电子计算机介绍

数字计算机的编译程序构造

〔美〕D. 格里斯

科学出版社

数字计算机的编译程序构造

〔美〕 D. 格里斯
曹东启 仲萃豪 姚兆炜 译

科学出版社
1982

内 容 提 要

本书较全面系统地介绍了编译程序构造的主要技术、实现方法以及必备的基本理论知识.

全书共分二十一章. 前七章讲述编译程序的有关概念、形式语言理论、扫描程序以及各种语法分析方法. 随后九章介绍运行时存贮的组织、各种符号表、源程序的内部形式、语义程序、变量的存贮分配、错误校正以及解释程序等内容. 最后五章进一步讨论了代码生成、代码优化、宏功能的实现以及编写编译程序的系统,并对建立编译程序时所应考虑的问题给出了若干扼要提示.

本书可供计算机软件工作者及有关专业人员学习参考之用. (为了方便读者,此次印刷将原来的上、中、下三册合为一册出版.)

David Gries

COMPILER CONSTRUCTION FOR DIGITAL COMPUTERS

John Wiley & Sons, Inc.

数字计算机的编译程序构造

〔美〕D. 格 里 斯

曹东启 仲萃豪 姚兆鳍 译

*

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

*

1982 年 12 月第 一 版 开本:787×1092 1/32

1982 年 12 月第一次印刷 印张:19 1/8

印数:00001—13,600 字数:435,000

统一书号:13031·2058

本社书号:2811·13—1

定价: 2.35 元

中译本再版前言

自从本书问世以来，国际上又相继出版了不少讨论编译理论和编译技术的书籍．它们的不足之处在于有些只侧重于介绍编译理论，有些则只针对具体系统．因此，从理论与工程两方面结合的角度来看，本书剪裁适当，是一本公认的好书．本书论及的很多观点和技术，至今仍具有较大的实用价值．译本在国内出版以后得到了广大读者和软件工作者欢迎．有的单位用它作为专业课或讨论班的教材，也有不少读者用它作为自学参考书．本书在普及编译技术和推动编译系统研制方面，起了一定的促进作用．

在这一段时间内，我们发现了译本中的一些错误，同时也收到了一些同志提出的意见，趁此再版之机，我们对这些错误做了改正．这里，特别要感谢舒裕录（N. N. Shulman）先生，他从头到尾仔细地审阅了全书，提出了很多宝贵的意见．我们还感谢徐家福、杨芙清、乔如良、张文宽、陈火旺、何新贵、钱家骅、朱三元、万举才、邢竞侯、李百令等同志，他们对本书也提出了不少意见．

译　者

1980年2月

序　言

鉴于编译程序和解释程序是任何计算机系统的一个必要组成部分，因此，没有它们，就要用汇编语言、甚至要用机器语言去编写程序．这样一来，就使得编译程序构造在计算机科学中成为一个重要而又有实际意义的研究领域．本书的目的是采用连贯方式介绍编写编译程序时所使用的一些主要技术，为的是初学者易于入门，专业人员便于备查参考之用．

本书以介绍所谓"语法制导的编译方法"作为准则．事实上，全书三分之一以上的篇幅是讨论"形式语言理论和自动语法识别"这一课题的．作者深深感到，对于任何一个从事编写编译程序的人来说，都应知道有关这一课题的基本知识．但是，这并不意味着，每个编译程序一定要用这种自动语法识别方法去编写．实际上，有许多程序设计语言不宜使用这些方法．然而，形式语言理论的基本知识，不仅使编译程序编制者更能了解他的编译程序中将会遇到一些什么问题，而且还有助于他更系统、更有效地设计和编写编译程序．

但是，语法分析只是编译程序结构中的一小部分，至于其它重要课题，诸如：符号表的组织、错误校正、代码生成、代码优化等等，作者已把它们写入有关章节之中．限于篇幅，略去了另外几个课题（例如，常数转换，可增编译程序等）．

本书是为了满足下述两种需要而编写的，对编译程序构造有兴趣或已从事这方面工作的专业程序设计者来说，它是一本自学书和参考书；另一方面，本书又可用作讲授"编制编译程序"这门课程的一学期课本．事实上，本书已包括：在1968

年3月《Communications of the ACM》上公布的并由ACM课程委员会推荐的课程I5(即编译程序构造)中所列的全部课题(甚至更多一些).

要求读者至少要有一年以上使用高级语言(如FORTRAN, ALGOL, PL/I)和汇编语言编写程序的经验,并能读懂ALGOL程序.在某些章节中,还假定读者熟悉初等布尔矩阵理论(书中附有简要介绍),懂得什么是集合,什么是两个集合的并,等等.除此之外,读者还应具备一些数学知识,比如说,学过大学二年级或三年级的数学专业课程.

要求读者具有使用高级语言的经验,这一点是非常明显的;因为,本书所要讨论的就是翻译用这类语言写出的程序.然而,更重要的是要有汇编语言的经验,因为它有助于我们熟悉计算机是如何工作的.但是,我们很少涉及任何一种特定的汇编语言.只有少数几个IBM360汇编语言程序分散在本书的有关章节中,但越过这些段落也不会影响对本书内容的理解.

编译程序本身也是用某种语言写出的程序.因此,编译程序某些部分的例子也必须用某种程序设计语言写出.为了易于阅读起见,我们选用一种类似ALGOL的语言.这些例子通常是简短的,以便容易领会它们.过于琐碎致使我们难于理解的地方,作者就使用英文(译文中已把这些译为中文——译者注)而不使用ALGOL语言去加以说明.作者十分仔细地静态检查了所有的程序段,但要告诫读者,并非全部例子都在计算机上调整过.

在本书的附录中,对所用的变形ALGOL给了一个扼要的介绍.此介绍是简短的,而且需要较多的ALGOL方面的知识.假如读者不熟悉ALGOL以及语言的语法描述,那末,建议读者在学完第二章后,再去读这一附录,就会更好一些.

本书已超出一学期课程的讲授内容．但我们认为至少要学习下述章节：

第一章　引言，
第二章　文法和语言(略去第 7 节)，
第三章　扫描程序(略去第 4、5、6 节)，
第四章　自顶向下识别算法(着重学习第 3 节)，
第五章　简单优先文法，
第八章　运行时存贮的组织(略去第 6、9 节)，
第九章　各种符号表的组织，
第十章　符号表中的数据(略去第 2 节)，
第十一章　源程序的内部形式(略去第 4、5 节)，
第十二章　介绍语义程序，
第十三章　类似 ALGOL 结构的语义程序(略去第 6 节)，
第十四章　运行时变量的存贮分配(略去第 3 节)，
第十六章　解释程序，
第二十一章　对编译程序编写者的一些提示．

读者可能意识到：在语法分析这部分内容中，作者特别强调简单优先技术．这并不是因为它是最好的(它或许是最坏的)，而是因为它最容易讲授．如果时间充裕，希望教师补充一些精美的自底向上语法分析方法，如算符优先文法(6.1节)，高阶优先文法(6.2 节)，转换矩阵方法(6.4节)，产生式语言(第七章)，或其它未列入的方法，等等．

本书各章节的介绍顺序也是可以改变的．事实上，在讲授本教程时，最好把一些实际素材穿插到语法理论中加以学习．第八章是独立的，讨论运行时的存贮管理；第九、十、十一和十六各章分别讨论了符号表、源程序的内部形式以及解释程序，这四章的内容只要按照上述顺序可安排在任何时间学习．

第二十一章应当特别地提一下．它把编译程序编制者所应熟悉的各种事实和见解收集在一起．这些内容或者不属于任何章节，或者是安排在个别章节中但需专门强调一下的重要内容．读者应反覆翻阅这一章，并精读当前感兴趣的段落．

“编制编译程序”的课程应当是一个实验的课程．学生应按一至三人分成小组，具体动手编写、调整某些简单语言的编译程序或解释程序．只有这样，才能使学生真正了解编译程序的实质内容．最好选择解释程序，因为学生无需在机器语言细节上自找麻烦；而且重要的在于思想，不在于细节．由此方针出发，整个计划应当是用高级语言进行程序设计．作者的经验是采用 PL/I 或类似 ALGOL 的语言，要比采用 FORTRAN 语言为好．因为用 FORTRAN 写出的编译程序偏大而且难读．如果能用编写翻译程序的系统，那么就使用这种系统．

为了便于改进和创新起见，应从基本而又简单的语言开始，其内容包括：整变量、赋值语句、表达式、标号和转移、条件语句以及简单的读写语句．然后，再令每个小组增加一两种特色来扩充这种简单的语言．例如：可以增加数组、记录(结构)、不同的数据类型、分程序结构、过程、宏功能以及循环语句等．

可以按照课程的进展程度，分期编出并检查编译程序．首先是扫描程序，其次是语法分析程序，再次是符号表程序，最后是语义程序．一旦学完解释程序的有关章节之后，就能设计和实现它．由于我们采用了这种办法，从而把整个工作均匀地分散到整个学期之中，而不是堆积到学期的终了．

在每章最后一节中，我们给出许多已发表的参考文献，其中有些文章可能已在其它章节中出现过。凡出现人名时，应自动理解成有参考文献列于本书的参考书目之中．该书目是

按作者的字母顺序和文章发表的年份排列的．如果某作者的参考文献多于一篇，那末，按〈名字〉(〈年份〉)去查阅相应的文献．例如 Gries(68)．假如作者某年发表的文章多于一篇，那末，该年第一篇用(〈年份〉a) 表明，第二篇用(〈年份〉b) 表明，等等．这样，Floyd (64b) 将查阅 Floyd 论述“程序设计语言的语法”那篇文章．

除标题和某些插图外，本书原稿是在 IBM360/65 计算机上产生的，所使用的程序称为 FORMAT，它是由 Gerald M. Berns (69) 编写的．作者还感谢 John Ehrman，他对该程序做了不少重要修改和补充．使用 FORMAT 程序不仅易于编辑原始题材，而且也易于把它分发给程度不同的学生．然而，这种方法也迫使作者放弃了一些习惯的表示方法．本书采纳了下述两点重要的变通的表示方法．由于用来打印本书的打印机上没有下角标和上角标（'到' 除外），所以书中采用运算符“!”表示方幂．这就是说，c!b 表示 c 的 b 次方．类似地，由于缺少下角标，所以，作者只能把 n 个符号的序列表为 S[1]，S[2]，…，S[n]．在含义明显的地方，我们可以把这个序列简写为S1，S2，…，Sn．

本书各章节最初是在斯坦福大学和康奈尔大学作为讲授“编制编译程序”这门课程的讲义，以后作者又有机会在康奈尔每年植树节期间举办的密执安暑期讲习班里，以及 1970 年在以色列举行的高级程序设计系统的国际讨论会上，讲述了原书的节选教程．作者对这几期学生深表谢意，他们评议了本书的原稿，很多人提出了有益的建议．作者真诚地感谢 Steve Brown，他从头到尾仔细地审阅了本书的全部手稿，发现不少错误，提出了很多宝贵意见和批评．

目　　录

第一章 引 言

1.1 编译程序、汇编程序、解释程序

翻译程序是一个把**源程序**翻译成等价的**结果程序**的程序. 源程序是用源语言编写的，而结果程序是由**目标语言**构成的. 翻译程序本身在**翻译时**执行.

如果源语言是 FORTRAN, ALGOL 或 COBOL 这类高级语言，而目标语言是某计算机的汇编语言或机器语言，那末我们就把这种翻译程序称为**编译程序**. 机器语言有时称为**代码**，所以结果程序有时又叫做**结果代码**. **编译时**，把源程序翻译为结果程序；**运行时**，才真正执行此结果程序.

汇编程序是这样一种程序，它把用汇编语言写的源程序翻译成某计算机的机器语言程序. 汇编语言十分接近机器语言. 事实上，很多汇编语言的语句恰好就是机器语言的语句的符号表示. 况且，汇编语言的语句通常具有固定格式，这种格式将使汇编程序更易于分析这些语句. 在这些汇编语句中通常不包含嵌套语句，分程序，等等.

一源语言的**解释程序**将以该语言写出的源程序作为输入接收并执行之. 编译程序和解释程序之间的差别在于：解释程序不产生欲被执行的结果程序，而是直接执行源程序本身.

对一个纯粹的解释程序来说，**每当**要执行一源程序的语句时，它就分析该语句，以便发现如何完成所要执行的功能. 显然，这种方法的效率非常低，且不常使用. 常用的方法是分两遍编写解释程序. 第一遍分析整个源程序（这和编译程序

所用的方法差不多)，并把它翻译为内部形式．然后，第二遍解释或执行这种内部形式的源程序．设计这种内部形式的目的是为了缩短执行每个语句所需的"译码"或分析的时间．

如前所述，编译程序自身也是用某种语言写出的程序——它的输入是源程序，而其输出是等价的结果程序．从历史的角度来看，我们过去是利用计算机的汇编语言、采用手工方式编写编译程序的．在很多情况下，汇编语言是唯一可用的语言！然而，目前倾向使用高级语言编写编译程序，因为，这不仅减少了大量的编程序和调程序的时间，而且还使最后得到的编译程序易于阅读．我们可以找到很多为快速编写编译程序而设计的语言．这些就是所谓的"编译程序的编译程序"，它是"编写翻译程序系统(TWS)"的一个子集．我们将在第二十章简要讨论它们．

本书重点是介绍编译程序的构造．但书中也讨论了有关解释程序的一些问题；因为编译程序构造中所用的技术也可用来编写解释程序，所以，这样只是稍微增加了本书的一些篇幅．然而，本书不讨论汇编程序，因为，任何一个熟悉编译程序构造的人，是不难了解一个汇编程序要做些什么工作，以及它如何实现它所要做的工作．

在本书中，读者将找不到一个完整的编译程序．其目的不是让读者了解作者是如何编写一个具体的编译程序的，而是想让读者学会如何去编写自己的编译程序．当然，就编译程序构造中所用的技术和方法而言，读者能在本书中找到很多有关的例子和讨论(显然这不是全部的，甚至作者不敢断言这是足够多的)．书中的例子是用变形 ALGOL 语言编写的，在附录中附有该语言的简要说明．如果读者把本书当作教科书，那末，希望读者用 ALGOL，FORTRAN，PL/1 或其它高级语言编写自己的编译程序或解释程序；这是学习编译程序

构造的最好方法.

1.2 编译过程概述

编译程序必须**分析**源程序,然后**综合**成结果程序. 因此,首先要把源程序分解为若干个基本部分; 其次再根据它们建立各个等价的结果程序部分.为了完成上述工作,编译程序就要在分析遍中建立很多表格,以便在分析及综合时使用它们. 图 1.1 较为详细地显示了编译的整个过程; 虚矢线表示信息

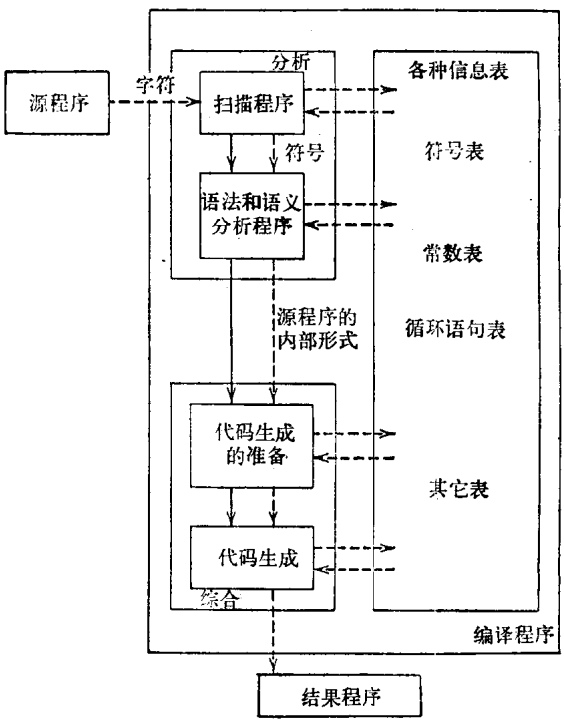

图 1.1 组成编译程序的各个逻辑部分