

<<双语版C程序设计>>

图书基本信息

书名：<<双语版C程序设计>>

13位ISBN编号：9787121190438

10位ISBN编号：7121190435

出版时间：2013-3

出版时间：电子工业出版社

版权说明：本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问：<http://www.tushu007.com>

<<双语版C程序设计>>

内容概要

<<双语版C程序设计>>

作者简介

作者: (爱尔兰) Paul Kelly (P. 凯利)

## 书籍目录

Chapter One Introduction to C (引言) 1.1 Brief history of C (C语言简史) 1.2 Why programmers use C (为什么程序员爱用C语言) 1.2.1 C is portable 1.2.2 C is a structured programming language 1.2.3 C is efficient 1.2.4 C is flexible 1.2.5 C is powerful 1.2.6 C is concise 1.3 Developing a C program (开发C程序) 1.4 Suggestions for learning C Programming (学习C语言程序设计的建议) Chapter Two C Data Types (C数据类型) 2.1 Constants (常量) 2.2 Variables (变量) 2.3 Simple output to the screen (简单的屏幕输出) 2.4 Comments (注释) 2.5 Data types (数据类型) 2.5.1 Short integer data types 2.5.2 Long integer data types 2.5.3 Unsigned integer data types 2.5.4 Double floating—point data type 2.6 Data type sizes (数据类型的大小) Programming pitfalls Quick syntax reference Exercises Chapter Three Simple Arithmetic Operations and Expressions (简单的算术运算和表达式) 3.1 C operators (C运算符) 3.1.1 The assignment operator 3.1.2 Arithmetic operators 3.1.3 Increment and decrement operators 3.1.4 Combined operators 3.2 Operator precedence (运算符优先级) 3.3 Type conversions and casts (类型转换与强制类型转换) Programming pitfalls Quick syntax reference Exercises Chapter Four Keyboard Input and Screen Output (键盘输入和屏幕输出) 4.1 Simple keyboard input (简单的键盘输入) 4.2 Using a width and precision specification in printf ( ) [在函数printf( )中使用域宽和精度说明] 4.3 Single—character input and output (单个字符的输入和输出) Programming pitfalls Quick syntax reference Exercises Chapter Five Control Statements: If and Switch (控制语句: if和switch) 5.1 The if statement (if语句) 5.2 The if—else statement (if—else语句) 5.3 Logical operators (逻辑运算符) 5.4 Nested if statements (嵌套的if语句) 5.5 The switch statement (switch语句) 5.6 The conditional operator ? : (条件运算符) Programming pitfalls Quick syntax reference Exercises Chapter Six Iterative Control Statements: while, do—while, and for (循环控制语句: while、do—while和for) 6.1 The while statement (while语句) 6.2 The do—while loop (do—while循环) 6.3 The for statement (for语句) 6.4 Nested loops (嵌套的循环) Programming pitfalls Quick syntax reference Exercises Chapter Seven Arrays (数组) 7.1 Introduction to arrays (引言) 7.2 Initialising arrays (数组初始化) 7.3 Two—dimensional arrays (二维数组) 7.4 Initialising two—dimensional arrays (二维数组的初始化) 7.5 Multi—dimensional arrays (多维数组) Programming pitfalls Quick syntax reference Exercises Chapter Eight Pointers (指针) 8.1 Variable addresses (变量的地址) 8.2 Pointer variables (指针变量) 8.3 The dereference operator \* (解引用运算符\*) 8.4 Why use pointers? (为什么使用指针) Programming pitfalls Quick syntax reference Exercises Chapter Nine Pointers and Arrays (指针和数组) 9.1 Pointers and one—dimensional arrays (指针和一维数组) 9.2 Pointers and multi—dimensional arrays (指针和多维数组) 9.3 Dynamic memory allocation (动态内存分配) 9.3.1 The malloc function 9.3.2 The calloc function 9.3.3 The realloc function 9.3.4 Allocating memory for multi—dimensional arrays Programming pitfalls Exercises Chapter Ten Strings (字符串) 10.1 String literals (字符串) 10.2 Long character strings (长字符串) 10.3 Strings and arrays (字符串和数组) 10.4 Displaying a string (显示一个字符串) 10.5 The puts function [puts函数] 10.6 The gets function [gets函数] 10.7 Accessing individual characters of a string (访问字符串中的单个字符) 10.8 Assigning a string to a pointer (用字符串为字符指针赋值) 10.9 String functions (字符串处理函数) 10.9.1 Finding the length of a string 10.9.2 Copying a string 10.9.3 String concatenation 10.9.4 Comparing strings 10.9.5 Other string functions 10.10 Converting numeric strings to numbers (数值字符串向数值的转换) 10.11 Arrays of strings (字符串数组) Programming pitfalls Quick syntax reference Exercises Chapter Eleven Functions (函数) 11.1 Introduction (引言) ..... Chapter Twelve Structures (结构体) Chapter Thirteen File Input and Output (文件的输入和输出) Chapter Fourteen The C Preprocessor (C编译预处理) Appendix A List of C Keywords Appendix B Precedence and Associativity of C Operators Appendix C ASCII Character Codes Appendix D Fundamental C Built—in Data Types

## &lt;&lt;双语版C程序设计&gt;&gt;

## 章节摘录

版权页：但是，由于CPU中寄存器的数量是有限的，因此编译器不能保证被定义为寄存器存储类型的变量一定会存储到CPU的寄存器中。

如果寄存器变量未存储在CPU的寄存器中，那么这个变量将变为默认的自动存储类型的变量。

注意，许多编译器为了优化程序的执行速度，会将一些像循环索引变量这样的变量自动存储在CPU寄存器中。

因此，多数情况下的循环索引变量可能自动指定为寄存器变量，而不必人为指定。

很多程序可以通过指定程序名和紧随其后的要传给程序的一个数据列表来执行。

例如在Linux操作系统中，我们可以使用cp命令来实现文件的复制。

old—file和new\_file称为命令行参数。

操作系统通过向程序传递两个信息来告知C程序命令行中有哪些内容，第一个信息是命令行参数的个数，第二个信息是存储命令行参数的数组的首地址。

请注意main（）函数是如何接收操作系统传来的参数值的：操作系统将main（）当做一个普通函数，像其他函数一样，操作系统传给main（）的数据值存储在argc（参数个数argument count的缩写）

和argv（参数向量argumentvector的缩写）这两个形参中，这两个形参名也可以是其他任何合法的变量名，但是，习惯上将它们命名为argc和argv。

程序的第4行，将argc定义为整型变量，将argv定义为字符指针数组，数组argv的每个元素中存储着各个命令行参数的首地址。

<<双语版C程序设计>>

编辑推荐

<<双语版C程序设计>>

版权说明

本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问:<http://www.tushu007.com>