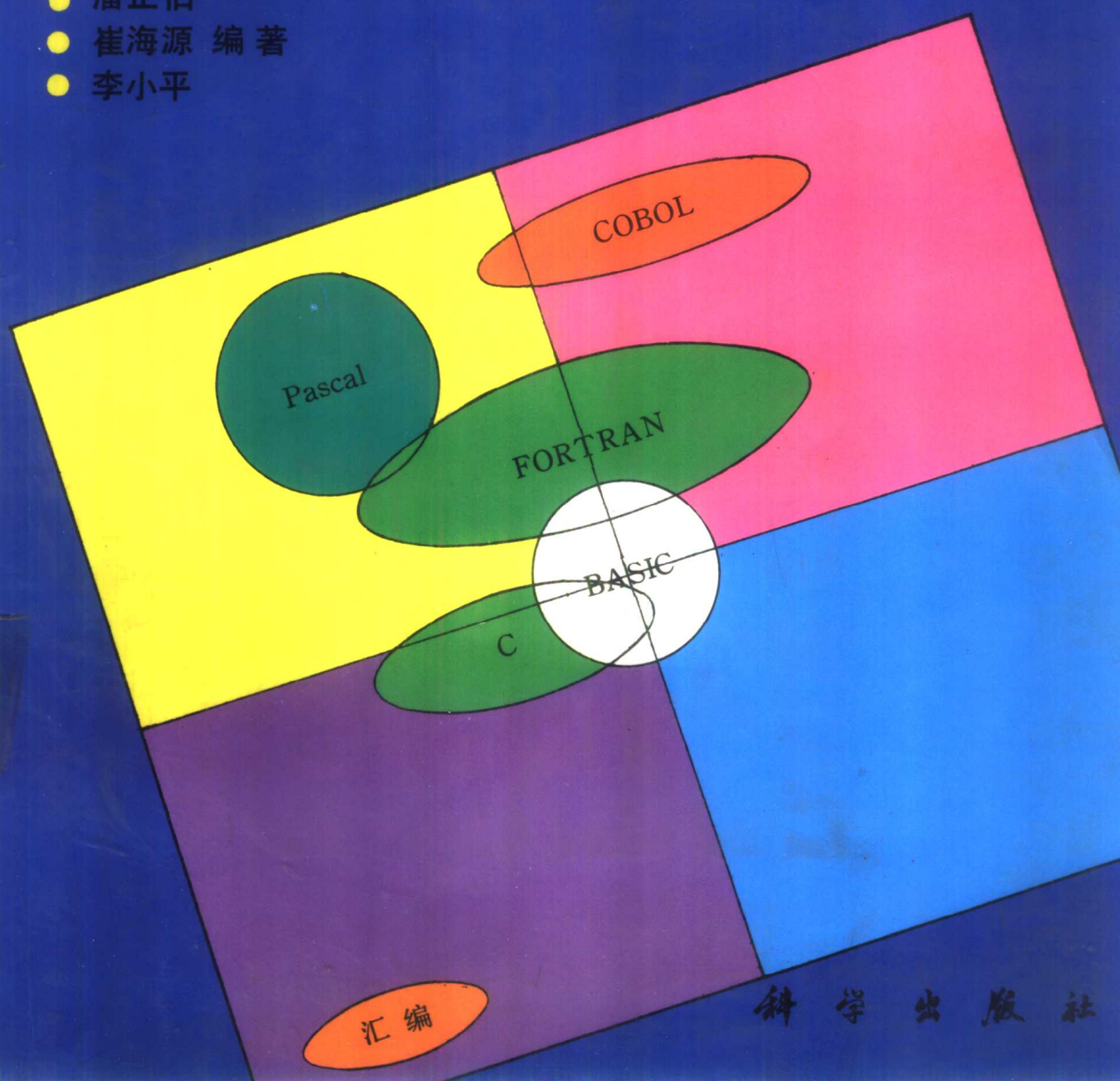


Quick BASIC

结构化程序设计

教程

- 潘正伯
- 崔海源 编著
- 李小平



科学出版社

Quick BASIC

结构化程序设计教程

潘正伯 崔海源 李小平 编著

科学出版社

1996

(京)新登字 092 号

内 容 简 介

第二代 BASIC(结构化 BASIC)的三种代表产品——True BASIC, Turbo BASIC 和 Quick BASIC 经过几年的激烈竞争,前两种已销声匿迹,唯独 Quick BASIC 仍保持着旺盛的势头。这当然与美国微软公司创办人、董事长兼总经理 Bill Gates 对它的偏爱和重点投入有关。他使得 Quick BASIC 成为第二代 BASIC 的集大成者,又使得 Quick BASIC 4.5 成为能直接进入第三代 BASIC(Windows BASIC)的唯一桥梁。

掌握 QB 4.5,是我国大专院校计算机专业和非计算机专业学生以及广大计算机用户为用好计算机、搞好本专业的应用开发并尽快适应 Windows 和 OS/2 下的开发环境最有效的途径。

本书集编者四年来的教学实践经验写成,书中对程序设计初学者常感困惑的问题(如变量的生存域、过程间的参数传递、递归等)作了通俗易懂的讲述。书中有大量经过反复推敲的例题和习题,便于读者摹仿和练习。

本书在介绍 QB 4.5 语法规则的同时,也纳入了指导编程工作必不可少的软件工程知识,使读者学习本书后,能写出质量更高的程序。

本书按教材体例编写,在附录中收入了一切必要的上机资料(如灵巧编辑器用法和出错信息等),且有完整的词条索引,读者可很方便地将本书当作 QB 4.5 的手册使用。

Quick BASIC 结构化程序设计教程

潘正伯 崔海源 李小平 编著

责任编辑 那莉莉

科学出版社出版

北京东黄城根北街19号

邮政编码:100717

双青印刷厂印刷

新华书店北京发行所发行 各地新华书店经售

*

1996年2月第一版 开本:787×1092 1/16

1996年5月第二次印刷 印张:24 1/4

印数:3001—7500 字数:551 000

ISBN 7-03-004980-2/TP·497

定价:30.00 元

前 言

软件界的骄子、美国微软公司的缔造者、现任董事长兼总经理 Bill Gates 六年前在一篇文章中写过这样两段话：

我至今仍是 BASIC 的狂热爱好者。它能处理其他语言迫使你不得不去应付的许多工作，从而解放出你的编程创造力。我已经向程序员们挑战，为解决任一问题他们可以选用任何工具编程，我打赌，我用 Quick BASIC 会更快地编出同样的程序。

70 年代后，正当 C 语言风行一时之际，有人预报了 COBOL 和 FORTRAN 语言的衰亡，有的则说 Pascal 敲响了 BASIC 的丧钟。然而，所有这些语言至今仍在使用。我相信，就 BASIC 而论，这种语言正在欣欣向荣地成长。一旦 BASIC 变成一种通用的宏语言，依我看它正朝此方向发展，那么它将比其他任何过程语言都长寿。实际上，BASIC 可能比我们这些人的寿命还长。

计算机技术发展很快，用“日新月异”形容它已不含任何夸张成分。近期人们已能目睹：

(1) 操作系统从 16 位向 32 位过渡，开发环境从 DOS 向 Windows, Windows 95, Windows NT 或 OS/2 转换。

(2) Pentium 芯片迅速占领市场，P6 芯片也将随之普及。

(3) 微机纷纷以多种形式联网，INTERNET 发展迅猛，Chinanet 工程全面启动。

(4) 多媒体技术迅速向各个领域渗透。

(5) DBMS, CAD, CAI, CAM, 人工智能，计算机模拟，模糊逻辑……得到广泛应用。

六年，在计算机发展史上不能说是一个短时期。Bill Gates 说的那些话还有效吗？

去年《BYTE》杂志上的两则广告给了我们答案：一家公司公然以 Power BASIC 作自己的公司名——可见它对 BASIC 寄托之深。它在介绍自己的产品 Power BASIC 的广告词中说：“它几乎 100% 与 Quick BASIC 兼容”。世界第二大软件公司 (Computer Associates International) 在介绍自己的新版 BASIC (CA-REALIZER 2.0) 时也强调“谁也不如它那样方便地移植你用 Quick BASIC 开发的成果”。在历数 CA-REALIZER 能给用户带来的 12 项收益中，再次强调“易于将 Quick BASIC 应用成果移植到 Windows 和 OS/2 下”。

商品生产高度发达的国家对广告是高度重视的，任何厂商绝不会把无关紧要的浮词滥语写入自己产品的广告词中。上述两段广告词至少向我们提供了以下三点信息：

(1) 进入 90 年代以后，BASIC 家族仍在迅速繁衍，新成员在继续增加，品质在继续提高，这反映了市场的需求，而市场是测试用户需求的最广泛、最灵敏的温度计。

(2) 在强烈追求技术进步的美国，到 1994 年下半年为止，尚存在着一个相当庞大的、举足轻重的 Quick BASIC 用户群，以至迫使任何厂家在开发自己的新产品时不得不考虑它与 Quick BASIC 的兼容性，而且在自己的广告中突出地强调这一兼容性以争取更多的用户。

(3) 用 Quick BASIC 开发的成果,很容易实现向 Windows 或 OS/2 环境下转换。

最近又从 Roger Jennings 写的《Access™2 开发指南(第二版)》(SAMS, 1994)上看到这样一段话:

Bill Gates 是 Microsoft 的 Chairman 和 CEO,他曾规定凡是需要一个嵌入式程序设计语言的 Microsoft 应用程序都要使用 Microsoft 的 BASIC 形式。

更坚定了我们把 Quick BASIC 4.5 介绍给中国计算机用户的决心。

自《BASIC 教程下卷 Quick BASIC》出版以来,我们已将它纳入自己的教学实践中,用了三年。对它的优点、缺点、难点、易点深有感受。在教学实践中对例题、习题和全书的结构、顺序都作了很多修改、补充和优化。

我们认识到,在我国的具体条件下,Quick BASIC 肩负着承上启下的历史使命,需要对它大推广、大普及。它能帮助我国大中专学生、广大的计算机用户尽快接触计算机的最新技术。因此我们重新编写了该书,力求在该书中溶入我们三年来在 Quick BASIC 教学中的点滴心得。

这是一本教科书又是一本手册,它的“索引”对读者是很有用的。

真正掌握一种程序设计语言,一方面是关于这种语言的知识的获得,另一方面是关于程序设计技能的训练和作为一名软件开发人员的良好素质的养成。过去常常是重视了前者而忽视了后者。后者是很难从分章习题中获得的。

建议在学完本书的基本内容后,可将学生几人一组,分组完成一个中等规模的课程设计。实践证明,这种课程设计深得学生欢迎,对提高学生的编程能力很有好处。

山东矿业学院应用教学与软件工程系 95,96,97 届数十位同学对《BASIC 教程下卷——Quick BASIC》写出了书面评论,他们的意见是形成本书的重要参考。他们中的薛庆军、韩莹洁、于志伟和张彦东,在潘正伯指导下参加了本书的编撰工作。在编撰过程中他们作出了贡献,也得到了提高和锻炼。

我们诚挚地感谢陈贺平先生以及在美国的杜殿海先生、曾瑞华先生和陈湧先生,感谢他们不断地提供最新的资料和书籍,使我们能把这本书写得更充实些。

虽然我们竭尽全力,对书稿作了反复修改,但终因水平所限,不能尽如人意,甚至有可能潜藏着某些我们尚未发现的错误,恳请读者指正,不胜感谢!

编者识

1995.10.28

目 录

第一章 基本知识	1
1.1 计算机和计算机语言	1
1.1.1 计算机发展概况	1
1.1.2 计算机语言	1
1.1.3 BASIC 百花园中的 Quick BASIC	3
1.2 流程图	4
1.2.1 流程图的作用	4
1.2.2 流程图的符号	5
1.2.3 流程图举例	6
1.3 语言元素	7
1.3.1 字符集	7
1.3.2 标识符	7
1.4 数据类型及其说明	8
1.4.1 数	8
1.4.2 串	9
1.4.3 量	9
1.4.4 类型说明	11
1.5 运算符和表达式	12
1.5.1 数字表达式	13
1.5.2 字符串表达式	13
1.5.3 逻辑表达式	13
1.6 赋值语句	16
1.6.1 赋值语句(LET 语句)	16
1.6.2 数据类型转换	17
1.7 内部函数	18
1.7.1 数值函数	19
1.7.2 数值转换函数	21
1.7.3 日期和时间函数	23
1.8 Quick BASIC 程序的组成	24
1.8.1 最简单的 BASIC 程序	24
1.8.2 复杂程序的结构	27
1.8.3 程序的书写	28
习题	29
第二章 标准输入/输出	31

2.1 数据的输入	31
2.1.1 键盘输入语句(INPUT 语句)	31
2.1.2 行输入语句(LINE INPUT 语句)	32
2.1.3 读数置数语句(READ/DATA 语句)	33
2.1.4 恢复读数据语句(RESTORE 语句)	35
2.2 程序的输入	35
2.2.1 Quick BASIC 屏幕初识	35
2.2.2 Quick BASIC 的灵巧编辑器	38
2.2.3 程序和数据的键入	40
2.2.4 文件	42
2.2.5 文档	43
2.3 运行结果的输出	45
2.3.1 屏幕显示语句(PRINT 语句)	45
2.3.2 格式显示函数	46
2.3.3 格式显示语句(PRINT USING)	48
2.3.4 打印输出语句(LPRINT/LPRINT USING)	50
2.4 程序的存储和运行	50
2.4.1 程序的存储	50
2.4.2 程序的运行	51
习题	51
第三章 顺序结构程序设计	53
3.1 几种 Quick BASIC 中常用的语句	53
3.1.1 初始化语句	53
3.1.2 REM 语句	55
3.1.3 END 语句	56
3.1.4 SWAP 语句	56
3.1.5 COLOR 语句	57
3.2 顺序结构程序设计	57
习题	60
第四章 字符串	61
4.1 字符和字符串	61
4.1.1 字符	61
4.1.2 字符串	61
4.2 输入语句中的串	63
4.2.1 赋值语句中的串	63
4.2.2 READ/DATA 语句中的串	64
4.2.3 在键盘输入语句中的串	65
4.3 串的比较	66
4.4 串函数	67

4.4.1 测串长函数(LEN 函数)	67
4.4.2 取字符函数	68
4.4.3 删左、右空格函数(LTRIM \$ 函数和 RTRIM \$ 函数)	68
4.4.4 串检索函数(INSTR 函数)	69
4.4.5 串的生成函数	70
4.4.6 改变串	70
4.4.7 字符串与数值之间的转换函数	71
4.4.8 ASCII 码与字符之间的转换函数	72
4.5 数制与数制转换	73
4.5.1 数制	73
4.5.2 数制转换	73
4.5.3 HEX \$ 函数与 OCT \$ 函数	75
4.6 综合示例	75
习题	77
第五章 分支结构程序设计	79
5.1 无条件转移语句(GOTO 语句)	79
5.2 条件语句	79
5.2.1 单行 IF~THEN~ELSE 语句	79
5.2.2 块 IF~THEN~ELSE 语句的最简形式	81
5.3 多路分支	82
5.3.1 块 IF~THEN~ELSE 语句的扩展形式	82
5.3.2 SELECT CASE 语句	84
5.3.3 开关语句(ON~GOTO 语句)	85
5.4 分支结构的嵌套	87
5.4.1 行、块 IF~THEN 语句的嵌套	87
5.4.2 混合嵌套	88
5.5 综合示例	89
习题	92
第六章 循环结构程序设计	94
6.1 FOR~NEXT 循环语句	94
6.1.1 FOR~NEXT 语句的一般形式	94
6.1.2 FOR~NEXT 语句的执行过程	95
6.2 WHILE~WEND 循环语句	96
6.3 DO~LOOP 循环语句	96
6.3.1 DO~LOOP 循环语句	96
6.3.2 DO~LOOP 循环中的 WHILE 修饰词	97
6.3.3 DO~LOOP 循环中的 UNTIL 修饰词	98
6.4 退出语句(EXIT 语句)	99
6.5 循环结构的框图表示法	100

6.6 多重循环	101
6.7 综合示例	104
习题	110
第七章 数组	112
7.1 数组的概念	112
7.2 数组的定义	112
7.2.1 DIM 语句	112
7.2.2 一维数组、多维数组与字符串数组	114
7.2.3 数组操作的其他语句和函数	116
7.3 静态数组与动态数组	117
7.3.1 元命令 \$DYNAMIC 和 \$STATIC	117
7.3.2 静态数组和动态数组	117
7.3.3 数组的释放(ERASE)与重定义(REDIM)	118
7.4 综合示例	119
习题	126
第八章 子程序和自定义函数	128
8.1 子程序	128
8.1.1 子程序的建立(RETURN 语句)	128
8.1.2 子程序的调用(GOSUB 语句)	129
8.1.3 子程序的流程图表示法	130
8.2 开关语句(ON~GOSUB 语句)	130
8.3 自定义函数	132
8.3.1 建立自定义函数(DEF FN 语句)	132
8.3.2 自定义函数的调用	133
8.4 综合示例	133
习题	135
第九章 过程	137
9.1 变量的生存域	137
9.2 说明类语句和属性	138
9.2.1 DECLARE 语句(BASIC 过程)	138
9.2.2 SHARED 语句和 SHARED 属性	138
9.2.3 CONST 语句	139
9.2.4 STATIC 语句与 STATIC 属性	140
9.3 SUB 过程	140
9.3.1 SUB 语句	140
9.3.2 SUB 过程的建立	141
9.3.3 SUB 过程的调用(CALL 语句——BASIC 过程)	142
9.4 FUNCTION 过程	143

9.4.1 FUNCTION 语句	143
9.4.2 FUNCTION 过程的建立	143
9.4.3 FUNCTION 过程的调用	144
9.5 参数传递	145
9.5.1 通过参数表传递参数	145
9.5.2 利用数据生存域的定义传递参数	145
9.5.3 各种数据的传递	147
9.5.4 传址和传值	150
9.6 递归	151
9.6.1 递归调用	151
9.6.2 FRE()函数	152
9.6.3 调整堆栈的容量	152
9.7 多模块调用与程序的连接	153
9.7.1 COMMON 语句	153
9.7.2 MAK 文件的建立	155
9.7.3 CHAIN 语句	156
9.8 综合示例	157
习题	159
第十章 文件	161
10.1 文件分类	161
10.1.1 文件说明	161
10.1.2 文件分类	161
10.1.3 记录和记录的定义(TYPE~END TYPE 语句)	163
10.1.4 文件号	164
10.2 与文件处理有关的语句和函数	164
10.2.1 打开数据文件或设备(OPEN 语句)	164
10.2.2 关闭数据文件或设备(CLOSE 语句)	166
10.2.3 文件操作语句和函数	167
10.3 顺序文件	170
10.3.1 顺序文件的输出(建立和添加数据)	170
10.3.2 顺序文件的输入	172
10.3.3 顺序文件的使用和维护	173
10.4 随机文件	178
10.4.1 随机文件的输出(建立和添加数据)	179
10.4.2 随机文件的输入	182
10.4.3 随机文件的维护	184
10.4.4 在过程间传递文件中的数据	184
10.5 包含文件	186
10.5.1 包含文件的用途	186

10.5.2 包含文件的建立	186
10.5.3 包含文件的引用	187
10.5.4 包含文件的观察和修改	187
10.6 索引文件	187
10.6.1 索引文件的基本概念	187
10.6.2 索引文件的建立	188
10.6.3 索引文件的应用	190
10.7 二进制文件	191
10.7.1 二进制存取的特点	191
10.7.2 SEEK 语句与 SEEK 函数	192
10.8 综合示例	194
习题	201
第十一章 图形和动画	203
11.1 屏幕坐标系	203
11.1.1 像素	203
11.1.2 屏幕坐标系	203
11.2 屏幕的设置	204
11.2.1 SCREEN 语句和 SCREEN 函数	204
11.2.2 PCOPY 语句	205
11.2.3 WIDTH,CLS,LOCATE 语句	206
11.2.4 设置颜色	208
11.3 基本绘图语句	211
11.3.1 画点(PSET,PRESET 语句)	211
11.3.2 画直线和矩形(LINE 语句)	213
11.3.3 画圆和画弧语句(CIRCLE 语句)	215
11.3.4 DRAW 语句	216
11.4 填图语句(PAINT 语句)	219
11.4.1 填充颜色	219
11.4.2 填充图案	221
11.5 视见窗口和逻辑坐标的设定	224
11.5.1 定义视见区(VIEW 语句)	224
11.5.2 定义文本窗口(VIEW PRINT 语句)	225
11.5.3 设定逻辑坐标系(WINDOW 语句)	226
11.5.4 屏幕颜色的测试函数(POINT 函数)	229
11.5.5 物理坐标与逻辑坐标的转换(PMAP 函数)	230
11.6 图形变换	232
11.6.1 平面图形的几何变换	232
11.6.2 平面图形变换的其他方法	233
11.7 动画设计	235

11.7.1 GET 语句	235
11.7.2 PUT 语句	236
11.7.3 动画	237
11.8 图形文件	239
11.8.1 建图形文件有关的语句	240
11.8.2 图形文件的建立(BSAVE 语句)	241
11.8.3 图形文件的使用(BLOAD 语句)	242
11.9 综合示例	243
习题	248
第十二章 音响和通信	250
12.1 音响	250
12.1.1 BEEP 语句	250
12.1.2 SOUND 语句	250
12.1.3 PLAY 语句	251
12.2 设备通信	252
12.2.1 OUT 语句与 INP 函数	252
12.2.2 访问设备	253
12.2.3 串行通信	254
12.2.4 取 DOS 代入参数的 COMMAND \$ 函数	258
12.3 访问内存	259
12.3.1 POKE 语句	259
12.3.2 PEEK 函数	259
12.4 定义功能键	260
12.5 综合示例	261
习题	263
第十三章 程序的调试和编译	264
13.1 预防程序故障	264
13.2 程序调试	265
13.2.1 供调试和编译的例子	265
13.2.2 调试与编程相结合	265
13.2.3 监视	266
13.2.4 中断	268
13.2.5 跟踪	268
13.2.6 程序运行控制命令	270
13.3 程序在 Quick BASIC 环境内编译和连接	271
13.4 分离编译和连接	272
13.4.1 编译和连接的过程	273
13.4.2 BC 编译命令的使用	273
13.4.3 LINK 连接命令的使用	274

习题	277
第十四章 错误和事件捕获	278
14.1 错误捕获	278
14.1.1 设置错误陷阱	278
14.1.2 错误的模拟与识别	280
14.1.3 错误的处理及返回	281
14.2 事件捕获	283
14.2.1 设置事件陷阱	284
14.2.2 事件处理	285
14.3 键事件陷阱	286
14.3.1 捕获键事件的语句	286
14.3.2 捕获键事件的实施	287
14.4 音乐事件陷阱	288
14.5 通讯口中断陷阱	290
14.6 过程及模块中的陷阱	291
14.7 在应用 BC 编译的程序中捕获错误和事件	293
习题	293
第十五章 库及 Quick BASIC 与其他语言连接	295
15.1 库的概念与使用	295
15.1.1 库的类型	295
15.1.2 库的优点	296
15.1.3 QLB 库的使用	296
15.1.4 建库所需的文件	298
15.2 在 Quick BASIC 环境下建库	298
15.2.1 在 Quick BASIC 环境内建库	298
15.2.2 对原有库的扩充	301
15.2.3 截取软件包中的过程进库	301
15.3 Quick BASIC 与其他语言的数据交换	304
15.3.1 通过数据文件交换数据	304
15.3.2 通过读取二进制文件交换数据	305
15.3.3 通过内存单元交换数据	308
15.4 Quick BASIC 调用其他语言的过程	309
15.4.1 DECLARE 语句(非 BASIC 过程)	309
15.4.2 在 DOS 环境下建库	310
15.4.3 对 Fortran 语言的过程建库	311
15.4.4 对 C 语言的过程建库	312
15.4.5 对汇编语言的程序建库	314
15.4.6 对混合语言的目标文件建库	315
15.5 混合语言编程实例	317

15.5.1 Quick BASIC 与 C 语言调用的约定	317
15.5.2 Quick BASIC 与 C 语言混合编程的数据传递	319
15.5.3 Quick BASIC 与 C 语言混合编程的编译和连接	323
15.5.4 Quick BASIC 与汇编语言的调用约定	324
15.5.5 Quick BASIC 与汇编语言的数据传递	327
习题	332
附录 A Quick BASIC 编程环境	333
附录 B ASCII 字符码和键盘扫描码	344
附录 C 基本适配器的屏幕方式	346
附录 D Quick BASIC 保留字	347
附录 E Quick BASIC 错误信息	349
参考文献	366
索引	367

第一章 基本知识

1.1 计算机和计算机语言

1.1.1 计算机发展概况

人类社会发展到今天,已经步入了以电子计算机技术为核心的新技术革命时代。它要求人们必须熟练地掌握多种工具(如外语、各种测试工具、作图和摄影工具、通信和交通工具、打字和文印工具……)。在多种工具中,电子计算机是非常重要的一种。

1946年,世界上第一台电子计算机在美国诞生。它使用了18800个电子管和1500多个继电器,重达30吨,占地150平方米,耗电150千瓦,每秒完成5000次加法运算。尽管这台庞然大物功能不够完善而且耗资巨大,但它仍是科学技术发展史上的一座丰碑。

这台计算机诞生后四十多年来,电子计算机以令人瞩目的速度迅猛发展着,先后经历了四代的发展变化。

第一代,1946~1958年间的电子管计算机。这代计算机体积大、成本高、运算速度一般只有几千到几万次。

第二代,1958~1965年间的晶体管计算机。不仅大大降低了成本和体积,而且将运算速度提高了近百倍。

第三代,1965~1970年间的集成电路计算机。由于采用了集成电路,这代计算机的体积和造价进一步降低,运算速度已提高到了每秒几十万次到几百万次,可靠性也有显著提高。

第四代,1970年以后出现的采用大规模集成电路的计算机。它的可靠性和运算速度更加提高,体积更加缩小,成本也进一步降低,运算速度可达每秒几千万次。

近年来正在研制第五代电子计算机。它将采用超大规模集成电路和约瑟夫逊等新型器件,是一种向着智能化发展的全新的智能计算机。

1.1.2 计算机语言

电子计算机作为一种高效、高精度、具有记忆及判断等多种能力的工具,已经广泛应用于科学计算、数据处理、自动控制、人工智能等许多领域。要学会使用计算机,最大限度地发挥计算机的功能就必须会用计算机语言为它正确地下达命令。在正确命令的操纵下,计算机才能不折不扣地完成主人交给它的任务,实现各种功能。因此,通晓计算机语言是计算机用户不可缺少的知识和技能。

计算机语言经历了由机器语言→汇编语言→高级语言的发展变化过程。

最初的机器语言是和计算机的硬件设计同时产生的。机器语言程序表现为一组由二进制代码(0和1)组成的指令和数据。

在实用中,机器语言有很多缺点:

(1) 程序难于看懂和调试。

(2) 必须将每一位代码逐个送入计算机,输入速度极慢。

(3) 程序太长,编程很费劲。

(4) 稍不小心就会出错,且错误难于查找和排除。

机器语言烦琐、单调,给使用和记忆都带来很多困难。所以,人们在实践中对它进行了改进,赋予每条指令一个名称,这种指令码的名称叫“助记符”。采用这种助记符的语言就是汇编语言。

汇编语言仍存在一些明显的缺点:

(1) 像机器语言一样,汇编语言的指令系统完成的都是最基本的操作(如累加器内容移位、二寄存器内容相加等)。而将用户的复杂任务(如检查模拟读数是否超过终限值,在适当时间启动继电器等)用这些最基本的操作命令表达出来,无疑,工作是繁重而费时的。

(2) 汇编语言要求用户深谙所使用计算机的硬件性能、寻址方式等许多知识,对编程人员的要求较高。

(3) 汇编语言对计算机硬件的依赖性很大,所以不能移植。

50年代末期,出现了高级语言。高级语言是面向过程的语言,它使用户可以用面向问题的方式而不是面向计算机的形式描述任务。高级语言的每一条语句完成一个可识别的功能,通常与多条汇编语言指令相对应。源程序要经过编译程序编译成机器语言指令后方可被计算机识别和运行。

当今流行的高级语言很多,如 Fortran, Pascal, ALGOL, COBOL, APL, BASIC 等。它们都有自己擅长的工作,如 Fortran 长于数学运算,COBOL 常用于处理商业任务,APL 和 BASIC 多用于分时系统和综合任务等。

一般说来,高级语言具有下列优点:

(1) 可以容易、迅速地写出程序。通常用高级语言写程序要比用汇编语言快 10 倍以上。

(2) 高级语言都有自己的语法。上述高级语言的语法,已由国家或国际标准确定,从而保证了其应用范围的广泛。

(3) 高级语言不依附于任何特定的硬件结构——中央处理器(CPU)的型号,这就使得编程人员无需深通所用计算机的硬件性能,减轻了工作的复杂程度。

(4) 高级语言编写的程序,一般是可以移植的。

自然,高级语言也有它的缺点:

(1) 高级语言必须配备相应的编译程序才能使用。有些编译程序的价格很贵,而且要占用大量的内存空间,使用户可用内存大大减少。

(2) 高级语言不直接面向机器,编译程序能进行的优化也并不全面,因此很难在程序执行效率和节省空间两方面实现代码最佳化,在一定程度上影响了程序的效率。

在众多的高级语言中,使用的范围和受欢迎的程度也不尽相同。1981年美国一家信息处理公司(IDG)对 995 位管理信息系统经理作过一次关于所用程序设计语言的调查。43.2%的调查对象认为 COBOL 是最重要的语言,并有 37.8%的人乐于用它;Fortran 的重要性占第二位,乐于选用者则占第四位;PL/I 因出世较晚,在重要性方面占第五位,乐于选用方面仅次于 COBOL 而占第二位。在上述调查之后的六年,即 1987 年,美国波士顿计算机学会对其成员作了类似的调查,结果如下:

BASIC	80~90 %	LISP	10 %
Fortran	60 %	COBOL	5 %
Pascal	35 %	PROLOG	5 %
C	35 %	其它	2 %
汇编	30 %		

当然,这类调查因其对象的业务差异会有不同的结果,但是 BASIC 语言跃居榜首,却绝不是偶然的。

1960 年美国达特姆斯学院 (Dartmouth College) 的 John G. Kemeny 和 Thomas E. Kurtz 教授,发表了一种叫做 BASIC 的高级语言。在一个长时期内,人们都认为 BASIC 不过是在它之前已经发表并广为流行的 Fortran 的一个子集,后来才认识到这种看法并不正确。

BASIC 一词在英汉词典上的释义是:基础的,基本的,根本的。然而,在这里它却是 Beginner Allpurpose Symbolic Instruction Code (初学者通用符号指令代码)的缩写。

BASIC 语言本身具有什么特点? 为什么它能博得如此广泛用户的青睐?

(1) BASIC 是普及面最广的计算机语言。从装机量看,数量最多的是微型机,任何一台微型机上都配有或仅配有 BASIC 语言。

(2) BASIC 是最易学的程序设计语言。它的语法简单,结构清楚,它的保留字几乎都是正宗的英语,易读易记而且很少超过 6 个字母 (据统计,Quick BASIC 中保留字超过 6 个字母的仅 16 个,占 7%)。

(3) BASIC 是变化最大、发展最快的语言。这从两个方面可以说明。首先是量的变化。BASIC 的版本繁杂,数量惊人。从 APPLE 机的 SOFT BASIC 到 PC 机的 PC BASIC, BASICA, GW-BASIC, NBASIC, 直到 80 年代后期出现的 True BASIC, Turbo BASIC, Quick BASIC, 其版本多不胜数。显然,每一版本都有自己的特点,而每一新版本总是要对老版本作出不同程度的改进。举例来说,基本 BASIC 只有 17 条语句,11 种内部函数,而 True BASIC 1.0 版本已经有 155 条语句,43 种内部函数,另外还增加了 30 条命令。其次是质的变化。80 年代中期, BASIC 是解释型的非结构化语言。从 1985 年凯梅尼和库尔茨两位教授发表 True BASIC 起, BASIC 就一跃变成为兼具解释与编译功能的“两栖型”结构化程序设计语言。相比之下,其他任何一种计算机语言都没有发生过这样大的变化。

尽管 BASIC 语言的变化很大,但是最新的 BASIC 却有极好的兼容性。基本 BASIC 中的全部语句和函数,在新版 BASIC 中仍然通用。特别是在 Quick BASIC 中,只要作极少修改,不仅能编译甚至能用更快的速度直接运行原先用 PC BASIC 或 GW-BASIC 等版本编写的程序。

(4) BASIC 对用户来说是最友好的语言,对软件开发来说是最生产率最高的语言。由于 BASIC 一直保持了解释型语言的性能,所以编制和调试程序时都很方便。

另外, BASIC 本身具有的功能,如人机对话功能、字符串功能、日期功能、时钟功能、通信功能、绘图功能、动画功能、音乐功能等等,其他任何一种语言的功能难得这样全面。

这大概就是美国计算机用户和程序员,纷纷放弃了他们原先采用的计算机语言而改用 BASIC 的基本原因。

1.1.3 BASIC 百花园中的 Quick BASIC

随着 True BASIC, Turbo BASIC 和 Quick BASIC 的诞生, BASIC 摆脱了原来“玩具语言”