计算机软件技术基础

陈慧南　主编

人民邮电出版社

内 容 提 要

软件技术已成为从事计算机应用的工程技术人员和各类高等工科院校非计算机专业有关计算机应用的一项技术基础，本书主要介绍了有关计算机软件技术的基础知识，包括软件基本概念，离散数学基础，基本数据结构，操作系统基础，软件开发技术和数据库概论等。各章有较丰富的例题并附有习题及实习题。

本书内容丰富，选材较新且有一定深度，可作为高等院校非计算机专业软件技术基础课程的教材或参考书，也可作为广大从事计算机应用工作的教师和工程科技人员自学参考。

计算机软件技术基础

陈慧南 主编

*

人民邮电出版社出版发行

北京崇文区夕照寺街14号

北京顺义振华印刷厂印刷

新华书店总店科技发行所经销

*

开本：787×1092 1/16 1997年7月 第1版

印张：20.25 1997年7月 北京第1次印刷

字数：504 千字 印数：1— 2500册

ISBN 7-115-06197-1/TP·328

定价：26.00元

前　言

随着计算机应用领域的扩大和深入，对于非计算机专业的工程技术人员掌握必要的计算机软件基础知识是提高计算机应用水平的重要途径。因此，高等院校非计算机专业，尤其是工程技术类专业的本科生和研究生掌握一定的软件基础知识是十分必要的。目前在高等院校非计算机专业开设《软件技术基础》课程的做法已越来越普遍。

本书根据国家教委关于全国高等工科院校非计算机专业《软件技术基础》课程的基本要求，以及邮电高校计算机类教指委关于该课程的教学大纲，并参考一些省、市非计算机专业计算机等级考试三级大纲的要求，结合本书作者多年的教学实践编写而成。

本书内容包括：软件概论，离散数学基础，基本数据结构，操作系统基础，软件开发技术和数据库概论等六个部分。各部分既相互独立，又存在必要联系。本书在重点讨论传统的受到广泛认同的软件基本原理、技术、方法和工具的同时，也注意介绍软件学科的新方法和新技术。此外，由于考虑到非计算机专业人员对计算机知识日益增长的需求，在内容选择上既注意覆盖最基本的软件知识，又有一定深度。本书各章都有一定数量的习题和一定数量的实习题。

全书共分六章，由陈慧南主编。第一、五、六章和附录由陈慧南编写，第二章由邱伟星编写，第三章由陈春玲编写，第四章由黄刚编写。全书由陈慧南统一审定。

本书在编写过程中，得到了南京邮电学院教务处，计算机工程系领导的关心和支持，在此表示感谢。本书涉及面广，由于作者水平有限，书中可能会存在不少问题，热忱希望读者指正。

编者

1996 年 2 月

目　录

第一章　软件概论

1.1　软件的分类和发展

1.1.1　软件和软件的分类

计算机资源包括硬件和软件。计算机硬件和软件组成的统一整体称为计算机系统。系统硬件是构成计算机系统所配置的全部设备，如中央处理机(CPU)，内存储器(RAM、ROM)，外存储器(软盘、硬盘等)以及各种输入输出设备(键盘、鼠标、显示器、打印机等)。但是为了充分发挥计算机硬件的使用效益，还需要有软件。所谓软件就是所有程序、数据及其文档的总称。软件可分为系统软件和应用软件两大类。系统软件是由计算机系统的设计者和生产者提供的程序及其使用说明和维护手册的总称。这些程序的作用在于扩充计算机功能，控制计算机高效率运行，并为用户提供更多的方便。系统软件主要包括：操作系统，各种程序设计语言处理程序(汇编程序、解释程序、编译程序)，数据库管理系统，计算机网络软件，诊断和故障处理程序，编辑程序和调试程序等。由于配制了各种系统软件，从而大大改善了用户使用计算机的环境。系统硬件和系统软件组成了计算机系统。这样，用户使用的计算机系统并不是一台仅有硬件的裸机，而是一台功能更强、效率更高、使用更方便的“虚拟计算机”。应用软件则是用户利用计算机系统提供的功能为解决特定问题而编制的程序及其使用说明和维护手册的总称。这类程序的种类繁多，不同的应用领域，不同的部门需要编制不同的应用程序。为减少重复劳动，提高软件重用率，应用软件也在逐步实现商品化，形成各种应用软件包。如各种科学计算软件包，事务管理软件包，辅助教学软件包等等。用户可根据自身需要选购，十分方便。

有时，系统软件和应用软件不能截然分开。如各种标准程序库，既可看成是应用软件，也可看成是计算机厂家提供的系统软件。对于一个使用计算机的工程技术人员来说，熟悉各种系统软件的目的是为了更好地发挥计算机的功能，为了更有效地从更高水平上来开发应用软件，编制应用程序。

1.1.2　软件的发展

一、程序设计语言和语言处理程序的发展

在1946年以来的半个世纪中，计算机硬件已经历了四代更迭：电子管、晶体管、集成电路和大规模集成电路。现代计算机正向着巨型、微型、并行、分布、网络化和智能化几个方面发展。与其相应的，软件也逐步丰富和完善。

在计算机产生的早期，人们直接使用机器语言编写程序。一台计算机有它能够执行的基本操作，这些操作可以用命令形式(一字节或多字节的二进制代码形式)来书写，这就是机器指令。它们是裸机唯一能识别的语句。每台计算机能识别和执行的指令的集合称为该计算机的

指令系统。机器语言即机器指令的集合。用机器语言编写的程序称机器语言程序。使用机器语言编写程序是很不方便的，它要求程序设计人员相当熟悉计算机的所有细节，例如指令系统、存储容量、寄存器的类型与个数等等。这样做工作量大，十分容易出错并且不易修改。又由于各种计算机的指令系统不同，所编程序只适用于特定机型，局限性很大。

为摆脱机器语言编程的困难，使计算机从少数专门人员手中解放出来，成为广大工程技术人员都能使用的工具，也为了减轻程序设计人员编制和调试机器语言程序中的繁重劳动，提高程序设计的效率，出现了以符号指令来编程的办法。用符号语言编制的程序称为符号程序。符号语言使用的指令助记符是指令英文名称的缩写。比如取数用 LD，加法用 ADD 等等。在编程时，只要记住指令助记符就可以了，这比单纯的"0"、"1"代码串容易记忆。这种符号语言的扩充就是汇编语言。用汇编语言编制程序要比用机器的指令代码方便得多，不仅易于检查错误和修改错误，而且指令、原始数据和结果数据的存放单元可由机器根据定位伪指令自动分配。例如，要汇编的源程序如表 1.1 所示。但是计算机内部结构只能识别和执行二进制代码表示的机器指令，不能识别和执行符号指令，因此必须将表 1.1 中的汇编语言源程序"翻译"成机器语言的"目标程序"，计算机才能执行。这个翻译工作由"汇编程序"完成。汇编程序是一种翻译程序，有了它，才允许用户在该计算机上使用汇编语言编制程序。

汇编语言是面向机器的，它与特定的计算机指令系统密切相关，程序员仍必须熟悉机器的硬件结构，因此程序设计仍很繁琐和低效。但是正因为它依赖于机器，熟练的程序员可结合机器特点设计出高质量的程序，所以直至今日，汇编语言仍起着重要作用，常用来实现对时间和空间效率要求较高的计算机系统核心程序和实时控制程序等。

高级语言是由表达各种意义的"词"和"数学公式"按照一定的"语法规则"组成的面向问题的语言。用高级语言进行程序设计比较接近人的习惯，编出的程序与具体的机器指令无关，可以独立于机器，通用性较强。如同汇编语言一样，用高级语言编写的源程序，也必须翻译成目标程序后才能在计算机上执行。这种翻译程序分两类：一类是解释程序，如 BASIC 语言解释程序。它先将源程序"扫视"一遍，然后一句句翻译，每译完一句，就执行一句。另一类是编译程序，如 FORTRAN、PASCAL 和 C 语言编译程序，它们将源程序完全翻译成目标程序后再交计算机执行。

目前，世界上有许许多多程序设计语言，较通行的也有好几十种。例如科学与工程计算语言 FORTRAN，会话型语言 BASIC，教学和系统程序设计语言 PASCAL，数据处理语言 COBOL，人工智能中使用的表处理语言 LISP、逻辑程序设计语言 PROLOG，大型通用语言 PL/1 等等。C 语言是一种结构化、模块化、可编译的通用程序设计语言，被广泛用于开发系统程序和应用程序。ADA 是美国国防部开发的语言，它支持抽象数据类型的概念。Smalltalk，C++和 Actor 等面向对象程序设计语言，近年来引起很大重视。

随着程序设计语言的发展，解释程序和编译程序等语言处理系统也应运而生并日趋完善。其中编译理论和技术是计算机科学中发展最为迅速的分支之一，现已形成了一套比较系统的理论和方法，能指导人们更好地设计和构造编译程序。

二、操作系统的发展

操作系统是计算机系统软件中另一个迅速发展的分支，它的发展与计算机硬件的发展以及计算机操作使用方式的改进有着密切的关系。

为了充分发挥计算机所具有的高工作效率，首先要处理好人对机器过多的干预或手动操作过多的问题，降低计算机的闲置率，这要有一个管理软件，它具有接受和处理用户所提交的

作业的功能。其次，要处理好中央处理机与外部设备在速度上快慢不匹配的问题，要求管理软件具有处理中断的功能，并提供一种简便的统一的使用外部设备的手段，统一管理外部设备的输入和输出。第三，为充分发挥计算机各种资源的作用，管理软件应具有能处理内存中多道作业的能力。几道程序共享系统资源，可使 CPU、内存及外部设备得到更加充分的利用，这就是多道批处理和分时系统的基本思想。

总之，要使计算机所有资源（包括中央处理机、存储器、各种外部设备和软件）协调一致，有条不紊地工作，并给用户提供更大的方便，必须要有一个自动化的管理机构，这就是操作系统。

操作系统的发展通常概括为五个阶段：

(1)手工操作阶段

(2)早期批处理阶段

(3)执行系统阶段

(4)多道批处理系统阶段

(5)操作系统的形成和发展

有关操作系统的详细内容将在第四章中介绍。多道批处理系统和分时系统的出现，标志着操作系统的正式形成。以后又出现了分布式操作系统和网络操作系统。

操作系统的出现是软件发展的一个重大转折，也是计算机系统发展的一个重大转折。操作系统本身是一个大程序，在它的控制下，计算机系统的每个部件（不仅是硬件部件，也包括软件部件）最大限度地发挥作用，因此，操作系统是软件系统的核心。

目前，操作系统已进入总结提高阶段，形成了几类较成熟的系统。UNIX 是当今最流行的多用户分时操作系统；PC-DOS 和 Windows 成为个人计算机上占优势的单用户操作系统。操作系统的理论研究仍在深入进行，如进程通讯、关于死锁和虚拟存储、共行的语言机制、分布式操作系统和网络操作系统的有关问题以及操作系统结构设计工具等。

三、计算机网络软件和数据库软件

计算机网络是计算机技术和通信技术高度发展和密切结合的结果。可以认为，网络软件是更高水平上的操作系统。它利用通讯线路把分布在不同地点上的多个独立的计算机系统连成一个网络，使网上用户能够实现数据传送，共享网络中所有硬件、软件和数据等资源。按照计算机网络的跨越距离，又可分成远程网和局部网。远程网如美国的 ARPA 网，局部网如 Ethernet 网等。

随着计算机广泛应用于国民经济的各领域、各部门，数据存储、数据处理的要求越来越高，因而在操作系统的支持下，建立和发展了各种类型的数据库系统软件。数据库管理系统是数据管理技术的新成果，它是数据管理经过人工操作和文件系统两个阶段后发展起来的。数据库中数据是从整体角度组织的，可供多个用户共享，具有低的数据冗余度。数据库管理软件起着用户或应用程序与总体数据库之间的接口作用和数据的统一管理作用。用户可以用查询语言或终端命令操作数据库，也可以用程序方式操作数据库。数据库管理软件为应用程序和数据库数据间提供了独立性。同时还提供了数据保护和并发控制功能等。有关数据库技术的内容，将在第六章中介绍。

四、用户界面工具和 CASE 工具

近年来，大量的应用软件以用户界面技术为基础，朝着图形化、菜单化和多窗口化方向发展，形成了新的学科分支——人—机通信，产生了许多用户界面开发工具，发展为用户界面管理系统(UIMS)。用户界面管理系统有助于实现交互应用软件中界面部分和计算部分的相对

独立。这是一种新的应用程序的开发方法，如果说操作系统将应用程序中与计算机资源管理有关的部分分离出来，由操作系统统一管理，那么，用户界面管理系统的目的在于实现应用程序中界面(对话)与计算(应用本身)的分离，即将涉及最终用户的输入处理，输出显示等人机对话与应用系统的计算部分分离。

计算机辅助软件工程(CASE)是软件开发和维护的新技术，也是一种新的软件工具。我们熟悉的微机操作系统 Windows 和 Turbo Pascal、Turbo C 语言集成环境所提供的程序设计环境主要集中在编码实现阶段。CASE 是一种通用的软件支持环境，为软件开发的全过程提供全面的支持。

1.2 程序设计和软件开发环境

1.2.1 程序设计方法和软件工程

由于早期的计算机速度低、内存小，评价程序质量的标准是正确性和时间、空间效率。程序员在编程时，为追求效率，总是以算法设计为中心，十分注重技巧，程序设计通常是个体劳动或少数人的合作劳动。程序清单常常是唯一的文档。

随着计算机应用领域的扩大，计算机应用从主要是科学计算发展到图象、语言、表格、声音等各种信息处理。这些非数值数据往往量大且结构复杂。计算机处理的对象改变了，程序设计也发生了变化，仅研究程序设计技巧已不能设计高效程序，必须研究数据的结构。

随着硬件的发展，计算机速度和存储容量迅速增加，程序的规模也越来越大。一方面，因程序的效率引起的矛盾减小，另一方面，设计、验证、维护和修改程序的代价急剧上升。为提高可靠性和节省开发、维护费用，必须重视程序的可维护性标准(包括可读性、可修改性、可测试性)。人们认识到，程序不仅是机器执行的，它是供人阅读的“文章”，它在程序开发和运行维护各阶段上由各类人员阅读。为此，人们提出了结构化程序设计方法。这是一种最早提出的程序设计方法，它采用自顶向下、逐步求精的设计方法和顺序、选择、循环三种基本控制结构构造程序。

结构化程序设计是一种成功的程序设计方法，但不能解决一个大系统的系统结构的设计问题，更不能解决大系统的总体模型的表达问题。为解决这些问题，产生了软件工程。软件工程研究的重点是如何用工程的方法开发大型软件。程序设计方法是软件工程学的一大支持。有关软件工程的知识将在第五章介绍。

1.2.2 软件开发环境

从前面对软件发展过程的简单介绍中，我们看到，最初的程序设计环境十分简陋。程序员面对的是功能十分简单的裸机，程序员用纸、笔和机器指令编写程序，程序设计效率极低，可靠性难以保证。

以后出现了汇编语言和汇编程序，还有了输入输出控制系统，程序员有了简单工具，程序设计环境虽有一定改善，但并没有本质变化。

50 年代末、60 年代初，计算机系统性能有了突破性发展，硬件功能更强、速度更快、存储容量大大增加，有了操作系统和各种高级语言及其语言处理程序，从而形成了较好的编程环境。

70年代以后，程序设计方法学和软件工程技术不断发展，提出了一系列先进的、有效的软件开发和维护的概念、方法、技术和工具，如数据抽象，信息隐蔽的原则，模块化局部化思想，结构化设计方法，面向数据和面向对象的设计方法，文档编制技术，测试技术，图形技术和数据库技术等等。产生了许多支持上述方法和技术的工具、工具箱和集成环境，如编辑、编译、调试工具，图形系统和窗口系统，还有在此基础上发展起来的CASE工具和工具箱。它使软件开发向自动化道路迈出了新的一步。

1.3 语言处理程序

如前所述，人们用汇编语言或高级语言编写的源程序，必须经过翻译成为机器语言程序后才能执行。由翻译所得的结果称为目标程序。目标程序可以用机器语言表示，也可以用汇编语言或其它中间语言表示。语言处理程序提供这种翻译功能。

1.3.1 汇编程序

汇编程序的主要功能是把汇编语言源程序翻译成机器能识别的目标程序(图1.1)。源程序通常是由ASCII码表示的符号化的指令串组成。当汇编程序加工源程序时，总是从头到尾一个符号接着一个符号地阅读，称为扫描源程序。从头到尾扫描一次源程序为扫描一遍。

具体地说，汇编程序必须完成几方面任务：

(1)处理语句中的操作符，生成与之相应的机器码，产生机器指令；

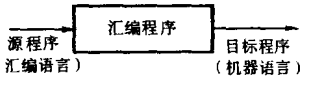

图1.1 汇编过程

(2)处理语句中的标号和名字，并代之以具体的单元地址；

(3)处理伪指令。

为了完成表1.1中汇编源程序的翻译工作，需要以下表格：

(1)机器码操作表(MOT)：用以确定指令的长度和把助记符转换成机器码(表1.2)；

(2)伪指令操作表(POT)：用以查找伪指令对应的操作(表1.3)；

(3)地址计数器(LC)：用以追踪和确定指令地址；

(4)符号表(SYMT)：用以记录各标号、名字(表1.4)。

借助上述表格，我们用手工方式模拟源程序被汇编的过程(表1.5)。

表1.1 汇编源程序举例

行号	标号	操作码	操作数
1		ORG	100 H
2	START:	LD	HL,DATA
3		XOR	A
4		LD	B,COUNT
5	LOOP:	ADD	A,(HL)
6		INC	HL
7		DJNZ	LOOP
8		LD	(RESULT),A

续表

行号	标号	操作码	操作数
9		NOP	
10		HALT	
11	DATA:	DB	1,2,3,4,5
12	COUNT:	EQU	$-DATA
13	RESULT:	DS	
14		END	

表 1.2　　MOT 表

指令助记符	机器码(16 进制)	指令长度(字节)
LD HL,nn	21	3
XOR A	AF	1
ADD A,(HL)	86	1
LD B,n	06	2
INC HL	23	1
DJNZ,e	10	2
LD (nn),A	32	3
HALT	76	1
·	·	·
·	·	·
·	·	·

表 1.3　　POT 表

伪指令码	处理伪操作的程序的入口地址
EQU	PIEQU
DB	PIDB
DS	PIDS
ORG	PIORG
·	·
·	·
·	·

第 1 行伪指令 ORG 100H，指出起始单元地址为 100H，即 START 的值为 100 H。第 2 行为指令，经过查 MOT 表，知其为 3 字节指令，操作码为 21，后两字节是立即数，其值是符号地址 DATA 的值，但至此还不知 DATA 的值，故暂先空出 2 字节。第 3 行是指令，其首地址可以从上一条指令的首地址及指令长度求得，所以其地址为 103，其操作码为 AF，指令长度为 1 字节；……。按同样方法，可以得到每条指令的首地址和操作码，并把处理(扫描)中遇到的符号

表 1.4　　SYMT 表

符号	值	注释
START	0	R
LOOP	6	R
DATA	F	R
COUNT	5	A
RESULT	13	R

其中 R 表示浮动地址，A 表示绝对地址

表 1.5　　手工汇编过程举例

行号	地址	代码	源程序		
1				ORG	100 H
2	100 H	21　0F　01	START:	LD	HL,DATA
3	103 H	AF		XOR	A
4	104 H	06　05		LD	B,COUNT
5	106 H	86	LOOP:	ADD	A,(HL)
6	107 H	23		INC	HL
7	108 H	10　FC		DJNZ	LOOP
8	10 AH	32　14　01		LD	(RESULT),A
9	10 DH	00		NOP	
10	10 EH	76		HALT	
11	10 FH		DATA:	DB	1,2,3,4,5
12		0005 H	COUNT:	EQU	$-DATA
13	114	1	RESULT:	DS	1
14				END	

(START、LOOP、COUNT、DATA、RESULT)填入符号表(SYMT，表 1.4)中。第 7 行 DJNZ LOOP 为相对转移指令，其偏移量 e 要经过计算才能得到(e＝LOOP－PC＝106－10A＝－4，其补码为 FC)。第 11 行 DATA：DB 1，2，3，4，5 是一条伪指令，由上一条指令可知，其首地址为 10 FH，伪操作 DB 的功能是定义 1～5 五个整数，并把它们依次存放在 DATA 单元开始的地址单元中。第 12 行伪指令定义标号 COUNT，其本身不占内存单元，但要求计算 EQU 右边的表达式 $－DATA，$ 是程序计数器的当前值。在本例中为 114 H，因 DATA 是 10 FH，所以 COUNT＝114 H－10 FH＝005 H。第 13 行伪指令定义存储空间为 RESULT 保留一个存储单元，其地址为 114 H。

由上述手工汇编过程可以看到，DATA、COUNT、RESULT 等符号地址的值在第一次扫描时还不能确定下来，只有在第一次扫描结束后，才能把程序中所有标号的值确定下来。因此需要进行第二遍扫描，以便将符号地址的值代到相应的标号中。注意在将 DATA 用 10 FH 代入时，注意低位字节在前，高位字节在后，因它是立即数。第一遍扫描将生成表 1.4 所示的符号表，此符号表在第二遍扫描中使用。

同手工模拟一样，汇编程序采用对源程序从头到尾的两遍扫描完成整个源程序的汇编。第

一遍扫描确定符号；第二遍扫描确定机器码(表 1.6)。

第一遍扫描过程要完成的任务为：

(1)确定每条指令的长度(查 MOT 表)；

(2)追踪地址计数器(LC)；

(3)建立符号表(由 LC 值确定每个标号的值)；

(4)处理伪指令操作码(如 EQU、DB、DS)。

第二遍扫描过程要完成的任务为：

(1)查符号的值；

(2)产生机器指令；

(3)确定伪指令 DB 定义的数据；

(4)处理伪指令操作码 DB。

此外汇编程序还完成一些附加的功能：

(1)查错：当用户写的汇编源程序不符合汇编语言所要求的书写格式，不符合语法要求时，汇编程序指出源程序的出错位置和错误性质。

(2)修改：汇编程序提供修改源程序的简便方法，用户把修改要求提供给汇编程序，由汇编程序实现对源程序的自动修改。

(3)打印：在汇编过程中，当发现错误时，将错误信息打印出来，必要时可打印出名字表及目标程序，还可打印出修改后的源程序文本等。

表 1.6　　两遍扫描汇编举例

源程序行号	第一遍扫描		第二遍扫描	
	LC	长度	LC	机器指令
1				
2	100 H	3	100 H	21　0F　01
3	103	1	103	AF
4	104	2	104	06　05
5	106	1	106	86
6	107	1	107	23
7	108	2	108	10　FC
8	10 A	3	10 A	32　14　01
9	10 D	1	10 D	00
10	10 E	1	10 E	76
11	10 E	5	10 E	01　02　03　04　05
12	114			
13	114	1		
14				

1.3.2　编译程序

用高级语言写的源程序可以通过编译途径将其翻译成语义上等价的目标程序。目标程序可以由机器语言、汇编语言或某种中间语言表示。如果目标程序是用汇编语言表示的，则它必

须再经汇编程序汇编成机器语言程序；如果目标程序是用某种中间语言表示的，则该目标程序可以进而由下面讨论的解释器解释执行或再经编译后执行。例如 PASCAL P_编译器所产生的中间代码叫 P_代码，这种 P_代码程序可以解释执行也可以再编译后执行。

编译途径一般可分为三个阶段，即编译阶段、连接装配阶段和运行阶段。在编译阶段，由编译程序扫描源程序并将其翻译成目标程序。当编译程序的输出是待装配的目标程序模块时，需经过连接装配阶段，即由连接装配程序把目标程序(模块)以及必须的运行子程序连接起来形成可执行的机器代码(图 1.2)。

一般地将编译过程分成五个阶段：词法分析，语法分析，语义分析和中间代码生成，代码优化，目标代码生成(图 1.3)。

此外，编译过程中还有两部分工作：一是建表和查表工作，二是进行出错处理。图 1.4 是编译程序的一种典型的逻辑结构。

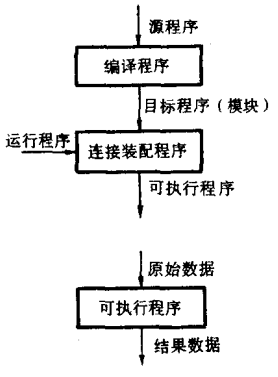

图 1.2 编译途径

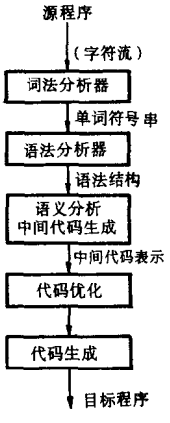

图 1.3 一般编译过程

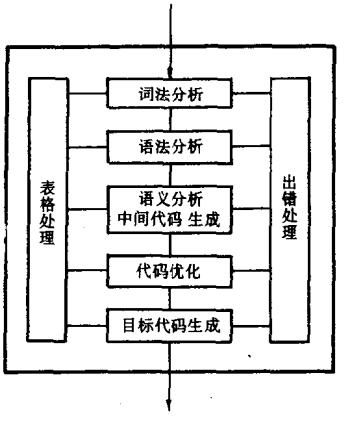

图 1.4 编译程序逻辑结构

1. 词法分析

词法分析程序又称扫描器。它对源程序进行自左向右扫描，识别出一个一个单词(保留字、标识符、常数、运算符和界限符等)，并将它们转换成相应的机内表示，从而形成相应的单词符号串(也报告词法错误)。

请看下面一段简单的 ALGOL 源程序。

```
begin
  real  R,H,S;
  S:=2*3.1416 * R * (H+R)
end
```

这是一个由 38 个字符组成的字符串(begin 和 real，real 和 R 之间各有一个空格)：

```
begin  real  R,H,S;S:=2*3.1416*R*(H+R)end
```

词法分析程序首先从左到右扫描此字符串，并根据词法规则，将其分解成一个个具有独立语法意义的单词符号(也称“单词”或“符号”)，如保留字(begin，real，end)、标识符(R，H，S)、常数(2，3.1416)、运算符和分界符(，　；　:=　+　(　)　*)等。这些单词是组成语句的基本符号。经词法分析，将上述源程序转换成由 22 个单词符号组成的单词符号串，它们通常由特定的内部形式表示。为此，我们可将单词分类，如分为表 1.7 中的四种类别，每种类别由一个整数表示。每个单词也给定一个内部编码(内部值)，如表 1.8 所示。