表示空符号串(什么符号也没有).

长度(length):设 x 为一符号串,则把 x 的长度定义为其中所包含的符号个数,并记为|x|.例如

$$|\varepsilon|=0,\ |a|=1,\ |aba|=3$$

连结(catenation):设 x 和 y 是符号串,则称 xy 为它们的连结.例如,假定有 x=aa, y=bb,则有 xy=aabb.特别对任一符号串 β,我们有

$$\varepsilon\beta=\beta\varepsilon=\beta$$

集合乘积(product):设 A 和 B 是符号串的集合,则用 AB 表示集合 A、B 的乘积,其具体定义如下:

$$AB=\{xy|x\in A, y\in B\}$$

因为 $\varepsilon\beta=\beta\varepsilon=\beta$, 所以有

$$\{\varepsilon\}A=A\{\varepsilon\}=A$$

假定 $A=\{a, ba\}$, $B=\{bb, cc\}$, 则我们有

$$AB=\{abb, acc, babb, bacc\}$$

空集(empty set): 不包含任何元素的集合称为空集合, 并记为$\varnothing$. 对任一集合 A, 我们有

$$\varnothing A=A\varnothing=\varnothing$$

这是由$\varnothing$和集合乘积的定义得到的. 要注意的是空集$\varnothing$并不包含空符号串 ε, 即 $\varepsilon\in\varnothing$.

方幂(power): 同一符号串的序列可表示成方幂形式. 如

$$X^0=\varepsilon$$

$$X^1=X$$

$$X^2=XX$$

$$\vdots$$

$$X^n=\overbrace{XX\cdots\cdots X}^{n}$$

对符号串集合也可定义方幂, 如

$$A^0=\{\varepsilon\}$$

$$A^1=A$$

$$A^2=AA$$

$$\vdots$$

$$A^n=A^{n-1}A$$

其中 A 是任一集合.

例如, 假定有 $A=\{a, b\}$, 则有

$$A^0=\{\varepsilon\}$$

$$A^2=\{aa, ab, ba, bb\}$$

$$A^3=\{aaa, aab, aba, abb, baa, bab, bba, bbb\}$$