电子计算机介绍

编译程序入门

程 虎 曹东启 编

科学出版社

编译程序入门

程 虎 曹东启 编

科学出版社

内 容 简 介

本书介绍有关编译程序的基本概念和常用方法。

全书共分六章。第一章是概括地介绍；第二章讲词法分析；第三、四章分别介绍两种控制语法分析的方法：递归子程序法和状态矩阵法，以及表达式、各种说明和语句的翻译；第五章主要叙述循环和表达式的优化；最后一章讲编译程序的其他功能。

本书是一本编译程序入门书，可供学习、设计和编制编译程序者参考。

编 译 程 序 入 门

程 虎 曹东启 编

*

科学出版社出版
北京朝阳门内大街137号
中国科学院印刷厂印刷
新华书店北京发行所发行 各地新华书店经售

*

1974年7月第一版 开本：787×1092 1/32
1974年7月第一次印刷 印张：3
印数：0001—20,450 字数：64,000

统一书号：15031·86
本社书号：403·15—8

定价：0.28元

前　言

在毛泽东思想的光辉照耀下，在无产阶级文化大革命的推动下，经过思想和政治路线方面的教育，我国的社会主义革命和社会主义建设正在取得新的胜利.

当代，电子计算机作为一种先进的计算和控制工具正在日益广泛地应用到国防尖端和国民经济的各个部门. 我国在毛主席关于**独立自主、自力更生**的伟大方针和社会主义建设总路线的指引下，计算技术事业（包括研制、生产和应用）发展很快.近年来，相继研究试制成功大型晶体管通用数字电子计算机和新型集成电路通用数字电子计算机. 电子工业取得了不少成绩，正在健康成长. 这就为在我国广泛使用电子计算机打下了良好的物质技术基础. 现在，在我国，电子计算机不仅用于科技数值计算，也用于控制生产过程和其他许多方面.

使用电子计算机解题，首先要根据题意进行数学加工，选择计算方法和拟定计算方案. 然后再将解题步骤用机器指令（机器语言）编成解题程序，这工作称为程序设计. 把程序和初始数据输入计算机，机器就按程序规定的操作对有关数据进行运算，最后得出计算结果并输出.

直接用机器指令编写程序（称为手编程序）是极其繁琐的工作，需要耗费大量的人力和时间，其中很大一部分是机械的、重复的工作，并且很不直观，难学、难写，容易写错，还不易检查出错误，查出错误也难以修改. 程序设计和检查错误所费时间往往比机器解题所需时间多数百倍甚至数千倍，而且机器指令因机器不同而异，所以程序设计要由受过一定训练

的程序员来做，这就大大限制了各部门的工作人员直接使用电子计算机.

为了解决这个问题，后来人们参考数学语言设计了一种算法语言，它既直观、通用，便于学习、编写和交流，又能精确描述算法，为计算机所接受.

采用算法语言写程序比手编程序大为方便，易学、易写，不易写错，易查错，查出错后也容易修改. 各部门的工作人员只要稍加学习就能掌握，从而大大有利于推广使用电子计算机. 用算法语言写程序还去掉了许多繁琐的工作，节省了编程序和查错、改错的时间，能够把时间和精力用在主要工作方面.

但计算机不懂算法语言，要由事先编好的称为编译程序的程序先把用算法语言写的程序翻译为机器指令程序，然后在机器上运算，得出结果.

近年来，我们在编译程序方面做了一点工作，经验极为有限. 从去年四月起曾先后办了几次介绍编译程序的讲座，内容较为浅显. 但为了适应我国计算技术事业迅速发展的大好形势，满足普及和提高的需要，现在把不成熟的讲义稍加改编、增补出版.

这本小册子以国际上常用的算法语言 ALGOL 60 为背景[1]，就有关编译程序的基本概念和目前最常用的编译方法（这些方法对 FORTRAN 等其他语言基本适用）作了一些介绍，可供学习、设计、编制编译程序时参考. 至于计算机和算法语言方面的知识，读者可参考其他有关书籍和文章.

目前，算法语言种类繁多，编译程序技术也在迅速发展，新方法不断涌现，书后提供的参考文献对从事软件方面工作

1) 请参阅 ALGOL 60 报告[1].

的读者可能会有所帮助.

伟大领袖毛主席教导我们:**"在生产斗争和科学实验范围内,人类总是不断发展的,自然界也总是不断发展的,永远不会停止在一个水平上.因此,人类总得不断地总结经验,有所发现,有所发明,有所创造,有所前进".** 希望读者在实际运用时,结合具体情况加以创造,奋发图强,赶超世界先进水平,为社会主义祖国做出新贡献.这就是编写这本小册子所想达到的目的.

本书部分内容取材于109乙和109丙计算机的编译程序所用的算法;同时,在编写本书的过程中参考了清华大学计算数学教研组编写的讲义以及其他资料.在此对有关同志表示感谢.编者还感谢董韫美同志,他曾审阅了全书,并提了宝贵的意见.

由于水平所限,小册子中难免有许多缺点、错误和不当之处,恳切地希望广大读者提出宝贵意见,给予批评指正.

编　者

1973年9月

目　　录

第一章 概 述

§1. 主 要 职 能

用算法语言写程序比用机器指令写程序方便得多，例如可写

$$a := b + c \times d;$$

此处符号 := 称为赋值号，是有方向性的，其意义是把右边的计算结果赋给左边的变量 a. 这样就不必用一条一条指令去写程序，也不必去做繁琐的存储分配和代真工作.

由于计算机不懂算法语言，所以用算法语言写的程序在机器上要分两步来实现：

第一步. 先把用算法语言写的程序翻译成等价的机器指令程序，这称为编译阶段. 这个工作由称为编译程序的程序来完成.

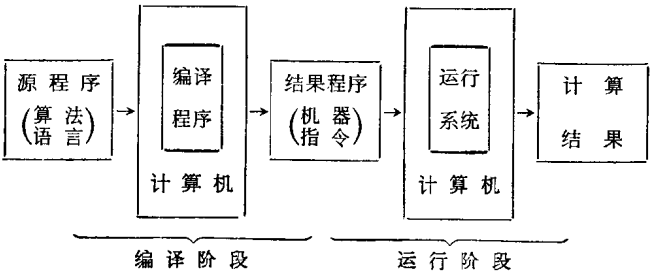
第二步. 执行由编译程序翻译出来的机器指令程序，对初始数据进行加工，算出最后结果，这称为运行阶段.

用算法语言作为源语言写的程序称为源程序，翻译出来的用机器指令(机器语言或称代码)作为结果语言(目标语言)所构成的程序称为结果程序(目标程序，代码程序或结果代码). 运行时还要有若干个固定的子程序(如数组动态存储分配子程序、下标变量直接地址计算子程序等)陪伴结果程序工作，这些子程序总起来称为运行系统. 编译程序和运行系统合起来称为编译系统.

另外，源程序也可通过解释系统进行解释执行，即逐个翻译并立即执行源程序的语句. 不是编出结果程序再执行，而是

直接解释得出结果，就像在定点计算机上使用浮点解释系统实现浮点运算那样．还可以把编译和解释结合起来，先把源程序编译成一种中间语言程序，再对此中间语言程序进行解释执行，得出结果．本质上是两种系统，本书不谈解释系统，只介绍编译系统．

必须指出，不能只注意编译程序把源程序翻译成结果程序这一点，实际上，使用编译系统解题，和手工程序设计解题方式相比较，除了使用者不像手工程序设计那样用机器指令编写程序外，整个解题过程都发生了变化．不考虑这些变化和新产生的问题，就不可能使编译系统实用．关于这方面的问题，在第六章中叙述．



§2. 实现步骤

编译程序不是一般的算题程序，而是一个把用算法语言写的源程序翻译成用机器指令写的结果程序的程序．因为通用数字电子计算机不仅能做算术运算，而且能做逻辑运算，所以能编出这种程序．和普通翻译外文书刊相仿，编译阶段大致要经过下列几个步骤：

第一步．扫视源程序(读符号)和换码；

第二步．词法分析(读单词)，包括分配单元，造名字特性表等；

第三步．语法分析(或称语法分解)；

第四步．修饰优化；

第五步．编出结果程序．

下面用一个具体例子来说明这些步骤．

例．计算园柱体的全面积

$$S = 2\pi R(H + R),$$

其中 R 为半径，H 为高．

用 ALGOL 60 可写出下列源程序：

```
begin
    real R,H,S; S:=2 × 3.14159 × R × (H + R)
end
```

其中 **begin**、**real** 和 **end** 的意思是开始、实型和结束．ALGOL 60 中定义符的英中文对照，见附录．

第一步．扫视源程序(读符号)和换码．

源程序经穿孔输入到机器的存储器中，编译程序再去扫视源程序(读符号)，认出是些什么符号．有的基本符号像 **begin** 是由五个符号拼成的，还要换成内部符号，便于以后对这些拼写定义符进行语法分析和语义加工，而无需临时现拼．

第二步．词法分析(读单词)，包括分配单元，造名字特性表等．

有些基本符号有独立意义，如 +、−、×、/等；有的没有独立意义，例如标识符中的字母、数字就没有独立意义，组成一个标识符才有意义．由数字和小数点等组成的数也是如此，其中数字和小数点等就没有独立意义，组成一个数才有意义．和翻译外文书刊先要把外文单词分辨出来一样，这里要把源程序中有独立意义的单词分辨出来．外文单词在书刊中由空白或标点符号隔开，在算法语言中单词的组成有一定规则，词法分析首先就是把单词分辨出来．

然后对标识符造名字特性表记录有关信息，对常数要造常数表，还有其他一些表.

经词法分析后，上例源程序加工成为内部编码程序

begin

$$M_3:=C_1\times C_2\times M_1\times(M_2+M_1)$$

end

其中 **begin** 和 **end** 表示两个内部编码，M_1、M_2、M_3 和 C_1、C_2 是和分配单元地址有关的内部编码. 这些内部编码均由等长的二进位数组成（计算机内的存储形式一般用二进制），M_1、M_2、M_3 代表标识符 R、H、S，C_1、C_2 代表常数 2 和 3.14159.

实际上，编码不再是符号，而是数码. 假定一个单词用两个八进数字表示，上例成为

01 13 03 14 04 15 04 11 04 06 12 05 11 07 02

begin $M_3:=C_1\times C_2\times M_1\times(M_2+M_1)$ **end**

其中 **begin** 和 **end** 的编码是 01 和 02；:=、×、+、(、) 的编码是 03、04、05、06、07；M_1、M_2、M_3 和 C_1、C_2 的编码是 11、12、13 和 14、15. 但为叙述方便，下面仍用符号表示.

除内部编码程序外，还有两张表如下：

名字特性表

	R
实型	M_1
	H
实型	M_2
	S
实型	M_3

常数表

C_1	2
C_2	3.14159

第三步．语法分析(或称语法分解)．

经过词法分析后，每个内部编码符号都代表有独立意义的单词．这些单词怎样组成短语和句子，要靠语法分析．因为算法语言有一整套必须严格遵守的语法规则，所以语法分析能够机械地进行，即可编出程序让计算机去做．语法分析方法很多，这里主要介绍三种常用的方法：优先数法，递归子程序法和状态矩阵法．语法分析主要为了识别各类语法成分(如表达式、说明、语句等)的结构，是翻译的核心部分．进行语法分析的同时，可做语法检查，查明源程序中哪些地方不合语法．

上例经语法分析得知，是一个赋值语句，而且知道要先算 $C_1 \times C_2$，再 $\times M_1$；把结果存起来，然后算 $M_2 + M_1$；其结果与 $C_1 \times C_2 \times M_1$ 的结果相乘，最后赋给 M_3．

第四步．修饰优化．

经编译程序编出的程序，一般不如手编程序质量好，不但结果指令条数多，更大的问题是结果程序的运算时间长．编译程序是统一处理各种源程序，手编程序是针对某个具体题目编写，当然会精巧．正像要使文章翻译得好，就要进行修辞一样，要使结果程序质量好(不但指令条数少，主要是结果程序的运算省时间)，就要对结果程序进行修饰优化．

如上例最好在编译时算出 $C_1 \times C_2$，并且先编 $M_2 + M_1$，这样省中间工作单元，省指令，省运行时间．因为编译时 $C_1 \times C_2$ 只要算一次，结果程序中直接用 $C_1 \times C_2$ 的结果 C_3．

修饰优化是在对源程序作了词法分析、语法分析的基础上，进一步做许多分析、比较工作，查出可优化部分，将源程序(或中间语言程序)进行等价变换，以便能够翻译出优化的结果程序．

第五步．编出结果程序．

首先要弄清相应于各语法成分的结果程序结构（无论是否优化），即对于每一个语法成分，对应到一组什么样的机器指令．在选定机器和算法语言后就可以做这项工作，它是编译程序进行翻译时的依据，其作用有点像翻译外文书刊时的字典（包括单词和成语等）．确定各语法成分的结果程序结构的工作必须认真做好，如有差错，就会影响编译结果的正确性．

上例的结果程序如下：

不优化的结果程序：

取 C_1
× C_2
× M_1
送 W_1
取 M_2
+ M_1
× W_1
送 M_3

共八条，用了一个工作单元 W_1．

优化的结果程序：

取 M_2
+ M_1
× C_3
× M_1
送 M_3

共五条，未用工作单元．

上面是根据逻辑功能的不同，把编译阶段分为几个步骤，各步骤不是绝然分开的，实现时可依具体情况而定．例如 **begin**，**end** 这些拼写定义符的处理可以放在扫视源程序（读符号）或换码这一步骤里，也可放在词法分析里；数的处理可以放在词法分析里，也可放在语法分析里去做．

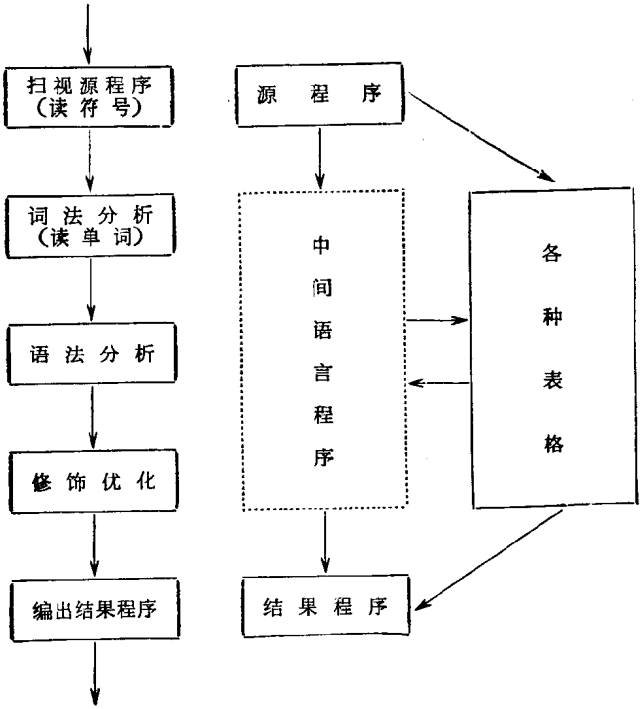
把编译阶段分为五个步骤后，编译程序本身的结构也可分成若干部分．对源程序从头到尾扫视一次并做有关加工称为一遍（或趟）．每一遍做一个或相连的几个步骤的工作，例如第一步，第二步可合为一遍，第三步，第四步又可合为一遍，最后一遍完成第五步的工作．每一遍产生一个中间结果，称为中间语言程序．前一遍的结果（中间语言程序）是后一遍的加工对象，最后一遍的结果就是结果程序．

一个编译程序是否分遍和如何分遍，要根据具体情况（如计算机存储容量的大小，算法语言的繁简，解题范围的宽窄，设计编制人员的多少等）而定．分遍的好处是使各遍功能独立单纯，相互联系简单，编译程序逻辑结构清晰，优化的准备工作充分，使优化做得较多较好；缺点是有一定的重复性工作，如各遍都有读符号、读单词、送符号等，这样就增加了编译程序的长度和编译的时间．

概括编译程序和翻译外文书刊相仿的编译步骤，可列表对比如下：

	翻译外文书刊	编译程序
分析	阅读原文	扫视源程序（读符号）
	识别单词	词法分析（读单词）
	分析句子	语法分析
综合	修辞加工	修饰优化
	写出译文	编出结果程序

编译程序的逻辑结构和相应信息流程可分别图示如下：


扫视源程序
(读符号)
词法分析
(读单词)
语法分析
修饰优化
编出结果程序
源程序
中间语言程序
各种表格
结果程序

第二章　词法分析

§1. 读符号、换码、送符号和送指令

一　读符号(扫视源程序)

这个工作与所用计算机的字长、内外存储器的容量和符号的编码等关系比较密切，一般源程序先放在外存储器中，用时分段调入内存储器，然后一个单元一个单元地取，每个单元中再一个符号（占若干个二进位）一个符号取出来，就像看外文书刊那样，从第一页第一行第一个词的第一个字母看起，直至全部看完．有时要向前'假读'几个符号，以判明情况，然后再回头处理已读过的符号．如果正好处在外存调入内存的分段交接处，就会从内存取不到，这个问题可采取重迭保存若干个单元的办法来解决．

只要设置一些计数器来标记当前读到第几个单元的第几个符号，就能实现内存的读符号，再加上判断是否读完一段，读完一段再从外存调入一段，直至整个源程序读完．

通常用一个子程序实现读符号工作供其他部分使用．

二　换码

换码是为了后面处理方便，也可放在读单词中一起处理．拼写定义符要通过换码换成一个内部符号编码．有时一个算法语言的源程序还允许几种文本，如汉字的、汉语拼音的、英文的，通过换码换成统一的一种内部编码．这样换了之后，就对几种文本作同样的处理．还有因穿孔设备的关系要换码