

# FORTRAN 77

## 语言及程序设计

乔沛荣 邱希春 主编

计算机应用基础  
教学系列丛书

华东理工大学出版社

机算机应用数学丛书

# **FORTRAN77语言及程序设计**

乔沛荣 邱希春 主编

华东理工大学出版社

## 上海市高校计算机应用教学丛书编委会

主 任 俞丽和

常务副主任 汪军波

副 主 任 乔沛荣 瞿彭志 章 鲁

委 员 (以姓氏笔画为序)

王修才 付铁华 李月明 许德因 沈海华

阮家栋 陆慧茜 邱希春 陈 健 郑志毅

张令初 张家骥 张慕蓉 夏明东 黄润发

黄俊民 潘高春 谢建华

秘 书 束建红

## 序

随着计算机硬件和软件的迅速发展,计算机在各行业中已得到普遍的使用,应用水平也不断地提高,计算机的应用能力已成为衡量科技人员和企事业管理人员素质的重要标志之一。

1990年上海市高等教育局决定建立上海普通高校非计算机专业学生计算机应用知识和应用能力等级考试制度,并于1992年3月组织了首次考试,以后每年都要进行这样的考试。

为了进一步提高高校非计算机专业的计算机教学质量,上海市高等学校计算机教学协作组(非计算机专业)组织编写了这套“计算机应用教学”丛书。

本丛书的作者均是各高校长期从事计算机教学第一线的骨干教师,教学经验丰富,实践能力强。

本丛书的主要对象是高等学校非计算机专业的学生,也可作为科技人员,管理人员计算机应用知识和应用能力的培训教材和自学参考书。

热忱欢迎广大读者对本丛书提出宝贵意见。

上海市高校计算机教学协作组

1994年1月

# 前 言

FORTRAN 语言是目前世界上较流行的程序设计语言之一。FORTRAN 77 是它的更新版本。后者在字符处理、文件等方面扩充了多种功能,这使得它不但能方便地进行科学计算,而且可以更广泛地应用到非数值计算的各个领域中去。目前,在各种类型的计算机上都配备了 FORTRAN 77 编译程序。为适应计算机发展的需要,根据国家教委高校计算机基础课程教学指导委员会 1993 年 9 月通过的《高级语言程序设计基本要术》提出的原则,编写了《FORTRAN 77 语言及程序设计》这本基础教材。可作为计算机程序设计入门课程的一学期教材,也可作为学习过其他语言学生的第二门程序设计技术课程和 FORTRAN 语言教材。学习本教材的数学基础是初等代数。

本教材以 FORTRAN 77 标准文本为依据,以结构化程序设计为主导,算法设计为基础,贯彻自顶向下,逐步求精的方法,充分重视字符处理和文件处理能力的训练,用通俗易懂的语言,由浅入深地介绍 FORTRAN 77 的语法规则和主要内容,力求在结构化程序设计方法和语言知识的安排上达到互相渗透,有机结合,随着语言知识教学的进程,互相补充,使编制的程序具有清晰的良好风格,易于编写,易于阅读,易于修改和调试。

本教材中还介绍了最基本的数值运算和非数值运算的算法及程序设计的基本知识,为读者继续提高解决较为复杂的运算的能力打下一定基础。在各章节中列举了若干数值运算和非数值运算的综合应用实例,供读者综合应用所学知识,提高阅读程序和编写程序的能力。

语言是一个工具,应该在使用过程中熟悉它,掌握它,而且应当以应用为目的,做到学精一门高级程序设计语言,达到灵活地得心应手地运用这门语言编写程序,去解决实际问题。学习本教材,除了听课或自学外,一定要多做练习、多编程序、多读程序、多上机调试程序。

本教材由乔沛荣、邱希春担任主编。参加编写的人员有吴建中(第一、二、三、六章)、乔沛荣(第四、五章)、邱希春(第七、八、九章)、郑志毅(第十、十一章)。全书由乔沛荣、邱希春修改定稿。本书编写工作得到了史济民的热情帮助,他认真审阅了全书,并提出了不少修改意见。

本教材在内容、结构和取材等方面的不足和错误之处,诚希读者提出批评和指正

编 者

1994年2月

# 目 录

i	概述 .....	1
1.1	计算机发展概况 .....	1
1.2	计算机系统 .....	2
1.2.1	存储器 .....	3
1.2.2	运算器 .....	4
1.2.3	控制器 .....	4
1.2.4	输入和输出设备 .....	4
1.3	计算机语言 .....	5
1.3.1	机器语言 .....	5
1.3.2	汇编语言 .....	6
1.3.3	高级语言 .....	6
1.4	操作系统初步 .....	7
1.4.1	操作系统的分类 .....	7
1.4.2	操作系统的基本功能 .....	8
1.4.3	PC-DOS 操作系统简介 .....	9
1.5	程序设计初步 .....	12
1.5.1	算法 .....	12
1.5.2	结构化程序设计 .....	13
1.5.3	程序设计的基本过程 .....	19
	习题 1 .....	21
2	<b>FORTRAN 77 语言的初步知识</b> .....	23
2.1	FORTRAN 语言的发展 .....	23
2.2	FORTRAN 程序结构 .....	24
2.3	FORTRAN 源程序的书写格式 .....	25
2.4	运行 FORTRAN 程序的方法 .....	28
	习题 2 .....	31
3	<b>FORTRAN 77 的基本成分及简单程序设计</b> .....	32
3.1	字符集和标识符 .....	32
3.1.1	字符集 .....	32
3.1.2	标识符 .....	33
3.2	常量、变量及其类型说明 .....	33
3.2.1	常量 .....	37
3.2.2	变量 .....	42
3.3	运算符及表达式 .....	42
3.3.1	算术运算符及算术表达式 .....	43
3.3.2	关系运算符及算术关系表达式 .....	45

3.3.3	逻辑运算符及逻辑表达式	47
3.3.4	字符表达式及其计算规则	49
3.4	内部函数	53
3.4.1	内部函数名	53
3.4.2	内部函数的引用	56
3.4.3	内部函数一览表	57
3.5	赋值语句	62
3.5.1	算术赋值语句	62
3.5.2	逻辑赋值语句	64
3.5.3	字符赋值语句	65
3.6	PARAMETER 语句	66
3.7	PAUSE 语句, STOP 语句和 END 语句	68
3.7.1	PAUSE 语句	68
3.7.2	STOP 语句	68
3.7.3	END 语句	69
3.8	基本输入输出语句	69
3.8.1	简单输入语句——表控格式	70
3.8.2	简单输出语句——表控格式	71
3.9	顺序结构程序设计及示例	72
	习题 3	76
4	转移与分支语句及选择程序设计	80
4.1	块 IF 结构	80
4.1.1	块 IF 语句的一般形式	80
4.1.2	单边选择结构	80
4.1.3	双边选择结构	82
4.1.4	多边选择结构	83
4.1.5	使用规则和说明	84
4.1.6	IF 级	88
4.1.7	选择结构程序实例	89
4.2	无条件转移语句和逻辑 IF 语句	94
4.2.1	无条件 GO TO 语句	94
4.2.2	逻辑 IF 语句	95
4.2.3	继续语句 (CONTINUE 语句)	98
4.2.4	算术 IF 语句	99
4.3	转移语句	103
4.3.1	计算 GO TO 语句	103
4.3.2	赋值 GO TO 语句	104
4.4	综合程序实例	106
	习题 4	112
5	循环及循环程序设计	114
5.1	循环及其分类	114

5.2.1 当型循环 .....	117
<b>5.2 当型循环和直到型循环</b> .....	<b>117</b>
5.2.2 直到型循环 .....	118
5.2.3 程序实例 .....	118
<b>5.3 DO 语句及计数型循环</b> .....	<b>122</b>
5.3.1 DO 循环的构成及循环次数 .....	122
<b>5.4 多重循环</b> .....	<b>133</b>
5.4.1 循环的嵌套 .....	133
5.4.2 DO 循环与块 IF 结构的嵌套 .....	137
5.4.3 控制转移语句的限制 .....	139
<b>5.5 综合程序实例</b> .....	<b>143</b>
习题 5 .....	150
<b>6 数据的输入输出</b> .....	<b>153</b>
<b>6.1 输入输出的基本概念</b> .....	<b>153</b>
6.1.1 记录 .....	153
6.1.2 文件与设备 .....	154
<b>6.2 输入/输出语句的分类</b> .....	<b>155</b>
<b>6.3 指定格式输入/输出语句</b> .....	<b>156</b>
6.3.1 指定格式输入语句(READ) .....	156
6.3.2 指定格式输出语句(WRITE、PRINT) .....	158
<b>6.4 格式说明语句(FORMAT)</b> .....	<b>160</b>
<b>6.5 编辑符</b> .....	<b>162</b>
6.5.1 I 编辑符 .....	162
6.5.2 F 编辑符 .....	165
6.5.3 E 和 D 编辑符 .....	167
6.5.4 G 编辑符 .....	168
6.5.5 L 编辑符 .....	169
6.5.6 A 编辑符 .....	170
6.5.7 H 编辑符 .....	172
6.5.8 撇号编辑符 .....	172
6.5.9 X 编辑符 .....	173
6.5.10 斜线编辑符 .....	173
6.5.11 冒号编辑符 .....	175
6.5.12 P 编辑符 .....	175
6.5.13 S 类编辑符 .....	175
6.5.14 B 类编辑符 .....	176
6.5.15 T 类编辑符 .....	177
6.5.16 复型数据的编辑 .....	178
6.5.17 重复系数与编辑描述组 .....	178
6.5.18 纵向走纸控制符 .....	179
习题 6 .....	182



<b>7 数组</b> .....	185
7.1 为什么要用数组.....	185
7.2 一维数组.....	187
7.2.1 数组说明.....	188
7.2.2 数组元素及数组下标.....	189
7.2.3 数组的基本操作.....	191
7.3 多维数组.....	203
7.3.1 多维数组的说明.....	204
7.3.2 多维数组元素的引用及下标.....	205
7.3.3 多维数组在内存中的存放顺序.....	206
7.3.4 多维数组的输入.....	208
7.3.5 多维数组的输出.....	209
7.3.6 多维数组的应用.....	209
习题 7.....	216
<b>8 语句函数</b> .....	221
8.1 语句函数概念.....	221
8.2 语句函数的定义规则.....	223
8.3 语句函数的调用规则.....	224
习题 8.....	227
<b>9 主程序与子程序</b> .....	229
9.1 概述.....	229
9.2 主程序及 PROGRAM 语句.....	229
9.3 FUNCTION 子程序.....	230
9.3.1 FUNCTION 子程序概念.....	230
9.3.2 FUNCTION 子程序的定义.....	231
9.3.3 FUNCTION 子程序的调用.....	234
9.4 SUBROUTINE 子程序.....	236
9.4.1 SUBROUTINE 子程序的定义.....	236
9.4.2 SUBROUTINE 子程序的调用.....	237
9.5 子程序调用过程中数据传送——虚参与实参的结合.....	239
9.5.1 虚参是变量.....	239
9.5.2 虚参是数组名.....	241
9.5.3 虚参是虚拟过程名.....	246
9.5.4 虚参是星号.....	250
9.6 可调数组.....	251
9.7 公用语句与等价语句.....	259
9.7.1 公用语句.....	259
9.7.2 等价语句.....	261
9.8 DATA 语句与 SAVE 语句.....	264
9.8.1 DATA 语句.....	264
9.8.2 SAVE 语句.....	265

9.9	BLOCK OATA 子程序 .....	267
9.10	应用示例 .....	268
	习题 9 .....	284
<b>10</b>	<b>文件 .....</b>	<b>289</b>
10.1	文件概念及分类 .....	289
10.1.1	记录 .....	289
10.1.2	文件 .....	290
10.2	文件的基本操作语句 .....	295
10.2.1	OPEN 语句 .....	296
10.2.2	CLOSE 语句 .....	299
10.2.3	数据文件的读写 .....	300
10.3	顺序文件 .....	302
10.3.1	文件定位语句 .....	303
10.3.2	有格式顺序文件的存取方式 .....	303
10.4	直接文件 .....	309
10.4.1	有格式直接文件的存取方式 .....	309
10.4.2	无格式直接文件的存取方式 .....	312
10.5	文件应用举例 .....	313
	习题 10 .....	317
<b>11</b>	<b>程序设计方法简介 .....</b>	<b>318</b>
11.1	概述 .....	318
11.2	自顶向下逐步求精的设计方法 .....	318
11.3	模块化设计 .....	320
11.4	程序风格 .....	321
11.5	程序的调试和测试 .....	321
	附录 .....	324

# 1 概 述

本章向读者简要地介绍计算机软硬件基本知识和程序设计初步, 主要内容包括,

- 计算机系统和基本组成;
- 计算机语言和操作系统分类及基本功能;
- 程序设计初步方法。

## 1.1 计算机发展概况

人类在长期的生产实践中, 创造了各种各样的工具, 以满足自身发展的需要。比如, 被称为千里眼的雷达拓展了视觉系统; 电话使人类的听觉系统延伸; 电视、收音机的发展大大地缩短了世界的距离; 各种各样的机械加强了手与足的能力, 极大地减轻了人们的体力劳动强度, 促进了社会生产力的发展。随着科学技术的进步和生产的实际需要, 各种计算工具也应运而生, 我国唐宋时期开始使用算盘; 17 世纪欧美出现计算尺和手摇计算机; 随后发明了电动计算机。但真正具有划时代意义的是在第二次世界大战的炮火中诞生的电子计算机, 这是人类计算工具发展的里程碑。

电子计算机是 20 世纪 40 年代、现代科学技术发展的产物。第一台电子计算机是在美国陆军部支持下, 由宾夕法尼亚大学摩尔电工系工程师 J.W. Mauchly 研制成功的。这台电子计算机定名为 ENIAC。它每秒钟可进行 5000 次加法或 300 次乘法。用了 18000 个电子管, 1500 个继电器, 耗电 1500 kW, 是一个重达 30 吨, 占地 1500(FOOT) 的庞然大物, 占据了宾夕法尼亚大学工程学院主楼的整个地下室。与现代的计算机相比, 虽然性能差异很大, 但毕竟跨出了具有划时代意义的一步, 为电子计算机的发展奠定了坚实的基础。

从 ENIAC 的问世到今天的 40 多年时间, 计算机科学得到飞速的发展, 并蕴藏着巨大的潜力。在推动计算机发展的诸多因素中, 电子逻辑器件的发展是最活跃的因素。从 50 年代以来, 电子计算机每隔 5 到 8 年就进行一次变革, 使速度提高 10 倍, 体积缩小 10 倍, 成本降低 10 倍, 40 多年来, 计算机的发展大致经历了四个阶段, 由于它的发展是以逻辑器件的更新换代为标志的, 因此, 人们也把计算机发展的不同阶段称为“四代”。

第一代电子计算机:

构成第一代电子计算机的基本逻辑器件是电子管电路, 以 ENIAC 为代表, 它的特点是体积大, 耗电量大, 计算速度慢, 存储容量小且工作不稳定。配制的外存储器主要是磁鼓和磁带存储器。主要用于科学计算。人们用机器语言编程序或配上汇编语言, 1956 年在 IBM 704 系统上安装了磁心存储器并出现 FORTRAN 程序设计语言。

第二代电子计算机:

第二代电子计算机使用的基本逻辑器件是晶体管, 以 1959 年 DEC 公司生产的 PDP-1

晶体管计算机为标志,其整机体积大为缩小,功耗低,运算速度每秒可达几万次,可靠性有较大的提高,配备了磁盘存储器,开始使用操作系统管理设备,计算机不仅用于科学计算,同时用于过程控制和数据处理。于是便出现了 COBOL 语言。

#### 第三代电子计算机:

60 年代中期,随着半导体工艺的进一步发展和微电子技术的形成,出现了集成电路(IC),促使计算机技术的又一次飞跃。第三代电子计算机是以中小规模集成电路作为主要逻辑部件,主存储器仍以磁芯存储器为主,整机体积进一步缩小,功能大幅度提高,使用灵活,价格便宜,外部设备配制更加完备,运算速度每秒达到几十万到几百万次。语言更为丰富,在操作系统方面提出了多道程序和分时系统的概念。1964 年 IBM 公司研制的 IBM 860 系列机,以其系列化、通用化、标准化而成为第三代计算机的代表受到了用户的欢迎,使用的领域已相当广泛。

#### 第四代电子计算机:

自集成电路出现以后,它的集成度以每 3 到 4 年提高一个数量级的速度在增长,而价格大幅度降低。第四代电子计算机开始于 70 年代初期,以美国英特尔(INTEL)公司生产的第一台微处理机为代表,采用中大规模集成电路作为主要的逻辑部件,采用半导体存储器,运算速度每秒可达几百万到几千万次。与第一台 ENIAC 计算机的功能相比,体积缩小 12000000 倍,价格不及 1/16000,而工作可靠性提高了几千倍。微型机发展很快,应用领域非常广泛,随着计算机技术的发展,功能不断提高,它的外围设备和软件日益完善,尤其是各种应用软件,为系统的开发和计算机的使用提供了方便。加上各种计算机网络的建成和使用,实现了不同地域之间的硬件和软件资源共享。由于微型机体积小,价格便宜,使用灵活,使计算机的普及得到了迅速的发展。

进入 80 年代以后,大规模集成电路朝着超大规模集成电路方向发展。每个芯片上集成 45 万个晶体管的 32 位处理机已经进入市场。大存储容量、强功能、运算速度每秒达数亿次的巨型计算机已相继在美、日、英、法等国研制成功。1983 年我国研制成功“银河”机,每秒运算速度达到 1 亿次。1993 年我国又研制成功运算速度 10 亿次的“银河 2”型计算机。巨型机的发展,集中地体现了当今的计算机科学的发展水平。

目前,计算机技术正在向巨型、微型、网络和人工智能等几个方向继续发展。日本等国正在着手研制第五代“智能”计算机,这是注重于逻辑推理,模拟人的智能的计算机系统。各国还重视光学技术、超导技术、电子仿生技术在计算机科学中的应用。如光盘存储器已进入实用阶段,超导元件构成的计算机部件已试验成功。随着计算机科学的发展,计算机技术必将会产生新的飞跃。

## 1.2 计算机系统

从第一台电子计算机问世到微型机和巨型机的产生,它的运算速度、存储容量、系统功能等发生了极大的变化。然而就计算机的基本结构而言,仍然属于“存储程序”计算机的基本原理。存储原理的概念最初是由匈牙利籍冯·诺依曼(Von Neumann)于 1946 年在“关于 EDVAC”的报告中提出来的。它的基本思想是计算机由存储器、运算器、控制器、输入设备和输出设备等五大部分组成。程序是一组计算机系统能直接识别的数字化的指令序列,

按地址顺序存放在存储器中。程序执行后，计算机能自动地按程序规定的顺序从存储器中取出当前需要执行的指令进行操作，直到程序结束。指令由二进制代码组成，分成操作码和地址码两部分，以决定计算机的操作性质和指明参加操作数所在的位置。这些特点是迄今为止各类计算机的共同特性，人们把按“存储程序”原理制造的计算机称为冯·诺依曼计算机。与初期的计算机相比，当今的计算机在构造和外围设备的配置上已经历了重大变化。然而按照其功能，一台计算机基本上还是由五大部分组成。一般我们把控制器和运算器称为中央处理机，简称CPU(Central Processor Unit)。它是计算机的核心部分，实现各种控制和运算功能。又把CPU和内存存储器合在一起称为主机。计算机各种设备之间相互协调，完成各种运算工作。冯·诺依曼计算机的基本组成及信息通路如图1.1所示。

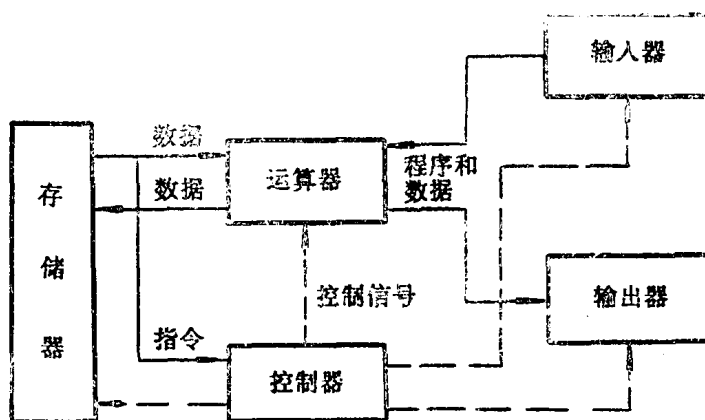


图 1.1 冯·诺依曼计算机的基本组成及信息通路

### 1.2.1 存储器

存储器是计算机存储信息的记忆装置，分为主存储器和辅存储器。主存储器简称主存或内存，辅存储器简称辅存或外存。

内存存储器是用磁芯体或逻辑部件和控制电路组成，用于贮存现行的程序和数据。在程序运行之前，必须将程序和初始数据通过输入设备送入内存；在运算过程中，保存各种运算的中间结果，也可取出内存保留的运算结果和各种信息，通过输出设备进行输出。总之，计算机中各种数据的传输都必须通过内存存储器来进行。因此，对内存存储器来说，数据存取速度将直接影响着计算机的运算速度。目前高速运行的计算机一般都选用半导体存储器。

程序和数据按顺序存放在存储单元中。一个内存存储器由许多存储单元所组成，每个单元又由若干个二进制位(Bit)组成，八个二进制位构成一个字节(Byte)。为区分不同的存储单元，按一定编址方式对它们进行编号，也就是说，每个单元有一个相应地址编码。通过查找单元地址就能实现对存储单元数据的存取。

辅存储器与内存存储器相比，辅存储器的存取速度比较慢，但是内存的存储容量是有限的，而辅存储器的存储容量一般不受限制。辅存储器通常是用磁性材料做成，如磁带、磁盘、磁鼓等。它的作用是存放暂时不用的数据和程序，当需要时，可通过输入/输出设备将数据或程序读入内存参加运算，也可把内存的有关信息写到辅存储器。辅存储器又称外存储器，简称辅存或外存。

存储器的一个重要特点是对存放在存储单元中的数据进行取操作时，存储单元中的数

据保持不变。倘若存入一个数据,则原数据直被新数据取代。人们习惯把它称为“取之不尽,一冲则了”。因此,在使用计算机时必须注意这个特点。

### 1.2.2 运算器

运算器是计算机实现加、减、乘、除等算术运算和其他一些常规运算(如关系运算、逻辑运算等)的基本部件。运算器主要由保存数据的寄存器和能够进行运算的加法器所组成;寄存器用来存放参与运算的数据或保存运算求得的结果。计算机的减法运算是通过对数据码的转换由加法实现的。在一些中、大型计算机系统中专门使用乘法器和除法器进行乘和除的运算,以提高整机的运算速度;然而在小型和微型计算机系统中是用软件的方法,用加法器一步一步完成的。应该说,加法器是计算机进行运算的主要构件。它的具体构造由系统设计时选定的运算方法及运算流程所提出的要求来确定,因此,不同的系统运算器的结构有所不同。

运算器在控制器的统一指挥下,在运算过程中,不断得到由存贮器提供的数据,并能把求得的结果送回到存贮器中。按照程序的要求顺序,有规律地完成各种运算任务。

### 1.2.3 控制器

控制器是计算机指挥和控制系统各部件活动的中央机构。在它的统一控制下,计算机各个设备协调动作,自动、有序地执行程序所规定的任务。

控制器是由指令寄存器、指令计数器、操作码译码器等基本部件所组成。在逻辑控制电路的驱动下,实现对整机的控制功能。我们已经知道“程序存贮”计算机系统的基本原理,是在程序执行之前,先将程序存入存贮器,程序指令由操作码和操作数两部分组成,操作数指出参加运算数据的地址。

在执行中,控制器从存贮器取出当前需要执行的一条指令,保存在指令寄存器中,由译码器对指令的性质进行分析,并根据操作码的指定要求,向主存、运算器及其他有关部件发出操作命令。当需要向运算器提供数据时,控制器要向主存发出“取数命令”,根据指令操作数指定的信息,决定所在主存的单元地址,将数据直接送给运算器,同时,控制器命令运算器执行规定的运算。

指令计数器的作用是存放当前执行指令的地址。当指令执行完毕,控制器就将计数器的内容加1,以便控制器接着执行下一条指令。因此,控制器对一条指令的执行过程,大致可分为取指、分析和执行三个阶段。

### 1.2.4 输入和输出设备

输入和输出设备属于计算机的外围设备,与计算机主机(指的是内存贮器、运算器、控制器)相联结构成一个计算机系统。这些设备在控制器的控制下进行工作,完成数据的输入输出,是人和计算机相互进行联系的部件。与高速运行的计算机相比,它们的速度都比较慢,输入和输出是两种不同的操作,然而有一些设备同时具有这两种功能,即既能进行输入又能进行输出(如磁带机、磁盘机)。由于计算机使用的对象和使用领域的不同,在输入输出设备的配置方面,存在一定的差异。

#### 1.2.4.1 输入设备

输入设备的作用是把数据和程序信息转换成计算机中的电信号，并顺序地存贮在计算机内存贮器。目前常用的输入设备是键盘打字机、软磁盘输入机、卡片读入机、电传打字机等。

用键盘打字机和电传打字机可以直接将数据和人工编写的原始程序通过击打符号键送入计算机内存贮器。

在使用卡片读入机时，必须首先把程序和数据用卡片穿孔机将卡片穿成穿孔卡片，然后将卡片上的程序信息读入到内存，这种卡片每张有12行80列，在某一系列的不同位置上穿上一个或一个以上(最多为三个)孔，表示一个字符。

对于事先通过计算机存贮在磁盘上的程序或其他信息，在需要时，可将相应的磁盘盘片插入磁盘输入机内，然后通过键盘向计算机打入有关的命令，将信息输入内存。

#### 1.2.4.2 输出设备

输出设备的作用与输入设备相反，是把内存的电信号转换成人们习惯的和直观形式的文字或符号，并且记录在某种外部介质上(如显示器、打印纸、磁盘、磁带等)。常用的输出设备有显示器、行式打印机、磁盘输出机、绘图机等。

在操作计算机时，输入计算机的程序和信息可直接通过显示器的屏幕显示出来，同时还能将一些随机的机器信息根据要求反映出来，以便操作人员及时了解程序的运行情况，修改程序和作必要的干预。通过键入适当的命令，按排计算机完成指定的任务，因此显示器和键盘打字机相互配合，成为人-机交换信息的主要桥梁。

行式打印机是将计算机的运算结果和程序以及其他需要的信息打印在纸上，供人阅读、交流和保存。打印机按行顺序打印，每一印刷行可打印80到160个字符。

对一些需要进一步修改和完善的程序，以及那些经常使用的程序和数据，可通过磁盘输出机存贮在磁盘上。磁盘的特点是存贮容量大，使用方便，便于携带。

输入输出设备是计算机不可缺少的重要设备，由于受到机械传动结构和信息传输等方面的影响，速度再快，也无法与主机的速度相比。主机就会因等待输入输出而处于闲置状态，浪费CPU执行时间。为此，需要设置专门的通道控制器，使得输入输出设备工作时，CPU同时处于工作状态，以协调主机与输入输出之间的速度矛盾。通道控制器在控制器的控制下，对输入输出设备进行管理。

## 1.3 计算机语言

计算机语言是人与计算机交换信息的中间媒介和工具，人们用它编制的程序描述指定问题的处理过程，然后按照程序的要求进行运算，计算出最终结果，实现运算的目的。随着计算机科学的发展，计算机语言经历了机器语言——汇编语言——高级语言的发展过程。

### 1.3.1 机器语言

计算机的每一步运算动作都是根据机器指令所指定的要求完成的，机器指令也称机器命令，是控制计算机各部件协调动作的命令。机器指令由若干位二进制码组成，一条指令分成操作码和操作数两部分。机器指令的具体表示形式和功能与计算机系统结构相关联，在计算机硬件总体设计时确定，与系统功能直接有关。因此，每个计算机系统都有一套指令系

统, 机器指令的集合称为机器语言。

用机器语言编写的程序, 计算机可以直接执行。但是, 由于这种程序是直接由二进制编码编写的, 因此, 程序的编写显得十分复杂、单调和枯燥, 且工作量大; 程序的直观性差、难以理解, 调试和修改有相当的难度; 它是一种面向机器的语言, 通用性差, 不易推广和交流。这些都给非计算机专业人员学习和使用计算机造成了很大的困难。

### 1.3.2 汇编语言

汇编语言是汇编指令的集合, 其指令的基本表示形式是以英文单词的缩写符号代表操作码, 例如用 ADD 代表加, SUB 代表减, MOVE 表示数据传送等。它的操作数一般以十六进制数表示。与二进制的机器指令相比, 汇编指令易于记忆, 因此通常也称它为助记码或记忆码。汇编指令与系统的机器指令基本一一对应, 所以汇编语言也是一种面向机器的语言, 与机器语言同属计算机低级程序设计语言。

用汇编语言编写的程序, 计算机不能直接执行, 要通过机器的“汇编系统”程序, 把汇编源程序的汇编指令翻译成机器指令, 然后计算机执行等价的机器语言程序, 我们把这个过程称为“汇编”。

汇编语言依赖于具体的机器, 因此同样存在通用性差、程序编写工作量大等弱点, 但是由于它能充分利用计算机的各种具体功能, 可以精心编制出运行速度快, 空间利用率高的程序, 因此, 目前有些程序, 尤其是系统程序的设计仍然采用汇编语言。

### 1.3.3 高级语言

机器指令的命令动作是非常简单, 而且是机械的。一个简单的算术运算需要由几条指令才能完成, 同时, 各计算机系统的指令系统又各不相同, 因此, 使用机器语言编写程序的工作显得十分复杂和困难, 尤其是对非计算机专业人员更为如此。也阻碍了计算机的推广应用和计算机技术的交流。汇编语言的出现也未从根本上解决这类问题。

计算机高级语言, 是一种比较接近人们习惯使用的自然语言和数学语言的程序设计语言。它由若干个程序语句所组成, 每个语句可表示多条机器指令的操作动作。它不依赖具体的指令系统, 是一种面向用户, 面向解题过程的计算机语言, 用高级语言编程序避免了与指令系统打交道, 大大地缩短了程序的编制周期, 减轻了劳动强度, 增强了易读性, 便于程序移植和人们的学习交流, 为计算机的推广和普及创造了良好的环境。

自 50 年代第一个计算机高级程序设计语言——FORTRAN 问世以来, 世界上至今已有一百甚至上千个不同种类和用途各异的高级语言。其中被广泛使用和具有代表性的有几十个。如: FORTRAN、ALGOL、COBOL、PASCAL、BASIC、PL/1、C、LISP、PROLOG、ADA、FORTH 等。各台计算机所配置的语言是有选择的, 计算机上能供使用的各种语言, 必须配置相应语言的“编译程序”或“解释程序”。

计算机高级语言对语句的书写和描述都有严格的语法和语义规定, 用户必须按照语句的语义描述算法。遵循语法规则书写程序。我们已经知道, 计算机最终执行的是机器指令, 因此, 用高级语言编写的源程序必定要由系统的“编译程序”将它翻译成由机器指令组成的目标程序, 通过执行完成运算任务; 或由“解释程序”将源程序的语句逐个转换成机器指令, 边转换, 边执行, 最终实现对整个程序的计算。



## 1.4 操作系统初步

计算机系统由硬件和软件两部分组成。硬件是由电气、电子及机械装置组成的设备,可分成存储器、运算器、控制器、输入设备和输出设备五大部分。软件指的是管理程序、操作系统、编译系统、数据库管理系统、应用程序等程序系统和相关的文档,分为系统软件和应用软件两大类,实现对计算机的管理,为用户操作计算机提供方便,以提高系统的使用效率。没有软件的计算机称为裸机,它的功能是十分有限的,而且操作起来也非常困难。配置了软件的计算机系统称为虚拟机,随着软件层次的增加,系统的功能也随之逐步增强。

操作系统是裸机的第一层软件,如图1.2所示,其他各种软件都是建立在操作系统之上,它的主要功能是实现对计算机系统资源(硬件和软件)进行有效的控制和管理;合理组织计算机的工作流程;改善系统的性能,提高其功能。它是用户使用计算机的接口,为用户提供方便。

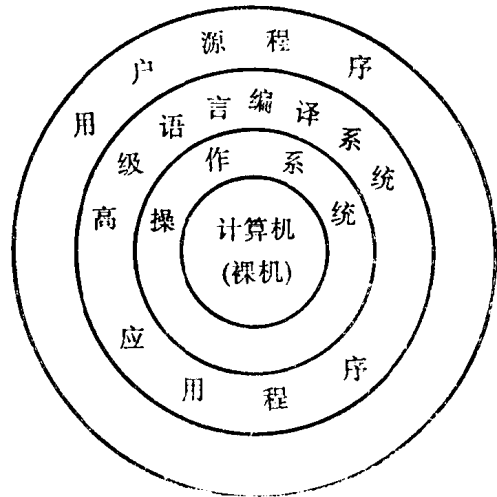


图 1.2 计算机系统资源分配层次图

### 1.4.1 操作系统的分类

早期的计算机是没有操作系统的。用户采用人工方式使用计算机,先把程序和数据纸带或卡片安装在输入设备上,然后启动设备将程序输入到内存存储器,程序开始执行后,直到程序运行完毕,计算机才能让给下一个用户使用。这种用户独占计算机的形式,系统资源的利用率很低,大量的 CPU 时间用于等待输入输出;只要与计算机联上多台终端设备,就可供多个用户同时使用计算机,有利于计算机的普及和推广;通过计算机文件系统,各分时用户之间可交流程序和各种信息。

实时操作系统可分为实时控制系统和信息服务系统。所谓实时,就是要求系统对用户提出的操作请求要立即给予处理和答复。实时系统的特征是:系统对用户请求的响应时间具有及时性,尤其是实时控制系统要以控制对象能接受的延迟来确定;实时系统仅允许用户访问有限数量的专用服务程序,不允许运行终端用户程序和修改已存程序;允许许多终端用户同时向系统提出服务请求,系统按分时原则为每个用户提供服务,对于实时控制系统,一般需具有多路采集现场信息和控制多个执行机构的功能;每个用户可通过各自的询问终端,向系统提出服务请求,互不干扰。

40 多年来,计算机操作系统的形成与发展,经历了一系列深刻的变革,从早期的手工操作,到多道程序操作系统的发展轨迹,可以看出,它的发展是以提高计算机操作的自动化程度,方便用户;利用计算机高速运行的特性,增强其处理问题的能力,以及加强系统的可靠性程度为主要目标。而它的发展是建立在硬件发展的基础之上的。操作系统的每一次变革都