



励志改变人生
编程改变命运



零基础学 C语言

(第2版)

11.5小时多媒体教学视频

康莉 李宽 陈国建 等编著

本书特色

- ◎ 由浅入深，循序渐进，从零开始学C语言，一点都不难
- ◎ 编程基础、编程进阶、编程应用、项目实战、上机练习、面试指南
- ◎ 210个实例、41个练习题、24个面试题

超值、大容量DVD

- ◎ 本书教学视频、本书源代码
- ◎ 本书教学PPT、本书习题答案



机械工业出版社
China Machine Press



TP312C
K253-2.02



郑州大学 *04010778422Z*

4

零基础学 C语言



(第2版)

11.5小时多媒体教学视频

康莉 李宽 陈国建 等编著



TP312C
K253-2.02



机械工业出版社
China Machine Press

零基础学
编程



DVD-ROM

本书站在零基础学习的角度，使初学者能尽快掌握C语言程序设计的精髓，并且少走弯路。在知识点讲解时，笔者采用由浅入深的逐级递进学习方式。本书一共分为4篇，循序渐进地讲述了C语言的语法规则和编程思想，从基本概念到具体实践、从入门知识到高阶主题、从语法语义到数据结构和算法都进行了详细的阐述。主要内容包括数据的存储和获取、屏幕的输入与输出、运算符、表达式、分支语句、循环语句、函数、数组、指针、字符串处理、结构体、共用体、枚举、位运算、文件处理、作用域、预处理、数据结构等知识点。最后一章通过一些常见的C语言面试题的解析，为读者踏入职场提供参考资料。

本书非常适合无C语言基础或基础薄弱的程序员阅读，并可作为开发人员的参考手册。

封底无防伪标均为盗版

版权所有，侵权必究

本书法律顾问 北京市展达律师事务所

图书在版编目 (CIP) 数据

零基础学C语言 / 康莉等编著. —2版. —北京: 机械工业出版社, 2012.3
(零基础学编程)

ISBN 978-7-111-37292-9

I. 零… II. 康… III. C语言—程序设计 IV. TP312

中国版本图书馆CIP数据核字 (2012) 第014300号

机械工业出版社 (北京市西城区百万庄大街22号 邮政编码 100037)

责任编辑: 陈佳媛

北京京师印务有限公司印刷

2012年3月第2版第1次印刷

185mm × 260mm · 26.5印张

标准书号: ISBN 978-7-111-37292-9

ISBN 978-7-89433-312-4 (光盘)

定价: 59.00元 (附光盘)

凡购本书, 如有缺页、倒页、脱页, 由本社发行部调换

客服热线: (010) 88378991; 88361066

购书热线: (010) 68326294; 88379649; 68995259

投稿热线: (010) 88379604

读者信箱: hzjsj@hzbook.com

前 言

C语言自1972年诞生于贝尔实验室以来，一直以其灵活和实用的特性得到了广大用户的喜爱，迅速发展成一种应用广泛的高级语言。从网站后台，到底层操作系统，从多媒体应用到大型网络游戏，均可使用C语言来开发。在工业领域，C语言也是首选的系统语言。各种操作系统，如Unix、Linux和Windows等的内核都是采用C语言和汇编语言来编写的。

创新推动着软件开发不断进步，在C语言之后，各种新的语言相继诞生，如C++、Java、C#等，但C语言的基础地位不可撼动，学完了C语言再去上面几种语言，用户会发现其中的机理是相通的，所谓万变不离其宗，改变的只是语法的形式，编程思想却没有变化。而且，很多语言的编译器或者解释器，就是用C语言编写出来的，比如风靡全球的PHP、Ruby等。

所以，C语言是程序开发的基石。希望本书能像一盏明灯，照亮读者学习C语言之路。

本书特色

本书系统全面地介绍了C语言各个方面的知识，从最简单的“Hello World”程序写起，逐步深化细化。书中对每个知识和技术要点都给予了翔实的示例及代码分析。和其他书籍中罗列代码的做法不同，本书中的代码力求短小精悍，直击要点，避免了细枝末节对读者思维的干扰。在知识点讲解的同时辅以笔者多年的C语言编程经验解析，可加深读者的理解。

本书的特点主要体现在以下几个方面。

- 本书的编排采用密切结合、循序渐进的方式，每章主题鲜明，要点突出，适合初中级读者逐步掌握C语言的语法规则和编程思想。
- 示例丰富，关键知识点都辅以示例帮助读者理解。示例程序简洁，但并不是简单的代码罗列，而是采用短小精炼的代码紧扣所讲的技术细节，并配以详细的代码解释和说明，使读者深刻印象深刻，对所学知识理解得更加透彻。
- 示例可移植性强，与编译环境和平台无关，读者可轻易地将代码复制到自己的机器上进行实验，自行实践和演练，直观体会所讲要点，感受C语言的无限魅力。本书的所有示例、源代码都附在随书光盘中，方便读者使用。
- 结构清晰，内容全面，几乎兼顾了C语言的所有特性。
- 图文并茂，帮助读者对知识点建立直观印象。
- 本书结合笔者多年的C语言编程和系统开发经验，特别标注出易错的技术点或初学者易误解的细节，读者在学习时可少走弯路，加快学习进度。
- 很多C语言书籍只讲语法规则，不讲数据结构，读者即便掌握了语法理论，也无法写出大型的C语言程序。本书介绍了数据结构和算法的知识，阐述了结构化程序设计的思想，

探讨了高质量编程的内容，为读者以后深入学习软件开发打下基础。

- 注重读者对技术点本质的理解，对诸如“编译器如何为程序实体分配内存”、“函数调用细节”等做了很多独创性的介绍。

本书内容

本书分为4篇，共23章，第一篇从C语言的基本知识讲起，使读者初步了解C语言语法和编程机制。如果将编写C语言程序比作盖房子，基本知识就相当于砖瓦水泥。第二篇讲述如何将这些基本知识组织起来以构成完整的C语言程序。第三篇介绍了进阶内容，讨论一些深层次的技术细节和理解困难、易出错的要点。第四篇介绍了案例实践和面试技巧。

第一篇（第1章~第9章）C语言基础

讲述了C语言的基础知识，包括C语言介绍、C语言程序开发步骤、不同的开发环境、C语言程序的组成、变量及数据类型、输入与输出、运算符和表达式、语句、分支、循环等。读者通过阅读本篇可对C语言有个初步而全面的认识，了解C语言的由来及强大功用，明确开发环境如何通过文本形式的代码生成二进制形式的代码，熟悉C语言程序的结构，知道如何声明变量，如何组织语句。学完本篇，读者便可自行书写简单的C语言程序。这9章的知识是进一步学习的基础。

第二篇（第10章~第15章）一窥C语言门庭

C语言博大精深，掌握了第一篇中的基础知识可以说只到了大门口。本篇从C语言的核心——函数讲起，介绍了与数组、指针、字符串和结构体相关的内容。指针是C语言的难点，也是C语言灵活性和实用性的直接体现。数组、字符串和结构体是C语言初学者容易感觉头疼的地方。学完本篇才算迈进了C语言的大门。

第三篇（第16章~第21章）C语言进阶主题

第二篇从较为独立的角度讲述了函数、数组、指针和结构体的知识，在实际应用中，这些要素彼此交叉，应用组合千变万化，这也是C语言灵活性的具体体现。本篇包括两章，分别介绍了指针和函数的技术细节，对初学者来说，理解起来可能略有难度，但这是通往高层次C语言学习的必经之路。此外，本篇还介绍了文件处理、编译及预处理、变量的生存期、作用域、可见域及数据结构方面的内容。本篇将使读者对C语言有更深入的体会和理解。

第四篇（第22章~第23章）C语言程序设计实例与面试题解析

本篇意在让读者掌握如何用C语言开发案例和实践项目。本篇提供了几种常见游戏的开发，帮助读者进一步掌握C语言的语法和经典算法。最后一章通过一些常见的C语言面试题，为读者踏入职场提供参考资料。

本书读者对象

本书作为C语言的基础教程，适合于：

- C语言的初中级读者
- 了解C语言，但所学不全面的人员
- 高等院校学习C语言课程的学生

- 使用C语言进行毕业设计的学生
- 使用C语言进行项目开发的人员
- 其他相关技术人员

本书作者

本书主要由康莉、李宽、陈国建编写，其他参与编写和资料整理的人员有：冯华君、刘博、刘燕、叶青、张军、张立娟、张艺、彭涛、徐磊、戎伟、朱毅、李佳、李玉涵、杨利润、杨春娇、武鹏、潘中强、王丹、王宁、王西莉、石淑珍、程彩虹、邵毅、郑丹丹、郑海平、顾旭光。

作者

2012年1月

目 录

前言

第一篇 C语言基础

第1章 踏上征程前的思想动员	1
1.1 为什么选择C语言	1
1.2 如何学好C语言	2
1.3 语言概述	3
1.3.1 什么是语言	3
1.3.2 什么是机器语言	3
1.3.3 什么是汇编语言	4
1.3.4 面向过程的语言	4
1.3.5 什么是C语言	4
1.4 深刻认识程序如何开发出来——程序开发周期	5
1.4.1 编辑C源代码	6
1.4.2 编译C源代码	6
1.4.3 连接目标文件	7
1.4.4 编译连接过程示例	7
1.4.5 运行程序	9
1.5 VC++、C++、C和TC——认识开发环境和语言的区别	9
1.6 小结	11
1.7 上机实践	11
第2章 跟我写Hello World	12
2.1 自己动手创建程序	12
2.2 认识LCC-Win32开发环境	12
2.2.1 为什么选择LCC-Win32	13
2.2.2 启动LCC-Win32	13
2.2.3 新建Hello World工程	13
2.2.4 定义新工程	13
2.2.5 添加源代码到工程	14
2.2.6 编译器设置	15
2.2.7 连接器设置	15

2.2.8 调试器设置	15
2.2.9 开始编辑代码	16
2.3 编译运行——欣赏你的杰作	17
2.4 查看代码如何一步一步被执行——调试排错 (debug)	18
2.4.1 debug一词的由来	18
2.4.2 设置断点	19
2.4.3 调试运行	19
2.5 小结	20
2.6 上机实践	20
第3章 分解HelloWorld——最简单C程序的组成	22
3.1 C程序概貌	22
3.2 为何有必要在代码里留下程序员的解释——注释	23
3.3 从哪里开始，到哪里结束——main函数	24
3.4 如何在屏幕上显示文字——函数调用	25
3.5 编译器如何认识printf函数——#include预处理器指示符	26
3.5.1 函数声明及其作用	26
3.5.2 试验寻找#include的作用	27
3.5.3 #include的作用	28
3.6 计算1+1——有些小作为的程序	29
3.7 数据从哪里来，又到哪里去——变量	30
3.7.1 变量在内存中的表现形式	30
3.7.2 编译器使用变量符号表	30
3.7.3 变量及其使用	31
3.8 自己设计C函数	32
3.8.1 在main函数中计算3个整数的平均数	33
3.8.2 在main函数中分3次计算3个整数的平均数	33
3.8.3 自编函数实现计算3个整数的平均数	34
3.8.4 如何自编写函数	35
3.8.5 试验观察总结函数声明和函数定义的意义	36
3.9 语句构成程序	37
3.10 向好程序员看齐——代码风格	38
3.11 小结	38
3.12 上机实践	38
第4章 如何存储和获取数据——变量及数据类型	40
4.1 二进制、八进制、十六进制	40
4.2 计算机中如何表示数据——字节和位	41
4.3 内存是存储数据的“房间”——认识计算机内存	43
4.4 数据类型	43
4.4.1 整数类型	44
4.4.2 整数的正负——有符号和无符号	46



4.4.3	实数类型	47
4.4.4	字符类型	48
4.4.5	数据类型总结	48
4.5	常量	49
4.5.1	直接常量(字面常量)和符号常量	49
4.5.2	符号常量的优点	51
4.5.3	字面常量的书写格式	52
4.5.4	转义字符	53
4.5.5	字符串	55
4.6	变量	55
4.6.1	变量的声明	56
4.6.2	变量声明的意义	56
4.6.3	标识符和关键字	57
4.6.4	变量在内存中占据的空间和变量的值	60
4.6.5	变量赋初值	62
4.7	几个与变量相关的经典算法	63
4.7.1	累加和累乘	63
4.7.2	交换两个变量的值	65
4.8	小结	66
4.9	上机实践	66
第5章	用屏幕和键盘交互——简单的输出和输入	68
5.1	输入—处理—输出：这就是程序	68
5.2	向屏幕输出——printf函数详解	68
5.2.1	printf函数的一般形式	69
5.2.2	格式字符串的一般形式	71
5.2.3	%d——十进制形式输出带符号整数	71
5.2.4	%最小宽度d格式符	72
5.2.5	%ld格式符——以长整型方式输出	73
5.2.6	%o格式符——以八进制形式输出整数	73
5.2.7	%x格式符——以十六进制的形式输出整数	74
5.2.8	%u格式符——以十进制形式输出unsigned型数据	74
5.2.9	%c格式符	75
5.2.10	%s格式符——输出字符串	76
5.2.11	%f格式符——输出实型数据	78
5.2.12	%e格式符——以指数形式输出实数	79
5.2.13	%g格式符——自动选择%f或者%e形式输出	80
5.2.14	printf函数小结	80
5.3	字符输出putchar和字符串输出puts	81
5.3.1	putchar函数(字符输出函数)	81
5.3.2	puts函数(字符串输出函数)	83

5.4 接收键盘输入——scanf函数详解	84
5.4.1 一般形式	84
5.4.2 scanf函数的工作原理	85
5.4.3 多数据输入分隔规则	86
5.4.4 格式字符	88
5.4.5 scanf使用方法说明	89
5.4.6 使用scanf函数的注意事项	91
5.5 字符输入函数getchar	93
5.6 输入输出程序举例	94
5.7 小结	97
5.8 上机实践	97
第6章 程序的基本构成——运算符和表达式	98
6.1 认识C中的运算符和表达式	98
6.1.1 认识运算符和表达式	98
6.1.2 操作数、运算符和表达式	99
6.1.3 C运算符简介	99
6.2 算术运算符和算术表达式	100
6.2.1 基本的算术运算符	100
6.2.2 ++自增、--自减运算符	102
6.2.3 算术表达式和运算符的优先级及结合性	104
6.3 逗号运算符和逗号表达式	105
6.3.1 逗号表达式的一般形式	105
6.3.2 逗号表达式的优先级和结合性	106
6.4 关系运算符和关系表达式	107
6.4.1 关系运算符的种类	107
6.4.2 关系表达式的一般形式	107
6.5 逻辑运算符和逻辑表达式	109
6.5.1 逻辑运算符	109
6.5.2 逻辑真值表	109
6.6 赋值运算符和赋值表达式	110
6.6.1 赋值表达式	110
6.6.2 复合赋值运算符	111
6.6.3 类型转换	111
6.6.4 左值与程序实体	112
6.7 强制类型转换和自动类型转换	112
6.7.1 强制类型转换	112
6.7.2 自动类型转换	113
6.7.3 函数调用时的转换	115
6.8 运算符的优先级	115
6.8.1 优先级、结合性汇总	115



6.8.2 短路表达式	116
6.8.3 sizeof运算符与sizeof表达式	116
6.9 小结	116
6.10 上机实践	117
第7章 程序的最小独立单元——语句	118
7.1 5种语句类型	118
7.1.1 表达式语句	118
7.1.2 函数调用语句	119
7.1.3 空语句	119
7.1.4 复合语句（块语句）	119
7.1.5 流程控制语句	120
7.2 结构化程序设计	120
7.2.1 什么是算法	120
7.2.2 算法的表示	120
7.2.3 算法的伪代码表示	121
7.2.4 算法的流程图表示	121
7.2.5 3种控制结构	121
7.2.6 算法示例	122
7.3 小结	122
7.4 上机实践	122
第8章 条件判断——分支结构	123
8.1 简单的条件判断——if语句	123
8.1.1 判断表达式	123
8.1.2 花括号和if结构体	124
8.2 两条岔路的选择——if else结构	125
8.2.1 关键在else	125
8.2.2 if else结构体	126
8.3 多分支语句和分支语句嵌套	127
8.3.1 多分支	127
8.3.2 多分支if结构	128
8.3.3 分支语句嵌套	130
8.4 开关语句——switch结构	131
8.4.1 一般形式	131
8.4.2 为什么叫开关语句	133
8.4.3 default语句	135
8.4.4 if结构和switch结构之比较	135
8.5 小结	135
8.6 上机实践	135
第9章 一遍又一遍——循环结构	137
9.1 构造循环	137

9.1.1 循环的条件	137
9.1.2 当型循环和直到型循环	138
9.2 简洁循环语句——while和do while结构	138
9.2.1 语法规则	139
9.2.2 代码块	140
9.3 最常用的循环语句——for结构	140
9.3.1 基本形式	140
9.3.2 表达式省略	142
9.3.3 循环终止和步长	143
9.4 循环嵌套	143
9.4.1 嵌套示例	144
9.4.2 嵌套的效率	144
9.5 与循环密切相关的流程转向控制语句	145
9.5.1 跳出循环——break	145
9.5.2 重来一次——continue	146
9.5.3 随心所欲——goto	147
9.6 小结	148
9.7 上机实践	148

第二篇 一窥C语言门庭

第10章 写程序就是写函数——函数入门	149
10.1 什么是函数——根据输入进行处理返回输出	149
10.1.1 分割	149
10.1.2 库函数和自定义函数	150
10.2 自定义函数	150
10.2.1 定义的语法	150
10.2.2 函数定义范例	151
10.2.3 不要重复定义	152
10.3 函数调用与返回	152
10.3.1 形参和实参	152
10.3.2 传址调用	154
10.3.3 函数返回	155
10.4 告诉编译器有这么一个函数——函数原型声明	156
10.4.1 函数声明的语法	156
10.4.2 声明不同于定义	158
10.4.3 标准库函数的声明	158
10.5 面向过程的程序结构	158
10.5.1 模块化	159
10.5.2 函数的调用过程——模块的配合	159

10.5.3	一个入口一个出口	159
10.5.4	自顶向下, 逐步求精	159
10.5.5	封装和可重用	160
10.5.6	高内聚, 低耦合	160
10.6	小结	160
10.7	上机实践	161
第11章	同一类型多个元素的集合——简单数组	162
11.1	什么是数组	162
11.1.1	数组是一大片连续内存空间	162
11.1.2	数组元素的访问	163
11.2	一维数组	163
11.2.1	一维数组的声明	163
11.2.2	一维数组元素的访问	164
11.2.3	数组的初始化	164
11.2.4	不合法的数组操作	165
11.3	二维数组	166
11.3.1	二维数组的声明	166
11.3.2	二维数组的初始化	167
11.3.3	二维数组应用举例	167
11.4	更高维的数组	168
11.4.1	高维数组的声明和元素访问	169
11.4.2	初始化	169
11.4.3	多维数组在内存中是如何排列元素的	170
11.5	小结	170
11.6	上机实践	171
第12章	C语言难点——指针初探	172
12.1	计算机中的内存	172
12.1.1	内存地址	172
12.1.2	内存中保存的内容	173
12.1.3	地址就是指针	173
12.2	指针的定义	173
12.2.1	指针变量的声明	174
12.2.2	指针变量的初始化	174
12.2.3	指针变量的值	174
12.2.4	取地址操作符&	175
12.2.5	指针变量占据一定的内存空间	175
12.2.6	指向指针的指针	176
12.3	使用指针	176
12.3.1	运算符*	176
12.3.2	指针的类型和指针所指向的类型	177

12.3.3	同类型指针的赋值	178
12.3.4	指针的类型和指针所指向的类型不同	178
12.4	指针的运算	180
12.4.1	算术运算之“指针+整数”或“指针-整数”	180
12.4.2	指针-指针	181
12.4.3	指针的大小比较	182
12.5	指针表达式与左值	183
12.5.1	指针与整型	183
12.5.2	指针与左值	183
12.5.3	指针与const	183
12.6	动态内存分配	185
12.6.1	动态分配的好处	185
12.6.2	malloc与free函数	185
12.6.3	calloc与free函数	187
12.6.4	free函数与指针	187
12.6.5	内存泄漏	188
12.6.6	释放已经释放了的内存会出问题	188
12.7	小结	189
12.8	上机实践	189
第13章	字符串及字符串操作	190
13.1	C风格字符串	190
13.1.1	C风格字符串的声明	190
13.1.2	C风格字符串在内存中的表示	190
13.2	字符数组的输入输出	191
13.2.1	字符数组可以进行整体输入输出	191
13.2.2	使用gets函数读入整行输入	192
13.2.3	访问字符数组中某个元素	193
13.2.4	更便捷的输出——使用puts函数	193
13.3	字符串处理函数	194
13.3.1	理解：数组名是常指针	194
13.3.2	strlen函数与size_t	196
13.3.3	字符串复制函数strcpy	197
13.3.4	字符串比较函数strcmp	198
13.3.5	字符串连接函数strcat	199
13.3.6	全转换为大写形式	200
13.3.7	链式操作	201
13.4	小结	201
13.5	上机实践	201
第14章	结构体、共用体、枚举和typedef	203
14.1	结构体	203

14.1.1 结构体的定义	203
14.1.2 声明结构体变量	204
14.1.3 初始化结构变量	204
14.1.4 访问结构体成员	205
14.1.5 结构体定义的位置	205
14.1.6 结构体变量赋值	206
14.2 特殊结构体	207
14.2.1 结构体嵌套	207
14.2.2 匿名结构体	210
14.3 共用体	210
14.3.1 什么是共用体	210
14.3.2 共用体的定义	210
14.3.3 声明共用体变量	211
14.3.4 共用体变量的初始化	211
14.3.5 共用体成员访问	211
14.3.6 共用体赋值	213
14.4 结构体和共用体的内存差异	213
14.4.1 结构体变量和共用体变量内存形式的不同	213
14.4.2 结构体变量的内存大小	213
14.4.3 字节对齐	214
14.4.4 最宽基本类型	215
14.4.5 共用体的大小	216
14.5 枚举类型	216
14.5.1 枚举类型的定义	216
14.5.2 声明枚举变量	217
14.5.3 枚举常量是什么	217
14.6 给类型取个别名——typedef	218
14.6.1 typedef基本用法	218
14.6.2 #define用法	218
14.7 小结	219
14.8 上机实践	219
第15章 如何节省内存——位运算	220
15.1 什么是位运算	220
15.1.1 开灯关灯	220
15.1.2 改变状态	220
15.2 位逻辑运算符	221
15.2.1 位取反操作	221
15.2.2 位与运算	221
15.2.3 位或运算	221
15.2.4 位异或	221

15.2.5 实例分析	222
15.3 移位运算	223
15.3.1 基本形式	223
15.3.2 移位举例	223
15.4 小结	224
15.5 上机实践	224

第三篇 C语言进阶

第16章 存储不仅仅局限于内存——文件	227
16.1 什么是文件	227
16.1.1 文件	227
16.1.2 流	228
16.1.3 重定向	228
16.1.4 文件的处理形式——缓冲区和非缓冲区	228
16.1.5 文件的存储形式——文本形式和二进制形式	229
16.2 C语言如何使用文件	229
16.2.1 文件型指针	229
16.2.2 文件操作的步骤	230
16.3 文件的打开与关闭	230
16.3.1 打开文件——fopen	230
16.3.2 打开是否成功	231
16.3.3 关闭文件——fclose	232
16.4 文件的读写	232
16.4.1 读写的相对参照	232
16.4.2 如何判断文件已经结束	232
16.4.3 字符读写函数——fgetc和putc	233
16.4.4 字符串读写函数——fgets和fputs	234
16.4.5 块读写函数——fread和fwrite	236
16.4.6 格式化文件输入输出——fprintf与fscanf	239
16.5 文件的定位	241
16.5.1 移到开头——rewind	241
16.5.2 得到当前位置——ftell	242
16.5.3 移动指针——fseek	242
16.6 小结	243
16.7 上机实践	243
第17章 灵活却难以理解——指针进阶	245
17.1 指针与数组	245
17.1.1 数组名指针	245
17.1.2 使用数组名常指针表示数组元素	246



17.1.3 指向数组元素的指针变量	246
17.1.4 指向数组的指针变量	247
17.1.5 指针数组	248
17.2 指针、结构体和结构体数组	249
17.2.1 两种访问形式	249
17.2.2 声明创建一个结构数组	249
17.2.3 结构数组的初始化	250
17.2.4 结构数组的使用	251
17.2.5 指向结构数组的指针	251
17.3 函数指针	252
17.3.1 函数名指针	252
17.3.2 指向函数的指针	253
17.3.3 函数指针类型	254
17.3.4 函数指针做函数参数	256
17.3.5 函数指针数组	256
17.3.6 指向函数指针的指针	258
17.4 小结	258
17.5 上机实践	258
第18章 更深入的理解——函数进阶	260
18.1 参数传递的副本机制	260
18.1.1 传值调用的副本机制	260
18.1.2 传址调用的副本机制	261
18.2 函数返回值的副本机制	264
18.2.1 return局部变量为什么合法	264
18.2.2 返回指针申请动态内存	265
18.2.3 不要返回指向栈内存的指针	266
18.2.4 返回指向只读存储区的指针	266
18.3 函数与结构体	267
18.3.1 结构体变量的传值和传址调用	267
18.3.2 结构体变量的成员作为函数参数	268
18.3.3 返回结构体的函数	269
18.3.4 返回结构体指针的函数	270
18.4 函数与数组	270
18.4.1 数组元素作为函数参数	270
18.4.2 数组名作为函数参数	271
18.4.3 多维数组名作为函数参数	273
18.4.4 数组名作为函数参数时的退化	274
18.5 递归	274
18.5.1 递归流程	274
18.5.2 递归两要素	276