

294821

程序设计初阶

〔美〕 M. H. 魯貝尔 著
仲萃豪 謝奇光 楊篤引譯

761

2014年7月

程序设计与应用

作者：陈国平 编著
出版社：清华大学出版社

程序設計初階

〔美〕 M. H. 魯貝爾 著

仲萃豪 謝奇光 楊箎引 譯

安其春 校

上海科學技術出版社

內 容 提 要

本书說明电子数字計算机的使用方法,是为初学者写的。内容包括使用电子計算机所需程序設計的一般基础知识,和循环、框图、子程序、程序庫、檢查程序、自动程序設計以及机器語言及 SOAP 指令等,簡明扼要,尤多实际例題。

本书可作綜合大学計算数学专业及高等院校有关专业的参考书;亦可供有关科学研究工作者、工程师、技术員等作为学习和参考之用。

程 序 設 計 初 阶

A PRIMER OF PROGRAMMING FOR DIGITAL COMPUTERS

原 著 者 (美) Marshal H. Wrubel

原出版者 McGraw-Hill Book Co., Inc.

1959 年版

譯 者 仲萃豪 謝奇光 楊篋引

校 者 安 其 春

*

上海科学技术出版社出版

(上海瑞金二路450号)

上海市书刊出版业营业許可証出 093 号

新华书店上海发行所发行 各地新华书店經售

大东集成联合印刷厂印刷

*

开本 850×1168 1/32 印張 6 24/32 字数 174,000

1962 年 8 月第 1 版 1962 年 8 月第 1 次印刷

印数 1-3,500

統一书号: 13119 · 466

定 价: (十) 0.78 元

引 言

本书是为要想以电子计算机作为研究工具的科学家、工程师和他们的学生写的。本书目的是回答这样的问题：电子计算机能解哪一类问题和我们应该如何使用它？整本书中，我们将以解决这些问题为目的，而不叙述电子计算机本身的构造。

即使科学家并不想独自将算题编好程序上计算机计算，但为了对程序设计者说明他的问题，并能用适合于机器计算的术语来解释它，对计算机有适当的了解仍是重要的。

本书目的不在于培养熟练的程序设计者，而是为初学者写的，所以为了简单明了，有些技巧性的内容只得略去。虽然书中引用了很多简单的例题，是从印地安那(Indiana)大学计算研究中心遇到的问题中选得的，但还是建议读者试将发生在本行中的问题编为例题来说明书中所提到的各点。实际上，开始在计算机上解题时最好是用熟悉的问题，并且要抱着解决这个问题的态度来读这本书。应当着重指出：在计算机上计算自己编出的问题，即使是一个简单的问题，所获得的经验也比读多少本这方面的书有价值得多。

对于初学者，困难不仅在于计算机的种类繁多，而且同一架计算机也有各种不同的程序设计方法。遗憾的是，计算机还没有象汽车那样标准化，不过它们还是有若干相同的部件和操作方法。本书中所叙述的一般技巧事实上可适用于任何厂家制造的存储程序式计算机。

为了说明书中所述各点，我们选择了一架计算机 IBM 650。所以选它，仅是因为这类计算机最为普遍，任何人想要使用 650 机来解题，大都能就近找到这种计算机，可以用来进行检查并进行计算。其他计算机的特点在适当的时候也将提到一些。

本书分为两部分：初等程序设计和高等程序设计。许多问题

可以完全用第一部分中所讲的方法解出。如果可能，讀者在讀第二部分之前最好能到計算机上試算一个簡單的問題。

本书附有一些程序設計術語詞汇，可以帮助被計算机的專門名詞所困惑的初学者把經常遇到的術語的意义搞清楚。

感謝 IBM 公司同意在这整本书中引用穿孔卡片技术和 IBM 650 型計算机的各种程序設計方法；并感謝 V. M. Wolontis 博士及貝尔電話實驗室同意引用他們的解釋代碼。

Marshal H. Wrubel



IBM 650 型电子计算机

目 录

引 言

第一部分 初等程序設計

第一章 一般介紹.....	1
1-1 什么是电子数字计算机?	1
1-2 程序的概念	2
1-3 程序庫.....	3
1-4 数值分析与程序設計	4
1-5 计算机的部件	5
1-6 几个典型的上计算机的問題.....	6
1-7 一个問題对电子计算机來說什么时候是太大或太小?	8
第二章 基础知識介紹.....	10
2-1 数制	10
2-2 字长	13
2-3 作为数据的字: 定点和浮点表示法	14
2-4 分地址的存貯器	17
2-5 作为指令的字	18
2-6 机器語言	18
2-7 解釋程序	19
2-8 貝尔 (Bell) 电话实验室的代碼	20
2-9 存貯器的功用	21
2-10 算术运算	22
2-11 地址 000 的特殊作用	25
2-12 简单的算术問題	25
2-13 穿孔卡片	29
2-14 准备穿孔卡片	32
2-15 輸入	35

2-16	讀入 (READ) 指令	37
2-17	穿孔 (PUNCH) 指令	38
2-18	无条件轉移 (UNCONDITIONAL TRANSFER)	39
2-19	一个完整的例題	40
2-20	一个补充例題	41
2-21	常用函数的代碼	42
	习 題	44
第三章	循环和分枝	46
3-1	一般說明	46
3-2	簡單的計数循环	46
3-3	具有地址改变的循环	47
3-4	不用計数的循环	49
3-5	简单循环的循环专用单元	50
3-6	簡單的計数循环的程序設計	54
3-7	一个求和的循环	54
3-8	移动 (MOVE) 指令	56
3-9	不用計数来終止的循环	57
3-10	非循环的分枝	61
3-11	无循环专用单元而有地址修改的循环	63
3-12	上述指令的符号表示法	67
3-13	无穷級数求和	68
3-14	循环套循环;作一有三个参数的表	71
	习 題	73
第四章	框图,子程序和程序庫	75
4-1	再談框图	75
4-2	按框程序設計	75
4-3	反复使用的框;初等子程序	78
4-4	子程序的进口和出口	79
4-5	一个典型的子程序	83
4-6	子程序的通用化	84
4-7	一个完整的例題	86
4-8	通用子程序的代碼字	86
4-9	程序的說明	91

4-10 程序庫的維護	92
習 題	93
第五章 檢查程序	94
5-1 一般說明	94
5-2 分塊檢查; 試算	95
5-3 存貯內容輸出 (Memory dumps)	95
5-4 抽印 (Snapshots)	96
5-5 插入指令	98
5-6 追蹤	99
5-7 停機代碼	103
5-8 比較法	105
5-9 地址尋找	106
5-10 在控制台上檢查程序	106
5-11 循環的追蹤	108
5-12 檢查包含有子程序的程序	109
5-13 檢查子程序	109
習 題	110
第六章 自動程序設計	111
6-1 一般概念	111
6-2 IT, FORTRAN 和 FORTRANSIT	113
6-3 符號變數	115
6-4 常數	116
6-5 “語句”的概念	116
6-6 源程序及結果程序	119
6-7 算術符號	120
6-8 算術語句	122
6-9 特殊函數	125
6-10 輸入及輸出語句	127
6-11 無條件轉移	129
6-12 暫停 (PAUSE) 和停止 (STOP)	129
6-13 簡單程序的例子	130
6-14 循環語句	131
6-15 數組	134

6-16	維數語句(DIMENSION).....	137
6-17	含有數組的循環.....	137
6-18	分枝及終止循環.....	139
6-19	子程序的分枝出口.....	141
6-20	調整.....	142
習 題		143

第二部分 高等程序設計

第七章	機器語言及SOAP算術指令.....	145
7-1	引言.....	145
7-2	字長、存貯容量及基本寄存器.....	145
7-3	算術運算時累加器的功能.....	147
7-4	650 機指令字的格式.....	148
7-5	最佳存取.....	149
7-6	介紹 SOAP.....	150
7-7	小數點位置.....	151
7-8	加法模式.....	152
7-9	加法、減法及存入的代碼.....	154
7-10	SOAP 卡片的格式；絕對地址.....	155
7-11	SOAP 地址.....	157
7-12	空地址.....	159
7-13	在加減法中的 S001、S002、S003 地址.....	160
7-14	在加減法中連起來用的累加器.....	162
7-15	做加減法時的溢出.....	163
7-16	乘法.....	164
7-17	除法.....	166
7-18	乘或除以 10：移位.....	166
7-19	對變數和常數加比例因子.....	170
7-20	對方程加比例因子.....	170
第八章	機器語言及SOAP指令(續).....	174
8-1	數據卡片格式.....	174
8-2	讀入段.....	175
8-3	讀入指令：輸入卡片.....	176

8-4	重新分配鼓上的信息	177
8-5	从控制台讀入开始卡片	178
8-6	单字輸入卡	178
8-7	其他輸入程序	180
8-8	穿孔段和穿孔指令	181
8-9	分枝代碼	182
8-10	SOAP 程序中的常数	186
8-11	SOAP 中的循环	186
8-12	进一步討論 SOAP 中的伪指令	189
8-13	伪指令 BLR 和 BLA	189
8-14	可用单元表	191
8-15	具有两个意义的符号地址:伪操作 HED	192
8-16	具有同一絕對地址的两个符号:伪操作 EQU	193
8-17	预先指定絕對存貯地址的符号:伪操作 SYN	194
8-18	和 SOAP 一起使用的子程序的两种形式	195
8-19	子程序的进口和出口	197
8-20	空操作和停止	198
8-21	SOAP 汇编的一般过程	199
8-22	調整	200
8-23	省略	200
	习 題	201
	術語詞汇	202

第一部分 初等程序設計

第一章 一般介紹

1-1 什么是电子数字计算机?

电子数字计算机是一种快速处理数的工具。一架快速计算机能在几微秒内完成两个数的加法运算；即使是一架速度比较慢的电子计算机，在一秒钟内也可完成上千次加法运算。用人工来做需时几年的计算量，若用电子计算机来做，则在几小时内就能完成。由于科学技术工作中很多问题的解决都有赖于数值计算，因此，电子计算机现已成为一种重要的研究工具了。

这里用“数字”这个术语是因为它使用了离散的数进行运算。与它相反，模拟计算机则处理连续的数量，如图表、长度、电压等。在本书中凡说到“电子计算机”时，意思就是指的数字计算机。

为了能很好地使用这些工具，必须学习一些新的技术。做一些说明指导，终究还是需要的，即使是学习使用计算尺或台式计算机也是如此。实际上，大多数的大学生，只要经过几小时的讲解和一些时间的实验，就能胜任编制简单而实用的问题的程序；而当他逐渐熟练之后，他最初学习计算技术时所花的时间，与他在计算中所节约的时间相比，将是微不足道的。

当然，在决定学什么时，必需加以选择。程序设计这一领域的发展是很快的；新的技术不断地出现，老的方法则将逐渐地过时。显然，我们不当去学那些过时的技术；另一方面，也往往容易偏于把全部时间花在学习最新的方法上，其结果是不能实际解决任何问题。既然假定读者最感兴趣的是在于解数值问题，所以本书的重点是放在能直接实际应用的技术上。

1-2 程序的概念

我們先看人們是如何使用一架台式計算機的。計算員把數打到鍵盤上，并根据運算要求去按標有 $+$ 、 $-$ 、 $\times$ 或 $\div$ 號的按鈕。于是機器就機械地算出結果。很多的时间是花費在按鍵和記結果上。最熟練的台式計算機計算員們是這樣安排他們的計算：當機器在進行下一步計算時，他們將前面的結果制成表格。這是一種高效率的辦法，因為人和機器幾乎都在不斷地工作。

由於相當多的時間是用在按鍵和記結果上，單靠提高計算機的速度已不能使這一類方法在速度上有明顯的提高。如果電子計算機的每一次運算也用按鍵的方法操作，那末當計算員在記結果時，計算機幾乎是一直在閑等着。在作一次加法所需的按鍵時間里，計算機實際上是能作幾千次加法的。這類計算機的運轉費用很大，這樣做法是非常不經濟的。

設計出一種完全不同的方法已成迫切需要。按照理想，在上計算機解題之前，人應先完成他的那部分工作。一旦計算機開始運算，人的干預應該是越少越好。要做到這一點，計算問題必須事先詳盡地并非常細心地安排好。

高速計算機能執行有限幾種基本運算——一些簡單的如“加法”或“乘法”以及其他較複雜的運算。解任何具體問題必須表示為這些基本運算的序列。這種序列稱為程序，編制它的過程稱為程序設計。在計算機接受程序前，程序必須用計算機結構所能遵循的代碼來表示。

計算的過程非常象鐵路運輸工作，它的綫路（程序）是預先設計好的。計算機從頭開始做并按照程序設計者安排的綫路進行。正象鐵路運輸一樣，計算機也時常停下來輸出或輸入信息。鐵路運輸的一個非常重要的特點是能根據轉轍器扳向那里而選擇運行的軌道。與此完全類似，計算機也可根據一定的開關而執行這一組或那一組運算。計算機本身能扳好在綫路的前面或后面的開關，或者由程序設計者預先扳好這些開關。綫路也能返回來形成

环綫,使计算机在同一段程序上做很多次运算,只有当控制开关扳到相反的方向时,才离开这特殊的指令組。

程序設計問題就是为了要解某类算題而去安排计算机所应遵循的綫路的問題。

程序設計过程中經常会出一些錯誤,因为这非常象和一个机器人打交道,他只具有很少的語汇,而且也只按字面来理解你所提出的一切。他不会發揮一点創造力,所以你必須預先估計到一切可能发生的情况。如果你說坐下,那么不管那里有沒有凳子他都坐下。因此如果你要他坐在凳子上,你必須說“到凳子那里去”,“把凳子拿来”,“坐在凳子上”。如果你說“坐在凳子上”但却沒有凳子,它就要出毛病了。

虽然不高明的程序設計不至于在算題时搞坏计算机,但会使机器算出完全不能理解的結果。计算机会无情地揭露程序設計者在邏輯上的缺点。它只会做那些叫它做的事情,不多也不少。如果它不去做程序設計者所打算叫它去做的事,程序設計者決不能怪別人,只能怪他自己。

1-3 程 序 庫

每当編出了一个程序时,常希望它能反复地用于不同的数据。例如,編了一个矩陣求逆的程序,便希望它能适用于許多式样的矩陣。如果把程序保存起来,那末便可节省重复編程序所需的腦力劳动。

在許多实用問題中某些类型的計算常常会反复出現。当程序設計者編一个求相关系数的程序时,这与他心目中所想解决的农业数据将无关系。这一程序同样也可用于心理学的、工程的或商业的数据。重要的是編好了一个求相关系数的程序。如果这个程序和它的使用說明书一起保存起来,則心理学家、工程师或商业分析者都可利用它而不需另編程序。

每个計算研究室,为了使这类广泛适用的程序可供大家利用,都設有一个程序庫。一个大的程序庫相当于几百小时程序設計工

作的成果，它成为每一个新的程序設計者的继承物。不同的机构只要設備基本上相同，就可以通过交换程序来扩充他們的程序庫。因此每个研究室可以視它的需要得到全国程序設計的成果。有时，只要另外增編很少一点程序，將庫中的若干小程序以适当的順序联起来，便可能解大的問題。

要是沒有程序庫，每一程序設計者必須从头开始工作；有了程序庫，他只需編对他的程序有特殊性的部分。

1-4 数值分析与程序設計

称为数值分析的这一数学分支，是專門从事解决数值解問題的学科。它的发展比电子計算机的出現早几百年，很多的数学家在这一学科上作出了重要的貢獻。数值分析者的任务之一是为求解具体問題而設計計算过程或算法(按它們的正确叫法)。

例如，考虑求一个数的平方根的方法。計算的方法有好几种，其中一个是我們小时候所学的方法，用求 10 的平方根作例子來說明：

$$\begin{array}{r}
 3.16 \\
 \sqrt{10.0000} \\
 \underline{9} \\
 61 \underline{100} \\
 61 \\
 626 \underline{3900} \\
 3756 \\
 \dots
 \end{array}$$

在台式計算机上經常使用另一种方法，即逐次减去相继的奇整数。而在电子計算机上則經常使用第三种方法，称为牛頓法。这种方法是基于按重复的过程有規律地改进一个近似值。当然，我們也可以用对数求平方根。因此我們看到，即使在这样一个簡單的問題上也有相当多的方法可选择。

同样，对求解一組綫性代数方程組如

$$8x - 5y + 2z = 31$$

$$x + y + 9z = 23$$

$$2x - y - 8z = -9$$

也有許多方法。現略舉幾種，就有克萊姆法、主元素消去法、高斯消去法及松弛法等。數值分析者研究這些方法時，特別重視象誤差的估計及收斂的速度等重要問題。

程序設計是算法和計算機間的鏈環。程序設計者必須詳細寫出計算機在實現這個算法中所執行運算的明確的序列。

很多數值分析工作必須走在程序設計的前面。但往往出現這種情況，即在程序設計過程中或已在計算機上進行試算以後，才發生一些數值分析上的問題。數值分析和程序設計通常是互相補充直到問題被解出為止。

許多算法是在使用現代計算機之前建立的，它們不適用於計算機的自動計算。因此目前不斷地設計出各種較適用的算法，並通過使用計算機來研究這些算法的性質。

本書不擬作為數值分析的教科書，所以將不去推導任何標準算法。關於這門學科，已有幾本完整的現代的書可以採用，它們已經是程序設計者藏書中的重要部分。為了實際使用，本書主要將敘述編制這些算法的程序。

1-5 計算機的部件

因為這並不是一本關於製造計算機的書，所以我們將不涉及計算機的內部電子裝置棧路。對我們來說，計算機可看作為一個能服從有限條命令的黑色箱子。我們只要知道它的五個基本部分就夠了：運算器、存貯器、控制器、輸入器和輸出器。

計算實際上是在運算器中完成的。運算器是計算機中處理數並轉換為和、積、商等的部分。

因為運算器一次只能處理少數幾個數，其他計算所用的信息必須保存在別的地方；但當需用時又必須能很快地得到它。存貯器就是為這目的服務的：——在計算過程中，信息從存貯器送到計算機的其他部分以及從其他部分取來放到存貯器中（是由程序來決定的）。其實程序本身也是存放在存貯器中的，這就是我們所要談的這種類型的計算機之所以稱為“存貯程序式”計算