

国家精品课程主讲教材
教育部高等理工教育教学改革与实践项目研究成果

C语言程序设计

The C Programming Language

苏小红 王宇颖 孙志岗 等编著
蒋宗礼 主 审



高等教育出版社
HIGHER EDUCATION PRESS

国家精品课程主讲教材

教育部高等理工教育教学改革与实践项目研究成果

C语言程序设计

C Yuyan Chengxu Sheji

苏小红 王宇颖 孙志岗 等编著
蒋宗礼 主审



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

内容简介

本书是一本兼具趣味性和实用性的 C 语言程序设计教材。全书由 13 章组成,内容包括:为什么要学 C 语言,C 数据类型,简单的算术运算和表达式,键盘输入和屏幕输出,选择控制结构,循环控制结构,函数,数组,指针,字符串,指针和数组,结构体和共用体,文件操作。

本书以应用为背景,面向工程实践和编程能力训练,从实际问题出发,在案例的不断深化中逐步引出知识点,形成逻辑清晰的脉络和主线。案例内容紧密结合实践,举一反三,融会贯通,尤其侧重错误案例的分析和讲解,在任务驱动下,由浅入深、启发引导读者循序渐进地编写规模逐渐加大的程序,让读者在不知不觉中逐步加深对 C 程序设计方法的了解和掌握。本教材配有多媒体课件、例题和习题源代码以及编程题考试自动评分系统等教学资源,免费向任课教师提供。

本书适合作为高等学校各专业的 C 语言程序设计课程教材,也可作为从事计算机相关工作的科技人员、计算机爱好者及各类自学人员参考。

图书在版编目(CIP)数据

C 语言程序设计/苏小红,王宇颖,孙志岗等编
著. —北京:高等教育出版社,2011.4
ISBN 978-7-04-031909-5

I. ①C… II. ①苏…②王…③孙… III. ①C
语言-程序设计-高等学校-教材 IV. ①TP312

中国版本图书馆 CIP 数据核字(2011)第 030698 号

策划编辑 武林晓 责任编辑 武林晓 封面设计 王洋 责任绘图 尹莉
版式设计 余杨 责任校对 姜国萍 责任印制 尤静

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 北京铭成印刷有限公司
开 本 787×1092 1/16
印 张 28.5
字 数 640 000
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.landaco.com>
<http://www.landaco.com.cn>
版 次 2011 年 4 月第 1 版
印 次 2011 年 4 月第 1 次印刷
定 价 43.00 元(含光盘)

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 31909-00

序

程序设计是一门非常重要的计算机课程,其重要性不仅仅体现在一般意义上的程序编制,更体现在引导读者实现问题求解思维方式的转换——培养人们的计算思维能力。也正是由于这一点,才使得这门课程具有很高的难度。这也是为什么初学者总感到这是一门很难的课程的关键所在——需要实现思维方式的转换。突破了 this 个难点,一切将变得比较自然。这本教材以 C 语言为背景,从初学者的需求出发,在面向工程应用型计算机专业人才的培养方面进行了有益的探索,体现了“学生易学,教师易用,变应试为应用”的编写理念,形成了如下一些特点。

1. 以实际问题的求解过程为引导,讲授程序设计的基本方法,以结构化与模块化程序设计为核心,沿着数据结构从简单到复杂这条主线逐步展开,侧重对程序设计方法、程序调试方法相关知识和方法的介绍,并将软件工程相关的思想和方法渗透其中,提高读者程序编制的规范性。

2. 重点放在解决“程序设计”的核心问题上,以讲授程序设计为主,将 C 语言的有关语法有机地结合到程序设计中,避免了生硬枯燥的语法叙述,真正体现了“程序设计”,在“把 C 语言从应试课程转变为实践工具”上做出了可贵的探索。

3. 明显地体现出作者多年来在该门课程上的教学积累,在写作上努力追求面向初学者进行“讲授”的风格,行文流畅,语言带有一定的人文气息,努力贴近读者,深入浅出,通俗易懂,逻辑性强,形成该书独特的风格。

4. 将作者丰富的程序设计经验融入教材编写,按照初学者的需求,适时引导进行程序错误分析、测试与调试,将一些容易被忽略的而且对高水平 C 程序设计很重要的“点”逐一展现给读者,进一步落实“程序设计”教学的需求。

5. 选择了一些趣味性强、有吸引力的例子和话题以提高读者的学习兴趣,选择一些实用性强的例子和话题,以努力提高读者的工程实践能力。精选的“不断提升”的引导性例题、习题和实验题,以及贯穿全书的综合实例,起到了开拓读者思路、引导读者探究问题求解方法、激发读者程序设计兴趣的目的。

此外,本书按照教学的需要还配套建设了丰富的教学资源,如《C 语言程序设计学习指导》、程序源代码、多媒体课件、编程题考试自动评分系统、学习自测软件以及课程教学网站等,构成了 C 语言程序设计课程教学的完整解决方案。

希望该书能够得到众多读者的喜爱!



2011年2月

前 言

我们所使用的工具深刻地影响着我们的思维方式和思维习惯,进而也将深刻地影响着我们的思维能力。

——Edsger W. Dijkstra

学习程序设计本身是一件既充满挑战、更充满乐趣的事情,然而它之所以常常给人以枯燥乏味的感觉,那是因为没有有人帮助他们发掘出趣味来。本书力图用最简明的语言、最典型的实例以及最通俗的类比和解释将这种趣味性挖掘出来,带给读者以全新的学习体验,和读者一起欣赏 C 语言之美,领悟 C 语言之妙,体会学习 C 语言之无穷乐趣。

本书以应用为背景,面向工程实践和编程能力训练,从实际问题出发,在一个实际案例的不断深化中逐步引出相关知识点,借助任务驱动的实例将相关知 识点像珠链一样串联起来,形成逻辑清晰的脉络和主线。案例内容紧密结合实践,举一反三,融会贯通,在任务驱动下,由浅入深、启发引导读者循序渐进地编写规模逐渐加大的程序,让读者在不知不觉中逐步加深对 C 程序设计方法的了解和掌握。

在内容的指导思想,本书以 C 语言为工具介绍程序设计的基本思想和基本方法,却又不拘泥于 C 语言的基本语法知识,而是面向实际应用,把程序设计中最基本、最新、最有价值的思想和方法渗透到经典的 C 语言中,目的是使读者在学习 C 语言以后,今后无论使用什么语言编程,都具有灵活应用这些思想和方法的能力。

在内容编排上全书共分 13 章,内容包括:为什么要学 C 语言,C 数据类型,简单的算术运算和表达式,键盘输入和屏幕输出,选择控制结构,循环控制结构,函数,数组,指针,字符串,指针和数组,结构体和共用体,文件操作。

在内容编排上,全书内容注重教材的易用性。每章开头都有内容导读,指导读者阅读,每章结尾以表格的形式给出本章知识点小结和常见错误小结,帮助读者整理思路。本书既适合于程序设计的初学者,也适合于想更深入了解 C 语言的人。书中设计了很多思考题,并在每章的扩充内容中增加了一些有一定深度和开放性的内容供希望深入学习程序设计的读者选学和参考,力求做到内容有宽度、有深度、有高度。

在内容写作上,本书力图避免以往教材写作中常常出现的通病和问题,如“实例不实,为解释语法而设计”、“语法堆砌,只见树木不见森林”、“忽视错误程序的分析 and 讲解”等。因此编写的主要特色是注重错误程序的讲解和分析以及与软件工程内容的联系,在分析常见错误案例的过程中,讲解程序设计的基本方法、程序测试方法以及程序调试和排错方法,帮助读者了解错误发生的原因、实质、排错方法及解决对策。

全书程序采用统一的代码规范编写,并且在编码中注重程序的健壮性。全书例题、习题和实验题的内容选取兼趣味性和实用性,习题以巩固基本知识点和强化程序设计方法为目的,难度分成多个阶梯,包括:改写例题的编程题,模仿例题的编程题,趣味游戏类编程题。题型包括:侧重程序阅读理解能力训练的写出程序运行结果题和程序填空题,侧重程序调试和排错能力训练的分析改错题,侧重编程实践能力训练的任务递进式编程题等。实验题从“巩固基础、综合设计、创新应用”三个层次进行设计,从第8章到第13章的实验题全部围绕一个综合应用实例来逐步展开和深入,以循序渐进的任务驱动方式,引导读者逐步编写出有一定规模的、贯穿全书的综合应用程序,即菜单驱动的学生成绩管理系统,该内容可作为课程设计的参考内容。

本书是国家精品课程主讲教材,在国家精品课程网站(<http://cms.hit.edu.cn/elite>)和教材网站(<http://book.sunner.cn>)上为读者提供了全方位、多元化的教学资源包括多媒体课件、全程教学录像、全部例题和习题的源代码。同时,我们还提供了如下教学支撑环境:包括C语言编程题考试自动评分系统(已获软件著作权专利),基于Moodle开发的具有作业在线评测和反抄袭功能的网络教学平台(<http://cms.hit.edu.cn>),以及针对本书习题的基于网络服务的uPlatform学习自测软件。其中,我们在国家自然科学基金项目资助下研制的C语言编程题考试自动评分系统可以免费提供给使用本教材一学期以上的教学单位。有需要上述资源者可直接与作者本人联系(sxh@hit.edu.cn)。

与本书配套出版的《C语言程序设计学习指导》主要包括习题解答和实验指导两部分内容。实验指导中介绍了在Visual C++6.0环境以及Code::Blocks+gcc+gdb环境下的标准C程序调试方法。多媒体教学课件、例题源代码、Code::Blocks+gcc+gdb(本书后面将其简称为Code::Blocks)集成开发环境的安装程序以及其他相关教辅资源的使用简介均附在本书的光盘中。

授课学时建议见前言后所附表格。

本书由苏小红主编,第1章由孙志岗执笔,第2章由王宇颖执笔,第3~13章由苏小红执笔,马培军和黄虎杰也参与了部分章节内容的编写。车万翔、赵巍、傅忠传、张卫、郭萍、李晗静、李希然、温东新、侯俊英、张洪志、王甜甜、陈惠鹏、秦兵、李秀坤、孙大烈、张彦航、单丽莉、刘旭东、刘国军、王庆北、刘劲锋、蔡江新、郝惠馨、李漾、娄久等参与了本书的内容校对工作。北京工业大学的蒋宗礼教授和国防科技大学的徐锡山教授在百忙之中仔细审阅了全部初稿,并提出了许多宝贵的意见和建议。在此对他们的辛勤付出表示衷心的感谢。

因编者水平有限,书中错误在所难免,恳请读者来信批评指正。作者E-mail地址为sxh@hit.edu.cn,wangyy@hit.edu.cn,sun@hit.edu.cn。高等教育出版社的网址为<http://computer.cncourse.com>。欢迎读者给作者或出版单位发送电子邮件或在网站上留言,对教材提出意见和建议。我们会在每次重印时及时予以更正,读者也可随时从我们的教材网站(<http://book.sunner.cn>)上下载教材的最新勘误表。

编著者

2011年于哈尔滨工业大学计算机科学与技术学院

“C 语言程序设计”授课内容和学时分配建议

章节内容	基本内容	30 学时	40 学时	50 学时
1. 为什么要学 C 语言	C 语言的发展历程	✓	✓	✓
	为什么要学 C 语言	✓	✓	✓
	什么是编程,编程的基础知识,软件开发的基本过程★	✓	✓	✓
	本章建议的学时	1 学时	1 学时	1 学时
2. C 数据类型	常量与变量★	✓	✓	✓
	简单的屏幕输出	✓	✓	✓
	基本数据类型★	✓	✓	✓
	计算变量或数据类型所占内存空间的大小★	✓	✓	✓
	变量和赋值和赋值运算符★	✓	✓	✓
	Code::Blocks 和 Visual C++ 6.0 集成开发环境简介	✓	✓	✓
	本章建议的学时	1 学时	1 学时	1 学时
3. 简单的算述运算和表达式	算术运算符和表达式★	✓	✓	✓
	复合的赋值运算符	✓	✓	✓
	增 1 和减 1 运算符★	✓	✓	✓
	宏常量与宏替换	✓	✓	✓
	const 常量★	✓	✓	✓
	自动类型转换与强制类型转换运算符★	✓	✓	✓
	常用的标准数学函数	⊗	✓	✓
	本章建议的学时	2 学时	2 学时	2 学时

续表

章节内容	基本内容	30 学时	40 学时	50 学时
4. 键盘输入和屏幕输出	字符常量★	✓	✓	✓
	字符的输入/输出	✓	✓	✓
	数据的格式化屏幕输出★	✓	✓	✓
	数据的格式化键盘输入★	✓	✓	✓
	本章建议的学时	2 学时	2 学时	2 学时
5. 选择控制结构	生活中的问题求解方法	✓	✓	✓
	计算机问题求解的基本特点	✓	✓	✓
	算法的概念和描述方法★	✓	✓	✓
	关系运算符与关系表达式★	✓	✓	✓
	逻辑运算符和逻辑表达式★	✓	✓	✓
	条件运算符和条件表达式★	✓	✓	✓
	用于选择控制的条件语句★	✓	✓	✓
	用于多路选择的 switch 语句★	✓	✓	✓
	程序测试★	✓	✓	✓
	对输入非法字符的检查与处理	⊗	⊗	⊗
	位运算	⊗	⊗	⊗
本章建议的学时	2 学时	4 学时	4 学时	
6. 循环控制结构	循环控制结构与循环语句★	✓	✓	✓
	计数控制的循环★	✓	✓	✓
	条件控制的循环★	✓	✓	✓
	嵌套循环★	✓	✓	✓
	流程的转移控制★ (goto, break, continue, exit())	✓	✓	✓
	结构化程序设计的核心思想★	✓	✓	✓
	常用的程序调试与排错方法	⊗	⊗	⊗
	类型溢出	⊗	⊗	✓
本章建议的学时	4 学时	4 学时	6 学时	

续表

章节内容	基本内容	30 学时	40 学时	50 学时
7. 函数	分而治之与信息隐藏	✓	✓	✓
	函数的定义★	✓	✓	✓
	函数调用,函数原型★	✓	✓	✓
	向函数传递值★	✓	✓	✓
	从函数返回值★	✓	✓	✓
	函数封装与防御性程序设计★	✓	✓	✓
	函数设计的基本原则	⊗	✓	✓
	递归函数★	✓	✓	✓
	变量的作用域和存储类型	✓	✓	✓
	自顶向下、逐步求精的模块化程序设计方法★	✓	✓	✓
	代码风格	⊗	✓	✓
	本章建议的学时	4 学时	6 学时	6 学时
8. 数组	一维数组的定义、初始化和引用★	✓	✓	✓
	二维数组的定义、初始化和引用★	✓	✓	✓
	向函数传递一维数组★	✓	✓	✓
	常用的排序和查找算法★	✓	✓	✓
	向函数传递二维数组★	✓	✓	✓
	本章建议的学时	4 学时	4 学时	6 学时
9. 指针	变量的内存地址	✓	✓	✓
	指针变量的定义和初始化★	✓	✓	✓
	间接寻址运算符★	✓	✓	✓
	按值调用与按地址调用★	✓	✓	✓
	用指针变量作函数参数★	✓	✓	✓
	函数指针及其应用★	⊗	✓	✓
	本章建议的学时	2 学时	4 学时	4 学时

续表

章节内容	基本内容	30 学时	40 学时	50 学时
10. 字符串	字符串常量★	✓	✓	✓
	字符指针★	✓	✓	✓
	字符串的存储、访问和输入/输出★	✓	✓	✓
	字符串处理函数★	✓	✓	✓
	向函数传递字符串★	✓	✓	✓
	从函数返回字符串指针	⊗	⊗	✓
	const 类型限定符	⊗	⊗	✓
	字符处理函数	⊗	⊗	⊗
	数值字符串向数值的转换	⊗	⊗	⊗
	本章建议的学时	2 学时	2 学时	4 学时
11. 指针和数组	指针和一维数组间的关系★	✓	✓	✓
	指针和二维数组间的关系★	⊗	✓	✓
	指针数组及其应用★	⊗	✓	✓
	动态内存分配函数	⊗	✓	✓
	长度可变的一维动态数组	⊗	⊗	✓
	长度可变的二维动态数组	⊗	⊗	✓
	常见的内存错误及其对策	⊗	⊗	✓
	缓冲区溢出攻击	⊗	⊗	⊗
	本章建议的学时	1 学时	4 学时	6 学时
12. 结构体和共用体	为什么要定义结构体类型	✓	✓	✓
	用 typedef 定义数据类型★	✓	✓	✓
	结构体变量的定义、初始化和成员的引用★	✓	✓	✓
	结构体数组的定义、初始化和成员的引用★	✓	✓	✓
	结构体指针的定义、初始化和成员的引用★	✓	✓	✓
	向函数传递结构体	✓	✓	✓
	共用体★	✓	✓	✓
	枚举数据类型	⊗	⊗	✓
	动态数据结构——单向链表	⊗	✓	✓
	本章建议的学时	3 学时	4 学时	4 学时

续表

章节内容	基本内容	30 学时	40 学时	50 学时
13. 文件操作	二进制文件和文本文件★	✓	✓	✓
	文件的打开和关闭★	✓	✓	✓
	按字符读写文件	✓	✓	✓
	按格式读写文件	✓	✓	✓
	按数据块读写文件	⊗	⊗	✓
	文件的随机读写	⊗	⊗	✓
	标准输入/输出重定向	⊗	⊗	⊗
	本章建议的学时	2 学时	2 学时	4 学时

注：✓表示课堂讲授内容，⊗表示选学、自学内容，★表示重点内容。

目 录

第 1 章 为什么要学 C 语言	1	3.2 宏常量与宏替换	31
1.1 引言	1	3.3 const 常量	34
1.2 游戏、黑客和 C 语言	1	3.4 自动类型转换与强制类型转换 运算符	35
1.3 C 语言, 不老的传说	4	3.5 常用的标准数学函数	38
1.4 C 语言的爱与恨	5	3.6 本章知识点小结	40
1.5 C 语言教给我们的事	6	3.7 本章常见错误小结	41
1.6 什么是“编程”	7	习题 3	42
1.7 本章小结	8	本章实验题	43
习题 1	8	第 4 章 键盘输入和屏幕输出	44
第 2 章 C 数据类型	9	4.1 单个字符的输入/输出	44
2.1 常量与变量	9	4.1.1 字符常量	44
2.1.1 常量	9	4.1.2 字符的输入/输出	45
2.1.2 变量	11	4.2 数据的格式化屏幕输出	47
2.2 简单的屏幕输出	13	4.3 数据的格式化键盘输入	50
2.3 数据类型	14	4.4 本章扩充内容	54
2.4 如何计算变量或数据类型所占 内存空间的大小	16	4.5 本章知识点小结	57
2.5 变量的赋值和赋值运算符	18	4.6 本章常见错误小结	58
2.6 本章扩充内容	19	习题 4	59
2.7 本章知识点小结	22	本章实验题	61
2.8 本章常见错误小结	23	第 5 章 选择控制结构	62
习题 2	23	5.1 生活中与计算机中的问题求解 方法	62
本章实验题	24	5.2 算法的概念及其描述方法	64
第 3 章 简单的算术运算和表达式	25	5.2.1 算法的概念	64
3.1 C 运算符和表达式	25	5.2.2 算法的描述方法	65
3.1.1 算术运算符和表达式	25	5.3 关系运算符与关系表达式	66
3.1.2 复合的赋值运算符	28	5.4 用于单分支控制的条件语句	68
3.1.3 增 1 和减 1 运算符	29		

5.5 用于双分支控制的条件语句	69	7.1 分而治之与信息隐藏	155
5.6 条件运算符和条件表达式	69	7.2 函数的定义	156
5.7 用于多分支控制的条件语句	70	7.2.1 函数的分类	156
5.8 用于多路选择的 switch 语句	75	7.2.2 函数的定义	157
5.9 逻辑运算符和逻辑表达式	80	7.3 向函数传递值和从函数返回值 ..	158
5.10 本章扩充内容	83	7.3.1 函数调用	158
5.10.1 程序测试	83	7.3.2 函数原型	160
5.10.2 对输入非法字符的检查 与处理	92	7.3.3 函数封装与防御性程序 设计	162
5.10.3 位运算符	95	7.3.4 函数设计的基本原则	171
5.11 本章知识点小结	98	7.4 递归函数	171
5.12 本章常见错误小结	99	7.4.1 递归问题的提出	171
习题 5	101	7.4.2 递归函数	174
本章实验题	102	7.5 变量的作用域和存储类型	179
第 6 章 循环控制结构	104	7.5.1 变量的作用域	179
6.1 循环控制结构与循环语句	104	7.5.2 变量的存储类型	181
6.2 计数控制的循环	114	7.6 模块化程序设计	189
6.3 条件控制的循环	116	7.6.1 模块分解的基本原则	189
6.4 嵌套循环	126	7.6.2 自顶向下、逐步求精	189
6.5 流程的转移控制	133	7.6.3 模块化程序设计实例	191
6.5.1 goto 语句	133	7.7 本章扩充内容	197
6.5.2 break 语句	133	7.8 本章知识点小结	200
6.5.3 continue 语句	135	7.9 本章常见错误小结	201
6.5.4 函数 exit()	137	习题 7	202
6.6 本章扩充内容	140	本章实验题	205
6.6.1 结构化程序设计的核心 思想	140	第 8 章 数组	206
6.6.2 常用的程序调试与排错 方法	141	8.1 一维数组的定义和初始化	206
6.6.3 类型溢出	142	8.2 二维数组的定义和初始化	211
6.7 本章知识点小结	144	8.3 向函数传递一维数组	215
6.8 本章常见错误小结	145	8.4 排序和查找	220
习题 6	146	8.5 向函数传递二维数组	234
本章实验题	152	8.6 本章知识点小结	238
第 7 章 函数	155	8.7 本章常见错误小结	239
		习题 8	240
		本章实验题	247

第9章 指针	248	11.3 指针数组及其应用	326
9.1 变量的内存地址	248	11.3.1 指针数组用于表示多个字 字符串	326
9.2 指针变量的定义和初始化	249	11.3.2 指针数组用于表示命令行 参数	333
9.3 间接寻址运算符	253	11.4 本章扩充内容	334
9.4 按值调用与按地址调用	255	11.4.1 C程序的内存映像	334
9.5 用指针变量作函数参数的程序 实例	261	11.4.2 动态内存分配函数	335
9.6 函数指针及其应用	264	11.4.3 长度可变的一维动态 数组	337
9.7 本章知识点小结	273	11.4.4 长度可变的二维动态 数组	339
9.8 本章常见错误小结	273	11.4.5 常见的内存错误及其 对策	341
习题9	274	11.4.6 缓冲区溢出攻击	351
本章实验题	276	11.5 本章知识点小结	353
第10章 字符串	278	11.6 本章常见错误小结	354
10.1 字符串常量	278	习题11	354
10.2 字符串的存储	278	本章实验题	358
10.3 字符指针	280	第12章 结构体和共用体	360
10.4 字符串的访问和输入/输出	281	12.1 从基本数据类型到抽象数据 类型	360
10.4.1 如何访问字符串中的单个 字符	281	12.2 结构体的定义	361
10.4.2 字符串的输入/输出	281	12.2.1 为什么要定义结构体 类型	361
10.5 字符串处理函数	286	12.2.2 结构体变量的定义	363
10.6 向函数传递字符串	290	12.2.3 用typedef定义数据类型	365
10.7 本章扩充内容	294	12.2.4 结构体变量的初始化	366
10.7.1 从函数返回字符串指针	294	12.2.5 嵌套的结构体	366
10.7.2 const 类型限定符	297	12.2.6 结构体变量的引用	368
10.7.3 字符处理函数	298	12.2.7 结构体所占内存的字 节数	372
10.7.4 数值字符串向数值的 转换	304	12.3 结构体数组的定义和初始化	373
10.8 本章知识点小结	306	12.3.1 结构体数组的定义	373
10.9 本章常见错误小结	306	12.3.2 结构体数组的初始化	374
习题10	308		
本章实验题	310		
第11章 指针和数组	312		
11.1 指针和一维数组间的关系	312		
11.2 指针和二维数组间的关系	319		

12.4 结构体指针的定义和初始化	376	13.4 按格式读写文件	413
12.4.1 指向结构体变量的指针	376	13.5 按数据块读写文件	419
12.4.2 指向结构体数组的指针	377	13.6 本章扩充内容	424
12.5 向函数传递结构体	377	13.6.1 文件的随机读写	424
12.6 共用体	384	13.6.2 标准输入/输出重定向	427
12.7 本章扩充内容	387	13.7 本章知识点小结	428
12.7.1 枚举数据类型	387	13.8 本章常见错误小结	429
12.7.2 动态数据结构——单向 链表	388	习题 13	429
12.8 本章知识点小结	398	本章实验题	429
12.9 本章常见错误小结	399	附录 A C 关键字	432
习题 12	400	附录 B Visual C++ 下基本数据类型的 取值范围	433
本章实验题	401	附录 C C 运算符的优先级与结合性 ..	434
第 13 章 文件操作	403	附录 D 常用字符与 ASCII 值对照表 ..	435
13.1 二进制文件和文本文件	403	参考文献	437
13.2 文件的打开和关闭	405		
13.3 按字符读写文件	406		

第 1 章 为什么要学 C 语言



内容导读

本章力图从一个客观的角度讲述学习 C 语言的意义,为不同需求的读者设计不同的学习目标。

1.1 引言

“为什么要学 C 语言?”这个问题每年在中华大地都会被问上几百万次。不同学校、不同专业、不同的人都可能会给出不同的回答。但是,很多学生,尤其是有独立思考精神的学生,往往会对得到的回答表示怀疑。这种怀疑直接导致的后果就是学习目标不明确,进而学习兴趣不足,再继续就是不爱学习了。为什么会这样?根源在于,学习 C 语言的真正理由并没有被揭示出来,因为很多人担心,如果说了真话,学生会更不爱学习了。但是,本书想冒一次险,说出真话。这么做,是因为我们相信,真诚永远好过虚假。

让我们先了解一下 C 语言的传奇身世,再解释为什么要学 C 语言的问题。

1.2 游戏、黑客和 C 语言

1969 年的美国贝尔实验室是当时科技界的梦工厂,集结着世界上最富创造力的科学家和工程师,其中包括数位诺贝尔奖获得者。他们一起创造了无数影响着全人类的发明,比如数码相机的核心——电荷耦合元件(Charge-coupled Device, CCD)就是那一年在贝尔实验室诞生的。

这些人的成就貌似高不可攀,但其实也都是凡人,在某些方面和我们也是非常相似的。比如当他们见到一台强大的计算机时,心里最先想到的也可能是用它来玩游戏。那个时代,计算机是大型机构才能拥有的奢侈品,在这上面玩游戏实在是暴殄天物。但谁在乎呢,追求快乐是最重要的,所以他们为了痛快地玩游戏而没少动脑筋。

那时候是没有商业游戏的,想有游戏玩,首先要发挥 DIY(Do It Yourself)精神,自己编。自己编的游戏被别人喜欢,是当时最有面子的事情。有一个叫 Ken Thompson(以下尊称为 ken)的工程师,26 岁,看到阿波罗 11 号载人飞船登月成功,觉得挺酷,就设计了一个叫“Space Travel”的游戏。在游戏中,玩家驾驶着宇宙飞船,在虚拟的太阳系里穿梭,欣赏美景的同时,还可以在各行星、卫星表面降落。这个游戏先是在 Mutlics 系统上编写,后来又在 GECOS 系统上重写。能运行这两个系统的机器都是笨重的大型机,虽然运算能力出众,但显示效果很差,而且机时费非常

高,玩一次需要支付 75 美元(当时美国人均月收入大约 200 美元)。这实在是太贵了。于是他同其同事 Dennis M. Ritchie(以下尊称为 dmr)一起寻找免费的“游戏机”。功夫不负有心人,还真被他们找到了一台。

这就是 PDP-7 小型机,由 DEC 制造,拥有当时最先进的图形处理能力。那时计算机的主要用途是数据处理,图形能力并不太重要,所以 PDP-7 更多的时候是静静地躺着,很少被使用,直到 ken 和 dmr 发现了它。

可以玩游戏了。然而,游戏的运行需要操作系统的支持。PDP-7 当时还是“裸机”,没有能在其上运行的操作系统。于是,他俩挽起袖子开始为 PDP-7 编写操作系统,并给这个系统起了一个名字——UNIX。直到今天,UNIX 仍然是最受信任的操作系统,它既支撑着军队、政府、电力、电信和银行等大型机构的关键业务,也是苹果 Mac 系列计算机的动力之源,甚至 iPhone、iPod Touch 的魅力也部分拜其所赐。图 1-1 展示了 ken 和 dmr 工作的情景。

UNIX 起初是用汇编语言编写的,那是一种更接近机器而不是人的语言。计算机能直接读懂的语言叫机器语言,它所有的语句都是由“0”和“1”两个数字构成的,根本就不是给人看的。当最后一个机器语言程序员“疯掉”以后,人们终于开始琢磨怎么让计算机读懂自然语言。基本思路是做一个翻译程序,直接把自然语言翻译成机器语言。这种翻译程序被命名为“编译器”。但是直接理解自然语言太难了,直到现在还没能完全实现,所以就折中一下,设计一种尽量接近自然语言,且还能被精确翻译为机器语言的语言,这种语言就是人们常说的编程语言。学编程的过程,其实就是学习怎样用编程语言说话让编译器听懂的过程。第一种编程语言肯定是最接近机器而远离人类的,它就是汇编语言。虽然看上去有几分像自然语言,比如加法叫“ADD”,减法叫“SUB”,但它的语法完全是机器的,每一行语句都和一条机器指令严格对应。不同计算机的机器指令是不一样的,所以针对一种计算机编写的汇编程序不能在另一种计算机上直接使用,必须重写(Space Travel 就被重写过很多次)。用专业术语来说,汇编语言缺少“可移植性”。

UNIX 的优雅加上 Space Travel 的吸引力,使很多人希望他们的计算机上也能安装 UNIX。于是 ken 和 dmr 决定改用高级语言编写 UNIX,这样它就可以在更多类型的计算机上运行。

除了机器语言和汇编语言以外,几乎所有编程语言都被统称为高级语言。它的特点是更接近自然语言,而与机器语言基本没有瓜葛。不同的高级语言编译器可以把同样的高级语言程序翻译成适应不同机器的指令,因而高级语言大多具有较好的可移植性。



站立者为 dmr,端坐者为 ken,
体积最大者为 PDP-11

图 1-1 ken 和 dmr 在工作或娱乐中