

河南省高校计算机基础教育系列教材

程序设计 与 算法基础

主编 张尧 段银田 杨秀金

FORTAN5.0 版



黄河水利出版社

河南省高校计算机基础教育系列教材

程序设计与算法基础

(FORTRAN 5.0 版)

主 编 张 尧 段银田 杨秀金

黄河水利出版社

内 容 提 要

本书以算法描述和程序设计方法为中心,在较系统地介绍算法描述和程序设计方法的基础上,讲述了从 FORTRAN 77 到 FORTRAN 5.0 的基本语法规则和程序设计特点。本书涉及到的算法有:查找和分类算法、一元非线性方程求根、一元函数插值、一元函数数值积分、线性代数方程组数值解法、矩阵的基本运算和变换算法等。通过对程序模块设计方法的介绍,强调了程序独立的重要性;通过数据文件概念的阐述和应用举例,强调了数据独立的重要性。本书还简述了 FORTRAN 屏幕绘图的基本方法,提供了函数、离散数据曲线绘制和在图形中加入文字注释的应用实例,丰富了 FORTRAN 的应用领域。可作为理工类本科学子程序设计课程的教学用书,也可作为工程技术人员和研究生的学习参考书。

图书在版编目(CIP)数据

程序设计与算法基础/张尧等主编. —郑州:黄河水利出版社,1999.1

河南省高校计算机基础教育系列教材

ISBN 7-80621-263-9

I. 程… I. 张… II. ①程序设计-高等学校-教材②电子计算机-算法理论-高等学校-教材 IV. TP3

中国版本图书馆 CIP 数据核字(98)第 40590 号

责任编辑:雷元静 杜亚娟

封面设计:谢 萍

责任校对:王才香

责任印刷:温红建

出版发行:黄河水利出版社

地址:河南省郑州市顺河路黄委会综合楼 12 层 邮编:450003

印 刷:黄河水利委员会印刷厂

开 本:787mm×1092mm 1/16

印 张:13.625

版 别:1999 年 1 月 第 1 版

印 数:1—3000

印 次:1999 年 1 月 郑州第 1 次印刷

字 数:340 千字

定 价:20.00 元

前 言

为了紧密结合当前计算机基础教育的特点,以及非计算机专业计算机基础教育第二层次的基本教学要求,我们在总结多年教学经验、吸取其他院校教学研究最新成果和探讨“面向 21 世纪”计算机基础教学发展方向的基础上,组织了本教材的编写工作。在安排编写内容时,考虑到多数院校都将“微机应用基础”课程作为计算机基础教育的第一门课程,当学习“程序设计”课程时学生已有了较好的计算机知识基础,本书采用了较高的起点;考虑到开设程序设计课程的目的在于培养学生应用计算机服务专业课程学习的能力,包括编程处理较复杂的习题、完成专业课程设计和毕业设计中的计算等,我们强调了以算法设计思路分析和程序设计方法的学习为主线,列举并分析大量的实用性程序,引导学生在程序分析和设计中学习算法语言,在上机实践中理解语法规则。

依据上述的编写指导思想,本书以算法描述和程序设计方法为中心,在较系统地介绍算法描述和程序设计方法的基础上,讲述了从 FORTRAN77 到 FORTRAN5.0 的基本语法规则和程序设计特点。本书涉及到的算法有:查找和分类算法、一元非线性方程求根、一元函数插值、一元函数数值积分、线性代数方程组数值解法、矩阵的基本运算和变换算法等。并通过对程序模块设计方法的介绍,强调了程序独立的重要性;通过对数据文件概念的阐述和应用举例,强调了数据独立的重要性。本书还简述了 FORTRAN 屏幕绘图的基本方法,提供了函数曲线、离散数据曲线绘制和在图形中加入文字注释的应用实例,丰富了 FORTRAN 的应用领域。

郑州轻工业学院张尧副教授、郑州工业大学段银田教授和郑州粮食学院杨秀金副教授担任本书的主编,负责全书的统稿和定稿工作。张尧编写了第七、八章和附录部分,段银田编写了第六章,杨秀金编写了第一、四章。另外,郑州轻工业学院张建伟编写了第二、五章,郑州工业大学张兰编写了第三章。在本书编写过程中,得到了各校教研室和河南省高等院校计算机基础教育研究会的多方帮助,在此一并表示衷心的感谢。对于书中的不当之处,敬请各位读者指正,共勉提高。

编 者

1998 年 10 月

目 录

第一章 计算机基础知识.....	(1)
1.1 计算机系统的组成	(1)
1.1.1 计算机硬件系统	(1)
1.1.2 计算机软件系统	(2)
1.2 计算机语言和语言处理程序	(3)
1.2.1 计算机语言	(3)
1.2.2 计算机语言处理程序	(3)
1.3 FORTRAN 语言的发展和应用特点	(4)
1.3.1 FORTRAN 语言的发展	(4)
1.3.2 Microsoft FORTRAN 5.0 对 FORTRAN 语言的新发展	(5)
1.3.3 FORTRAN 语言的应用特点	(6)
1.4 MS-FORTRAN 程序开发环境	(6)
1.4.1 MS-FORTRAN 集成开发环境 PWB	(6)
1.4.2 DOS 命令方式	(10)
1.4.3 CodeView 动态调试程序	(11)
第二章 算法和程序设计概要	(13)
2.1 算法和算法的描述方法	(13)
2.1.1 算法的概念	(13)
2.1.2 算法描述	(14)
2.2 FORTRAN 源程序的基本构成	(20)
2.2.1 FORTRAN 程序举例	(20)
2.2.2 FORTRAN 程序单位的构成、分类和书写规则	(21)
2.3 FORTRAN 程序中的基本量、表达式和标准函数	(22)
2.3.1 常量和变量	(22)
2.3.2 数据类型及变量类型的定义	(24)
2.3.3 数组和下标变量	(26)
2.3.4 基本内部函数	(27)
2.3.5 表达式	(29)
第三章 基本程序结构设计	(32)
3.1 顺序结构设计	(32)
3.1.1 计算赋值语句、DATA 赋值语句、END 语句和 STOP 语句	(32)
3.1.2 自由格式输入和固定格式输出	(34)
3.1.3 格式输入/输出	(37)
3.1.4 语句函数的定义和引用	(42)

3.2 选择结构设计.....	(44)
3.2.1 关系运算和逻辑运算.....	(44)
3.2.2 逻辑 IF 语句、块 IF 语句和 IF 算术语句.....	(45)
3.2.3 多分支结构设计.....	(47)
3.3 循环结构设计.....	(52)
3.3.1 DO 语句, CONTINUE 语句和 END DO 语句.....	(52)
3.3.2 DO WHILE 循环和 UNTIL 循环.....	(55)
3.3.3 循环的嵌套设计.....	(59)
3.3.4 在 I/O 语句中的隐式 DO 循环.....	(61)
3.3.5 循环结构内的控制转移语句 EXIT 和 CYCLE.....	(62)
第四章 数组的应用	(65)
4.1 数组的存储结构.....	(65)
4.1.1 数据和数据的逻辑结构.....	(65)
4.1.2 FORTRAN 语言中数组的存储结构.....	(65)
4.2 数组元素赋值和输出.....	(66)
4.2.1 在 I/O 语句中使用数组名.....	(66)
4.2.2 利用 DO 循环进行数组元素的赋值.....	(67)
4.2.3 在 I/O 语句和 DATA 语句中利用隐式 DO 循环 进行数组的输入和输出.....	(67)
4.2.4 利用随机函数过程和 DATA 语句给数组赋初值.....	(68)
4.2.5 给数组赋特定初值的操作举例.....	(69)
4.3 数组元素间和数组间的简单运算.....	(71)
4.3.1 同一数组的数组元素间的简单运算.....	(71)
4.3.2 不同数组间的简单运算举例.....	(72)
4.4 查找和排序算法.....	(73)
4.4.1 一维数组的查找算法.....	(73)
4.4.2 一维数组的排序算法.....	(75)
4.4.3 二维数组的排序算法.....	(80)
4.4.4 二维数组的查找算法.....	(81)
4.5 字符运算和字符数组的应用.....	(83)
4.5.1 字符运算、字符函数及其应用.....	(83)
4.5.2 利用字符数组绘制函数的近似图形曲线.....	(86)
4.6 数组的一些其他运算和 FORTRAN 5.0 数组的运算表达式.....	(88)
4.6.1 一维数组元素的删除操作.....	(88)
4.6.2 两个一维数组的数组元素的归并处理.....	(89)
4.6.3 FORTRAN 5.0 的数组表达式举例.....	(90)
第五章 FORTRAN 语言程序中的过程设计	(93)
5.1 过程的基本概念和分类.....	(93)
5.2 函数过程设计.....	(94)

5.2.1	外部函数过程的构成	(94)
5.2.2	外部函数的定义及引用举例	(95)
5.3	例行子程序	(98)
5.3.1	外部函数和例行子程序间的两点重要差别	(98)
5.3.2	例行子程序的定义和调用方法	(98)
5.4	程序单位之间参数传递方法的比较和总结	(100)
5.4.1	两种基本的参数传递方式	(100)
5.4.2	变量作形参	(101)
5.4.3	数组作形参	(102)
5.4.4	过程名作形参	(107)
5.5	在过程中的 SAVE 语句和 DATA 语句	(110)
5.6	FORTRAN 语言中专用的数据通信语句及数据置初值	(111)
5.6.1	等价语句(EQUIVALENCE)	(111)
5.6.2	共用语句(COMMON)	(112)
5.6.3	数据块子程序(BLOCK DATA)	(114)
第六章	常用数值算法及程序设计	(118)
6.1	一元非线性方程求根	(118)
6.1.1	迭代法	(118)
6.1.2	牛顿切线法	(121)
6.1.3	弦位迭代法	(123)
6.1.4	加速迭代收敛的 δ^2 法	(124)
6.1.5	二分法	(126)
6.1.6	几种方法的评价	(127)
6.2	一元函数插值	(128)
6.2.1	一元函数插值简介	(128)
6.2.2	拉格朗日插值多项式	(129)
6.2.3	用低次多项式分段插值	(132)
6.3	一元函数数值积分	(134)
6.3.1	方法的基本思想	(134)
6.3.2	矩形法	(135)
6.3.3	梯形法	(136)
6.3.4	辛普森法	(138)
6.3.5	三种近似积分法的比较	(140)
6.4	线性代数方程组数值解法	(140)
6.4.1	高斯-若当消去法	(141)
6.4.2	列主元素消去法	(144)
6.4.3	全主元素消去法	(146)
6.4.4	追赶法	(149)
6.5	矩阵的运算和变换	(152)

6.5.1	矩阵的加减运算	(152)
6.5.2	两矩阵的乘法运算	(153)
6.5.3	求逆矩阵	(154)
6.5.4	求行列式的值	(157)
第七章	FORTRAN 数据文件应用简介	(159)
7.1	文件的概念	(159)
7.1.1	文件的组织和分类	(159)
7.1.2	有关文件操作的术语	(159)
7.2	外部文件操作的基本步骤和基本语句	(160)
7.2.1	外部文件操作的三个基本步骤	(160)
7.2.2	外部文件操作语句和函数	(161)
7.3	数据文件应用举例	(164)
7.3.1	有格式顺序文件的操作和应用举例	(165)
7.3.2	有格式随机文件的操作和应用举例	(169)
7.3.3	无格式文件的操作和应用举例	(174)
7.3.4	文件应用举例小结	(177)
第八章	FORTRAN 5.0 屏幕绘图的基本方法	(178)
8.1	FORTRAN 5.0 的文件包含和复合数据类型定义	(178)
8.1.1	文件包含	(178)
8.1.2	结构、联合体和记录类型	(178)
8.2	屏幕绘图的基本概念	(180)
8.2.1	绘图函数和子程序	(180)
8.2.2	图形模式及设置图形模式	(180)
8.2.3	绘图坐标系	(181)
8.3	基本绘图操作的实现	(182)
8.3.1	绘图颜色、背景颜色、线型等的设定	(183)
8.3.2	点和直线的绘制及文字的加入	(184)
8.3.3	矩形、圆和圆弧的绘制	(186)
8.3.4	图形区域的填充和其他操作	(187)
8.4	绘图应用程序综合举例	(187)
附录 A	Microsoft FORTRAN 5.0 中常用的内部函数和附加过程	(195)
附录 B	7 位 ASC II 代码表	(199)
附录 C	FORTRAN 5.0 常用绘图函数和子程序	(200)
附录 D	FORTRAN 5.0 错误信息及改正措施说明	(204)

第一章 计算机基础知识

电子计算机自 1946 年问世以来,以迅猛的速度发展着,现在,计算机的应用已延伸到社会各个领域。每一个科技人员和管理人员,都应具有计算机基础知识和应用计算机的能力。为使用好计算机,必须首先了解和掌握计算机的基础知识。

1.1 计算机系统的组成

一个计算机系统由硬件系统和软件系统组成。如图 1-1 所示。

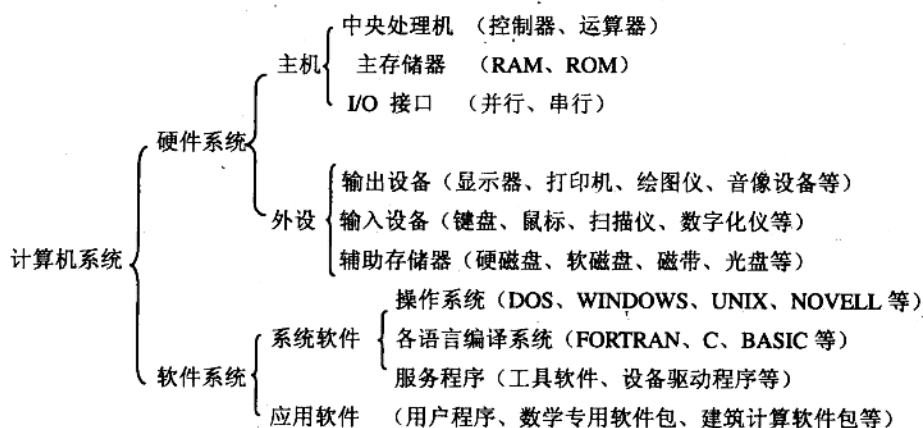


图 1-1 计算机系统的组成

1.1.1 计算机硬件系统

计算机的硬件系统由五大功能部件组成,即运算器、控制器、存储器、输入设备和输出设备。它们之间的关系如图 1-2 所示。

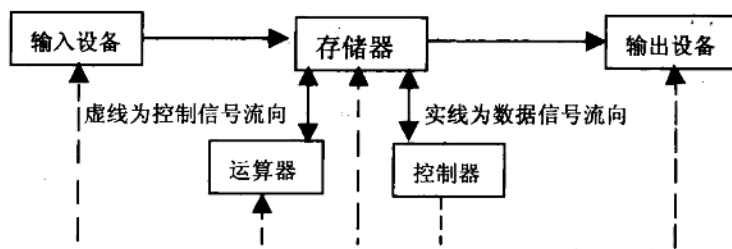


图 1-2 计算机硬件组成示意图

下面我们分别简要介绍各部件的主要功能。

1.1.1.1 运算器

运算器是用来对各种数据进行算术运算和逻辑运算的部件。当需要进行某种运算时,由控制器发出命令,将存放在内存储器中的数据送到运算器中,在运算器中进行计算,然后将计算结果送回到内存储器中存起来,以便下次运算时使用或把它输出出来。

1.1.1.2 控制器

控制器是用于控制计算机各个部件协调工作的一种装置,是计算机的“神经中枢”。控制器根据事先存入计算机的指令向运算器,存储器,输入、输出设备发出控制信号,使其按要求工作。

控制器和运算器是计算机硬件的核心,合称中央处理器,简称 CPU。

1.1.1.3 存储器

存储器是计算机的记忆装置,用来存放数据和程序。存储器分为主存储器和辅助存储器两类。其中主存储器又称内存储器,简称内存;辅助存储器又称外存储器,简称外存。

一个存储器分为许多存储单元,被存储的信息按信息单位分别存放在这些存储单元内,每个单元都有自己的编号,称为该单元的“地址”。每个存储单元包含若干个二进制位,每个二进制位称为一个位(Bit)。为便于管理,将 8 个位称为一个字节(Byte),把计算机运算器一次能处理的数据的位数称为该计算机的字长,字长一般是字节的整倍数。

内存储器用来存放当前运行所需要的程序和数据,也存放计算结果和中间结果,它可与 CPU 直接交换信息。早期的计算机用的是磁芯存储器,现在的计算机大多采用半导体存储器。内存储器分为随机存取存储器(RAM)和只读存储器(ROM)两种。随机存取存储器又称读写存储器,可随时读出和写入信息。但当主机电源被切断时,RAM 中的信息会丢失。只读存储器则只能读出,不能写入,即使关闭电源,ROM 中的信息也不会消失。因此,常用 ROM 存放固定的程序和数据。存储器的容量一般按字节计算,简记为 B。并称 1024B 为 1KB,称 1024KB 为 1MB,称 1024MB 为 1GB。微型计算机内存储器的容量一般为 1MB~64MB。

外存储器用来存放计算机中暂不参与运行的程序和数据,不直接与 CPU 交换信息,但可与内存成批地交换信息。常用外存有磁盘、光盘和磁带。目前硬盘的容量一般为 1GB~6.4GB,光盘的容量一般为 640MB。

1.1.1.4 输入设备

计算机的输入设备用来向计算机内部输入程序、数据、命令及其他控制信息。输入设备有键盘、外存储器、鼠标、模数转换器等。目前微型计算机中普遍使用键盘和鼠标输入信息。

1.1.1.5 输出设备

输出设备用来将计算机内的程序和运算结果等信息展示给人们。常用的输出设备是显示器、打印机和外存储器。

输入、输出设备通过 I/O 接口电路与主机相连。

1.1.2 计算机软件系统

如果计算机只有硬件,那它什么事也干不成,要想使计算机完成多种多样的信息处理任务,必须配备各种各样的程序系统。每种程序系统连同它们的文档统称为软件系统。可以说,软件是程序与相关数据和文档的总称。按照其在计算机系统中的作用和地位,软件分为系统软件和应用软件两大类。

1.1.2.1 系统软件

系统软件是便于用户使用计算机、充分发挥计算机的功能而提供的基础性的支撑软件。如操作系统、编译系统、解释系统、汇编系统等。其中操作系统是核心,它相当于一个总调度,它管理着整个计算机系统的资源,使计算机系统的软、硬件资源得到充分合理的应用。它是人和机器的一个接口,人们可通过它所提供的一系列命令来操纵计算机系统的运行。

1.1.2.2 应用软件

应用软件是指专为计算机在某个特定领域中的应用而开发的软件。例如办公自动化软件(OA)、计算机辅助设计软件(CAD)、计算机辅助教学软件(CAI)、管理信息系统(MIS)等。

1.2 计算机语言和语言处理程序

1.2.1 计算机语言

能让计算机直接或间接接受与识别的语言,就是计算机语言。

计算机语言可分为三大类,即机器语言、汇编语言和高级语言。机器语言与汇编语言都是依赖于具体机器的语言,使用不便,不易推广。人们也常把机器语言和汇编语言称为低级语言。高级语言是面向过程和对象的语言,即在用高级语言编写程序时,不必再考虑机器的内部构造和不同机器的特点,只要按照描述问题的算法和语言的规定写出解题过程(程序),计算机就能执行。用这种语言描述问题的过程如同人们叙述一件事情,容易被理解,它在计算机的推广应用发挥着巨大的作用。

高级语言种类很多,其中应用广泛的有以下几种:

(1)FORTRAN(Formula Translation)语言。该语言是最早问世的高级语言。自1954年以来被不断地完善和发展,至今仍被广泛地应用于科学研究和工程设计领域。

(2)BASIC语言。该语言以其易学、易用和人机交互式功能强等特点成为应用最广的高级语言。自1960年以来,已有多次更新,True BASIC、Quick BASIC等都是1987年以后发表的BASIC语言的新版本,既适用于数值计算又适用于非数值处理。

(3)C语言。它是70年代为表达UNIX操作系统而诞生的一种高级语言。既可方便地用于开发系统软件,又可用于开发应用软件。

以上均属“过程化语言”,用它解决问题,必须详细描述解题步骤。人们更希望一种只要向计算机说明要解决的问题,而不需描述“怎么做”的语言。能实现人们这种愿望的语言被称为“非过程化语言”。具有非过程化语言特征的高级语言C++,已在推广应用,完全的非过程语言在进一步研究与发展中。

1.2.2 计算机语言处理程序

虽然用高级语言编写程序比用低级语言方便得多,但用高级语言编写的程序计算机无法直接识别,必须先把它翻译成计算机能识别的机器代码,即机器语言程序。实现这个翻译工作的程序,称为语言处理程序,一般包括编译程序和解释程序。

编译程序要完成的工作包括词法分析、语法和语义分析、生成目标程序和优化目标程

序。完成这些工作,一般要分为翻译和连接两个步骤。翻译时实现翻译和查错,当发现有语法错误时,输出出错信息,用户可据此修改源程序,然后再次编译,直到语法、语义正确,才产生目标块程序(.OBJ)。连接时则由装配连接程序将翻译得到的目标块程序和系统所提供的函数库、过程库等连接在一起,形成可直接被计算机执行的目标程序。编译方式下处理并执行一个程序的具体过程见图 1-3。

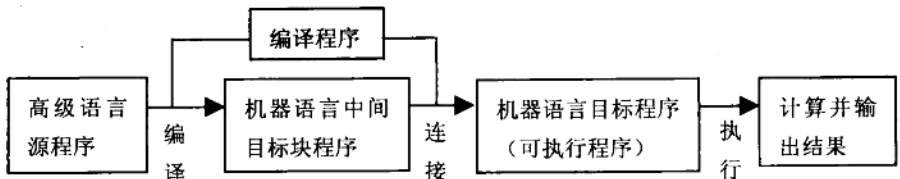


图 1-3 编译程序工作过程示意图

解释程序也是将高级语言程序翻译成机器指令,但它与编译程序不同,它不是先产生出目标程序再交付执行,而是边翻译边执行。可以理解为翻译一句执行一句。若翻译执行过程中遇到错误,需进行修改后再解释执行。若无错误,则翻译并执行到程度结束。比如有一个源程序,共 15 行,第 10 行有错,则前 9 个句子都能边解释边执行,直到执行第 10 句时才会发现错误,停止执行,并显示出错信息。用户修改此句后,需重新执行该程序。这种翻译方式的工作过程见图 1-4。

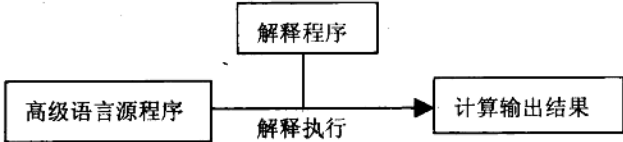


图 1-4 解释程序工作过程示意图

由上述可以看出,编译方式如同笔译,一次翻译出整个译文;而解释方式如同口译,说一句译一句,执行一句。编译方式产生独立的可执行文件,能脱离原语言环境单独执行,且运行速度快,但对程序的调试操作不够方便;解释方式下执行程序必须有具体的语言处理程序环境的支持,且速度较慢,但调试程序较为方便。FORTRAN 语言程序使用编译方式,BASIC 语言程序多采用解释方式。目前有些语言可选择使用解释或编译方式。

1.3 FORTRAN 语言的发展和应用特点

1.3.1 FORTRAN 语言的发展

FORTRAN 语言是国际上广泛使用的一种计算机高级语言。50 年代初,以 IBM 公司的 J. Backus 为首的研究小组,致力于研究开发适用于科学家和工程师使用的高级语言。到 1954 年 4 月,该语言研究成功,被命名为 FORTRAN(即 FORTRAN I)。该语言 1956 年被

正式使用。J. Backus 也因此于 1977 年获得图灵奖。

自 FORTRAN 语言问世以来,先后推出了不同的版本。其中影响较大的有以下几个版本:

1966 年美国国家标准化协会(American National Standard Institute,简称 ANSI)公布的美国标准文本:标准 FORTRAN(X3.9-1966)。

1972 年国际标准化组织(International Standard Organization,简称 ISO)公布的标准文本:程序设计语言 FORTRAN ISO 1539-1972,俗称 FORTRAN 66。

1978 年公布的标准文本:ANSI X3.9-1978 FORTRAN,俗称 FORTRAN 77。

FORTRAN 77 在许多方面对 FORTRAN 66 作了改进。如增加了“块 IF”语句,使其能方便地用于结构化程序设计;增加了字符型变量和字符型数组,允许字符串的运算,使其也能用于非数值运算领域;增强了输入、输出的功能;允许不同类型数据混合运算;数组下界可以等于或小于零;下标表达式可以为任意的整型表达式等。同时,FORTRAN 77 和 FORTRAN 66 还有较高的兼容性,即用 FORTRAN 66 编写的源程序可以不加修改或稍加修改就能在 FORTRAN 77 编译系统支持下运行,保持了 FORTRAN 语言的继承性。

1980 年,FORTRAN 77 被接受为国际标准,即程序设计语言 FORTRAN ISO 1539-1980,该标准分为全集和子集。一般微型计算机上配置的 FORTRAN 77 仅为子集,即不具备 FORTRAN 77 的某些功能,建议读者在使用时先了解一下所用计算机上 FORTRAN 77 子集的功能。

我国 1983 年公布了 FORTRAN 国家标准,基本采用了国际标准,标准号为 GB3057-82。

在计算机软件研制开发领域内,新标准的制订总是落后于产品的开发和应用。尽管 FORTRAN 77 之后,又产生了新的国际标准 FORTRAN 90。但和它相应的软件产品同时又对其进行了取舍和扩充。Microsoft FORTRAN 5.0 就是其中的一个具有代表性的产品。

1.3.2 Microsoft FORTRAN 5.0 对 FORTRAN 语言的新发展

在与已有的 Microsoft FORTRAN 语言兼容的基础上,FORTRAN 5.0 在如下几个方面增加或增强了 FORTRAN 语言的功能:

- (1)第一次提供了图形库,支持丰富的绘图操作和字体输出操作。
- (2)支持所有的 IBMSAA 扩展和许多的 VAX 扩展。
- (3)既可以在 DOS 环境下运行,又可以在 OS/2 环境下运行。
- (4)支持条件编译和多种语言混合编程。
- (5)提供了新的数据结构——由程序员定义的复合数据类型(结构体和记录)。
- (6)新增了 SELECT CASE、DO WHILE、EXIT、CYCOL 等结构控制语句。

(7)提供了和 C 语言集成程序调试环境完全相似的程序调试环境,支持程序的源程序级调试。但基本系统(包括绘图)仅占用 1.3MB 的硬盘空间,显得非常小巧方便。

本书将以 FORTRAN 77 为基础,适当地插入介绍了 FORTRAN 5.0 新增的结构控制语句的功能和应用方法。

1.3.3 FORTRAN 语言的应用特点

FORTRAN 语言主要应用于数值计算。它几乎统治了所有的数值计算领域,许多大型科学计算程序都是用 FORTRAN 语言编写的。如著名的 SAP5 软件包和大型结构分析程序 ADINA/ADINAT。其中,ADINA/ADINAT 程序是在美国麻省理工学院(M. I. T)K. J. Bathe 教授领导下,总结 SAP NONSAP 程序的编制经验,结合当前有限元计算的最新发展研制而成的,是国际上著名的有限元分析程序。它属于专利程序。1981 版本连绘图程序在内,共有 8 万多条 FORTRAN 语句。能处理结构的线性和非线性静力与动力分析问题,瞬态和稳态温度场问题,热弹塑性分析、液体与结构相互作用等问题。这一程序与 SAP 程序相配合,能解决机械、土建、造船、水利、动力、核能等许多部门的大型结构分析计算问题。又如,数学软件库 STYR/MATH 全部采用可移植的 FORTRAN 语言 PFORT 编写,并有 FORTRAN N 和 FORTRAN 77 两种版本。STYR/MATH 库以数学为主,包括求解常微分方程、计算矩阵特征值和特征向量、快速富里叶变换等等,适应于较广泛的科技计算。由此看来,用 FORTRAN 解决数值计算领域的问题十分方便。

由于早期的 FORTRAN 语言是专为数值计算而设计的,因此不适用于非数值处理。而 FORTRAN 5.0 则使 FORTRAN 语言不但能应用于非数值处理领域,也能应用于对绘图有要求的计算机辅助教育和计算机辅助设计等领域。

1.4 MS-FORTRAN 程序开发环境

1.4.1 MS-FORTRAN 集成开发环境 PWB

MS-FORTRAN 5.0 和当前流行的其他高级语言一样,也有一个集成开发环境,即本系统的程序员工作台 PWB(Programmer's WordBench)。在 PWB 中可以独立完成 FORTRAN 语言程序文件的编辑、编译、连接和调试运行等全部工作。PWB 还带有联机帮助功能(Help),可以随时查询有关命令、术语、菜单功能等信息,它是一个有效的程序开发辅助工具。

1.4.1.1 PWB 的启动和退出

在 DOS 提示符 ">" 下键入 PWB 和回车键,即可启动集成开发环境。此时显示主屏幕如图 1-5 所示。第一行是主菜单区,第二行是标题行,底行是提示行。

在第一次进入 PWB 环境时,一般光标停在文本区左上角,这时可以直接键入程序文本。但是此时文本没有标题(文件名)而在标题行显示 NUTITLE。完成文本输入后,选择主菜单中的 File 项的下拉子菜单中的 SAVE 命令,在对话框提示下输入程序文件名,例如键入 EXPROG.FOR,这时程序就以此名存盘。文本区上方的 NUTITLE 也换成相应的文件名。如要退出 PWB,选择主菜单中的 File 项的下拉子菜单中的 Exit 命令。这时系统将控制权交还 DOS,显示 DOS 提示符 ">"。

1.4.1.2 FORTRAN 源程序的输入和编辑

(1)打开或创建程序文件。在 PWB 环境中,可以用主菜单的 File 项中的子命令 New 来创立一个新的源文件,或用 Open 子命令来打开一个已经建立的程序文件。无论以何种方式

File	Edit	View	Search	Make	Run	Option	Browse	Help
NUTITLE								
用户文本编辑区								
Compile Result								
编译信息区								
<General Help> <F1=Help> <Alt=Menu>							Psendo 00001.001	

图 1-5 PWB 主屏幕示意图

打开一个文件后，屏幕中心的文本区均成为文件的编辑窗口，在这个区域内可以键入、修改、删除一段文本。最简单的编辑方式是移动光标到预定的位置，键入要增加的内容，新字符就显示在光标所在的位置。编辑缺省的状态为插入字符状态，即将新字符插入到光标所在的字符位置之前，而原光标之后的字符自动后移。这时如果按 Ins 键，编辑系统将由插入转为改写，即原光标之后的字符不移动，而以新字符代替原字符。无论是在插入还是在改写状态，按键盘上方的 $\leftarrow$ 键，均为删除光标前的一个字符，而按 Del 键为删去光标后的一个字符。在删除一段字符后，光标后的字符将依次向前移动，填补删去的空位。

(2) 成段文本的编辑。成段文本的选择是编辑中常做的工作，可以用 Shift 加方向键 ($\leftarrow$, $\rightarrow$) 来选择一段文本，即按住 Shift 再用方向键来移动光标，或按住鼠标左键并拖动光标来选择文本，这时所选定的内容将反白显示。在主菜单的 Edit(编辑)项的下拉子菜单中还提供另外两个命令(Set Anchor 和 Select to Anchor)进行成段文本的选择。将光标放在要选择的段落开头一端，用 Set Anchor 命令来设置一个段首标记，然后移动光标到文本段的另外一端，用 Select to Anchor 命令来设置段尾标记，就将两次光标定位之间的文本全部选中了。这种方法较适合于选择一段要跨越几屏的大段落。

在选择一段文本后，Edit(编辑)下拉子菜单中的命令 Cut、Copy、Clear 可以对选择好的段落进行操作。Cut 命令将选择的文本裁剪下来，放入文本缓冲区中；Copy 命令则只是将选中文本复制到缓冲区内，而原文件内容不变；Clear 命令正相反，全部删除这段文本，并且不向缓冲区送入任何内容。如果要删除所选文本段落，也可以直接敲击键盘上的 Delete 键，如果要将所选文本段落移到或复制到某一位置，则先将光标移到待插入的位置，然后用 Paste 命令将缓冲区内的内容(用 Cut 或者 Copy 命令剪辑的文本内容)复制到光标所在的文件位置。在用 Paste 将缓冲区的内容粘贴到文件中之后，缓冲区的内容仍然不变。因此，剪辑的同一内容可以多次复制到文件的不同地方或不同的文件内部。

(3) FORTRAN 源程序存盘。源文件的编写应该特别注意书写格式。按标准 FORTRAN 的规定，源文本区在 7~72 列内，1~6 列专门用于标号或续行标志，72 列以后用作注释。而 MS-FORTRAN 5.0 支持类似 Pascal 和 C 程序的自由格式的源文件，这时源文件

的书写可以不受列控制的限制,有助于编写更易于阅读的程序。但要在程序的开头加上元命令 \$FREEFORM。

建立或修改过的源程序应随时存盘,可选择 File 下拉子菜单中的 Save 或 Save As 来保存编辑好的源程序文件。假设用户源程序命名为 EXPROG.FOR,那么磁盘上就生成一个名为 EXPROG.FOR 程序文件。

1.4.1.3 源程序的编译、连接和运行

(1)初步编译,语法检查。编辑完成的源程序文件可以在 PWB 下进行编译和连接。在 PWB 主菜单中有 Make 和 Run 两项提供编译和连接的功能。Make 下拉子菜单中的 Compile File 命令可单独对活动窗口中的源程序进行编译。选择该命令后,PWB 调用带参数(/C)的驱动程序 FL.EXE 来对当前源程序进行编译,完成对源程序的语法检查,在正常编译结束后,系统将显示相应的编译结果信息,同时生成后缀为 .OBJ 的目标文件。对于 EXPROG.FOR 源程序文件编译,生成的目标文件为 EXPROG.OBJ。如果程序存在语法错误,系统将显示出错信息,提示用户修改源程序,并不生成目标文件。用户通过主菜单的 Edit 项,重新进入活动窗口对源程序进行修改编辑,然后存盘,重新编译,直到编译通过为止。

(2)在 Make 之下编译连接生成 EXE 可执行文件。在 PWB 中也可以不区分编译和连接阶段,而是将二者合成一步,此步称为生成(Build)。调用系统程序 FL.EXE 来编译、连接,最终建立可执行文件(*.EXE)。首先应使被编译源程序文件成为当前活动窗口中的内容,然后选择主菜单 Make 项的下拉子菜单中的 Build 命令,这时当前文件名出现在生成命令 Build 之后,并有 .EXE 后缀。选择该命令开始编译和连接,屏幕中心出现一个工作区,显示当前正在调用的编译命令。在完成编译和连接之后,出现一个对话框,如图 1-6 所示。

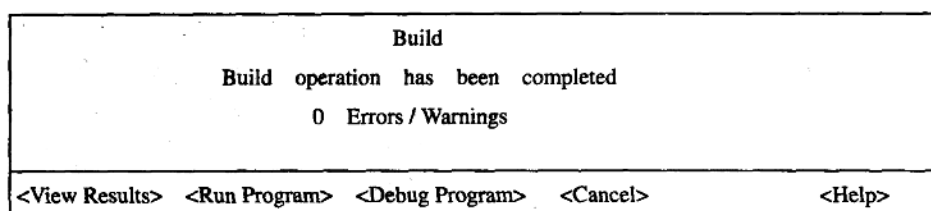


图 1-6 编译连接成功后的对话框

可以在对话框中选择<View Results>来查看编译结果信息,包括编译错误或警告的提示内容。如果如图所示(0 Errors/Warnings),程序没有错误,此时以 .EXE 为后缀的可执行文件在磁盘上生成,对于 EXPROG.FOR 源程序文件来说,生成的可执行文件即是 EXPROG.EXE。然后可选择<Run Program>立即运行该程序。如果系统检查出错误,则需修改程序,重新编译连接。在顺利生成一个可执行文件后,还可以主菜单 Run 项的下拉子菜单中选择运行程序命令 Execute 来运行这个可执行文件。

(3)在 Run 之下运行程序。在 Run 子菜单之下用 Execute 命令可以运行当前已生成的可执行文件。如果当前窗口中是新建立的源程序,过去没有进行过编译,Execute 命令启动后系统找不到对应的可执行文件,则它将首先显示如图 1-7 所示的对话框。

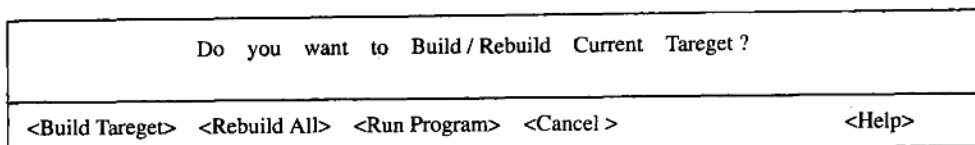


图 1-7 执行程序对话框

这时可选择生成命令<Build Target>,自动进行编译、连接,并生成可执行程序 EX-PROG. EXE,然后返回图 1-6 对话框,再转去执行该程序。

决定 PWB 编译、连接条件的参数是由主菜单 Option 下拉子菜单的 FORTRAN Compiler Option 命令决定的,其中 Memory Model 选择编译时采用的内存模式,FORTRAN Library 选用所使用的库类型,Language 决定源程序采用的语言版本。

在完成编译或发生编译错误退出编译过程后,除了可以在如图 1-6 对话框上选择<View Results>来查看编译信息外,还可在主菜单 View 下拉子菜单中选择 Compile Result 命令来进入编译信息窗口,如果编译成功显示正常编译\连接后的信息窗口。在发生编译错误的情况下,在显示内容之后还会加上若干条错误或警告信息说明行。其中前面是错误类型,括号()之中的数字表示发生错误的行号,后边是错误性质的简要说明。可以用主菜单 Search 下拉子菜单中的 Set Error 命令从错误信息窗口返回到引起该错误的源文件的出错行首,以便进行改正。这方面的详细资料请读者查阅有关参考书。

1.4.1.4 多模块程序的编译连接

对于大型的程序开发,比较好的方法是将程序分为若干个功能模块,每个块分别形成一个文件,称为模块文件。对各模块分别进行编辑,最后再把各模块合成一个完整的程序。将多个模块文件合成一个程序的方法很多,一种是直接拷贝成一个大文件,这样生成的源文件太大,不便于阅读和修改;另一种是对各模块分别编译后再连接起来,这是常用的方法。PWB 还提供了一种比较好的多模块组织技术,即用一個称为项目的文件来组织各个相关模块,其操作过程如下:

设一个大型程序包括以下各个模块文件:FA. FOR, FB. FOR, ..., FF. FOR。首先,建立一个以.MAK 为扩展名的项目文件,其中应包括以下各程序行:

FA. FOR

FB. FOR

.....

FF. FOR

具体方法是在主菜单 Make 下拉子菜单中选择 Set Program List 命令,在弹出的对话框的 File. Name 域内键入以 .MAK 为扩展名的文件名,例如,MAIM. MAK,并用<OK>保存。此时该文件还是个空文件。然后,在 Make 的下拉子菜单中选择 Edit Program List 命令,在弹出的对话框的 File Name 域内再次键入项目文件名 MAIN. MAK,再在 File List 子框内选择 F1. FOR, ..., F5. FOR,这样将各模块文件加入到 Program List 子框中,再用<Save List>来保存生成的项目文件 MAIN. MAK。