

Mc
Graw
Hill
Education

中国矿业大学图书馆藏书



C02342824

Fortran程序设计

(第四版)

```
FUNCTION vector_div_real(vec_1, real_2)
  TYPE(vector), INTENT(IN) :: vec_1
  REAL, INTENT(IN) :: real_2
  vector_div_real%x = vec_1%x / real_2
  vector_div_real%y = vec_1%y / real_2
  vector_div_real%z = vec_1%z / real_2
END FUNCTION vector_div_real

FUNCTION vector_div_int(vec_1, int_2)
  TYPE(vector) :: vector_div_int
  TYPE(vector), INTENT(IN) :: vec_1
  INTEGER, INTENT(IN) :: int_2
  vector_div_int%x = vec_1%x / REAL(int_2)
  vector_div_int%y = vec_1%y / REAL(int_2)
  vector_div_int%z = vec_1%z / REAL(int_2)
END FUNCTION vector_div_int

FUNCTION dot_product(vec_1, vec_2)
  REAL :: dot_product
  TYPE(vector), INTENT(IN) :: vec_1, vec_2
  dot_product = vec_1%x*vec_2%x + vec_1%y*vec_2%y &
    + vec_1%z*vec_2%z
END FUNCTION dot_product

FUNCTION cross_product(vec_1, vec_2)
  TYPE(vector) :: cross_product
  TYPE(vector), INTENT(IN) :: vec_1, vec_2
  cross_product%x = vec_1%y*vec_2%z - vec_1%z*vec_2%y
```

Stephen J. Chapman 著
王志强 李浩亮 等 译

Mc
Graw
Hill
Education



中国电力出版社
CHINA ELECTRIC POWER PRESS

Fortran程序设计

(第四版)

Stephen J. Chapman 著

王志强 李浩亮 林慕清 章小莉 等 译

```
TYPE :: vector
  REAL :: x, y, z
END TYPE vector

FUNCTION dot_product(vec_1, vec_2)
  TYPE(vector) :: vec_1, vec_2
  REAL :: dot_product
  dot_product = vec_1%x*vec_2%x + vec_1%y*vec_2%y + vec_1%z*vec_2%z
END FUNCTION dot_product

FUNCTION cross_product(vec_1, vec_2)
  TYPE(vector) :: cross_product
  TYPE(vector), INTENT(IN) :: vec_1, vec_2
  cross_product%x = vec_1%y*vec_2%z - vec_1%z*vec_2%y
  cross_product%y = vec_1%z*vec_2%x - vec_1%x*vec_2%z
```



中国电力出版社
CHINA ELECTRIC POWER PRESS

内 容 提 要

本书介绍了 Fortran 语言基础知识, 以及结构化程序设计思想, 该设计思想使得大型 Fortran 程序的维护更易于实现。本书在讲述 Fortran 知识过程中, 按适用于大型项目开发的模式来展开。本书的主要内容有: 计算机和 Fortran 语言简介, Fortran 基础知识, 程序设计与分支结构, 循环和字符操作, 基本的 I/O 概念, 数组, 过程, 数组的高级特性, 过程的附加特性, 字符变量的更多特性, 附加的内置数据类型, 派生数据类型, 过程和模块的高级特性, 高级 I/O 概念, 指针和动态数据结构, Fortran 面向对象程序设计, 优化数组和并行计算, 冗余、废弃以及已被删除的 Fortran 特性。

本书是一本理想的 Fortran 语言资料。

图书在版编目 (CIP) 数据

Fortran 程序设计: 第四版/ (美) 史蒂芬·查普曼 (Stephen J.Chapman) 著; 王志强等译.

—北京: 中国电力出版社, 2018.10

ISBN 978-7-5198-2294-1

I. ①F… II. ①史…②王… III. ①FORTRAN 语言—程序设计 IV. ①TP312.8

中国版本图书馆 CIP 数据核字 (2018) 第 174504 号

著作权合同登记号 图字: 01-2018-3788

Stephen J. Chapman

Fortran for Scientists and Engineers, Fourth Edition

ISBN: 978-0-07-338589-1, Copyright © 2018 by McGraw-Hill Education.

All Rights reserved. No part of this publication may be reproduced or transmitted in any form or by any means, electronic or mechanical, including without limitation photocopying, recording, taping, or any database, information or retrieval system, without the prior written permission of the publisher.

This authorized Chinese translation edition is jointly published by McGraw-Hill Education (Asia) and China Electric Power Press LTD. This edition is authorized for sale in the People's Republic of China only, excluding Hong Kong, Macao SAR and Taiwan.

Copyright © 2018 by McGraw-Hill Education and China Electric Power Press LTD.

版权声明

版权所有。未经出版人事先书面许可, 对本出版物的任何部分不得以任何方式或途径复制或传播, 包括但不限于复印、录制、录音, 或通过任何数据库、信息或可检索的系统。

本授权中文简体字翻译版由麦格劳-希尔 (亚洲) 教育出版公司和中国电力出版社有限公司合作出版。此版本经授权仅限在中华人民共和国境内 (不包括港澳台地区) 销售。

本书封面贴有 McGraw-Hill 公司防伪标签, 无标签者不得销售。

出版发行: 中国电力出版社

地 址: 北京市东城区北京站西街 19 号 (邮政编码 100005)

网 址: <http://www.cepp.sgcc.com.cn>

责任编辑: 刘 焯 (liuchi1030@163.com)

责任校对: 黄 蓓 常燕昆

装帧设计: 王红柳

责任印制: 杨晓东

印 刷: 三河市航远印刷有限公司

版 次: 2018 年 10 月第一版

印 次: 2018 年 10 月北京第一次印刷

开 本: 787 毫米×1092 毫米 16 开本

印 张: 51

字 数: 1322 千字

印 数: 0001—3000 册

定 价: 148.00 元

版 权 专 有 侵 权 必 究

本书如有印装质量问题, 我社发行部负责退换

在此谨将本书献给我 8 岁的儿子 Avi，
他是一个年龄只有 8 岁，
但能实际动手编写软件的可爱小伙子！

——Stephen J.Chapman

译 者 的 话

Fortran 起源于 1954 年,是最早出现的高级程序设计语言之一,它主要适合用来解决科学计算方面的问题。今天即使是已经进入到面向对象编程时代, Fortran 语言仍然被专注于数据计算的科学家和工程师广泛使用。本书为第四版,相较于第三版以 Fortran 95/2003 为蓝本,这次主要讲解 Fortran 2008,并在第 17 章增加了并行处理和优化数组两个 Fortran 2008 中的全新内容。但是原书的结构与风格依然未变: 章节学习目标明确,自上而下的程序设计方法贯穿始终,理论阐述翔实,例题讲解清晰,代码测试完整,验证学习效果的测验问题和练习内容丰富。所有这些特性都是作者为方便读者自学、掌握 Fortran 语言程序设计而精心设计的,这也成为用 Fortran 语言编程的人们乐于购买本书的亮点。

本书不仅是 Fortran 语言初学者的入门必选,也是熟练掌握 Fortran 语言人员的首选参考书。本书第 1~7 章,从计算机基本结构出发,讲述了计算机中数据的表示。Fortran 语言的发展历史,以及编程良好习惯如何培养,详尽介绍了 Fortran 语言基本知识。第 8~15 章介绍了 Fortran 语言高级特性,为初学者掌握和用好 Fortran 语言提供了强有力的支持。第 16 章介绍了 Fortran 语言面向对象编程方法,第 17 章介绍了 Fortran 2008 中的并行处理和优化数组,第 18 章逐一简述已经宣布废弃的那些 Fortran 旧特性,为遗留的旧版 Fortran 程序继续使用提供支持,这些新知识和旧特性是熟练使用 Fortran 编程人员最关心的内容。

本书自始至终强调树立实际工程编程思想,致力于培养读者编写良好的可读性、维护性和完整性的程序的能力。书中有大量的工程实例,在每章后面的习题中配有很多练习,其中不少练习涉及各个学科的实际工程计算。例如,相对论、电子工程、万有引力、逃逸速度、双曲余弦、振动周期、无线电接收机等编程。这极利于读者明白所学知识在实际工程中的运用,掌握所学知识,增加学习兴趣。

本书主要译者有刘博雅(第 1~4 章)、林慕清(第 5~7 章)、李浩亮(第 8~10 章)、王志强(第 13~15 章)、张克君(第 16 章)、李援南(第 17 章)、章小莉(第 18 章、附录及其他内容),还得到了张悦、钱榕、李伟、杨志成、霍刚、周志全、曹长宏、章晓盛等人的大力支持。最后章小莉对全部译稿进行了审阅。在此对大家的精诚合作表示深深的感谢!

我们希望这本新修订的第四版图书能为用计算机处理数据计算、完成科学研究的读者们带去更多的帮助。限于时间和水平,书中难免存在不足之处,敬请读者批评指正。

译 者

前 言

本书第一版是我编写和维护国防和地球物理领域的大型 Fortran 程序的产物。在工作期间,很显然,成功编写大型程序取决于策略和技术,那时候对一个年轻的工程师来说,维护 Fortran 程序与在学校学习 Fortran 程序的编写完全不同。一旦程序投入服务,维护和修改大型程序绝对需要高得令人难以置信的费用。因为编程的人容易理解程序,而修改程序的人相较于原程序员来说就很难明白它们的内容。我编写本书的目标是,既传授 Fortran 语言的基础知识,也很好地讲解编写和维护程序的技术。另外,还希望本书对今后打算从事编程工作的学生有参考价值。

在学生学习程序设计过程的早期,要教会他们花更多的精力来保证程序的可维护性是非常难的。因为课堂上的编程任务都很简单,一个人足以在短时间内完成编写任务,且这些程序也不需要多年的维护。正因为项目简单,一个学生只要参加了课程学习,就能完成所有编程任务,通过考试,即使不学实际工作中参加大型编程项目时需要的习惯,常常也能胜任从“编写”到产生代码整个过程的任务。

本书一开始介绍编写的 Fortran 程序就适用于大型项目开发,这样可以强调在编代码前认真进行设计的重要性。其中设计过程使用的技术是自顶向下设计,即把大型程序分解为可以单独实现的若干个逻辑部分。书中还强调完成单个逻辑部分编写过程的重要性和在开始将各个独立部分集成为最终产品的过程开始之前单元测试的重要性。

另外,本书讲述的 Fortran 程序,工程师和科学家在实际工作与研究中都会遇到。所有编程环境中一个问题是常见的:必须维护大量遗留代码。在特殊场合的遗留代码起初是用 Fortran IV (或甚至更早版本!)编写的,今天这些程序的结构已不再被使用。例如,这些代码可以通过使用 IF 语句来完成,或计算转到或赋值转到 GO TO 语句。第 18 章将介绍 Fortran 语言中那些不再被使用,但是在遗留代码中还是会遇到的旧特性。同时也强调在新程序中应该永不再用这些特性,并教会学生在遇到这些特性时如何处理它们。

第四版的变化

本书第四版直接基于 Fortran 95/2003 (适用科学家和工程师) 的第三版而编写,保留了上一版的编写结构,但全书都穿插 Fortran 2008 的新知识(以及 Fortran 2015 标准建议的相关内容),可喜的是,Fortran 起源于 1954 年,至今还生机勃勃。

从逻辑上来说,Fortran 2008 大部分新增技术是对 Fortran 2003 的扩展,各章的相应地方都介绍了这些新技术。但是,并行处理和优化数组为全新内容,第 17 章专门对它们有介绍。

绝大多数 Fortran 语言课程授课时间限定为三个月或一个学期,学生从中掌握 Fortran 语言的基础知识和编程基本概念。这些课程的内容为本书第 1~7 章,如果有时间的话,可选学第 8 章、第 9 章。给学生打下良好基础,以便他们在实践中更好地使用 Fortran 语言。

有能力的学生、工作中的科学家和工程师会需要第 11~15 章的 COMPLEX (复数)、派生数据类型和指针知识。工作中的科学家和工程师几乎肯定需要第 18 章中陈旧、丰富和

已删除的 Fortran 特性，这些知识很少在课堂中传授。但是本书包含了这些知识，目的在于实际中用 Fortran 语言解决真实问题时，本书依然有参考价值。

本书特点

本书设计了很多特性来强调如何用恰当的方法编写可靠性高的 Fortran 程序。这些特性对于首次学习 Fortran 的学生和实际工程中的人们都很有用。它们是：

1. 现代技术

本书在例题中始终用的是最新特性。许多 Fortran 2008 现代特性中不仅一直保留着 Fortran 语言旧版本的特性，还有了可取代它们的新特性。在这种情况下，例题中用的是现代新技术。旧技术的使用大部分被移到第 18 章中讲述，在那里强调了它们是旧版本的/不受欢迎的。保留的旧版本 Fortran 特性有：使用模块替代 COMMON（通用）块来实现数据共享，DO... END DO 循环替代 DO...CONTINUE 循环，内部过程替代语句函数、CASE 结构替代计算转向 GOTO 语句。

2. 强制类型

全书一直使用 IMPLICIT NONE 语句来强制每个程序中的每个变量类型要显式声明，以便编译时捕捉到常见的打字错误。与程序中每个变量显式声明一起，书中强调创建数据字典的重要性。该字典描述程序中每个变量的作用。

3. 自顶向下设计方法

本书第 3 章介绍了自顶向下设计方法，并且随后的其他章节一直在用该方法。这一方法鼓励学生在开始编代码前，仔细思考，对程序进行好的设计。强调明确定义问题的重要性，以便在开始任何其他工作之前准备好需要的输入和输出数据。一旦问题被恰当定义，紧接着就教授学生逐步细化问题，即将问题分解得更小，把单个子任务设计为子例程或函数，最后告诉学生每个阶段中测试的重要性，包括关于构建程序的单元测试和最终产品的集成测试。书中给出了几个程序示例说明如何进行测试，这些程序可以在一些数据集上正确运行，而在另一数据集上运行时却可能失败。

通过学习本书可以知道标准的程序设计过程是：

- (1) 清晰地说明要解决的问题。
- (2) 定义程序需要的输入和将产生的输出。
- (3) 描述打算用于程序的算法。这一步涉及自顶向下、逐步分解、伪代码或流程图。
- (4) 把算法转换成 Fortran 程序。
- (5) 测试程序。这一步包括对于特定子例程的单元测试，也包括用许多不同数据完成最终程序的集成测试。

4. 过程

本书强调用子例程和函数来实现在逻辑上大任务分解出来的子任务，并利用过程隐藏数据，还强调要重视在将子任务集成为最后的程序之前的单元测试。另外，书中还介绍了使用过程时常见的错误，以及怎样避免这些错误（参数类型不匹配、数组长度不匹配等）。书中强调对过程要用好显式接口，因为它使得在对 Fortran 编译时，编译器能尽可能多地捕捉常见的编程错误。

5. 简版和标准版 Fortran

本书强调编写简洁 Fortran 代码的重要性，因为这样才可以很容易地实现代码在不同类

型计算机之间的移植。书中还教授学生在自己的程序中一定要用标准版的 Fortran 语句，以便代码达到最大限度的简洁。另外，书中还教授多用像 `SELECTED_REAL_KIND` 函数这样的特性，以避免在不同机器上运行程序时，发生数据精度和类别不同的问题。

本书也教授学生不要在少数特殊的过程中用与机器型号有关的代码（如调用与机器系统有关的类库文件），以免在移植程序时不得不重写这些代码。

6. 良好的编程习惯

当介绍良好编程习惯时，便于对学生强调它们，这些知识点被突出表示，以示强调这是好的编程做法。另外，每章介绍的“良好的编程习惯”在每章最后都进行了小结。下面举例说明书中如何标示“良好的编程习惯”知识点。

良好的编程习惯
书写代码时，请保证用多个空格来缩进 IF 语句的语句体，从而使代码的可读性好。

7. 编程警示

书中对编程时需要注意的事项进行了突出表示，以提示要避免它们的发生。下面举例说明书中如何标示“编程警示”知识点。

编程警示
要关注整型数运算，因为整型数除法常常会得出难以预料的结果。

8. 指针和动态数据结构

第 15 章详细讨论了 Fortran 指针，包括指针使用不正确可能带来的问题。如内存不足，指针指向的空间将得不到分配。书中给出了很多动态数据结构示例，包括链表和二叉树。

第 16 章讨论了 Fortran 的对象和面向对象编程，包括涉及程序多态性的动态指针的使用。

9. 注意事项

本书中有很多的注意事项，这些注意事项列出的是学生可能感兴趣的其他信息。某些注意事项实质上展示了 Fortran 的发展史。例如，第 1 章有一条注意事项描述了 IBM Model 704，这是第一台运行 Fortran 的机器。另有一条注意事项是对书本知识的补充。再如，第 9 章的一条注意事项回顾和概述了 Fortran 中数组的多处不同。

10. 知识点完整参考资料

最后，书中给出了现代 Fortran 语言的完整参考资料，以便读者在实际应用中能快速查找需要的相关知识点。这里把特别关注点列入在特性表中，很容易查阅，其中包括晦涩和难于理解的特性，如通过地址引用传递过程名，在表控输入语句中的默认值等。

本书教学特点

为了便于学生理解书中所讲内容，本书有如下设计特点：

每章开头列出了本章学习目标。共有 27 个测验分布于全书之中，相应答案在附录 F。这些测验能帮助读者自我检验学习效果。另外，有大约 360 个练习出现在各章节之后，全部练习答案在图书网站可以找到，当然练习答案在教师参考手册上也有包含。各章中“良好的编程习惯”均加粗强调，常见的错误在“编程警示”中给出了提醒，各章末尾也对“良好的编程习惯”和 Fortran 语句和结构进行了小结，最后附录 C 给出了 Fortran 内置过程的详

细说明，附录 E 列出了全部的术语。

本书有教师参考手册，其中含有各章练习答案，教师也可在该图书网站的教师参考手册中找到答案，书中全部例题源代码和相关的补充材料也可以从图书网站上下载获得。

关于 Fortran 编译器的注意事项

在写作本书的时候，我用过两种 Fortran 编译器：Intel Visual Fortran Version 16.0 和 GNU G95 Fortran 编译器。两个编译器对 Fortran 2008 的完整功能基本实现，仅有非常少的几项功能不支持，同时这两种编译器对 Fortran 2015 未来的实现特性的建议给予了关注。

在此，我对潜在的用户强烈推荐该两编译器，因为 Intel Fortran 的优点是很好地集成了调试环境，但其缺点是价格偏高。G95 编译器则可以免费使用，但其调试功能相对难用。

用户最后的注意事项

无论我多么努力地审校本书，书中还是会存在一些印刷和打印错误，如果读者发现这些错误，请通过出版社告诉我，我将在重印和改版时努力修正它们。对于你的帮助与支持我深表感谢。

在本书网站上我将提供完整的勘误表和错误改正说明，网址是 www.mhhe.com/chapman4e，如果需要获取修订信息和最新更新请查看该网站。

致谢

在此我对 Raghu Srinivasan 和 McGraw-Hill 教育团队为本次图书修订所做的工作深表感谢。另外，我要感谢我的妻子 Rosa 和女儿 Devorah，她们在本次图书修订过程中给予了我大力支持（在前一版的图书修订中，我也对其他 7 个孩子给予了感谢，但是这次他们均已长大成人）。

Stephen J. Chapman
Melbourne, Victoria, Australia

目 录

译者的话

前言

第 1 章 计算机和 Fortran 语言简介	1
1.1 计算机	2
1.2 计算机中数据的表示	3
1.3 计算机语言	9
1.4 Fortran 语言发展史	9
1.5 Fortran 的演进	11
1.6 小结	14
第 2 章 Fortran 基础知识	17
2.1 介绍	17
2.2 Fortran 字符集	17
2.3 Fortran 语句结构	18
2.4 Fortran 程序结构	19
2.5 常数与变量	21
2.6 赋值语句和算术运算	27
2.7 内置函数	35
2.8 表控输入和输出语句	37
2.9 变量初始化	41
2.10 IMPLICIT NONE 语句	42
2.11 程序举例	43
2.12 调试 Fortran 程序	49
2.13 小结	50
第 3 章 程序设计与分支结构	61
3.1 自顶向下设计技术入门	61
3.2 伪代码和流程图的使用	65
3.3 逻辑常数、变量和运算符	65
3.4 控制结构：分支	71
3.5 有关调试 Fortran 程序的问题	89
3.6 小结	90
第 4 章 循环和字符操作	96
4.1 控制结构：循环	96
4.2 字符赋值和字符操作	118
4.3 Fortran 循环的调试	129

4.4	小结	130
第 5 章	基本的 I/O 概念	139
5.1	格式和格式化 WRITE 语句	139
5.2	输出设备	140
5.3	格式描述符	142
5.4	格式化 READ 语句	159
5.5	文件及文件处理介绍	164
5.6	小结	180
第 6 章	数组	191
6.1	声明数组	192
6.2	在 Fortran 语句中使用数组元素	193
6.3	在 Fortran 语句中使用整个数组和部分数组	203
6.4	输入和输出	206
6.5	程序举例	211
6.6	什么时候该用数组?	224
6.7	小结	225
第 7 章	过程	231
7.1	子例程	232
7.2	用模块共享数据	249
7.3	模块过程	255
7.4	Fortran 函数	257
7.5	过程作为参数传递给其他过程	263
7.6	小结	267
第 8 章	数组的高级特性	281
8.1	二维数组	281
8.2	多维数组	290
8.3	对数组使用 Fortran 内置函数	293
8.4	加掩码的数组赋值: WHERE 结构	295
8.5	FORALL 结构	297
8.6	可分配数组	299
8.7	小结	306
第 9 章	过程的附加特性	316
9.1	给子例程和函数传递多维数组	316
9.2	SAVE 属性和语句	325
9.3	过程中的可分配数组	329
9.4	过程中的自动数组	329
9.5	在过程中作为形参的可分配数组	335
9.6	纯过程和逐元过程	338

9.7	内部过程	340
9.8	子模块	341
9.9	小结	347
第 10 章	字符变量的更多特性	356
10.1	字符比较操作	357
10.2	内置字符函数	360
10.3	把字符变量传入子例程或函数	363
10.4	可变长字符函数	367
10.5	内部文件	369
10.6	例题	369
10.7	小结	374
第 11 章	附加的内置数据类型	379
11.1	REAL 数据类型的可选择类别	379
11.2	INTEGER 数据类型的可选长度	398
11.3	CHARACTER 数据类型的可选类别	399
11.4	COMPLEX 数据类型	400
11.5	小结	407
第 12 章	派生数据类型	411
12.1	派生数据类型简介	411
12.2	派生数据类型的使用	412
12.3	派生数据类型的输入与输出	413
12.4	在模块中声明派生数据类型	414
12.5	从函数返回派生类型	421
12.6	派生数据类型的动态内存分配	425
12.7	参数化派生数据类型	425
12.8	类型扩展	426
12.9	类型绑定过程	427
12.10	ASSOCIATE 结构	431
12.11	小结	432
第 13 章	过程和模块的高级特性	438
13.1	作用范围和作用域	438
13.2	块	443
13.3	递归过程	444
13.4	关键字参数和可选参数	446
13.5	过程接口和接口块	450
13.6	通用过程	454
13.7	用用户自定义操作符和赋值符扩展 Fortran	464
13.8	绑定赋值符和操作符	474
13.9	限制对模块内容的访问	474

13.10	USE 语句的高级选项	477
13.11	内置模块	480
13.12	访问命令行参数和环境变量	480
13.13	VOLATILE 属性和语句	482
13.14	小结	483
第 14 章	高级 I/O 概念	495
14.1	更多格式描述符	495
14.2	表式输入的默认值	503
14.3	Fortran I/O 语句详述	504
14.4	I/O 名称列表	522
14.5	未格式化文件	524
14.6	直接访问文件	526
14.7	流访问模式	529
14.8	派生数据类型的非默认 I/O	530
14.9	异步 I/O	536
14.10	访问特定处理器相关的 I/O 系统信息	538
14.11	小结	539
第 15 章	指针和动态数据结构	546
15.1	指针和目标变量	547
15.2	在赋值语句中使用指针	551
15.3	使用数组指针	553
15.4	使用指针的动态内存分配	554
15.5	指针当作派生数据类型的元素	557
15.6	指针数组	567
15.7	在过程中使用指针	569
15.8	过程指针	573
15.9	二叉树结构	576
15.10	小结	591
第 16 章	Fortran 面向对象程序设计	597
16.1	面向对象程序设计介绍	598
16.2	Fortran 类的结构	602
16.3	CLASS 保留字	602
16.4	在 Fortran 中实现类和对象	604
16.5	第一个例子: timer 类	606
16.6	方法的分类	611
16.7	对类成员的访问控制	617
16.8	析构函数	618
16.9	继承性和多态性	621
16.10	禁止在子类中重载方法	633

16.11	抽象类	633
16.12	小结	649
第 17 章	优化数组和并行计算	654
17.1	Fortran 中优化数组的并行处理	655
17.2	创建简单并行程序	655
17.3	优化数组	657
17.4	映像间的同步	658
17.5	例题：排序大数据集合	664
17.6	动态优化数组和派生数据类型	669
17.7	优化数组传给过程	670
17.8	临界区	671
17.9	并行程序中的极大危险	671
17.10	小结	675
第 18 章	冗余、废弃以及已被删除的 Fortran 特性	680
18.1	Fortran 90 前的字符限制	680
18.2	已被废除的源码格式	681
18.3	冗余数据类型	682
18.4	过时、废弃、和/或不必要的说明语句	682
18.5	共享内存空间：COMMON 和 EQUIVALENCE	685
18.6	不必要的子程序特性	690
18.7	其他执行控制特性	696
18.8	被废除的分支和循环结构	698
18.9	I/O 语句的冗余特性	702
18.10	小结	702
附录 A	ASCII 字符集	708
附录 B	Fortran/C 交互操作	709
附录 C	Fortran 内置过程	717
附录 D	Fortran 程序中的语句序列	758
附录 E	术语表	760
附录 F	各章测验的答案	782
作者介绍		801

第 1 章

计算机和 Fortran 语言简介

本章学习目标：

- 了解计算机的基本组成。
 - 理解二进制、八进制和十六进制数。
 - 学习 Fortran 语言发展历史。
-

计算机可以说是 20 世纪最重要的发明，它以多种方式深刻地影响着我们的生活。当我们去食品店购物，收银台的扫描设备通过计算机识别我们购买的食物。客户在银行的账户是用计算机来管理，自动柜员机、信用卡和借记卡使得我们无论白天还是晚上什么时候都可以办理业务，因为有很多的计算机一直在服务着。计算机也控制着我们的电话和电力系统、微波炉和其他设备的运行，甚至它还控制着汽车的引擎。今天，如果突然使它们的计算机失灵，那么许多国家几乎一夜之间就会崩溃。考虑到计算机对人们生活的重要性，很难想象第一台计算机是在大约 65 年前才发明出来的东西。

这是个怎样的设备呢？为什么对我们的生活产生如此大的冲击？计算机（computer）是一种特殊的机器，它可以存储信息，并能对信息以令人难以置信的极高速度进行算术计算。存储在计算机内存中的程序（program）能告诉计算机需要按怎样的顺序执行计算任务，从而保证对信息的计算顺利完成。大多数计算机都非常灵活。例如，如果在它的上面执行相应的程序，我用来编写本书的计算机也可以进行银行账户管理。

计算机能够存储巨大量的信息和相应的处理程序，当需要的时候，也可以使信息被立即有效利用。例如，银行的计算机能存储每个客户所有账户上每一笔资金进出的细目表。更大规模的信息存储例子是，在美国信用公司使用计算机存储每个公民的信用历史，有数十亿条信息记录。当需要的时候，公司可以搜索这十几亿条记录信息，从而确认每个人的信用记录，并在几秒钟的时间内向用户展示他的信用状况。

认识到计算机的能力不会超出人们的想象力这一点非常重要。计算机仅仅是按照存储在它内存的程序步骤在工作着，当计算机展示它做事情的聪明才智时，是因为聪明的人已经编写好了它正执行的程序，即那个舞台人类已经到达，是人类的集体智慧使得计算机创造了奇迹。本书将教会读者如何编写自己需要的程序，以便计算机完成你所需要它完成的工作。

1.1 计算机

图 1-1 所示框图是典型的计算机结构图，计算机上重要的组成部件（component）是中央处理单元（Central Processing Unit, CPU）、主存（Main Memory）、辅助存储器（Secondary Memory）和输入输出设备（Input and Output Device）。这些组成部件用框图描述如下：

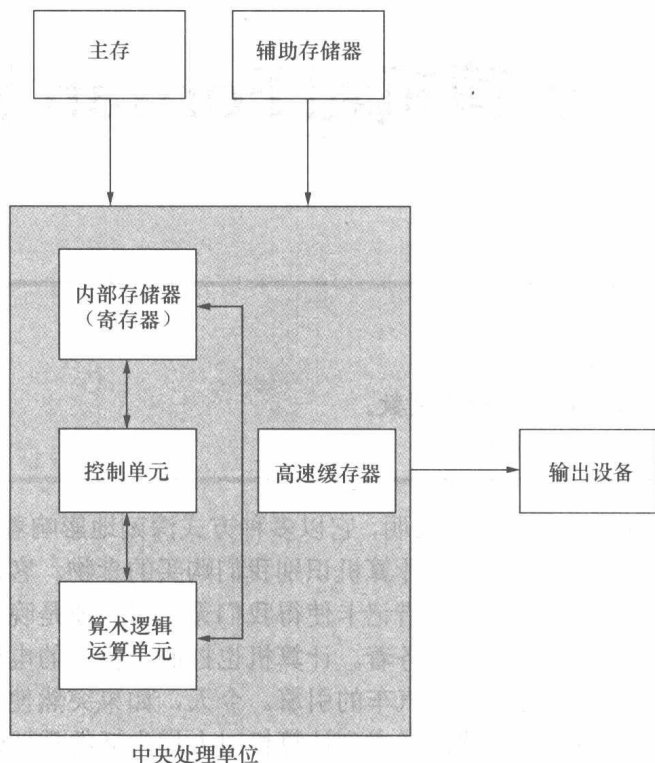


图 1-1 典型的计算机结构框图

1.1.1 CPU

中央处理单元是任何计算机的核心，它由一个控制单元、一个算术逻辑运算单元（ALU）和内部存储器组成。当 ALU 在完成实际算术运算时，CPU 中的控制单元控制着计算机所有其他部件的操作。CPU 中的内部存储器由一系列寄存器组成，用于临时存储运算时产生的中间结果，加上高速缓存，以临时存储在不久的将来需要使用到的数据。

CPU 中的控制单元不仅解释计算机程序指令，也负责从主存（或者高速缓存）获取数据值，存储到寄存器中，还负责把数据值从寄存器传送到输出设备或主存。例如，如果程序要求实现两个数的相乘，并保存结果。控制单元将从主存获取两个数据，存储它们到寄存器，之后，把数据送入 ALU，同时控制 ALU 完成两个数据相乘，再存储结果到另一个寄存器。最后在 ALU 完成数据相乘之后，控制单元取出目标寄存器中的结果，将它存储到高速缓存中（CPU 的其他部分在后续的时间内将数据从高速缓存复制到主要的内存）。

现代 CPU 通过并行操作多个 ALU，变得更快，能够在给定的时间内执行更多的操作。

它们还在 CPU 芯片上集成了更多的高速缓存，从而可以非常快速地获取和保存数据。

1.1.2 存储器

计算机存储器主要划分为三大类：高速缓存、主存或内存，以及辅助存储器。高速缓存存储器是存储在 CPU 芯片内部的存储器。CPU 可以非常快速地访问该存储器，所以允许以非常高的速度进行其内部存储的计算。控制单元预判在程序中将需要哪些数据，并将其从主存储器提前预取到高速缓存中，以便数据可以以最小的延迟得到使用。当数据长时间不再需要时，控制单元会将计算结果从高速缓存中复制回主内存。

主存通常由半导体芯片组成，这些半导体芯片通过称为内存总线的导线与 CPU 相连，其速度非常快。与 CPU 内部的内存相比，相对便宜。存储在主存储器中的数据可以在现代计算机上以几纳秒或更少（有时少得多）时间获取。因为它是如此的快速和便宜，主存储器被用来临时存储当前由计算机执行的程序，以及该程序需要的数据。

主存不用于存储永久保留的程序或数据。大多数主存是不稳定的（volatile），这意味着一旦计算机电源关闭，它的内容就会丢失。此外，主存价格相对更高，于是我们仅购买足以容纳下任何时候运行的最大程序的主存。

辅助存储器是由比主存速度慢和廉价的设备组成。购买它们的价钱可以比主存低很多，但是它们却能存储下比主存多的多的信息。另外，大多数辅助存储设备是稳定的（nonvolatile），这意味着无论计算机电源是否关闭，它都能保存程序和数据，且不丢失。典型的辅助存储器有硬盘（hard disk）、固态硬盘（SSD）、USB 存储棒和 DVD 等。辅助存储设备通常被用于存储不需要立即使用、但将来某个时候还是要用到的程序和数据。

1.1.3 输入和输出设备

数据通过输入设备输入到计算机，通过输出设备从计算机输出。现代计算机上最常见的输入设备有键盘和鼠标。人们可以用键盘敲入计算机程序或数据到计算机中。在某些计算机上也可以发现其他类型的输入设备，如触摸屏、扫描仪、麦克风和摄像机。

输出设备允许人们使用存储在计算机中的数据。今天计算机上最常见的输出设备是显示器和打印机，其他类型的输出设备有绘图仪和扬声器。

1.2 计算机中数据的表示

计算机存储器是由单个数十亿的开关组成的，它们每一个都可以开（ON）或关（OFF），且可以在两种状态下随意切换。每个开关表示一位二进制（也称为 bit，位）数据，ON 表示二进制的 1，OFF 表示二进制的 0。获取开关状态，单个开关就可以唯一表示数字 0 和 1。显然，由于人们日常需要使用的数据不仅仅是 0 和 1，所以在计算机中需要把多个数字位组合在一起来表示数据。当几个位被组合在一起的时候，可以用来表示二进制（基数为 2）系统中的数据。

最通用的一组位被称为字节（byte）。一个字节是 8 位二进制，它用来表示二进制数据。字节是用于度量计算机内存容量的基本单元。例如，我用来编写本书的计算机的主存大小为