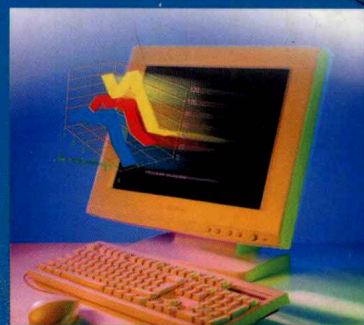




普通高等学校非计算机专业计算机基础系列教材



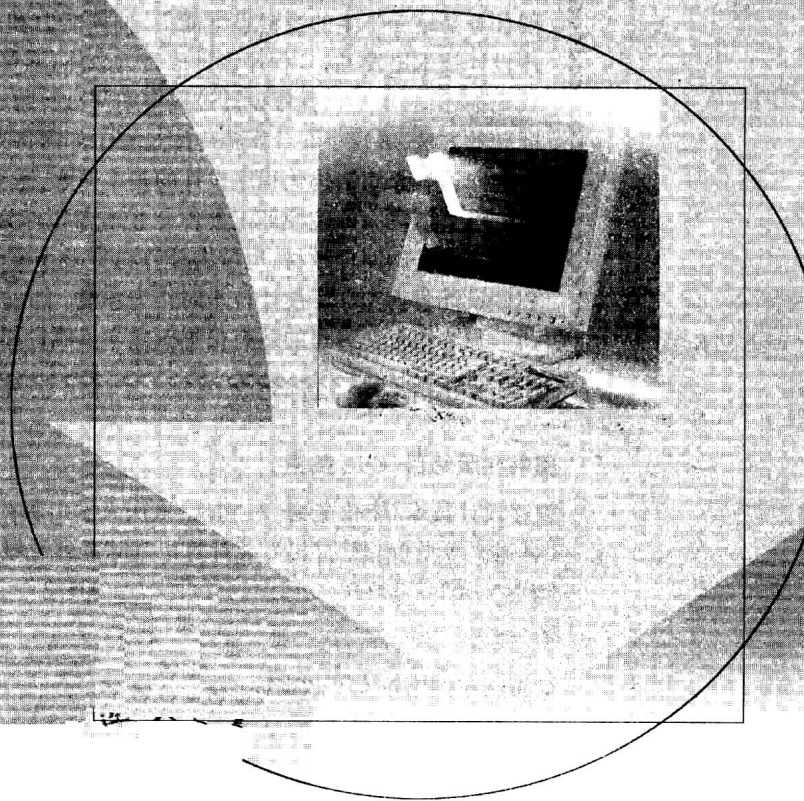
程 序设计技术

◆ 主 编 熊 壮
◆ 主 审 曾 一

重庆大学出版社



普通高等学校非计算机专业计算机基础系列教材



程 序设计技术

主 编 熊 壮
主 审 曾 一
编 者 张全和 陈 策
何 颖 刘慧君

重庆大学出版社

内容简介

程序设计技术和程序设计语言是大学计算机基础系列课程中的重要组成部分,其主要任务是培养学生的逻辑思维能力、抽象能力和基本的程序设计能力。本书从结构化程序设计技术出发,以 C 程序设计语言为载体,通过对典型实例的算法描述以及相应 C 语言代码描述,展现了在程序设计过程中如何对问题进行分析、如何组织数据和如何描述解决问题的方法,展现了在计算机应用过程中如何将方法和编码相联系的具体程序设计过程,进而向读者介绍结构化程序设计的基本概念、基本技术和方法。

本书可供高等院校理工类各专业本专科作为程序设计技术、程序设计语言或计算机软件技术基础课程教材,也可供计算机专业本专科学子以及计算机应用开发人员在学程序设计语言和程序设计技术时作为参考。

与本书配套编写的还有由重庆大学出版社出版的《程序设计实验指导书》。

图书在版编目(CIP)数据

程序设计技术/熊壮主编. —重庆:重庆大学出版社,2005.2

(普通高等学校非计算机专业计算机基础系列教材)

ISBN 7-5624-3339-9

I. 程... II. 熊... III. 程序设计—高等学校—教材 IV. TP311.1

中国版本图书馆 CIP 数据核字(2005)第 011794 号

普通高等学校非计算机专业计算机基础系列教材

程序设计技术

主 编 熊 壮

主 审 曾 一

责任编辑:王 勇 吴庆佳 版式设计:吴庆渝

责任校对:廖应碧 责任印制:秦 梅

*

重庆大学出版社出版发行

出版人:张鸽盛

社址:重庆市沙坪坝正街 174 号重庆大学(A 区)内

邮编:400030

电话:(023) 65102378 65105781

传真:(023) 65103686 65105565

网址:<http://www.cqup.com.cn>

邮箱:fxk@cqup.com.cn(市场营销部)

全国新华书店经销

四川外语学院印刷厂印刷

*

开本:787×1092 1/16 印张:24 字数:599 千

2005 年 2 月第 1 版 2005 年 2 月第 1 次印刷

印数:1—5 000

ISBN 7-5624-3339-9

定价:30.00 元

本书如有印刷、装订等质量问题,本社负责调换

版权所有,请勿擅自翻印和用本书

制作各类出版物及配套用书,违者必究。

编委会

顾问 吴中福 邱玉辉

主任 陈流汀

副主任 杨天怡 严欣平 张鸽盛

委员 (以姓氏笔画为序)

王世迪 邓亚平 程小平

杨国才 范幸义 洪汝渝

郭松涛 黄 勤 曾 一

谭世语 熊 壮 莫 堃



序言

计算机技术的飞速发展,加快了人类进入信息社会的步伐,改变了世界,改变了人们的工作、学习和生活,对社会发展产生了广泛而深远的影响。计算机技术在其他各学科中的应用,极大地促进了各学科的发展。不掌握计算机技术,就无法掌握最先进、最有效的研究开发手段,将影响到其所从事学科的发展。因此,计算机技术基础是 21 世纪高校非计算机专业大学生必须掌握的、最重要的基础之一。

经过多年的探索和实践,按“计算机文化基础”、“计算机技术基础”、“计算机应用基础”三个层次组织教学已被公认为高校非计算机专业计算机基础教学的基本模式。第一层次开设“计算机文化基础”课程,教学的主要任务是使学生掌握计算机基础知识和基本操作能力;第二层次开设“计算机软件技术基础”和“计算机硬件技术基础”课程,教学的主要任务是使学生掌握计算机软、硬件技术的基本知识和基本开发技术;第三层次按专业群开设“计算机信息管理基础”、“计算机辅助设计基础”、“计算机网络技术基础”、“计算机控制技术基础”等不同课程,教学的主要任务是培养学生应用计算机技术分析解决本学科及相关领域问题的能力。

为了适应计算机技术的飞速发展和广泛应用对高校非计算机专业人才培养提出的新要求,我们组织一批

长期从事计算机技术教学和科研的教师,编写了这套计算机基础教学系列教材。本系列教材有如下特点:

1. 适合于层次教学模式。系列教材内容覆盖了高校各类非计算机专业三层次计算机基础教学要求,既有适合理工类专业使用的,也有适合文经类专业使用的,各类专业都可从中选择到相应的教材。

2. 内容新。系列教材较好地反映了计算机技术的新发展,如《计算机文化基础》介绍了图形窗口界面和网络、多媒体基础;《计算机软件技术基础》介绍了软件基本概念和基本工具、结构化及面向对象程序设计的概念与方法、软件工程的基本思想和最先进的开发环境及平台;《计算机信息管理基础》以大型关系数据库管理系统为背景,介绍了关系数据库的基本知识和数据的构造方法以及网络技术在系统中的应用等等。

3. 强调应用和实用。非计算机专业的计算机基础教学以应用为目的,因此,本系列教材在编写上特别注意应用需要,强调实用性。主要课程教材都配有实验教程,基本知识理论讲深讲透,使用技术主要通过学生上机实验来掌握。

4. 便于自学。为了充分调动学生的学习主动性和能动性,本系列教材在写法上,既注意概念的严谨与清晰,又特别注意用易读易懂的方法阐述问题,应用举例丰富,便于自学。

总而言之,本系列教材的编写指导思想是:内容要新,要体现计算机技术的新发展和适应教学改革的要求;概念要清晰、通俗易懂,便于学生自学;应用性、实用性要强,切实在培养学生应用能力上下功夫;层次配套,可选择性强,适用面宽,既是普通高校非计算机专业本专科学学生教材,亦可作为高等教育自学教材和工程技术人员的参考书。

限于编者水平,系列教材的内容及体系难免有缺点错误,诚恳希望读者和专家给予指正。

编委会
2002年1月



前言

“程序设计技术”和“程序设计语言”是大学计算机基础系列课程中的重要组成部分,其主要任务是培养学生的逻辑思维能力、抽象能力和基本的程序设计能力。本书从结构化程序设计技术出发,以 C 程序设计语言为载体,通过对典型实例的算法描述以及相应 C 语言代码描述,展现了在程序设计过程中如何对问题进行分析、如何组织数据和如何描述解决问题的方法,展现了在计算机应用过程中如何将方法和编码相联系的具体程序设计过程,进而向读者介绍结构化程序设计的基本概念、基本技术和方法。

程序设计的主要任务之一就是需要将需要用计算机处理的实际问题抽象为数学模型,并设计出解决这个问题所需要的方法和步骤即算法。在处理实际问题的程序中应该包含两个方面的内容:要处理的对象和对这些对象的处理方法。对于那些要处理的对象,用数据和数据之间的关系即数据结构来表示,而对对象的处理方法用算法来表示。本书中用了较多实例向读者介绍了算法的基本概念、算法的特征以及算法的描述方法。

程序设计的另外一个主要任务就是用合适的计算机程序设计语言对所设计的算法进行编码处理,即编制程序。C 语言功能丰富、表达能力强、程序执行效率高、

可移植性好;C 语言既有高级计算机程序设计语言的特点,同时又具有部分汇编语言的特点,因而 C 语言具有较强的系统处理能力;C 语言是一种结构化程序设计语言,支持自顶向下、逐步求精的程序设计技术,通过 C 语言函数结构可以方便地实现程序的模块化;在 C 语言的基础之上发展起来的面向对象程序设计语言如 C++、Java、C#等与 C 语言有许多的共同特征,掌握 C 语言对学习进而应用这些面向对象的程序设计语言有极大的帮助。综上所述,C 语言应该是学习结构化程序设计技术的首选语言。

2 本书的主要内容分为 3 个大部分:第 1 部分主要讨论了算法描述的方法及若干算法描述实例。第 2 部分主要讨论了 C 程序设计语言和进行程序设计的主要技术,如数据的基本类型、程序控制结构、数组、指针、字符串、标准库、模块化程序设计技术及其在 C 语言中的实现等。对非验证性实例均给出了问题的分析方法、主体的算法描述和相应的 C 程序实现,以帮助读者尽快提高使用 C 语言解决实际问题的能力。第 3 部分主要讨论了 C 语言的应用问题,如线性数据结构的处理问题、常用的排序技术、常用查找技术以及 C 语言图形处理基本概念和 C 语言图形程序设计基本方法。

本书每章均提供了与内容紧密相关的习题以帮助读者巩固所学知识,习题的参考答案或算法提示、常用 C 标准库函数、ASCII 码对照表等常用教辅资料在与本书配套的《程序设计实验指导书》中提供。本书可供高等院校理工类各专业本专科作为程序设计技术、程序设计语言或计算机软件技术基础课程教材,也可供计算机专业本专科学生以及计算机应用开发人员在学程序设计语言和程序设计技术时作为参考。

本书由熊壮主编,曾一主审。各章节编写分工如下:熊壮,第 1 章、第 9 章;张全和,第 4 章;陈策,第 6 章、第 7 章;何频,第 2 章、第 3 章;刘慧君,第 5 章、第 8 章;全书由熊壮进行内容调整、修改,统一定稿。

本书在编写和出版过程中一直得到重庆大学教务处、重庆大学计算机学院和重庆大学出版社的支持和帮助,编者在此表示衷心的感谢。

限于编者水平,书中错误和不妥之处在所难免,恳请读者不吝指教。

联系地址:重庆,重庆大学计算机学院。

E-Mail: xiongz@cqu.edu.cn, xiongz@cqcnc.com

编者
2005 年 1 月



1 程序设计语言和算法描述

1.1 程序设计语言	(1)
1.1.1 程序设计语言概述	(1)
1.1.2 语言处理程序概述	(2)
1.2 算法特征及算法的描述方法	(3)
1.2.1 程序中数据的表示	(4)
1.2.2 算法的概念与特征	(4)
1.2.3 算法的描述方法	(7)
1.2.4 算法描述示例	(8)
习题 1	(12)

2 C 程序设计入门

2.1 C 语言的发展简史与特点	(13)
2.2 C 程序的基本结构	(15)
2.2.1 C 程序的基本结构	(15)
2.2.2 C 程序中的函数及其分类	(17)
2.3 C 语言的基本数据类型	(17)
2.3.1 C 语言数据类型概述	(17)
2.3.2 C 语言的基本数据类型	(20)
2.4 基本运算符和表达式	(25)

2.4.1	运算符的分类、优先级别和结合规则	(25)
2.4.2	基本运算符和算术表达式	(26)
2.4.3	赋值运算符和赋值表达式	(27)
2.4.4	自反运算符和自增、自减运算符	(28)
2.4.5	逗号运算符和逗号表达式	(30)
2.4.6	sizeof 运算符	(31)
2.5	不同类型数据混合运算及数据转换	(32)
2.5.1	混合运算的基本概念	(32)
2.5.2	不同数据类型隐式转换	(33)
2.5.3	不同数据类型显式转换	(33)
2.6	C 程序设计初步	(34)
2.6.1	C 语句概述	(34)
2.6.2	C 程序设计 7 大步骤	(36)
2.6.3	运行一个 C 程序的基本过程	(38)
2.6.4	基本输入输出函数	(38)
2.7	简单程序设计举例	(45)
习题 2		(48)

3 C 程序的控制结构

3.1	C 程序的控制结构概述	(52)
3.2	C 语言中的关系运算和逻辑运算	(53)
3.2.1	关系运算符和关系表达式	(53)
3.2.2	逻辑运算符和逻辑表达式	(54)
3.3	分支结构及其应用	(57)
3.3.1	分支结构的基本概念	(57)
3.3.2	C 语言的条件句与分支结构的实现	(58)
3.4	循环结构及其应用	(73)
3.4.1	循环结构概述	(73)
3.4.2	while 型循环结构	(74)
3.4.3	do ~ while 型循环结构	(75)
3.4.4	for 型循环结构	(77)
3.4.5	3 种循环语句的比较	(80)
3.4.6	循环的嵌套	(81)
3.4.7	循环语句的效率讨论	(84)
3.5	其他简单控制结构	(85)
3.5.1	goto 语句和标号语句	(85)
3.5.2	break 语句	(86)
3.5.3	continue 语句	(88)



3.6 算法与控制结构应用程序举例	(90)
3.6.1 控制结构程序设计举例	(90)
3.6.2 简单算法程序设计举例	(97)
3.7 程序设计风格讨论	(105)
3.7.1 空行	(105)
3.7.2 代码行	(106)
3.7.3 代码行内的空格	(107)
3.7.4 对齐	(108)
3.7.5 长行拆分	(109)
3.7.6 修饰符的位置	(109)
3.7.7 注释	(110)
习题3	(111)

4 函数与程序结构

4.1 模块化程序设计基本概念	(116)
4.1.1 模块化的基本思想	(116)
4.1.2 C语言实现模块化程序设计	(117)
4.2 函数的定义、声明和调用	(118)
4.2.1 函数的定义	(118)
4.2.2 函数的声明	(121)
4.2.3 函数的调用	(123)
4.2.4 函数调用时的参数传递	(126)
4.3 函数的嵌套调用	(128)
4.4 函数的递归调用	(133)
4.5 编译预处理	(137)
4.5.1 宏定义	(138)
4.5.2 文件包含	(140)
4.5.3 条件编译	(144)
4.6 C应用程序结构及其处理方法	(147)
4.6.1 变量的作用域和生存期的概念	(147)
4.6.2 自动变量及其作用域	(148)
4.6.3 寄存器变量	(149)
4.6.4 静态变量及其作用域	(150)
4.6.5 外部变量及其作用域	(151)
4.6.6 内部函数和外部函数	(156)
习题4	(157)

5 指针与函数

5.1 指针的概念	(161)
5.1.1 指针变量的定义	(161)
5.1.2 指针变量的引用	(163)
5.2 指针变量作函数的参数	(170)
5.2.1 指针变量作为函数的参数	(170)
5.2.2 指针变量与指针指向的变量	(172)
5.3 函数的指针与函数调用	(173)
5.3.1 用指向函数的指针变量来调用函数	(174)
5.3.2 指向函数的指针变量作函数参数	(176)
5.4 返回指针值的函数	(177)
5.5 使用指针的一些问题	(178)
习题 5	(179)

6 构造数据类型与指针

6.1 数组、指针及应用	(184)
6.1.1 数组的概念与定义	(184)
6.1.2 数组元素的引用和初始化	(186)
6.1.3 数组作为函数参数的使用	(192)
6.1.4 数组与指针	(193)
6.1.5 数组应用——排序数组	(196)
6.1.6 数组应用——查找数组	(203)
6.1.7 二维数组	(207)
6.1.8 二维数组与指针	(215)
6.2 字符串及应用	(220)
6.2.1 字符串的概念及定义	(220)
6.2.2 字符串的输入输出	(221)
6.2.3 字符串标准处理函数	(223)
6.2.4 指针数组和多级指针	(230)
6.2.5 命令行参数	(233)
6.3 结构体类型及应用	(237)
6.3.1 结构体类型定义	(237)
6.3.2 结构体变量的定义方法和成员引用	(238)
6.3.3 类型定义 typedef	(242)
6.3.4 结构体数组	(243)
6.3.5 结构体与函数	(245)



6.3.6 结构体与指针	(247)
6.4 共用体类型及其应用	(250)
6.4.1 共用体类型定义	(250)
6.4.2 共用体变量定义及成员引用	(251)
6.5 枚举类型及应用	(255)
6.5.1 枚举类型定义	(255)
6.5.2 枚举变量的定义和操作	(256)
习题6	(258)

7 位运算

7.1 计算机中数据表示方法	(261)
7.1.1 位与字节	(261)
7.1.2 计算机中数的表示方法	(262)
7.2 位运算及应用	(264)
7.2.1 “按位与”运算及应用	(264)
7.2.2 “按位或”运算及应用	(265)
7.2.3 “异或”运算及应用	(266)
7.2.4 “取反”运算及应用	(267)
7.2.5 “左移”运算及应用	(267)
7.2.6 “右移”运算及应用	(268)
7.2.7 有关位操作的自反算符	(269)
7.3 位运算举例	(269)
7.4 位段及应用	(273)
7.4.1 位段的概念和定义方法	(273)
7.4.2 位段的引用方法	(275)
习题7	(276)

8 文件

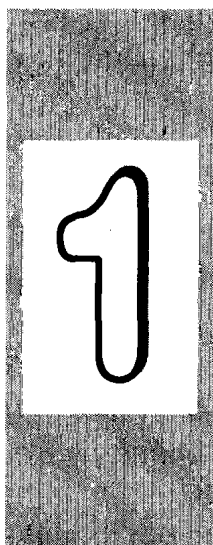
8.1 文件概念与文件类型指针	(278)
8.1.1 文件的概念	(278)
8.1.2 文件的分类	(278)
8.1.3 文件类型指针	(280)
8.2 文件的打开与关闭	(281)
8.2.1 文件的打开	(281)
8.2.2 文件的关闭	(283)
8.3 文件的读写	(283)
8.3.1 文件中单个字符的读写操作	(283)

8.3.2	文件中字符串读写操作	(290)
8.3.3	文件中的格式化读写操作	(292)
8.3.4	文件中的数据块读写操作	(293)
8.4	文件的定位和随机读写	(297)
8.4.1	文件的内部记录指针和文件定位	(297)
8.4.2	文件的随机读写	(299)
8.5	文件操作的错误检测	(300)
8.5.1	文件读写错误检测函数 <code>ferror</code>	(300)
8.5.2	清除文件错误标志函数 <code>clearerr</code>	(301)
习题 8		(301)

9 C 语言应用

9.1	线性数据结构及其处理技术	(306)
9.1.1	C 语言的自引用结构和存储分配	(306)
9.1.2	线性表的存储结构及基本运算	(310)
9.1.3	栈的基本概念和应用	(323)
9.1.4	队列的基本概念和应用	(328)
9.2	常用的排序技术	(335)
9.2.1	直接选择排序	(335)
9.2.2	直接插入排序	(338)
9.2.3	二分插入排序	(340)
9.2.4	冒泡排序	(342)
9.2.5	希尔排序	(344)
9.2.6	快速排序	(346)
9.3	常用查找技术	(348)
9.3.1	线性查找	(348)
9.3.2	二分查找	(350)
9.4	图形程序设计基础	(352)
9.4.1	C 语言图形处理基本概念	(352)
9.4.2	C 语言常用图形库函数的使用	(353)
9.4.3	C 语言图形程序设计基本方法	(364)
习题 9		(368)

主要参考文献



程序设计语言和算法描述

1.1 程序设计语言

1.1.1 程序设计语言概述

从某种意义讲,计算机应用问题就是让计算机能够理解执行人类解决某种问题的方法,从而达到解决实际应用问题的目的。为了使计算机能够理解人的意图,就必须解决人类和计算机相互交流的问题,将人解决问题的思路、方法和手段通过某种计算机能够理解的形式告诉计算机,使得计算机能够根据人的指令去一步一步的工作进而完成某种特定的任务。这种人和计算机之间交流的语言就称为计算机程序设计语言。从计算机发明至今,随着计算机硬件技术和软件技术的发展,计算机程序设计语言经历了机器语言、汇编语言、面向过程的程序设计语言以及面向对象的程序设计语言等阶段。

1) 机器语言

在计算机系统中,一条机器指令规定了计算机系统的一个特定动作。一个系列的计算机在硬件设计制造时就用若干指令规定了该系列计算机能够进行的基本操作,这些指令一起构成了该系列计算机的指令系统。在计算机应用的初期,程序员使用机器的指令系统来编写计算机应用程序,这种程序称为机器语言程序。编写机器语言程序要求程序员对计算机系统的所有细节如指令系统、存储容量等都要十分熟悉,使得计算机程序设计成为了一种非常艰苦的工作。使用机器语言编制的程序,由

于每条指令都对应计算机一个特定的基本动作,所以程序占用内存少、执行效率高。但其缺点也十分明显,例如:编制程序的工作量大,容易出错;依赖具体的计算机系列,因而程序的通用性、移植性都很差。

2) 汇编语言

为了解决使用机器语言编写应用程序所带来的一系列问题,人们首先想到了使用助记符号来代替不容易记忆的机器指令。这种用助记符号来表示计算机指令的语言称为符号语言,亦称为汇编语言。在汇编语言中,每一条用符号来表示的汇编指令与计算机机器指令一一对应,但记忆难度大大减小了,不仅易于检查和修改程序错误,而且指令、数据的存放位置可以由计算机自动分配。用汇编语言编写的程序称为源程序,计算机不能直接识别和处理源程序,必须通过某种方法将它翻译成为计算机能够理解并执行的机器语言,执行这个翻译工作的程序称为汇编程序。

使用汇编语言编写计算机程序,程序员仍然需要十分熟悉计算机系统的硬件结构,所以从程序设计本身上来看仍然是低效率的、繁琐的。但正是由于汇编语言与计算机硬件系统关系密切,在某些特定的场合如对时空效率要求很高的系统核心程序以及实时控制程序等,迄今为止汇编语言仍然是十分有效的程序设计工具。

2

3) 高级语言

计算机程序设计高级语言是一类接近于人类自然语言和数学语言的程序设计语言的统称。按照其程序设计的出发点和方式不同高级语言分为了面向过程的语言和面向对象的语言,如 Fortran 语言、C 语言等都是面向过程的语言,到目前为止还具有强大的生命力,是设计中小型计算机程序的优秀程序设计语言;而以 C++、Smalltalk 等为代表的面向对象语言与面向过程语言有着许多不同,这些语言支持“程序是相互联系的离散对象集合”,这样一种新的程序设计思维方式,具有封装性、继承性和多态性等特征。

高级语言按照一定的语法规则,由表达各种意义的运算对象和运算方法构成。使用高级语言编写程序的优点是:编制程序相对简单、直观、易理解、不容易出错。高级语言是独立于计算机的,因而用高级语言编写的计算机程序通用性好,具有较好的移植性。

用高级语言编写的程序称为源程序,计算机系统不能直接理解和执行,必须通过一个语言处理系统将其转换为计算机系统能够认识、理解的目标程序才能为计算机系统执行。

1.1.2 语言处理程序概述

如前所述,无论使用哪种计算机高级程序设计语言编写的计算机程序都称为源程序,它们都不能为计算机直接识别、理解和执行,都必须通过某种方式转换为计算机能够直接执行的机器语言。这种将高级程序设计语言编写的源程序转换到机器目



标程序的方式有 2 种:解释方式和编译方式。

解释方式下,计算机对高级语言书写的源程序一边解释一边执行,不形成目标文件和执行文件。

编译方式下,首先通过一个对应于所用程序设计语言的编译程序对源程序进行处理,经过对源程序的词法分析、语法分析、语意分析、代码生成和代码优化等阶段将所处理的源程序转换为用二进制代码表示的目标程序。然后通过连接程序处理将程序中所用的函数调用、系统功能调用等嵌入到目标程序中,构成一个可以连续执行的二进制执行文件。调用这个执行文件就可以实现程序员在对应源程序文件中所指定的相应功能。用编译方式处理高级语言源程序的简单过程如图 1.1 所示。

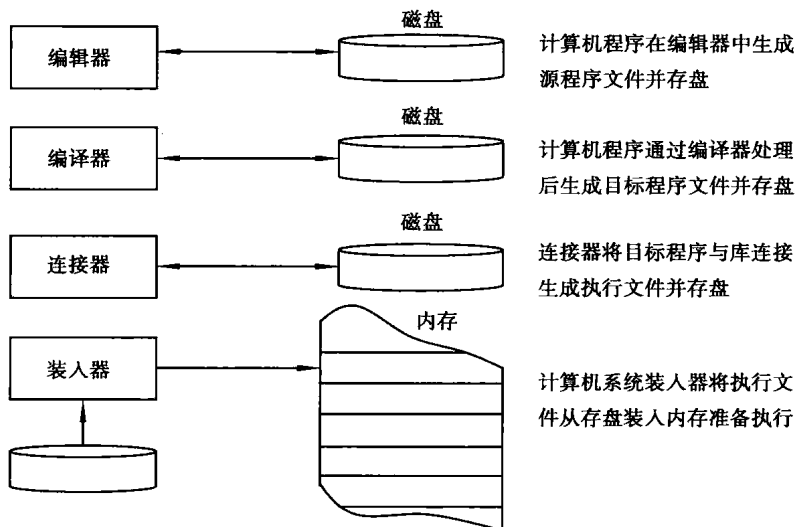


图 1.1 高级语言源程序处理过程

1.2 算法特征及算法的描述方法

计算机程序设计的目的是通过计算机解决实际问题。程序设计主要有两个方面的任务,首先是将需要用计算机处理的实际问题抽象为数学模型,并设计出解决这个问题所需要的方法和步骤,即算法。然后用合适的计算机程序设计语言对所设计的算法进行编码处理,即编制程序。在处理实际问题的程序中应该包含两个方面的内容:要处理的对象和对这些对象的处理方法。要处理的对象用数据和数据之间的关系即数据结构来表示,对对象的处理方法用算法来表示。著名计算机科学家沃思(Nikiklaus Wirth)用公式:“程序 = 数据结构 + 算法”诠释了程序设计的本质。