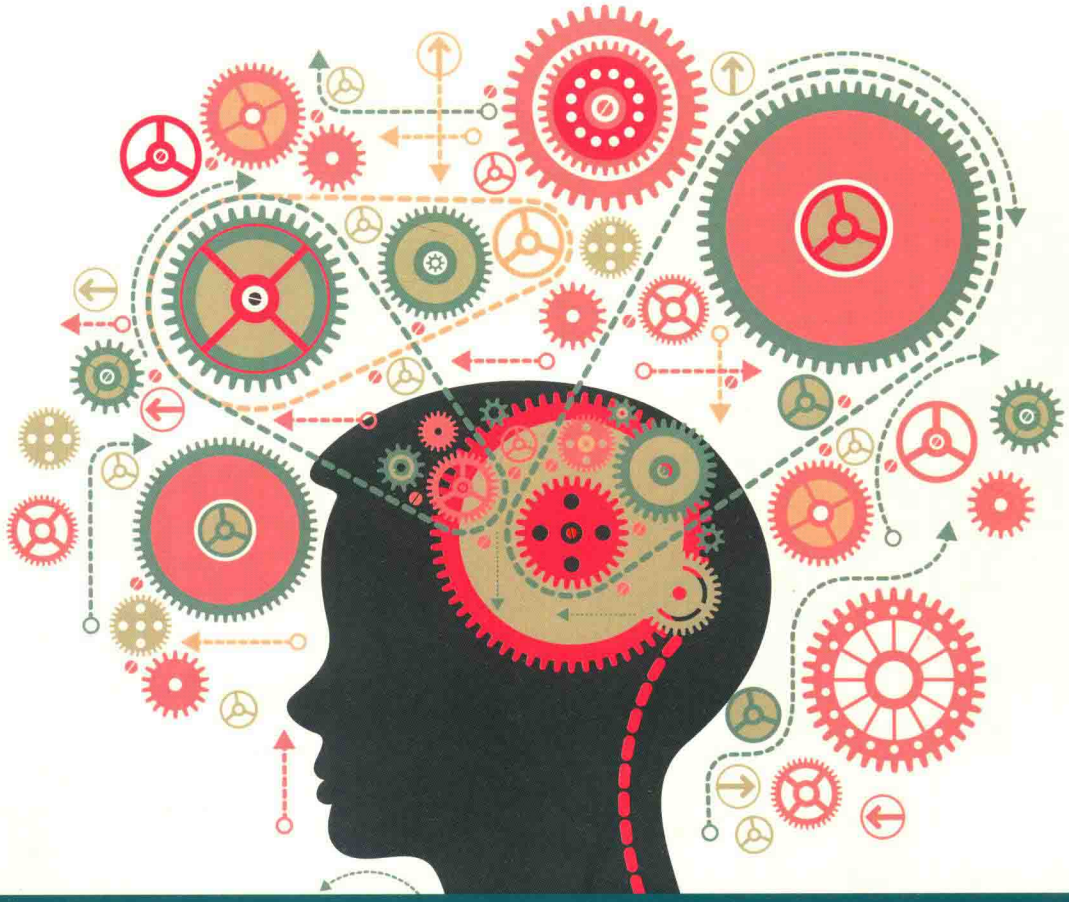
新工科建设之路·计算机类专业规划教材



编译原理

龚宇辉　王丽敏　主　编
周瑞红　韩旭明　副主编

中国工信出版集团
電子工業出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
http://www.phei.com.cn

新工科建设之路·计算机类专业规划教材

编译原理

龚宇辉　王丽敏　主　编

周瑞红　韩旭明　副主编

電子工業出版社
Publishing House of Electronics Industry
北京·BEIJING

内 容 简 介

本书共分 9 章。第 1 章介绍了编译程序的基础知识，包括编译工作的基本过程及各阶段的基本任务；第 2 章介绍了文法及语言的基本概念、文法分类、词法分析程序的设计原理与构造方法等；第 3 章介绍了自顶向下语法分析的基本思想和分析技术，包括语法分析的任务、LL(1)文法、LL(1)分析法和递归下降分析法；第 4 章介绍了自底向上语法分析的基本思想和分析技术，包括算符优先分析、LR 分析法等；第 5 章介绍了语义分析与中间代码的生成；第 6 章介绍了符号表的组织与管理，包括符号表的作用、符号表的组织和使用方法；第 7 章介绍了运行时的存储组织与分配技术；第 8 章介绍了代码优化的基本概念、基本块的划分、局部优化和循环优化方法等；第 9 章介绍了目标代码生成的基本技术。

本书系统性强，概念清晰，内容简明通俗，每章章首配有本章的学习目标和学习要求，章末配有本章小结和适量的习题，使学习者快速掌握书中的内容。

图书在版编目（CIP）数据

编译原理 / 龚宇辉，王丽敏主编. — 北京：电子工业出版社，2018.6

ISBN 978-7-121-33731-4

I. ①编… II. ①龚… ②王… 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(2018)第 033240 号

策划编辑：竺南直
责任编辑：张　京
印　　刷：三河市鑫金马印装有限公司
装　　订：三河市鑫金马印装有限公司
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：787×1 092　1/16　印张：12.25　　字数：313.6 千字
版　　次：2018 年 6 月第 1 版
印　　次：2018 年 6 月第 1 次印刷
定　　价：35.00 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）88254888，88258888。

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

本书咨询联系方式：davidzhu@phei.com.cn。

前　言

编译技术是计算语言发展的支柱，也是计算机科学中发展最迅速、最成熟的一个分支。“编译原理”是一门研究设计和构造编译程序的原理和方法的课程，是计算机及其相关专业的一门核心课程，在教学中占有极其重要的地位。该课程蕴涵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。编译程序是计算机系统软件的重要组成部分，其基本原理和实现技术也适用于一般软件的设计和实现，因此对应用软件和系统软件的设计与开发有一定的启发和重要的指导作用，在软件工程、软件自动化、语言转换、程序分析及其他领域具有广泛的应用。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。通过该课程的学习，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及方法，为今后从事应用软件和系统软件的开发奠定一定的理论和实践基础。

本书共分 9 章。第 1 章介绍了编译程序的基础知识，包括编译工作的基本过程及各阶段的基本任务；第 2 章介绍了文法及语言的基本概念、文法分类、词法分析程序的设计原理与构造方法等；第 3 章介绍了自顶向下语法分析的基本思想和分析技术，包括语法分析的任务、LL(1)文法、LL(1)分析法和递归下降分析法；第 4 章介绍了自底向上语法分析的基本思想和分析技术，包括算符优先分析、LR 分析法等；第 5 章介绍了语义分析与中间代码的生成；第 6 章介绍了符号表的组织与管理，包括符号表的作用、符号表的组织和使用方法；第 7 章介绍了运行时的存储组织与分配技术；第 8 章介绍了代码优化的基本概念、基本块的划分、局部优化和循环优化方法等；第 9 章介绍了目标代码生成的基本技术。

本书系统性强，概念清晰，内容简明通俗，每章章首配有本章的学习目标和学习要求，章末配有本章小结和适量的习题，可使学习者快速掌握书中的内容。本书附录中的程序代码可扫相应的二维码查看，或者登录华信教育资源网 www.hxedu.com.cn 注册后下载。

本书根据作者多年的教学经验编写而成，在成书的过程中，编著参考了书末所列出的有关文献，在此，向这些书籍的作者一并表示感谢。由于编著者水平有限，时间仓促，书中难免存在一些缺点和不足，敬请读者多提宝贵意见，以便再做修订补充。

编著者

目　录

第1章 编译简述

学习目标

正确理解什么是编译程序；掌握编译程序工作的基本过程并了解各阶段的基本任务。

学习要求

- 掌握：编译方式与解释方式的根本区别和编译程序的基本过程。
- 了解：编译程序各阶段的基本任务及实现编译程序的方法。

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。编译器的编写涉及程序设计语言、计算机体系结构、语言理论、算法和软件工程等学科。本章主要介绍翻译程序、汇编程序和编译程序的概念，实现高级语言的编译方式，一个典型的编译程序的组成，遍、编译程序前端和后端及编译程序的构造方法等。

1.1 程序的翻译

语言是人与人之间传递信息的媒介和手段。世界上存在多种语言，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。在计算机领域里，程序是计算机系统中计算任务的处理对象与处理规则的描述，是计算机实现各类信息处理的工具，是人与计算机的接口。使用过现代计算机的人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的，但是计算机不能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够执行。

1.1.1 程序设计语言

程序设计语言是人与计算机之间进行信息通信的载体，是用来编写程序的工具。程序设计语言从结构上可分为两类，分别是低级语言和高级语言。

1. 低级语言

低级语言包括机器语言和汇编语言。

(1)机器语言

机器语言由能被计算机直接执行的机器指令组成，每条机器指令是由二进制数字 0、1 代码组成的。

特点：对计算机的依赖性强、直观性差、可读性差，编写程序的工作量大，容易出错，出错后难于调试和修改，只有对相应计算机的结构比较熟悉，且经过一定训练的程序人员才能较好地使用。基于上述原因，人们引入了汇编语言。

(2)汇编语言

汇编语言是符号化了的机器语言,引入一些助记符表示机器指令中的操作码和存储地址。

特点:与机器语言相比,大大提高了编程的速度和准确度。也有很多缺点:不易编写,阅读和理解困难,不便移植,因此人们又引入了高级语言。

2. 高级语言

高级语言是便于理解的自然语言,有几百种之多,但除了一些专用语言之外,得到广泛运用的只有其中少数几种,常用的有 Basic、Fortran、Pascal、C、Java 等。高级语言无论是在算法描述的能力上,还是在编写和调试程序的效率上,都远比低级语言优越。

特点:可读性强,具有通用性,可在不同机器上运行,便于移植,便于编写、调试和修改。

1.1.2 编译程序

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。人们虽然可以直接用机器语言编写程序,但很不方便,这是因为机器语言程序不易读、不易写、结构性很差。另一种方法是人们先用较为接近自然语言的高级程序设计语言(或简称高级语言)来编写程序,再借助特定的软件将它翻译成机器语言程序。

1. 翻译程序(Translator)

将用汇编语言或高级语言编写的程序转换成等价的机器语言程序,执行这种转换功能的程序统称为翻译程序。转换前的程序也就是翻译程序的输入对象称为**源程序**(Source Program),它是用汇编语言或高级语言编写的程序;输出对象称为**目标程序**(Object Program),目标程序可以是机器语言程序、汇编语言程序或自定义的某种中间语言程序。

2. 汇编程序(Assembler)

源程序为汇编语言程序,目标程序为机器语言程序的翻译程序,称为汇编程序。

3. 编译程序(器)(Compiler)

编译程序是将用高级语言编写的程序(源程序)翻译成汇编语言或机器语言形式的程序(目标程序)的一种翻译程序。世界上第一个编译程序 Fortran 于 20 世纪 50 年代中期研制成功,它是现今所有计算机系统中最重要的系统程序之一,即高级语言的翻译程序。

1.1.3 实现高级语言的编译方式

由于计算机硬件只懂自己的指令系统,即只能直接执行相应机器语言格式的代码程序,而不能直接执行用高级语言或汇编语言编写的程序。因此,要在计算机上实现除机器语言之外的任一程序设计语言,首先应使此种语言被计算机所“理解”。解决这一问题的方法有两种:一种是对程序进行编译;另一种是对程序进行解释。

1. 解释方式

解释方式是接受用程序语言(源语言)编写的程序(源程序),然后直接解释执行源程序,如图 1-1 所示。解释器相当于源程序的抽象执行机,是语言的实现系统。其执行过程为一个语句一个语句地读入源程序,边翻译边执行,在翻译过程中不产生目标程序。

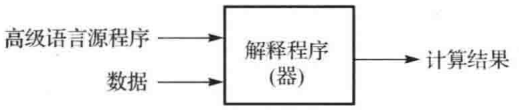


图 1-1 解释方式

2. 编译方式

编译方式下的源程序为高级语言程序，目标语言是低级语言程序(汇编或机器语言程序)的翻译程序。执行过程为对整个源程序进行分析，翻译成等价的目标程序，翻译的同时做语法检查和语义检查，然后运行目标程序。

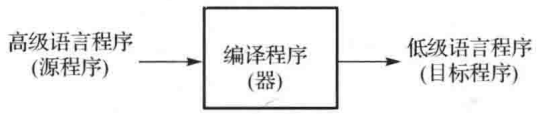


图 1-2 编译方式

3. 编译方式和解释方式的区别

解释方式特别适合于交互式语言，而且解释方式允许程序执行时改变自身，如调试程序。这种情形编译程序不易胜任，因为它需要动态编译，而解释程序可以毫无困难地胜任；此外，解释程序不依赖于目标机(运行编译程序所产生目标代码的计算机)，因为它不生成目标代码，可移植性优于编译程序。但是和编译程序相比，解释程序开销大，运行速度慢。解释方式和编译方式的最根本区别在于：在解释方式下，并不生成目标代码程序，而是直接执行源程序本身。

1.2 编译程序的组成

编译过程是编译程序工作时的动态特征。一个典型的编译程序一般包含以下八个阶段，分别是词法分析、语法分析、语义分析、中间代码生成、中间代码优化、目标代码生成、表处理、错误处理，各阶段之间的关系如图 1-3 所示。

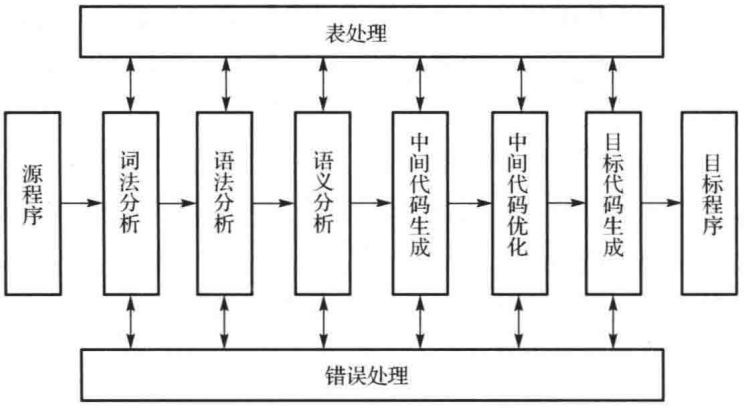


图 1-3 编译器的功能结构图

1.2.1 编译程序的构成

图 1-3 是编译程序总框，编译程序的结构可以按照下列八个阶段的任务分模块进行设计。

1. 词法分析

词法分析(Lexical Analysis)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，称为**单词符号**，简称**符号**。例如保留字(begin、end、for、while 等)、标识符、常数、运算符和界符(左右括号)等。编译器的词法分析也叫作**线性分析**或**扫描**。

词法规则是单词符号的形式规则，它规定了哪些字符串构成一个单词符号。

例如 C 程序语句：

```
a[i] = 4 + 2;
```

这个代码包括了 12 个非空字符，但只有 8 个单词符号：

(1)a：标识符。

(2)[：左括号。

(3)i：标识符。

(4)]：右括号。

(5)=：赋值号。

(6)4：数。

(7)+：加号。

(8)2：数。

每一个单词符号均由一个或多个字符组成，在进一步处理之前它已被收集在一个单元中。

2. 语法分析

语法分析(Syntax Analysis)的任务是在词法分析的基础上，根据语言的语法规则，把单词符号串分解成各类语法单位(根据文法的产生式进行推导或归约)，如短语、子句、句子(语句)、程序段、程序等，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。通过语法分析，确定整个输入串是否构成语法上正确的“程序”，语法分析也称为**层次分析**。

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。

3. 语义分析

语义分析(Semantic Analysis)的任务是首先对每种语法单位进行静态的语义审查(静态语义是指程序在语义上要遵守的规则)，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。同时审查源程序有无语义错误，为代码生成阶段收集类型信息。

4. 中间代码生成

中间代码生成(Intermediate Code Generation)即将源程序转换成一种称为中间代码的内部表示形式。中间代码是一种简单的、含义明确的记号系统。

5. 中间代码优化

中间代码优化(Intermediate Code Optimization)的任务是对前阶段产生的中间代码进行等价的变换或改造，以期获得更为高效的(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。

6. 目标代码生成

目标代码生成(Object Code Generation)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。

7. 表格管理(Symbol-Table Management)

编译程序在工作的过程中需要建立一些表格，以登记源程序中所提供的或在编译过程中所产生的一些信息，各个阶段的编译工作都涉及构造、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。

8. 错误处理

各个阶段还存在错误处理(Error Detection and Reporting)模块，当有错误出现时，由相应的错误处理模块给出解决方案。一个好的编译程序在编译的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，以便用户查找和纠正，因此在编译程序中必须有一个出错处理程序。

1.2.2 遍

具体实现上，受不同源语言、设计要求和计算机硬件条件的限制，往往将编译程序组织成若干遍(Pass)。所谓“遍”，就是对源程序或源程序的中间表示形式从头到尾扫描一次，并进行加工处理，生成新的中间结果或目标程序。既可以将编译过程中的几个不同阶段合为一遍，也可以把一个阶段的工作分为若干遍。例如，词法分析这一阶段可以作为单独的一遍，但更多的时候把词法分析程序作为语法分析程序的子程序来加以调用，将词法分析阶段和语法分析阶段合并为一遍。

1.2.3 编译程序前端和后端

概念上我们常把编译程序划分为编译前端和编译后端。**前端**主要由与源语言有关但与目标机无关的那些部分组成。编译前端通常包括词法分析、语法分析、语义分析、中间代码生成，与目标机无关的中间代码优化部分也包含在前端，当然前端也包括相应部分的错误处理。

编译**后端**包括与目标机有关的中间代码优化部分和目标代码生成等。一般来说，这些部分与源语言无关，而仅依赖于中间语言。很明显，编译后端是面向目标语言的，编译前端则不是，它几乎独立于目标语言。

1.3 编译程序的构造

编译程序是一个复杂的系统程序，它是实现高级语言编程的一个最重要的工具。一般开发编译程序有如下几种可能途径。

1. 转换法(预处理法)

假如我们要实现 L 语言的编译器，现在有 L'语言的编译器，那么可以把 L 语言程序转换成 L'语言程序，再利用 L'语言的编译器实现 L 语言，这种方法通常用于语言的扩充。如对于 C++语言，可以把 C++程序转换成 C 程序，再应用 C 语言的编译器进行编译，而不用重新设计和实现 C++编译器。常见的宏定义和宏扩展都属于这种情形。

2. 移植法

假设在 A 机器上已有 L 语言的编译程序，现在想在 B 机器上开发一个 L 语言的编译程序。这里有两种实现方法。

实现方法一：最直接的办法就是将 A 机的代码直接转换成 B 机的代码。

实现方法二：假设 A 机和 B 机上都有高级程序设计语言 W 的编译程序，并且 A 机上的 L 语言编译程序是用 W 语言写的，我们可以修改 L 编译程序的后端，即把从中间代码生成 A 机的目标代码部分改为生成 B 机的目标代码。这种在 A 机上产生 B 机目标代码的编译程序称为交叉编译程序(Cross Compiler)。

3. 自展法

自展法的实现思想：先用目标机的汇编语言或机器语言书写源语言的一个子集的编译程序，然后将这个子集作为书写语言，实现源语言的编译程序。通常这个过程会分成若干步，像滚雪球一样，直到生成预计源语言的编译程序为止。我们把这样的实现方式称为自展技术。使用自展技术开发编译器时，要求这种高级语言必须是能够编译自身的。

4. 工具法

20 世纪 70 年代，随着诸多种类的高级程序设计语言的出现和软件开发自动化技术的提高，编译程序的构造工具陆续诞生，如 20 世纪 70 年代 Bell 试验室推出的 LEX 和 YACC 至今还在广泛使用。其中 LEX 是词法分析器的自动生成工具，YACC 是语法分析器的自动生成工具。然而，这些工具大都用于编译器的前端，即与目标机有关的代码生成和代码优化部分。由于对语义和目标机形式化描述方面还存在困难，虽研制了不少生成工具，但还没有广泛应用的。

5. 自动生成法

如果能根据对编译程序的描述，由计算机自动生成编译程序，那么将是最理想的方法。但需要对语言的语法、语义有较好的形式化描述工具，才能自动生成高质量的编译程序。目前，语法分析的自动生成工具比较成熟，如前面提到的 YACC 等，但是整个编译程序的自动生成技术还不是很成熟，虽然有基于属性文法的编译程序自动生成器和基于指称语义的编译程序自动生成器，但产生目标程序的效率很低，离实用尚有一段距离，因此，要想实现真正的自动化，必须建立形式化描述理论。

1.4 小 结

本章从总体上概要介绍了编译相关的原理和技术及典型编译器的逻辑结构，使学生对编

译程序有一个初步的认识。本章重点和难点为对各基本概念的理解和对整个编译程序各个阶段所承担任务的理解和掌握。

复习思考题

1．选择题

(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)编译程序的工作过程一般可划分为下列基本阶段：词法分析、_____、代码优化和目标代码生成。

A．出错处理　　B．语法分析

C．表格管理　　D．语义分析和中间代码生成

(4)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

(5)编译程序是_____。

A．应用软件　　B．系统软件

C．操作系统　　D．用户软件

2．判断题

(1)一个程序是正确的是指该程序的语法是完全正确的。

(2)高级语言程序必须经过编译程序的翻译才能被计算机识别和执行。

(3)编译程序的输入是高级语言程序，输出是机器语言程序。

(4)每一个编译程序都由完成词法分析、语法分析、代码优化、代码生成工作等五部分组成。

(5)编译程序生成的目标程序一定是可执行的程序。

3．简答题

(1)高级程序设计语言具有哪些特点？

(2)什么是编译程序？

(3)编译程序在逻辑功能上一般由哪几部分组成？

(4)编译方式和解释方式的根本区别是什么？

第 2 章　形式语言与词法分析

学习目标

正确理解上下文无关文法、语言的形式定义、短语、简单短语、语法树和文法二义性等。系统学习编译过程中词法分析的方法，单词符号的描述工具、表示方法、识别机制及其不同描述方式之间的转换，进而理解词法分析的过程和理论，明确词法分析在编译过程所处的阶段和作用。

学习要求

- 掌握：词法分析程序的手工实现方法，单词符号的描述工具、表示方法、识别机制及其不同描述方式之间的转换。掌握符号串及其运算、上下文无关文法和语言的形式定义。
- 了解：文法的分类，词法分析程序的自动构造。

在对源程序进行的编译过程中，首先要识别出源程序中的**单词符号**，然后分析每个语句的意义并进行翻译。识别单词符号的任务是由词法分析器(lexical analysis)[也称为扫描器(scanner)]来完成的，这里主要介绍词法分析程序的设计原则、单词符号的描述、识别机制及词法分析器的自动构造，并介绍编译理论中用到的有关形式语言理论的最基本概念，包括字母表、符号串、短语、直接短语、句柄、语法树和文法二义性等。

2.1　字母表和符号串的基本概念

通过第 1 章的介绍，我们知道编译程序的功能是将高级语言编写的源程序翻译成与之等价的机器语言或汇编语言的目标程序。也就是说，我们所要构造的编译程序是针对某种程序设计语言的，编译程序要对它进行正确的翻译，首先要对程序设计语言本身进行精确的定义和描述。为了精确地定义和描述程序设计语言，需采用形式化的方法。所谓形式化的方法，是用一整套带有严格规定的符号体系来描述问题的方法。从形式语言的角度来看，任何程序设计语言的程序都可看成一定字符集(称为字母表)上的一个字符串(有限序列)。一个合适的程序至少应满足词法、语法和语义规则。语言的词法规则规定了单词符号的构成形式，语言的语法规则规定了如何从单词符号形成更大的结构(语法单位)。就现今的多数程序语言来说，描述一个程序语言的语法规则的数学工具是上下文无关文法。一个上下文无关文法是给出程序设计语言的精确的、易于理解的语法说明。

正如英语是由句子组成的集合，而句子又是由单词和标点符号组成的序列那样，程序设计语言 Pascal 或 C 语言是由一切 Pascal 或 C 程序所组成的集合，而程序是由类似 if、begin、end 的符号、字母和数字这样一些基本符号所组成的。从字面上看，每个程序是一个“基本符号”串。设有一基本符号集，则 Pascal 或 C 语言可看成是在这个基本符号集上定义的、按一定规则构成的一切基本符号串组成的集合。下面给出符号和符号串的有关概念。