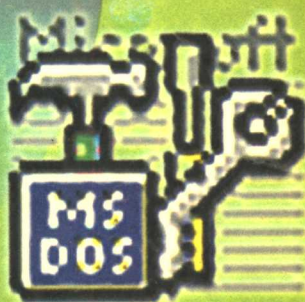
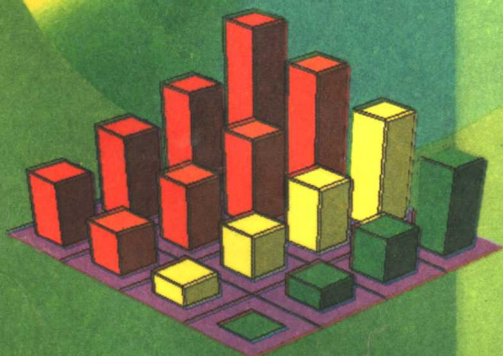


Quick Basic

语言程序设计教程

主编 屈玉泉



Microsoft
Quick Basic

科学技术文献出版社

73.9621

QYQ

QUICK BASIC 语言 程序设计教程

主 编:屈玉泉

副主编:别祖杰 梁玉前 龙冯文

编 者:别祖杰 屈玉泉 龙冯文
崔轩辉 梁玉前 曾元贵

科学技术文献出版社

(京)新登字 130 号

图书在版编目(CIP)数据

QUICK BASIC 语言程序设计教程/屈玉泉主编. -北京:
科学技术文献出版社, 1996
ISBN 7-5023-2840-8

I. Q... I. 屈... III. BASIC 语言-程序设计-教材 IV.
TP312BA

中国版本图书馆 CIP 数据核字(96)第 15833 号

科学技术文献出版社出版

(北京复兴路15号 邮政编码 100038)

中国科学技术信息研究所重庆分所印刷厂印刷 新华书店重庆发行所发行

1997年1月第1版 1997年1月第1次印刷

787×1092毫米 16开本 18印张 438千字

印数:1—5000册

定价:19.80元

前 言

随着科学技术的迅速发展,计算机在经济与社会发展中的地位越来越重要,未来的经济是知识化的经济和数字化的经济。面向 21 世纪,计算机知识与应用能力不仅是本、专科学生必备的,而且已成为国家公务员、企事业管理者和广大技术人员任职的条件之一。在计算机技术普及应用中,具有“大众化”特色的 BASIC 高级编程语言起了重要作用,它本身也在快速发展,正如专家所言“BASIC 大旗不倒”。

Quick BASIC 是美国 Microsoft 公司 1987 年 11 月推出的,它吸收了 FORTRAN、PASCAL、C 及 Ada 等众多高级语言的优点,成为运算速度快、功能强、易学易用的结构化程序设计语言。

Quick BASIC 采用下拉式菜单,对话方式操作,对源程序解释与编译并行,它把源程序的编辑、编译、连接、调试有机地结合起来,构成一个用户所期望的灵活的集成编程环境;它既具有解释型 BASIC 的方便性,又具有编译型 BASIC 执行的快速性,并能通过简单操作生成在 MS-DOS 下直接执行的 EXE 文件。Quick BASIC 具有很强的库管理功能,利用这种功能可把编写的过程和例行程序变为 Quick BASIC 的虚拟扩充部分,使用户能更有效地组织程序。

跟以前的 BASIC 版本相比,Quick BASIC 对语言功能作了许多扩充,数据类型更全,流程控制更灵活,模块独立存储更方便实现结构化程序设计,增强了非数值计算(如文字、图形和声音处理)的功能,可充分利用内存资源使源程序不受限制,灵活的联机求助功能以及与以前版本程序的兼容性,等等。它具有其他高级语言的许多优点,使用更方便,适合于开发各种应用软件。

本书以应用为出发点,内容由浅入深,循序渐进,既强调基础知识的系统性,又突出基本技能的培养。叙述采用从具体例子引出基本概念,比较自然地得出结论的方式。为使读者获得编程能力训练和养成良好程序设计风格,书中给出不同类型可直接在机器上运行的程序实例,同时又提供各部分上机实习内容,不但方便课堂教学,而且适合读者自学和实际操作。笔者在编写中注意了内容的科学性、实用性、通俗性,同时将多年教学经验和自己的成果融入,使其具有一定的特色。

本书是几位多年从事高级语言教学和软件开发应用的教师集体辛勤劳动的成果,在主编拟定编写大纲的基础上,经反复讨论修改,再分工

编写，最后由主编审订定稿。编写组分工如下：主编屈玉泉副教授编写第四、五章；副主编别祖杰副教授、梁玉前和龙冯文分别编写第一、二、三章，第十、十一章和第六、七章；崔轩辉编写第八、九章；曾元贵副教授编写第十二、十三章；全组共同编写第十四章。本书的编写和出版，得到重庆石油高等专科学校教务处的的大力支持，也得到了重庆钢铁高等专科学校和重庆建筑高等专科学校的支持和帮助，在此一并表示感谢！

编者热切希望本书在推广 Quick BASIC 高级语言、促进大中专院校计算机教学改革中起到积极作用。无论是学过 BASIC 有一定基础的读者，还是初学者，学习本书会有新颖之感，更可向 Visual BASIC for Windows 顺利过渡。

由于编著者水平有限，书中难免有错谬疏漏，恳请读者批评指正。

编者

1996 年 9 月

目 录

第一章 QUICK BASIC 概论	(1)
§ 1.1 计算机语言简介	(1)
1.1.1 面向机器的语言	(1)
1.1.2 高级语言	(2)
§ 1.2 BASIC 语言的发展	(3)
1.2.1 初期 BASIC	(3)
1.2.2 中期 BASIC	(3)
1.2.3 近期 BASIC	(4)
1.2.4 最先进的 BASIC 版	(4)
§ 1.3 Quick BASIC 特点	(5)
1.3.1 Quick BASIC 在编程环境方面的特点	(5)
1.3.2 Quick BASIC 在语言功能上的扩展	(5)
§ 1.4 Quick BASIC 的系统文件及安装	(6)
1.4.1 Quick BASIC 的系统盘	(6)
1.4.2 Quick BASIC4.5 以上版本包含的文件	(6)
1.4.3 Quick BASIC 系统的安装	(7)
§ 1.5 Quick BASIC 启动	(7)
§ 1.6 Quick BASIC 屏幕环境	(8)
1.6.1 Quick BASIC 主屏幕	(8)
1.6.2 Quick BASIC 菜单系统	(9)
1.6.3 Quick BASIC 对话框	(11)
§ 1.7 Quick BASIC 的退出	(11)
§ 1.8 编辑、运行一个 Quick BASIC 源程序	(11)
习题一	(12)
第二章 Quick BASIC 语言的基础知识	(13)
§ 2.1 BASIC 源程序结构	(13)
2.1.1 BASIC 源程序的语句行结构	(13)
2.1.2 Quick BASIC 源程序的模块结构	(13)
§ 2.2 Quick BASIC 字符集	(14)
§ 2.3 数据类型	(14)
2.3.1 数值型	(15)
2.3.2 字符型	(15)
§ 2.4 常量	(16)
2.4.1 数值常量	(16)
2.4.2 字符常量	(16)
2.4.3 符号常量	(17)
§ 2.5 变量	(17)
2.5.1 变量的类型	(17)

2.5.2 变量的命名规则	(17)
2.5.3 变量的取值	(18)
§ 2.6 标准数值函数	(19)
§ 2.7 BASIC 表达式	(19)
2.7.1 算术运算	(20)
2.7.2 关系运算	(20)
2.7.3 逻辑运算	(21)
习题二	(21)
第三章 结构化程序设计的基本概念	(23)
§ 3.1 算法的概念	(23)
§ 3.2 算法的描述	(24)
3.2.1 顺序结构的 N-S 图	(24)
3.2.2 选择结构的 N-S 图	(24)
3.2.3 循环结构的 N-S 图	(25)
§ 3.3 结构化程序设计	(26)
3.3.1 软件工程要求	(26)
3.3.2 结构化程序设计方法	(26)
习题三	(27)
第四章 顺序结构程序设计	(28)
§ 4.1 顺序结构处理模式	(28)
§ 4.2 赋值语句(LET)	(29)
§ 4.3 输出语句(PRINT)	(30)
4.3.1 语句格式	(31)
4.3.2 标准输出格式	(31)
4.3.3 紧凑输出格式	(32)
4.3.4 混合输出格式	(33)
4.3.5 用输出函数控制的输出格式	(34)
§ 4.4 数据输入语句	(36)
4.4.1 键盘输入语句	(36)
4.4.2 读数语句(READ)和置数语句(DATA)	(38)
4.4.3 恢复数据指针语句(RESTORE)	(39)
§ 4.5 END、STOP、REM、CLS、SLEEP 语句	(40)
4.5.1 END 语句	(40)
4.5.2 STOP 语句	(40)
4.5.3 REM 语句	(40)
4.5.4 CLS 语句	(41)
4.5.5 SLEEP 语句	(41)
习题四	(41)
第五章 选择结构程序设计	(43)
§ 5.1 GOTO 语句	(43)

§ 5.2 行 IF 语句	(44)
5.2.1 语句格式及功能	(44)
5.2.2 举例	(45)
§ 5.3 逻辑表达式	(46)
5.3.1 关系运算符和关系表达式	(46)
5.3.2 布尔表达式	(47)
§ 5.4 块 IF 语句	(49)
5.4.1 格式与功能	(49)
5.4.2 举例	(50)
§ 5.5 选择结构的嵌套	(51)
5.5.1 行 IF 语句的嵌套	(51)
5.5.2 块 IF 语句的嵌套	(52)
§ 5.6 多路选择结构	(54)
5.6.1 多分支 IF 结构	(54)
5.6.2 SELECT CASE 语句	(56)
§ 5.7 多分支(开关)选择语句(ON-GOTO)	(58)
§ 5.8 综合应用举例	(60)
习题五	(63)
第六章 循环结构程序设计	(66)
§ 6.1 关于循环的概念	(66)
§ 6.2 FOR 循环语句	(66)
6.2.1 FOR 语句格式	(66)
6.2.2 循环执行过程及使用说明	(67)
6.2.3 FOR 语句的嵌套(多重循环)	(69)
§ 6.3 WHILE-WEND 语句	(71)
§ 6.4 DO-LOOP 语句	(73)
6.4.1 DO 语句的四种格式	(73)
6.4.2 简单 DO 循环语句及两个退出循环的语句	(75)
6.4.3 DO 循环的使用说明	(76)
§ 6.5 循环结构应用举例	(77)
§ 6.6 循环优化问题	(81)
习题六	(84)
第七章 数组	(87)
§ 7.1 数组与下标变量的概念	(87)
§ 7.2 数组说明语句(DIM)	(88)
7.2.1 常规 DIM 语句	(89)
7.2.2 语句功能及使用说明	(89)
7.2.3 一维数组应用举例	(90)
§ 7.3 Quick BASIC 的扩充 DIM 语句	(91)
7.3.1 扩充 DIM 语句	(91)

7.3.2	LBOUND 和 UBOUND 函数	(91)
7.3.3	静态数组和动态数组	(92)
7.3.4	数组删除语句 ERASE 和重定维语句 REDIM	(92)
7.3.5	数组和变量类型说明语句 DIM...AS...	(93)
§ 7.4	数组应用程序举例	(94)
习题七		(106)
第八章	字符串数据处理	(110)
§ 8.1	字符串的基本概念	(110)
8.1.1	字符串数据类型	(110)
8.1.2	字符串数据的运算	(112)
8.1.3	字符串检索	(117)
§ 8.2	取子字符串函数	(118)
8.2.1	LEFT\$函数	(119)
8.2.2	RIGHT\$函数	(119)
8.2.3	MID\$函数	(119)
§ 8.3	字符串生成函数	(120)
8.3.1	STRING\$函数	(120)
8.3.2	SPACE\$函数	(120)
§ 8.4	字符串与数值互相转换的功能函数	(121)
8.4.1	ASCII 码与字符转换 ASC 和 CHR\$函数	(121)
8.4.2	数字与字符串转换的 STR 和 VAL 函数	(122)
8.4.3	测试字符串长度的 LEN 函数	(122)
8.4.4	删除字符串首尾空格的 (LTRIM\$和 RTRIM\$) 函数	(123)
8.4.5	大小写字母转换函数 (LCASE\$和 UCASE\$)	(123)
8.4.6	替换字符串中字符的 MID\$语句	(124)
§ 8.5	系统日期和时间的读取和设置	(124)
8.5.1	设置和读取系统日期	(124)
8.5.2	设置和读取系统时间	(125)
§ 8.6	应用举例	(126)
习题八		(131)
第九章	自定义函数, 子程序与过程	(133)
§ 9.1	自定义函数	(133)
9.1.1	自定义函数的概念	(133)
9.1.2	自定义函数的定义与调用	(133)
§ 9.2	子程序	(136)
9.2.1	子程序的定义及返回方式	(136)
9.2.2	子程序的调用	(136)
§ 9.3	过程	(139)
9.3.1	过程的概念	(139)
9.3.2	FUNCTION 过程的定义及其调用	(139)

9.3.3	SUB 子程序的定义及其调用	(141)
9.3.4	参数传递	(144)
9.3.5	过程的说明	(148)
§ 9.4	变量的作用域和共享	(151)
9.4.1	全局变量与局部变量	(151)
9.4.2	共享变量	(152)
9.4.3	静态变量和自动变量	(154)
§ 9.5	过程嵌套和递归调用	(155)
9.5.1	过程嵌套	(155)
9.5.2	过程的递归调用	(156)
§ 9.6	过程应用举例	(157)
§ 9.7	Quick 库	(161)
9.7.1	Quick 库概述	(161)
9.7.2	Quick 的建立	(162)
§ 9.8	程序的链接(CHAIN 语句)	(164)
9.8.1	CHAIN 语句	(164)
习题九		(165)
第十章	屏幕控制和作图	(167)
§ 10.1	屏幕显示的两种方式	(167)
10.1.1	屏幕显示的文本方式	(167)
10.1.2	屏幕显示的图形方式	(168)
§ 10.2	辅助性语句, 函数和变量	(168)
10.2.1	WIDTH 语句	(168)
10.2.2	LOCATE 定位语句	(169)
10.2.3	清屏语句 CLS	(170)
10.2.4	POS 和 CSLIN 函数	(170)
§ 10.3	显示状态控制语句	(170)
10.3.1	SCREEN 语句	(170)
10.3.2	调色语句 COLOR	(171)
§ 10.4	绘图语句	(173)
10.4.1	画点语句 PSET 和擦点语句 PRTESET	(174)
10.4.2	画线语句 LINE	(175)
10.4.3	连续画线语句 DRAW	(176)
10.4.4	画圆、椭圆和画弧语句 CIRCLE	(178)
10.4.5	着色语句 PAINT	(179)
§ 10.5	Quick BASIC 的窗口操作	(181)
10.5.1	窗口概念和定义视见窗语句	(181)
10.5.2	使用 WINDOW 语句重新定义图形视见窗坐标	(183)
§ 10.6	应用举例	(185)
习题十		(187)

第十一章 文件	(188)
§ 11.1 文件的基本概念	(188)
11.1.1 文件的定义及分类	(188)
11.1.2 文件与记录	(188)
11.1.3 文件名	(188)
11.1.4 数据文件的打开与关闭	(189)
11.1.5 文件指针	(189)
§ 11.2 顺序文件	(189)
11.2.1 顺序文件的概念及特点	(189)
11.2.2 建立顺序文件的步骤	(190)
11.2.3 读取顺序文件的步骤	(193)
11.2.4 修改顺序文件	(194)
§ 11.3 随机文件	(196)
11.3.1 随机文件的概念及特点	(196)
11.3.2 建立随机文件的步骤	(197)
11.3.3 读取随机文件的步骤	(199)
§ 11.4 Quick BASIC 对读写随机文件方法的改进	(201)
11.4.1 定义记录类型和记录变量	(201)
11.4.2 随机文件打开语句	(202)
11.4.3 读写随机文件语句	(202)
§ 11.5 应用举例	(203)
习题十一	(209)
第十二章 输入输出设计	(210)
§ 12.1 输入输出技术	(210)
§ 12.2 格式输出	(212)
12.2.1 用 PRINT USING 语句输出数值	(213)
12.2.2 用 PRINT USING 语句输出字符串	(215)
§ 12.3 菜单技术	(216)
习题十二	(218)
第十三章 应用程序综合举例	(219)
§ 13.1 打印万年历	(219)
§ 13.2 快速排序	(221)
§ 13.3 用高斯消元法求解线性方程组	(222)
§ 13.4 陷阱技术应用	(224)
13.4.1 出错陷阱	(224)
13.4.2 事件陷阱	(225)
§ 13.5 模拟技术应用	(226)
13.5.1 确定性模拟	(227)
13.5.2 概率性模拟	(227)
§ 13.6 用辛普生法求定积分	(230)

习题十三	(232)
第十四章 上机实习篇	(234)
§ 1 Quick BASIC 的基本操作	(234)
§ 2 顺序结构源程序编制, 调试与运行	(235)
§ 3 常量, 变量, 数值型标准函数的应用练习	(236)
§ 4 选择结构及其程序设计	(237)
§ 5 循环结构程序编制与调试	(239)
§ 6 一维数组应用程序编制与调试	(240)
§ 7 二维数组应用程序编制与调试	(242)
§ 8 字符串数据处理	(243)
§ 9 子程序和过程	(244)
§ 10 屏幕图形处理程序设计	(248)
§ 11 数据文件的应用	(249)
§ 12 综合应用程序设计	(250)
附录一 ASCII 码表	(252)
附录二 PRINT USING 语句的格式码	(254)
附录三 Quick BASIC 语句和函数一览表	(256)
附录四 出错信息	(262)

第一章 Quick BASIC 概论

§ 1.1 计算机语言简介

计算机是信息处理的有力工具，但目前的计算机尚不能直接理解人类的自然语言，人们要指挥计算机运行，不得不使用特定语言与之打交道。使用计算机处理信息时，通常要对待处理的问题预先编排好明确的工作步骤，并把它们用特定的语言表示出来，即编写程序。这种用计算机系统所能接受的语言编写程序的过程称为程序设计。

1.1.1 面向机器的语言

面向机器的语言包括机器语言和汇编语言。

一、机器语言 (Machine language)

计算机能够直接识别与执行的基本操作，如取数、送数、加、减等，是由计算机硬件在设计制造时所确定的。这些由二进制编码表示的基本操作称为机器指令。每一种计算机在设计时，对于能执行什么样的指令，能执行多少条指令，怎样表示这些指令，都有自己的规定。这些指令的总和称为机器的指令系统。

机器语言就是由二进制代码按照一定的规则组成的指令系统。

机器语言程序的优点是计算机可以直接识别、占用内存少、执行速度快。但也存在两个严重的缺点：其一是机器语言难懂、难记，编程工作十分繁琐、易出错，程序不便修改；其二是机器语言不能移植，没有通用性。由于不同的机型有不同的指令系统，因此用一种计算机器语言编写的程序，不能在另一种计算机上运行。

在计算机发展的初期，只有少数计算机专门技术人员才懂机器语言，这就极大地限制了计算机的推广应用。目前除某些单板机外，大多不直接使用机器语言编写程序了。

二、汇编语言 (Assembly language)

汇编语言是符号化的、面向机器的低级程序设计语言。

汇编语言程序是把机器指令组成的机器语言程序符号化，并与机器语言程序一一对应。

与机器语言相比，用汇编语言编制程序要方便得多，且易于学习、记忆、修改。但用汇编语言编制的程序（称为汇编语言源程序）不能直接由机器执行，必须经过“汇编”（即翻译）转换成机器语言程序（称为机器语言目标程序），再由计算机执行。

汇编语言源程序的运行需经过两大阶段，即汇编阶段和运行阶段，如图 1-1 所示。

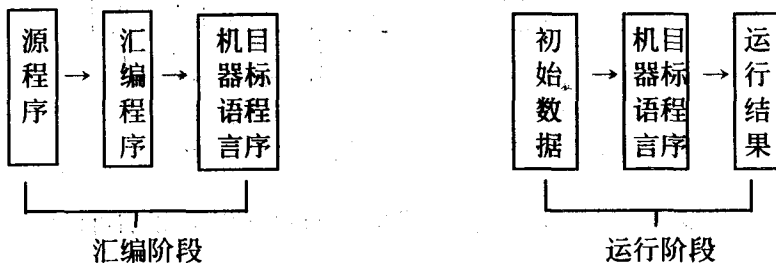


图 1-1 汇编语言源程序的执行

汇编语言源程序经汇编后得到的机器语言目标程序可以保留在计算机系统中，以后可任意运行，而不必每次运行前都汇编。

由于每种计算机都有不同的设计思想和应用目的，那么它就有自己的机器指令系统和相

应的机器语言及汇编语言。要掌握某种计算机的汇编语言，就必须熟悉该种计算机（包括编程时要涉及到的其硬件内部结构和功能）。因此，汇编语言设计者应有相当的专业知识。

基于以上特点，汇编语言适合于编写实时控制程序和系统软件。

1.1.2 高级语言(High level language)

50年代末，60年代初开始，相继出现了多种高级程序设计语言，它独立于机器，且接近于习惯的数学表达式和自然语言。所谓高级，主要是指不必对计算机的指令系统及内部结构有深入的了解就可以编写程序，而且程序通用性好，具有可移植性，即用同一种高级语言编写的程序能在不同型号的计算机上至多稍加修改便可运行。

我们把用高级语言编写的程序称为源程序。和汇编语言程序一样，高级语言源程序必须翻译成机器语言才能执行。高级语言的翻译工作通常有两种方式：一种是编译方式(Compilation)，另一种是解释方式(Interpretation)。

编译方式是把高级语言源程序翻译成机器语言或汇编语言程序，同时具有对源程序检错的功能。编译程序与汇编程序有些类似，但比汇编程序复杂得多。原因是：汇编语言是面向机器的低级语言，汇编语言源程序与目标程序存在著基本的对应关系；而高级语言更接近于自然语言，源程序与目标程序之间是一对多的关系，即一条语句通常可能转换成几条到几十条指令。编译过程要经过先分析后综合等相当复杂的处理过程。有的编译甚至不能一次将高级语言源程序翻译成机器语言程序，而是分两步，先将源程序翻译成汇编语言中间程序，然后再将中间程序翻译成机器语言目标程序。源程序的执行需要分成两大阶段即编译阶段和运行阶段(图1-2)，或三大阶段即编译阶段、汇编阶段和运行阶段(图1-3)完成。

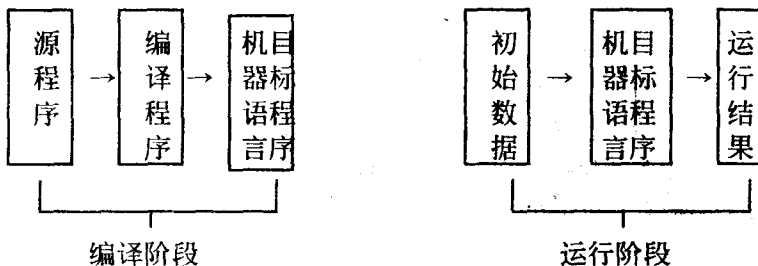


图 1-2 高级语言源程序执行的两阶段过程

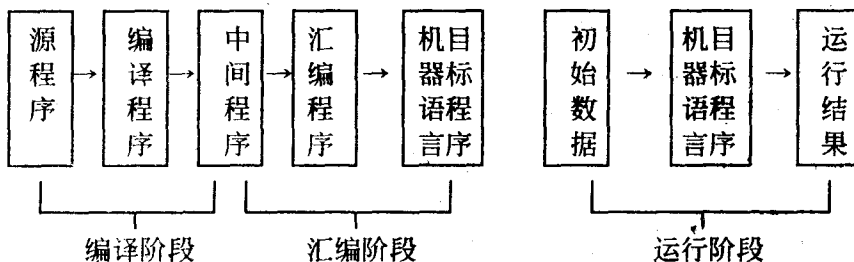


图 1-3 高级语言源程序执行的三阶段过程

经编译、汇编后的机器语言目标程序可以保留在计算机系统中，以后可以任意运行，而不必每次运行前都编译。

采用解释方式是对源程序按语句的动态顺序进行逐句分析，边翻译边执行，直至程序结束。它与编译方式的根本区别在于对源程序不生成目标程序。用解释程序翻译源程序只有一个阶段——解释执行阶段，如图1-4所示。

采用解释方式的高级语言程序易于实现人机对话，便于学习和使用；不足的是占用机器

的时间长, 运行效率低。

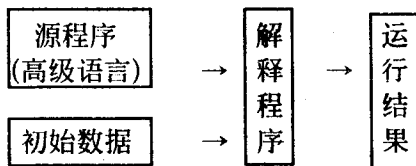


图 1-4 高级语言源程序的解释执行过程

高级语言的出现, 摆脱了枯燥无味的 0、1 代码的束缚, 使计算机的普及与应用结束了二进制的时代, 进入了十进制以及接近于自然语言和数学语言的新天地。高级语言的优点还在于易学易用易修改, 程序通用性强, 即有较好的可移植性。此外, 高级语言适合各类人员根据不同需要编制用户程序, 这就使计算机的应用领域不断扩大。难怪阿兰·图灵说过: “高级语言的产生是计算机发展史上最惊人的成就”。

§ 1.2 BASIC 语言的发展

BASIC 语言是国内外广泛使用、简单易学的一种计算机语言, 在我国计算机应用的起步与发展中起了重要作用。

1.2.1 初期 BASIC (1964 ~ 1971)

1964 年 Dartmouth 学院的 John G. Kemeny 和 Thomas E. Kurtz 等人首创了 BASIC 语言, 它原是为适应分时系统而研制的一种交互式语言, 可用于一般的数值计算与事务处理。由于这种语言采用人机对话方式, 从心理学观点看, 对编程过程较有利, 同时, 因其简单明了, 不久便成为计算机的入门语言。

BASIC 是 Beginner's All-purpose Symbolic Instruction Code (初学者通用符号指令代码) 的缩写。该语言开始时仅包括初学者所需的功能, 随著各大学、计算机厂商和研究所在实际应用中加入了各自所需的内容, 其功能得以扩大。1978 年美国国家标准协会 (ANSI) 制定了 BASIC 语言的最小标准, 或称基本 BASIC。

基本 BASIC 有 17 种语句和 11 种标准函数。因其易学易用被普遍用于各种计算机系统, 而各种计算机系统又按实际的需要扩展了 BASIC 的内容。具有扩展部分的 BASIC 称为扩展 BASIC, 我国的国产 DJS-100 系列计算机都配有该 BASIC 语言。国内从 70 年代末开始普遍以此为内容讲授 BASIC 语言, 并把它广泛用于数值计算、信号分析与处理、概率计算与统计、最优化技术等领域。

1.2.2 中期 BASIC (1971 ~ 1984)

BASIC 的简单性也使得它很小, 当计算机也开始变小的时候语言的大小是至关重要的。麻省理工学院的 Altair 计算机诞生于 1975 年, 它迎来了微机的革命。Microsoft 公司的共同缔造者 Bill Gates 和 Paul Allen 承担了为 Altair 开发一个 BASIC 语言系统的任务, 该 BASIC 版本在计算机仅有 4K 内存中运行, 最后它发展成为了个人计算机工业中最流行的产品。

进入 80 年代后, 当 IBM PC 替代了早期的微机时, Microsoft 公司的 GW-BASIC 和与之完全兼容的高级 BASIC (即 BASIC) 成为了标准。它们包含基本 BASIC 和扩展 BASIC 的全部功能, 增加了绘图、著色、消隐等语句和图形变换与窗口操作, 使计算机屏幕作图功能得以增强。此外, 还提供了事件陷阱、读取键盘按键、通讯及光笔等技术, 方便交互式输入程序及作图显示; 结合音乐语句, 还广泛用于游戏程序。

BASIC 因采用解释方式, 程序执行速度慢是最大缺点。为克服此缺点, 吸取编译方式的高

级语言优点, 后又产生了编译 BASIC。它具备 BASIC 的大部分功能, 使用编译方式, 程序执行速度比解释 BASIC 加快 3 ~ 10 倍, 并有节省内存空间和可提供保密措施等优点。编译 BASIC 与解释 BASIC 在一些国家同被列入教科书中。

1.2.3 近期 BASIC (1985 ~ 1990)

BASIC 在一段时期也被人们冷落过, 因为 BASIC 的初衷是面向初学者的, 所以一问世就带有不少与生俱来的长处和短处: 易学易用, 透明度高, 但“解释一句执行一句”的速度毕竟不尽人意。此外, 它还违背时下流行的“结构化程序设计”思想, 即软件质量以“结构清晰及可读性好”为首要条件。传统的 BASIC 在研制一些较大软件系统时, 便潜伏著“软件危机”, 因而, 被一些人称为“street BASIC”(街头 BASIC), 认为不能登大雅之堂。

到了 1985 年, BASIC 语言的原创始人在美国国家 BASIC 标准(1984)的基础上, 吸收各种高级语言的优点, 研制推出了取名为“True BASIC”的新版本。它保留了简单易学的特点, 完全支持结构化程序设计的概念, 增加或加强了绘图、窗口、矩阵运算等功能, 给大众化的 BASIC 注入了新的活力。1987 年, Borland 公司推出了 Turbo BASIC, 它也是一种结构化程序设计语言, 标准度高, 增加了流程控制语句、多行函数和过程、标号代替行号作转移语句的目标, 且它是编译型语言, 运行速度快, 调试中将程序的编辑、编译、运行和跟踪等工作融为一体, 全部命令及环境参数设置以菜单方式提供, 可四个窗口显示, 人机界面焕然一新, 因而在适应各种较大应用课题方面又前进一步。

1987 年 11 月, Microsoft 公司推出了 Quick BASIC。它吸取了 FORTRAN、PASCAL、C 及 Ada 等语言的优点, 克服了传统 BASIC 结构化程序差和运算速度慢的缺点。从难度上说, 它与传统 BASIC 差不多, 而功能则大为扩充, 其完善程度可以与 PASCAL、FORTRAN 及 C 相媲美。

据 IDG 统计, 在美国, 1981 年以前 47.3% 的人使用 BASIC, 到了 1987 年使用 BASIC 的人骤然上升到 80 ~ 90%。对此, 只有一种解释, 那就是 BASIC 具有非凡的吸引力, 它已成为“效率最高的语言”。

1.2.4 最先进的 BASIC (1991 ~ 今)

1991 年 Microsoft 公司开发出 Visual BASIC, 它是采用事件驱动编程机制的计算机语言。事件驱动是一种非常适合图形用户界面的编程方式。传统的编程是一种面向过程按顺序进行的工作, 即需要描述“怎样做”, 很象烹饪菜肴的菜谱: 打鸡蛋、加牛奶、加糖搅拌、烤 20 分钟。这种编程方式的缺点是写程序的人总是要关心什么时候要发生什么事情。这对于烹饪来说是可以接受的, 但在现代的计算机应用中, 目的是让用户来操纵程序的运行。Visual BASIC 语言作为一种事件驱动编程语言, 程序员只要编写响应用户动作的程序, 即只需说明“做什么”, 如选择命令、在窗口中撤击鼠标按钮、移动鼠标等, 而不必编写按精确次序执行的每个步骤。你不必编写一个大型程序, 而是创建一个由若干个小程序组成的应用程序, 这些小程序都由用户启动的事件来激发。利用 Visual BASIC, 程序员能以空前的速度, 方便地编写各类应用程序。

Visual BASIC 的出现, 使从前掌握过 BASIC 语言的用户对其趋之若鹜, 因熟悉它比熟悉一种新的语言要快得多。相信学过 BASIC 的用户和正在学 BASIC 的用户, 以及即将学习 BASIC 的用户, 都将比掌握其它高级语言的用户更快地学会 Visual BASIC。BASIC 已不再是人们称道的“丑小鸭”, 而是名符其实的“白天鹅”。

§ 1.3 Quick BASIC 特点

Quick BASIC 把解释型 BASIC 的方便性和编译型 BASIC 的执行速度快有机地结合在一起,既可以同时完成编辑、调试和运行,又提高了执行速度,并对 BASIC 语言的功能进行了较大的扩充。此外,所有操作都在内存中完成,而不是对盘上的文件进行读写,从而使程序调试运行更快、更容易。Quick BASIC 与 BASICA 基本兼容,大多数用 BASICA 开发的程序不用作任何修改即可在 Quick BASIC 环境下运行或编辑、连接。可以说 Quick BASIC 集它之前的所有 BASIC 版本优点的总和,更适合教学和程序开发。

Quick BASIC 的特点主要表现在两个方面,一是用户所期望的编程环境,二是对语言功能所作的扩充。

1.3.1 Quick BASIC 在编程环境方面的特点

1. 交互式编辑

Quick BASIC 环境的核心部分是一个性能优越的编辑程序。输入一个程序行后,马上就能对其进行语法检查,如果没有语法错误,则该程序行即被翻译成可执行代码,否则给出错误描述信息。该编辑程序还能自动将关键字转换为大写,校正遗漏和错误,并可显示程序的逻辑结构。

2. 交互式调试

Quick BASIC 把编辑、运行和调试有机地结合起来,不必来回转换。在调试程序时,可以将一个运行著的程序在任何一处暂停,对它进行编辑,然后从停止处继续执行。用户可利用屏幕底部的“直接”窗口测试程序的语句,甚至可以修改在运行著的程序中的数据,以便观察修改后的效果,而不必重新启动程序。

3. 快速生成可执行文件

程序调试通过后,如果需要,则只要进行很简单的操作,就能生成在 DOS 状态下可直接运行的文件(.EXE 文件)。

4. 多个模块同驻内存

Quick BASIC 4.0 以上的版本支持多个模块同时驻留内存,如要将内存中的全部模块编译、连接成一个可执行文件,则只用一步操作即可实现。

5. 库的自动生成与管理

Quick 库可将用户编写的过程和例行程序变为 BASIC 语言的虚拟扩充部分。只要在 RUN 菜单中选取一条命令,即可构造一个 Quick 库,而且每当建立或修改一个 Quick 库时,Quick BASIC 将自动地建立或更新与之相应的独立库。

6. 灵活的联机求助功能

Quick BASIC 提供了灵活的求助功能,它既可为每个语句提供语法检查,同时也为菜单命令、快捷键和 ASCII 提供帮助信息。

7. 完全图形支持

Quick BASIC 支持 VAG、MCGA、CGA、EGA 及 PS/2 的图形方式和单色方式,同时支持增强图形卡。

1.3.2 Quick BASIC 在语言功能上的扩展

1. Quick BASIC 把微机内存看作是一张大的空白程序纸,在这张纸上的每一行可含有一个或多个 BASIC 语句(各语句间用“:”隔开),程序按物理行的顺序执行,因而源程序可不写行号。原 BASIC 中的行号在 Quick BASIC 中作为标号来处理。标号既可用数值表示,也可用字符串表示。