

BASIC 语言
计算机程序设计导引
(上册)

Richard Forsyth 著

卢泽愚译

中国科学院自然资源综合考察委员会
系 统 分 析 训 练 班

一九八〇年八月

前 言

本书的目的是给出Basic计算机程序设计语言的一个导引。分时系统及最近微处理机的流行，已使计算机的能力扩大到愈来愈多的人，其中许多人甚至从未考虑过使用计算机。有许多理由认为Basic是应用这种新出现功能的最好手段之一。Basic的固有结构使它特别适用于初次用户。

Basic虽有其缺点，但具备三个非常重要的长处：（1）易学，在很短时间内就能开始用它做些题目；（2）对要求更高的用户提供了有效的附加功能；（3）它是为交互式会话使用而非为批量作业而设计的。这些优点使它特别适用于非专业人员。因此Basic不断普及是不足为奇的。

进而，我们可看到不久的将来，一台完备的具有足够内存和大量后援存贮的台式计算机将与一台彩色电视机的价格差不多；极有可能五年十年之后这些私人计算机会用Basic编程序。这样一来Basic会变得更加普及。

Basic交互式的性质值得强调，它与另一种“批量”方式大不相同。虽然Basic有批量式的变型，其它许多语言也能用于交互式的运行（还有像APL和POP—2也是按交互式设计的），但是Basic的主要特点之一：它是专为用户与计算机相互会话使用而设计的。这意味着程序员不必予先详细说明每种意外故障，而能在意外情况出现时修改错误或做别的处置。用惯了单纯批量系统的人都会了解这免去了大量的挫折，在批量系统中一个普通的错误可以破坏运行而使最后结果被延误一两天。交互式处理对新用户也极为重要，它允许学员在系统上试验。自然这能显著地加快学习过程。本书始终贯穿着一个主旨：“如有疑问就试试看。”如果你怀疑能否做某些工作，你可编一个小的试验程序去看出现什么结果。可以非常迅速地联机试验，如果不成功，计算机很快就会告诉你。总之，Basic适合于进行实际试验来学习它。

这些特点使Basic成为人机通讯的一种非常有效的工具。对于许多计算任务，应用Basic系统去完成它简直是最容易、最快速而且最经济的方法；随着计算机时间比人的时间更便宜，这一点将更加如此。

因为Basic比一些别的计算机语言更少标准化，所以必须决定讲述什么文本。本书采取的办法是依附于Basic的一个有代表性的实现。就是说，所有说明的程序和练习的答案都在指定的机器——DEC系列—10机上实际运行过。DEC—10机在教学范围内是通用的，而且它的Basic比起其它许多扩充的Basic系统来更接近于Dartmouth学院原始Basic的特点。但是，这表明某些程序例子不经修改不能在别的机器上运行。大多数情况下要做的修改是微小的。第一到第四章讲述的语言基本点实际上是通用的，所有机器都一样；但第五到第八章的许多进一步的特性不是标准的，用户可以发现与其它机器的差别。主要的不同点在有关章节更充分地讲到。一旦再需要了解某些事情在你的机器上

如何办，方法仍然是“试试看”。学习了这里介绍的语言，在应用 Basic 的任何文本时不会有太大困难。

我感谢伦敦市中心商业区的工艺品计算中心，本书所有的程序例子都用他们的 DEC—10机运行过。还感谢S.罗斯，M.怀特，我的妻子和女儿以及两瓶帮助我设计出本书标题的进口葡萄酒。

Richard Forsyth 1977.3.

目 录

前 言

第一章 引论

1.1 分时.....	1
1.2 键盘.....	2
1.3 进入和退出系统.....	3
1.4 Basic 系统	3
1.5 终端会话实例.....	6
1.6 次要的措置.....	8

第二章 基本Basic——程序设计初步

2.1 Basic 的算术运算	11
2.2 LET (赋值) 语句.....	13
2.3 六种基本语句.....	14
2.4 编辑.....	17
2.5 终端会话实例.....	18
2.6 练习.....	21

第三章 循环与数组——迭代过程

3.1 FOR (循环) 和NEXT (循环结束) 语句.....	24
3.2 数组.....	25
3.3 程序例子.....	27
3.4 练习.....	30

第四章 输入与输出——数据传输

4.1 READ (读), DATA (数据) 及 RESTORE (复原) 语句.....	31
4.2 PRINT USING (格式输出) 语句.....	32
4.3 程序例子.....	34
4.4 练习.....	38

第五章 函数和子程序——例程的利用

5.1 GOSUB (转子) 和 RETUEN (返回) 语句.....	40
--------------------------------------	----

5.2	预定函数.....	41
5.3	用户定义的函数.....	42
5.4	程序例子.....	44
5.5	练习.....	47

第一章 引 论

Basic是Beginner's All-purpose Symbolic Instruction Code（初学者通用符号指令码）一词的首字母缩写，是由美国新罕布什尔州Dartmouth学院的Kemeny和Kurtz两教授于1964年设计的一种计算机程序设计语言。在此以后由不同制造厂家和学会改进，必然有相当程度的分歧。因此现在有着若干变型或者说“方言土语”。这里介绍的是用于DEC系列—10机的文本。美国国家标准化协会马上要出版标准Basic的说明书，虽然可能具有某些DEC Basic所没有的附加性质，但是不大可能缺少这里讲的任何功能，两者不会有矛盾。换句话说，标准Basic将是本入门书介绍语言的一个扩充。

Basic最大的吸引力在于它易学。初学者只花一个下午在终端上与Basic系统交互操作，就应该能编写并运行简单的Basic程序。Basic的交互式性质是非常重要的，它是易学的一个原因：它是专为交互式应用设计的，而交互式操作是最好的学习方法。

1.1 分时

图1粗略地画出一个非常标准的中型教学计算机系统中各主要部分之间关系的略图。

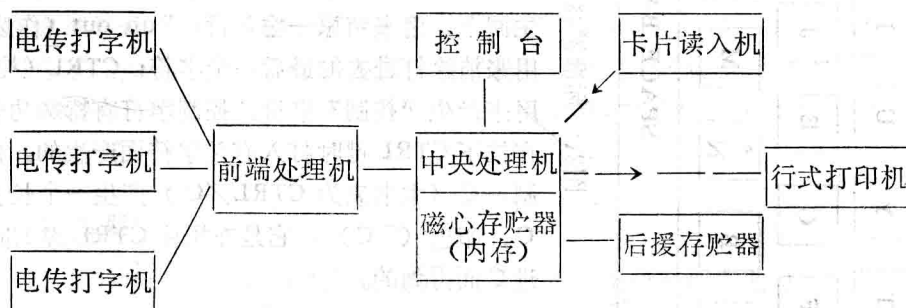


图1. 一个教学分时系统（DEC10）图解

图中各项如下：每个电传打字机代表一个终端，用户通过它与系统通讯；前端处理机是处理终端与中央处理机之间通讯的小型计算机；中央处理机是实际进行运算的地方，所以直接与快速磁心存储器（内存）相连；这里存放着程序和数据（在处理时按形状被认为是这两者之一）；后援存储器（外存）是磁性的后备存储设备，可以长期存放程序和数据；卡片读入机和行式打印机是为较多信息输入输出的外围设备；控制台供操作员控制用。（DEC—10机的后援存储器是磁盘，为更通用，凡是磁盘都译作较广泛的后援存储器）。

图1不是准确的设备图，只是系统各主要部分及其间关联的略图。特别是只画出三个终端，实际上在任何时候联上系统的终端数从没有到大约60个不等。

一台计算机能同时为许多终端服务，所用方法称为分时，实质上就是把主处理机的

时间分配给各个不同的任务。因为处理机很快，虽然计算机一般说来对任一终端只花费很少的时间比例（通常每秒钟只用 $1/50$ 秒），但除了非常高的负载以外，用户很少觉察到有任何中断。一项作业在没处理的其余时间里被挂起，它的程序、数据和有关信息都保存在后援存贮里，因此一当下次再轮到处理它时可以在正确的位置上重新开始运算。

DEC 10上的分时是由一组统称为监督程序的若干管理程序控制的，用户靠发布监督程序命令与系统相互联系。监督程序控制任务间的资源分配，在机器运行的任何时候都在起作用，它是DEC 10的操作系统。

1.2 键盘

对于 Basic 用户而言，终端是与计算机通讯的主要工具，所以必须熟习它的键盘。图 2 表示一种标准的键盘配置图，虽有其它的排列但这是最通用的。

字母和数字的位置与普通打字机相同，但还有一些特殊按键。最重要的有：Return（回车）键产生回车，用来结束一输入行；Rub-out（作废）键用来清除打进去的最后一个字符；CTRL（控制）键用来产生“控制”字符。控制字符有特殊功能，是在按下 CTRL 键时打入有关字符而产生的。例如控制—C（本书表为 CTRL/C）产生一个长音符和 C，印出（ $\sim C$ ），它是在压住 CTRL 键的同时打进 C 而得到的。

CTRL/C使正在等待输入的 Basic 程序停止运行；CTRL/O禁止计算机印出任何内容；CTRL/U删除输入的一整行；等等。这些特殊字符可以在 Basic 系统内部应用。附录 C 中更详细地介绍它们。

对初学者来说必须要会用键盘，这是练习的主要要求。虽然不需要成为打字明星，但反过来说，如果你要几秒钟才找到一个字符，那末你在终端对话期间不会有乐趣或成果。

特别注意数字 1 与字母 I，数字 0 与字母 O 是不同的。在计算机上这些差别非常模糊，如果你想按的键按错了就会出问题。不幸，除了位置不同外所有键盘都没突出这些差别，因此要注意它们所在

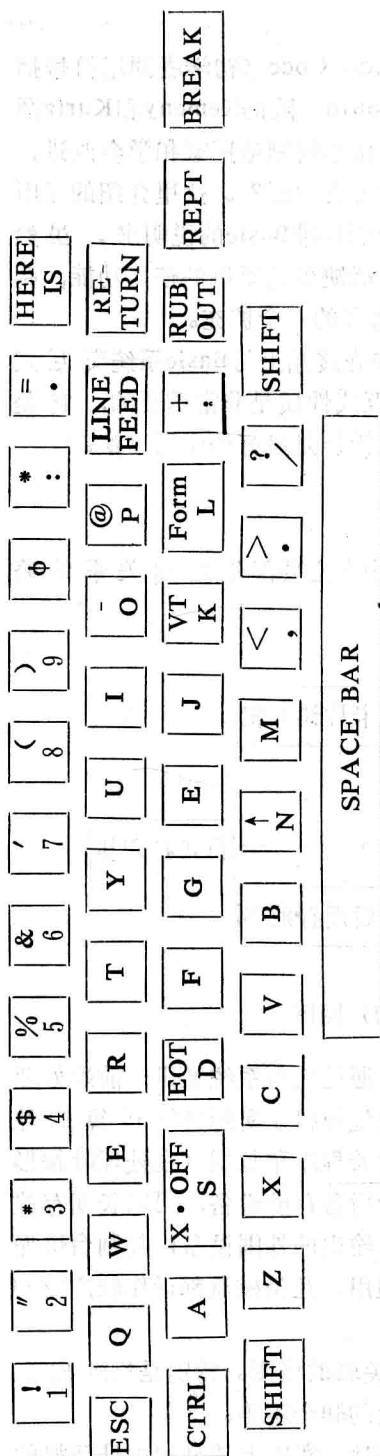


图2. 标准键盘配置图

的地方。在准备程序时奉劝你采取如下手写的约定，在把它们打进计算机时会省去许多麻烦：数字写 ϕ ，1，2，字母写O，I，Z。

1.3 进入和退出系统

为能访问计算机几乎总要通过某种称为“注册”的定名过程来实现。DEC10的注册过程在下面简单介绍。大多数分时系统的注册过程有同种的外形但有若干不同，在进行下去以前要查对你在所用机器上的帐号、口令等细节。如果你还没有的话，必须得到一个用户号或用户名。

对新来者，特别是不大会打字的人来说，注册可能是个负担，但要在计算机上做有用的工作这是必不可少的。

1.3.1 DEC10的注册

首先保证你的终端真正与计算机联上，这包括接通线路并检查路线。

当计算机在左边印出一个圆点（句号）表示它准备就绪时，输入LOGIN（一个字，不是两个字），然后按回车键。

当计算机印出编号符（*）后，你要回答用户号，在DEC10上它是要与计算机中一致的由逗号隔开的两个数，例如17022，1。

当系统印出PASSWORD（口令）时，给出你选用的口令（1—6个字符），以让计算机确认你有权使用那个用户号。为保密起见，口令在终端不反应出来。

现在你可得到一个引导信息，表出年月日，星期和时间，并在一新行开始跟一句点。如果你不工作，就按CTRL/C键以作废注册并回复到初始状态；如果你要工作，则先输入BASIC，再按回车键就进入到了Basic系统。系统应回答READY（就绪），你就能应用Basic系统了。

1.3.2 注销

几乎所有Basic系统对BYE（再见）命令的反应都是用户有秩序地脱离系统。但有的机器（不是DEC10），BYE命令只产生退出Basic子系统；要整个脱离系统还要别的命令。无论哪种情况，重要的是不要忘记注销。

不注销而离开联机的终端，会浪费时间和费用，而给下一个使用终端的陌生人提供了清除或破坏你存贮的程序及文件的可能性；很少有人能完全抵制不干这种恶作剧。

在1.5节将会看到一个从注册到注销的完整的终端对话期间。

1.4 Basic系统

我们开始已说到Basic系统和Basic语言。在你试图写出并运行自己的Basic程序之前，必须要知道有关该系统工作的一些事情。

表现给用户的DEC10Basic系统大致如图3，一个普通用户不需要比它更详细的图形。但是应该了解当你输入Basic程序，并且运行、存放它的时候，并不是直接与计算机而是与一些非常复杂的软件打交道。例如，计算机不能遵照组成Basic程序的一系列

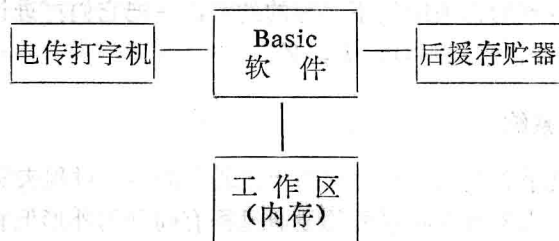


图3. Basic系统的说明

语句行事，而必须先把它们翻译成机器指令。进行这项工作的程序是 Basic 软件的一部分。整个Basic软件可分成三个主要的子系统：（1）命令译码程序，它接受并执行运行程序，把程序存入后援存储器以及从后援存储器调出程序等等命令；（2）行编辑程序，能使你将内存工作区中的程序按行删除或修改；（3）编译程序，将Basic程序翻译成机器能懂的形式。借用监督程序的BASIC命令即可调用该系统。

上述系统的主要部分都是对处于终端的用户而言的，用户与Basic软件相互作用以能在后援存储和工作区之间移动程序，或者运行、编辑在工作区里的程序，而后援存储可以保存程序和数据供以后使用。重要的是弄清工作区与后援存储之间的差别：工作区中的程序能够运行或者编辑，而后援存储只能保存它们。在你执行或修改一个程序以前，必须先把程序放入工作区。这可以从终端输入；如果后援存储中存放有程序的话，也可以从那里调出。后援存储的用处就在于允许你在终端对话期之间（即两次用机的时间之间）保存程序，因此程序运行时可以保存它以备后用，而不必每次都要重新输入它。

1.4.1 使用Basic系统

使用Basic系统时，可处于两种状态之一。

（1）输入状态：Basic接收输入行到工作区去，同时执行命令。

（2）执行状态：对当前在工作区的程序进行编译并按此执行。

在输入状态（1）下，Basic将任一开头有整数（行号）的行看做一行送到工作区去，此数标出存放位置。任何开始没有整数的行都被看成命令。如果Basic不认识某个命令，它只抱怨（印出：？WHAT？）并不受你偶然打入胡扯命令的影响；但是，如果你输入一行号再跟上一些零乱字符，则此行将原封不动地送入工作区的适当位置去，而且直到要执行它以前不会引起错误。（有些Basic系统，每输入一行就翻译此语句，但DEC10要等到执行程序前才检查语法错误。）

在执行状态（2）下，Basic按当前在工作区的程序运行。

要从状态（1）转到状态（2）需打入RUN（运行）命令。要强行从状态（2）退出并返回到状态（1）需按两次CTRL/C键，Basic系统将印出READY（就绪）。正常情况下，程序执行完毕后，Basic回到状态（1）。

注意，“语句”（或“指令”）是构成Basic程序的编号行，可以用命令RUN去执行它；而“命令”是没有编号的行，它指示Basic系统下面干什么而立即被执行。

你可以对系统以任何顺序打入有编号的语句而输入一个Basic程序——在工作区一行一行地顺次排好。另外，如果后援存贮中已有程序，也可以用OLD（旧程序）命令去调用它。一旦程序在工作区，就可用命令RUN去执行它。

1.4.2 一些Basic命令

附录C中包括全部的Basic命令。下面给出的是少数现在就需用的基本命令，这里写出系统可接受的三个字母缩写，完整的形式在附录C中。

命令	意义
BYE（再见）	脱离系统
CAT（目录）	列出用户在后援存贮中的存放目录
LIS（列表）	列出工作区中的当前内容
NEW程序名（新程序）	清工作区以输入具有指定名的新程序
OLD程序名（旧程序）	恢复后援存贮中的指名程序并放到工作区去
RUN（运行）	执行当前在工作区的程序
SAV程序名（存贮）	将工作区的内容按指定的名称及“扩充名”存放在后援存贮中
UNS项目名（清除）	清除后援存贮器中的指定项目

表中的程序名、项目名之类的说明项，对DEC10而言是跟在命令名后隔一空格（或文件）的说明。系统识别每个程序要通过一个名字以及用圆点隔开的“扩充名”。名字是由用户选用的1—6个字母或数字；扩充名可以是0—3个字符。如果用户没给出扩充名，则Basic系统指定它为BAS。一个典型的说明项可以是TEST1.BAS，它可出现于如下命令中

OLD TEST1.BAS

这就告诉系统从用户的后援存贮区找到TEST1.BAS，并把它调入工作区（其中任何内容都被重写了）。

如果在命令OLD和NEW中丢掉了说明项，Basic系统将要你补上；但如果在命令SAV和UNS中漏了它，系统就假定是工作区中无论什么内容的当前名和扩充名。在给出命令之前，你总要考虑它将对工作区的内容及后援存贮中的存贮有什么影响。

1.4.3 Basic程序库

如果让CAT（目录）命令加上三个星号（CAT***），就得到能为所有用户访问的公用的Basic库程序的目录。这个程序库中包括许多实现公共要求的计算，以及有助于使用键盘和系统的一些策略与说明。要得到它们中的某个程序，需要在OLD命令的说明项加上三个星号。例如OLD MASTER • CLP***，就从Basic库中将程序MASTER • CLP加载到工作区，以备由命令RUN运行。

所有分时系统都保存有某种供公共访问的Basic程序，但是调用它们的准确方法，以及库存内容会因设备而异。

1.5 终端会话实例

下面是一个完整的终端会话的输出，提供了正确地注册和注销的例子，并使你看看Basic程序是什么模样。如果你还不明白它如何工作的话，不必着急，在下一章将解释所用的语句。现在你只需注意：Basic 程序是由一系列语句组成的，每一语句均前置一个行号。

在本例中，与以下本书各处的例子一样，由用户打入的部分加黑点印出以区别于计算机的输出。由计算机印出的一切均只用白体，并不加黑点。

• LOGIN (注册命令)

JOB 9 CITY POLY MONITOR 507B10 TTY146 (计算机印出街区、终端号，
工作号；管理程序号)

*311/1 (用户名)

PASSWORD: (口令：用户的回答因保密不印出)

1730 12-MAR-76 FRI (时间，年月日，星期)

• BASIC (Basic命令)

READY, FOR HELP TYPE HELP. (就绪，如要帮助打进HELP命令)

OLD DEGREE.XXX (调旧程序命令)

READY

LIS (列表命令)

DEGREE.XXX 17:31 12-MAR-76

10 REM *CENTIGRADE/FAHRENHEIT CONVERTER (摄氏、华氏温度
换算)

20 REM $F = 9/5 * C + 32$

30 REM $C = (F - 32) * 5/9$

40 PRINT "WHEN THE COMPUTER TYPES? GIVE DEGREES SCALE"

(当计算机印出? 时，给出温度与标度)

50 PRINT "SCALE SHOULD BE 1 (C) OR 2 (F)" [标度为1 (摄氏)
或2 (华氏)]

60 PRINT

100 INPUT D, S

110 IF S=1 THEN 200

120 IF S=2 THEN 300

130 PRINT "2ND NUMBER MUST 1 (C→F) OR 2 (F→C)" [第二个
数必为1 (摄变华) 或2 (华变摄)]

140 PRINT "PLEASE TRY AGAIN" (请再试)

150 GOTO 100

200 REM *CENTIGRADE TO FAHRENHEIT: (摄氏变华氏)

210 LET C=D

```

220 LET F = 9/5 * C + 32
230 GOTO 400
300 REM * FAHRENHEIT TO CENTIGRADE:      (华氏变摄氏)
310 LET F = D
320 LET C = (F - 32) * 5/9
400 PRINT C "CELSIUS IS" F "FAHRENHEIT"
410 IF C < -273 THEN 430
420 GOTO 100
430 PRINT "WHICH IS BELOW ABSOLUTE ZERO!" (低于绝对0度)
440 REM * STOP HERE (TOO COLD)           [停机(太冷)]
999 END
READY

```

RUN (运行命令)

DEGREE.XXX 17:32 12-MAR-76

WHEN THE COMPUTER TYPE? GIVE DEGREES, SCALE

SCALE SHOULD BE 1 (C) OR 2 (F)

? 0, 1

0 CELSIUS IS 32 FAHRENHEIT (摄氏0度是华氏32度)

? 37, 1

37 CELSIUS IS 98.6 FAHRENHEIT (摄氏37度是华氏98.6度)

? 0, 2

-17.7778 CELSIUS IS 0 FAHRENHEIT (摄氏-17.7778度是华氏0度)

? 100, 1

100 CELSIUS IS 212 FAHRENHEIT (摄氏100度是华氏212度)

? 100, 2

37.7778 CELSIUS IS 100 FAHRENHEIT (摄氏37.7778度是华氏100度)

? -4, 0

2ND NUMBER MUST BE 1 (C->F) OR 2 (F->C)

PLEASE TRY AGAIN

? -40, 2

-40 CELSIUS IS -40 FAHRENHEIT (摄氏-40度是华氏-40度)

? 32, 2

0 CELSIUS IS 32 FAHRENHEIT (摄氏0度是华氏32度)

? 20, 1

20 CELSIUS IS 68 FAHRENHEIT (摄氏20度是华氏68度)

? -460, 2

-273.333 CELSIUS IS -460 FAHRENHEIT (摄氏-273.333度是华氏-460度)

WHICH IS BELOW ABSOLUTE ZFRO! (低于绝对0度!)
 TIME: 0.38 SECS. (程序执行时间: 0.38秒)
 REACY
 BYE (再见命令)
 JOB 9, USER [31.11] LOGGED OFF TTY146 1735 12-MAR-76
 (印出用户、终端号, 注销时间)
 SAVED ALL FILES (197 BLOCKS) (全部文件存在197区)
 RUNTIME 1.34 SEC (运行时间1.34秒)

在这个简单的会话期中, 用户以户号311,1注册, 然后进到Basic系统。叫做DEGREE.XXX的程序是用OLD命令从后援存贮中恢复并负载到工作区去的。此程序是换算华氏和摄氏温度。它要求用户给出两个数: 温度和标志测温标度的数字, 然后程序印两种标度的温度值。如果温度低于绝对零度, 则程序转换了数值后即停机; 否则还要讯问下次的输入。程序先用LIS命令印出, 再由RUN命令运行。最后由打入命令BYE而退出系统。

1.6 次要的措置

要取得在终端上的技巧需要时间。学习Basic系统的最快方法是实践, 因其交互式设计, 此系统本身就有点类似教学机器的作用。在你胡乱地使用键盘以前, 要阅读这一节讨论的初学者遇到的一些共同问题。

1.6.1 实用的细则

为了使用计算机, 程序员必须熟习一定的看来好像不足道的技巧。首先你应该确有把握知道如何做好下述事情:

- (1) 建立与系统的联系: 这可能包括拨通电话网, 检查线路及终端的故障。
- (2) 进入系统: 这可能包括申请(或购买)用户号, 账号或别的什么要求。还包括按一定的格式和次序打入正确的上述编码和口令。这里你首先要注意计算机非常苛求于一些微小的细节, 诸如逗号, 空格, 新行等等。
- (3) 进入Basic系统: 这可能在注册时自动进入; 但对具有多种可供选择语言的大系统来说往往不能自动实现。
- (4) 正确地注销并退出系统。
- (5) 调用存贮的程序(包括库程序)。
- (6) 执行工作区中的程序。
- (7) 印出程序。
- (8) 存放程序以备后用。
- (9) 从长期存贮中删除程序。
- (10) 在出现某种错误时停止运行的程序。
- (11) 在输入未传送到计算机以前, 从你当时打入的行中清除一个或多个字符, 或者清除一整行。

(12) 认识大多数较常用的错误信息，并了解其含义。

建议你最好在一张纸上或旧卡片上写出这些操作的简短注意（加上例子），在前几个星期带着它，直到没用为止。卡片上还应有你的用户编号，但口令要记在心里。

这些必要点在从未用过计算机的人们中，能引起大多数的混乱和麻烦。学习Basic的困难比较而言是较小的。

重要的事情是：得到会话运算的实感；了解键盘及各个键的作用；熟习诸如注册、注销的基本过程；在进行自己的程序设计前要有一定使用计算机的能力。如果你在输入、编辑和执行你自己的程序时，对系统还很陌生，那你会经受到不必要的挫折。练习这些基本技巧的一个好办法是运行某个公共库的程序。

1.6.2 程序设计

计算机可以用，也可以被错用。你与计算机开始会话时要避免两种窘态：其一是坐在终端而不知所措；其二是过急地试图解决太复杂的问题而败兴。一方面第一次遇到终端时不要发呆，另一方面在你初次会话期间也不要奢求。

还要记住程序设计是解决问题的训练，不仅仅是编制的问题。一旦你分析了问题，编制解法相对说来是容易的；但是如果你没有弄清问题或者严格说来你还不明白要计算机给你干什么，就很难编写也容易出错。

好的程序设计有一定原则，这个题目在第8章再用较大篇幅讨论。这里只消指出，许多初学者（正象许多有经验的程序员做过的那样）忽视这些原则，大都因为他们急于得到运行的程序，而很少花时间去分析此程序所要解决的问题。事先仔细考虑所花的时间往往因节约其后修正粗糙程序设计带来错误的时间而得到相当大的补偿。

每个程序员形成各自的风格，但最好一开始就解决尽可能有条理的问题，不要忽略通过预备分析的重要性。

第二章 基本 Basic

要让计算机为你计算，必须严格指定它需实现的运算顺序。换句话说，必须用它能懂得的语言给出程序（一系列指令）。这种语言之一就是Basic。Basic程序由一组有序的“语句”构成，每一语句指示计算机进行一步运算。这些语句是英文的关键字，如LET，PRINT等，有时还跟以数学表达式或别的符号，每语句总要前置一个行号。

因此，利用Basic要计算机解决特殊问题，首先必须把问题分解成详细的一系列小步骤，然后依正确的顺序写出做这些步骤需要的语句。这就是该项任务的Basic程序，再把它输入系统并试运行；如不能完成任务就修改它，直到得出一个成功的文本，可以存入外援贮器以备下次再用。这一过程如图4所示。

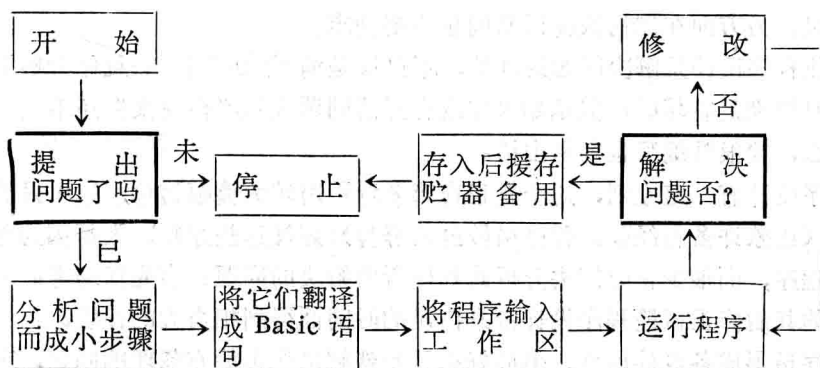


图4 用 Basic 编程序的过程

当然这只是一个简化的略图，但可得到两个要点：首先，程序设计技巧（或科学或业务）是由若干多少不等的划分明确的阶段所组成的一个过程；此过程基本上是循环的。程序设计过程的循环性是个关键性质：一旦程序在工作区，程序员可以反复地检验它，修改它，直到完满完成并存入后援存贮为止。

还要注意此图形采用了流程图（或称程序框图）的形式。在初始阶段，即阐明问题阶段，程序框图常常是有用的。框图非常简明地显示了过程的主要步骤，区别了判别（用粗方框表示）与行动或运算（用长方框表示），并且用箭头联系它们以表示执行的次序。框图对于开初帮助分析问题和最后表示解法的逻辑这两方面往往都是好办法。框图有它自己的标准及约定，这里不更多地讲它。但初学者仍然可以看到仅用判别框和运算框的简单框图也是很有用的。

本章介绍七个基本的Basic语句。它们构成了本语言的基础，在此之上还可发展。只用这七个语句有许多程序设计任务很难完成或者根本不能完成；但同样，它们也能完成非常广泛的种种任务。而且，很难发现任何大小的Basic程序不用这几个语句的大多数或者全部。因此，你会看到不了解所有这些语句就难于写出Basic程序，同样你也会看到只用它们就可能写出许多有意义的程序。顺便提一下，1.5中的程序例子DEGREE，XXX用到了全部这七个语句，同时没有别的语句。

2.1 Basic的算术运算

许多人认为计算机只是一个超型计算器，是除了算术运算（或数字捣弄）外，不能干别的事情的大机器。后面（例如第六章）就会了解，Basic还可用于其它类型的 数据处理，当然数值计算是非常重要的，Basic为此提供了广泛的功能。

2.1.1 Basic的数

Basic中的数值常数写成一串十进数字，前面可随便加一符号，还可随便含有一小数点。它们的后面还可以跟着指数部分——字母E后一任选符号，再后跟一位或两位数字。常数的引导0（即前面的0）没有意义。Basic中合法的数值常数例子如下：

整数：0, 1, -999, +1, 12345678, 002, 200

定点数：0.7, 1.6, -89.305, 1000.0001, 72.0, -9.87654321

浮点数：1E7, 0.314159265E01, 4E+2, 5E-11, -99.99E9

（注意：这里定点数、浮点数与一般的叫法不同，这两种数一般称为基本实数和带指数的实数）。

一个数内部不允许逗号，因此，一万2千是12000，而不是12,000。上面插入的逗号只用做数间的分隔符号。在浮点记法中，E代表“乘上10的多少方”，是用作比例标度的。于是1E7等于 $1 \times 10^7 = 10000000$ （1千万）；0.314159265E01是 π 的近似值3.14159265。有时这种记法被称为“科学记数法”。

在机器内部，数按两部分（小数部分和指数部分）存贮，但初学者不必管它。DEC10的Basic中最大可能的数约为1.7E38（即17后跟37个0），但最小数近似为1.4E-39。更大或更小的数将分别引起“上溢”或“下溢”。在出现上溢时，印出一个警告并用计算机能保存的最大数（1.7E38）进行计算；在出现下溢时，印出一个信息，并用0进行计算。负数限制在带有负号的相同范围内。

写出多于8个有效位的常数，虽然是合法的，但没有什么意义，因为计算机的计算限于准确到8位数字。（要写出能够克服这里提到在大小和精度方面限制的Basic程序也是可能的，但要求格外精心地编程序。）

2.1.2 变量

程序中一个数表示固定的值，这是常数。在程序执行期间值可变化者称为变量。

在Basic中，变量是给出的名字，它可以是单个字母或者一个字母后跟一个数字。于是A1, B, X, Z9都可以是变量名，而1A, QB, X50, Z99都不是。变量名是由程序员选用的（往往带有助记忆的关系）以表示一个变化的量。你可以把变量看成是保存一个数的存贮单元，在计算过程中可以改变。程序中用到一个变量名就表示该单元中当前保存的数，（在不同上下文中用到它，也改变了该单元的内容）。

注意：在DEC10的Basic（并非所有的Basic文本）中，数值变量在程序运行之初都予置0。但是，依赖这个性质是愚蠢的，较好的应用是由你给所有变量置初值，那怕开始就是0。）

2.1.3 算术表达式

Basic的算术表达式尽可能严格地遵从通常的代数习惯，其中限制所有的符号必须摆在一条线上，而且所有运算符都是明确的而非隐含的。基本的算术表达式（或数值公式）取如下两种形式之一：

项

项 运算符 项

这里的项是常数或者变量，运算符是算术运算符之一。较复杂的表达式可以由完整的表达式去代替项而构成，根据的模式是

表达式 运算符 表达式

这里的表达式是上述两种简单形式之一，或者它本身也是一个较复杂的表达式。表达式及子表达式可以用括号括上。

算术运算符按优先次序是

^ 或者 * * 乘方

* 和 / 乘法和除法

+ 和 - 加法和减法

优先次序是重要的，在计算表达式时，Basic遵照如下规则去产生最后结果：

(1) 计算括号中的子表达式，如果有的话。

(2) 计算任何乘方（从左到右）

(3) 计算乘法和除法（从左到右）

(4) 计算加法和减法（从左到右）

于是 $A + B * 2$ 表示先算B乘以2，然后加A；而 $(A * B) * 2$ 表示先算 $A * B$ ，然后将此和乘以2。注意如果没有适当的优先次序，可以用成对括号来强行进行要求的任何次序的运算。由于加减号可以担任三种不同的角色，使表达式有些复杂。例如减号的三种角色是：做为一个数的符号（-5），只有一个运算元素要求的负运算符（-B），以及减法运算符（5 - B）。由适当插入0和括号可以避免混乱。对初学者来说，最好的办法是避免太长太复杂的表达式，而把它分成较简单的步骤，并在可能出现含糊的地方，那怕并不一定如此，都插入括号。

下列例子有助于弄清楚这些规则，其中假定 $A = 1$ ， $B = 2$ ， $C = 10$ ， $X = -1$ ， $E = 0$

表达式	结 果
-----	-----

+ 1	1
-----	---

X	-1
---	----

- A	-1
-----	----

- X	1
-----	---

A + 10	11
--------	----

A - B	-1
-------	----

C * 10	100
--------	-----

A / B	0.5
-------	-----