

全国高职高专教育规划教材

程序设计基础

■ 余苏宁 主编

■ 赵 杰 乌云高娃 副主编

全国高职高专教育规划教材

程序设计基础

Chengxu Sheji Jichu

余苏宁 主 编

赵 杰 乌云高娃 副主编



高等教育出版社·北京
HIGHER EDUCATION PRESS BEIJING

内容提要

本书系统地介绍了 C 语言和算法设计的基础知识、结构化程序设计的基本理论和方法,通过大量生动有趣的实例,深入浅出地讲述了基本问题的求解方法和算法设计的基本方法。以极具特色的例子介绍了程序调试技术和 Windows 平台上基于开放式 Allegro C 库的绘图、动画和游戏编程知识和技术。全书共 8 章,主要内容包括:程序与算法基础、C 语言基础、算法的基本知识、结构化程序设计、基本问题的求解方法、算法设计的基本方法、程序调试和 Allegro 绘图、动画及游戏编程等。附录中给出了 VC6 集成开发环境的安装、使用和调试方法,以及 C 语言常用语法提要等。

本书可作为应用性、技能型人才培养的各类教育“程序设计基础”课程的教学用书,也可供各类培训、计算机从业人员和爱好者参考使用。

图书在版编目(CIP)数据

程序设计基础/余苏宁主编. —北京:高等教育出版社, 2012. 2

ISBN 978 - 7 - 04 - 033633 - 7

I. ①程… II. ①余… III. ①C 语言 - 程序设计 - 高等职业教育 - 教材 IV. ①TP312

中国版本图书馆 CIP 数据核字(2011)第 273253 号

策划编辑 杜 冰
插图绘制 尹 莉

责任编辑 杜 冰
责任校对 刘 莉

封面设计 杨立新
责任印制 胡晓旭

版式设计 马敬茹

出版发行 高等教育出版社
社 址 北京市西城区德外大街 4 号
邮政编码 100120
印 刷 北京四季青印刷厂
开 本 787mm × 1092mm 1/16
印 张 24.25
字 数 590 千字
购书热线 010 - 58581118

咨询电话 400 - 810 - 0598
网 址 <http://www.hep.edu.cn>
<http://www.hep.com.cn>
网上订购 <http://www.landraco.com>
<http://www.landraco.com.cn>
版 次 2012 年 2 月第 1 版
印 次 2012 年 2 月第 1 次印刷
定 价 34.50 元

本书如有缺页、倒页、脱页等质量问题,请到所购图书销售部门联系调换

版权所有 侵权必究

物 料 号 33633 - 00

前 言

“程序设计基础”是为计算机系一年级新生开设的专业基础课程,其目的是帮助学生突破算法设计难关,培养其对一些常见基本问题进行分析、算法设计、编程和调试的能力,为今后学习其他专业课程奠定良好的基础。本课程的内容主要分为两大部分,其一是算法设计,其二是计算机语言。但重点是算法设计,语言为算法设计服务。

本教材是根据编者多年从事高职教学的经验、针对高职学生的特点、在不断总结教学改革经验的基础上写作出来的,力求体现如下特色。

- 遵循高职教育理念,注重培养应用型技术人才

本书遵循理论知识以必需、够用为度,重在技术能力培养的原则,根据职业岗位的需求安排教学内容,不追求面面俱到,但求切实掌握生产应用所必需的基本理论和技能,对于必须掌握的理论知识力争做到使读者知其然也知其所以然,注重应用型技术人才基础知识和基本技能的培养。

- 实例引导,循序渐进,项目驱动

本书以“实例引入,改造拓宽,项目综合”的方式组织内容,以一个或多个从实际中归纳总结出来的实例引入教学内容,实例反映基本、核心内容,并尽可能简单,以降低学习门槛。实例大多给出了程序代码和运行结果,这被称为“活代码”(live-code),它使教材由“静态”变成了“动态”,变抽象为具体,更便于理解和掌握。还有些实例则鼓励学生自己动手编写代码和上机调试,并通过对实例进行改造、拓宽以及项目综合等方式,提高他们对问题进行分析处理和综合应用的能力。

- 目标明确,取材适当

本教材秉承“两个内容、一个重点”的宗旨,针对刚刚入校的高职学生的特点,精心挑选实例,合理组织内容。本教材基于如下理由选用C语言作为实现算法、编写程序的语言。首先,C语言实用性较强,在系统设计、应用设计、硬件设计等方面都有广泛的应用;其次,C语言是一种结构化的语言,有丰富的数据结构和较强的表达能力,便于教学和应用;再次,C语言能为后续课程如C++、Java以及面向对象技术的学习奠定基础。

但是也应看到,C语言灵活多变,较难学习。为此,本教材本着“服务算法,基本实用,降低难度”的原则选取C语言的一个基本子集作为本教材的教学工具,不包括结构、联合、文件等较高级的内容,对于数组和函数都只涉及简单应用,数组只用到二维。对指针数组、返回指针的函数等都不做要求。对于指针,只在第8章中应Allegro C库的要求才用到,但不做更多的要求,只要会用就行。对位运算以及较复杂的运算表达式也不做要求。使所选用的C子集能很好地服务于本课程的既定目标,使学生既能用它来编写程序实现算法,而又不会过多地将精力陷于学习C语言的语法细节中,从而将主要精力集中到学习算法设计上,学好算法,尽快完成由不会编程到会编程的飞跃。在后续的C语言和数据结构课程中,还会进一步学习C语言,这对于分散难点、循序渐进地学好C语言也是大有好处的。

- 实用有趣,通俗易懂

求知欲望是学习的驱动力,而兴趣是求知欲的导火索,本教材所选实例来源于日常生活和工作,科学实用,有启迪性,益智,通俗易懂。其中包括大量有趣、经典和极具动感的例子(如字符动画、人机游戏、警察抓小偷、迷宫探索、哥德巴赫猜想、百钱百鸡、物有几何、 n 皇后、约瑟夫问题、音乐图像展示器、会飞的小鸟、经典排序算法动画、汉诺塔游戏等),生动有趣,能极大地激发学生的学习兴趣,变被动学习为主动学习。

- 反映技术的新进展

本教材不仅融入了C语言最新标准C99的有关内容,采用了VC6可视化的开发平台,给出了基于VC6集成调试器、能体现动态调试特点的调试用例,而且引入了Windows平台上基于开放式Allegro C库的绘图、动画和游戏编程的有关知识和技术。本教材的所有例程都可以在VC6以上的编译器中编连运行,之所以选VC6而不是更高版本作为本教材的开发平台,是因为相对来说VC6难度较低,使用也方便。学会用VC6,再学更高版本的编译器就容易得多了。除此之外,本教材的所有例程也可以在Dev-C++、MinGW等较新的开发平台上编连运行。Allegro C库在这些开发平台上的安装和使用也都大同小异。

结构化定理可以说是所有程序设计技术理论的基础,虽说它已经不是什么新东西了,但是,向高职学生介绍结构化定理的相关内容和它的来龙去脉,可以说是一次新的尝试。

除以上特点外,本教材还有结构合理、内容安排适当、由浅入深、循序渐进的特点,在内容叙述上思路清晰、概念清楚、步骤明确、语言流畅、深入浅出。

本教材内容包括程序与算法基础、C语言基础、算法的基本知识、结构化程序设计、基本问题的求解方法、算法设计的基本方法、程序调试等。其中,算法的基本知识、基本问题的求解方法和算法设计的基本方法是重点,每一章的内容也有主次之分,教学中应该有所侧重。对于算法描述工具,本教材给出了6种,重点要求掌握一到两种,设计过程中建议使用“伪C”,因为它书写、修改方便,是C语言和自然语言的组合,弱化语法,注重解题思路的顺畅发挥,这对于初学者是至关重要的。图形工具对于表现算法的逻辑关系是十分有用的,建议掌握一种图形工具。基本问题的求解方法一章中给出的是一些常见、基本问题的算法,是算法入门内容。算法设计的基本方法一章中重点是枚举法和归纳法,分治法和回溯法要求基本了解。学习和模仿一些成熟算法,特别是一些经典算法是初学者学习算法的必由之路,本教材中给出了大量的成熟算法和经典算法。对第8章Allegro绘图、动画及游戏编程的内容不做要求。教师可以少讲甚至不讲,根据学生的情况提出弹性要求。相信这一章实例程序本身会对他们的学习产生较大的影响。

程序调试技能是一个程序设计人员最重要的基本素质之一。本教材在第7章中,由简单到复杂、由静态到动态给出了3个极具特色的例子,能引导读者循序渐进地掌握程序调试的基本知识和技能。这部分内容排在第7章,并不等于它必须排到课程的后面去学习,实际上只要开始编程,就可以开始这方面内容的学习,可以将这方面内容分散穿插到其他章节和例程中去学习,并结合实际进行运用。能较早地学习和使用这方面的知识和技能,就能较早地使其惠及整个课程的学习,也有利于学生对该技能的熟练掌握。

外语知识和算法知识、计算机语言知识一样,都是一个程序设计人员必备的基本知识,本教材对一些重要概念和名词给出了英文注释,应要求学生掌握。除此之外,还应指导学生学会使用联机帮助,看懂屏幕上的英文信息。这对于今后使用、维护计算机系统,从事计算机编程、调试和

检测工作都是大有好处的。

在本课程的教学中还应注意如下问题:首先,不要将本课程与本科院校计算机专业开设的“算法设计”之类的课程等同起来,因为两者的目标是不同的。其次,不要将本课程与本科院校非计算机专业开设的“程序设计基础”之类的课程等同起来,因为后者是为已学过计算机编程的非计算机专业学生进一步学习软件设计方面较深入的知识开设的一门课程,与高职学生当前的情况大不相同,不能生搬硬套这些课程的内容和要求。

本书由深圳职业技术学院的余苏宁主编,深圳职业技术学院的赵杰和乌云高娃为副主编,焦作大学信息工程学院的丰树谦老师参编。第1、2章由丰树谦编写,第3、6章由赵杰编写,第4、5章由乌云高娃编写,附录由赵杰和丰树谦编写,其余各章由余苏宁编写,余苏宁负责全书的统稿。此外,王静霞、肖丹、孙甄、余轶、孙贤在文稿的录入和整理、例程的调试和审核方面做了大量的工作,在此一并表示衷心的感谢。

需要本书电子教案、例程源代码、Allegro C库以及相关资料的读者可以通过发送邮件至 gjs-gzjsj@163.com 索取,也可以登录高等教育出版社高职高专教学资源网(<http://hv.hep.com.cn>)获取。

由于编者水平有限,书中难免有错漏之处,请广大读者批评赐教,编者的 E-mail 是 szysnysn@sina.com。

编 者
2011 年 12 月

目 录

第 1 章 程序与算法基础	1	2.3.4 基本类型	33
1.1 程序与算法	1	2.3.5 数值型数据的混合运算	40
1.1.1 实例程序演示	1	2.4 运算符和表达式	43
1.1.2 程序与算法	4	2.4.1 算术表达式	44
1.2 简单程序实例	5	2.4.2 赋值表达式	46
1.3 C 程序的编辑、编译、连接和 执行	5	2.4.3 关系表达式	47
1.3.1 C 程序的编辑、编译、连接和 执行概述	6	2.4.4 逻辑表达式	48
1.3.2 编程工具的选择	7	2.4.5 条件表达式	49
1.3.3 C 程序的开发过程	8	2.4.6 逗号表达式	51
1.4 实例程序的解读	12	2.4.7 算术、关系、逻辑、赋值 运算符的混合使用	51
1.5 系统生成的文件和程序配置	15	2.5 格式化输出输入函数	53
1.6 程序设计语言基础知识	16	2.5.1 格式化输出函数 printf	53
1.6.1 机器语言	17	2.5.2 格式化输入函数 scanf	57
1.6.2 汇编语言	17	习题 2	59
1.6.3 高级语言	18	第 3 章 算法的基本知识	62
1.7 C 语言的产生和特点	19	3.1 算法的基本概念	62
1.7.1 C 语言的产生	19	3.2 算法的特性与要素	63
1.7.2 C 语言的特点	21	3.2.1 简单算法实例	63
习题 1	23	3.2.2 算法的特性	64
第 2 章 C 语言基础	24	3.2.3 算法的基本要素	65
2.1 会动的字符	24	3.3 怎样表示一个算法	66
2.2 C 语言的基本字符、标识符 和关键字	25	3.3.1 用自然语言描述算法	66
2.2.1 基本字符	26	3.3.2 用流程图描述算法	66
2.2.2 标识符	27	3.3.3 结构化程序与结构化定理	68
2.2.3 关键字	27	3.3.4 用 N-S 图描述算法	72
2.2.4 特定字	28	3.3.5 用 PAD 图描述算法	73
2.3 数据与数据类型	28	3.3.6 用伪代码描述算法	75
2.3.1 猜生日游戏	28	3.3.7 用计算机语言表示算法	79
2.3.2 数据类型分类	31	习题 3	80
2.3.3 常量和变量	31	第 4 章 结构化程序设计	81
		4.1 结构化程序设计概述	81
		4.2 顺序结构程序设计	82

4.2.1	C 语句	82	5.1.1	整除的判断、整数的 判断	148
4.2.2	顺序结构程序设计	83	5.1.2	奇偶判断	149
4.3	选择结构程序设计	83	5.1.3	数字分离	150
4.3.1	if 语句	84	5.1.4	素数判断	151
4.3.2	if - else 语句	84	5.1.5	素因数分解	152
4.3.3	if 语句的嵌套和 else if 语句	87	5.1.6	局部截取	153
4.3.4	由条件表达式构成的选择 结构	91	5.1.7	最大公约数和最小 公倍数	154
4.3.5	switch 语句	92	5.1.8	综合举例	156
4.4	循环结构程序设计	97	5.2	数学计算	160
4.4.1	while 循环语句	97	5.2.1	数值计算	161
4.4.2	do - while 循环语句	100	5.2.2	日期计算	168
4.4.3	for 循环语句	102	5.2.3	模拟	172
4.4.4	循环嵌套	105	5.2.4	随机数的应用	174
4.4.5	break 语句和 continue 语句	107	5.3	逻辑判断	177
4.4.6	goto 语句和标号语句	109	5.3.1	逻辑判断的方法	177
4.4.7	循环语句小结	110	5.3.2	逻辑判断与推理题	180
4.5	数组	111	5.4	数制转换	185
4.5.1	数组的定义和引用	111	5.4.1	二进制数转化为 十进制数	185
4.5.2	数组的初始化	112	5.4.2	十进制数转化为 二进制数	187
4.5.3	一维数组应用举例	113	5.5	数据的排序算法	188
4.5.4	字符数组	114	5.5.1	选择法	188
4.6	模块化程序设计	116	5.5.2	冒泡法	191
4.6.1	模块化程序设计概述	116	5.5.3	插入法	192
4.6.2	函数	117	习题 5		194
4.6.3	变量的存储属性	125	第 6 章	算法设计的基本方法	197
4.6.4	预处理	132	6.1	枚举法	197
4.7	自顶向下、逐步求精	137	6.1.1	枚举方法	197
4.8	结构化程序设计风格	142	6.1.2	枚举范围和枚举方法 的确定	216
4.8.1	标识符命名	142	6.1.3	用枚举法解不定方程	219
4.8.2	注释	143	6.2	归纳法	223
4.8.3	程序编写	144	6.2.1	递推	223
4.8.4	输入输出	145	6.2.2	递归	225
习题 4		145	6.3	分治法	239
第 5 章	基本问题的求解方法	147			
5.1	数字问题	147			

6.4 回溯法	249	8.7 缩放和定时器	309
习题6	260	8.7.1 缩放位图	311
第7章 程序调试	263	8.7.2 定时器	311
7.1 错误分类	263	8.8 速度控制和文本显示	313
7.2 编译错误的查找、定位 与排除	263	8.8.1 速度控制	314
7.2.1 利用编译器查找和定位 错误	264	8.8.2 文本显示	316
7.2.2 利用编辑器辅助查找和 定位错误	266	8.9 绘图及动画	318
7.3 连接错误的查找、定位与排除 ...	274	8.9.1 绘图基本技术	319
7.4 运行错误的查找、定位与排除 ...	274	8.9.2 动画基本技术	321
7.4.1 在程序中插入调试语句 ...	275	8.9.3 双缓冲技术	326
7.4.2 利用调试工具动态跟踪 调试程序	278	8.9.4 页切换技术	329
习题7	285	8.10 Allegro C 库动画实用例程	332
第8章 Allegro 绘图、动画及游戏 编程	289	8.10.1 经典排序动画程序	332
8.1 引言	289	8.10.2 汉诺塔动画程序	336
8.2 安装 Allegro C 库	289	习题8	339
8.3 Hello Allegro	290	附录A Visual C++ 6.0 集成开发环境 的安装和使用	340
8.3.1 在 VC6 中创建 Allegro 工程	290	A.1 Visual C++ 6.0 集成开发 环境的安装	340
8.3.2 添加源程序代码	291	A.2 VC6 的使用	343
8.3.3 在工程设置中添加 alleg.lib 库名	293	A.2.1 VC6 简介	343
8.3.4 编译连接和运行	294	A.2.2 菜单栏	345
8.3.5 程序解读	294	A.2.3 工具栏	346
8.4 图像展示器	295	A.2.4 在 VC6 中开发 C 程序	348
8.4.1 展示器锥形	295	A.3 Visual C++ 6.0 集成调试器 的使用	351
8.4.2 位图显示方法	297	A.3.1 工程配置	352
8.5 播放音乐	302	A.3.2 动态跟踪调试的一般 过程	353
8.5.1 音乐播放方法	304	A.3.3 断点设置	353
8.5.2 指定屏幕颜色	305	A.3.4 Debugger 的调试运行	357
8.5.3 中心对齐和剪裁	305	A.3.5 Debugger 的查看工具	358
8.6 显示多幅位图和键盘控制	306	附录B C 语言中的关键字	364
8.6.1 指定不同文件名的方法 ...	306	附录C C 语言运算符的优先级和 结合性	366
8.6.2 key 数组和 KEY 常量	307	附录D C 语言常用语法提要	369
		附录E ASCII 字符编码一览表	373
		参考文献	376

第1章

程序与算法基础

1.1 程序与算法

1.1.1 实例程序演示

为了对程序有一个感性的认识,下面首先演示几个简单的程序实例,学习完本课程之后,相信大家都能够编写这类程序。

演示程序 1 打印万年历

这是一个非常实用的小程序。这个程序也是初学编程者一个很好的综合练习,它用到了数组、逻辑运算、求模运算、条件表达式、条件语句、循环语句、自定义函数以及按要求的格式进行打印等方面的知识。该程序通过指定年份和月份,打印出该年该月的日历。例如,指定 2012 年 2 月,打印出的日历如图 1.1 所示,2012 是闰年,所以 2 月有 29 天。

```
Enter year month: 2012 2
      2012 Year  2 Month
SUN MON TUE WED THU FRI SAT
          1  2  3  4
5    6    7    8    9   10   11
12   13   14   15   16   17   18
19   20   21   22   23   24   25
26   27   28   29
```

图 1.1 2012 年(闰年)2 月日历

演示程序 2 二分检索的可视化输出和字符动画

“分治”是从现实世界中总结出来的一套处理大型复杂问题的有效方法。人们为了求解一个较大的问题或一个复杂的问题,常常将其划分成一些较小或简单的子问题进行求解。当这些子问题解决之后,把它们的解连接起来,便得到原问题的解。这是一种“各个击破”、“分而治之”的思想。

在数据检索中,二分检索(或称二分查找、折半查找等)算法是运用分治策略的典型例子。采用顺序查找算法时,若表长为 $n=1\,024$ 项,则最多要找 1 024 次,平均要找 512 次。但若采用二分检索算法对有序表进行查找,最多只要找 10 次($2^{10}=1\,024$),次数明显降低,而且数据量越大,它与顺序查找的差别就越大,这是应用分治算法的一个很好的例子。

设 $(a_1, a_2, \dots, a_n)$ 为升序表,共含有 n 项。现欲从表中找出其值为 x 的项。二分检索的基本思想是:首先将 x 与表中间的一项 a_{mid} 比较,若 $x = a_{\text{mid}}$,则找到,查找过程结束;若 $x < a_{\text{mid}}$,则在表的左半部分中用同样的方法查找;若 $x > a_{\text{mid}}$,则在表的右半部分中用同样方法查找。以上过程不断重复,直至找到 x ,或得出 x 不在表内的结果为止。二分检索的具体内容见第 6 章。

本演示程序以可视化输出和字符动画的方式演示了二分检索过程。程序运行时,要求输入一个待查找的数,其后,程序显示出查找所在的数组的下标和数组元素的值以供观察,接着以箭头的闪烁和移动可视化地显示出以上查找的 3 步过程:查找中间项、查找左半表或右半表等,如果最终找到该数,则显示出它在数组中的下标,否则显示出“not found”的信息,图 1.2 是用二分法检索 34 的过程。

```
Enter a number between 1 and 73(any letter to quit):34↵
i: 0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24↵
a: 1  4  7 10 13 16 19 22 25 28 31 34 37 40 43 46 49 52 55 58 61 64 67 70 73↵
-----
1  4  7 10 13 16 19 22 25 28 31 34 37 40 43 46 49 52 55 58 61 64 67 70 73↵
  ↑                                     ↑                                     ↑↵
1  4  7 10 13 16 19 22 25 28 31 34↵
  ↑           ↑           ↑↵
                19 22 25 28 31 34↵
                  ↑   ↑   ↑↵
                    28 31 34↵
                      ↑ ↑ ↑↵
                        34↵
                          ↑↵

34 found in array element 11↵
```

图 1.2 二分检索 34 的过程

演示程序 3 排序动画

排序和搜索技术是程序中经常会使用到的基本技术,在程序设计领域中是一个专门的研究

课题,在数据结构课程中有专门的论述。本动画程序演示了 6 种经典排序方法的排序过程以及相关的统计资料,如排序时间、交换次数、比较次数等,是初学者学习程序设计和有关排序知识的一个很好的范例。图 1.3 展示的是快速排序的过程。

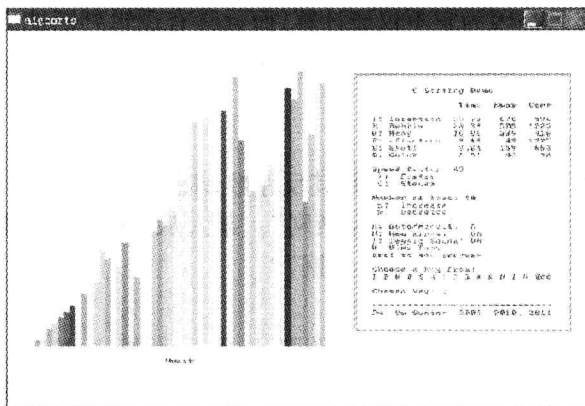


图 1.3 快速排序过程

演示程序 4 汉诺塔(Towers of Hanoi)游戏

汉诺塔程序是递归程序设计的经典范例。汉诺塔游戏据传是古代印度布拉玛庙里的僧侣们玩的一种游戏。游戏的装置是一块铜板,上面有 3 根柱子,左边柱子(A 柱)自上而下、由小到大的顺序地串有 64 个金碟,呈一个塔形。游戏要求把左边柱上的金碟全部移到右边柱子(C 柱)上,条件是,一次只能移动一个碟子,碟子只能在 3 个柱子之间移动,并且不允许将大碟放在小碟上面。

本演示程序以动画的形式模拟了该游戏过程中碟子的移动过程,对学习和理解递归过程是很有帮助的。同时对于研究和探索汉诺塔的非递归程序设计也是一个很有用的辅助工具。图 1.4 展示了给定碟子数为 10 时汉诺塔游戏的运行过程。

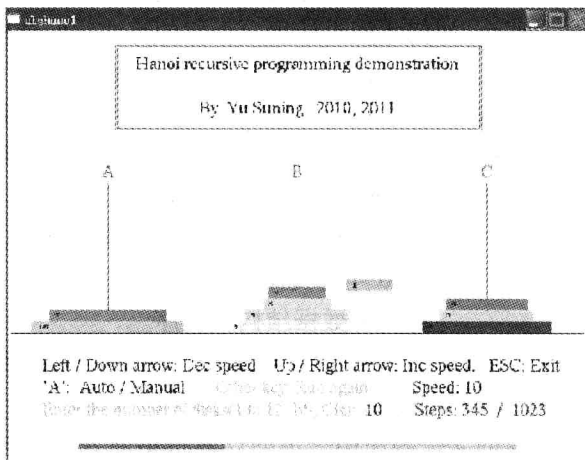


图 1.4 汉诺塔游戏动画($n=10$)

1.1.2 程序与算法

计算机被称为“电脑”，是因为它不仅能计算，而且具有逻辑判断能力，能代替人们去完成一些脑力劳动，如分析问题、控制过程、处理事务等，是一种具有“思维”能力的机器。下面是生活中一些常见的可以用计算机解决的问题。

问题 1：求自然数 n 的阶乘 $n!$ 。

问题 2：一个四位数，乘以 9 结果恰好是其反序数，求这个四位数。

问题 3：中国有句俗话：“三天打鱼两天晒网”。某人从 2000 年 1 月 1 日起开始“三天打鱼两天晒网”，问这个人今天是“打鱼”，还是“晒网”？

问题 4：在计算机屏幕上画一个圆。

问题 5：在一个集体婚礼上有 3 对情侣（新郎 $A、B、C$ ，新娘 $X、Y、Z$ ），他们都喜欢开玩笑，有人不知道谁和谁结婚，于是询问了他们中的 3 人，但得到的回答是这样的： A 说他的未婚妻是 X ；而 X 说她的未婚夫是 C ，但 C 说他将与 Z 结婚。他们的回答把这个人搞懵了，你能告诉他谁和谁结婚吗？

问题 6：编制一个学籍管理系统，完成学籍管理中的有关事项。

上面这些问题有的是数值计算问题，如问题 1~3，有的则不是数值计算问题，如问题 5、6，有的既有数值计算，又有非数值计算，如问题 4。总的来说，计算机处理的问题可以分为两大类，**数值计算问题**和**非数值计算问题**。不论是数值计算还是非数值计算问题，要想在数字计算机中解决，都要对其进行数字化，即用计算机所能“理解”的方式来描述问题才有可能用计算机去解决。通俗地说，就是必须为它编制程序。**程序**是控制计算机完成特定功能的一组有序指令的集合。计算机的工作就体现为程序的执行。现代数字计算机的基本原理和特征之一就是**存储程序**，正是由于程序存储在计算机的内存中，才使计算机能够自动地进行计算。

那么，如何编写程序？怎样才能编写出一个较好的程序呢？程序的灵魂是**算法**。所谓**算法**，就是解决一个问题的方法和步骤，而程序则是算法在计算机上的特定实现。要编写出好的程序，首先必须设计出好的算法。那么，如何才能设计出一个好的算法呢？要回答好这个问题，可不那么简单，但它是这门课程的目标。本书将通过大量的实例，由浅入深地介绍如何分析问题、设计算法、编写程序，学习这方面的基本知识、技能和技巧，以便让计算机能听从指挥，为人们服务。

从程序设计的角度来看，每个问题都涉及两个方面的内容：**数据**和**操作**。所谓**数据**是泛指计算机要处理的对象，包括数据的类型、数据的组织形式和数据之间的相互关系，这些又被称为**数据结构**（data structure）。所谓**操作**是指处理的方法和步骤，也就是**算法**（algorithm）。因此，著名计算机科学家沃思（Nikiklaus Wirth）1976 年曾提出一个著名的公式

算法 + 数据结构 = 程序

随着软件技术的进一步发展，人们进一步认识到，一个程序除了以上两个主要要素之外，还涉及采用什么样的设计方法（如结构化程序设计方法、面向对象程序设计方法等）去进行设计，并且还要有一定的计算机环境的支持和使用一定的计算机语言及工具才能完成。因此，有学者将上述公式拓展为

程序 = 算法 + 数据结构 + 程序设计方法 + 语言工具和环境

由此可见,作为一名程序设计人员,必须熟练掌握以上 4 个方面的知识和技能,并能综合运用这几个方面的知识和技能,这样才能设计出一个好的程序出来。本书会涉及上述 4 个方面的内容,但不可能全面介绍这些内容,它们都有专门的课程去研究。本书的重点在算法上,同时也会包含较多的程序设计语言方面的内容,但本书不是一本专门介绍算法的高深教材,也不是一本全面介绍某种语言的教科书,所涉及的语言知识以够用为度,服务于算法。本书的目的是帮助读者完成由不会编程到会编程的飞跃,为培养优秀的程序设计人员奠定坚实的基础。使读者通过本书的学习,能够对于生活、学习中常见的、不太复杂的问题进行分析,写出算法,编出程序,并在计算机上调试通过,计算出正确结果。

前面通过演示和说明已经使读者对程序的运行、功能和基本知识有了一些基本的感性认识 and 了解,为了更好地学习算法,下面还将进一步展示一些简单程序的代码,使读者对计算机程序的结构、组成以及如何编写程序等有更进一步的感性和理性认识。

1.2 简单程序实例

前面已经说明了本书以 C 语言作为实现算法、编写程序的语言,因此,下面介绍的简单程序的例子都是用 C 语言写成的。至于什么是计算机语言,什么是 C 语言,如何编写程序等,初学者一定会有很多问题,但随着下面的学习将会逐步掌握。

下面要展示的第一个实例是 C 语言的创始人 Brian W. Kernighan 和 Dennis M. Ritchie(简称 K&R)在他们的《C 程序设计语言》(The C Programming Language)一书中首先使用的“Hello, world”例程,这个程序虽然非常简单,但却反映了 C 语言的一些最基本的特点,因此被人们反复传用,几乎所有介绍 C 语言的教科书都用它作为入门的例子,只不过所显示的文字有所不同而已。

例 1 “Hello, world”例程,这个程序在计算机屏幕上显示一行信息“Hello, world”。

```
# include <stdio. h >
main( )
{
    printf(" Hello, world\n" );
}
```

这个例程只有上述 5 行代码,程序运行时将在计算机屏幕上显示一行信息:

Hello, world

在具体解释这个程序及其基本特点之前,首先得让它在计算机上运行起来,为此需要做下面这些工作。

1.3 C 程序的编辑、编译、连接和执行

在前一小节中,读者已经看到了用 C 语言编写的程序,那还只是“纸上谈兵”,如何能让它在

计算机上运行呢?这一节就来解决这一问题。在这一节中,将介绍如何将编写出来的程序转换成计算机上可执行程序的过程和方法。

1.3.1 C 程序的编辑、编译、连接和执行概述

现代数字电子计算机所能识别和执行的是由 0、1 所组成的二进制的指令,而不能识别和执行用高级语言编写的程序,有关计算机语言方面的知识后面有进一步的论述。为了使计算机能执行高级语言所写的程序,必须先用一种称为**编译程序**的软件,把程序翻译成二进制形式的程序,称为**目标程序**(target program),然后将该目标程序与系统的函数库和其他目标模块连接起来,形成**可执行的目标程序**(简称**可执行程序**),这样才能被计算机所识别和执行。相对于目标程序,用高级语言编写的程序被称为**源程序**(source program)。

因此,写在纸上或正在头脑中设想的源程序,要在计算机上执行,通常要经过下面的 4 个步骤。

1. 编辑

在计算机上书写(录入和修改)源程序并形成**源程序文件**,保存到计算机的磁盘中。**源程序文件**通常也简称为**源程序**或**源文件**(source file),文件名通常应该根据内容来命名,而文件的扩展名(也称后缀)则为“.c”(代码文件)和“.h”(说明性文件,通常称为头文件),例如上面的程序可命名为“hello.c”。对于本书,为了使程序与例题编号相对应而将其命名为“Ex1_01.c”。需要说明的是:一个较大的程序,往往是由多个源程序文件组成的。

用于编写(录入、修改和保存)程序的软件称为**编辑器**(editor)。编写的过程称为编辑。

2. 编译

将源程序翻译成目标程序并形成目标程序文件,扩展名通常为“.obj”或“.o”,多个源程序文件将生成多个目标程序文件。编译工作是由称为**编译器**(compiler)的软件完成的。

3. 连接

将编译生成的一个或多个目标程序文件与系统提供的库函数目标程序模块连接在一起生成一个完整的可执行程序,扩展名通常为“.exe”(在 UNIX 操作系统中为“.out”)。完成连接工作的软件称为**连接器**(linker)。

4. 执行

在计算机上运行可执行程序,观察和分析程序执行情况。开发阶段程序的执行通常是为了调试。给定一些数据,观察、跟踪程序的执行情况,如果程序执行不正确,则要分析原因,修改程序,直至对于所给定的一组数据执行都正确为止。用于程序调试的软件称为**调试器**(debugger)。调试后正确的程序才能交付使用。

对于 C 程序而言,以上过程如图 1.5 所示。图中,经过编辑,生成了 n 个源程序文件($fl.c, \dots, fn.c$)和若干个头文件($fi.h, \dots, fm.h$),这些源程序文件经编译后得到目标程序文件($fl.obj, \dots, fn.obj$),这些目标程序与库函数等目标模块经连接器连接后生成为一个可执行程序 $f.exe$,最后运行的是 $f.exe$ 。在编译、连接和执行过程中如果出现错误,则会重复前述过程直至没有错误为止。

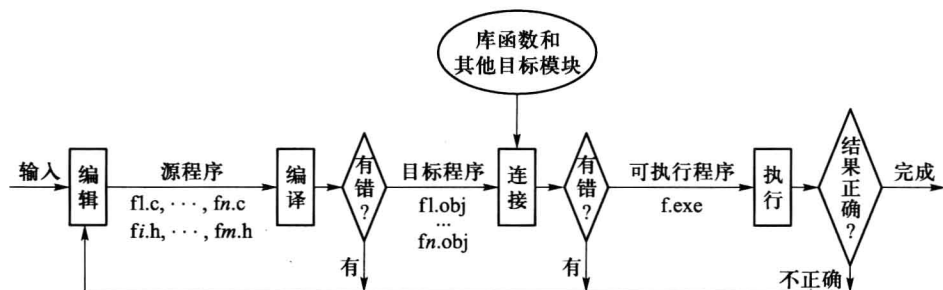


图 1.5 C 程序的编辑、编译、连接和执行过程

1.3.2 编程工具的选择

由上面的过程可以看出,整个过程需要用到多种工具。在集成开发环境(integrated development environment, IDE)未诞生之前,上面说明中的每一步都是由独立的工具分别完成的,极不方便,且效率低。1983年11月,创立不久的 Borland 公司,在以主设计师 Anders Hejlsberg 为首的科技人员的努力下,推出了第一个具有划时代意义的产品——Turbo Pascal 1.0,从此开创了 IDE 的先河。IDE 是一种一站式的开发环境,在 IDE 中可以对源程序进行编辑、编译、连接和调试运行。目前常用的 C/C++ 集成开发工具有 Turbo C、Borland C++、C++ Builder、Microsoft Visual C++ 以及 GNU C compiler(GCC)等。

Turbo C 是最早采用 IDE 的 C 编译器,其经典之作 Turbo C 2.0 曾被广泛使用,虽然自身的规模很小,但它的编译速度却很快,效率高、功能强、使用方便。由于它是早期针对 DOS 平台的产品,不支持鼠标,如果要在程序的字符串或注释中使用中文,则需要中文 DOS 的支持,这是它的不方便之处,不过国内已有 Turbo C 的汉化版,如 Turbo C 2.0 汉化版、Turbo C/C++ 3.0 汉化版以及 Windows 平台开发工具 WIN-TC 和 Turbo C For Windows 等,这些 C/C++ 语言集成开发环境是以 Turbo C 2.0/3.x 或 Visual C++ 6.0 为内核的,使用较为方便,属于免费软件,可以很容易地从 Internet 上得到。

继 Turbo C 2.0 之后, Borland C++ 3.1 是又一个很成功的集成开发工具。它既具有针对 DOS 平台的 C/C++ 编译器(BC.EXE),又具有针对 Windows 平台的编译器(BCW.EXE),能同时打开多个文档,支持鼠标操作。在中文 Windows 下,既可使用 BC.EXE 也可使用 BCW.EXE, BC.EXE 的用法与 Turbo C 2.0 基本上是一样的,而 BCW.EXE 可以在程序的字符串和注释中使用中文,并且运行时可以输出中文信息和输入中文字符串。Borland C++ 3.1 自身的规模也不大,并保持了编译速度快、效率高、功能强、使用方便的特点。

Microsoft Visual C++ 6.0 是一个面向对象、多文档、多窗口、能自动生成程序框架、功能强大、可视化的程序开发系统。在中文 Windows 下,它能够在程序的字符串和注释中使用中文,运行时也能够输出中文信息、输入中文字符串。同时由于它的编辑器功能强大,对文字的选取、复制、粘贴、剪切、移动等操作都可用鼠标来进行,其用法与 Microsoft Word 基本相同,所以使用起来十分方便。继 Visual C++ 6.0 之后,Microsoft 公司又推出了 Visual C++ 7.0(即 Visual C++ 2003)、Visual C++ 8.0(即 Visual C++ 2005)等,都是不错的 C/C++ 开发环境。就 C 语言开发

来说, Visual C++ 6.0 较为方便, 对于单文件程序的编译连接, 不需要事先创建工程, 系统会自动创建一个默认的工程, 而后两者都需要事先创建工程。

如果需要在 UNIX 或 Linux 平台上开发 C 程序, 则可以选用 GCC, GCC 早期的含义是 GNU C compiler, 即 GNU C 编译器, 现在已扩展为 GNU compiler collection, 即 GNU 编译器集, 除了 C 编译器之外, 还包括很多其他编译器, 如 C++、Objective-C、Java、Pascal、Ada、COBOL、FORTRAN 以及 Mercury 等语言的编译器。由于 GCC 是一个高效且可自由“使用、复制、修改和发布”的编译器, 因此现在, GCC 也有了许多 Windows 下的移植版本, 如 MinGW Developer Studio (MinGW 是指只用自由软件来生成纯粹的 Win32 可执行文件的编译环境, 它是 minimalist GNU on Windows 的简称)、Dev-C++ 等。

值得推荐的是 Dev-C++, 这是一款由 Bloodshed Software 公司推出的 Windows 下的 C/C++ 集成开发环境, 可以从地址为 <http://www.bloodshed.net/devcpp.html> 的网站上免费得到。

Dev-C++ 使用 MinGW32/GCC 编译器, 支持 C99 标准和多国语言, 包括简体中文语言界面以及技巧和提示, 入门简单, 使用方便, 能够满足初学者与编程高手的不同需求。如果要试验 C99 的一些独有的功能, 则建议安装 Dev-C++。

可以根据自己的具体情况来选用编程工具, 如果系统资源充足, 也可同时安装多个编程工具, 各取所长。

由于本书很多例子都使用了中文注释, 有些例子还在字符串中使用了中文, 因此, 本书将以 Visual C++ 6.0 作为开发环境。附录 A 给出了 Visual C++ 6.0 的安装和使用方法。如果使用 Turbo C 2.0, 本书第 1~7 章中的例子, 除了字符串和注释中的中文不能显示外, 其他功能并不受影响, 无须修改就可在 Turbo C 2.0 中编译、连接和运行。

1.3.3 C 程序的开发过程

下面以例 1.1 (程序名为 Ex1_01.c) 为例具体地谈一谈在 Visual C++ 6.0 集成开发环境中编辑、编译、连接和运行 C 程序的操作过程, 为了突出这一过程, 将 Visual C++ 6.0 的安装放在附录 A 中, 有需要的读者, 可参阅本书附录 A。

1. 启动 Visual C++ 6.0 集成开发环境

在 Windows 环境中, 选择“开始”→“程序”→“Microsoft Visual Studio 6.0”→“Microsoft Visual C++ 6.0”命令即可启动 Visual C++ 6.0 集成开发环境 (以下简称 VC6、集成开发环境、开发环境或环境, 在上下文清楚的情况下有时也直接称为系统), 如图 1.6 所示。

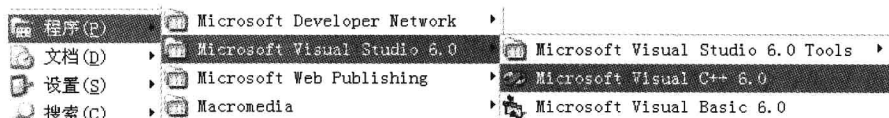


图 1.6 启动 Visual C++ 6.0 集成开发环境

另外, 为了方便启动 VC6, 可以在桌面上或“开始”菜单中用 Windows 的方法创建其快捷方式。