

李芳芸 沈被娜
王选民 编著

计算机软件技术基础

清华大学出版社

计算机软件技术基础

李芳芸 沈被娜 王选民 编著

清华大学出版社

内 容 简 介

本书是非计算机专业用计算机基础教育教材之一。本书从计算机使用者、各类应用软件设计及开发人员的实际需要出发，用较短的篇幅系统地介绍了计算机软件技术的基础知识，包括软件基本概念、数据结构、操作系统、计算机网络、数据库系统和应用软件开发技术等。每部分既有基本原理的叙述，又有常用实例的介绍。最后还介绍了适用于各种领域的功能很强的PASCAL语言。

本书主要读者对象为非计算机专业的大学本科生、研究生和从事计算机应用的科技人员。

计 算 机 软 件 技 术 基 础

李芳芸 沈被娜 王选民 编著

责任编辑 蔡鸿程

★

清华大学出版社出版

北京 清华园

昌平区振南排版厂排版

保定市科技印刷厂印刷

新华书店北京发行所发行

★

开本：787×1092 1/16 印张：20 字数：524千字

1987年10月第1版 1989年4月第2次印刷

印数：15001—30000 定价：4.70元

ISBN 7-302-00108-1/TP·43

前 言

近几年来我国出现了使用计算机的热潮。过去，计算机多用于科研和教学，现在已迅速普及到生产、管理和社会生活的各个领域。今天，要求计算机使用者不仅要掌握计算机的基本原理、汇编语言和某些高级语言，还要求具有一定的计算机软件基础知识，以提高应用软件方面的研制能力。

本书是在清华大学自动化系试用教材“计算机软件技术基础”的基础上扩充和改编的，适用对象为非计算机专业的大学本科生及研究生。内容包括：计算机软件的基本概念及其发展，软件的基础数据结构原理、基本算法、应用实例和 PASCAL 语言简介，操作系统的基本原理和实例——CP/M 及 UNIX，计算机网络的基本概念和实例——DECNET 及 ETHERNET，数据库系统的基本概念、设计方法和数据库管理系统的实例——dBASE II 及 DATATRIEVE-11，并介绍了用软件工程概念开发应用软件的分析、设计、编程、调试技术。

教材的第一章、第五章、第六章由李芳芸编写，第二章、第七章由王选民编写，第三章、第四章由沈被娜编写。

计算机软件涉及的内容非常多，我们仅从应用的角度加以介绍，其中每个部分都是计算机科学中的一门重要学科。由于我们编写这样一本综合性软件教材是一次尝试，缺点和问题在所难免，望广大读者对本教材提出宝贵意见。

在出版本教材的过程中受到清华大学自动化系系统模拟实验室全体同志和清华大学出版社本书责任编辑的大力协助，在此表示感谢。

编者

1986.7.20

目 录

第一章 概论	1
§ 1.1 计算机软件的发展.....	1
1.1.1 汇编语言的出现.....	2
1.1.2 高级语言的出现.....	3
1.1.3 操作系统的形成.....	3
1.1.4 计算机网络软件、数据库软件的出现.....	4
§ 1.2 计算机软件.....	5
1.2.1 汇编程序.....	6
1.2.2 各种高级语言.....	7
1.2.3 编译程序和解释程序.....	8
1.2.4 各种标准程序库.....	9
主要参考书.....	10
第二章 数据结构	11
§ 2.1 什么是数据结构.....	11
§ 2.2 线性表结构.....	12
2.2.1 线性表.....	12
2.2.2 栈.....	14
2.2.3 队.....	16
2.2.4 线性链表.....	18
§ 2.3 树结构.....	20
2.3.1 什么是树.....	20
2.3.2 二叉树.....	22
2.3.3 二叉树的遍历.....	24
2.3.4 树的应用.....	26
§ 2.4 检索.....	31
2.4.1 线性检索.....	31
2.4.2 对半检索.....	32
2.4.3 分块检索.....	34
2.4.4 散列检索.....	35
§ 2.5 排序.....	39
2.5.1 选择排序.....	39
2.5.2 冒泡排序.....	40
2.5.3 线性插入排序.....	41

2.5.4 对半插入排序.....	42
2.5.5 快速排序.....	43
主要参考书.....	44
第三章 操作系统.....	45
§ 3.1 绪论.....	45
3.1.1 操作系统的任务和功能.....	45
3.1.2 操作系统的发展过程.....	46
3.1.3 一些基本概念和术语.....	50
§ 3.2 存储管理.....	53
3.2.1 存储管理的功能.....	53
3.2.2 存储管理方式.....	54
3.2.3 存储管理的发展趋势.....	64
§ 3.3 CPU管理.....	65
3.3.1 概述.....	65
3.3.2 作业管理.....	66
3.3.3 进程管理.....	69
3.3.4 交通控制.....	70
§ 3.4 设备管理.....	77
3.4.1 概述.....	77
3.4.2 设备管理程序.....	80
3.4.3 虚拟设备——假脱机系统.....	82
§ 3.5 文件管理.....	84
3.5.1 概述.....	84
3.5.2 文件结构及文件目录管理.....	86
3.5.3 文件的使用及文件系统的一般模型.....	89
§ 3.6 实际操作系统介绍.....	93
3.6.1 CP/M 操作系统.....	93
3.6.2 UNIX 操作系统.....	102
主要参考书.....	117
第四章 计算机网络.....	118
§ 4.1 概述.....	118
4.1.1 计算机网络的形成过程.....	118
4.1.2 计算机网络的组成.....	121
§ 4.2 网络的拓扑结构和访问控制方式.....	122
4.2.1 网络拓扑结构.....	122
4.2.2 网络中信道访问控制方法.....	123
§ 4.3 网络的体系结构网络协议层.....	124
4.3.1 网络层次模型.....	124

4.3.2 链路级协议	125
4.3.3 主机之间通信协议	126
4.3.4 应用级协议	127
4.3.5 开放系统互连网络基准模型(OSI)	128
§ 4.4 网络环境下的操作系统	129
4.4.1 概述	129
4.4.2 通信方式与中断处理程序	130
4.4.3 进程管理	130
4.4.4 通信软件和网络软件	131
§ 4.5 DECnet 远程网络介绍	132
4.5.1 概述	132
4.5.2 DIGITAL 网络体系结构	133
4.5.3 逻辑链路	135
4.5.4 任务间通信	138
4.5.5 远程文件访问	147
4.5.6 DECnet 的终端功能	150
§ 4.6 Ethernet 局部网络介绍	151
4.6.1 概述	151
4.6.2 Ethernet 结构方式	151
4.6.3 Ethernet 网络控制策略	153
4.6.4 Ethernet 网络通信协议	154
4.6.5 Ethernet 信息交换过程	156
4.6.6 Ethernet 功能介绍	157
主要参考书	160

第五章 数据库系统 161

§ 5.1 绪论	161
5.1.1 什么是数据库系统	161
5.1.2 数据描述	163
5.1.3 三类数据模型	165
5.1.4 数据库管理系统	168
5.1.5 用户访问数据的过程	169
5.1.6 关系代数	170
§ 5.2 数据库设计	174
5.2.1 设计过程	174
5.2.2 概念结构设计	176
5.2.3 逻辑结构设计	182
5.2.4 物理结构设计	191
5.2.5 数据字典、数据安全性、保密性及完整性问题	193
§ 5.3 微型机数据库管理系统 dBASE I 简介	195

5.3.1	dBASE II 的特点	195
5.3.2	dBASE II 的系统结构	195
5.3.3	dBASE II 使用环境及限制	196
5.3.4	dBASE II 命令	197
5.3.5	dBASE II 命令文件	208
5.3.6	dBASE II 与其它高级语言接口	212
5.3.7	dBASE II 文件类型	212
§ 5.4	小型机数据库管理系统 DATATRIEVE-11	213
5.4.1	DATATRIEVE-11 简介	213
5.4.2	DATATRIEVE-11 命令语句	215
5.4.3	报表生成系统	221
5.4.4	描述表的使用	224
5.4.5	口令表的使用	226
5.4.6	DATATRIEVE 的层次结构和视图的使用	228
5.4.7	DATATRIEVE-11 间接命令文件	230
§ 5.5	数据库系统发展方向和分布式数据库系统	234
5.5.1	概述	234
5.5.2	分布式数据库系统	236
	主要参考书	241
第六章	应用软件开发技术	242
§ 6.1	绪论	242
6.1.1	软件工程学的概念	242
6.1.2	用软件工程技术进行应用软件开发	243
6.1.3	应用软件开发的原则和方法	244
§ 6.2	结构化分析方法	247
6.2.1	分析阶段的任务	247
6.2.2	结构化分析的基本思想和步骤	247
6.2.3	数据流图	248
6.2.4	数据词典	249
6.2.5	小说明	250
§ 6.3	结构化设计方法	253
6.3.1	设计阶段的任务	253
6.3.2	结构化设计的基本思想和原则	253
6.3.3	结构化设计的步骤	255
6.3.4	数据结构方法	258
6.3.5	Parnas 方法	261
§ 6.4	结构化编程方法	262
6.4.1	编程阶段的任务	262
6.4.2	结构化程序设计	262

§ 6.5 测试方法.....	264
6.5.1 测试阶段的任务.....	264
6.5.2 测试过程.....	266
6.5.3 白盒法.....	268
6.5.4 黑盒法.....	271
主要参考书.....	275
第七章 PASCAL语言简介.....	276
§ 7.1 PASCAL语言概述.....	276
§ 7.2 PASCAL程序的一般结构.....	276
§ 7.3 PASCAL语言的词汇.....	277
§ 7.4 数据和数据类型.....	278
§ 7.5 标准函数.....	285
§ 7.6 语句.....	288
§ 7.7 过程和函数.....	297
§ 7.8 程序举例.....	300
主要参考书.....	311

第一章 概 论

§ 1.1 计算机软件的发展

自 1946 年世界上出现了第一台电子数字计算机以来, 仅仅四十年时间, 计算机系统得到飞速发展。人们习惯按元件工艺的演变将计算机的发展划分为四个阶段, 或称四个时代。

第一代电子管时代 (1945-1959 年), 计算机线路上的主要元件为电子管, 存储器为水银延迟线或静电屏。

第二代晶体管时代 (1959-1965 年), 计算机的主要逻辑元件为晶体管, 存储器由磁芯构造。

第三代集成电路时代 (1965-1971 年), 计算机主要逻辑元件为集成电路, 存储器除磁芯外还增加了镀线。

第四代大规模集成电路时代 (自 1971 年开始), 大规模集成电路的集成度超过 100 个门电路, 每平方英寸包含 50000 个元件以上, 存储器采用 MOS 和双极型元件。

现在正在研制更新型的第五代计算机 FGCS(Five Generation Computer System), 计算机向智能化方向发展。具有人工智能的知识信息处理、非数值运算的高性能计算机, 即是第五代计算机。FGCS 超越于现代计算机的根本点在于具有推理能力, 其主要功能是问题求解和推理, 知识库管理和智能接口。其关键技术包括超大规模集成电路 (VLSI) 结构、并行处理方法、逻辑程序设计、基于关系数据库的知识库、以及应用人工智能和模式处理。

日本第五代计算机方案的目标是替广泛用途的知识工程设计和制造出计算机硬件和软

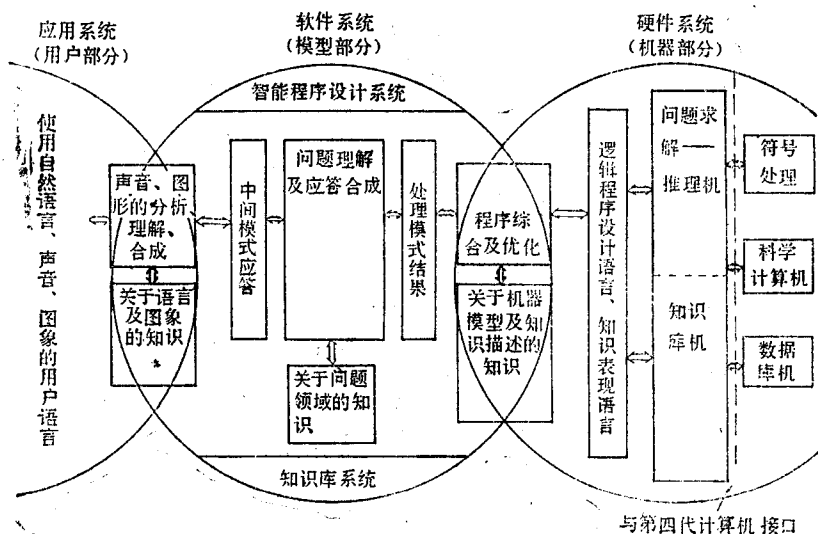


图 1.1

件,包括专家系统、能理解自然语言的机器以及机器人。能支持庞大的知识库,能迅速做相联检索,能进行逻辑推理操作,在程序结构和软件中利用并行方式而达到高速运行,而且做到机器和使用者之间的通信,以便可以使用自然语言和图象。

第五代按照三个子系统组织工作,一是管理知识库的子系统,第二是从事问题解决和推理子系统,第三是人与机器之间的交互方法的子系统。

图 1.1 表示第五代计算机系统的硬件、软件及与人的关系。

随着计算机硬件技术的不断发展及广泛使用,软件也逐步丰富与完善,而软件的发展又大大地促进了硬件的进展。

所谓软件应是程序的集合,这种程序不是用户为解决某一个具体问题而编制的程序,这种程序具有支持计算机工作和扩大计算机功能的作用。

软件有四个发展阶段:

1.1.1 汇编语言的出现

开始,一台计算机中没有装入任何软件,我们称它为裸机,裸机只认识“0”,“1”两种代码,程序设计人员只能用机器指令或称机器语言来编写程序,这就要求程序设计人员熟记计算机的全部指令,工作量大,易于出错且不易于修改。又由于每种计算机的机器指令不相同,所编制的程序只适用于某一特定的机器,局限性很大。因此在早期的计算机中,采用了建立标准子程序或标准程序库的办法,这虽然在一定程度上缓和了工作量大的矛盾,但不能从根本上解决问题。为了摆脱用机器指令编码的困难,出现了用指令符号来编制程序的办法。用符号语言编制的程序称为符号程序,在编制程序时,只要记住指令的助记符就可以了,指令助记符是指令英文名称的缩写,较指令编码容易记忆。例如取数用 LDA、加法用 ADD 等。这种符号语言的扩大就是汇编语言。

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。例如国产小型机 DJS-130 上用机器语言和汇编语言编制的计算 $Z = X + Y$ 程序,如表 1-1 所示。

表 1.1

机器语言程序		汇编语言程序		
地址码	指令代码	地址标号	指令符号	注 解
			•LOC 101;	定位伪指令程序从101号地址开始存放
101	920108	START;	LDA 0,CN01;	N→L0
102	624107		LDA 1,CN02;	Y→L1
103	107099		ADD 0,1;	X+Y→L1
104	044119		STA 1,CN03;	Z=N+Y→(CN03)
105	963077		HALT;	停机
106	X	CN01;	X;	原始数据
107	Y	CN02;	Y;	原始数据
110	Z	CN03;	Z;	计算结果
			•END;	程序汇编结束

不过，计算机的内部结构是根据指令代码设计的，它只能“识别”和“理解”用二进制代码表示的机器指令，不能“识别”和“理解”指令助记符。因此，人们用汇编语言编出计算题后，必须将此程序“翻译”为机器语言程序，机器才能执行，并算出结果。这个翻译工作是由预先装入计算机中的“汇编程序”完成的，汇编程序是一种“编译”程序，计算机有了它，才能允许用户在该计算机上使用汇编语言编制程序，因此汇编程序是计算机必不可少的软件。机器语言程序是它的工作目的，称为目标程序（或目的程序），而用汇编语言编制的程序称为汇编语言程序（或汇编源程序、源程序）。其过程如图 1.2 所示。

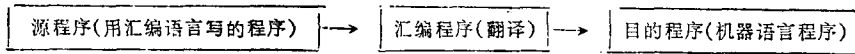


图 1.2

1.1.2 高级语言的出现

采用汇编语言编制程序，仍要记住机器指令的助记符，且所编的程序只针对某一类机器。为解决这些问题，出现了高级语言程序，它使得程序编制工作从程序设计员手中解放出来。高级语言的出现大大方便了程序的设计工作，也促进了计算机的应用，使得计算机的使用范围由计算机专业人员扩大到了各行各业。

高级语言也称为程序设计语言，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。通常把用程序设计语言编制的程序称为源程序，而计算机进行算题仍要根据目的程序来进行。这中间也如同汇编程序一样，要有一个“编译程序”，把源程序“翻译”成目的程序。因此在计算机中要先放好“编译程序”。这种翻译过程有两大类，一是解释程序，如 BASIC 语言解释程序，是先将源程序“扫视”一遍，然后一句句翻译成目的程序，每译完一句，就执行一句，当源程序翻译完了，目的程序也就执行完了。另一类是编译程序，如 FORTRAN 语言编译程序，是将源程序完全翻译为目的程序后，再交计算机执行。

这种程序设计的方法比较接近于人们的习惯，编出的源程序与具体机器的指令无关，可适用于任何机器，因此人们把程序设计语言称为高级语言。

目前，世界上有着几百种程序设计语言，较通用的也有几十种，例如科学与工程计算的语言 FORTRAN，风格严谨的 ALGOL 60 语言，数据处理语言 COBOL，系统程序设计语言 PASCAL，人工智能语言 LISP，大型通用语言 PL/1 和 ACGOL 68，会话式语言 BASIC 此外还有各种模拟语言、数据库语言等。

1.1.3 操作系统的形成

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，并要降低计算机的闲置率，这就要求用户组织好交给计算机来完成的每个作业。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围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，应提供一种简便的、统一的调用外围设备的手段，统一管理外围设备的输入和输出。

第三点是给多个用户提供使用计算机的方便，给中央处理机设置多个终端。要有一个管理软件它具有能处理来自不同终端的多道作业的功能。为了处理多道作业，可采用分时方

法。

总之要使计算机所有资源（包括中央处理机、存储器、各种外部设备和各种软件）协调一致、有条不紊地工作，就要靠操作系统来进行统一调度和管理。

操作系统的发展通常概括为五个阶段：

- 手工操作阶段；
- 早期批处理阶段；
- 执行系统阶段；
- 多道程序系统阶段；
- 分时系统阶段。

每一阶段的特点见第三章所述，多道程序系统和分时系统的出现，标志着操作系统的正式形成。以后又出现实时系统及网络操作系统等。

操作系统的出现是软件发展的一个重大转折，也是计算机系统的一个重大转折。操作系统本身是一个庞大的程序，在它的控制下，计算机系统的每个部件（不仅硬件部件，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如 UNIX 分时操作系统，RSX-11M 实时操作系统等等。现正进行多方面的理论研究工作如操作系统结构设计、工具语言、进程通讯、关于死锁和各种调度算法、可靠性问题、性能测试和分析等。

1.1.4 计算机网络软件、数据库软件的出现

计算机的应用正向纵深发展：由辅助人们进行科学计算、实时控制、数据处理等发展到辅助设计、信息管理，甚至辅助人们进行思维和决策；由单机系统发展到简单的联机系统、复合计算机系统、分时系统直到计算机网络系统。由少数高等院校、科研单位走向整个社会，日益广泛地应用于国民经济的各个领域。

计算机网络软件是计算机技术和通信技术两者高度发展和密切结合的结果。从某种意义上讲，它是更高水平上的操作系统。它是利用通信线路把分布在不同地点上的多个独立的计算机系统连接成为一种网络，使网上用户能够实现数据传送，共享网络中的所有硬件、软件和数据等资源。这不仅提高了计算机的可靠性，均衡了网络中各种计算机的加载情况，并且便于系统的扩展。

目前，按照计算机网的跨越距离，又可分成远程计算机网和局部计算机网。远程网如美国 ARPA 网、DEC 网，局部网有 Ethernet、PC-net、PLAN 4000 等等。

当前计算机网络研究的课题有：高速数据通信和使用效率，不同类型的计算机结合，不同类型计算机网络彼此互连，网络操作语言，网络操作系统，以及完善各种服务级协议，建立网络服务系统，扩大网络服务范围等等。

随着计算机广泛应用于工业生产，商业管理、财政贸易，交通运输等部门，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。

数据库系统软件是数据管理经过人工管理、文件管理两个阶段发展起来的，大型企业的信息管理需要大量数据的较长时间的存储和检索、删除、更新加工等处理，用数据库管理系统软件来统一管理这些数据。

数据库管理软件具有数据结构化、最低冗余度、较高的程序与数据的独立性，易于扩充、

易于编制应用程序等优点，此外这种系统软件还具有安全性（防止不合法使用）、完整性（保证数据的正确性、有效性、相容性）及并发控制（避免并发程序间互相干扰）等功能，因此较大的信息系统都是建立在数据库设计之上的。

当前数据库系统软件的发展动向有：数据库标准化——为各种数据库提供统一的基础，建立共用数据库结构；数据库规范理论——研究构成规范的数据模型；数据库机器——改进数据库现行体系结构，部分地用硬件去完成数据库软件的工作；分布式数据库系统——是数据库技术与计算机网络相结合的产物；高级数据库和知识库——具有更强的处理能力，灵活性和方便接口的工程数据库、知识库等。有关数据库系统的发展可见第五章。

可以预见，随着计算机硬件发展和计算机广泛的应用，以及科学技术的不断进步，计算机软件还会有更大的飞跃。

§ 1.2 计算机软件

计算机资源包括硬件和软件，其硬件受到原设计的局限，增添和更新有一定的限度，但其软件的扩充是大有可为的。计算机硬件和软件组成的统一的整体，称为计算机系统。对于使用计算机的人来讲，面对的是以中央处理机和外围设备为物质基础的扩充了的计算机，即虚拟计算机。因此计算机系统使用得好不好，不只是指物质基础即硬件的使用是否正确，运行是否可靠，而且包括软件掌握得如何？发挥了多少效用。这是更为关键的一个方面。

软件至今没有一个确切的定义。软件的功能是利用计算机本身提供的逻辑功能来合理地组织计算机的工作，以便简化或代替人们使用计算机过程中的各个环节，并为用户提供一个便于掌握、操作简便的工作环境。

计算机软件有以下几部分：

- 汇编程序
- 各种高级语言
- 各种语言的编译或解释程序
- 各种标准程序库
- 操作系统
- 数据库或信息管理系统
- 计算机网络软件
- 各种应用程序

软件可分为系统软件与应用软件，但有时不是截然可分的。例如各种标准程序库，可看作是应用软件，也可以看作是计算机厂家提供的系统软件。因为用户稍加改造，甚至不必改造就能将它们编到自己的程序内。又例如大型企业的信息管理系统是把计算机技术和控制论、信息论、系统论等方法紧密结合起来而建立的系统工程应用系统。这种系统建立的过程是通过系统分析、设计、调试等阶段，应用了系统软件的多方面技术。这样建立的信息管理系统具有一定的通用性，便于推广。如美国 IBM 公司为机械制造业建立的管理信息系统 COPI-CS。对于一个使用计算机的人员来讲，熟悉系统软件的目的，就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和编制应用程序。

在此书中我们除了编入了操作系统、数据库系统、计算机网络外，还编入了数据结构和应用软件开发技术。

数据结构是软件的基础，在计算机科学中是一门综合性的专业基础课。汇编程序、编译程序、操作系统、数据库系统、人工智能等均要用到数据结构的概念和算法。数据结构是包括分析研究计算机加工的数据对象的特性，选择合适的数据结构和存储结构以及相应的算法，并且包括各种算法在时间和空间上的分析技巧。通过数据结构的學習，以进行复杂程序设计的训练。此外，为了进行大型应用软件的开发，本书还介绍了用系统的观点、软件工程的观点进行应用软件开发的技术。

为对软件有个全面的了解，在这一节中我们简述软件中的前四个部分，即汇编语言、各种高级语言、各种语言的编译或解释程序、各种标准数据库。

1.2.1 汇编程序

汇编语言的程序和对应的机器语言程序有相同之处，又有不同之处。相同之处是，程序主体部分几乎是一一对应的；不同部分是数码换成了符号，地址换成了名字，另外还增加了关于工作单元和常数单元的成分。这些不同之处正是汇编语言的优点，使得汇编语言的程序和机器语言程序比较起来，好写、好读、好改。

汇编语言也是依赖于机器的，因此称它为面向机器的语言。使用时必须了解机器的某些细节，如累加器、每条指令的执行速度、内存容量等等，但正由于它依赖于机器，所以就可以与机器语言程序一样可结合机器特点编出高质量的程序，即程序又短，执行速度又快。所以直至今日，汇编语言仍起着重要作用，在一些国外计算机公司中仍用汇编语言编写系统软件，来保证高质量的软件功效。

汇编程序的主要功能是把用汇编语言编写的源程序加工成机器语言写的目标程序。其过程可以分成三步：

- 用汇编语言编制出源程序；
- 将源程序输入到计算机内，由汇编程序把它加工成计算机能够执行的目标程序；
- 执行目标程序，得到计算结果。

源程序是由一串符号化的指令组成，国内采用标准的 ASCII 码，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的阅读，称为扫描源程序，从头到尾扫描一次源程序为扫描一遍，一般汇编程序是两遍扫描源程序的。第一遍把源程序中所有出现的名字进行造表，确定每个名字将占用的内存位置。第二遍扫描时，按所造出的表，把每条符号化指令代换成数码形式的机器指令。

此外汇编程序还具有一些附加的功能。

查错：当用户写的汇编语言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某个位置出了什么性质的错误。

修改：汇编程序提供修改源程序的简便办法，用户把修改的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。

打印：在汇编过程中，当发现错误时，将错误信息印出来，必要时还可印出名字表及目标程序，还可印出修改后源程序的文本等。

实用的汇编语言中还可以包括一些用户编程序时很需要的指令，这些指令并无对应的机器指令，称为伪指令。

有关汇编程序的内容在微机原理课程中已进行详细讲解，在此不再阐述。

1.2.2 各种高级语言

现世界上流行的高级语言很多，我们以常用的几种语言为例，介绍其特点。

FORTRAN 语言是用于科学和工程计算的语言，其程序结构是分块结构，每一块不包含进一层的结构。一个 FORTRAN 程序包含一个主程序块和若干个子程序块。程序的执行由主块开始，主块可以调用子块，子块间也可以调用，但子块不可以调用主块。另外在 FORTRAN 程序中变量的地址是静态分配的，因此不能包含递归子程序和动态数组。总的来讲，FORTRAN 程序的结构比较简单，而且可以分块书写和分块编译，故比较灵活方便。

ALGOL 60 语言是用于科学和工程计算的语言，其程序结构是分程序嵌套结构。整个 ALGOL 60 是一个大分程序，其嵌套的深度可以任意，内层分程序可以使用外层分程序定义的量，反之则不可以。其变量地址是动态分配的，是在进入分程序时分配，在退出分程序时释放，且允许递归子程序和动态数组。其程序结构比较严谨、明晰。动态分配地址比较节省存储。

COBOL 语言是通用商业语言，或称管理语言。对于数据处理，FORTRAN 和 ALGOL 60 语言都不是合适的计算机语言。为了解决大量数据的处理，发明了这种适用于商业数据处理的语言。在 COBOL 语言中引入了数据部分，按层次结构来描述，数据描述形式完全适合于现实生活中各种管理信息。另外，其程序设计采用了自然语言的程序设计风格。程序的书写是采用接近英语的形式。由于这些原因，COBOL 语言在商业、工业和行政管理部们被广泛地使用。

LISP 语言为表处理语言，是人工智能程序设计最常用的语言。在计算机应用方面，一个非常有意义的领域是非数值处理，例如公式推导、用机器证明定理、下棋、印刷排版、编辑修改、版面设计、机器翻译、语言研究等人工智能问题。这些非数值处理问题是 FORTRAN、ALGOL 60、COBOL 等语言不能解决的，如 $(X+Y)^2$ 的计算，在 FORTRAN 语言中只能先给出变量 X 和 Y 的数值，然后才能计算，绝不能推导出 $(X+Y)^2$ 等于 $X^2+2XY+Y^2$ 。采用 LISP 语言则可解决这种非数值处理问题。LISP 语言有两个基本的数据类型，叫做原子和表。一般把原子定义为以大写字母打头的大写字母和数字串，其长度不超过三十个字符。表是递归定义的，也就是在表的定义中又用到了表。两个表可以构造一个复合表。还定义了一组简单而有力的操作。例如由两个表构造一个复合表的运算，从一个表中选出其第一个分量的运算和从一个表去掉其第一个分量而取余下部分的运算。通过原子、表、表的运算以及定义一些 LISP 函数，就可以进行符号演算、公式推导、博弈及其它种种非数值处理的工作。

LISP 语言的理论基础是数理逻辑，它很简单但又有很强的表达能力。在运行过程中能动态地分配大量的存储，具有随时把无用的存储单元收集回来再用的技术，这也是非数值处理语言的一个共同特点。

PL/1 语言是一种大型通用语言，它包含了各种功能的语言成分，既能进行数值计算又能进行数据处理，既可以作表处理，又可以作符号串处理。此外，它还具有格式丰富的输入/输出设施和中断并行等能力。用户可以根据需要选用它的一部分或全部。

PL/1 是综合了 FORTRAN 语言、ALGOL 60 语言、COBOL 语言等的特点设计出来的一种大型语言。它的表达式和语法来自 FORTRAN 语言，分程序结构和类型说明来自 ALGOL 60 语言，数据描述取自 COBOL 语言，控制结构和总体结构则兼有 FORTRAN 和 ALGOL

60语言之所长。此外它还具有表处理能力、字符串处理能力以及静态、动态、和受控的三种存储分配能力，并有程序中断和多重任务等设施。

PASCAL 语言是一种按结构化程序设计原则设计的描写算法的语言，是一种适用于各种领域的功能较强的通用语言。它概念简单明晰，容易学习使用，适合于结构程序设计。PASCAL 语言的出现和结构程序设计技术的发展推动了编译程序工程技术的发展。PASCAL 语言强调程序的可靠性和易于验证，强调概念清晰、实现简化和方便用户。这使得程序易于阅读，有较好的结构，减少了错误的发生，并提高了效率和灵活性。

除以上通用语言外，还有许多专用于某种特定用途的语言。例如飞机工业中的机翼设计语言、造船设计中的船体放样的语言、专用于计算机线路设计的语言，专用于连续系统模拟的语言、专用于离散事件的模拟语言以及专用于编辑排版的语言等。

1.2.3 编译程序和解释程序

编译程序的工作过程是把高级语言写的源程序译成目标程序。采用的方法是先分析词法和语法，然后进行综合，包括代码优化、存储分配和代码生成三部分。

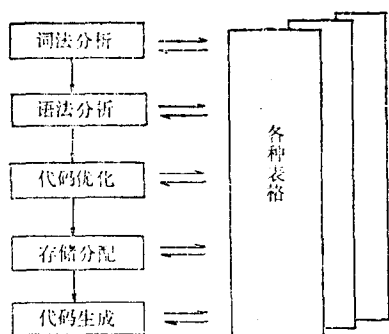


图 1.3

和语法，然后进行综合，包括代码优化、存储分配和代码生成三部分。词法分析程序是编译程序中的最基础的程序。要完成这么复杂的任务，编译程序要对源程序进行多次扫描。每扫描一遍完成一项或几项任务。不同语言的编译程序采用的具体做法不同。词法分析、语法分析、代码优化、存储分配和代码生成五部分都生成各种表格。每扫描一遍造出或修改各种表格，交下一遍使用，完成各遍之间的交换工作，每遍扫描就把源程序或内部表示形式从头到尾查看一遍，同时完成编译过程中的一部分工作。如图 1.3 所示。

词法分析过程中造出各种表格，登记了词法分析扫描中所获得的有关信息。在以后加工源程序的每个阶段中，还要不断地在这些表格中增加一些信息，或对原有的信息作些修改、增补。在生成机器指令的阶段，就要在表格中增设各变量（或标号等）所分配的内存地址，最后编译程序就根据这些表格生成机器指令，即目标指令。在这些表格中最为重要的，有代表性的是名字特性表和数表。每当读到一个新的名字时，就在名字特性表中增添一项，登记名字和有关特性的信息。名字特性表主要是生成目标程序用的，词法分析程序还把源程序中的数加工成机器内部表示形式的数并送到数表里。词法分析程序加工后，需要进行换码。所谓换码是将源程序中的字符换成代码。

语法分析程序的重要功能之一是找出源程序的语法错误，有时还能发现一些语义错误。语法分析的另一重要功能是通过语法分析弄清了句子的结构，以便将其翻译成目标程序。实际上，语法分析程序一方面将各种表格进一步改造或写一些从语法分析得到的信息；另一方面，经过语法分析后就可以编出接近于目标程序的过渡形式。这种过渡形式近似于汇编语言那样的格式，所有程序中的名字还只能是代码。

代码优化是编译技术中重要的问题之一。它为生成目标程序更加简洁、更加合理，为生成高质量的目标程序打下基础。代码优化的内容很多，例如合并常数，即如果一个运算的操