

赵雄芳 张克强 陈怀义 李兆盛 詹皇业 编著

程序设计教程

FORTRAN 77

中国人民解放军国防科学技术大学

前 言

FORTRAN 是一种效率高、用途广的程序设计语言，它是从事科学、工程和数据处理工作者的一种重要的计算机工具语言，在国内已广为流行，并且仍处于不断的完善和发展过程中。近年来有关 FORTRAN 程序设计的书虽不少，但多是以基本 FORTRAN（相当于 FORTRAN I）或 FORTRAN66（相当于 FORTRAN IV）标准文本为背景的，而目前国内许多计算机上均已配置了 FORTRAN 语言国际标准的最新版本 FORTRAN77，它与 FORTRAN66 等相比，语言规模和语言功能都有较大的扩充。为了适应这一新的形势，满足读者进一步学习 FORTRAN 的要求，也为了在我国推广和普及计算机应用，加速四化建设方面尽绵薄之力，我们编著了《FORTRAN77 程序设计教程》一书。

本书依据 FORTRAN77 标准文本，全面、系统地叙述了 FORTRAN77 程序设计语言的基本规则，采用由浅入深、循序渐进的方式介绍了程序设计的基本概念、方法和技巧。为了开阔视野，增加了“向量程序设计浅谈”一章，介绍了向量语言的基本知识。为了方便读者学习，增设了《FORTRAN66 与 FORTRAN77》等附录。

本书文字通顺，条理清楚，例题丰富。因此，对于不具备计算机专业知识和较深数学基础的读者来说，也不难学到 FORTRAN77 的程序设计方法和技巧。通过教学使用证明，本书是一本较好的程序设计入门书，是一本较好的 FORTRAN77 标准的参考书，可作为大专院校计算机课程的教材，也可供计算机技术人员和自学者阅读。

参加本书编著工作的有赵雄芳、张克强、陈怀义、李兆盛和詹皇业等同志。在成书过程中，自始至终得到了研究所陈火旺教授、软件教研室齐治昌主任、卢贤民讲师等许多同志的关心和支持。刘毅同志在教学中对本书提出了有益的建议，特此表示衷心的感谢。

由于我们水平有限，书中错误和不当之处在所难免，恳请读者批评指正。

编 著 者

一九八四年四月

目 录

前 言

第一章 电子计算机简介

§1 计算机的发展概况	1
§2 计算机的组成及特点	2
§3 二进制、八进制与十六进制	4
§4 机器语言、汇编语言与高级语言	7
习 题	12

第二章 FORTRAN 语言基础

§1 FORTRAN 语言简史	13
§2 源程序结构	13
§3 源程序的书写格式	17
习 题	20

第三章 表达式与赋值语句

§1 数据类型和常数	23
§2 符号名与变量	26
§3 内部函数	30
§4 表达式	31
§5 赋值语句	35
习 题	38

第四章 输入输出初步

§1 基本概念	40
§2 格式输出	43
§3 格式输入	50
§4 输入/输出表和格式说明的相互作用	58
§5 表控输入输出	55
§6 举例	56
习 题	57

第五章 控制语句

§1 STOP 语句、PAUSE 语句和 END 语句	59
§2 无条件 GOTO 语句	61
§3 算术 IF 语句	61
§4 逻辑 IF 语句	66

§5	计算 GOTO 语句	75
§6	赋值 GOTO 语句	78
§7	块 IF 结构	80
	习 题	88
第六章	循环与数组	
§1	DO 循环	92
§2	关于循环的规定	102
§3	多重循环	108
§4	数组的引入和说明	112
§5	数组元素与下标值	116
§6	数组的输入/输出	119
§7	应用举例	122
	习 题	129
第七章	数据类型	
§1	双精度型数据	132
§2	复型数据	136
§3	字符型数据	140
§4	逻辑型数据	143
	习 题	152
第八章	过程	
§1	语句函数	155
§2	外部函数	160
§3	子程序	172
§4	可调数组	181
§5	ENTRY 语句	187
§6	RETURN 语句	190
§7	EXTERNAL 语句和 INTRINSIC 语句	192
§8	SAVE 语句	195
	习 题	197
第九章	数据联系语句和数据初值语句	
§1	EQUIVALENCE 语句 (等价语句)	200
§2	COMMON 语句 (公用语句)	204
§3	DATA 语句和数据块辅程序	211
	习 题	215
第十章	输入输出	
§1	记录 and 文件	217
§2	数据传输的输入/输出语句	221
§3	数据传输	231

§4 编辑.....	240
§5 辅助输入/输出语句	254
习 题.....	261
第十一章 常用算法.....	263
第十二章 向量程序设计浅谈	
§1 概述.....	292
§2 向量程序初析.....	295
§3 数组操作数.....	297
§4 数组赋值语句与数组表达式.....	306
§5 条件数组赋值语句.....	311
§6 一致性语句.....	318
§7 向量程序设计举例.....	335
附录1 程序单位内的语句顺序和注释行.....	345
附录2 FORTRAN77 语句形式表	345
附录3 可执行语句和不可执行语句表.....	347
附录4 内部函数表.....	349
附录5 FORTRAN66 与 FORTRAN77	354
参考文献.....	356

第一章 电子计算机简介

自古以来,人类为了提高自身的劳动效率,加快计算速度,增强计算能力,已经发明了各种各样的计算工具,例如算盘和计算尺等。这些工具千百年来一直沿用至今。但是,无论何种计算工具,也比不上电子计算机对人类所产生的巨大影响,而且,这种影响还在日益扩大,日益深入,用“无所不在、无所不能”来概括当今世界上电子计算机的作用,可以说是当之无愧的。当前,电子计算机的应用远远地超出了纯计算的范围,进入国民经济的各个领域,在科学研究、工业、农业、国防、商业以至于日常生活的各方面都得到越来越广泛的应用。计算机的科学技术水平、生产规模和应用程度,已经成为衡量一个国家现代化水平的重要标志。

本章简单介绍电子计算机的基础知识。

§1 计算机的发展概况

一九四六年,世界上诞生了第一台电子计算机“埃尼阿克”(ENIAC)。这台计算机每秒可作五千次运算,它用了18,800个电子管,占地1500平方英尺,耗电150千瓦,重量达30吨。虽然这台计算机又大又笨,但是,它的诞生,却标志着人类科学技术史上一次意义重大的突破。

三十多年来,计算机的发展非常迅速。计算机科学已经成为一门独立的学科,计算机的设计与制造已经成为一个完整的、独立的工业体系,其产值在国民经济总产值中所占的比例越来越大,新的、功能完善的机种层出不穷。在今天,完成ENIAC同样功能的微型计算机,体积仅ENIAC的百万分之一,重量仅几十克。

计算机的发展,已经历了四代的变化:第一代电子管计算机,运算速度每秒几千次到几万次;第二代晶体管计算机,运算速度每秒几万次到几十万次;第三代集成电路计算机,运算速度每秒几十万次到几百万次。第四代大规模集成电路计算机。大型计算机运算速度可达每秒几千万次,巨型机每秒上亿次,甚至几亿次。目前,已经着手研制新一代计算机——智能计算机,同时,也开始研制速度更快的计算机——光计算机。

据统计,大约每五到八年,计算机的运算速度提高十倍,体积缩小十分之九,成本降低十分之九。在数量上,世界上的机器总数每十年增加1~2个数量级,而微型机则是每二年就翻一番。由此,我们可以看到计算机发展的惊人速度。今后的计算机将不断采用新的设计思想,新材料、新技术和新工艺,使计算机速度更快,功能更完善,使用更方便。在研制方面,将向大型、巨型和小型、微型两个方向发展;在应用方面,将趋

向于系统化和网络化。

在党和政府的领导下，近年来我国的计算技术发展很快。一九八三年，每秒运算千万次以上的大型计算机和每秒向量运算亿次以上的银河亿次巨型机相继研制成功，使我国进入了世界巨型机研制的先进行列。另一方面，微型机的数量和质量也在稳步提高。特别是中文信息处理系统的研制已经取得可喜的成果，这就为我国计算机的推广和普及开辟了一个广阔的道路。随着社会主义建设事业的发展和进步，我国现代计算技术水平必将赶上和超过世界先进水平。

§2 计算机的组成及特点

2.1 计算机的组成

现代电子计算机由硬件（硬设备）和软件（软设备）两大部分组成。

一、计算机硬件

计算机硬件大致可分为五个部分：

1. 输入设备

它是计算机的“入口”。各种信息（程序和数据）首先由输入设备送入计算机。因此，它是人——机联系的主要设备之一。

常用的输入设备有：光电输入机、卡片输入机、电传打字机、控制打字机、终端键盘、磁带机和磁盘机等。

2. 存储器

用来存放各种信息。按其作用又分为内存存储器（简称内存）和外存存储器（简称外存）。由输入设备输入的信息直接送往内存，运算过程中所产生的中间结果和最终结果也放在内存中。存储器不仅可以存入信息，也可以取出信息，这种功能相当于人的大脑的“记忆”功能，因此，存储器也称记忆装置。

内存被分成许多小单元，每个单元（或相邻的几个单元）可用来存放一个数据或者一条机器指令。内存单元从0开始依次编有一个号码，该号码称为地址。存储单元的总数，称为存储器的容量。容量是以1K为单位的（这里1K=1024）。例如某台计算机的内存容量为32K，则表示内存有 $32 \times 1024 = 32768$ 个存储单元。内存容量越大，则存放的信息越多。

每个存储单元又分为若干位，每一位可存放一位二进制代码。若干位二进制代码的组合，可表示一个有独立意义的信息（一个数据或者一条指令）。一个存储单元的位数，称为字长，字长通常有8位，16位，32位，64位等，随不同机器而异。显然，字长越大，则所表示的数值范围也越大。

常用的内存有磁芯存储器和半导体存储器，尤其是微型计算机，都采用半导体存储器。由于体积的限制，一台计算机的内存容量与字长是有限的。

外存亦可用于存放信息，常用的外存有磁带机和磁盘机。由于磁带和磁盘的可拆卸性，因此，可以说外存的容量是无限的。和内存相比，外存容量大，存取速度慢，内存容量小，但存取速度快。

存储器有一个重要特点，即从它的任一单元取出信息之后，该单元原有的信息并不破坏，通俗地，可以说成“取之不尽”。另一方面，只有当存入新的信息，才取代原来的信息。通俗地，又可以说成“以新替旧”。

3. 控制器

控制器是整个机器的指挥中枢，由它指挥和协调计算机各部分自动工作。控制器是根据人们事先编制的程序来指挥的，人们把事先考虑好的意图（如计算机先做什么，后做什么，如何处理可能遇到的各种情况等）表达在程序中。因此，计算机自动工作的过程，实质上就是自动执行程序的过程。

4. 运算器

运算器完成各种算术运算和逻辑运算。例如加、减、乘、除以及逻辑加、逻辑乘等。此外，运算器还能完成某些其它动作，如数码传送、移位等。

5. 输出设备

它是计算机的“出口”。运算过程中的中间结果和最终结果既可以送往外存保存起来，也可以送往外界，输出的形式可以是印刷、绘图、显示或者穿成卡片或纸带等。

常用输出设备有：行式打印机、卡片或纸带穿孔输出机、电传打字机、自动绘图仪、终端屏幕、磁带机和磁盘机等。

计算机硬件各部分的联系如图1.1所示。

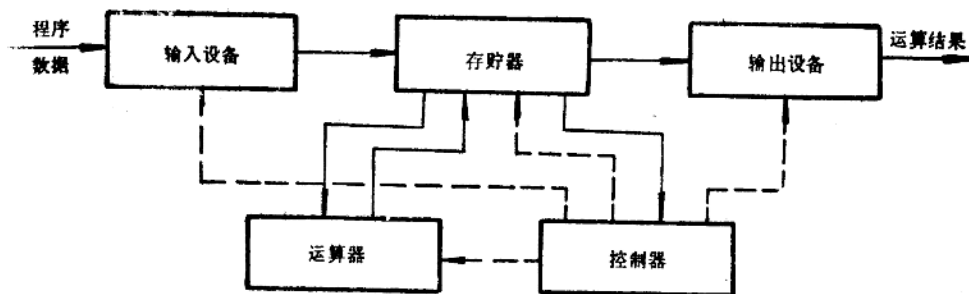


图 1.1 计算机各部分联系图

图 1.1 中实线表示信息的流向，虚线表示控制讯号的流向。

运算器、控制器和内存合称为计算机主机，其中运算器与控制器合称为中央处理机（简称 CPU）。输入和输出设备合称为外部设备（外围设备），外部设备的种类和数量是根据机器功能大小和使用要求而配置的。

二、计算机软件

为了充分发挥计算机的效能，保证机器正常工作，除了硬件之外，还需要配置功能完善的软件。

软件大体上可分为两大类：系统软件和应用软件。系统软件包括操作系统和各种编译程序、汇编程序、实用程序等，它是面向机器的，目的在于提高机器的使用效率；应用软件包括各种应用程序，标准程序等，它是面向用户的，目的在于提高用户的工作效率。计算机配备了功能完善的软件后，大大地提高了机器的使用效率，也极大地方便了用户，如今，软件已经成为计算机系统不可缺少的一部分。

2.2 计算机的特点

计算机和其它计算工具相比较,有如下几个显著的特点。

一、运算速度快

现代大型机的运算速度每秒上千万次,而巨型机的速度每秒上亿次,甚至几亿次。如此之高的运算速度是其它计算工具所望尘莫及的。直到今天,才使得某些以前依靠人力根本无法完成的计算有了完成的可能。有了计算机,人类从事于计算的速度提高了好几个数量级。

二、存储容量大

大型和巨型计算机能存储几十万以至于几千万个数据。因此,平时可以利用计算机保存大量的数据和资料,以备随时查阅。某些特大型运算,例如气象预报、地震探矿和航空航天飞行器研制等,既要求高速,又要求大容量。这些要求,只有计算机才能满足。

三、精确度高

台式计算器只有七、八位有效数字,而大型计算机的有效数字有十几位,如果要求更高的精度,还可以用双倍精度运算,有效数字可达四十位以上。

四、具有逻辑判断能力

计算机除可进行算术运算以外,还可以作逻辑运算,例如比较两个数值的大小,判断一个数值是否有正值或负值等。因此,计算机能完成各种复杂的计算任务。

五、自动化程度高

计算机的整个运算过程都是在程序控制下自动进行的,无需人工参与。能否自动运算,这是计算机与计算器的本质区别。

§3 二进制、八进制与十六进制

电子计算机接受的信息都是用二进制表示的代码。这里,我们简单介绍二进制以及与之有关的八进制与十六进制。

3.1 二进制

为了搞清楚什么是二进制,先看看我们日常习惯的十进制。

十进制在我们日常生活中是屡见不鲜的。例如一元十角,一丈十尺,一米十分米,一市斤十两等等,俯拾即是。之所以采用十进制,这大概与人类的十个手指有关。

十进制的基数是10,它有10个基本数字,即0,1,2,...,9,逢十进一。例如十进制数256.8可表示成

$$256.8_{(10)} = 2 \times 10^2 + 5 \times 10^1 + 6 \times 10^0 + 8 \times 10^{-1}$$

在日常生活中,还可以见到其它的计数制。例如1小时60分,这是六十进制,一打铅笔12支,这是十二进制。还可以举出其它例子。

相应地,二进制的基数是2,它只有两个基本数字,即0和1,并且逢二进一,比如二进制数1011.11可表示成

$$1011.11_{(2)} = 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 + 1 \times 2^{-1} + 1 \times 2^{-2}$$

如果我们把上式右边的多项式按十进制 计算出来, 则将该 二进制数化成了十进制数, 计算过程为:

$$\begin{aligned} 1011.11_{(2)} &= 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 + 1 \times 2^{-1} + 1 \times 2^{-2} \\ &= 8 + 2 + 1 + 0.5 + 0.25 \\ &= 11.75_{(10)} \end{aligned}$$

下边简单介绍怎样将十进制数化成二进制数。

对于十进制整数, 可用除 2 取余法: 将十进制数除以 2, 得到一个商和余数; 再将商除以 2, 又得到一个新的商和余数; 如此做下去, 直到商等于零为止。将各次所得余数, 以最后余数为二进制最高位数字, 最先余数为最低位数字, 依次排列, 即为所求的二进制数。

例如: 将 $29_{(10)}$ 化成二进制数的计算过程如下:

2	29	余 1	↑
2	14	余 0	
2	7	余 1	
2	3	余 1	
2	1	余 1	
	0	余 1	

所以, $29_{(10)} = 11101_{(2)}$

对于十进制纯小数, 用乘 2 取整法: 先用 2 乘十进制纯小数, 然后去掉乘积中的整数部分, 再用 2 去乘剩下的纯小数部分。如此做下去, 直到满足精度要求或纯小数部分等于零为止。把各次乘积的整数部分由上而下依次排列, 即为所求的二进制纯小数。

例如: 将 $0.375_{(10)}$ 化成二进制数的过程可简写为:

0.	3 7 5
	2
0	7 5 0
	2
1	5 0 0
	2
1	0 0 0

所以, $0.375_{(10)} = 0.011_{(2)}$

当十进制数既含整数又含小数时, 则可按上述方法分别转换, 然后拼成一个二进制数。

计算机为什么采用二进制而不采用十进制呢? 这是因为二进制较之十进制有许多优点:

1. 二进制数容易表示

自然界有许多截然对立的现象。例如高和低，开和关，亮和灭，有和无等等。这些都是对立的状态。如果用“高”状态表示1，则可用“低”状态表示0。因此，例如二进制数101就可以用“高低高”这个状态组合表示。

在电子技术中，晶体管的通导与截止，双稳态电路的高电位与低电位，门电路的正脉冲与负脉冲，都是具有两种稳定状态的元件。这两种状态就可以用来表示0和1，并且很容易控制。但是要找到具有十种稳定状态的元件来表示十进制数，则相当困难。

2. 二进制的运算简单

以乘法为例，二进制乘法的运算规则如下：

$$\begin{array}{ll} 0 \times 0 = 0 & 1 \times 0 = 0 \\ 0 \times 1 = 0 & 1 \times 1 = 1 \end{array}$$

我们只须掌握四条口诀，便可以做乘法运算。但是十进制的乘法口诀却有81条，即小学生的九九表。

3. 采用二进制，有利于节省设备，提高速度和可靠性。

用二进制表示一个数当然也有不方便之处，例如数511_(十)，只有3位，但是若用二进制表示，则表示为111111111，共9位，所需位数要多得多，看起来不直观，读起来不方便。在实际工作中，为了弥补二进制的不足，往往还采用八进制与十六进制。

3.2 八进制与十六进制

八进制的基数是8，它有8个基本数字，即0, 1, 2, ..., 7，逢八进一。例如：

$$56_{(八)} = 5 \times 8^1 + 6 \times 8^0 = 46_{(十)}$$

八进制与二进制之间有一个简单的转换关系，即每三位二进制相当于一位八进制。因此将二进制数化成八进制数时，以小数点为界，整数部分向左，小数部分向右，每三位二进制对应一位八进制，例如

$$10111011.11_{(二)} = 010 \ 111 \ 011. \ 110 = 273.6_{(八)}$$

相反，将八进制数化成二进制数时，只须将每一位八进制展开成三位二进制即可，例如

$$346_{(八)} = 011 \ 100 \ 110_{(二)} = 11100110_{(二)}$$

八进制较接近于十进制，无论从书写和阅读都比二进制直观、方便，因此，八进制常作为二进制的缩写使用。

不难理解，十六进制的基数是16，它有16个基本数字，即0, 1, 2, 3, ..., 9, 0, 1, 2, 3, 4, 5。逢十六进一。数字0, 1, 2, 3, 4, 5是表示十进制的10, 11, ..., 15这六个数。在有些场合，还经常将这六个数字写成A, B, C, D, E, F。

例如：

$$1B6_{(十六)} = 1 \times 16^2 + 11 \times 16^1 + 6 \times 16^0 = 438_{(十)}$$

十六进制也有类似于八进制的转换关系和作用。

表1.1是某些数的几种计数制对照表。

表 1.1

数 \ 计数制	十	二	八	十六
零	0	0	0	0
一	1	1	1	1
二	2	1 0	2	2
三	3	1 1	3	3
四	4	1 0 0	4	4
五	5	1 0 1	5	5
六	6	1 1 0	6	6
七	7	1 1 1	7	7
八	8	1 0 0 0	1 0	8
九	9	1 0 0 1	1 1	9
十	1 0	1 0 1 0	1 2	A
十一	1 1	1 0 1 1	1 3	B
十二	1 2	1 1 0 0	1 4	C
十三	1 3	1 1 0 1	1 5	D
十四	1 4	1 1 1 0	1 6	E
十五	1 5	1 1 1 1	1 7	F

§4 机器语言、汇编语言与高级语言

我们已经粗略地了解了计算机，那么，如何使用它来解决我们的实际问题呢？

一般地，使用计算机解题的全过程如图 1.2 所示。

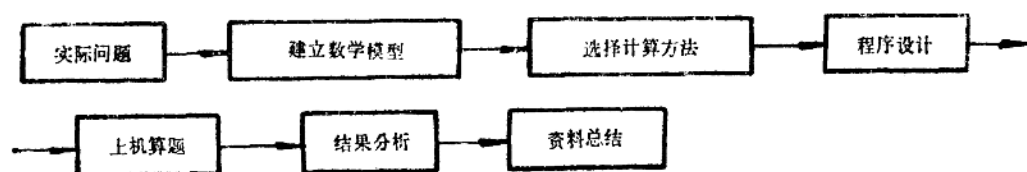


图 1.2 用计算机解题全过程

如何从实际问题出发，建立相应的数学模型，又如何选择合适的计算方法，这些问题的讨论已经超出了本书的范围。我们所要解决的问题是：在确定了计算方法之后，如何进行程序设计这一工作，并且局限于如何用 FORTRAN 语言进行程序设计。至于上机算题以后的几个步骤，那是不言而喻的。

计算机能够根据人们事先编制好的程序自动工作，所以，程序编制得好坏与否，将直接影响到计算机的工作效率。在逐步解决如何编制程序这一问题之前，首先要弄清楚什么是程序？

一个程序实质上是一组有序的指令。这组指令描述了一个计算过程，广义地，可以说描述了一个算法。当我们把一个计算过程适当地分细，直到计算的每一步都可以用一条指令描述出来，并且采用某种工具来书写这一条条指令，最终形成一个实现这个计算过程的程序，这一工作过程就叫做程序设计。通俗地说，当我们确定下来要求计算机“做什么”时，首先就应当告诉它“怎么做”，即告诉它第一步该做什么，第二步该做什么，等等。

人与人对话需要使用某种语言，人与机器“对话”，也需要某种语言。因此，上文所说的用某种工具就是指用某种语言来编制程序。目前在程序设计过程中常用到下列几类语言，对于不同的语言，指令的含义也不一样。

4.1 机器语言

任何一台计算机都有它固有的一套指令，称为机器指令。每条指令规定了计算机的一个动作，例如做一个加法，实现某种转移等等。不同机器其指令数目相差甚远，少则几十条，多则一、二百条。显然指令越多，机器的功能越强。这一整套指令称为计算机的指令系统，也称为机器语言。

用机器语言编写的程序称为手编程序。手编程序中的每一条指令都是用二进制表示的机器指令。一般地，机器指令的格式如下：

操 作 码	地 址 码
-------	-------

即一条机器指令大致可分为两部分。第一部分为操作码，指明计算机的动作，例如加法，转移等等；第二部分为地址码，指明操作对象所在的内存单元地址。另外，程序中所使用的数据事先也需要人工化成二进制形式。手编程序输入计算机之后，便可立即执行。这是因为这种程序能够直接被机器所识别和接受，无需对它进行加工。所以，用机器语言算题的过程如图 1.3 所示。

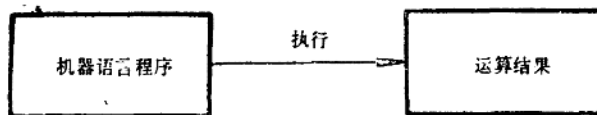


图 1.3 用机器语言算题过程

我们来看一个简单例子。

例 1 计算 $A \times B + C$ ，结果放在 Y 中。

用某台计算机的指令可编成如下一段程序。

指令地址	操作码	地址码	
200	0 0 3	0 0 0 1 0 0	(把 A 取到累加器)
201	0 0 6	0 0 0 1 0 2	($A \times B$)
202	0 0 2	0 0 0 1 1 2	(把 $A \times B$ 暂存 112 号单元)
203	0 0 3	0 0 0 1 0 4	(把 C 取到累加器)
204	0 0 4	0 0 0 1 1 2	($A \times B + C$)
205	0 0 2	0 0 0 1 1 0	(把结果存入 Y 中)

这里省略了必要的输出结果的指令，也没有给出 A 、 B 、 C 的具体数值。我们假定这台计算机的某些动作作用如下的代码表示：

存数	0 2
取数	0 3
加法	0 4
乘法	0 6

初始数据 A 、 B 、 C 分别存入100, 102, 104号单元, 110号单元存放运算结果 Y , 112号单元存放中间结果 $A \times B$, 程序存入 200~205 号单元。程序以及地址都是二进制的, 这里我们用八进制书写, 以便看起来直观一些。

用机器语言编写程序有两件事情是必须做到的。第一要了解和掌握指令系统, 第二要为数据(初始数据、中间结果和最终结果)和程序本身分配相应的存贮单元。这些事情都要求用户自己完成。

机器指令是一些二进制代码, 要做到运用自如, 已属不易。另外, 一条指令动作简单, 这就要求用户将计算过程分得很细, 以满足机器本身的要求, 这是一项十分繁琐、细致、费时的的工作, 必须周密考虑, 万无一失。稍有不慎, 则牵一发而动全身, 引起整个程序的大范围修改。还有, 程序难读难懂, 不利于交流, 无通用性。

总之, 用机器语言编写程序工作量大, 重复性劳动多, 直观性差, 不利于计算机的普及和推广。

4.2 汇编语言

为了克服使用机器语言的种种困难, 在计算机问世不久, 人们就开始寻找新的程序设计途径。

初期人们创造了“汇编语言”。汇编语言实质上就是机器语言符号化。用一组便于记忆的符号来编写程序, 而将分配存贮单元、翻译二进制数等某些工作交给计算机去完成。

仍以例1为例。

例 1' 计算 $A \times B + C$, 结果放在 Y 中。

用汇编语言可编写成如下一段程序。

L D	A	(取出 A)
$\times$	B	($A \times B$)
S T	W	(存入 W)
L D	C	(取出 C)
+	W	($W + C$)
S T	Y	(存入 Y)

我们已经知道, 计算机只能识别和接受机器语言程序, 并不“认识”用汇编语言编写的程序。为了解决这一矛盾, 需要有一个“翻译员”, 负责作等价的变换工作, 同时兼管分配内存单元等项事情。这个“翻译员”实际上也是一个程序, 它由软件设计者编制好事先输入计算机。因此, 用汇编语言解题要分为两步。第一步是翻译, 第二步才是执行。

用汇编语言编写的程序称为源程序，担任“翻译”的程序称为汇编程序，翻译后的程序称为目标程序，汇编程序的工作过程称为汇编。算题过程如图 1.4 所示。

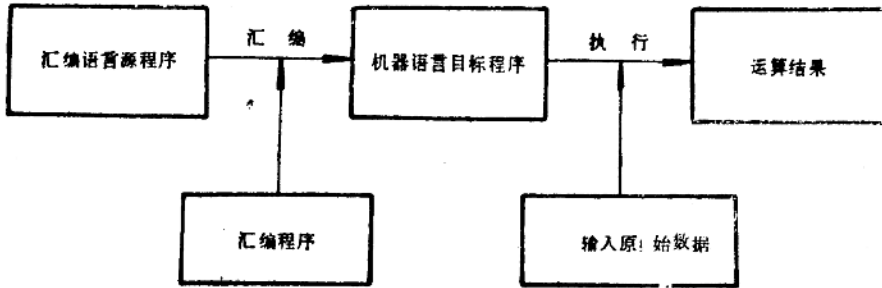


图 1.4 用汇编语言算题过程

汇编语言比机器语言前进了一步，比较好学，便于记忆。但是，它仍然依赖于具体机器，是面向机器的。所以，汇编语言在克服机器语言的种种弊端方面，并没有实质性的突破。

4.3 高级语言

随着高速计算机的发展，编制程序的慢速与实际计算的快速之间的矛盾越来越尖锐，甚至到了难以忍受的地步。这对于充分发挥机器的效率来说是十分不利的。为了缓和这对矛盾，除了研制复杂的服务性程序之外，各种高级语言也应运而生。

高级语言的一个更为流行的名称是程序设计语言。相对地，人们把机器语言或者汇编语言称为低级语言。

还是以例 1 为例。

例 1' 计算 $A \times B + C$ ，结果放在 Y 中。

如果用 FORTRAN 语言来编写这段程序，则只需一个语句：

$$Y = A * B + C$$

再加上必要的输入输出部分，完整的程序如下：

```

      READ (5, 15) A, B, C
15   FORMAT (3F8.2)
      Y = A * B + C
      WRITE (6, 20) Y
20   FORMAT (1X, F8.3)
      STOP
      END
  
```

高级语言的一条指令就是一个语句。前述例 1 和例 1' 都用了六条指令来完成这个计算，这里只需一个语句便达到同样目的，仅此一斑，便可略知高级语言的优越性。

用高级语言编写的程序亦称源程序。算题时同样需要一个“翻译”，以便把源程序翻译成等价的目标程序。这个翻译程序称为编译程序，编译程序的工作过程称为编译。算题过程如图 1.5 所示。

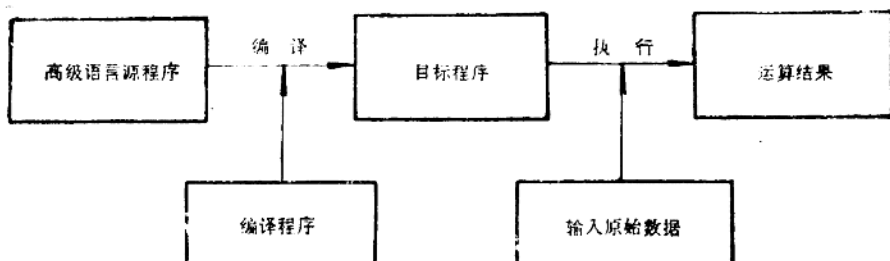


图 1.5 编译方式的解题过程

用高级语言解题同样需要两步，第一步是编译源程序，第二步是执行目标程序，这是指的一般情况。实际上，某些高级语言（例如 BASIC 语言）不是采用先编译后执行的方式，而是采用边解释边执行的方式。这种方式的解题过程是：源程序输入计算机之后，由事先存入的“解释程序”来解释它的每一个语句，每解释一个语句之后，便立即让它执行，当解释完最后一个语句，程序的执行也告结束，并取得相应结果。解释方式的解题过程如图1.6所示。

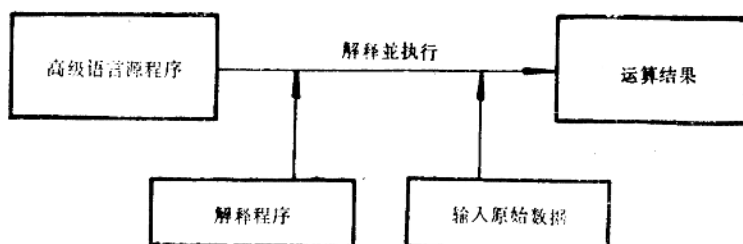


图 1.6 解释方式的解题过程

计算机配置了高级语言之后，人们不必学习机器指令，也无需了解计算机的内部结构与工作原理，便可以方便地使用计算机。另外，高级语言是独立于具体机器，面向问题的。不同型号的机器只要配置了同一种高级语言，同一个程序就可以在不同的机器上运算，并且结果完全相同。还有，高级语言与日常习惯用语（主要是英语）颇有相似之处。这样一来，程序的可读性、通用性等等问题便迎刃而解，程序的正确性也相应提高，编写的速度也要快得多。现在，相当多的算法已经有了用高级语言编写的标准程序，需要时，抄过来即可使用。从而大大缩短了编制程序的周期，减轻了工作量，也为计算机的普及和推广扫除了极大的障碍。

目前，国内外高级语言有数百种之多，其中绝大多数是专用语言，比较流行的有十几种，适用的范围各不相同，其中常用的有：

FORTRAN, ALGOL 60

(数值计算语言)

COBOL

(事务处理语言)

LISP

(表处理语言)

PL/I, ALGOL 68

(大型通用语言)

BASIC

(小型会话式语言)

高级语言的研制方兴未艾,在不久的将来,将会有功能更强的、使用更方便的语言问世。

最后,必须指出,使用计算机解题的过程是一个循环往复,渐趋正确的过程,尤其是一个大型复杂的程序更是如此。事实上,上机算题一次成功的可能性是很小的,编制好的程序,往往需要反复修改,调试,试算,最后才正式计算,以保证结果的正确性。当我们在使用电子计算机这一现代化的计算工具为四化出力时,需要的是严肃认真、耐心细致、一丝不苟的科学态度。

习 题

1. 一个计算机系统的硬设备包括哪几部分?请画出计算机的结构略图。
2. 化下列各二进制数为等价的十进制数:
 - (a) 110.010
 - (b) 1101
 - (c) 0.1011
3. 化下列十进制数为等价的二进制数:
 - (a) 1984
 - (b) 18.75
4. 指出并说明用计算机解算问题必须遵照的步骤。
5. 什么是汇编程序?它的功能是什么?
6. 用图表示出计算机处理 FORTRAN 语言程序的两个阶段?
7. 指出下面每组中两个概念之间的区别:
 - (a) 机器语言与汇编语言
 - (b) 汇编与编译
 - (c) 源程序与目标程序
 - (d) 与机器相关和与机器不相关