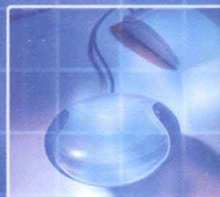




高等学校**应用型特色**规划教材

C语言程序设计

C YUYAN CHENGXU SHEJI



陈锐 白政民 主编
刘宇 刘河 王佩 副主编

赠送
电子教案

- 新颖的教材内容——基础理论知识与实例完美融合
- 丰富的教学资源——免费提供电子课件、习题库及其参考答案



清华大学出版社

高等学校应用型特色规划教材

C 语言程序设计

陈 锐 白政民 主 编

刘 宇 刘 河 王 佩 副主编

清华大学出版社

北 京

内 容 简 介

C 语言是计算机专业的基础课和核心课程。本书内容包括 C 语言概述、C 语言程序开发环境、基本数据类型、运算符与表达式、顺序结构程序设计、选择结构程序设计、循环结构程序设计、数组、函数、指针、预处理命令、结构体与联合体、位运算、文件、链表。本书内容全面,结构清晰,语言通俗,重难点突出,例题丰富,所有程序都能够直接运行。

本书可作为高等院校的计算机及相关专业的 C 语言教材,也可作为计算机软件开发、等级考试和软件资格考试人员的参考书。

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。
版权所有,侵权必究。侵权举报电话:010-62782989 13701121933

图书在版编目(CIP)数据

C 语言程序设计/陈锐,白政民主编;刘宇,刘河,王佩副主编.——北京:清华大学出版社,2011.7
(高等学校应用型特色规划教材)
ISBN 978-7-302-25915-2

I. C… II. ①陈… ②白… ③刘… ④刘… ⑤王… 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 108890 号

责任编辑:黄 飞

装帧设计:杨玉兰

责任校对:李玉萍

责任印制:李红英

出版发行:清华大学出版社

地 址:北京清华大学学研大厦 A 座

<http://www.tup.com.cn>

邮 编:100084

社 总 机:010-62770175

邮 购:010-62786544

投稿与读者服务:010-62776969, c-service@tup.tsinghua.edu.cn

质 量 反 馈:010-62772015, zhiliang@tup.tsinghua.edu.cn

印 刷 者:北京市人民文学印刷厂

装 订 者:三河市新茂装订有限公司

经 销:全国新华书店

开 本:185×260 印 张:22.25 字 数:536 千字

版 次:2011 年 7 月第 1 版 印 次:2011 年 7 月第 1 次印刷

印 数:1~4000

定 价:39.00 元

产品编号:041266-01

前 言

C 语言是自诞生之日起在国内外一直被广泛使用的一种计算机高级程序设计语言。C 语言功能丰富，表达式能力强，使用灵活方便，应用面广，运行效率高，可移植性好，既具有高级语言的优点，又具有低级语言的许多特点。C 语言既可以编写系统软件，也可以编写应用软件。正是如此众多的优点，C 语言被作为计算机专业必修的一门课程。

许多高等院校的计算机专业不仅开设了 C 语言课程，而且非计算机专业也开设了 C 语言课程。全国计算机等级考试、软件资格(水平)考试都将 C 语言列入考试范围。在数据结构与算法课程、研究生入学考试中，C 语言也成为常用的描述语言。学习 C 语言已经成为广大计算机爱好者和青年学生的迫切要求。

本书作者多年来一直从事 C 语言程序设计的教学与研究工作，并从事数据结构与算法、自然语言理解的研究，具有较为丰富的科研实践经验与程序开发能力，对程序设计有深刻的研究，书中很多地方都是作者总结的教学经验。

本书的特点

1. 内容全面，讲解详细

为了方便学生学习，本书首先介绍了 C 语言的特点及优势，然后介绍 C 语言常用开发工具。本书内容全面，覆盖全部 C 语言的知识，对于每个知识点，都使用具体实例进行讲解，以便学生迅速掌握 C 语言。

2. 结构清晰，内容合理

本书将 C 语言按章、节和小节划分知识点，将知识点细化，结构框架清晰，内容讲解合理，易于学生理解与学习。在知识点的讲解过程中，循序渐进，由浅入深，最后通过实例强化知识点，这样的讲解方式使学生更容易理解和消化。

3. 语言通俗，叙述简单

针对每个知识点，使用比较通俗的语言进行讲解，避免使用晦涩难懂的语句，以便学生理解。这样，学生可以更加容易学习并掌握 C 语言。

4. 实例典型，深入剖析

在讲解每一个 C 语言知识点时，都结合具体实例进行剖析。在实例的选取上，都是一些最为常见且每个实例涵盖知识点丰富的典型程序。在每一章的最后或比较大的知识点后面，都给出了一个完整的程序，并对程序通过图进行具体讲解，深入分析。

5. 配有小结和习题，巩固知识点

在每一章的最后，都有一个小结，对本章的知识点进行总结，方便读者清楚本章的重点、难点。为了让读者熟练编写 C 语言程序，每一章的最后都配有一定数量的习题，在学习了每一章的内容之后，可以通过这些习题试着编写 C 语言程序，以便巩固本章的学习内容。



本书的内容

第1章：介绍C语言的发展历史与优势，学习C语言的一些基础知识。

第2章：介绍两种最为常用的C语言开发工具：Turbo C 2.0 和 Visual C++ 6.0。

第3章：介绍基本数据类型：整型、实型、字符型以及相应的常量与变量。

第4章：介绍运算符与表达式。

第5章：先介绍语句的概念，然后介绍输入/输出函数的使用，最后通过具体实例讲解如何编写顺序结构的程序。

第6章：主要介绍单分支的if选择语句、双分支的if选择语句、多分支的if选择语句、switch选择语句和条件运算符与条件表达式。

第7章：主要介绍几种常用的循环语句——顺序结构的C语言程序、while循环语句、do-while循环语句、for循环语句、goto语句、break语句和continue语句、多重循环结构的程序设计。

第8章：主要讲解了一维数组、二维数组、字符数组。

第9章：主要讲解了函数的分类、函数的定义形式、局部变量和全局变量、函数的参数和函数的返回值、函数的调用、函数的嵌套调用、函数的递归调用、数组作为函数的参数。

第10章：首先讲解指针与地址的区别，然后讲解了指针变量、数组与指针、字符串与指针、指针数组与二级指针、函数与指针、返回指针值的函数。

第11章：介绍宏定义、文件包含命令、条件编译命令。

第12章：首先讲解结构体的定义、引用、初始化，然后讲解了结构体数组、指针与结构体、联合体、枚举类型。

第13章：主要介绍位运算，了解计算机的底层操作。

第14章：首先讲解文件的分类，然后讲解了打开和关闭文件、读取文件和写入文件、文件的定位等操作，最后讲解常见的其他文件操作函数。

第15章：首先讲解链表的概念、动态存储分配，然后讲解链表的创建、链表的查找、链表的插入、链表的删除等操作。

由于编者水平有限，书中难免有疏漏和不妥之处，敬请广大读者批评指正。

本书由陈锐、白政民(许昌学院)任主编，刘宇(空军航空大学)、刘河(华中师范大学汉口分校)、王佩任副主编，王鹤(辽宁工程技术大学)、汪东方、王三中、夏敏捷、宋海声(西北师范大学)、李红、张思卿任编委。

编 者

目 录

第 1 章 C 语言概述.....	1	2.2.3 使用 Visual C++ 6.0 开发环境 运行 C 语言程序	28
1.1 C 语言的产生与发展	1	2.3 小结	29
1.2 为什么要学习 C 语言	2	第 3 章 基本数据类型	30
1.2.1 学习 C 语言的好处	2	3.1 数据类型的分类	30
1.2.2 如何学好 C 语言	3	3.2 常量与变量	32
1.3 C 语言的特点	5	3.2.1 常量	32
1.4 程序设计基础——进制转换	6	3.2.2 变量	33
1.4.1 二进制数的表示	6	3.3 整型数据	34
1.4.2 二进制数与十进制数、 八进制数、十六进制数 的对应关系	7	3.3.1 整型常量	35
1.4.3 二进制数、十六进制数和 八进制数转换为十进制数	7	3.3.2 整型变量	36
1.4.4 十进制数转换为二进制数	8	3.4 实型数据	40
1.4.5 十进制数转换为十六进制数	10	3.4.1 实型常量	40
1.4.6 十进制数转换为八进制数	12	3.4.2 实型变量	41
1.4.7 二进制数与十六进制数、 八进制数的相互转换	13	3.5 字符型数据	43
1.5 计算机中数的表示	14	3.5.1 字符型常量	43
1.5.1 计算机中的正数与负数表示	14	3.5.2 字符型变量	44
1.5.2 原码、补码	15	3.5.3 字符串常量	46
1.5.3 浮点数	17	3.6 小结	47
1.6 小结	19	习题	47
习题	20	第 4 章 运算符与表达式	49
第 2 章 C 语言程序开发环境	21	4.1 变量赋初值	49
2.1 Turbo C 2.0 开发环境	21	4.2 自动类型转换与强制类型转换	50
2.1.1 Turbo C 2.0 开发环境介绍	21	4.3 算术运算符与算术表达式	52
2.1.2 C 程序的开发步骤	24	4.3.1 双目运算符	52
2.2 Visual C++ 6.0 开发环境	24	4.3.2 算术表达式	52
2.2.1 使用 Visual C++ 6.0 开发环境 新建 C 程序文件	25	4.3.3 单目运算符	53
2.2.2 Visual C++ 6.0 开发环境常用 命令介绍	27	4.4 赋值运算符与赋值表达式	55
		4.4.1 赋值运算符与赋值表达式 简介	55
		4.4.2 类型转换	57
		4.5 关系运算符与关系表达式	60
		4.5.1 关系运算符	60



4.5.2 关系表达式	61	第7章 循环结构程序设计	110
4.6 逻辑运算符与逻辑表达式	62	7.1 goto 语句	110
4.6.1 逻辑运算符	62	7.2 while 语句	111
4.6.2 逻辑表达式	63	7.3 do...while 语句	113
4.7 逗号表达式	64	7.4 for 语句	117
4.7.1 逗号运算符与逗号表达式	64	7.5 循环的嵌套	119
4.7.2 应注意的问题	65	7.6 break 语句和 continue 语句	121
4.7.3 逗号表达式应用举例	65	7.6.1 break 语句	121
4.8 小结	66	7.6.2 continue 语句	122
习题	66	7.7 循环结构的应用举例	123
第5章 顺序结构程序设计	69	7.8 小结	125
5.1 语句	69	习题	125
5.1.1 C 语句	69	第8章 数组	131
5.1.2 C 语句分类——简单语句和 复合语句	69	8.1 数组的概念	131
5.2 基本输入/输出在 C 语言中的实现	71	8.1.1 数组的引入	131
5.3 字符数据的输入/输出	72	8.1.2 数组的维数	132
5.3.1 putchar 函数——输出字符	72	8.2 一维数组	134
5.3.2 getchar 函数——输入字符	72	8.2.1 一维数组的定义和初始化	134
5.4 格式输入/输出	73	8.2.2 计算数组的大小——sizeof 运算符	138
5.4.1 printf 函数——格式化数据 的输出	73	8.2.3 一维数组的应用举例	138
5.4.2 scanf 函数——格式化数据 的输入	80	8.3 二维数组	146
5.5 顺序结构程序设计举例	83	8.3.1 二维数组的定义	146
5.6 小结	85	8.3.2 二维数组初始化	148
习题	85	8.3.3 二维数组应用举例	150
第6章 选择结构程序设计	89	8.4 字符数组	153
6.1 if 语句	89	8.4.1 字符数组和字符串	153
6.1.1 if 语句的三种形式	89	8.4.2 字符串处理函数	159
6.1.2 if 语句的嵌套	94	8.4.3 二维字符数组	162
6.1.3 条件运算符	97	8.4.4 字符数组应用举例	165
6.2 switch 语句	99	8.5 数组的应用举例	167
6.3 选择结构的应用举例	101	8.6 小结	172
6.4 小结	105	习题	173
习题	105	第9章 函数	176
		9.1 函数的定义	176
		9.2 函数的参数与函数返回值	179
		9.2.1 形式参数与实际参数	180

9.2.2 函数的返回值.....	180	10.4 字符串与指针.....	240
9.3 函数的调用	182	10.4.1 字符串指针.....	240
9.3.1 函数的一般调用.....	182	10.4.2 字符串指针作为函数的 参数.....	241
9.3.2 函数的嵌套调用.....	185	10.4.3 字符数组与字符指针变量的 比较.....	242
9.4 函数的递归调用.....	188	10.5 指针数组与二级指针	243
9.4.1 递归调用的定义.....	188	10.5.1 指针数组.....	244
9.4.2 递归调用应用举例.....	189	10.5.2 二级指针.....	244
9.5 数组作为函数参数.....	193	10.5.3 指针数组作为 main 函数的 参数.....	246
9.5.1 数组元素作为函数参数.....	193	10.6 函数与指针.....	247
9.5.2 数组名作为函数参数.....	194	10.6.1 函数指针——指向函数的 指针.....	247
9.6 变量的作用域	198	10.6.2 函数指针作为函数参数	249
9.6.1 局部变量	198	10.6.3 指针函数——返回指针值 的函数.....	251
9.6.2 全局变量	199	10.6.4 void 指针.....	252
9.7 变量的存储类别.....	202	10.7 小结.....	253
9.7.1 自动变量——auto 变量.....	202	习题	254
9.7.2 静态变量——static 变量	203		
9.7.3 寄存器变量——register 变量	204	第 11 章 预处理命令	255
9.7.4 外部变量——extern 变量.....	204	11.1 宏定义.....	255
9.8 内部函数与外部函数.....	206	11.1.1 不带参数的宏定义	255
9.8.1 内部函数	207	11.1.2 带参数的宏定义.....	257
9.8.2 外部函数	207	11.2 文件包含命令.....	259
9.9 函数的应用举例.....	208	11.2.1 文件包含命令—— #include.....	259
9.10 小结	216	11.2.2 多文件结构.....	260
习题	217	11.3 条件编译命令.....	262
第 10 章 指针.....	220	11.3.1 第一种条件编译命令—— #ifdef.....	262
10.1 地址与指针的概念.....	220	11.3.2 第二种条件编译命令—— #ifndef.....	263
10.1.1 地址	220	11.3.3 第三种条件编译命令—— #if.....	263
10.1.2 指针	221	11.3.4 第四种条件编译命令—— #if...#elif...#endif.....	264
10.2 指针变量	222	11.4 小结.....	265
10.2.1 指针变量的定义.....	222		
10.2.2 指针变量的引用.....	223		
10.2.3 指针变量作为函数参数.....	227		
10.3 指针与数组	229		
10.3.1 指向数组元素的指针与指向 数组的指针.....	229		
10.3.2 数组指针作为函数的参数.....	231		
10.3.3 指向多维数组的指针变量.....	234		



习题	265	13.1.7 与位运算符相结合的赋值 运算.....	298
第 12 章 结构体与联合体	267	13.2 位运算应用举例.....	298
12.1 结构体	267	13.3 位段.....	302
12.1.1 结构体类型的定义.....	267	13.4 小结.....	304
12.1.2 结构体变量的定义.....	268	习题	305
12.1.3 结构体变量成员的引用.....	270	第 14 章 文件	306
12.1.4 结构体变量的初始化.....	271	14.1 C 文件与文件类型指针	306
12.2 结构体数组	273	14.1.1 C 文件的分类	306
12.2.1 结构体数组的定义及 初始化	273	14.1.2 文件类型指针	307
12.2.2 结构体数组的初始化.....	274	14.2 文件的打开与关闭.....	307
12.2.3 结构体数组应用举例.....	275	14.2.1 文件的打开.....	307
12.3 指针与结构体.....	277	14.2.2 文件的关闭.....	309
12.3.1 指向结构体变量的指针.....	277	14.3 文件的读写.....	310
12.3.2 指向结构体数组的指针.....	279	14.3.1 fputc 函数与 fgetc 函数.....	311
12.3.3 结构体变量作为函数 参数	280	14.3.2 fgets 函数和 fputs 函数	316
12.4 typedef 类型定义.....	281	14.3.3 fread 函数与 fwrite 函数	317
12.5 联合体	283	14.3.4 fscanf 函数与 fprintf 函数	320
12.5.1 联合体的定义及引用.....	283	14.4 文件的定位.....	324
12.5.2 使用联合体应注意的问题.....	284	14.5 小结.....	326
12.5.3 联合体的应用举例.....	285	习题	327
12.6 枚举类型	287	第 15 章 链表	329
12.6.1 枚举的定义及引用.....	287	15.1 链表的相关概念.....	329
12.6.2 枚举的应用举例.....	288	15.1.1 链表.....	329
12.7 小结	290	15.1.2 动态存储分配.....	330
习题	290	15.2 链表的操作.....	331
第 13 章 位运算	292	15.2.1 链表的创建.....	331
13.1 位运算符与位运算.....	292	15.2.2 链表的输出.....	333
13.1.1 位与运算	292	15.2.3 链表的查找.....	334
13.1.2 位或运算	293	15.2.4 链表的插入操作.....	334
13.1.3 异或运算	294	15.2.5 链表的删除操作.....	336
13.1.4 取反运算	296	15.3 链表的应用举例.....	338
13.1.5 左移运算	296	15.4 小结.....	341
13.1.6 右移运算	297	习题	341

第 1 章 C 语言概述

C 语言是经典的程序设计语言之一，它是 1972 年由 Dennis M. Ritchie 设计发明的，并首次在基于 UNIX 操作系统的 DEC PDP-11 计算机上使用。随后 C 语言在各个领域得到了广泛应用，至今，C 语言仍然是应用最为广泛的语言之一。

C 语言的诞生是现代程序语言革命的起点，是程序设计语言发展史中的一个里程碑。自 C 语言出现后，以 C 语言为基础的 C++、Java 和 C# 等面向对象语言相继诞生，并在各领域的使用获得巨大成功。即使今天，C 语言在系统编程、嵌入式编程等领域依旧占据着统治地位。

1.1 C 语言的产生与发展

C 语言是国际上广泛流行的、很有发展前途的计算机高级语言。它适合于作为系统描述语言，既可用于来写系统软件，也可用来写应用软件。

早期的系统软件主要使用汇编语言编写，由于汇编语言依赖计算机硬件，程序的可读性和可移植性比较差，因此为了提高系统软件的可读性和可移植性，而改用了高级语言编写系统软件。但是，一般的高级程序设计语言不具备汇编语言的功能(汇编语言可以直接对硬件进行操作，例如，对内存地址操作、位一级的操作)而 C 语言兼具了一般高级语言和低级语言的优点，因此得到了广泛的应用，并且一直以来经久不衰。

C 语言是在 B 语言的基础上发展起来的，它的根源可以追溯到 ALGOL 60。1960 年，图灵奖获得者 Alan J. Perlis 在前人的基础上发明了 ALGOL 60 语言。ALGOL 60 是程序设计语言发展史上的一个里程碑，它标志着程序设计语言成为一门独立的科学学科，并为后来软件自动化及软件可靠性的发展奠定了基础。

1963 年，剑桥大学将 ALGOL 60 语言发展成为 CPL(Combined Programming Language)语言，CPL 语言更接近硬件，但规模比较大，难以实现。1967 年，剑桥大学的 Martin Richards 对 CPL 语言进行了简化，推出了 BCPL(Basic Combined Programming Language)语言。1970 年，Ken Thompson 又将 BCPL 进行了修改，并为它起了一个有趣的名字——B 语言。

由于 B 语言过于简单，功能有限，1972—1973 年，贝尔实验室的 Dennis M. Ritchie 在 B 语言的基础上设计出了 C 语言。C 语言保持了 BCPL 语言和 B 语言的优点，既简练又接近硬件，同时，克服了它们的缺点——过于简单，数据无类型。最初，C 语言只是为了描述和实现 UNIX 操作系统，在 1973 年，Ken Thompson 和 Dennis M. Ritchie 合作，将 UNIX 的 90% 以上用 C 语言改写。刚开始，C 语言主要在贝尔实验室使用，直到 1975 年，UNIX 的第 6 版发布后，C 语言的突出优点才引起人们的注意。

为了使 UNIX 操作系统推广，1977 年 Dennis M. Ritchie 发表了不依赖于具体机器系统的 C 语言编译文本《可移植的 C 语言编译程序》。1978 年，Brian W. Kernighan 和 Dennis M. Ritchie 出版了名著《The C Programming Language》，从而使 C 语言成为目前世界上流行



最广泛的高级程序设计语言之一。

随着微型计算机的日益普及,出现了许多 C 语言版本。1983 年,美国国家标准化协会(ANSI)对 C 语言进行了扩充,并制定了新的标准,被称为 ANSI C。1988 年, Brian W. Kernighan 和 Dennis M. Ritchie 根据 ANSI C 标准重新修订了他们的经典著作《The C Programming Language》。1987 年, ANSI 又公布了新标准——87 ANSI C。

1990 年,国际标准化组织 ISO(International Organization for Standards)接受了 87 ANSI C 为 ISO C 的标准(ISO 9899—1990)。1994 年, ISO 又对 C 语言标准进行了修订。

目前流行的 C 语言编译系统大多是以 ANSI C 为基础进行开发的,不同版本的 C 编译系统所实现的语言功能和语法规则略有差别。

1.2 为什么要学习 C 语言

C 语言经历了 30 多年的发展,仍然能够经久不衰,最重要的原因是它写出的程序能高效运行,功能强大,能够进行系统级的开发。因此, C 语言与其他语言相比,具有独特的先天优势。在学习 C 语言之前,首先让我们先了解一下学习 C 语言的好处,以及如何学好 C 语言。

1.2.1 学习 C 语言的好处

对于计算机专业及相关专业的学生来说, C 语言是必须要掌握的语言。可以说 C 语言是每个计算机专业的学生入门的首选语言,只有学好了 C 语言,才能掌握好计算机专业的后续课程。C 语言是今后学习各种语言的基础,也起到了理解计算机的桥梁和纽带作用。

1. C 语言是各种程序设计语言的基础

目前最为流行的程序设计语言有 C 语言、C++语言、Java 语言、C#等,而 C 语言是这些语言的基础。C 语言相对于 C++、Java、C#等语言来说,语法结构简单,没有涉及太多的技术细节与开发环境的使用,学习起来比较容易掌握。在学了 C 语言之后,再学习 C++、Java、C#会更加容易。

C 语言与其他语言相比,具有以下显著特点。

(1) 如果有了 C 语言作为基础,掌握其他编程语言如 Java、C++、C#会更加容易,这是因为它们都是以 C 语言作为基础的语言。

(2) 与其他编程语言(如 Java、C#)相比较, C 语言是一门低级语言,而学习低级语言能够使你更加清楚地了解计算机的工作原理。与汇编语言相比, C 语言又是一门高级语言, C 语言更加容易掌握。

(3) C 语言是一门比较低级的语言,更接近硬件,因此运行速度非常快。在实时性要求比较高的情况下, C 语言是最佳选择。

(4) C 语言允许直接访问物理地址,可以进行位一级的运算,能实现汇编语言的大部分功能,因此 C 语言又是一门介于高级语言和汇编语言之间的语言。

(5) 学习了 C 语言中的指针,我们会更加清楚地了解到计算机操作的本质。而 C#和 Java 则没有这个概念,因此难以领略到指针的强大和语言的本质。

(6) 相对于其他编程语言(如 C++、Java)C 是更底层的语言。在一个较底层水平编程,能够在总体上让我们进一步加深对计算机的理解。

(7) 与汇编语言先比, C 语言的可移植性非常好。

2. C 语言是一种主流的语言

目前最为流行的开发工具主要有 Visual C++、Visual Basic.net、Eclipse、Visual C#等,从技术角度来看,可以分为.NET 阵营和 Java 阵营,这两个阵营都分别有强大的公司支持。不管是今后我们选择.NET 还是 Java, C 语言都是这两种技术的基础。学好 C 语言对于今后继续掌握.NET 和 Java 都是大有裨益的。并且,掌握好 C 语言还为我们今后选择系统开发打下良好的基础。因此, C 语言成为主流的程序设计语言是必然的。它的主流地位主要体现在以下几个方面。

(1) C 语言从诞生到现在已经有将近 40 年的历史,因此给我们留下了相当多的 C 语言现成可用的代码,这些代码都是比较宝贵的资源,另外有相当多的团体仍然在支持并使用 C 语言。

(2) 设备驱动程序和操作系统只能用 C 语言编写。只有精通了 C 语言,才能编写和改进它们。

(3) 目前的计算机等级考试、软件水平考试都把 C 语言作为一门必考的科目,在大多数企业招聘考试中, C 语言也是几乎唯一的考试内容。

(4) 程序设计语言是计算机专业学生今后要学习的数据结构和算法等课程的基础。目前,数据结构和算法的教材基本都采用 C 语言进行描述。

(5) C 语言是开源社区的主流语言。很多开源程序以及 Linux 都是用 C 语言编写的。如果我们掌握了 C 语言,就能通过阅读源代码深入理解计算机系统了。

1.2.2 如何学好 C 语言

想要学好 C 语言,必须把握好以下几点。

1. 确立离散的思维方式,摒弃连续性思维方式

在学习任何一门编程语言时,一定要确立离散性的思维方式。这决定着你是否真正掌握好编程语言的重要因素,学习 C 语言也是如此。刚开始学习 C 语言时,不少学生很难摆脱旧有的惯性思维方式的影响。在我们学习 C 语言时,不要考虑太多的数学证明、公式推导,在计算机内部,数据是以二进制形式存储,是一种离散的数据。在利用 C 语言处理类似连续性函数、积分等问题时,需要将这些问题的离散化处理。在今后学习 C 语言的过程中,你会深刻地体会到这一点。

2. 熟练掌握二进制与十进制、十六进制、八进制之间的相互转换

在计算机内部,所有的数据都是以二进制形式存储的。为了我们习惯考虑,在显示时需要将二进制转换为十进制数,而在存储时,又需要将十进制数转换为二进制数。但是,同样一个数,使用二进制数表示起来又太长,为了方便,常需要将二进制用十进制、十六进制、八进制表示。因此,在学习 C 语言时,需要我们熟练掌握二进制、十进制、十六进



制、八进制之间的相互转换。

3. 理解字符与 ASCII 码之间的关系

通过键盘输入的数据是字符数据，而计算机是以二进制形式存储。这需要将字符转换为对应的二进制形式并存放起来。美国的标准协会 ANSI 专门制定了字符与 ASCII 之间的对应关系。字符与 ASCII 码之间的对应关系可参见附录。

4. 掌握运算符的结合性与优先级

C 语言提供了 34 种运算符，每种运算符都有优先级与结合性。如果有多个运算符出现在同一个表达式中，需要选择优先级别高的运算符进行计算。如果运算符相同，则需要根据运算符的结合性进行运算。

5. 掌握 3 种程序控制结构

C 语言是一种结构化的程序设计语言，它具有 3 种控制结构：顺序结构、选择结构和循环结构。这 3 种程序控制结构是目前结构化程序设计语言都具有的。编写一个完整的 C 语言程序几乎都要使用这 3 种结构，使用这 3 种结构可以解决几乎所有问题。

6. 掌握一些常用的算法

在学习 C 语言的过程中，常常需要对一些数据进行排序及查找给定的数据，这就是排序算法和查找算法。排序算法和查找算法是在程序设计过程中常用的算法，排序算法可以分为冒泡排序、插入排序、选择排序等，查找算法可以分为顺序查找、折半查找等。掌握一些常用的算法对今后学习数据结构和算法是大有裨益的。

7. 熟练使用指针

指针是 C 语言区别于其他语言的一个重要标志。指针是 C 语言的灵魂，熟练使用指针可以使程序编写更加灵活，编写出来的程序运行效率更加高效。指针是一把双刃剑，使用得好；可以提高运行效率；使用得不当，则会造成难以意料的错误。因此，这就需要大家在学习的过程中，熟练掌握指针。

8. 熟练掌握一个开发工具

工欲善其事，必先利其器。学好一门语言，需要我们熟练地掌握一个开发工具。只有多上机练习，才能知道程序是否正确。C 语言的开发工具有许多，目前比较流行的有 Turbo C 2.0、Turbo C 3.0、Visual C++ 6.0、Win-TC、LCC-Win32 等。我们建议初学者可以学习 Turbo C 2.0 或 Turbo C 3.0，有了基础之后再选择 Visual C++ 6.0，Visual C++ 6.0 是一个非常专业的开发工具。

9. 多上机、多练习

C 语言是一门实践性非常强的课程，除了多看书外，还需要我们多上机、多练习，只有这样，我们才能熟练掌握 C 语言的开发工具，才能熟练调试 C 语言，遇到错误才可以很快地解决。

1.3 C语言的特点

C语言之所以具有如此强大的生命力，并成为最受人们欢迎的语言，除了它优于其他编程语言的先天优势之外，C语言本身具有的一些内在的特点也是很重要的原因。许多著名的系统软件，如PC-DOS、Linux都是由C语言编写的。C语言的特点主要有以下几个。

1. C语言是结构化语言

结构化程序层次清晰，易读性强，容易维护，便于调试。在传统的程序流程中，可以使用比较随意的转移语句使程序的控制流程转向任意一处。如果一个程序中多处出现转移情况，将会导致程序结构杂乱无章，容易出错，这样的程序易读性差，尤其是在实际软件产品的开发中，更多地追求软件的可读性和可修改性，像这种结构和风格的程序是不允许出现的。

结构化程序设计方法主要由以下三种逻辑结构组成。

- (1) 顺序结构：顺序结构是一种线性、有序的结构，它依次执行各语句模块。
- (2) 选择结构：选择结构是根据条件成立与否选择程序执行的通路。
- (3) 循环结构：循环结构是重复执行一个或几个模块，直到满足某一条件为止。

对于结构化程序来说，由于每个模块执行单一功能，模块间联系较少，使程序编制比过去更简单，程序更可靠，而且增加了可维护性，每个模块可以独立编制和测试。

2. C语言是中级语言

C语言把高级语言的基本结构和语句与低级语言的实用性结合起来。C语言可以像汇编语言一样对位、字节和地址进行操作，而这三者是计算机中最基本的工作单元。

- C语言语法限制不太严格，程序设计自由度大。虽然C语言也是强类型语言，但书写形式自由，实现同一个功能的语句可以有多种写法，给程序编写者有较大的自由度。例如，在for循环语句中，可以将赋初值放在for语句的前面，也可以放在for语句内部。循环的判断条件还可以省略，循环变量的增值可以在循环条件中，也可以在循环语句的内部。
- C语言允许直接访问物理地址，可对硬件进行操作。由于C语言允许直接访问物理地址，可以直接对硬件进行操作，所以它能够像汇编语言一样对物理地址进行操作。C语言既具有高级语言的功能，又具有低级语言的许多功能，可用来编写系统软件。
- 生成目标代码质量高，程序执行效率高。一般只比汇编程序生成的目标代码效率低10%~20%。

3. C语言功能齐全

C语言具有丰富的运算符和数据类型，便于实现各类复杂的数据结构，并引入了指针概念，可使程序效率更高。另外，C语言也具有强大的图形功能，支持多种显示器和驱动器。而且计算功能、逻辑判断功能也比较强大，可以用来实现各种程序。



4. 运算符丰富

C 语言的运算符包含的范围非常广泛, 共有 34 种运算符。C 语言把括号、赋值、强制类型转换等都作为运算符处理。从而使 C 语言的运算类型极其丰富, 表达式类型多样化。灵活使用各种运算符可以实现其他高级语言中难以实现的运算。

5. 数据结构丰富

C 语言的数据类型有: 整型、实型、字符型、数组类型、指针类型、结构体类型、共用体类型等。这些数据类型可以用来实现基于各种复杂的数据结构的运算。

6. C 语言适用范围大、可移植性好

C 语言有一个突出的优点就是适用于多种操作系统, 如 Windows、UNIX; 也适用于多种机型。C 语言具有强大的绘图能力, 可移植性好, 并具备很强的数据处理能力, 因此适于编写系统软件以及二维、三维图形和动画软件。此外, C 语言也是数值计算常用的高级语言。

1.4 程序设计基础——进制转换

计算机内部的信息可分为两大类: 控制信息和数据信息。控制信息是一系列的控制命令, 用于指挥计算机如何操作; 数据信息是计算机操作的对象, 一般又可分为数值数据和非数值数据。数值数据用于表示数量的大小, 有确定的数值; 非数值数据没有确定的数值, 主要包括字符、汉字、逻辑数据等。但是不论是什么信息, 在输入计算机内部时, 都必须用二进制编码表示, 以方便存储、传送和处理。

1.4.1 二进制数的表示

我们最熟悉和最常用的是十进制数, 十进制数的特点是: 逢十进一, 借一当十。十进制数需要用到的数字符号为 10 个, 分别是 0~9。实际上, 在日常生活中, 使用各种进位计数制。进位计数制是一种计数的方法。例如, 六十进制(1 小时=60 分、1 分=60 秒), 十二进制(1 英尺=12 英寸、1 年=12 月), 七进制(1 周=7 天), 二进制(1 双袜子=2 只袜子)等。

但是对计算机而言, 不论是控制命令还是数据信息, 它们都要用 0 和 1 两个基本符号(二进制码)来表示, 原因如下。

(1) 二进制码在物理上最容易实现。例如, 用 1 和 0 分别表示高、低两个电位或脉冲的有无等, 可靠性都很高。

(2) 二进制码用来表示二进制数, 其编码、加减运算规则简单。

(3) 二进制码的两个符号 1 和 0 正好与逻辑数据“真”与“假”相对应, 为计算机实现逻辑运算带来了方便。

因此, 在计算机中使用的是二进制计数。二进制数的特点是“逢二进一, 借一当二”, 需要用到的数字符号是两个, 分别为 0 和 1。

任意一个十进制数可以用位权表示, 位权就是某个固定位置上的计数单位。在十进制数中, 个位的位权为 10^0 , 十位的位权为 10^1 , 百位的位权为 10^2 , 千位的位权为 10^3 , 而在

小数点后第一位上的位权为 10^{-1} ，小数点后第二位的位权为 10^{-2} ，依此类推。例如，如果有十进制数 534.23，则百位上的 5 表示五个 100，十位上的 3 表示三个 10，个位上的 4 表示四个 1，小数点后第一位上的 2 表示两个 0.1，小数点后第二位上的 3 表示三个 0.01，用位权表示如下：

$$(534.23)_{10} = 5 \times 10^2 + 3 \times 10^1 + 4 \times 10^0 + 2 \times 10^{-1} + 3 \times 10^{-2}$$

因此，对于一个任意进制的数 N ，用 r 表示该数的基数，那么该数 N 可以表示为：

$$N = a_n \times r^n + a_{n-1} \times r^{n-1} + a_{n-2} \times r^{n-2} + \cdots + a_0 \times r^0 + b_1 \times r^{-1} + b_2 \times r^{-2} + b_3 \times r^{-3} + \cdots + b_m \times r^{-m}$$

其中， $a_i (0 \leq i \leq n)$ 和 $b_j (1 \leq j \leq m)$ 是 $0 \sim r-1$ 个数字中任意一个， r^i 是各位上对应的权。

同理，二进制数 101.11，用位权表示如下：

$$(101.11)_2 = 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 + 1 \times 2^{-1} + 1 \times 2^{-2}$$

在二进制数中，各位上的权是 2^k ，因此任意的一个二进制数 N 可以表示为：

$$N = a_n \times 2^n + a_{n-1} \times 2^{n-1} + a_{n-2} \times 2^{n-2} + \cdots + a_0 \times 2^0 + b_1 \times 2^{-1} + b_2 \times 2^{-2} + b_3 \times 2^{-3} + \cdots + b_m \times 2^{-m}$$

1.4.2 二进制数与十进制数、八进制数、十六进制数的对应关系

我们已经知道，计算机内部采用二进制数表示，而人们习惯使用十进制数表示。为了方便，需要将十进制数与二进制数进行相互转换。二进制数与十进制数的对应关系如表 1.1 所示。

表 1.1 二进制数与十进制数的对应关系

二进制数	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001
十进制数	0	1	2	3	4	5	6	7	8	9

在计算机中，用二进制数表示某个数时，由于位数太长不便于记忆与书写，需要将它转换为十六进制数。十六进制数的基数是 16，用于表示的数字除了 $0 \sim 9$ 外，还包括字母 A、B、C、D、E 和 F，分别用来表示十进制数的 10、11、12、13、14 和 15。八进制数的基数是 8，各个数位的数字包括 $0 \sim 7$ 共八个数字。二进制数、八进制数、十六进制数和十进制数的对应关系如表 1.2 所示。

表 1.2 二进制数、八进制数、十六进制数和十进制数的对应关系

二进制数	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
十六进制数	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
八进制数	0	1	2	3	4	5	6	7								
十进制数	0	1	2	3	4	5	6	7	8	9						

从表 1.2 中不难看出，表示一个十六进制数需要使用 4 位二进制数，表示一个八进制数需要使用 3 位二进制数，表示一个十进制数需要使用 4 位二进制数。

1.4.3 二进制数、十六进制数和八进制数转换为十进制数

要将一个二进制数、十六进制数或八进制数转换为对应的十进制数非常容易，方法为：将二进制数(十六进制数或八进制数)中的各位数乘以对应的权，然后求和就可以得到相应



的十进制数。例如,要将二进制数 101101 和 1011.01 转换为对应的十进制数,只需要将各位上的数字乘以相应的权 2^k ,就可以得到对应的十进制数,其转换过程如下。

$$(101101)_2 = 1 \times 2^5 + 0 \times 2^4 + 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 = 32 + 8 + 4 + 1 = (45)_{10}$$

$$(1011.01)_2 = 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 + 0 \times 2^{-1} + 1 \times 2^{-2} = 8 + 2 + 1 + 0.25 = (11.25)_{10}$$

同理,要将一个 16 进制数转换为对应的十进制数,只需要将各位上的数字乘以相应的权 16^k ,就可以得到对应的十进制数,其转换过程如下。

$$(5DE)_{16} = 5 \times 16^2 + 13 \times 16^1 + 14 \times 16^0 = 1280 + 208 + 14 = (1502)_{10}$$

$$(3DA.B6)_{16} = 3 \times 16^2 + 13 \times 16^1 + 10 \times 16^0 + 11 \times 16^{-1} + 6 \times 16^{-2} = 768 + 208 + 10 + 0.6875 + 0.0234375 = (986.7109375)_{10}$$

同理,要将一个八进制数转换为对应的十进制数,只需要将各位上的数字乘以相应的权 8^k ,就可以得到对应的十进制数,其转换过程如下。

$$(602)_8 = 6 \times 8^2 + 0 \times 8^1 + 2 \times 8^0 = 384 + 0 + 2 = (386)_{10}$$

$$(732.54)_8 = 7 \times 8^2 + 3 \times 8^1 + 2 \times 8^0 + 5 \times 8^{-1} + 4 \times 8^{-2} = 448 + 24 + 2 + 0.625 + 0.0625 = (474.6875)_{10}$$

说明:下标 2 表示该数是一个二进制数,下标 10 表示该数是一个十进制数。

1.4.4 十进制数转换为二进制数

我们知道,一个完整的十进制数可以分为两个部分:整数部分和小数部分。要将十进制数转换为对应的二进制数,需要分成两个部分进行转换,即整数部分的转换和小数部分的转换。将一个十进制数的整数转换为二进制数的方法主要有两种:除 2 取余法和降幂法。

1. 除 2 取余法——十进制整数转换为二进制整数

除 2 取余法,就是将要转换的十进制数不断地除以 2,得到商和余数,并记下余数;然后将商作为新的被除数,继续除以 2,得到商和余数,并记下余数。重复以上的过程,直到商为 0 为止。将每次得到的余数按照先后顺序构成的 0 和 1 的序列,就是转换后的一个二进制整数从低位到高位序列。

例如,将一个十进制数 94 转换为对应的二进制数,其转换过程如图 1.1 所示。

$$\begin{array}{rcl}
 2 & \overline{) 94} & \\
 2 & \overline{) 47} & 94/2 \text{ 的余数为 } 0, a_0=0 \\
 2 & \overline{) 23} & 47/2 \text{ 的余数为 } 1, a_1=1 \\
 2 & \overline{) 11} & 23/2 \text{ 的余数为 } 1, a_2=1 \\
 2 & \overline{) 5} & 11/2 \text{ 的余数为 } 1, a_3=1 \\
 2 & \overline{) 2} & 5/2 \text{ 的余数为 } 1, a_4=1 \\
 2 & \overline{) 1} & 2/2 \text{ 的余数为 } 0, a_5=0 \\
 & 0 & 1/2 \text{ 的余数为 } 1, a_6=1
 \end{array}$$

$$(94)_{10} = (a_6 a_5 a_4 a_3 a_2 a_1 a_0)_2 = (1011110)_2$$

图 1.1 使用除 2 取余法将十进制数转换为二进制数的过程示意图

因此,将一个十进制数 $(94)_{10}$ 转换后可以得到相应的二进制数 $(1011110)_2$ 。