

.....典藏版.....

百科全书式的C语言编程宝典，值得每个程序员珍藏
囊括C语言从基础知识到高级开发技术的全景式解读

C语言 开发手册

薛园园◎等编著



超值DVD内容

- ★ 本书涉及的实例源文件
- ★ 本书配套教学视频
- ★ 本书教学PPT
- ★ 赠送40小时相关技术多媒体视频教程



超值DVD内容

- ★ 356个实例，12.5小时教学视频
- ★ 一线工程师多年C语言开发经验的总结
- ★ 内容全面、深入，囊括多个开发领域
- ★ 深入分析指针操作，解决疑难问题
- ★ 特别介绍了接口开发和不同环境的C语言开发



电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
<http://www.phei.com.cn>

.....典藏版.....

C语言 开发手册

薛园园◎等编著

电子工业出版社

Publishing House of Electronics Industry

北京·BEIJING

内 容 提 要

毋庸置疑, C 语言从产生到现在的各个阶段, 都是最重要和最流行的编程语言之一。熟悉和掌握 C 语言程序设计, 已经成为程序员的一门必修课。本书言简意赅、通俗易懂, 知识点覆盖全面, 详细介绍了 C 语言的程序设计及最新的 C 语言编程技术。

本书分 4 篇, 共 29 章, 分别介绍了 C 语言基础、C 语言进阶、C 语言的接口技术和现代 C 语言的扩展。首先, 介绍了 C 语言及其开发环境, 并全面介绍了 C 语言的基本语法知识; 接着, 详细讲解了 C 语言的一些高级程序应用; 然后, 介绍了 C 语言与其他主流编程语言的接口; 最后, 讲解了现代 C 语言的一些扩展应用和高级技术。

本书结构安排紧凑、讲解详细、实例丰富。C 语言程序设计的初学者通过本书可以快速掌握 C 语言程序设计方法。同时, 本书对现代 C 语言程序的接口和高级技术进行了全面讲解, 对具有一定开发经验的设计人员也有非常好的参考价值。本书可以作为 C 语言初学者的入门教材, 也可作为 C 语言程序员的参考手册, 还可作为大中专院校学生及电脑培训班的参考书。

未经许可, 不得以任何方式复制或抄袭本书之部分或全部内容。

版权所有, 侵权必究。

图书在版编目(CIP)数据

C 语言开发手册(典藏版) / 薛园园等编著. —北京: 电子工业出版社, 2011.4
ISBN 978-7-121-12120-3

I. ①C… II. ①薛… III. ①C 语言—程序设计 IV. ①TP312

中国版本图书馆 CIP 数据核字(2010)第 208712 号

责任编辑: 高洪霞

印 刷: 涿州市京南印刷厂

装 订: 涿州市桃园装订有限公司

出版发行: 电子工业出版社

北京市海淀区万寿路 173 信箱 邮编 100036

开 本: 787×1092 1/16 印张: 51.5 字数: 1271 千字

印 次: 2011 年 4 月第 1 次印刷

印 数: 4000 册 定价: 99.00 元(含 DVD 光盘 1 张)

凡所购买电子工业出版社图书有缺损问题, 请向购买书店调换。若书店售缺, 请与本社发行部联系, 联系及邮购电话: (010) 88254888。

质量投诉请发邮件至 zltz@phei.com.cn, 盗版侵权举报请发邮件至 dbqq@phei.com.cn。

服务热线: (010) 88258888。

前言

PREFACE

C 语言从产生到现在经历了几十年的发展，一直经久不衰，已成为最重要和最流行的编程语言之一。目前虽然产生了许多新的编程语言，例如 C++、Java、C51 等，但这些语言都是借鉴 C 语言而发展起来的。因此，C 语言是每一个计算机技术人员的基本功之一。

C 语言具有高级语言的强大功能，却又有很多直接操作计算机硬件的功能（这些都是汇编语言的功能），因此 C 语言通常又被称为中级语言。学习和掌握 C 语言，既可以增进对于计算机底层工作机制的了解，又为进一步学习其他高级语言打下了坚实的基础。同时，随着时代的发展，C 语言也有了很大的发展，例如嵌入式领域的应用、并行计算、和其他语言接口等。

和其他书籍相比，本书有如下优点

(1) 本书不仅详细介绍了 C 语言程序设计的基础知识，还对常用的高级编程技术进行了详细的讲解。

(2) 本书在讲解每个知识点的同时，均给出了其在程序设计中的应用实例，每个实例都可以通过编译执行，使得读者可以快速掌握对应知识点如何在程序设计中应用。

(3) 本书不仅全面介绍了 C 语言的程序设计方法，还详细讲解了 C 语言如何与最新流行的编程语言进行接口，这是 C 语言的最新发展。

(4) 针对 C 语言的最新发展，本书还介绍了其在嵌入式、单片机及并行计算方面的应用。这些发展使得 C 语言保持了经久不衰的活力。

(5) 本书对每一个实例的程序代码都进行了详细的注释和分析，并给出了运行结果，使得读者更加容易理解。

(6) 本书采用应用最为广泛的编译环境，并且适用于多种操作系统。读者既可在 Windows 环境下、又可在 Linux 环境下使用本书学习 C 语言。

本书的内容

本书以实用性、系统性、完整性和前沿性为重点，详细介绍了 C 语言的程序设计方法、接口技术及现代的一些最新发展。本书分为 4 篇，共 29 章内容。

篇章序号		主要内容
第 1 篇是 C 语言基础篇，共分为 13 章，详细介绍了 C 语言的基础知识和基本语法等	第 1 章 C 语言概述	介绍了 C 语言的发展、特点、C 语言的标准、开发流程，并以一个简单的 C 程序演示了编写 C 语言程序的方法。同时本章还分析了 C 程序构成、编程规范以及 C 语言在不同领域的扩展应用
	第 2 章 典型的 C 语言开发环境	介绍了 C 语言几种常见的开发环境，包括 Turbo C、Dev-C++、Visual C++、Code::Blocks 开发环境的使用等内容

续表

篇章序号		主要内容
第1篇是C语言基础篇，共分为13章，详细介绍了C语言的基础知识和基本语法等	第3章 C语言数据类型	介绍了C语言数据的表示方法，包括数据的存储、常量、变量、整数类型、字符类型、实数类型、混合运算及类型转换、C99 新增数据类型等内容
	第4章 运算符和表达式	详细介绍C语言中的表达式、各种运算符及运算优先级
	第5章 C语句及流程控制结构	详细讲解了C语言中的各种语句类型
	第6章 格式化输入/输出	介绍了格式化输入/输出函数的使用，包括printf()函数、scanf()函数、getchar()函数、getch()函数、gets()函数、putch()函数、puts()函数等的使用
	第7章 数组	介绍在C语言中使用数组的方法，主要介绍了一维数组、二维数组、字符数组和字符串的使用
	第8章 函数	介绍在C语言中创建和使用函数的方法，包括函数的概念、编写函数、设置函数的参数、函数调用、递归函数的编写等内容
	第9章 指针	介绍C语言中指针的使用，指针是C语言最显著的特征。这里首先介绍了变量的保存形式，然后介绍了指针和简单变量、指针和数组、指向多维数组的指针、指针和字符串、指针数组、指针和函数等内容
	第10章 结构	包括结构的概念、结构的嵌套、结构数组、结构指针、向函数传递结构等内容
	第11章 联合、枚举和位域	讲解了联合、枚举和位域的概念及使用
	第12章 预处理	介绍了宏定义命令、条件编译、其他预处理命令、内联函数等编译预处理内容
	第13章 存储管理	介绍用C语言进行存储管理的方法，包括内存动态分配和回收等内容
第2篇为C语言进阶篇，共分为7章，详细讲解了C语言程序设计中的高级技术	第14章 字符及字符串处理	介绍了C语言处理字符串的方法，包括字符串的存储、字符和字符串检测函数、查找字符串、比较字符串、连接字符串、复制字符串、字符和字符串的转换等内容
	第15章 文件操作	主要介绍了数据流的概念、文件的打开与关闭、从文件中读写字符、从文件中读写字符串、二进制文件的读写、文件检测函数、文件的随机读写、管理缓冲区、输入/输出的重定向、文件管理等内容
	第16章 典型库函数应用	主要介绍了随机函数、数学函数、日期和时间函数的应用
	第17章 模块化的程序设计方法	包括模块化程序设计概念、C 程序的组织、变量的作用域、变量的存储类型等内容
	第18章 典型算法应用	介绍常用算法的C语言程序，包括用C语言编写的完成排序、查找、队列、堆栈、链表等程序
	第19章 图形绘制	介绍C语言底层图形函数的编程，首先介绍了显示系统的基本概念，然后介绍了直接读取显存编写图形函数的方法，在此基础上完成绘制线、圆、矩形、多边形等函数，最后还介绍了编辑图形的方法
	第20章 鼠标控制	介绍用C语言编写鼠标的底层操作函数，包括初始化鼠标、复位鼠标、显示鼠标指针、隐藏鼠标指针、获取按钮信息等操作鼠标的底层函数

续表

篇章序号	主要内容
第3篇为C语言接口篇，共分为5章，主要介绍了目前一些主流的编程语言，以及如何使用C语言及其接口来发挥各自的优势，从而完成程序设计任务	第21章 C语言与汇编的接口 主要介绍了C语言调用汇编程序的方法，包括AT&T汇编简介、GCC内嵌汇编、调用汇编程序等内容
	第22章 C语言与LabVIEW的接口 主要介绍了图形化程序设计语言LabVIEW、CIN结点，以及如何使用CIN结点来实现C语言与LabVIEW的接口
	第23章 C语言与MATLAB的接口 主要介绍了MATLAB语言，以及如何使用MATLAB引擎、COM组件、MEX文件和Matcom等实现C语言与MATLAB的接口
	第24章 C语言与JavaScript的接口 首先简单介绍了JavaScript，然后重点介绍了JavaScript-C引擎及JavaScript-C引擎的开发，并通过实例讲解了如何在C/C++中执行JavaScript程序
	第25章 C语言硬件端口操作 首先介绍了并行端口，然后介绍如何在C语言环境下对并行端口进行操作。这部分内容充分体现了C语言既有高级语言的特点，又有低级语言操作硬件能力的特点
第4篇为C语言扩展篇，共分为4章，主要介绍了C语言在嵌入式、Linux平台及高性能并行计算等方面的应用	第26章 Linux下的C语言开发 首先介绍了Linux系统及Linux常用命令，然后介绍了Linux C开发环境及其程序开发
	第27章 单片机的C语言开发 主要讲解了C语言在单片机开发中的扩展应用。重点介绍了C51语言的开发工具及其与标准C语言的区别，本章还通过实例介绍了单片机的C语言开发
	第28章 嵌入式系统C语言开发 主要介绍了嵌入式系统、嵌入式Linux C开发工具、集成开发环境Eclipse以及嵌入式C语言开发流程
	第29章 基于C语言的高性能并行GPU编程 主要介绍了GPU的概念、NVIDIA的基于C语言的CUDA、CUDA应用程序编程接口。然后，重点讲解了如何在Visual Studio环境中使用CUDA，并通过实例讲述了基于C语言的CUDA并行编程

本书结构紧凑，知识点涉及全面，内容翔实，案例丰富。由于本书内容较多，编写时间较仓促，书中如有遗漏或不足之处，恳请广大读者提出宝贵意见，以便于进一步改进。

适合的读者

- 大学、大专等相关专业的学生及教师
- 系统开发人员
- 程序设计初学者
- C程序员
- 电脑培训班
- 计算机爱好者

第 1 篇 C 语言基础篇

第 1 章 C 语言概述

2

C 语言是现代应用最为广泛的编程语言，深受广大程序员的喜爱。本章将简要介绍 C 语言的产生发展、C 语言的特点、标准及 C 语言流程等内容。并通过一个完整的例子演示 C 程序开发过程中的项目创建、源代码编写、编译及调试的全过程。同时，将分析 C 语言程序构成特点。

1.1 C 语言发展历史	2
1.2 C 语言的特点	3
1.2.1 C 语言的基本特点	3
1.2.2 C 语言的优点	4
1.2.3 C 语言的缺点	5
1.3 C 语言的标准	5
1.3.1 C 标准概述	5
1.3.2 C 语言的新标准	6
1.3.3 编译器对 C 标准的支持	10
1.4 C 语言开发流程	11
1.4.1 功能分析	11
1.4.2 划分任务	11
1.4.3 编写源代码	12
1.4.4 编译和链接	12
1.4.5 修改源代码	13
1.4.6 功能测试	13
1.4.7 发布程序	13
1.4.8 维护和修改	13
1.5 第一个 C 程序	13
1.5.1 创建项目	14
1.5.2 输入源代码	15
1.5.3 编译链接	16
1.5.4 调试	16
1.6 C 程序构成分析	17
1.6.1 头文件	17
1.6.2 主函数	17
1.6.3 其他组成部分	18
1.7 良好的编程规范	21
1.7.1 完备和详尽的程序注释	21
1.7.2 整齐的代码书写格式	21
1.7.3 简洁清晰的命名	22

1.8 C 语言在不同领域的扩展应用	22
1.8.1 单片机方面	22
1.8.2 LabVIEW 方面	22
1.8.3 LabWindows/CVI 方面	23
1.8.4 MATLAB 方面	23
1.8.5 JavaScript 方面	23
1.8.6 并行计算方面	24
1.8.7 Linux 平台	24
1.9 小结	24

第2章 典型的 C 语言开发环境

25

C 语言的成功及大范围的应用也造就了丰富的 C 语言编译器。对于用户来说,学习 C 语言要紧密结合 C 编译器或者 C 集成开发环境来进行。程序员要想快速高效地开发 C 程序,必须首先熟悉 C 语言开发环境的使用。本章将介绍几种主流的 C 语言开发环境。

2.1 C 语言开发环境简介	25
2.1.1 C 语言开发环境的构成	25
2.1.2 典型的 C 语言开发环境	26
2.2 使用 Turbo C 开发	30
2.2.1 Turbo C 的开发环境简介	30
2.2.2 Turbo C 的配置文件	31
2.2.3 用 Turbo C 创建 C 程序	31
2.2.4 编译链接程序	32
2.2.5 运行程序	33
2.3 使用 Visual Studio 开发	34
2.3.1 Microsoft Visual C++ 的开发环境简介	34
2.3.2 使用 Visual Studio 2008 创建 C 程序	35
2.3.3 编译链接程序	38
2.3.4 调试运行程序	38
2.4 使用 Dev-C++ 开发	38
2.4.1 Dev-C++ 的开发环境简介	39
2.4.2 Dev-C++ 开发环境的设置	39
2.4.3 使用 Dev-C++ 创建 C 程序	41
2.4.4 编译和运行程序	41
2.4.5 升级 Dev-C++	42
2.5 使用 Code::Blocks 开发	42
2.5.1 Code::Blocks 的开发环境简介	42
2.5.2 Code::Blocks 开发环境的设置	43
2.5.3 使用 Code::Blocks 创建 C 程序	45
2.5.4 编译运行程序	46
2.6 小结	46

第3章 C 语言数据类型

47

应用程序一般在计算机的内存中运行,对各种数据进行操作。C 语言中的数据具有不同的类型,用于不同对象的表示。数据是 C 语言基本的操作对象。本章将首先介绍内存中程序和数据的

存储,然后将介绍 C 语言中各种数据类型及其特点。

3.1	数据的存储	47
3.1.1	内存单元	47
3.1.2	数据的存储	48
3.1.3	程序在内存中的映射	49
3.2	数据类型简介	49
3.2.1	基本类型	49
3.2.2	构造类型	51
3.2.3	指针类型	51
3.2.4	空类型	51
3.3	标识符和关键字	51
3.3.1	标识符	52
3.3.2	关键字	52
3.4	分隔符	53
3.4.1	单分隔符	54
3.4.2	组合分隔符	54
3.5	常量	54
3.5.1	直接常量	54
3.5.2	符号常量	55
3.6	变量	58
3.6.1	变量声明	58
3.6.2	变量初始化	59
3.7	整型数据	60
3.7.1	整型数据的声明	60
3.7.2	整型数据的存储	61
3.7.3	整型常量	65
3.7.4	整型数据的输入	66
3.7.5	整型数据的输出	68
3.8	字符型数据	69
3.8.1	字符型变量的声明	69
3.8.2	字符型的存储	70
3.8.3	字符型常量	71
3.8.4	转义字符	72
3.8.5	字符型数据的输入	73
3.8.6	字符型数据的输出	74
3.9	浮点型数据	75
3.9.1	浮点型数据的声明	75
3.9.2	浮点型数据的存储	76
3.9.3	浮点型常量	77
3.9.4	浮点型数据的输入	78
3.9.5	浮点型数据的输出	79
3.10	类型转换	80
3.10.1	混合运算中的类型转换	80
3.10.2	强制类型转换	83

3.11 C99 新增数据类型.....	84
3.12 小结.....	86

第4章 运算符和表达式

87

第3章介绍了C语言中的基本数据类型,这些是C语言最基本的操作对象。而在第4章中,读者将了解到最基本的操作方法,即运算符和表达式。在C语言中,通过运算符将常量、变量等操作对象组成表达式,而后,由表达式进一步构成C语句。可以说,运算符和表达式是C语言中最基本的操作方法。

4.1 运算符.....	87
4.1.1 运算符概述.....	87
4.1.2 算术运算符.....	88
4.1.3 赋值运算符.....	90
4.1.4 关系运算符.....	92
4.1.5 逻辑运算符.....	93
4.1.6 条件运算符.....	94
4.1.7 位运算符.....	95
4.1.8 其他运算符.....	98
4.1.9 运算符优先级和结合性.....	102
4.2 表达式.....	104
4.2.1 算术表达式.....	104
4.2.2 赋值表达式.....	105
4.2.3 逗号表达式.....	106
4.2.4 关系和逻辑表达式.....	107
4.3 小结.....	109

第5章 C语句及流程控制结构

110

第4章介绍了C语言中的运算符和表达式,这些是构成C语句的基本单元。C语句,即C语言中的操作命令,用于使计算机完成特定的功能。一条完整的语句必须以“;”结束。C语言的源程序是由一系列的语句组成的,这些语句可以完成变量声明、赋值和控制输入/输出等操作。由于计算机能识别的是机器指令,因此,编译系统将一条语句转换成若干条机器指令来执行。C语言中的语句包括说明语句、表达式语句、循环语句、条件语句、开关语句、复合语句、空语句和返回语句等,下面分别进行介绍。

5.1 说明语句.....	110
5.2 表达式语句.....	110
5.3 复合语句.....	111
5.4 循环语句.....	112
5.4.1 while 循环语句.....	112
5.4.2 do...while 循环语句.....	113
5.4.3 for 循环语句.....	114
5.5 条件语句.....	116
5.5.1 单分支 if 条件结构.....	116
5.5.2 双分支 if 条件结构.....	116
5.5.3 阶梯式 if...else...if 条件结构.....	117

5.5.4	if 条件结构总结	119
5.6	开关语句	120
5.7	跳转语句	121
5.7.1	goto 跳转语句	121
5.7.2	break 跳转语句	122
5.7.3	continue 跳转语句	123
5.8	函数调用语句	124
5.9	空语句	125
5.10	返回语句	126
5.11	C 语言的流程控制结构	127
5.12	小结	128

第 6 章 格式化输入/输出

129

使用计算机程序解决问题时，需要由用户输入数据，通过程序进行运算后，再输出结果。由此过程可以看到，程序中输入/输出占有相当重要的地位。在 C 语言中，所有的数据输入/输出都是由库函数完成的，因此都是函数语句。在前面各章的实例中，曾使用过 printf() 函数和 scanf() 函数进行数据的输出和输入。本章将详细介绍这两个函数的使用，同时，还将介绍其他一些用于输入和输出的函数。

6.1	格式化输出函数 printf()	129
6.1.1	printf() 函数的格式	129
6.1.2	printf() 函数的格式字符	130
6.1.3	printf() 函数的修饰符	131
6.1.4	典型的 printf() 函数实例	134
6.1.5	动态设置输出宽度和精度	138
6.1.6	printf() 函数的返回值	139
6.1.7	理解输出列表	140
6.1.8	printf 格式化输出函数综合实例	142
6.2	格式化输入函数 scanf()	143
6.2.1	scanf() 函数的格式	143
6.2.2	scanf() 函数格式字符串	145
6.2.3	scanf() 函数注意问题	146
6.2.4	scanf() 函数的返回值	148
6.2.5	格式化输入函数 scanf() 综合实例	148
6.3	其他常用的输入/输出函数	149
6.3.1	字符读入函数 getchar()	149
6.3.2	字符读入函数 getch()	150
6.3.3	字符串读入函数 gets()	151
6.3.4	字符输出函数 putchar()	152
6.3.5	字符输出函数 putchar()	153
6.3.6	字符串输出函数 puts()	153
6.3.7	格式化内存缓冲区输出函数 sprintf()	154
6.3.8	格式化内存缓冲区输入函数 sscanf()	155
6.4	小结	156

第7章 数组

157

在前面章节中介绍了 C 语言的基本数据类型,包括整型、字符型和实型。以这些基本数据类型为基础, C 语言还提供几种构造数据类型。构造类型数据是由基本类型数据按一定规则组成的,数组就是其中一种构造数据类型。在程序中用循环结构可以很方便地处理数组。本章将详细介绍 C 语言中数组的使用。

7.1 了解数组	157
7.1.1 使用数组的好处	157
7.1.2 数组的概念	158
7.1.3 数组的维数	160
7.2 一维数组	161
7.2.1 一维数组的声明	161
7.2.2 一维数组的存储	164
7.2.3 一维数组的引用	164
7.2.4 向函数传递一维数组	166
7.2.5 一维数组的初始化	166
7.3 二维数组	169
7.3.1 二维数组的声明	169
7.3.2 二维数组的存储	170
7.3.3 二维数组的引用	171
7.3.4 二维数组的初始化	173
7.4 字符数组和字符串	175
7.4.1 一维字符串数组	176
7.4.2 了解字符串	177
7.4.3 字符串的操作函数	178
7.4.4 字符串的输入输出	179
7.4.5 二维字符串数组	181
7.5 多维数组	182
7.6 数组的基本应用	182
7.6.1 反转字符串	182
7.6.2 查找最大值	183
7.6.3 数据排序	185
7.7 小结	186

第8章 函数

187

使用 C 语言编程,就肯定会使用到函数。C 语言允许用户使用一些编译环境自带的库函数,例如使用 `printf()` 函数输出变量的值,使用 `scanf()` 函数接收用户的输入等。合理使用库函数可以大大简化程序设计过程。同时,为了有利于程序的模块化,促进程序资源的共享。C 语言也支持用户使用自定义函数。本章将详细介绍函数的概念、调用、作用域等相关知识。

8.1 函数的概念	187
8.1.1 什么是函数	187
8.1.2 函数的分类	188
8.1.3 函数的定义	189
8.1.4 <code>main()</code> 函数	191
8.2 函数的工作过程	192

8.2.1	程序结构	192
8.2.2	函数执行过程	193
8.3	编写函数	193
8.3.1	函数头	193
8.3.2	返回类型	194
8.3.3	参数列表	195
8.3.4	函数体	195
8.3.5	函数原型	196
8.4	函数的参数	197
8.4.1	形参和实参	197
8.4.2	参数传递过程	198
8.4.3	值调用	198
8.4.4	引用调用	199
8.4.5	数组作为函数参数	201
8.4.6	指针作为函数参数	205
8.4.7	main()函数的参数	206
8.5	函数调用	208
8.5.1	函数调用方式	208
8.5.2	被调函数的说明	209
8.5.3	函数的返回值	210
8.6	几种典型的函数调用形式	212
8.6.1	赋值调用与引用调用	212
8.6.2	嵌套调用	213
8.6.3	递归调用	215
8.7	函数及其变量的作用域	221
8.7.1	函数的作用域	221
8.7.2	函数的变量作用域	221
8.8	小结	222

第9章 指针

223

指针是C语言中广泛使用的一种数据类型。利用指针可以操作各种基本的数据类型,以及数组等复合数据结构,甚至使用指针还可以访问函数。正确理解和使用指针对于成功进行C语言程序设计是至关重要的。

指针是C语言最显著的特征,同时又是C语言最危险的特征。例如,在使用指针的程序中,常常因为用错指针导致程序出错,而这类错误却很难发现。更严重的是,对未初始化的指针进行操作可能会导致系统崩溃。

9.1	内存和变量	223
9.1.1	计算机内存	223
9.1.2	变量的存储	224
9.2	指针和简单变量	225
9.2.1	指针的概念	225
9.2.2	创建指针	226
9.2.3	初始化指针变量	227
9.2.4	指针变量的引用	228

9.2.5	给函数传递指针	232
9.3	指针变量的赋值	232
9.3.1	初始化赋值	232
9.3.2	取地址赋值	233
9.3.3	指针之间赋值	233
9.3.4	数组赋值	233
9.3.5	字符串赋值	234
9.3.6	函数入口赋值	234
9.4	指针和数组的关系	234
9.4.1	指针、数组和地址间的关系	234
9.4.2	指针变量的运算	236
9.4.3	用指针操作数组元素	237
9.5	指向多维数组的指针	240
9.5.1	理解二维数组的地址	240
9.5.2	多维数组的指针表示	242
9.5.3	指向多维数组的指针变量	243
9.5.4	数组名作为函数的参数	247
9.5.5	指向数组的指针小结	250
9.6	指针和字符串	251
9.6.1	字符串的指针表示	251
9.6.2	字符串指针作函数参数	253
9.6.3	字符数组和字符指针的区别	254
9.7	指针数组	256
9.7.1	指针数组的概念	256
9.7.2	用指针数组处理字符串	258
9.7.3	指针数组作函数参数	259
9.8	指向指针的指针	261
9.8.1	理解指向指针的指针	261
9.8.2	二级指针变量与数组	263
9.9	指针和函数	268
9.9.1	返回指针的函数	268
9.9.2	指向函数的指针	269
9.10	指针和 const 变量	272
9.10.1	用 const 控制指针	273
9.10.2	const 的几种特殊用法	274
9.11	指针总结	274
9.11.1	明确分辨各种指针类型	275
9.11.2	正确理解指针	275
9.12	小结	276

第 10 章 结构

277

在实际的程序设计中，经常需要处理具有不同数据类型的一组数据。例如，在学生成绩统计表中，应该包含如下几项：姓名（字符型）、学号（整型或字符型）、年龄（整型）、性别（字符型）和成绩（整型或浮点型）。由于数组中各元素的类型和长度都必须一致，因此不能用一个数

组存放这一组数据。在这种情况下，C 语言中可以使用“结构”，这是一种构造数据类型，相当于其他高级语言中的记录类型。

10.1	结构的定义	277
10.2	结构变量的定义	278
10.2.1	先定义结构，再定义结构变量	279
10.2.2	在定义结构的同时，定义结构变量	279
10.2.3	直接说明结构变量	279
10.3	结构变量的使用	280
10.4	结构变量的初始化	281
10.5	结构数组	283
10.5.1	结构数组的定义和引用	283
10.5.2	结构数组的初始化	285
10.5.3	结构数组实例	285
10.6	结构指针	286
10.6.1	定义结构指针	286
10.6.2	结构指针的引用	287
10.6.3	用指针处理结构数组	289
10.7	嵌套结构	291
10.7.1	包含数组的结构	291
10.7.2	包含指针的结构	292
10.7.3	包含结构的结构	292
10.8	向函数传递结构	296
10.8.1	传递结构变量的值	296
10.8.2	传递结构指针到函数	297
10.9	小结	298

第 11 章 联合、枚举和位域

299

在 C 语言中，除了数组和指针外，还定义了联合和枚举两种聚合数据类型，并且可以进行类型说明和定义位域，这些是特殊形式的数据类型。下面分别进行讲解。

11.1	联合	299
11.1.1	定义联合类型	299
11.1.2	定义联合变量	300
11.1.3	联合变量成员的引用	301
11.1.4	联合变量数组	301
11.1.5	联合变量指针	302
11.1.6	在结构中嵌套联合类型	303
11.1.7	结构和联合的区别	305
11.2	枚举	306
11.2.1	定义枚举类型	306
11.2.2	定义枚举变量	307
11.2.3	枚举类型变量的赋值	308
11.3	类型说明	310
11.4	二进制数据存储	311

11.4.1	数据的存储	312
11.4.2	不同数据类型的二进制表示	312
11.5	位运算	315
11.5.1	位逻辑运算符	315
11.5.2	移位运算符	318
11.5.3	位运算的复合赋值运算符	318
11.5.4	位运算的用途	319
11.6	位域	320
11.6.1	定义位域结构	320
11.6.2	位域变量的声明	321
11.6.3	位域的使用	322
11.6.4	位域与联合的使用	323
11.7	小结	324

第 12 章 预处理

325

C 语言的程序中可包括各种以符号#开头的编译指令,这些指令称为预处理命令。预处理命令属于 C 语言编译器,而不是 C 语言的组成部分。通过预处理命令可扩展 C 语言程序设计的环境。

预处理命令通常在程序编译时进行一些符号处理,其并不执行具体的硬件操作。C 语言中的预处理命令主要有宏定义指令、文件包含指令和条件编译指令,还有其他一些调试时使用的指令。本章将介绍预处理程序各命令的使用方法。

12.1	预处理简介	325
12.1.1	预处理功能	325
12.1.2	预处理命令	325
12.2	宏定义命令	326
12.2.1	无参数的宏定义	326
12.2.2	带参数的宏定义	329
12.2.3	#undef 命令	331
12.2.4	预处理操作符#和##	331
12.3	文件包含指令	334
12.4	条件编译指令	335
12.4.1	#if、#else、#endif 命令	336
12.4.2	#elif 命令	337
12.4.3	#ifdef、#ifndef 命令	339
12.4.4	使用#define 和#undef	340
12.5	其他预处理命令	340
12.5.1	预定义的宏名	340
12.5.2	重置行号和文件名命令#line	341
12.5.3	修改编译器设置命令#pragma	342
12.5.4	产生错误信息命令#error	342
12.6	内联函数	343
12.7	小结	344

第 13 章 存储管理

345

在大多数情况下,存储管理都是操作系统的功能。一般的程序设计语言不提供对内存的管理。

而 C 语言提供了管理内存的相关函数,可实现内存的动态存储管理。这是 C 语言功能强大的一个表现,本章将介绍 C 语言的内存组织方式和动态存储管理。

13.1 内存组织方式	345
13.1.1 内存组织方式	345
13.1.2 堆和栈的比较	346
13.2 C 语言的动态存储管理	347
13.3 分配内存函数——malloc()	347
13.4 带初始化的分配内存函数——calloc()	349
13.5 调整已分配内存函数——realloc()	350
13.6 释放分配的内存函数——free()	352
13.7 丢失的内存	354
13.8 小结	356

第 2 篇 C 语言进阶篇

第 14 章 字符及字符串操作

358

在实际的程序设计中,经常需要处理文本数据。在 C 语言中,文本数据按字符串格式保存。对于字符串,前面已经介绍过如何用字符数组保存字符串、用字符指针指向一个字符串常量、字符串的输入与输出函数等。

在 C 语言的库函数中提供了丰富的处理字符及字符串的函数,本章将介绍这些函数的用法。

14.1 字符的操作	358
14.1.1 检查英文字母函数	359
14.1.2 检查字母数字函数	360
14.1.3 检查控制字符函数	360
14.1.4 十进制数字检查函数	361
14.1.5 可打印字符检查函数	362
14.1.6 包含空格的可打印字符检查函数	363
14.1.7 格式字符检查函数	364
14.1.8 小写英文字母检查函数	365
14.1.9 大写英文字母检查函数	366
14.1.10 控制字符检查函数	367
14.1.11 十六进制数字检查函数	368
14.1.12 大写字符转换函数	368
14.1.13 小写字符转换函数	369
14.1.14 ASCII 字符转换函数	370
14.1.15 大写字符宏转换函数	371
14.1.16 小写字符宏转换函数	371
14.2 字符串的存储	372
14.2.1 字符串的静态存储	372
14.2.2 字符串的动态存储	373
14.3 字符串的操作	374
14.4 检测字符串长度函数	375
14.5 查找字符串	376