

<<标准C程序设计>>

图书基本信息

书名：<<标准C程序设计>>

13位ISBN编号：9787302197959

10位ISBN编号：7302197954

出版时间：2009-5

出版时间：(印)巴拉古路萨米 清华大学出版社 (2009-05出版)

作者：[印] 巴拉古路萨米

页数：543

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>

<<标准C程序设计>>

前言

About the Author: E. Balagurusamy, former Vice Chancellor, Anna University, Chennai, is currently Member, Union Public Service Commission, New Delhi. He is a teacher, trainer, and consultant in the fields of information Technology and Management. He holds an M.E. (Hons) in Electrical Engineering and Management. He holds an M.E. (Hons) in Electrical Engineering and Ph.D. in Systems Engineering from the Indian Institute of Technology, Roorkee. His areas of interest include Object-Oriented Software Engineering, Electronic-Oriented Software Engineering, Electronic Business, Technology Management, Business Process Re-engineering, and Total Quality Management. A prolific writer, he has authored a large number of research papers and several books. His best selling books, among others, include: Programming in C#. Programming in Java, 3/e. Object-Oriented Programming with C++, 3/e. Electronic-Oriented Programming with C++, 3/e. Programming in BASIC, 3/e. Numerical Methods. Reliability Engineering. A recipient of numerous honours and awards, he has been listed in the Directory of Who's Who of intellectuals and in the Directory of Distinguished Leaders in Education.

Preface to the Fourth Edition: C is a powerful, flexible, portable and elegantly structured programming language. Since C combines the features of high-level language with the elements of the assembler, it is suitable for both systems and applications programming. It is undoubtedly the most widely used general-purpose language today. Since its standardization in 1989, C has undergone a series of changes and improvements in order to enhance the usefulness of the language. The version that incorporates the new or porates the new features is now referred to as C99. The fourth edition of ANSI C has been thoroughly revised and enlarged not only to incorporate the numerous suggestions received both from teachers and students across the country but also to highlight the enhancements and new features added by C99.

Organization of the book: The book starts with an overview of C, which talks about the history of C, basic structure of C programs and their execution. The second chapter discusses how to declare the constants, variables and data types. The third chapter describes the built-in operators and how to build expressions using them. The fourth chapter details the input and output operations. Decision making and branching is discussed in the fifth chapter, which talks about the if-else, switch and goto statements. Further, decision making and looping is discussed in Chapter six, which covers while, do and for loops. Arrays and ordered arrangement of data elements, which covers while, do and for loops. Arrays and ordered arrangement of data elements are important to any programming language and have been covered in chapters seven and eight. Strings are also covered in Chapter eight. Chapters nine and ten are on functions, structures and unions. Pointers, perhaps the most difficult part of C to understand, is covered in Chapter eleven in the most user-friendly manner. Chapters twelve and thirteen are on file management and dynamic memory allocation respectively. Chapter fourteen deals with the preprocessor, and finally Chapter 15 is on developing a C program, which provides an insight on how to proceed with development of a program. The above organization would help the students in understanding C better if followed appropriately.

New to the edition: The content has been revised keeping the updates which have taken place in the field of C programming and the present day syllabus needs. As always, the concept of 'learning by example' has been stressed throughout the book. Each major feature of the language is treated in depth followed by a complete program example to illustrate its use. The sample programs are meant to be both simple and educational. Two new projects are added at the end of the book for students to go through and try on their own. Each chapter includes a section at the beginning to introduce the topic in a proper perspective. It also provides a quick look into the features that are discussed in the chapter. Wherever necessary, pictorial descriptions of concepts are included to improve clarity and to facilitate better understanding. Language tips and other special considerations are highlighted as notes wherever essential. In order to make the book more user-friendly, we have incorporated the following key features: Codes with comments are provided throughout the book to illustrate how the various features of the

<<标准C程序设计>>

language are put together to accomplish specified tasks. Supplementry imp~rtion abed notes that complement but stand apart from the general text have been included in boxes. Guidelines for developing efficient C programs are given in the last chapter, together with a list of some common mistakes that a less experienced C programmer could make. Cause studies at the end of the chapters illustrate common ways C features are put together and also show real-life applications. The Joest Remember section at the end of the chapters lists out helpful hints and possible problem areas. Numerous chapter-end questions and exercises provide ample opportunities to the readers to review the concepts learned and to practice their applications. Progrrrmming projects discussed in the appendix give insight on how to integrate the various features of C when handling large programs. Supplementary Mntertul With this revision we have tried to enhance the online learning center too. The supplementary material would include the following: For the instructor Q Solutions to the debugging exercises For the Student Q Exclusive project for implementation with code, step-by-step description and user manual Q Code for the two projects (given in the book) U Two mini projects Q Reading material on C This book is designed for all those who wish to be C programmers, regardless of their past knowledge and experience in programming. It explains in a simple and easy-to-understand style the what, why and how of programming with ANSI C.

<<标准C程序设计>>

内容概要

本书旨在教授读者如何使用C语言进行程序设计。

全书贯彻了“用示例学习”的概念。

在深入介绍了C语言的每个特性之后，给出了一个完整的示例程序，用于演示说明其应用。

每章末尾的“案例学习”不仅介绍了把C语言的特性集成在一起的常用方法，而且还显示了它在实际生活中的应用。

最后一章介绍了开发高效、无错误的C程序的一些指导原则。

本书在前一版的基础上增加介绍了C99的一些新特性，并增加了不少复习题和项目设计题，而且在附录IV中给出了4个完整的应用程序开发示例，使得本书的实际应用性更强。

<<标准C程序设计>>

作者简介

作者：巴拉古路萨米

书籍目录

第1章 C语言概述第2章 常量、变量及数据类型第3章 运算符与表达式第4章 输入输出操作管理第5章 判断与分支第6章 判断与循环第7章 数组第8章 字符数组与字符串 第9章 自定义函数第10章 结构与联合第11章 指针第12章 文件管理第13章 动态内存分配与链表第14章 预处理器第15章 C程序开发指导原则附录I 位级程序设计附录II附录III附录IV 项目设计附录V C99的特性Preface to the Fourth Edition xilace to the Fourth Edition xi1 Overview of C 11.1 History of C 11.2 Importance of C 31.3 Sample Program 1: Printing a Message 31.4 Sample Program 2: Adding Two Numbers 61.5 Sample Program 3: Interest Calculation 81.6 Sample Program 4: Use of Subroutines 101.7 Sample Program 5: Use of Math Functions 111.8 Basic Structure of C Programs 121.9 Programming Style 141.10 Executing a 'C' Program 141.11 Unix System 161.12 Ms-Dos System 18Review Questions 19Programming Exercises 202 Constants, Variables, and Data Types 232.1 Introduction 232.2 Character Set 232.3 C Tokens 252.4 KeyWords and identifiers 25y words and Identiners 252.5 Constants 262.6 Variables 302.7 Data Types 31~'2.8 Declaration of Variables 342.9 Declaration of Storage Class 372.10 Assigning Values to Variables 382.11 Defining Symbolic Constants 442.12 Declaring a Variable as Constant 452.13 Declaring a Variable as Volatile 45Vi Contents2.14 Overflow and Underflow of Data 46Review Questions 49Programming Exercises 513 Operators and Expressions 523.1 Introduction 523.2 Arithmetic Operators 523.3 Relational Operators 553.4 Logical Operators 573.5 Assignment Operators 573.6 Increment and Decrement Operators 593.7 Conditional Operator 613.8 Bitwise Operators 613.9 Special Operators 613.10 Arithmetic Expressions 633.11 Evaluation of Expressions 643.12 Precedence of Arithmetic Operators 653.13 Some Computational Problems 673.14 Type Conversions in Expressions 683.15 Operator Precedence and Associativity 723.16 Mathematical Functions 74Review Questions 78Programming Exercises 814 Managing input and Output Operations 844.1 Introduction 844.2 Reading a Character 854.3 Writing a Character 884.4 Formatted input 894.5 Formatted Output 98Review Questions 110Programming Exercises 1125 Decision Making and Branching 1145.1 Introduction 1145.2 Decision Making with iF Statement 1145.3 Simple iF Statement 1155.4 The IF.....ELSE Statement 1195.5 Nesting of IF....ELSE Statements 1225.6 The ELSE IF Ladder 1265.7 The Switch Statement 1295.8 The '?' Operator 1335.9 The GOTO Statement 136Review Questions 144Programming Exercises 148Contents Vii6 Decision Making and Looping 1526.1 Introduction 1526.2 The WHILE Statement 1546.3 The DO Statement 1576.4 The FOR Statement 1596.5 Jumps in LOOPS 1666.6 Concise Test Expressions 174Review Questions 182Programming Exercises 1867 Arrays 1907.1 Introduction 1907.2 One-dimensional Arrays 1927.3 Declaration of One-dimensional Arrays 1937.4 Initialization of One-dimensional Arrays 1957.5 Two-dimensional Arrays 1997.6 Initializing Two-dimensional Arrays 2047.7 Multi-dimensional Arrays 2087.8 Dynamic Arrays 2097.9 More about Arrays 209Review Questions 223Programming Exercises 2258 Character Arrays and Strings 2298.1 Introduction 2298.2 Declaring and initializing String Variables 2308.3 Reading Strings from Terminal 2318.4 Writing Strings to Screen 2368.5 Arithmetic Operations on Characters 2418.6 Putting Strings Together 2428.7 Comparison of Two Strings 2448.8 String-handling Functions 2448.9 Table of Strings 2508.10 Other Features of Strings 252Review Questions 257Programming Exercises 2599 User-defined Functions 2629.1 Introduction 2629.2 Need for User-defined Functions 2629.3 A Multi-function Program 2639.4 Elements of User-defined Functions 2669.5 Definition of Functions 2679.6 Return Values and their Types 2699.7 Function Calls 2709.8 Function Declaration 272Vin Contents9.9 Category of Functions 2749.10 No Arguments and no Return Values 2749.11 Arguments but no Return Values 2779.12 Arguments with Return Values 2809.13 No Arguments but Returns a Value 2849.14 Functions that Return Multiple Values 2859.15 Nesting of Functions 2869.16 Recursion 2889.17 Passing Arrays to Functions 2899.18 Passing Strings to Functions 2949.19 The Scope, Visibility and Lifetime of Variables 2959.20 Multifile Programs 305Review Questions 311Programming Exercises 31510 Structures and Unions 31710.1 Introduction 31710.2 Defining a Structure 31710.3 Declaring Structure Variables 31910.4 Accessing Structure Members 32110.5 Structure initialization 32210.6 Copying and Comparing Structure Variables 32410.7 Operations on individual Members 32610.8 Arrays of Structures 32710.9 Arrays within Structures 32910.10 Structures within Structures 33110.11 Structures and Functions 33310.12 Unions 33510.13 Size of Structures 33710.14 Bit Fields

337Review Questions 344Programming Exercises 34811 Pointers 35111.1 Introduction 35111.2 Understanding Pointers 35111.3 Accessing the Address of a Variable 35411.4 Declaring Pointer Variables 35511.5 Initialization of Pointer Variables 35611.6 Accessing a Variable through its Pointer 35811.7 Chain of Pointers 36011.8 Pointer Expressions 36111.9 Pointer increments and Scale Factor 36211.10 Pointers and Arrays 36411.11 Pointers and Character Strings 36711.12 Array of Pointers 369Contents ix11.13 Pointers as Function Arguments 37011.14 Functions Returning Pointers 37311.15 Pointers to Functions 37311.16 Pointers and Structures 37611.17 Troubles with Pointers 379Review Questions 385Programming Exercises 38812 File Management in C 38912.1 Introduction 38912.2 Defining and Opening a File 39012.3 Closing a File 39112.4 Input/Output Operations on Files 39212.5 Error Handling During I/O Operations 39812.6 Random Access to Files 40012.7 Command Line Arguments 405Review Questions 408Programming Exercises 40913 Dynamic Memory Allocation and Linked Lists 41113.1 Introduction 41113.2 Dynamic Memory Allocation 41113.3 Allocating a Block of Memory: MALLOC 41313.4 Allocating Multiple Blocks of Memory: CALLOC 41513.5 Releasing the Used Space: Free 41513.6 Altering the Size of a Block' REALLOC 41613.7 Concepts of Linked Lists 41713.8 Advantages of Linked Lists 42013.9 Types of Linked Lists 42113.10 Pointers Revisited 42213.11 Creating a Linked List 42413.12 Inserting an item 42813.13 Deleting an item 43113.14 Application of Linked Lists 433Review Questions 440Programming Exercises 44214 The Preprocessor 44414.1 Introduction 44414.2 Macro Substitution 44514.3 File inclusion 44914.4 Compiler Control Directives 45014.5 ANSI Additions 453Review Questions 456Programming Exercises 457X Contents15 Developing a C Program: Some Guidelines 45815.1 Introduction 45815.2 Program Design 45815.3 Program Coding 46015.4 Common Programming Errors 46215.5 Program Testing and Debugging 46915.6 Program Efficiency 471Review Questions 472Appendix i: Bit-level Programming 474Appendix II: ASCII Values of Characters 480Appendix III: ANSI C Library Functions 482Appendix IV: Projects 486Appendix V: C99 Features 537Bibliography 545

章节摘录

插图：Use the size of operator to determine the size of a linked list. When using memory allocation functions malloc and calloc , test for a NULL pointer return value. Print appropriate message if the memory allocation fails. Never call memory allocation functions with a zero size. Release the dynamically allocated memory when it is no longer required to avoid any possible “ memory leak”. Using free function to release the memory not allocated dynamically with malloc or calloc is an error. Use of an invalid pointer with free may cause problems and , sometimes , system crash. Using a pointer after its memory has been released is an error. It is an error to assign the return value from malloc or calloc to anything other than a pointer. It is a logic error to set a pointer to NULL before the node has been released. The node is irretrievably lost. It is an error to declare a self-referential structure without a structure tag. It is an error to release individually the elements of an array created with Calloc. It is a logic error to fail to set the link field in the last node to null.

1. Insertion in a Sorted List

The task of inserting a value into the current location in a sorted linked list involves two operations : 1. Finding the node before which the new node has to be inserted . We call this node as ‘ Key node ’ .2. Creating a new node with the value to be inserted and inserting the new node by manipulating pointers appropriately. In order to illustrate the process of insertion , we use a sorted linked list created by the I create function discussed in Exemple 13.3. Figure 13.11 shows a complete program that I creates a list (using sorted input data) and then inserts a given value into the correct place using function insert.

<<标准C程序设计>>

编辑推荐

《标准C程序设计》第3版融入了过去10年中采用本书的学生和老师的很多反馈意见。

本版的特点包括：| 基于最新的C语言标准。

| 在本书的最后给出了4个完整的应用程序开发示例。

| 扩展讨论了C的指针。

| 每章后面的“谨记”一节给出了很有用的编程提示以及可能容易出错的问题。

| 20多个真实的开发案例，展示了C程序的设计过程。

| 80多个程序设计范例，向读者阐述了良好程序设计的基本原则。

| 还有200多个复习题和150多个项目设计题。

总之，本书的语言简洁易懂，示例非常丰富且具有很强的实际指导意义，是一本很好的C语言程序设计的教材。

<<标准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>