

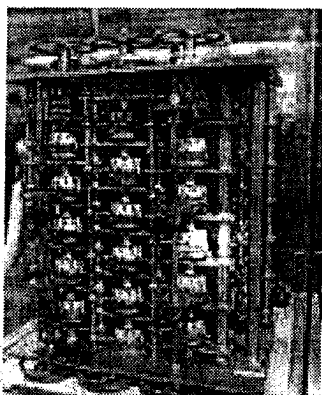
第一章 计算机基础知识

在开始 Fortran 90 程序设计的学习之前，先来简要介绍一下计算机的有关理论，以了解计算机软件和硬件的基本运行原理。这样对实际的程序编写会有很大的帮助。本章将从计算机的发展史开始，介绍计算机中的二进制的表示方法，计算机系统的组成，计算机机器语言、汇编语言及高级语言等计算机语言知识。

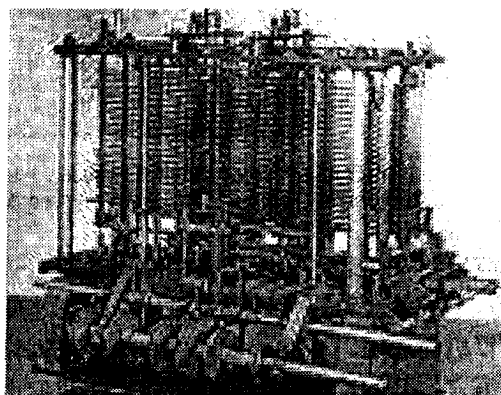
第一节 计算机的发展概述

如果能够查阅 1940 年以前的英文字典的话，就可以发现那时对于计算机（英文 Computer）的解释是指进行计算的人。尽管当时也有用来帮助人类做算术运算的机器及工具，包括中国古老的算盘，但都统称为计算器，而非我们现在命名的计算机。

工业革命后，科学家一直在尝试着制造功能更强大的计算工具，在 19 世纪出现了一些可以实际应用的机械式计算设备（如图 1-1 所示）



1812 年差分机



1834 年分析机

图 1-1 早期的机械式计算工具

在第二次世界大战中，由于军事和高科技的发展，需要解决一些十分复杂的数学问题，如破解对方密码、计算大炮弹道、飞机辅助设计等方面的问题。1942 年，美国国防部委派宾夕法尼亚大学的莫克利负责弹道的计算工作。他提出了研制新型计算机的建议，1943 年耗资 40 万美元实施，于 1946 年研制出世界上第一台由程序控制的电子数字计算机，名为“埃尼阿克”（ENIAC Electronic Numerical Integrator And Calculator 电子数字积分计算机）。用它计算弹道只要 3s，比机械计算机快 1000 倍，比人工计算快 20 万倍（见图 1-2）。

早期的计算机大都使用真空管来进行计算，体积非常庞大，而且编写程序时要使用机器语言，也就是程序员必须使用数字命令及开关来指挥计算机。另外，它的应用也只局限在军事和少数尖端科研方面。但尽管 ENIAC 有许多不足之处，但毕竟是计算机的始祖，揭开了电子计算机时代的序幕。从它问世以后，计算机技术飞速发展。20 世纪中晶体管的出现，使得改用晶体管的第二代计算机的体积减小、速度增加、耗电减少、价格降低、可靠性和性能提高。软件业也在此时期开始萌芽，这段期间开发出了编译器，可以把机器语言改用文本来表示（即汇编语言），然后再把这些文本翻译成机器语言，Fortran 及 COBOL 等高级语言也在同时期问世。



图 1-2 ENIAC 共用了 18000 个电子管和 86000 个其他电子元件，1500 个继电器，耗电 150 kW/h，运算速度却只有 300 次/s(各种运算)或 5000 次/s(加法运算)，占地 170m²，质量达 30 t。

集成电路发明后，可以把多个晶体管放在一小块芯片上，开创了半导体时代。这也让计算机进入了所谓的第三代计算机时代。运算速度进一步提高，每秒可达千万次；软件逐渐完善，计算机向通用、标准、多样化和机种系列化发展；高级程序语言发展，出现操作系统和会话式语言；开始广泛应用于各个领域。我们现在使用的计算机是第四代计算机，它的芯片采用大规模集成电路，运算速度达到每秒几千万次到几亿次，广泛应用于科学计算、数据处理、过程控制、计算机辅助系统以及人工智能等各个领域，进入以计算机网络为特征的时代。现在计算机拥有的计算能力及存储容量都远远超过了早期的超级计算机。

第二节 数 字 化

在数字计算机中，数据是用逻辑电路的状态来表示的，由于数字逻辑电路只有导通和截止两种稳定状态，只能对应表示 0 和 1 两个数据，因而数字计算机内部的数据全部使用二进制数。记录一个二进制数所需要的容量大小，称为一个位（bit）。但通常在使用时，并不是

以一个位为最小的单位，而是组合 8 个位(bit)为一个字节 (byte) 来做基本单位的。在此基础上，1024 个字节为 1KB，1024KB 为 1MB，1024MB 为 1GB。

1byte=8bits

1KB=1024bytes= 2^{10} bytes

1MB=1024KB= 2^{20} bytes

1GB=1024MB= 2^{30} bytes

在计算机技术中会使用到十进制、二进制、八进制和十六进制。为了符合人们的日常习惯，在计算机源程序和应用程序的人机界面上仍然使用十进制数，但计算机内部只能采用二进制数。

计算机处理的数据不只是数值性数据，还有非数值性数据的处理，诸如文本、多媒体图像、声音等。所以需要在数据输入计算机的过程中，系统能自动将用户输入的各种数据按编码的类型转换成相应的二进制形式进入到计算机中以便处理。同样在输出过程中，再由计算机系统自动将二进制编码数据转换成用户可以识别的数据格式输出。常用的编码有 ASCII、BCD、汉字编码。

ASCII码定义的是欧美语系的文本拼音字母，使用 $2^7=128$ 个七位二进制进行 128 个字符的表示。它包含了所有的英文字母、数字及图形符号，每个英文字母使用一个字节就可以进行记录。

用计算机处理汉字时，汉字种类繁多，编码比拼音文字困难，而且在一个汉字处理系统中，输入、处理和输出的编码不同，标准采用的是国标码。每个汉字要用 2 个字节来记录。

其他如多媒体信息，在计算机中的处理都是先把声音、图像等转换成数字后，再以二进制方法来保存这些数字。计算机本身并不会知道现在它所处理的二进制数据，原本记录的是是什么。所以二进制数据需要先说明所保存的是什么数据，计算机才会知道该如何去解释。举例来说，如果有一个文件的扩展名是 TXT，那么它的原始数据就应该是记录的数字化后的文本，经过 ASCII 或汉字编码表对照后可以转换为英文或中文来显示。

第三节 计算机系统

计算机系统由硬件系统与软件系统两部分组成。硬件指的是所能够看得见的组成计算机的物理设备，例如：显示器、主机等，是构成计算机的实体；软件是用来指挥计算机完成具体工作的程序和数据，是整个计算机的灵魂。

一、计算机的硬件系统

美籍匈牙利科学家冯·诺依曼在研制世界第一台计算机时提出了计算机体系结构的概念，他将计算机硬件系统分为 5 个主要部分：运算器、控制器、存储器、输入设备和输出设备（见图 1-3）。50 多年来，计算机系统基本结构没有变化。

1. 运算器

运算器也称算术逻辑单元，是计算机的核心部件，主要负责对信息的加工处理。运算器不断地从存储器中得到要加工的数据，对其进行算术运算和各种逻辑运算，并将最后的结果送回存储器中，整个过程在控制器的指挥下有条不紊地进行。

2. 控制器

控制器是计算机的指挥中枢。提供控制信号，控制、协调各部分操作的部件，使计算机能够自动地执行命令。运算器和控制器合称为中央处理器（Central Processing Unit，简称 CPU），CPU 是计算机的核心。

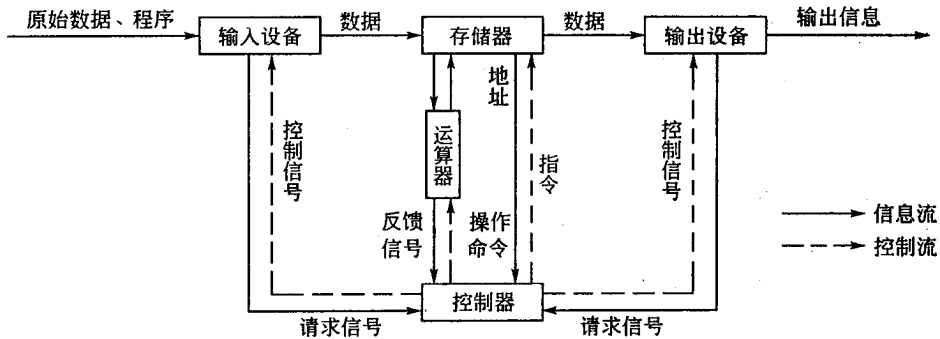


图 1-3 计算机硬件系统结构示意图

3. 存储设备

存储设备主要用来存放程序和信息，具有记忆功能。存储器划分为若干存储单元，每个存储单元以字节（Byte）进行编址。存储器的容量是存储器中所包含的字节数，一般用 KB、MB 和 GB 来表示。

存储器分为内存储器（主存储器）和外存储器，而内存储器又分为随机读写存储器（RAM）和只读存储器（ROM），RAM 中的信息可以随机地进行读写，断电后其中的信息将丢失，而 ROM 中的数据只能读出不能写入，断电后也不会丢失其中的数据。内存储器分别与 CPU 和输入输出设备相连，直接配合 CPU 工作。内存储器的特点是速度快，但造价高。现在微型计算机的内存越来越大。外存储器又称为辅助存储器，其容量相当大，但速度慢，主要有硬盘、软盘、光盘等。计算机在运行时，会先把程序及数据从光盘或是硬盘中取出，放到内存中再开始运行。

4. 输入设备

输入设备指任何可以让用户来操作计算机，或用来向计算机传送信息的工具。它能将信息变为计算机能识别的形式存放在内存中。常用的输入设备有键盘、鼠标、扫描仪、数码相机等。

5. 输出设备

输出设备是用来将计算机中的信息传送到外部设备的工具。常用的输出设备有显示器、打印机、绘图仪等。

二、计算机的软件系统

软件是微机的灵魂。没有安装软件的微机称为“裸机”，无法完成任何工作。软件系统是程序系统和文档的总称。软件分为系统软件、应用软件和支持软件等。

1. 系统软件

系统软件是用来对计算机系统进行管理、监控以及对计算机资源进行维护，是计算机的

基础软件。包括整个计算机的操作系统、各种计算机语言处理程序（如汇编程序、编译程序和解释程序等）、数据库管理系统等。

计算机只是一部接受程序命令运行的工具。在计算机启动以后，第一个步骤会自动去找存放在 BIOS 中的程序来执行，这段程序通常会检查计算机本身的硬件状态是否正常，确认内存和硬盘是否正常地安装在主板上等等。完成检查之后，BIOS 会把硬盘启动扇区中的程序取出执行，通常这个程序就是操作系统。

操作系统是帮助配置计算机资源的工具。如提供用户的文件管理、提供用户硬件管理、提供多任务执行功能等。每个操作系统使用计算机资源的方法是不同的。

2. 应用软件

应用软件是利用计算机及其系统软件，为解决实际问题而编制的、具有专门用途的软件，如各种字处理软件（WS、WPS、CCED）、科学计算用的软件包、计算机辅助制造、辅助设计软件、各种图形软件等。应用程序是面向用户的最高层程序。

3. 支持软件

支持软件是在计算机硬件和系统软件的基础上，为支持其他软件的研制和开发的软件，如追踪调试程序、格式化程序等。

第四节 计算机语言

任何一个计算机系统都是按照人们用某种程序设计语言编写的程序进行工作的，所以计算机语言是人机交流的有效工具。从计算机诞生之日起，程序设计语言从低级语言到高级语言不断地向前发展，新的、功能强大的程序设计语言不断地涌现。现在，世界上正在使用的各种程序设计语言不下百种。

程序设计语言分为低级语言和高级语言两种。而低级语言又分为机器语言和汇编语言。

一、机器语言

最初的计算机语言是机器语言，由二进制代码（0和1组成的一序列数字代码）按照确定的规则描述和表达，这种代码能够被计算机直接理解、识别和执行，称为机器语言。

机器语言与具体的计算机硬件系统有关，不同种类的计算机配置有不同的机器语言，因而尽管执行效率高，但可移植性差，不易维护，难学难懂，出错后难改。

二、汇编语言

为了解决机器语言存在的突出问题，出现了汇编语言或符号语言。汇编语言是用一些易于理解的助记符来代替机器语言中的二进制数字代码。用汇编语言编写程序，与机器语言相比程序的可理解性、可读性、可靠性、可维护性和可移植性都大大提高了。汇编语言程序代码与机器指令相对应，而且汇编语言同机器语言一样，与计算机硬件系统有关，在字符型上看起来非常不直观，都属于低级语言。

三、高级语言

为了解决程序的通用性和易读性的问题，开发出了高级语言。高级语言采用人们熟悉的

英语和数学公式来表达，它的编写方式比较直观，程序员可以用比较抽象与机器无关的方法来解决。高级语言克服了低级语言的缺点，它面向过程、易读易学、易于编写、可靠性高、可维护性好，基本与计算机硬件系统无关，能在不同种类的计算机上运行。这也导致硬件制造跟软件研发分成了不同的专业领域。

高级语言程序不能被计算机直接识别、理解和执行，必须要经过编译程序进行翻译工作，把高级语言转换成计算机所能识别的机器语言。如常用的高级语言 BASIC、PASCAL、SQL、JAVA、C、C++ 等。随着 Windows 操作系统的普及，以上基于 Windows 系统的可视化的程序语言已经相当普遍。

复习思考题

1. 一个完整的计算机系统由什么组成？
2. 计算机内部使用的进制数有哪几种？
3. 位 (bit)、字节 (Byte)、千字节 (KB)、兆字节 (MB) 之间的关系是什么？
4. 简述计算机硬件系统的组成。
5. 机器语言、汇编语言和高级语言特点分别是什么？有哪些异同点？

第二章 FORTRAN 程序设计基础

在这一章中我们先来了解一下 Fortran 90 程序设计的相关基础知识，主要包括了 Fortran 语言本身的发展简史，Fortran 90 的标准特性，关于程序设计的概念和方法的了解，Fortran 90 字符集的规定，以及 Fortran 90 编译器的使用。

第一节 Fortran 语言的发展历史

Fortran 是为科学、工程问题中的那些能够用数学公式表达的问题而设计的语言，主要用于数值计算。Fortran 是英文词 Formula Translation 的缩写，其含义为“公式翻译”。这种语言简单易学，因为可以像抄写数学教科书里的公式一样书写数学公式，它比英文书写的自然语言更接近数学语言，而且语法规则严格，编写的程序安全可靠。Fortran 语言至今已有 40 多年历史，但仍历久不衰，始终是数值计算领域所使用的主要语言。

第一代 Fortran 语言是在 1954 年提出来的，当时由 John Backus 率领的 IBM 公司的一个小组，于 1956 年在 IBM 704 计算机上研究开发的一套程序，它可以把使用接近数学语言的文本翻译成机器语言，称为 Fortran I。Fortran 语言推出后，由于运行效率高而成为最早的广泛使用的高级程序设计语言。

正因为 Fortran 满足了现实的需要，所以它传播得很快，在传播和使用过程中不可避免地产生了多种版本。其中最为流行的是 1958 年出现的 Fortran II，它对 Fortran I 进行了很多重要的扩充（如引进了子程序等概念）。其后出现的 Fortran III 未在任何计算机上实现。1962 年初出现的 Fortran IV 对原来的 Fortran 做了一些改变使得 Fortran II 源程序在 Fortran IV 编译程序下不能全部直接使用，导致了语言不兼容的问题。这样就形成了当时同时使用 Fortran II 和 Fortran IV 两种程序设计语言的局面。

由于 Fortran 语言的种类很多，各种 Fortran 语言的语义和语法规则又不完全一致，这给用户带来了极大的不便。使用者热切希望有一种能脱离实现机器，在各种机型上都能互换通用的 Fortran 语言，从而提出了 Fortran 语言标准化的问题。

1962 年 5 月，当时的美国标准化协会 ASA (American Standard Association)（后来改名为美国国家标准学会 ANSI—American National Standards Institute，现名为 NIST—National Institute of Standards and Technology）成立了有关工作组开展此项工作。1966 年 3 月正式公布了两个美国标准文本：一是美国国家标准 Fortran (ANSI X3.9-1966)（相当于 Fortran IV）；二是美国国家标准基本 Fortran (ANSI X3.10-1966)（相当于 FORTRAN II）。美国国家标准基本 Fortran 是美国国家标准 Fortran 的一个子集，从而达到了语言上的向上兼容。

Fortran IV（即 Fortran 66）流行了十几年，几乎统治了所有的数值计算领域。但很多编译程序往往为实现一些有用的功能而忽略标准。另外，在结构化程序设计方法提出以后，人

们开始感到 Fortran IV 已不能满足要求。Fortran IV 不是结构化的语言，没有直接实现三种基本结构的语句。

其后，根据语言的发展和计算机硬件的进步的需要，1978年4月美国标准化协会正式公布新的美国国家标准，即 ANSI X3.9-1978 Fortran，称作 Fortran77。1980年，Fortran77被正式采纳颁布为国际标准，即《程序设计语言 Fortran ISO 1539-1980》，这种新标准大致上保留了已有的 Fortran 66 标准，删去了部分内容，又添加了一些逻辑判断及输入输出方面的功能。此外，还扩充了字符处理功能。使 Fortran 不仅可用于数值计算领域，还可以适用于非数值运算领域。Fortran 77 增加了一些结构化的语句，能用于编写结构化程序，但还不是完全结构化的语言。

为了引入一些新的功能，适应语言的发展，同时又要确保程序的高效率，其标准化的工作花了十几年，最终在 1991年5月通过了 Fortran 90 新标准 ANSI X3.198-1991，相应的国际化标准组织的编号为 ISO/IEC1539:1991。新的 Fortran 标准废弃了过时的严格的源程序书写格式，改善了语言的正规性，并提高了程序的安全性，功能有更大的扩充，是一个能适应现代程序设计思想的现代程序设计语言。在此新标准中增加了许多新的功能，其主要特色是加入了面向对象的概念及工具、提供了指针、加强了数组的功能、增加了多字节字符集的数据类型及相应的内在函数等。为了保护对 Fortran 77 用户在软件开发上的巨大投资，整个 Fortran77 被作为 Fortran 90 的一个严格子集。

随着其他程序设计语言的迅速发展，Fortran 语言不再是唯一适用的程序设计语言。然而，尽管在一些特殊领域，使用其他程序语言更为合适，但在数值计算、科学和工程技术领域，Fortran 仍具有强大的优势。如可移植性强、运行速度快、应用广泛。Fortran 经历了长期的发展，各种配套软件相当丰富，适合各种计算机系统。其强大的生命力在于它能紧跟时代的发展，不断更新标准，每次新的版本推出都在功能上有一定突破性进展。Fortran 90 不仅仅是将已有的语言进行标准化，更重要的是发展了 Fortran 语言，吸取了一些其他语言的优点。所以，虽然 Fortran 语言历史悠久，但仍在日新月异地发展。

在当前程序设计语言层出不穷的今天，学习 Fortran 语言的意义在于继承传统和紧跟时代。不仅一些爱好者推崇 Fortran 语言，而且科学计算编程的专家也认为，科学与工程相关专业的学生应该采用 Fortran 而非 C 和 C++编程。这是因为，Fortran 90 具有 C++所有的重要功能（尚不具备的预计将在 Fortran2k 版本中推出）然而 C 语言主要是用于微机上的廉价开发，而 Fortran 的目的是为了产生高效最优化运行的可执行程序，用 Fortran 编写的大型科学计算软件较 C 语言编写的通常要快一个量级，其程序编写更为自然和高效，且易学易懂。尤其是在高性能并行计算逐渐成为时代必然的今天，不仅巨型机而且微机和工作站也有了多处理器，其串行机上的线性内存模式已不再适用，而只有 Fortran 具备处理相应问题的标准并行化语言，其独特的数组操作充分体现了它的先进性。

Fortran 90 的现代特性表现在：

(1) 加强了程序的可读和可维护性：淘汰所有转移语句，用新的控制结构实现选择分支与重复操作，使程序结构化。同时增加了结构块、模块及过程的调用灵活形式，使程序易读易维护，新的模块装配取代了 Fortran 77 的许多旧语句，使程序员更为清晰明确地定义全局数据。增加了新的数据种别说明，使得 Fortran 程序在不同计算机编译环境下有更自由的移植性。

(2) 发展了现代算法功能：加强了数组的算法功能，引进了多种数组操作功能与概念，使数组像一个变量一样自由操作，使数组的并行化运算成为可能。增加了适于操作数据结构的派生类型，提高了文字处理功能，胜任信息管理系统、办公自动化的任务。特别是动态存储功能的引进极大地加强了它在数值计算领域中应用的威力。

(3) 扩大与编程者的友好界面：新的编程形式减少了繁琐与格式的束缚，接近自然语言与公式演算。允许在字符数据中选取不同种别，在字符串中可使用各国文字（例如汉字），还可任意使用化学、物理、数学的各种专业字符。

第二节 程序和程序设计

计算机对于任何问题的解决，都是在程序支持的基础上来完成的。计算机依赖的程序是用计算机能识别的语言所描述的解决实际问题的方法和步骤。计算机能直接识别的语言是机器语言，但机器语言用二进制代码表示机器指令，而且机器指令与具体的计算机结构有关，程序直观性差、通用性不强。所以初学者一般都学习利用一种高级语言来编写程序。FORTRAN语言便是在科学计算领域应用十分广泛的一种高级语言。

使用计算机解决实际问题，通常是先要对问题进行分析并建立数学模型，然后考虑数据的组织方式和算法，并使用某一种程序设计语言（选择编译器）编写程序，最后调试程序，使之运行后能产生预期的结果，这个过程就是程序设计。所以在拿到一个实际问题之后，应对问题的性质与要求进行深入分析，从而确定求解问题的数学模型和方法，接下来进行算法设计，并画出流程图。有了算法流程图，再来编写程序是很容易的事情了。

一、程序设计与程序结构

1966年 Bohra 和 Jacopini 提出了组成结构化算法的三种基本结构：顺序结构、选择结构和循环结构。

顺序结构是最简单的一种基本结构，依次顺序执行不同的程序块，如图 2-1 (a) 所示。

选择结构根据条件满足或不满足而去执行不同的程序块。在图 2-1 (b) 中，当条件 P 满足时执行 S1 程序块，否则执行 S2 程序块。

循环结构是指重复执行某些操作，重复执行的部分称为循环体。循环结构分为当型循环 (DO WHILE) 和直到型循环 (DO UNTIL) 两种，见图 2-1 (c) 和 (d)。当型循环先判断条件是否满足，当条件 P 满足时反复执行 S 程序块，每执行一次测试一次条件 P，直到 P 不满足时为止，跳出循环体执行下面的基本结构。直到型循环先执行一次循环体，再判断条件 P 是否满足，如果不满足则反复执行循环体，直到 P 满足为止。直到型循环至少执行一次循环体，而当型循环有可能一次也不执行循环体。

三种基本程序结构具有的共同特点为：只有一个入口；只有一个出口；结构中无死语句（即结构内的每一部分都有机会被执行）；结构中无死循环。

结构化定理声明，任何一个复杂问题的程序，都可以用以上 3 种基本结构组成。具有单入口单出口性质的基本结构之间形成顺序执行关系，使不同基本结构之间的接口关系简单，相互依赖性少，从而呈现出清晰的结构。

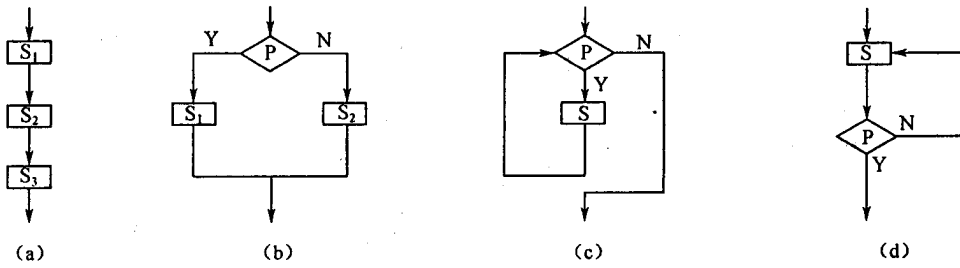


图 2-1 程序的三种基本结构

二、程序设计方法

早期程序设计的好坏常以运行速度快、占用内存少为主要标准，然而在计算机的运算速度大大提高，存储容量不断扩大的情况下，程序具有良好的结构成为第一要求。一个结构良好的程序虽然在效率上不一定最好，但结构清晰，易于阅读和理解，便于验证其正确性。从而促使了结构化程序设计方法的产生。

结构化程序设计采用自顶向下、逐步求精和模块化的分析方法。

(1) 自顶向下是指对设计的系统要有一个全面的理解，从问题的全局入手，把一个复杂的问题分解成若干个互相独立的子问题，然后对每个子问题再做进一步的分解，如此重复，直到每个问题都容易解决为止。

(2) 逐步求精是指程序设计的过程是一个渐进的过程，先把一个子问题用一个程序模块来描述，再把每个模块的功能逐步分解细化为一系列的具体步骤，以至能用某种程序设计语言的基本控制语句来实现。逐步求精总是和自上向下结合使用，一般把逐步求精看作自顶向下设计的具体体现。

(3) 模块化是结构化程序的重要原则。所谓模块化就是把大程序按照功能分为较小的程序。一般一个程序是由一个主控模块和若干个子模块组成的。当然主模块是相对于子模块而言。主控模块用来完成某些公用操作及功能选择，而子模块用来完成某项特定的功能。子模块可以有更下一层的子模块。一个复杂的问题可以分解成若干个较简单的子问题来解决。这种设计风格便于分工合作，将一个大的模块分解为若干个子模块来分别完成，之后用主控模块控制、调用子模块。这种程序的模块化结构如图 2-2 所示。

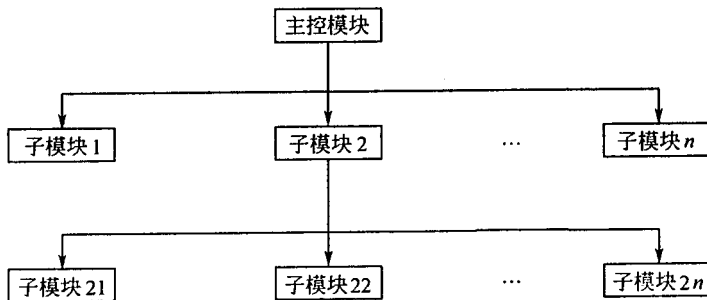


图 2-2 程序的模块化结构

随着硬件性能的提高和图形用户界面的推广，应用软件的规模高速增长。由此引起的复

杂性单靠结构化程序设计方法已无法解决，于是产生了面向对象的程序设计（Object-Oriented Programming）。

传统的程序设计是基于求解过程来组织程序流程的，数据和施加于数据的操作是独立设计的，以对数据进行操作的过程作为程序的主体。面向对象程序设计则以对象作为程序的主体。对象是数据和操作的“封装体”，封装在对象内的程序通过“消息”来驱动运行。在图形用户界面上，消息可通过键盘或鼠标的某种操作来传递。但要注意的是，使用面向对象的程序设计方法并非摒弃现有的结构化程序设计方法，这两种方法并不是相互孤立的。事实上，在对象内部的实现上，常常使用结构化程序设计方法。

第三节 Fortran 90 字符集和符号名的规定

Fortran 语言中最基本的成分是字符，任何其他的 Fortran 元素都要由字符按照一种规则组成。字符的集合称为 Fortran 语言的字符集。在 Fortran 90 中共有 58 个字符。它们是：

字母字符 (26 个)	A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
数字字符 (10 个)	0 1 2 3 4 5 6 7 8 9
特殊字符 (22 个)	: + - * / = () , . ' ! " % & ; < > ? \$ _ (空格)

Fortran 90 中使用的字母都是大小写字母不加区分。其中特殊字符的意义大多数是明显的，但*，/，.等在不同的位置时有不同的意义。其中“\$”和“？”号在程序中没有确切意义。特殊字符中货币符号因国家而异。空格字符的使用往往是为了使程序清晰。

在 Fortran 90 中还允许有该处理系统中可以表示的附加字符，但只可出现在字符常量、字符串编辑描述符、注解和输入输出记录中。

在 Fortran 90 中规定了符号名中下划线可以作为有效字符使用。符号名的最大长度是 31 个字符，其中第一个字符必须为字母。合法的符号名的例子为：

ABC1, NAME_LENGTH, S_P_A_C_E_M_A_N, TRANSFER_

非法的符号名如：

23LT, A (I), L-10, a2

注意尽量不要使用 Fortran 语言中的关键字和内部函数名作为符号名。在每一种计算机系统所用的 Fortran 编译器中，可能对字符集或其功能做某些扩充。因此在使用某一具体的计算机时，应了解它的规定。例如“\$”号可用于从屏幕上连接输出输入字符于同一行。标准还允许 Fortran 90 扩充进各国文字，各种专业用符号，这要看厂商装入的编译系统是否支持这些字符的使用。某些系统还可以用这些字符作为字符常量或注释，注意这类字符是 2 字节长度，在计算字符串长时要加倍。

第四节 源代码的编写格式

Fortran 程序代码的编写格式有两种：Free Format（自由格式）及 Fixed Format（固定格式）。在添加一个程序文件时就有这两个格式可以选择。其中固定格式是属于旧式的写法，它在编写版面上有很多限制。而自由格式是 Fortran 90 之后的新写法，取消了很多旧的限制。

3. 语句区

第 7~72 个字符，是编写语句的区域。语句区的空格除引号内字符串中的空格外是无效的。程序书写时不要求一定从第 7 列开始写语句，可以从第 7 列以后（72 列以前）的任何一列开始写，但一行只能写一个语句。如果写满了 72 列，一旦在终端上修改程序时在该行又插入了一些字符，就会使本行最后几个字符超出语句区而引起意料不到的错误。特别注意到语句最后的空格将可能溢出 72 列，在某些计算机系统上将导致难以查出的错误。

4. 注释区

第 73~80 个字符，为了便于阅读，Fortran 利用此 8 列为程序行编序号以便查找。程序员一般注释区只对程序员提供辨别信息，不是语句的一部分，在编译时不对 72~80 列作处理。也可在每行注释行对程序进行注释，但超过的部分会被忽略或发出错误信息。

固定格式举例：

EX0201.FOR

```
1. C      FIXED  FORMAT  DEMO      ! 第一个字符 C表示该行文本为批注
2.      program main
3.      write(*,*) 'Hello'
4.      write(*,*)                  !4、5行是把第3行程序拆成两行，故5行的第6个字符是一个
5.      !'Hello'                    非0字符，这样5行会当成是4行的续行
6. 100   write(*,*) 'Hello'
7. 10    stop
8.      end
```

从以上程序可以知道，每一行程序代码的最前面至少都会留有 6 个字段。因为前 6 个字段不能用来作为程序代码。第 6、7 行的最前面的数字是行代码，行代码可以是任何数字，只要数值大小在 5 位数范围内即可，它和程序代码的执行先后顺序无关。另外程序代码命令之间的空格，不具有任何意义。例如在 WRITE 命令后的 'Hello' 字符串间插入了不同数目的空格，但代码的含义是相同的。

二、Fortran 自由格式

随着穿孔卡片的淘汰，固定格式不再继续使用下去。在 FORTRAN 90 中，用自由格式编写程序有很大的自由度。它没有规定每一行的第几个字符有什么作用，与固定格式相比，不易产生键入位置的错误，而且易读易懂。但要注意以下几点：

(1) 起始用注释符“!”号，此行其后的所有字符均作为注释（字符串中的!号除外，如 print *, 'help!!!!'）。

(2) 每行可以编写 132 个字符。可以有复数条程序语句，语句间用分号“；”分开，且语句没有位置规定。

(3) 当一个语句非常长以致于 132 个字符都书写不下时，允许有 39 个续行。在语句行最后如果有续行符号“&”，表示下一行程序会和这一行连接；如果一行程序代码的开头是续行符号“&”，则表示它会和上一行程序连接。注意语句的有效字符是从“&”前和续行符“&”之后的位置算起。行号放在每行程序的最前面。

当程序员要将自己编写的 Fortran 90 程序与现有的用 Fortran 77 编写的子程序库在源码级

结合起来的话，需要特别注意格式的差别。

第五节 Fortran 程序的基本结构

在开始程序设计之前，首先来了解一下 Fortran 90 程序的总体构成。看下面的例子：

EX0202.FOR

```
1. PROGRAM    ex0202           !主程序
2. INTEGER A(4,4), SUM, M
3. READ(*,*) M
4. READ(*,100)((A(I,J), J=1, M), I=1, M)
5. 100  FORMAT(4I2)
6. 10  CALL SUB(A, M, SUM)
7.     IF(SUM.GT.30) THEN
8.     GOTO 10
9.     END IF
10.    M=M+1
11.    WRITE(*,*) SUM, FUN(A, M)
12.    END
13.    SUBROUTINE SUB(B, N, S) !外部子程序
14.    INTEGER B(N,N),N,S
15.    S=0
16.    DO 20 I=1, N
17.    S=S+B(I, I)
18. 20  CONTINUE
19.    N=N-1
20.    END
21.    FUNCTION FUN(B, N)      !函数子程序
22.    INTEGER B(N, N), N
23.    FUN=1
24.    DO 30, I=1, N
25.    FUN=FUN*B(I, I)
26. 30  CONTINUE
27.    END
```

上面程序中行号 1~27 只为说明方便加上的，并不是程序的内容。一个 Fortran 程序具有类似的结构形式。这种结构是在编写 Fortran 程序时必须遵循的。归纳起来有：

(1) 一个 Fortran 程序由若干个程序单元组成。程序单元可以是主程序、外部子程序或模块，其中模块包含可以被其他程序访问的各种程序实体。一个 Fortran 程序包含一个主程序和若干个其他类型的程序单元。若一个程序包含多个程序单元，则可对各程序单元分别进行编译，然后将它们连接成一个统一的可执行文件。

(2) 不管是主程序还是子程序，它们均以开始语句作为程序单元的第一条语句，然后是程序体，最后是该程序单元的结束语句。不同类型的程序单元的开始语句不尽相同，主程序用 PROGRAM 后接程序名开头，也可省略；而子程序则要以 FUNCTION 加上程序名或 SUBROUTINE 加上程序名开头。程序体包括说明部分和可执行部分。每一个程序单元都必须以 END 语句结束。例如一个简单的 Fortran 程序的结构如下：

```
程序开始  —————> [PROGRAM MAIN] <————— MAIN  是自定义的名称
                                     ...
主程序代码  —————> WRITE(*,*)"Hello"
                                     ...
程序结束  —————> STOP <————— 此程序可以省略
主程序代码结束  —————> END [PROGRAM[程序名]]
```

Fortran 程序通常以 PROGRAM 描述开头，PROGRAM 后面还要接一个自定义的程序名。位于 PROGRAM 和 END 之间的程序代码为主程序。注意 Fortran 程序的最后还要有 END 描述，表示程序代码写到这一行结束，其中方括号中的内容为可选项。END 语句是程序结束的标志，任何一个程序单位最后一个语句必须是 END 语句，END 必不可少。

第六节 Fortran 90 编译器

一个 Fortran 程序要在计算机上运行，要有相应的编译系统的支持。Microsoft 的 Fortran PowerStation 4.0 就是基于 Fortran 90 标准的编译器，下面简要介绍其操作方法，更完整的描述见上机手册。

一、编译器的介绍

编译器可以把 Fortran、C 等高级语言翻译成机器码，也就是常常看到的扩展名为 EXE、COM 的文件。也有一些语言不需要编译器就可以运行，例如 Visual Basic、Delphi 等，这些语言被称为解释语言，早期的解释语言要放在解释器下才能运行。现在的解释语言通常提供两种运行格式，第一种格式必须在解释器下才能运行，另一种格式与 Fortran 和 C 一样，可以编译成运行文件独立运行。

相同的程序代码，让不同厂商的编译器来编译，所生成的运行文件也不会完全相同。编译器的好坏主要就取决于翻译结果。比较编译结果好坏的衡量标准有：翻译要正确；看谁翻译出来的执行文件的运行效率高，以及翻译出来的执行码的长短等。

程序员通常还会注意编译过程所花费的时间、编译器所提供的调试 (DEBUG) 工具的完整程度、使用帮助是否清楚等。

各家的编译器除了要支持高级语言的标准语法之外，通常都还会有自己的扩充功能，某些应用上这些扩充功能是必要的。在制定程序语言标准时，通常不会在输出方面定义太多功能，这些功能需要经过个编译器自行提供。如想要用 FORTRAN 来编写 Windows 图形接口的程序，就要使用编译器所提供的扩充链接库。

有关详细的编译器的使用方法，如调试工具 (DEBUG) 等参见上机手册。

二、Microsoft Fortran PowerStation 4.0

本书使用 Microsoft 的 Fortran PowerStation 4.0 来做实际的范例。

1. 系统安装与启动

Microsoft 的 Fortran PowerStation 4.0 的安装和 Windows 的大多数程序相同 这里不再介绍。在此只有三点建议：

(1) 除非是硬盘空间不够，否则请务必安装帮助文件。

(2) 最好不要使用默认的目录位置来安装，默认目录会安装在 C:\Program Files\Fortran PowerStation 4.0，建议可以直接放在根目录下，例如使用 C:\MSDEV 目录。

(3) 安装到 90%时，会出现对话框询问是否要更新一些环境参数的值以方便命令行下使用，建议单击 OK 按钮更新。

Fortran 被组合在一个叫做 Microsoft Studio 的图形接口开发环境中，Microsoft Studio 提供一个同一的使用接口，这个接口包括文字编辑功能、Project 的管理、调试工具等。而编译器则是使用类似 PlugIn 的方法组合到 Microsoft Studio 中。Microsoft 的 Fortran PowerStation 4.0 除了完全支持 Fortran 90 的语言外，扩展功能方面提供完整的 Windows 程序开发工具，专业版还内含 IMSL 数值链接库。另外它还可以和 Visual C++ 直接互相链接使用，也就是把 Fortran 和 C 语言的程序代码混合编译成同一个运行文件。

安装好 Microsoft 的 Fortran PowerStation 4.0 后 运行菜单中的 Microsoft Developer Studio 就可以开始编译 Fortran 程序了。这时将看到图 2-5 所示的 Fortran PowerStation 4.0 工作窗口。整个窗口由菜单栏、工具栏、项目管理区、文本编辑区和结果输出区 5 部分组成。

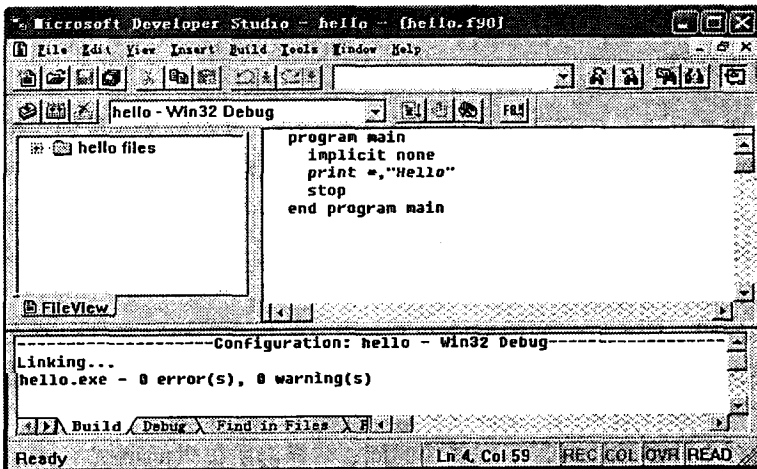


图 2-5 Fortran PowerStation 4.0 工作窗口

2. 上机步骤

Fortran 程序上机运行要经过编辑、编译、连接和运行 4 个步骤。在 Fortran PowerStation 4.0 环境下，具体操作如下：

(1) 建立一个项目 (Project)。在应用 Fortran PowerStation 4.0 开发 Fortran 程序时，首先要建立一个项目。项目中存放了建立特定程序所需要的全部信息，这些信息包括源程序文件

名及其相互关系、所需库文件清单、编辑器、连接器和建立该程序所需要的其他信息的清单。

建立项目的具体步骤是在 Fortran PowerStation 4.0 工作窗口的 File 菜单中选择 New 命令，再在出现的 New 对话框中选择 Project Workspace，接着输入工程名和存放该工程的文件夹名称，在 Type 列表框中选择 Console Application，最后单击 Creat 按钮。

(2) 建立源程序文件。具体步骤为在 Fortran PowerStation 4.0 工作窗口的 File 菜单中选择 New 命令，再在出现的 New 对话框中选择 Text File，接着单击 OK 按钮。这时文本编辑区域为空白区域，可以在其中输入、编辑修改源程序，编辑完成后将源程序保存到存放项目的文件夹中。在 Fortran PowerStation 4.0 中，源程序的缺省扩展名是.F90。

(3) 将源文件加入项目。建立和保存了源程序文件后，要将它加入到项目中去。具体步骤为在 Fortran PowerStation 4.0 工作窗口的 Insert 菜单中选择 File into Project 命令，再在出现的 Insert Files into Project 对话框中选择要加入到项目的文件，接着单击 Add 按钮。

(4) 编译和运行。在 Fortran PowerStation 4.0 工作窗口的 Build 菜单中选择 Compile、Build、Execute 等命令可以完成源程序的编译、连接和运行。

复习思考题

1. 下列符号中为合法的 Fortran 90 标识符的有哪些？并解释非法的原因。

A123B	5YZ	COS (X)	U.S.S.R
M%10	X+Y	A.2	min*2
X C2	F (X)	'A'ONE	PRINT

2. Fortran 90 自由格式书写时，应注意哪些事项？

3. 简述 Fortran 语言的发展过程和特点。

4. 简述程序设计的基本步骤。

5. 简述 Fortran 程序的基本结构和书写格式。

6. 简述 Fortran 90 的程序组织结构。

7. 何谓源程序的固定书写格式和自由书写格式？Fortan 90 允许采用何种书写格式？

8. 简述 Fortran 程序的上机操作步骤。

9. Fortran 90 字符集有何规定？

10. 何谓关键字？判断下列名称，那些是 Fortran 90 的关键字？哪些不是关键字？PRINT, MAIN, TOTAL, COUNT, WHILE, QUARD, THEN, AS, IF, OTHER.