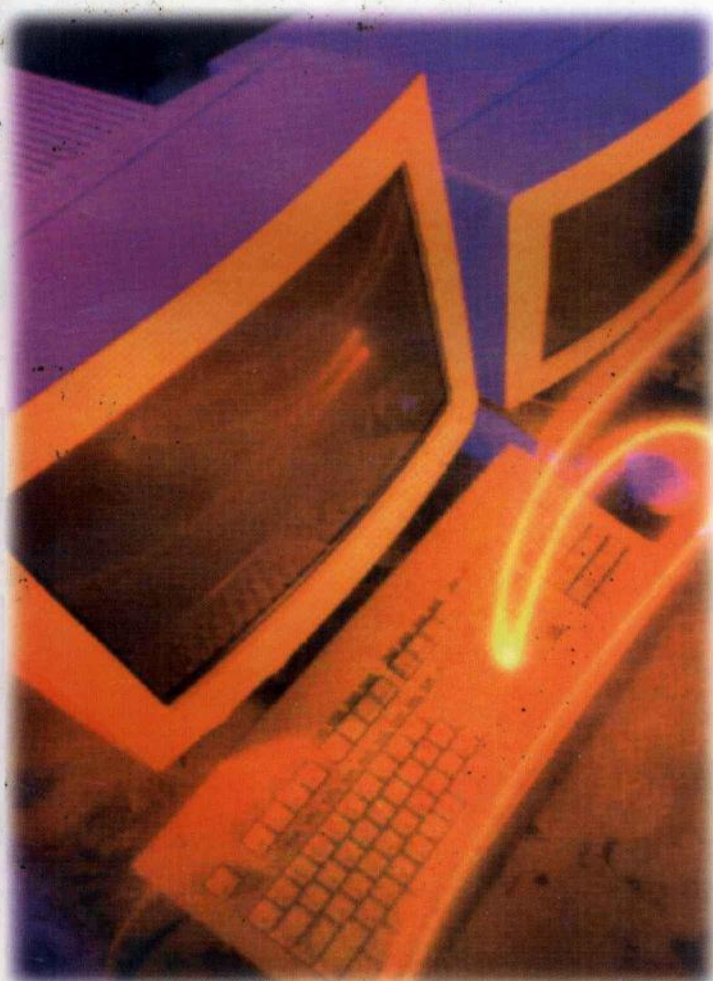


# Quick BASIC

## 程序设计

主编 冯玉山 王秀义

主审 王仁迪



计算机系列

中国矿业大学出版社

# Quick BASIC 程序设计

主 编 冯玉山 王秀义

主 审 王仁迪

中国矿业大学出版社

## 内 容 提 要

本书详细叙述了 Quick BASIC 程序设计的基础知识及程序设计方法。包括 Quick BASIC 语言基础、基本结构程序设计、模块化程序设计、数据文件、图形与动画、输入输出设备控制等内容。

本书融 Quick BASIC 程序设计基本知识与实验教学为一体,是计算机程序设计语言的一本较为理想的教科书。可作为各类高、中级职业学校“计算机应用基础”课程的教学用书,也可作为全国计算机二级考试培训教材。

责任编辑 朱明华

## 图书在版编目(CIP)数据

Quick BASIC 程序设计/冯玉山,王秀义主编,一徐州:  
中国矿业大学出版社,1999.7  
ISBN 7-81070-029-4

I. Q… II. ①冯…②王… III. Basic 语言-程序设计  
IV. TP312

中国版本图书馆 CIP 数据核字(1999)第 20447 号

中国矿业大学出版社出版发行  
(江苏徐州 邮政编码 221008)

出版人 解京选

赣榆县赣中印刷厂印刷 新华书店经销

开本 787×1092 1/16 印张 12.75 字数 321 千字

1999 年 7 月第 1 版 1999 年 7 月第 1 次印刷

印数 1~5100 册 定价 15.00 元

# 前 言

BASIC 语言以它的通俗易懂、便于掌握、适用范围广等特点,受到广大微机用户的欢迎和喜爱。

Quick BASIC 是第二代 BASIC(结构化 BASIC)语言的代表产品,用 Quick BASIC 开发的成果,很容易向 Windows 或 OS/2 环境下转换。

为适应各类在校学生及社会上广大微机用户学习计算机程序设计语言的需要,我们根据国家教委考试中心编制的“全国计算机二级考试大纲”,组织编写了这本教材。

本书根据职业教育的特点,将 Quick BASIC 程序设计的基本知识与上机实验融为一体,具有以下一些特点:

## 1. 全书循序渐进、由浅入深、逐步深化

全书分四部分共由十一章组成。第一部分(第一章~第二章)主要介绍 Quick BASIC 开发环境及其语言基础;第二部分(第三章~第六章)从结构化程序设计的角度出发,重点讨论了顺序、选择、循环结构程序及数组的有关知识;第三部分(第七章~第九章)研究模块化程序的设计方法,内容包括子程序、自定义函数与过程,以及数据文件;第四部分(第十章~第十一章)介绍图形与动画、输入输出设备控制等内容。

## 2. 重视基本概念和编程能力的培养

本书在介绍数据的表达,语句或函数的格式时,尽量结合语法要求阐述其功能,同时给出必要的说明。为加深对基本概念的理解,各章均安排了适量的例题和习题;为培养学生的程序开发能力,书中还配备了一定数量的实验,以便学生上机操作。

## 3. 注重结构化、模块化程序设计方法

本书在介绍程序结构、程序设计、程序调试方法的过程中,贯穿了结构化、模块化程序设计思想。书中所列举的编程实例,均按结构化程序要求设计,从问题的分析、程序结构的设计以及程序功能,都从模块化的角度进行介绍和说明,并配以程序流程图。

选用本教材时,建议在教学实施过程中,可根据微机的配置情况,酌情选取章节内容。总学时宜安排 70~80 学时左右,课堂教学与上机操作的课时比以 3:1 为宜。如有可能,在课程结束后,安排一至两周课程设计,对培养和提高学生的软件开发能力是非常有益的。

本书由冯玉山、王秀义主编,王仁迪主审。参加本书编写的有赵国淮(第一章),冯玉山(第二章),杨志(第三章)、陈建平(第四章),孙淑芳、李福亮(第五章),于永春(第六章),朱延美(第七章),王秀义(第八章)、王仁迪(第九章)、吕明析(第十章)、张利群(第十一章)。

由于编者水平有限,书中错误及不妥之处在所难免,敬请读者不吝赐教。

编 者  
1999 年 3 月

# 目 录

|                               |    |
|-------------------------------|----|
| 第一章 绪论                        | 1  |
| § 1.1 计算机概述                   | 1  |
| 1.1.1 计算机的功能部件及工作原理           | 1  |
| 1.1.2 计算机的组成                  | 3  |
| § 1.2 计算机语言                   | 4  |
| 1.2.1 机器语言                    | 4  |
| 1.2.2 汇编语言                    | 5  |
| 1.2.3 高级语言                    | 5  |
| § 1.3 Quick BASIC 概述          | 6  |
| 1.3.1 Quick BASIC 的特点、运行环境及安装 | 6  |
| 1.3.2 Quick BASIC 的工作屏幕       | 7  |
| 1.3.3 Quick BASIC 程序开发过程      | 9  |
| 习题一                           | 10 |
| 第二章 Quick BASIC 语言基础          | 11 |
| § 2.1 Quick BASIC 语言元素        | 11 |
| 2.1.1 基本字符集                   | 11 |
| 2.1.2 保留字                     | 11 |
| § 2.2 数据类型                    | 11 |
| 2.2.1 数据类型                    | 12 |
| 2.2.2 常量和变量                   | 13 |
| 2.2.3 变量的类型说明                 | 14 |
| § 2.3 Quick BASIC 表达式         | 15 |
| 2.3.1 算术表达式                   | 15 |
| 2.3.2 关系表达式                   | 16 |
| 2.3.3 逻辑表达式                   | 16 |
| 2.3.4 字符串表达式                  | 17 |
| § 2.4 常用标准函数                  | 17 |
| § 2.5 Quick BASIC 程序的结构       | 20 |
| § 2.6 QBasic 的上机基本操作          | 21 |
| 习题二                           | 24 |

|                                 |           |
|---------------------------------|-----------|
| 实验一 QB 程序编辑器的使用 .....           | 24        |
| 实验二 QB 语言基础 .....               | 25        |
| <b>第三章 顺序结构程序设计 .....</b>       | <b>26</b> |
| § 3.1 程序的基本结构与流程 .....          | 26        |
| 3.1.1 程序的基本结构 .....             | 26        |
| 3.1.2 流程图 .....                 | 27        |
| § 3.2 数据输入 .....                | 29        |
| 3.2.1 赋值语句(LET) .....           | 29        |
| 3.2.2 键盘输入语句(INPUT) .....       | 30        |
| 3.2.3 行输入语句(LINE INPUT) .....   | 31        |
| 3.2.4 读数/置数语句(READ/DATA) .....  | 32        |
| 3.2.5 恢复读数语句(RESTORE) .....     | 33        |
| 3.2.6 交换变量值语句(SWAP) .....       | 34        |
| § 3.3 屏幕输出语句 .....              | 34        |
| 3.3.1 屏幕输出语句(PRINT/WRITE) ..... | 34        |
| 3.3.2 格式输出语句(PRINT USING) ..... | 35        |
| 3.3.3 光标定位语句(LOCATE) .....      | 38        |
| § 3.4 格式输出函数 .....              | 38        |
| 3.4.1 TAB 函数 .....              | 38        |
| § 3.5 常用初始化语句 .....             | 39        |
| 3.5.1 CLS 语句 .....              | 39        |
| 3.5.2 CLEAR 语句 .....            | 39        |
| 3.5.3 RANDOMIZE 语句 .....        | 40        |
| § 3.6 注释、暂停及结束语句 .....          | 41        |
| 3.6.1 注释语句(REM) .....           | 41        |
| 3.6.2 暂停语句(STOP) .....          | 41        |
| 3.6.3 终止程序执行语句(END) .....       | 42        |
| 习题三 .....                       | 42        |
| 实验三 建立与运行顺序结构的程序设计 .....        | 43        |
| <b>第四章 选择结构程序设计 .....</b>       | <b>46</b> |
| § 4.1 无条件转移语句(GOTO) .....       | 46        |
| § 4.2 条件语句(IF-THEN-ELSE) .....  | 47        |
| 4.2.1 单行结构 .....                | 47        |
| 4.2.2 块结构 .....                 | 49        |
| 4.2.3 条件嵌套 .....                | 50        |
| § 4.3 多路分支 .....                | 51        |
| 4.3.1 SELECT CASE 语句 .....      | 51        |

|                                |           |
|--------------------------------|-----------|
| 4.3.2 ON—GOTO 语句               | 53        |
| § 4.4 选择结构的嵌套                  | 55        |
| 习题四                            | 55        |
| 实验四 建立与运行选择结构程序                | 57        |
| <b>第五章 循环结构程序设计</b>            | <b>59</b> |
| § 5.1 FOR/NEXT 循环结构            | 59        |
| 5.1.1 FOR/NEXT 循环结构            | 59        |
| 5.1.2 关于步长值的讨论                 | 61        |
| 5.1.3 举例                       | 62        |
| § 5.2 WHILE/WEND 循环结构          | 63        |
| § 5.3 DO/LOOP 循环结构             | 65        |
| 5.3.1 无测试条件的 DO/LOOP 循环结构      | 65        |
| 5.3.2 在 DO/LOOP 语句中加 WHILE 修饰词 | 66        |
| 5.3.3 在 DO/LOOP 语句中加 UNTIL 修饰词 | 66        |
| § 5.4 退出语句(EXIT)               | 67        |
| § 5.5 多重循环                     | 68        |
| 5.5.1 多重循环的概念                  | 68        |
| 5.5.2 多重循环的应用                  | 69        |
| 习题五                            | 71        |
| 实验五 建立与运行循环结构程序                | 73        |
| <b>第六章 数组</b>                  | <b>75</b> |
| § 6.1 数组的概念                    | 75        |
| § 6.2 数组的定义                    | 75        |
| 6.2.1 数组说明语句(DIM)              | 75        |
| 6.2.2 数组的引用                    | 76        |
| 6.2.3 与数组操作有关的语句和函数            | 78        |
| 6.2.4 静态数组与动态数组                | 79        |
| § 6.3 数组的应用                    | 80        |
| 6.3.1 统计计算                     | 81        |
| 6.3.2 数组的排序                    | 82        |
| 6.3.3 数组的查找操作                  | 84        |
| 6.3.4 矩阵的运算                    | 85        |
| 习题六                            | 88        |
| 实验六 数组                         | 89        |
| <b>第七章 子程序和自定义函数</b>           | <b>90</b> |
| § 7.1 子程序                      | 90        |

|                                      |    |
|--------------------------------------|----|
| 7.1.1 子程序的概念 .....                   | 90 |
| 7.1.2 转子语句(GOSUB)和返回语句(RETURN) ..... | 91 |
| 7.1.3 子程序的嵌套 .....                   | 92 |
| § 7.2 开关转子语句(ON-GOSUB) .....         | 93 |
| § 7.3 自定义函数 .....                    | 94 |
| 7.3.1 自定义函数的建立(DEF FN 语句) .....      | 95 |
| 7.3.2 自定义函数的调用 .....                 | 95 |
| 习题七 .....                            | 96 |
| 实验七 子程序和自定义函数的调用 .....               | 97 |

|                            |     |
|----------------------------|-----|
| <b>第八章 过程</b> .....        | 99  |
| § 8.1 变量的作用域及其属性说明 .....   | 99  |
| 8.1.1 变量的作用域 .....         | 99  |
| 8.1.2 变量属性的说明 .....        | 99  |
| § 8.2 SUB 过程 .....         | 102 |
| 8.2.1 SUB 过程的建立 .....      | 102 |
| 8.2.2 SUB 过程的调用 .....      | 103 |
| § 8.3 FUNCTION 过程 .....    | 103 |
| 8.3.1 FUNCTION 过程的建立 ..... | 103 |
| 8.3.2 FUNCTION 过程的调用 ..... | 104 |
| § 8.4 过程参数的传递 .....        | 105 |
| § 8.5 递归 .....             | 109 |
| 8.5.1 递归调用 .....           | 109 |
| 8.5.2 调整堆栈容量 .....         | 109 |
| § 8.6 模块化程序设计 .....        | 110 |
| 习题八 .....                  | 112 |
| 实验八 过程的建立与调用 .....         | 113 |

|                            |     |
|----------------------------|-----|
| <b>第九章 文件</b> .....        | 115 |
| § 9.1 文件的概念 .....          | 115 |
| 9.1.1 文件、记录与字段 .....       | 115 |
| 9.1.2 文件的分类 .....          | 117 |
| § 9.2 文件操作语句和函数 .....      | 117 |
| 9.2.1 打开文件语句(OPEN) .....   | 117 |
| 9.2.2 关闭文件语句(CLOSE) .....  | 118 |
| 9.2.3 文件操作语句 .....         | 119 |
| 9.2.4 文件操作函数 .....         | 120 |
| § 9.3 顺序文件 .....           | 121 |
| 9.3.1 顺序文件的输出(顺序写文件) ..... | 121 |

|                              |            |
|------------------------------|------------|
| 9.3.2 顺序文件的输入(顺序读文件)         | 122        |
| 9.3.3 顺序文件中数据的添加、检索、排序和修改    | 123        |
| § 9.4 随机文件                   | 126        |
| 9.4.1 随机文件的输出(随机写文件)         | 127        |
| 9.4.2 随机文件的输入(随机读文件)         | 129        |
| 习题九                          | 131        |
| 实验九 顺序文件的建立、修改和查询与随机文件的建立和修改 | 131        |
| <b>第十章 图形与动画</b>             | <b>133</b> |
| § 10.1 屏幕设置                  | 133        |
| 10.1.1 屏幕显示方式与屏幕坐标系          | 133        |
| 10.1.2 设置屏幕方式语句(SCREEN)      | 134        |
| 10.1.3 设置屏幕颜色语句(COLOR)       | 135        |
| § 10.2 绘图语句                  | 136        |
| 10.2.1 画点语句(PSET/PRESET)     | 136        |
| 10.2.2 画线语句(LINE)            | 137        |
| 10.2.3 连续画线(DRAW)            | 138        |
| 10.2.4 画圆和画弧语句(CIRCLE)       | 139        |
| § 10.3 图形的填充                 | 139        |
| § 10.4 屏幕坐标的设定               | 140        |
| 10.4.1 定义视见区语句(VIEW)         | 140        |
| 10.4.2 定义文本窗口语句(VIEW PRINT)  | 141        |
| 10.4.3 定义逻辑坐标系语句(WINDOW)     | 142        |
| 10.4.4 物理坐标与逻辑坐标的跟踪          | 144        |
| § 10.5 动画设计                  | 144        |
| 10.5.1 用 GET 语句存储图形          | 145        |
| 10.5.2 用 PUT 语句重放图形          | 145        |
| 10.5.3 动画技术                  | 146        |
| 习题十                          | 151        |
| 实验十 图形与动画设计                  | 151        |
| <b>第十一章 输入输出技术</b>           | <b>153</b> |
| § 11.1 键盘的控制                 | 153        |
| § 11.2 输出设备的控制               | 155        |
| 11.2.1 打印机的控制                | 155        |
| 11.2.2 扬声器的控制                | 156        |
| 11.2.3 显示器的控制                | 158        |
| § 11.3 报表生成                  | 159        |
| § 11.4 “菜单”技术                | 161        |

|                                       |     |
|---------------------------------------|-----|
| 习题十一.....                             | 166 |
| 实验十一 输入输出技术 .....                     | 167 |
| 附录 I Quick BASIC 语言函数、语句、命令功能一览表..... | 168 |
| 附录 I Quick BASIC 开发环境命令一览表.....       | 172 |
| 附录 II Quick BASIC 错误信息 .....          | 175 |
| 参考文献.....                             | 192 |

# 第一章 绪 论

本章主要介绍计算机的组成及工作原理,计算机语言的分类及 Quick BASIC 的一般知识,以使读者对计算机及 Quick BASIC 有一个基本的认识 and 了解。

## § 1.1 计算机概述

### 1.1.1 计算机的功能部件及工作原理

电子计算机问世半个多世纪以来,已经跨越了电子管、晶体管、集成电路和大规模集成电路等发展历程,正在向超大规模集成电路和人工智能为主要特征的第五代计算机迈进。尽管计算机技术发展迅猛,计算机种类繁多,但其基本结构却大都体现了一个著名原理,即冯·诺依曼原理。

匈牙利数学家冯·诺依曼(Von Neumann)在 1946 年提出了“存储程序控制”的原理,其基本思想是:将一个问题的计算方案即计算机程序,连同所需要的数据存储在计算机中,使计算机按照程序步骤顺序执行指定操作,能够及时取得初始数据,暂存中间结果,输出最终结果。这样,计算机就必须有一个存储器,用来存储程序和数据;有一个运算器,用来执行规定的运算;有一个控制器,以按照程序要求实现自动操作。此外,还必须有输入输出设备,以输入原始数据和输出计算结果。这些,就是“存储程序式计算机”即计算机的基本构成。

综上所述,计算机一般由 5 个功能部件构成:存储器、运算器、控制器、输入设备和输出设备。下面简述各个部件的功能。

#### 1. 存储器

存储器是计算机的记忆装置,用于存放程序和数据,即计算步骤、初始数据、中间结果和最终结果。存储器分为两种,一种是内存储器(简称内存);另一种是外存储器(简称外存)。

(1) 内存储器(主存储器)。为一组集成电路芯片,可与运算器、控制器、输入输出设备直接交换或传递信息,存取速度快,但容量有一定限制。按存取方式,内存储器又分为只读存储器和随机存储器。

只读存储器 ROM(Read Only Memory)将确定不变的信息固化进电路芯片,只能读出不能写入,断电后信息不会消失。微型机常将专用程序、监控程序或操作系统模块等固化于只读存储器中,以实现计算机最基本的功能。

随机存储器 RAM(Random Access Memory)用于存储计算机中正在执行的程序和数据,可以随意读写,断电后信息自动消失。随机存储器的容量大小是计算机的一个重要指标,对程序的运行有重要意义。

(2) 外存储器(辅助存储器)。为内存储器的后备和扩充,只能与内存储器成批交换数

据,存取速度较慢,但存储容量大,数据可永久性保存,用于存放当前暂时不用的程序和数据。常见的外存储器有软磁盘、硬磁盘、磁带、光盘等。

内存储器是计算机硬件的重要资源,其主要指标是:存取时间、存取周期和存储容量。存取时间是指对存储器写入或读出一个数据所用的时间;存取周期是指对存储器连续两次写入或读出数据所需的最短时间。存取时间和周期通常用毫微秒表示,其值大小表明了计算机的运行速度。存储容量指计算机中可存储的数据总量,通常以字节(Byte)为基本单位,常用的容量单位为 KB(1024 B)、MB(1024 KB)、GB(1024 MB)等,存储容量越大,计算机的数据处理能力越强。

2. 运算器

运算器是计算机对数据进行运算和操作的部件,由一系列寄存器和加法器等电子器件和电路组成。可对数据进行加、减、乘、除等基本算术运算,并能实现逻辑运算。数据由内存储器提供,运算结果可暂时存于寄存器中,也可以送回内存储器中保存。

3. 控制器

控制器是计算机的指挥中心,对计算机资源进行总调度的部件,能够按照用户预先编好的程序或发出的指令进行工作。计算机执行的每一条指令,包括取出指令、翻译指令,形成各种操作信号指挥各个部件之间按恰当的顺序协调地完成各种操作,无一不是在控制器的统一调度下进行的。

运算器和控制器构成中央处理器,简称 CPU(Central Processing Unit)。中央处理器是计算机最核心的部分,也是衡量计算机种类和档次的最重要的标志。

4. 输入设备和输出设备

输入与输出设备是实现人机对话的设备。计算机通过输入设备将解题程序和原始数据输入存储器保存;通过输出设备将计算结果及有关信息输出给用户。常用的输入设备有键盘、鼠标器、扫描仪、触摸屏等;常用的输出设备有显示器、打印机、绘图仪等。

在上述部件中,中央处理器与内存储器结合起来称为计算机的“主机”,外存储器、输入设备和输出设备等称为计算机的外部设备(或外围设备,简称“外设”)。

计算机各功能部件之间的联系如图 1.1 所示。

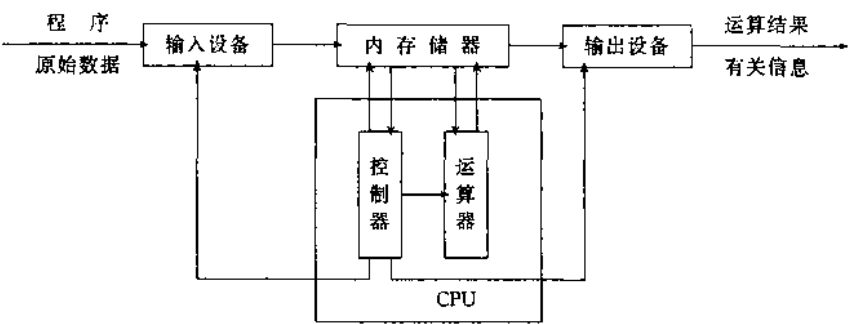


图 1.1 计算机功能部件之间的联系

计算机的工作过程可概述为:在控制器统一指挥下,由输入设备输入程序和数据,并把它们存放在内存储器中;在程序运行时,计算机依次从内存中读出程序的各条指令(取指令);随之,分析指令执行何种功能、何数参与运算、产生相应的控制信号,并发送到各个执行

部件,由计算器执行相应的运算,同时根据指令要求,决定是否把存储器中的运算结果或有关信息输送到输出设备上(执行指令)。一条指令执行完后,再处理下一条应执行的指令,直到程序中的指令全部执行完为止。

### 1.1.2 计算机的组成

计算机系统是由硬件系统和软件系统两大部分组成的。

#### 1. 硬件系统

硬件系统是计算机系统物理部件的总称,上述计算机的五大功能部件组成计算机的主要硬件部分。仅有硬件的计算机称为“裸机”,裸机仅仅提供了特定的基本功能,难以充分发挥其能力,要想充分发挥计算机的功能,就必须要有相应软件的支持。

#### 2. 软件系统

软件系统是计算机运行所需的各种程序、数据及其有关的文档资料的总称。软件的作用在于负责管理计算机的各种硬软件资源,并充分发挥其功能。从用户角度来看,软件是用户与计算机之间交流信息的桥梁,人们可以通过编写软件来规定计算机硬件应作的一系列动作,而不必掌握更多的硬件知识。软件极大地开发了计算机的应用领域,充分发挥了计算机的效率,并为用户使用计算机提供了良好的可靠的运行环境。

软件系统可分为系统软件和应用软件两大类:

##### (1) 系统软件

系统软件是管理、控制、协调、维护、监控计算机软硬件资源,并为计算机开发提供支持的一类软件。主要有操作系统、语言处理程序、服务程序等。

① 操作系统,是最基本的系统软件,为一系列管理和控制计算机资源的程序集合,为用户与计算机之间提供的第一层界面。主要实施处理器管理、存储管理、文件管理、设备管理和作业管理,从而可扩大机器功能、提高运行效率、方便用户使用、减少系统故障,为高效合理地利用计算机资源提供完善的运行环境。

操作系统按其功能分为:单用户操作系统、多用户操作系统、网络操作系统和实时操作系统等。微型机操作系统有代表性的是 DOS(Disk Operator System)和 Windows。前者已开发出 DOS 1.0—6.22 等多个版本,拥有最广大的用户;后者以多窗口图形界面提供处理多任务的能力,正受到越来越多用户的欢迎。

② 语言处理程序,是支持软件开发和维护的程序集合,为用户开发和维护应用软件提供中间支持和开发环境。解决问题的计算方案(即计算机程序)需要用特定的语言表达出来,这些特定的语言是指计算机能够接受和处理的语言,语言处理程序正是为用户的程序提供特定的语言支持和开发环境。如微型机配备的机器语言、汇编语言、BASIC 语言解释程序、PASCAL、FORTRAN、COBOL、C 等语言的编译系统等。

③ 服务程序,是系统软件中的实用程序组,为软件开发、实施和维护等提供服务。如编辑程序、测试程序、诊断程序、排错程序以及调试工具软件等。

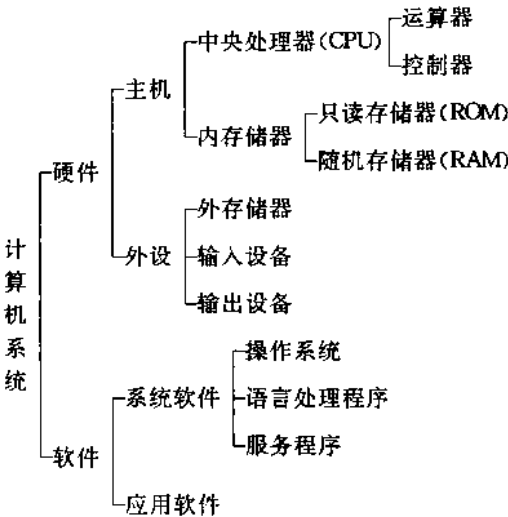
##### (2) 应用软件

应用软件是用户利用计算机软硬件资源为解决各类应用问题而编写的软件。这类软件一般专用于特定应用领域,品种多、范围广。如科学计算、工程设计、情报检索、工资管理、自动控制程序等等。

综上所述,计算机是由硬件和软件两大部分组成的。硬件能力的发挥需要软件提供支

持,软件的运行需要以硬件为基础,二者相辅相成,缺一不可,合起来才构成完整的计算机系统。

计算机组成的结构层次大致如下:



## § 1.2 计算机语言

电子计算机以其自动、高速、精确地完成信息处理、存储和传输的特点,已成为现代化信息处理不可替代的有力工具,在科学计算、数据处理、自动控制、人工智能等领域发挥着巨大的作用。要使计算机能解决任何问题,最大限度地发挥计算机的功能,就必须学会用计算机能识别的符号或代码编制程序,这些符号或代码即构成了计算机语言。

随着计算机的发展,计算机语言经历了机器语言、汇编语言、高级语言的发展过程,这三类语言以各自的特点在人机信息交流中发挥着极其重要的作用。

### 1.2.1 机器语言

机器语言是与计算机硬件的设计同时产生的,是一种依附于机器的语言,不同的计算机(CPU)使用的机器语言是不同的。机器语言由计算机能直接识别的二进制代码(0和1)组成,其语言程序表现为一批二进制编码的机器指令群,通常称为目标程序。计算机在执行程序时,先将指令从内存的指令代码区取至CPU,然后对指令代码进行译码,从而针对不同指令产生不同的操作控制信号,以完成指令所规定的功能或操作。显然,机器语言是计算机惟一能够直接识别的语言,但是,只有当机器语言的指令代码符合其与计算机事先约定的代码时,计算机才能理解和执行它。

机器语言的优点是机器能够直接执行,因而速度快,占用内存单元少。但其缺点也是显而易见的,主要表现为:各种代码不便于记忆、阅读和理解;编程十分繁琐,程序直观性差,非常容易出错,给程序的检查和调试带来较大困难;由于不同的计算机有不同的指令系统,这就造成在不同的计算机之间,机器语言不能通用,因而不便于推广。

### 1.2.2 汇编语言

由于机器语言与人们惯用的语言差别太大,为了便于理解和记忆机器指令,人们采用能帮助记忆的英文缩写为符号(机器指令助记符)来代替机器指令的操作码。采用这种助记符的语言就是汇编语言。用指令助记符及符号地址所书写的指令称为汇编格式指令,用汇编格式指令编写的程序称为汇编程序。

用汇编语言书写程序比用机器语言方便得多,它对程序的执行有一定程度的描述,便于人们阅读和记忆。目前,一般系统程序多是用汇编语言编写的。但是,由于汇编语言程序与机器语言程序是一一对应的,因而仍然没有摆脱对计算机硬件的依赖性,程序的移植性很差;汇编语言要求熟悉掌握计算机的硬件性能、寻址方式等许多知识,对编程人员的要求较高,这些都在一定程度上妨碍汇编语言的交流与推广。

需要指出的是,虽然汇编语言为人们带来了一定的方便,但由于计算机只能直接识别机器语言,因此在计算机执行指令之前,必须将汇编语言源程序“翻译”成机器语言的目标程序,这种工作可由计算机自动完成,称为机器汇编(见图 1.2)。

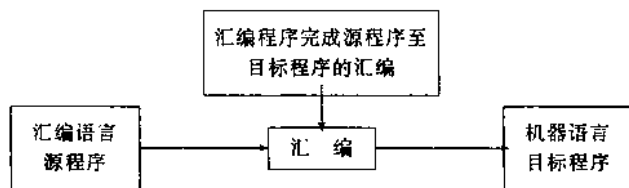


图 1.2 源程序的汇编过程

### 1.2.3 高级语言

为了便于计算机应用的推广与普及,50 年代末期出现了高级语言。高级语言的出现是计算机发展中“最惊人的成就”,高级语言不需要编程人员对机器指令系统有深入的了解,是一种面向使用者(用户)而非面向机器的语言,这是它与机器语言和汇编语言最根本的区别。高级语言的优点是相当突出的,主要体现在:

- (1) 高级语言在形式上更接近于人们习惯的自然语言和数学语言,使用起来十分方便;
- (2) 高级语言源程序由语句组成,每条语句的功能相当于若干条机器指令,因此语句的功能很强,同时用语句编写程序要容易得多;
- (3) 一个高级语言源程序可以在不同型号的机器上运行(只要配有该种语言的解释或编译程序),程序的可移植性很强,便于交流和推广。

高级语言的种类很多,目前常用的高级语言有: BASIC、FORTRAN、PASCAL、COBOL、C 语言等,它们各具特点。

高级语言程序同样不能被计算机直接识别,程序被执行前必须进行语言处理,即通过语言处理程序将其“翻译”成计算机能够接受的信息。这种“翻译”通常以解释和编译两种方式进行。

解释方式是对源程序逐句进行翻译和执行,即边解释边执行,直至程序结束,见图 1.3(a)所示。编译方式是将源程序一次性翻译为目标程序,然后再执行目标程序得到结果,见图 1.3(b)所示。

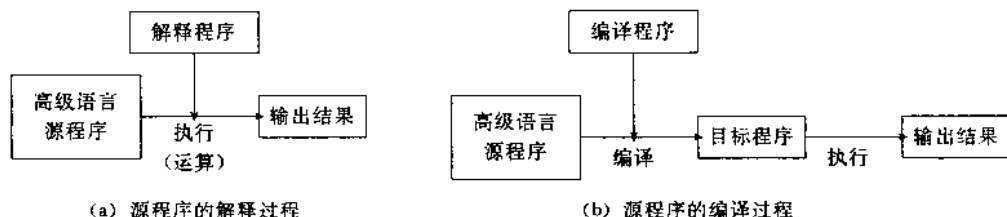


图 1.3

编译方式执行效率较高,但占用较多的内存空间;解释方式效率较低,但使用方便且占内存较少。基本 BASIC 一般采用解释方式;FORTRAN、PASCAL、C 等语言采用的是编译方式。

高级语言虽然使用方便,但也有其缺点,主要是:必须配备相应的编译程序才能使用,这就占用了大量的内存空间,使用户内存大为减少;由于不能直接面对机器,因而程序执行效率较低。

### § 1.3 Quick BASIC 概述

Quick BASIC 是美国 Microsoft 公司于 1987 年推出的 BASIC 语言版本,它吸收了 Pascal、C 语言的特点,集解释型与编译型的优点于一体,既具有解释型的简单易学与调试、检测的方便性,又具有编译的快速与通用性。同时还对 BASIC 语言的功能进行了较大的扩充,尤其在结构化程序设计方面,在屏幕编辑、编译、连接、调试等方面,已达到能与 Pascal、C 语言相媲美的程度。

#### 1.3.1 Quick BASIC 的特点、运行环境及安装

##### 1. Quick BASIC 的特点

Quick BASIC 是兼有解释与编译功能的结构化程序设计语言,除了具有集成化的编程环境、友好的菜单式界面、完备的调试手段外,还具有以下特点:

(1) 易学易记。QB(Quick BASIC 的简称,下同)使用的保留字均为英语单词,且很少超过 6 个字母(据统计 QB 中超过 6 个字母的保留字仅 16 个,占 7%),易读易记;另外,QB 具有在线帮助功能,提供了所有的指令定义、语法以及实例,十分适合初学者学习。

(2) 功能丰富。具有人机对话、字符串、日期、时钟、通讯、绘图、动画、音乐等诸多功能,适用范围广泛,可解决科技计算、数据处理等多种类型的问题。特别是在灵巧编辑器、声音、图像、动画、通讯等方面的处理以及 Quick 库、独立库的应用方面都有明显优势。

(3) 使用快速。其“快速”体现在:运行快,可进行仿真(协处理器)运算,运行速度为基本 BASIC 的 20 至 50 倍;编辑快,具有灵巧高效的编辑功能,可立即检查刚键入语句的语法,并立即将正确的语句译成可执行代码;调试快,可在任何一点暂停程序运行,编辑后从断点处恢复运行;建立执行文件快,通过菜单选择,只需四步即可建立在 DOS 状态下能直接执行的 EXE 文件。

(4) 兼容性强。对基本 BASIC 及其增强型(如 PC BASIC、BASICA、GW-BASIC 等)向下兼容,用这些版本的语言开发的程序,只要稍加改动就可以在 QB 上运行,且效率更高;在

QB 环境下开发的程序很容易移植到 Windows 环境下的 Visual BASIC 中运行。

(5) 相容性突出。QB 与 MS-DOS、Quick C、Microsoft C 以及 C++等都是 Microsoft 公司的产品,其窗口、界面以及命令风格相当接近,与之有突出的相容性,掌握 QB 对熟悉和掌握其它语言环境的使用将会更加方便。

## 2. Quick BASIC 运行环境及安装

### (1) QB 的运行环境

QB 对硬件有如下要求:

IBM-PC 机及其兼容机,内存在 320 KB 以上,至少有一个软盘驱动器。

对软件的要求为:需 MS-DOS 2.1 以上版本的支持。同时应包括下列文件(视版本而定,这里列出的是 4.0 版本):

|            |            |
|------------|------------|
| QB.EXE     | 集成开发环境     |
| BC.EXE     | 编译程序       |
| QB.QLB     | Quick 程序库  |
| QB.LIB     | 独立程序库      |
| QB.BI      | 包含文件       |
| BRUN40.EXE | 运行模块       |
| BQLB40.LIB | Quick 程序库  |
| BRUN40.LIB | 运行模块库      |
| BCOM40.LIB | 交互式运行库     |
| LINK.EXE   | MS 覆盖连接程序  |
| LIB.EXE    | MS 程序库处理程序 |
| NOCOM.OBJ  | 目标文件       |
| MOUSE.COM  | 鼠标器驱动程序    |
| QB.HLP     | 在线帮助       |

为给 QB 建立合适的工作环境,系统配置文件 CONFIG.SYS 最好作相应改动,譬如:

|                   |        |
|-------------------|--------|
| DEVICE=MOUSE.COM  | 支持鼠标器  |
| PATH=C:\ZM1;C:\QB | 缺省路径   |
| SET LIB=C:\CXK    | 程序库路径  |
| SET NO87=(空格)     | 协处理器仿真 |

### (2) QB 的安装

QB 的安装很简单,如果源盘上有 SETUP.BAT 安装文件,直接运行即可;也可以把相应文件直接拷贝到硬盘。若在软盘上工作,则可根据计算机设备配置情况,按照 QB 运行需要准备工作盘。

需要指出的是,为方便用户,MS-DOS 5.0 以上版本自带有 Quick BASIC 的简化版本 QBasic,用户在系统下可直接运行。

## 1.3.2 Quick BASIC 的工作屏幕

进入 QB 的工作环境只需在 DOS 下打入 QB 命令(使用简化版本则打入 QBASIC 命令),如: C:\QB>QB(回车),即进入 QB 环境。

QB 环境表现为一个多窗口的工作屏幕和下拉式菜单提供的命令系统。其工作屏幕如