



新大纲

全国计算机等级考试指导

QBasic语言 程序设计考试指导 (二级)

李大友 主编
彭波 殷光复 编著



电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
URL: <http://www.phei.com.cn>

清华大学出版社

清华大学出版社

Objective C语言 程序设计与考试指导

第二版

作者：王 强



清华大学出版社



全国计算机等级考试指导

QBasic 语言程序设计考试指导(二级)

李大友 主编

彭 波 殷光复 编著

電子工業出版社

Publishing House of Electronics Industry

内 容 简 介

本书共 11 章,前 2 章介绍了结构化程序设计的基本概念、QBasic 语言的特点、基本操作以及常用的菜单命令。第 3~11 章结合实例讲述了 QBasic 语言的程序设计语句、全局变量与局域变量、函数、数组及绘图功能等内容。附录 A~E 列出了 QBasic 语言的保留字、语句、函数和常见错误等内容。附录 F、G 给出了计算机等级考试二级考试大纲和一套二级考试模拟试题(含答案)。附录 H 为上机考试说明。

本书可以作为等级考试用书,也可以作为高等院校的教学参考书。

未经许可,不得以任何方式复制或抄袭本书之部分或全部内容。

版权所有,翻版必究。

图书在版编目(CIP)数据

QBasic 语言程序设计考试指导:二级/李大友主编;彭波等编著. - 北京:电子工业出版社,2000.1

(全国计算机等级考试指导)

ISBN 7-5053-5424-8

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

中国版本图书馆 CIP 数据核字(1999)第 31327 号

丛 书 名:全国计算机等级考试指导

书 名:QBasic 语言程序设计考试指导(二级)

主 编:李大友

编 著:彭 波 殷光复

责任编辑:吕 迈

排版制作:电子工业出版社计算机排版室

印 刷 者:北京李史山胶印厂

出版发行:电子工业出版社 URL:<http://www.phei.com.cn>

北京市海淀区万寿路 173 信箱 邮编 100036

经 销:各地新华书店

开 本:787×1092 1/16 印张:18 字数:460 千字

版 次:2000 年 1 月第 1 版 2000 年 1 月第 1 次印刷

书 号:ISBN 7-5053-5424-8
TP·2723

印 数:8000 册 定 价:20.00 元

凡购买电子工业出版社的图书,如有缺页、倒页、脱页、所附磁盘或光盘有问题者,请向购买书店调换。

若书店售缺,请与本社发行部联系调换。电话 68279077

序 言

《全国计算机等级考试指导》和《模拟试题详解与模拟试卷》两套系列丛书是按 1998 年修订大纲的要求而编写的。

自从全国计算机等级考试推出以来,已有上百万人参加考试,有力地推动了计算机应用技术的发展。

计算机技术是一种日新月异、飞速发展的技术。而全国计算机普及教育又是以普及和提高应用水平为目的。这就提出了一个问题,如何处理普及和提高的关系。以普及为主线,兼顾提高,则是应遵循的原则。为此,教育部考试中心组织部分专家、教授,对考试大纲进行了必要的修改。

全国计算机等级考试,根据计算机应用水平的不同,分为 4 个等级、7 种类型。其中一级 A 类和一级 B 类,以面向文字处理和数据库应用系统为主,以满足办公自动化领域的基本要求。

一级 A 类分为 DOS 环境和 Windows 环境两种,应试者可以任选其一。要求应试者掌握计算机基础知识;微机系统基本组成;了解操作系统的基本功能,掌握一种操作系统的使用方法(DOS 环境和 Windows 环境,分别会用相应的操作系统);了解文字处理基本知识、掌握一种字表处理软件的使用方法;了解数据库应用系统的基本功能、掌握数据库应用系统的操作方法;了解计算机网络和因特网的初步知识;了解计算机病毒的防治常识。

一级 B 类是面向公务员的考试。要求掌握计算机基础知识、DOS 操作系统的功能和使用;汉字处理系统 WPS 的功能和使用;FoxBASE+ 数据库应用系统的基本概念和基本操作。

二级考 5 种高级语言程序设计。要求应试者掌握计算机基础知识、操作系统的功能和使用;具有使用一种高级语言(C、Pascal、FORTRAN、QBasic 或数据库语言)编程、调试和运行的能力。

三级仍分为 A、B 两类。

三级 A 类面向测控领域的技术人员。要求掌握微机原理、汇编语言程序设计、接口技术、计算机网络、软件技术基础以及微机在测控领域的应用技术。

三级 B 类面向软件应用领域的技术人员。要求掌握计算机基础知识、计算机网络、数据结构与算法、操作系统、软件工程以及微机在管理信息系统或数值计算机或辅助设计方面(三者任选其一)的应用能力。

四级相当于大学计算机专业本科水平。要求具有计算机软、硬件系统的设计开发能力。要求掌握计算机系统原理、计算机体系结构、计算机网络与通信、离散数学、数据结构与算法、操作系统、软件工程和数据库系统原理等方面的基础理论知识和软、硬件系统的开发能力。

这两套丛书重点面向一级和二级应试人员,严格按 1998 年修订大纲的要求编写。丛书层次清楚、结构严谨、深入浅出、便于自学。

这两套丛书不仅可作为等级考试用书,还可作为高等院校的教学参考用书。

主编 李大友

前 言

现代计算机技术正在以惊人的速度发展着,它已经渗透到了人类生活的每一个角落,计算机知识已经成为当代人类文化不可缺少的重要组成部分。程序设计是计算机应用人员的基本功,学习程序设计,才能懂得计算机是怎样工作的。大部分高等学校都把程序设计列为学生的必修课程,全国计算机等级考试二级考试内容就是程序设计。

QBasic 是 MS-DOS 5.0 以上版本系统随盘提供的一种结构化编程语言。它是 GW-BASIC 的超集,是 Quick BASIC 和 Visual BASIC 的一个子集,是学习计算机高级语言与结构化编程理想的首选语言,是由 DOS 平台过渡到 Windows 平台的最佳途径,是没有程序设计经验的人学习程序设计的较佳起点。

程序设计语言的选择呈现多样化的趋势。QBasic 语言始终体现 BASIC 语言初创者宗旨的一项承诺:计算机对于所有学生而言将变成非常容易掌握的工具。

概括说来,QBasic 具有以下主要特点:

1. 在体现先进性的同时,保持了“易学易用”的特点,特别适合于初学者;
2. 所有关键字均用人们比较习惯的英文或英文缩写表示,便于对关键字功能的理解;
3. 完全结构化和模块化,与其他先进的高级语言无异;
4. 功能上具有较大的扩充,融入了其他高级语言的精华,提供了新的选择结构,改进和增加了循环结构,丰富了函数和子程序等。
5. 上机操作方便,界面友好,既能使用鼠标,又具有灵活方便、功能齐全的编辑环境。

为此,作者将本书热诚地推荐给广大读者。

作者长期从事计算机教学,根据个人的经验,学习 QBasic 程序设计语言时,读者应该注意三个方面:深入理解各种语句和标准函数的功能及使用;通过大量的编程练习不断提高编程技术;保证充分的上机训练时间,加强对有关知识的感性认识。

本书由李大友教授主编,QBasic 语言程序设计部分由彭波副教授编写,附录 H 上机考试说明由殷光复教授编写。全书由李大友教授统稿和审定。

编者
1999.9

目 录

第 1 章 程序设计概述	(1)
1.1 程序设计语言	(1)
1.1.1 计算机语言的层次	(1)
1.1.2 常用高级语言简介	(3)
1.1.3 高级语言的执行方式	(4)
1.2 计算机程序算法	(5)
1.2.1 算法的概念	(5)
1.2.2 算法的表示	(6)
1.2.3 程序流程图与 N-S 图	(7)
1.2.4 举例	(9)
1.3 结构化程序设计	(9)
1.3.1 结构化程序设计的概念	(10)
1.3.2 结构化程序设计的方法	(11)
本章小结	(12)
习题	(12)
第 2 章 QBasic 语言基础	(13)
2.1 BASIC 语言的发展	(13)
2.1.1 最初发展时期	(13)
2.1.2 蓬勃发展时期	(13)
2.1.3 调整巩固时期	(14)
2.1.4 走向未来时期	(14)
2.2 QBasic 语言的特点	(14)
2.3 QBasic 的基本操作	(16)
2.3.1 QBasic 的启动	(16)
2.3.2 QBasic 的操作环境	(17)
2.3.3 QBasic 程序的编辑与运行	(19)
2.3.4 联机屏幕帮助	(21)
2.3.5 退出 QBasic 系统	(22)
2.4 QBasic 常用菜单命令	(22)
2.4.1 打开主菜单及子菜单和执行子菜单命令的方法	(22)
2.4.2 “File”子菜单	(23)
2.4.3 “Edit”子菜单	(25)
2.4.4 “View”子菜单	(26)
2.4.5 “Search”子菜单	(27)
2.4.6 “Run”子菜单	(28)

2.4.7 “Debug”子菜单	(29)
2.4.8 “Option”子菜单	(31)
2.4.9 “Help”子菜单	(32)
本章小结	(33)
习题	(34)
第3章 QBasic 程序设计初步	(35)
3.1 QBasic 程序的结构	(35)
3.1.1 QBasic 源程序的组成规则	(35)
3.1.2 QBasic 系统的规范化处理	(36)
3.2 常量	(36)
3.2.1 数值型常量	(36)
3.2.2 字符串常量	(38)
3.2.3 符号常量	(39)
3.3 变量	(40)
3.3.1 变量的命名	(40)
3.3.2 变量的类型	(41)
3.3.3 变量的标识	(41)
3.3.4 几点说明	(42)
3.4 标准函数	(42)
3.4.1 三角函数	(43)
3.4.2 算术函数	(44)
3.4.3 取整函数	(44)
3.5 运算符和表达式	(45)
3.5.1 算术运算符	(45)
3.5.2 算术表达式及运算顺序	(46)
3.5.3 算术表达式中数据类型的匹配规则	(46)
3.5.4 简单字符串表达式	(47)
3.6 编写和运行一个简单的程序	(47)
3.6.1 编写程序	(47)
3.6.2 输入并运行程序	(48)
3.6.3 存储程序	(49)
本章小结	(49)
习题	(51)
第4章 顺序结构程序设计	(53)
4.1 赋值语句(LET 语句)	(53)
4.1.1 LET 语句的格式与功能	(53)
4.1.2 LET 语句的使用说明	(54)
4.1.3 LET 语句的应用举例	(55)
4.2 键盘输入语句(INPUT 语句)	(55)
4.2.1 INPUT 语句的格式与功能	(56)
4.2.2 INPUT 语句的执行过程	(56)
4.2.3 INPUT 语句的使用说明	(57)
4.2.4 行输入语句 LINE INPUT	(57)
4.2.5 INPUT 语句的应用举例	(58)

4.3 读数/置数语句(READ/DATA 语句)	(58)
4.3.1 READ/DATA 语句的格式与功能	(58)
4.3.2 READ/DATA 语句的执行过程	(59)
4.3.3 READ/DATA 语句的使用说明	(59)
4.3.4 恢复数据指针 RESTORE 语句	(60)
4.3.5 三种提供数据语句的比较	(61)
4.4 输出语句(PRINT 语句)	(62)
4.4.1 PRINT 语句的格式与功能	(62)
4.4.2 PRINT 语句的执行过程	(62)
4.4.3 PRINT 语句的输出格式	(63)
4.4.4 打印输出语句(LPRINT 语句)	(66)
4.5 顺序结构中的其他语句	(67)
4.5.1 注释语句(REM 语句)	(67)
4.5.2 暂停语句(STOP 语句)	(67)
4.5.3 清屏语句(CLS 语句)	(67)
4.5.4 结束语句(END 语句)	(67)
4.6 应用程序实例	(68)
本章小结	(69)
习题	(70)
第 5 章 选择结构程序设计	(74)
5.1 关系表达式和逻辑表达式	(74)
5.1.1 关系运算符和关系表达式	(74)
5.1.2 逻辑运算符和逻辑表达式	(75)
5.2 行 IF 语句	(77)
5.2.1 行 IF 语句的格式与功能	(77)
5.2.2 行 IF 语句的三种常用格式	(77)
5.3 块 IF 结构	(79)
5.3.1 块 IF 结构的格式与功能	(79)
5.3.2 块 IF 结构的使用说明	(79)
5.3.3 块 IF 结构举例	(80)
5.4 在块 IF 结构中使用 ELSEIF	(83)
5.4.1 条件语句的嵌套	(83)
5.4.2 ELSEIF 结构的格式与功能	(83)
5.4.3 ELSEIF 结构举例	(84)
5.5 多分支选择结构(SELECT CASE)	(85)
5.5.1 SELECT CASE 语句的格式与功能	(86)
5.5.2 SELECT CASE 语句的使用说明	(86)
5.5.3 SELECT CASE 语句举例	(88)
5.6 多分支转移语句(ON GOTO)	(91)
5.6.1 ON GOTO 语句的格式与功能	(91)
5.6.2 ON GOTO 语句举例	(91)
5.7 几种选择结构语句的比较	(92)
5.8 应用程序实例	(93)
本章小结	(95)

习题	(96)
第6章 循环结构程序设计	(100)
6.1 循环结构的概念	(100)
6.1.1 循环的概念	(100)
6.1.2 QBasic 实现循环的方法	(100)
6.2 WHILE 循环结构	(102)
6.2.1 WHILE 循环的格式与功能	(102)
6.2.2 WHILE 循环的使用说明	(102)
6.2.3 WHILE 循环举例	(103)
6.3 FOR 循环结构	(103)
6.3.1 FOR 循环的格式与功能	(104)
6.3.2 FOR 循环的使用说明	(104)
6.3.3 FOR 循环的参数	(105)
6.3.4 EXIT FOR 语句	(107)
6.3.5 FOR 循环举例	(107)
6.4 DO 循环结构	(108)
6.4.1 DO 循环结构的--般格式	(108)
6.4.2 简单 DO 循环及 EXIT DO 语句	(109)
6.4.3 带 WHILE 子句的 DO 循环	(110)
6.4.4 带 UNTIL 子句的 DO 循环	(110)
6.5 循环的嵌套	(111)
6.5.1 循环嵌套的形式	(112)
6.5.2 循环嵌套的执行过程	(112)
6.5.3 循环嵌套的使用说明	(112)
6.6 应用程序实例	(114)
本章小结	(116)
习题	(117)
第7章 数组	(121)
7.1 有关数组的基本概念	(121)
7.1.1 数组与数组元素	(121)
7.1.2 数组元素	(122)
7.1.3 数组的定义	(123)
7.1.4 静态数组和动态数组	(124)
7.1.5 相关的语句和函数	(125)
7.2 一维数组及其应用	(127)
7.2.1 一维数组的基本操作	(127)
7.2.2 一维数组的实际应用	(128)
7.3 二维数组及其应用	(134)
7.3.1 二维数组的基本操作	(134)
7.3.2 二维数组的实际应用	(137)
7.4 应用程序实例	(141)
本章小结	(145)
习题	(146)
第8章 函数与子程序	(149)

8.1 自定义函数	(149)
8.1.1 DEF 函数	(149)
8.1.2 FUNCTION 函数	(151)
8.2 子程序	(154)
8.2.1 GOSUB 子程序	(154)
8.2.2 SUB 子程序	(158)
8.3 全局变量与局部变量	(160)
8.3.1 局部变量	(161)
8.3.2 全局变量	(162)
8.3.3 共享变量	(163)
8.4 参数传递	(164)
8.4.1 参数传递的概念	(164)
8.4.2 地址传递方式	(165)
8.4.3 值传递方式	(166)
8.5 过程的嵌套调用与递归调用	(167)
8.5.1 过程的嵌套调用	(167)
8.5.2 过程的递归调用	(169)
8.6 应用程序实例	(170)
本章小结	(174)
习题	(176)
第9章 字符串处理	(178)
9.1 字符串常量和变量	(178)
9.1.1 字符串常量	(178)
9.1.2 字符串变量	(179)
9.1.3 字符串的输入与输出	(180)
9.2 字符串函数	(181)
9.2.1 子字符串函数	(181)
9.2.2 转换类字符串函数	(182)
9.2.3 其他字符串函数	(184)
9.3 字符串表达式	(185)
9.3.1 字符串的连接	(185)
9.3.2 字符串的比较	(186)
9.4 字符串数组	(187)
9.5 自选输出格式	(189)
9.5.1 屏幕定位语句(LOCATE)	(189)
9.5.2 自选打印格式语句(PRINT USING)	(190)
9.6 应用程序实例	(193)
本章小结	(195)
习题	(197)
第10章 数据文件	(199)
10.1 文件概述	(199)
10.1.1 文件的基本概念	(199)
10.1.2 文件属性	(201)
10.1.3 文件缓冲区及读写操作	(201)

10.2 顺序文件及其操作	(201)
10.2.1 顺序文件的存放格式及特点	(201)
10.2.2 顺序文件的常用语句和函数	(203)
10.2.3 建立顺序文件	(204)
10.2.4 调用顺序文件	(205)
10.2.5 修改顺序文件	(206)
10.3 随机文件及其操作	(207)
10.3.1 随机文件的存放格式及特点	(207)
10.3.2 随机文件的常用语句和函数	(208)
10.3.3 建立随机文件	(210)
10.3.4 调用随机文件	(211)
10.3.5 修改随机文件	(212)
10.4 记录型变量	(213)
10.4.1 记录类型及记录变量	(213)
10.4.2 用于记录变量的读写语句	(214)
10.4.3 记录变量的举例	(215)
本章小结	(216)
习题	(218)
第 11 章 屏幕控制和作图	(220)
11.1 文本模式与图形模式	(220)
11.1.1 文本模式	(220)
11.1.2 图形模式	(220)
11.2 屏幕控制语句	(221)
11.2.1 SCREEN 语句	(221)
11.2.2 COLOR 语句	(222)
11.2.3 其他屏幕控制语句	(223)
11.3 画点和画线	(224)
11.3.1 画点	(224)
11.3.2 画线	(226)
11.4 画圆、椭圆和圆弧	(228)
11.4.1 画圆	(228)
11.4.2 画椭圆	(228)
11.4.3 画圆弧	(229)
11.5 图形着色及图案填充	(230)
11.5.1 图形着色	(230)
11.5.2 图案填充	(231)
11.6 图形窗口	(232)
11.6.1 窗口语句(WINDOW 语句)	(232)
11.6.2 视窗语句(VIEW 语句)	(233)
11.7 应用程序实例	(234)
本章小结	(236)
习题	(237)
附录 A 标准 ASCII 字符集	(238)
附录 B QBasic 保留字	(240)

附录 C	QBasic 语句及功能	(242)
附录 D	QBasic 标准函数及功能	(246)
附录 E	QBasic 常见错误信息	(248)
附录 F	计算机考级二级考试大纲	(250)
附录 G	二级笔试模拟试题	(253)
附录 H	上机考试说明	(265)

第 1 章 程序设计概述

现在,计算机已经被广泛地应用于社会生活的各个领域,成为大众化的工具了。计算机已不再是令人感到不可捉摸的“神秘之物”了。世纪之交在即,所有人都应该具有一定的计算机知识,并且能够使用计算机。

计算机的工作方式与我们日常用于数值计算的计算器不同。使用计算器的时候,输入数据和运算符号以后,计算器立即显示计算的结果,而计算机是按照程序所规定的内容和步骤进行工作的。每一个学习计算机知识的人,都应当建立“程序”的概念,并且学习程序设计的知识。

什么是程序?打个比方,我们开会,首先要制定一个“大会程序”,比如第一项“全体起立,高唱国歌”;第二项“通过主席团名单”;第三项“领导作报告”;第四项“举手表决某项提议”……由此可见,“程序”包括两方面的内容:一个是“需要做什么事”;另一个是“按照什么顺序进行”。那么,计算机程序就是为使计算机完成一个预定的任务而设计的一系列语句或指令。要让计算机实现一组操作,必须先编写程序,然后再使计算机执行此程序。

什么是“程序设计”?设计、书写及检查调试程序的过程就称为程序设计。程序设计包括:分析问题确定解题方法、写出具体操作步骤、用高级语言编写程序、上机调试程序、修改程序、运行程序、整理结果、书写程序文档等过程。

1.1 程序设计语言

语言可以分成两大类,即自然语言和人工语言。我们人类所使用的语言一般被称为自然语言,它们是人类在自身发展过程中为了表达思想、交流经验、互通信息而自然形成的语言。人工语言是人们为了到达某种目的而专门设计的语言。

计算机是人类科学技术发展历史上的一个伟大进步,可以看成是人脑的延伸,但是目前它还不能理解人类的自然语言,只能接受特定的机器代码。要想使计算机按照人的意愿去处理某件事情,就必须有人通过某种途径来生成计算机所能够操作的代码。计算机语言就是用来生成这些可以为计算机直接识别的代码的人工语言之一。

1.1.1 计算机语言的层次

计算机语言可以分成三个层次。

1. 机器语言

计算机基本的物理构成是复杂的电子线路。这些线路可以在不同的电信号作用下,通过各种装置产生不同的动作。计算机的核心部分(CPU、存储器等等)只能识别高、低电平两种状态,即二进制的0、1。由0、1的不同组合所形成的可以为计算机直接识别的二进制指令代码的集合就称为机器语言。计算机在接收到这些二进制代码后,由译码器产生各种控制信号送往有关的单元,从而产生相应的操作。

机器语言是最基本的、出现最早的计算机编程语言,是唯一可以为计算机直接执行的语

言。用机器语言编写的程序冗余小,执行效率高,节省内存,运行速度快,可以直接控制计算机的硬件。但是不同类型计算机的硬件结构可能有所不同,它们各自具有不同的指令系统,因此用机器语言编程对程序设计者的水平要求很高,他们必须对所使用计算机的硬件工作原理以及线路连接关系十分清楚,而且一般在某种机器上编写的程序不能在另一种机器上运行,因此人们称机器语言为“面向机器”的程序设计语言。此外,用机器语言编写的程序可读性差、容易出错,出错后很难查找,因此,人们往往不直接使用机器语言进行编程,而是借助于其他更为简单、编程效率更高的计算机语言,通过间接的方法来产生可以为计算机直接接受的代码。

2. 汇编语言

为了克服机器语言的缺点,便于一般人员进行编程,在 50 年代初出现了汇编语言,其特点是引入了一些助记符来表示操作,用一些特定的符号来表示计算机中的某些单元,例如用“ADD”表示加运算,用“AL”、“AH”表示某一寄存器等等。这些助记符比较接近英文,其他符号也比较简单,便于记忆、理解,从而简化了编程过程。

使用助记符和有关符号编写的程序称为汇编语言程序。由于计算机只能识别二进制代码,不能识别这些符号,因此,还必须通过某种方法将汇编语言程序“翻译”成相应的二进制代码,由这些二进制代码组成的程序称为目标程序,“翻译”的过程称为汇编。汇编的方法有人工汇编和机器汇编两种,人工汇编是由人根据机器的指令系统通过查表进行转换,机器汇编则是借助于专门的汇编程序由计算机来完成转换。

使用汇编语言进行编程也要求编程人员有丰富的计算机硬件知识,这是因为汇编语言仍然是一种“面向机器”的程序设计语言,编程过程离不开机器的硬件结构,助记符、符号及汇编过程都要依赖于机器的特定指令系统。所以,汇编语言程序的通用性仍然较差,应用范围受到一定的限制。但是汇编语言程序仍然具有运行速度快、能直接控制硬件等特点,在实时性要求较高的计算机自动控制系统中或者对计算机硬件设备要求特殊管理、编写系统软件的情况下,仍然得到了广泛的应用。

3. 高级语言

机器语言和汇编语言都是“面向机器”的程序设计语言,人们习惯上称它们为“低级语言”。由于它们难学、难记、程序难调试、可移植性差,所以在非计算机专业人员中不易推广,随着计算机的迅速普及和人们解决日益增加的实际问题的需要,出现了各种形式的高级语言。

高级语言是普及型的计算机程序设计语言,其各种命令的形式接近于自然语言(英语)和数学算式的格式,它们有着各自的特点,有着各自严格的语法规则,便于记忆、书写、阅读和修改。使用高级语言编写出的程序的每一条命令,从字面上就能看出其含义。高级语言基本上摆脱了机器类型的影响,程序设计者在进行程序设计时可以不考虑机器的硬件结构,只需要掌握应用问题的解决方法和有关的算法,按照语言的语法规则书写命令,就可以编出程序。

使用某种高级语言编写出来的程序称为该语言的源程序。计算机不能直接辨认用高级语言编写的程序中的指令,必须将高级语言程序“翻译”成计算机可以直接辨认的机器语言程序。然而,用人工进行这样的“翻译”,在实际上是不可能的。因此,人们在创造高级语言的时候,同时要编写出用计算机将高级语言程序“翻译”成机器语言程序的软件。这样的“翻译”软件就叫做高级语言的编译软件(程序)。在编辑和执行高级语言程序的时候都需要有该种语言的编译软件参与。例如:随着 MS-DOS 操作系统 3.0 版本发行的 BASICA.EXE 软件是老版本的

BASIC 语言编译软件;随着 MS-DOS 5.0 以及更高版本发行的 QBasic 语言是一种改进的 BASIC 语言的编译软件。

使用高级语言编程的方法和思路很接近我们人与人之间的自然语言交流和数学描述,因此编程效率高,编程的时候基本上不涉及计算机的硬件知识,便于普及,程序的通用性好。但是高级语言程序不如机器语言简练,翻译转换后生成的目标程序冗余大,运行时占用内存多,速度较慢。

1.1.2 常用高级语言简介

自 1954 年诞生了第一种高级语言——FORTRAN 语言以来,到目前为止已陆续出现了上百种计算机高级语言。高级语言的发展趋势是:标准化、模块化和结构化。这些高级语言各有特点,所侧重的应用领域也不同,除少数几种高级语言外,基本上是不同的领域使用不同的高级语言。比较常用、流行的高级语言主要有以下五种。

1. FORTRAN 语言

FORTRAN 语言是世界上最早出现的高级语言,它特别适合于科学和工程中的数值计算。它所支持的数据类型比较丰富,可以进行复数运算。除此之外,它的数据输入输出格式也很灵活。

2. COBOL 语言

COBOL 语言是专门为处理商业事务而设计的一种通用语言,适用于商业和管理领域。其特点是针对商务和管理过程中要处理大量记录形式的数据的特殊需要,侧重于数据项和输入输出记录的处理。

3. PASCAL 语言

PASCAL 语言是最早出现的结构化程序设计语言,它特别强调程序的可靠性和易于验证性。用它编写出来的程序结构性好,程序易读。这种语言广泛用于一些计算机课程的教学,也可用于科学计算。

4. C 语言

C 语言是一种通用的结构化程序设计语言,特别是在软件工程编写系统软件方面应用很广,它既有高级语言的诸多特点,也具有汇编语言控制硬件的能力,还可以比较方便地调用其他语言编写的程序,因此人们习惯把它称为“中级语言”,其应用日益广泛。随着 Microsoft Windows 的出现,近年来又出现了可视化的 Visual C 语言。

5. BASIC 语言

BASIC 语言是继 FORTRAN 语言之后推出的又一计算机高级语言。它是于 1964 年由美国 Dartmouth 学院两位计算机科学家专门为初学者设计的。BASIC 语言是一种易学易用、易推广又有较高使用价值的高级语言,特别适合于初学者使用,曾一度风靡世界。三十多年来,BASIC 语言与日新月异的计算机技术同步发展,不断推出新版本,但是它始终遵守原有的宗旨:为初学者着想。它的风格依旧,被誉为“BASIC 之树常青”。当前,能够在 Windows 环境运

行的 Visual BASIC,也已经成为“能迎接计算机世界挑战的最好例子”。

Microsoft 公司认为:“BASIC 语言由于已经具有较高的知名度,将仍然是熟知其专业程序设计人员、咨询专家和实力用户的首选语言。”

1.1.3 高级语言的执行方式

正如前面所述,使用高级语言编写出来的源程序是不能直接被计算机执行的,必须要通过特定的翻译程序将其转换成机器语言才能为计算机所执行。对高级语言源程序进行翻译的方式有两种,即编译执行方式和解释执行方式。

1. 编译执行方式

当源程序编辑好以后,利用高级语言系统自身所具备的编译工具(功能),一次将源程序统一翻译成机器语言程序,该程序称为源程序的目标程序,该过程称为编译,如果源程序有错误,则编译时给出所有的出错信息,不生成新的目标程序,只有当用户将全部错误改正以后,编译才生成最终的目标程序文件,该目标程序是由二进制代码组成的机器语言文件,可以被计算机执行。

编译执行方式有如下的特点:

- (1)一次将源程序翻译成目标程序后,该目标程序可以多次运行而无需再进行翻译。
- (2)编译方式所形成的目标程序的装载和运行速度比一般的解释方式要快。
- (3)编译后所形成的目标程序一般可以脱离相应的语言环境运行。
- (4)编译后的目标程序的运行不需要源程序,有利于源程序的保密。
- (5)不同语言环境下生成的目标程序(包括汇编得到的目标程序)可以互相调用,有利于充分发挥各种语言的长处。
- (6)编译过程中可以一次发现程序中的全部错误,并且修改方便(有些错误是互相关联的,可能修改一处就可以消除其他的一些错误)。
- (7)源程序有任何错误都不能生成目标程序。
- (8)源程序变动后必须重新编译才能生成新的目标程序。

由于编译方式具有很多优点,所以大部分高级语言都采用这种执行方式,例如 Quick BASIC、Turbo C 等等。

2. 解释执行方式

这种执行方式是直接运行源程序。运行的时候逐句将源程序翻译成机器语言调入,内存再加以运行,这个过程称为解释。解释一句,运行一句,不在磁盘上生成永久的目标程序,若程序中的某些语句有错误,则执行到出错处即停止运行,并给出出错的位置以及出错的类型(有些系统还给出出错的原因),便于及时修改。

解释执行方式的优点是不需要编译工具,故整个语言系统较小,出错前的程序可以正常运行,可以得到部分运行结果,便于用户调试,不足之处是由于解释一句运行一句,致使一般运行速度较编译方式要慢,每次运行程序都需要计算机进行解释,较大程序的调试比较麻烦。早期的 BASIC 语言都采用解释执行方式。QBasic 采用的是较为先进的解释方式,其环境下的运行速度与一般的解释方式相比是比较快的。