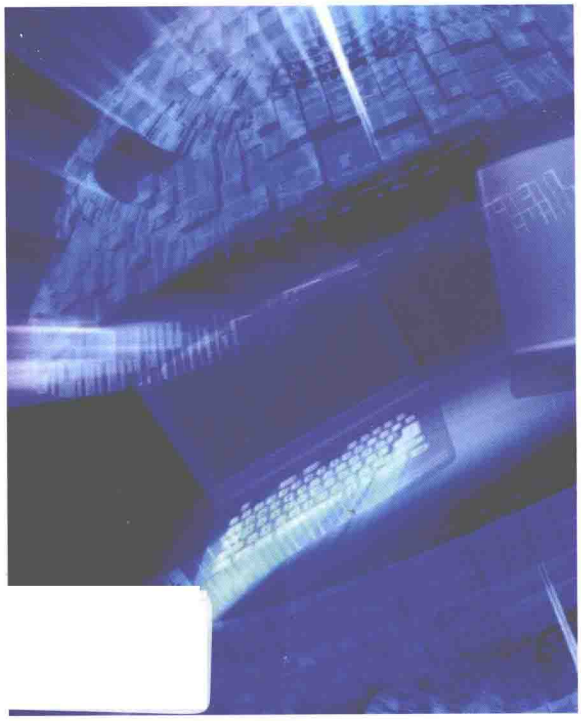
高等学校计算机应用规划教材

C语言程序设计

- 程序设计概述
- C语言概述
- 数据类型、运算符和表达式
- 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、函数和指针
- 编译预处理指令
- 结构体、共用体与枚举类型
- 文件操作
- C语言的扩展简述



刘韶涛　潘秀霞　应晖　编著

清华大学出版社

高等学校计算机应用规划教材

C语言程序设计

刘韶涛 潘秀霞 应 晖 编著

清华大学出版社
北 京

内 容 简 介

本书是根据近年来实际教学过程中，学生学习 C 语言程序设计遇到的各种问题和反馈意见，进行了总结讨论和分析提炼，纠正、修改和进一步完善了之前使用的教材的基本内容，增加了扩展 C 程序设计的相关新章节。

本书力求对 C 语言程序设计中涉及的基本概念、基本理论、典型应用和语法规则等的表述更为规范、科学和准确，文字叙述更加精炼通顺、实验数据更为准确有据。并对本书的全部习题和案例程序等都给出了完整的注释、运行结果分析和解题说明等。

在本书中，不仅仅局限于对 C 语言程序设计知识的描述，也把与 C 程序设计相关的其他知识加以阐述，特别介绍 C 语言在其他交叉学科和相关领域中的新应用，让读者对 C 程序设计在整个学科体系、不同的软件开发环境、工程实践背景等都有一个较清楚的了解和认识。

本书既可作为高等学校 C 语言程序设计课程的教材，也可作为 C 语言程序开发人员的参考书。

图书在版编目(CIP)数据

C 语言程序设计 / 刘韶涛，潘秀霞，应晖　编著. —北京：清华大学出版社，2015
(高等学校计算机应用规划教材)
ISBN 978-7-302-38899-9

Ⅰ. ①C…　Ⅱ. ①刘…　②潘…　③应…　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(2014)第 308047 号

责任编辑：王　定
封面设计：牛艳敏
版式设计：思创景点
责任校对：邱晓玉
责任印制：李红英

出版发行：清华大学出版社
网　　址：http://www.tup.com.cn，http://www.wqbook.com
地　　址：北京清华大学学研大厦 A 座　　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
课 件 下 载：http://www.tup.com.cn,010-62794504
印 装 者：清华大学印刷厂
经　　销：全国新华书店
开　　本：185mm×260mm　　**印　　张**：23.75　　**字　　数**：623 千字
版　　次：2015 年 2 月第 1 版　　**印　　次**：2015 年 2 月第1 次印刷
印　　数：1～3500
定　　价：45.00 元

产品编号：057581-01

前　言

目前，国内外C语言程序设计的相关教材较多，但是大多数教材都着重于对C语言基本语法规则和基本概念的阐述，学生学完之后，并不能真正掌握和灵活使用C来解决一些实际应用问题。特别是国内的很多教材，是为了让学生学完后，参加全国或者省计算机等级考试而编写的，其内容完全是为了对付等级考试的考试内容。再加上目前计算机等级考试中的C语言程序设计的考试内容、考试方法等都存在很多不完善的地方，考试内容和考试成绩并不能反映考生真正运用C语言进行程序设计来解决实际问题的能力和水平。因此，编写高质量的C程序设计教材和辅导学习丛书，培养学生思考问题、分析问题和解决问题的能力和水平，提高其计算机的应用能力水平，对学生来说既是非常必要，也是非常重要的。

本书根据初学者的特点，由浅入深，循序渐进，旨在帮助学生掌握C语言程序设计的基本方法，理解领会C语言的特点和本质，提高学生运用C解决实际问题的综合能力。并进一步丰富增加了一些典型的应用案例和模拟试题分析等，对基本概念和规则的表述更为科学、文字更为精炼通顺、数据更为准确有据。案例程序都给出完整的注释、运行结果和分析说明，所有习题和上机实践题也都增加了参考答案或分析和解答提示等，以利于学生自我解题时进一步参考和对比。对考试模拟试卷的分析，力求更为详尽，重点突出，让学生明白解决问题的思路和方法，起到举一反三的作用，更方便学生进行自我练习、检查以及理解和掌握，尽量做到一题多解，着重对学生分析及思考能力的培养和训练。

我们根据近年来实际教学过程中，学生使用初版教材遇到的各种问题和反馈意见，组织编著教师和授课教师，总结讨论，分析提炼，经过细心筛选和整理，重新编著了《C语言程序设计》和《C语言程序设计学习指导与上机实践》，纠正、修改和进一步完善之前版本中的内容，增加和学科发展及知识更新相关的新章节。另外，在本套书中，我们信奉“不求全面，但求实用”的理念。尽量让读者都能寻找到各种知识点的方向和途径，指引出什么问题应该从哪些地方寻找答案。不仅仅局限于C语言程序设计知识的描述，也把与程序设计相关的其他知识加以阐述，特别介绍C语言在其他交叉学科和相关领域中的新应用，让读者对C程序设计在整个学科体系、不同的软件开发环境、工程实践背景等都有一个较清楚的了解和认识。

本教材的编写，我们所追求的目标是：

(1) 既能作为一本学习C语言程序设计的学习教材，又能作为一本C语言程序设计的实验指导教材，也可作为一本探讨C程序设计学习和实践的艺术书籍。

(2) 突出C的应用重点和难点，不拘于具体语法细节的学习指导，而更注重C的应用实践环节的上机实训。紧密联系教学实践，在教材中力求反映出学生学习相关知识的各类疑难问题，从学生学习的角度，对相关知识加以阐述和提炼。

(3) 引导读者良好的程序设计风格和程序设计思路，让读者能理解解决问题的方法，以

达到触类旁通的效果。应用举例讲究经典实用而且丰富有趣，注重前后章节例题的连贯、一致和逐步深入。

(4) 主要面向初、中级读者群，又能兼顾高级读者的一些需求。内容的深浅选择上，既适合大多数初学者，又能满足少数高级读者深入学习的需求。先讲解基本知识，再探讨深层次的若干问题，以引起高级读者的兴趣。尽量把教学实践中学生学习中的问题反映到教材编写中，并加以解决，所以不但适用于学生读者，对教师读者也有一定的作用。

(5) 进一步完善初版教材中的学习指导内容和上机实践题目的设计，突出重点，加强应用，力求表述更为科学、阐述更加准确。所有例题、习题和上机操作题，都经过调试、运行和分析，以更好地方便学生进行自我测试、自我检查和自我提高。

(6) 增加大量的例题、实验上机题和考试模拟试题等，对各个考点的知识进行详尽的分析、研究和探讨，旨在帮助学生通过学习和练习，真正理解和掌握 C 程序设计的基本概念、基本理论知识和基本实践能力。设置各种不同层次等级的题目，以适用于不同的读者对象。

(7) 增加 C 在其他工程实践项目中的应用，让学生进一步理解 C 在各个工程领域中的应用实践情况，激发他们运用 C 解决专业问题的兴趣，切实提高他们应用 C 程序设计解决工程实际问题的能力和水平。

本书由刘韶涛、潘秀霞、应晖主编，其中所有章节内容由刘韶涛副教授进行了全面的修订、补充和完善。计算机科学与技术学院的陈维斌院长、陈锻生副院长、潘孝铭副院长、范慧琳副教授、余坚副教授等对教材的编写给予了全程的指导和关心，并给出了很多建设性的意见和建议。华侨大学教务处也对教材的编写和立项等工作都给予了大力的支持，在此一并表示衷心的感谢！

由于时间仓促，加上编者水平有限，书中难免存在不妥之处，恳请广大读者批评指正，我们的联系邮箱是 shaotaol@hqu.edu.cn。

编　者

2014 年 11 月

目　录

第 1 章　程序设计概述 ……1
1.1　计算机系统概述 …… 2
1.1.1　硬件基础知识 ……2
1.1.2　软件基础知识 ……5
1.1.3　计算机中数据的表示 ……6
1.1.4　数据在存储器中的特性 …… 12
1.2　程序设计语言 …… 14
1.2.1　机器语言 …… 14
1.2.2　汇编语言 …… 15
1.2.3　高级语言 …… 15
1.3　高级语言程序的创建和运行过程 …… 16
1.4　算法与数据结构概述 …… 17
1.4.1　算法的特性 …… 18
1.4.2　算法的表示 …… 19
1.4.3　算法示例 …… 21
1.4.4　数据结构的基本概念 …… 23
1.5　结构化程序设计 …… 24
1.5.1　结构化程序设计思想 …… 24
1.5.2　3 种基本程序结构 …… 25
1.5.3　结构化程序设计举例 …… 26
1.6　习题 …… 28
1.7　参考答案或解答提示 …… 28

第 2 章　C 语言概述 ……29
2.1　C 语言的发展和特点 …… 29
2.1.1　C 语言的发展背景 …… 29
2.1.2　ANSI C 的特点 …… 30
2.2　C 语言的程序结构与基本词汇符号 …… 31
2.2.1　C 语言的程序结构 …… 33
2.2.2　C 语言的基本词汇符号 …… 34
2.3　C 语言的编写风格 …… 36
2.4　运行 C 程序的步骤和方法 …… 39
2.4.1　基本过程 …… 39
2.4.2　错误处理 …… 40
2.5　习题 …… 41
2.6　参考答案或解答提示 …… 42

第 3 章　数据类型、运算符和表达式 …… 43
3.1　基本数据类型 …… 43
3.1.1　void 类型 …… 45
3.1.2　字符类型 …… 45
3.1.3　整数类型 …… 46
3.1.4　实数类型 …… 47
3.2　变量 …… 47
3.2.1　变量声明与定义 …… 47
3.2.2　变量初始化 …… 48
3.3　常量 …… 49
3.3.1　常量的表示 …… 49
3.3.2　代码常量 …… 52
3.4　运算符和表达式 …… 54
3.4.1　赋值运算符和赋值表达式 …… 55
3.4.2　算术运算符及表达式 …… 57
3.4.3　逗号运算符及逗号表达式 …… 58
3.4.4　关系运算符和逻辑运算符 …… 59
3.4.5　条件运算符 …… 61
3.4.6　常用标准函数的调用 …… 61
3.4.7　位运算符 …… 63
3.5　表达式求值 …… 65
3.5.1　优先级 …… 65
3.5.2　结合性 …… 65

3.5.3 表达式求值中的类型转换 …… 65
3.6 习题 …… 67
3.7 参考答案或解答提示 …… 69

第 4 章 顺序结构程序设计 …… 72
4.1 C 语言的语句 …… 73
4.1.1 空语句 …… 73
4.1.2 表达式语句 …… 73
4.1.3 复合语句 …… 74
4.1.4 控制语句 …… 75
4.2 C 语言中的输入输出 …… 75
4.2.1 流 …… 75
4.2.2 标准输入输出 …… 76
4.3 字符输入输出 …… 76
4.3.1 字符输出函数 putchar() …… 76
4.3.2 字符输入函数 getchar() …… 77
4.4 格式化输入输出 …… 78
4.4.1 格式化输出函数 printf() …… 78
4.4.2 格式化输入函数 scanf() …… 86
4.5 顺序结构程序设计举例 …… 92
4.6 习题 …… 95
4.7 参考答案或解答提示 …… 97

第 5 章 选择结构程序设计 …… 99
5.1 if 语句概述 …… 99
5.2 if 语句的使用 …… 100
5.2.1 单分支 if 语句 …… 100
5.2.2 双分支 if 语句 …… 101
5.2.3 多分支 if 语句 …… 103
5.2.4 if 的嵌套 …… 104
5.3 条件运算符与条件表达式 …… 107
5.4 switch 语句 …… 108
5.5 习题 …… 111
5.6 参考答案或解答提示 …… 112

第 6 章 循环结构程序设计 …… 115
6.1 while 当型循环 …… 115
6.2 do…while 型循环 …… 118
6.3 for 语句 …… 120
6.4 循环嵌套及其使用 …… 122
6.5 break 和 continue 语句 …… 129
6.5.1 break 语句 …… 129
6.5.2 continue 语句 …… 131
6.6 无条件跳转语句 goto …… 132
6.7 习题 …… 133
6.8 参考答案或解答提示 …… 133

第 7 章 数组 …… 136
7.1 数组的基本概念 …… 136
7.2 一维数组 …… 138
7.2.1 一维数组的定义 …… 138
7.2.2 一维数组的初始化 …… 141
7.2.3 一维数组的应用 …… 144
7.3 二维数组 …… 157
7.3.1 二维数组的定义 …… 157
7.3.2 二维数组的初始化 …… 159
7.3.3 二维数组的应用 …… 160
7.4 字符数组与字符串 …… 167
7.4.1 字符数组的定义 …… 167
7.4.2 字符数组的初始化 …… 170
7.4.3 字符数组与字符串 …… 171
7.4.4 字符串处理函数 …… 172
7.5 多维数组 …… 175
7.6 数组小结 …… 176
7.7 习题 …… 176
7.8 参考答案或解答提示 …… 177

第 8 章 函数 …… 184
8.1 函数的概念与定义 …… 184
8.1.1 函数的概念和分类 …… 184
8.1.2 函数的定义 …… 188
8.2 函数的参数与函数的返回值 …… 190
8.2.1 函数的参数 …… 190
8.2.2 函数参数的求值顺序 …… 192
8.2.3 函数的返回值 …… 192
8.3 函数的调用 …… 193
8.3.1 函数调用的概念 …… 193

8.3.2 函数调用的方式……194
8.3.3 函数的原型说明……194
8.3.4 函数的嵌套调用与递归调用……195
8.3.5 函数使用 const 形参……203
8.4 函数与数组……203
8.4.1 数组元素作为函数的实参……203
8.4.2 数组作为函数的参数……204
8.5 变量的类型……209
8.5.1 局部变量和全局变量……209
8.5.2 说明存储类型……212
8.6 全局函数和静态函数……216
8.6.1 全局函数……216
8.6.2 静态函数……218
8.7 参数类型与数量可变的函数……218
8.8 函数小结……218
8.9 习题……219
8.10 参考答案或解答提示……219

第 9 章 指针……224
9.1 指针的基本概念……224
9.1.1 指针变量的定义……226
9.1.2 与指针运算有关系的两个运算符……226
9.1.3 指针变量的引用……227
9.1.4 const 指针……229
9.2 指针与数组……231
9.2.1 指向数组元素的指针变量的定义……231
9.2.2 通过指针变量使用数组元素……232
9.2.3 指针与二维数组……234
9.2.4 指针与数组作为函数的参数……236
9.2.5 指针数组……239
9.3 指针与字符串……242
9.3.1 指向符串的指针……242
9.3.2 字符串指针作为函数参数……243
9.4 指针与函数……244
9.4.1 指针作为函数的参数……244
9.4.2 返回指针值的函数……245
9.4.3 指向函数的指针……248
9.4.4 命令行参数……251
9.5 多级间址……252
9.6 void 指针与动态内存分配……255
9.6.1 void 指针……255
9.6.2 动态内存分配……257
9.7 指针小结……262
9.8 习题……262
9.9 参考答案或解答提示……263

第 10 章 编译预处理……265
10.1 C 预处理程序……265
10.2 #define……266
10.3 #include……267
10.4 条件编译指令……268
10.4.1 #if、#else、#elif 和#endif……268
10.4.2 #ifdef 和#ifndef……269
10.5 #undef……270
10.6 小结……271
10.7 习题……271
10.8 参考答案或解答提示……273

第 11 章 结构体、共用体与枚举类型……275
11.1 结构体……275
11.1.1 结构体类型定义……275
11.1.2 结构体变量的定义、初始化及引用……276
11.1.3 结构体成员为结构体……278
11.1.4 结构体变量的初始化……278
11.1.5 结构体变量的引用……280
11.2 结构体数组……285
11.2.1 结构体数组的定义……286
11.2.2 结构体变量数组的初始化……287
11.2.3 结构体数组的引用……288

11.3 指向结构体类型的指针 …… 289
11.4 链表 …… 296
11.4.1 包含指针成员的结构变量 …… 296
11.4.2 单向链表的简单操作 …… 302
11.5 共用体 …… 311
11.5.1 共用体类型定义 …… 313
11.5.2 共用体变量的声明 …… 314
11.5.3 共用体变量的引用 …… 315
11.6 枚举类型 …… 316
11.6.1 枚举类型变量的声明 …… 317
11.6.2 枚举变量的引用 …… 317
11.7 typedef 定义类型 …… 319
11.8 习题 …… 321
11.9 参考答案或解答提示 …… 322

第 12 章 文件 …… 328
12.1 流和文件 …… 328
12.1.1 流 …… 328
12.1.2 文件 …… 329
12.1.3 文件类型的指针 …… 329
12.1.4 标准文件 …… 330
12.2 缓冲型文件的打开、关闭与读写 …… 330
12.2.1 fopen()函数 …… 331
12.2.2 fclose()函数 …… 331
12.2.3 fgetc()与 fputc()函数 …… 332
12.2.4 fread()与 fwrite()函数 …… 334
12.3 文件 I/O …… 336
12.3.1 fprintf()与 fscanf()函数 …… 336
12.3.2 fgets()与 fputs()函数 …… 338
12.3.3 文件读写指针移动函数 fseek()与 rewind() …… 339
12.3.4 ftell()和 feof()函数 …… 339
12.4 习题 …… 341
12.5 参考答案或解答提示 …… 341

第 13 章 C 的扩展简述 …… 342
13.1 面向对象的程序设计语言 C++ …… 342
13.1.1 程序设计方法的发展历程 …… 342
13.1.2 面向对象的基本概念 …… 343
13.1.3 面向对象的程序设计方法(OOP) …… 344
13.1.4 关于 C 与 C++ …… 345
13.2 Linux 下 C 编程简介 …… 347
13.2.1 Linux 简介 …… 347
13.2.2 Linux 下 C 编程环境概述 …… 348
13.2.3 Linux 程序设计的特点 …… 350
13.3 C 的其他应用简介 …… 350

附录 A ASCII 表 …… 353
附录 B 标准 C 函数库 …… 357
附录 C 运算符的优先级与结合性 …… 369
参考文献 …… 371

第1章　程序设计概述

基本内容：

- 计算机系统基础知识。
- 数据在内存中的存储。
- 程序设计语言基础知识。
- 高级语言编写程序的过程。
- 算法和数据结构的基础知识。
- 结构化程序设计的基本概念。

重点内容：

- 数据在内存中的存储特性。
- 程序设计语言的基础知识和高级语言编写程序的过程。
- 算法和数据结构的基本概念。
- 结构化程序设计的基本概念。

我们用 C 语言书写本书的第一个 C 程序，欢迎大家开始学习 C 程序设计，欢迎大家步入 C 语言的世界。

```
#include <stdio.h>
void main(void){
    printf("Welcome to learn C programming!\n");
    printf("Welcome to enter the C world!\n");
}
```

程序运行结果为：

```
Welcome to learn C programming!
Welcome to enter the C world!
```

本章将介绍和计算机程序设计有关的基本概念，包括计算机系统基础、程序设计语言的相关背景、程序设计的基础算法与数据结构以及程序设计方法等的基本概念和基础知识，为以后进一步学习 C 语言程序设计打下坚实的基础。

我们知道，计算机是通过指令与程序来完成许多激动人心的工作的，而且计算机的工作过程是一个系统的、多部件协同的复杂过程。要理解计算机的工作原理，得先从计算机的系统开始说起。

1.1　计算机系统概述

1.1.1　硬件基础知识

一个完整的计算机系统是由计算机硬件系统和计算机软件系统两部分组成的。硬件是计算机的实体，又称硬设备，是所有固定装置的总称。它是计算机实现其功能的物质基础，其基本配置可分为主机和外设，主机主要包括中央处理器和内存储器，外设包括输入和输出设备。

自 1946 年第一台电子计算机 ENIAC(Electronic Numerical Integrator And Computer)在美国诞生以来，至今已有近 70 年的历史。虽然电子计算机在外形、性能和应用领域等方面发生了巨大的变化，但是至今的电子计算机仍然沿用美籍匈牙利数学家冯·诺依曼等人提出的“存储程序与程序控制”的设计电子数字计算机的一些基本思想，该思想概括起来有如下一些要点：

(1) 计算机硬件由五大基本部件组成：运算器、控制器、存储器、输入和输出设备。

(2) 采用二进制形式表示计算机的指令和数据。

(3) 将程序(由一系列指令组成)和数据存放在存储器中，使计算机在工作时能够自动高速地从存储器中取出指令加以执行。

这些概念奠定了现在计算机的基本结构，并开创了程序设计的时代。半个多世纪以来，虽然计算机结构经历了重大的变化，性能也有了惊人的提高，但就其结构原理来说，至今占有主流地位的仍是以存储程序原理为基础的冯·诺依曼型计算机，如图 1.1 所示。

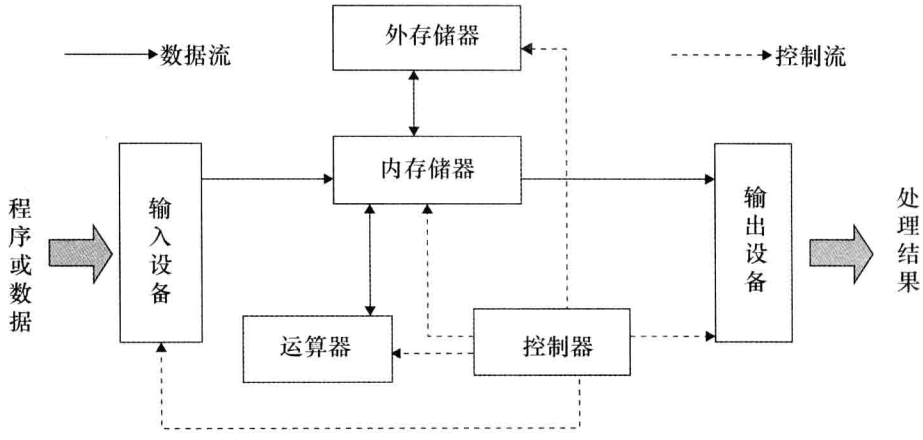


图 1.1　冯·诺依曼结构计算机

图 1.1 描述了计算机的工作流程，也就是应用计算机求解问题、执行程序的过程。首先将事先设计好的程序通过系统的输入设备，并在操作系统的统一控制下将程序送入内存储器。控制器通过程序指令的一步步执行来处理数据。最终执行的结果再由输出设备输出。

1. 硬件基本组成

计算机硬件系统是指所有构成计算机的物理实体，包括计算机系统中的一切电子、机械和光电设备等。计算机硬件的五大部分是输入设备、中央处理器(CPU，包括控制器和运算器)、主存储器和输出设备以及辅助存储设备等，如图 1.2 所示。

图 1.2　计算机硬件系统

输入设备通常是键盘，它将程序和数据输入计算机。其他的输入设备有鼠标、书写笔、触摸屏或基于声音的输入设备等。

CPU 负责执行算术、数据比较、移动数据(至系统中的不同位置)等指令。主存储器也称主存或内存，是在处理过程中临时存放数据或程序的场所。当关闭个人计算机或从一台分时计算机退出时，主存的数据会被删除。

输出设备通常是呈现输出信息的显示器或打印机。呈现在显示器上的输出信息称为软拷贝，打印机打印出的信息则称为硬拷贝。

辅助存储器也称外存，用于输入及输出，它是程序和数据的永久存放之处。当关闭计算机时，程序和数据仍然存放于辅助存储器中，可以满足其后的进一步使用需求。

2. 中央处理器

在一个计算机系统中，中央处理器(CPU)是由两个主要部分——运算器和控制器组成的。它是计算机的控制、运算中心，通过总线和其他设备进行联系。中央处理器的类型和品种非常多，各种中央处理器的性能差别也很大，有不同的内部结构、不同的指令系统。但由于基于冯·诺依曼结构，基本部分组成都很相似。

(1) 控制器

控制器是计算机系统的指挥中心，它把运算器、存储器和输入输出设备等部件组成一个有机的整体，然后根据指令的要求指挥全机的工作。

(2) 运算器

运算器是在控制器的控制下实现其功能的，运算器不仅可以完成数据信息的算术逻辑运算，还可以作为数据信息的传送通路。

基本的运算器组织包含如下几个部分：实现基本算术、逻辑运算功能的 ALU，提供操作数与暂存结果的寄存器组，有关的判别逻辑和控制电路等。将这些功能模块连接成一个整体时，运算器内的各功能模块之间的连接也广泛采用总线结构，这个总线称为运算器的内部总线，ALU 和各寄存器都挂在上面。

它的核心部分是加法器。因为四则运算加、减、乘、除等算法都归结为加法与移位操作，所以加法器的设计是算术逻辑线路设计的关键。

(3) 寄存器

寄存器是 CPU 中的一个重要组成部分，它是 CPU 内部的临时存储单元。寄存器既可以

用来存放数据和地址，也可以存放控制信息或 CPU 工作时的状态。在 CPU 中增加寄存器的数量，可以使 CPU 把执行程序时所需的数据尽可能存在寄存器中，从而减少访问内存的次数，提高其运行速度。但是寄存器的数目也不能太多，除了增加成本外，寄存器地址编码增加也会增加指令的长度。CPU 中的寄存器通常分为存放数据的寄存器、存放地址的寄存器、存放状态信息的寄存器和其他寄存器等类型。

3. 指令系统

计算机系统包括硬件和软件两大组成部分。硬件是指构成计算机的中央处理器、主存储器、外围输入输出设备等物理装置；软件则指软件厂家为方便用户使用计算机而提供的系统软件和用户用于完成自己的特定事务和信息处理任务而设计的用户程序软件。计算机能直接识别和运行的软件程序通常由该计算机的指令代码组成。

通常情况下，一条指令要由两部分内容组成，其格式为：

操作码	操作数地址

第一部分是指令的操作码。操作码用于指明本条指令的操作功能。例如，是算术加运算、减运算还是逻辑与、或运算功能，是否读、写外设操作功能，是否程序转移和子程序调用或返回操作功能等，计算机需要为每一条指令配一个确定的操作码。

一台计算机的指令系统往往由几十条到几百条指令组成。一般包括如下指令：

(1) 算术与逻辑运算指令。
(2) 移位操作指令。
(3) 数据传送指令。
(4) 转移指令、子程序调用与返回指令。
(5) 特权指令。
(6) 其他指令。

第二部分是指令的操作数地址，用于给出被操作信息(指令或数据)的地址，包括参加运算的一个或多个操作数地址、运算结果的保存地址、程序的转移地址、被除数调用的子程序的入口地址等。

寻址方式解决的是如何在指令中表示一个操作数的地址，如何用这种表示得到操作数或计算出操作数的地址。表示在指令中的操作数地址，通常被称为形式地址：用这种形式地址并结合某些规则可以计算出操作数在存储器中的存储单元地址，这一地址被称为物理(有效)地址。计算机中常用的寻址方式有如下多种：

(1) 立即数寻址。
(2) 直接寻址。
(3) 寄存器寻址、寄存器间接寻址。
(4) 变址选址。
(5) 相对寻址。
(6) 基地址寻址。
(7) 间接寻址。
(8) 堆栈寻址。

关于主存和辅存、I/O 接口和 I/O 设备等其他计算机硬件基础知识，可参考《计算机组成原理》等相关书籍，这里不再阐述。

1.1.2　软件基础知识

人们用各种电路器件制造的计算机称为物理计算机或裸机。裸机使用的语言是二进制形式表示的机器语言。在机器语言的基础上，人们开始了抽象层次更高、更接近人类自然语言的计算机语言，包括汇编语言和高级语言。汇编语言和高级语言需要进行翻译才能被计算机硬件识别。不同层次的用户使用不同层次的计算机语言与计算机进行交互，使计算机实现用户的要求，因此，可以把计算机看成是一个多层次的系统。裸机以上的用户所操作的计算机可以看成是一台相应的虚拟计算机。

软件是指挥计算机运行的程序集，计算机软件系统是计算机运行时所需的各种程序、数据及其相关文档的总称。不管硬件系统体系结构如何，按照功能，计算机软件一般可以分为两类：系统软件和应用软件。

系统软件管理计算机资源，是硬件和用户间的接口，但不直接服务于用户的需求。而另一方面，应用软件则直接负责帮助用户解决问题。图 1.3 描述了计算机软件的一般分类。

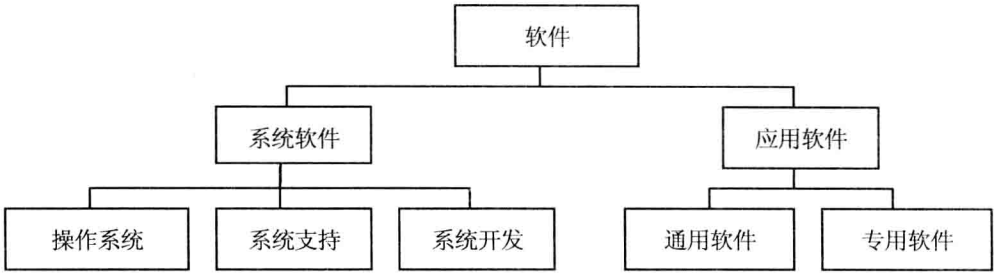


图 1.3　软件的类型

系统软件包括管理计算机硬件资源和执行信息处理任务时所需的程序。这些程序可分为：操作系统、系统支持和系统开发软件。操作系统提供了诸如用户界面、文件和数据库存取以及通信协议等功能。这种软件的主要目的是使系统以一种有效的方式运行，同时允许用户访问系统。系统支持(支撑)软件提供了系统工具和其他的运作服务：系统工具的实例包括排序程序和磁盘格式化程序等；运作服务则包括为保护系统和数据而给系统操作员和安全监控器提供系统性能统计数据的程序等。系统开发软件包括将程序翻译为可执行的机器语言的翻译器、保证程序免于错误的调试工具和计算机辅助软件工程(CASE)等。

应用软件分为两类：通用软件和专用软件。通用软件购买于软件开发商，可以用于多个应用。例如，字处理软件、数据库管理系统和计算机辅助设计系统等。它们被称为通用软件是因为它们能帮助用户解决多种计算机的普遍应用问题。专用软件仅被用于特定的目的。例如，设计人员使用的计算机辅助设计(CAD)软件，会计师使用的分户总账系统，建筑商使用的材料需求规划系统等。它们仅用于设计时指定的任务，不能应用于其他的一般性任务。

图 1.4 是系统软件和应用软件之间的关系。图中的圈表示界面，内部核心是硬件，用户位于外层。使用系统时，一般用户使用的是应用软件，应用软件和位于系统软件层的操作系统交互，而系统软件则提供了和硬件直接交互的能力。同时，用户在需要时可通过底部开口直接和操作系统交互。

如果用户买不到能满足需要的软件，那么就需要定制开发他们需要的软件。本书讲授的 C 语言就是当今众多软件开发工具中的一种。

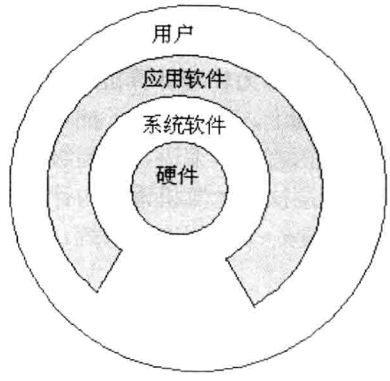


图 1.4　系统软件和应用软件间的关系

1.1.3　计算机中数据的表示

1. 数制

在采用进位记数的数字系统中，如果用 r 个基本符号(如 0,1,2,…,r−1)通过排列起来的符号串表示数值，则称其为基 r 数制，r 称为该数制的基。假定用 $m+k$ 个自左向右排列的符号 $D_i(-k \leqslant i \leqslant m-1)$表示数值 N，即

$N=D_{m-1}D_{m-2}\cdots D_1D_0D_{-1}D_{-2}\cdots D_{-k}$

式中的 $D_i(-k \leqslant i \leqslant m-1)$为该数制采用的基本符号，可取值 0,1,2,…,$r$−1，小数点位置隐含在 D_0 与 D_{-1} 之间，则 $D_{m-1}D_{m-2}\cdots D_1D_0$ 为 N 的整数部分，$D_{-1}D_{-2}\cdots D_{-k}$ 为 N 的小数部分。

如果每个 D_i 的单位值都赋以固定的值 W_i，则称 W_i 为该位的权，此时的数制称为有权的基 r 数制。此时 N 代表的实际值可表示为：

$$N=\sum_{i=-k}^{m-1}(D_i \times W_i)$$

如果该数制编码还符合“逢 r 进位”的规则，则每位的权(简称位权)可表示为：

$$W_i=r^i$$

式中的 r 是数制的基，i 为位序号。所以 N 代表的实际值也可以用下式表示：

$$N=\sum_{i=-k}^{m-1}(D_i \times r^i)$$

式中：r——这个数制的基；i——表示这些符号的列次序，即位序号；D_i——位序号为 i 的位上的符号；r^i——第 i 位上的一个 1 所代表的值(位权)；$D_i \times r^i$——第 i 位上的符号所代表的实际值；N——代表一个数值。

此时该数制称为 r 进位数制，简称 r 进制。表 1.1 所示是计算机中常用的几种进位数制。

表 1.1　计算机中常用的几种进位数制

数　制	进 位 数 制	基 本 符 号
二进制	r=2	0，1
八进制	r=8	0，1，2，3，4，5，6，7

(续表)

数　制	进 位 数 制	基 本 符 号
十六进制	r=16	0，1，2，3，4，5，6，7，8，9，A，B，C，D，E，F。其中 A~F 分别表示十进制的 10，11，12，13，14，15
十进制	r=10	0，1，2，3，4，5，6，7，8，9

如果每一数位都具有相同的基，即采用同样的基本符号集来表示，则称该数制为固定基数制，这是计算机内普遍采用的方案。在个别应用中，也允许对不同的数位或位段选用不同的基，即混合采用不同的基本符号集来表示，则该数制称为混合基数制。

图 1.5 示例为几种不同数制下表示的数值。

十进制数

位置：3　2　1　0　−1　−2

　　1　2　3　5.　4　5　　$=1\times10^3+2\times10^2+3\times10^1+5\times10^0+4\times10^{-1}+5\times10^{-2}$

权重：10^3　10^2　10^1　10^0　10^{-1}　10^{-2}

二进制数

位置：3　2　1　0　−1　−2　−3

　　1　0　1　1.　0　0　1　　$=1\times2^3+0\times2^2+1\times2^1+1\times2^0+0\times2^{-1}+0\times2^{-2}+1\times2^{-3}$

权重：2^3　2^2　2^1　2^0　2^{-1}　2^{-2}　2^{-3}

十六进制数

位置：2　1　0

　　A　0　E　$=10\times16^2+0\times16^1+14\times16^0$

权重：16^2　16^1　16^0

八进制数

位置：2　1　0

　　7　0　5　$=7\times8^2+0\times8^1+5\times8^0$

权重：8^2　8^1　8^0

图 1.5　各种数制表示的数值

2. 数制转换

计算机中有几种不同的进位数制，包括二(八和十六)进制和十进制。二进制数据容易用逻辑线路处理，更接近计算机硬件能直接识别和处理的电子化信息的使用要求，而十进制使人更容易接受。应熟练掌握两者之间的进制转换问题。

(1) 十进制数到二(八和十六)进制数的转换

$$N=\sum_{i=-k}^{m-1}(D_i\times r^i)\ (\text{其中 } r^i \text{ 为 } D_i \text{ 对应的位权})$$

所确定的运算规则是不同进位计数制数据之间完成进制转换的依据。

十进制到二进制的转换通常要区分数的整数部分和小数部分，并分别按除以 2 取余数部分和乘以 2 取整数部分两种不同的方法来完成。

对整数部分，要用除以 2 取余数的方法完成十进制到二进制的转换，其规则是：

① 用 2 除十进制的整数部分，取其余数为转换后的二进制数整数部分的低位数字。

② 再用 2 去除所得的商，取其余数为转换后的二进制数高一位的数字。

③ 重复前一步，直到商为 0，结束转换过程。

对小数部分，要用乘以 2 取整数的方法完成十进制数到二进制数的转换，其规则为：

① 用 2 乘十进制数的小数部分，取乘积的整数为转换后二进制小数的最高位数字。

② 再用 2 乘上一步乘积的小数部分，取新乘积的整数为转换后二进制小数低一位数字。

③ 重复前一步操作，直至乘积部分为 0，或已得到的二进制小数位满足要求，结束转换过程。

对小数进行转换的过程中，转换后的二进制已达到要求位数，而最后一次乘积的小数部分不为 0，会使转换结果存在误差，其误差值小于得到的最低一位的位权。

对既有整数部分又有小数部分的十进制数，可以先转换其整数部分为二进制数的整数部分，再转换其小数部分为二进制数的小数部分，把得到的两部分结果合并起来得到转换后的最终结果。例如，$(38.43)_{10}=(100110.01101)_2$。

十进制到八进制和十六进制数的转换方法与前述方法类似，只是乘除 8 和 16 时手工运算不太方便。

(2) 二进制数到八进制数、十六进制数的转换

用二进制表示一个数值 N，所用的位数 K 为 $\log_2 N$，如表示 4096，K 为 13，写起来位串较长。为此，计算机中也常常采用八进制和十六进制来表示数值数据。为表示 N，分别有如下对应关系：

$$N=\sum_{i=-k}^{m-1}(D_i\times 8^i)\ (D_i\text{的取值为 0~7})$$

例如，$(7.44)_8=7\times8^0+4\times8^{-1}+4\times8^{-2}=(7.5625)_{10}$。

$$N=\sum_{i=-k}^{m-1}(D_i\times 16^i)\ (D_i\text{的取值为 0~9 和 A~F})$$

例如，$(1A.08)_{16}=1\times16^1+10\times16^0+8\times16^{-2}=(26.03125)_{10}$。

在把二进制数转换成八进制或十六进制表示时，应从小数点所在位置分别向左、向右对每 3 位或每 4 位二进制位进行分组，写出每一组数对应的 1 位八进制数或十六进制数。若小数点左侧(整数部分)的位数不是 3 或 4 的整数倍，可以按在数的最左侧补 0 的方法处理；对小数点右侧(小数部分)，应按在数的最右侧补 0 的方法处理，否则容易转换错。对不存在小数部分的二进制数(整数)，应从低位开始向左每 3 位划分一组，使其对应 1 个八进制位，或把每 4 位划分成一组，使其对应 1 个十六进制位。例如，$(10.101)_2$ 变成八进制时，应把它理解为$(010.101)_2$，是$(2.5)_8$，即八进制的 2.5。当把它转换为十六进制时，应首先变为$(0010.1010)_2$，是$(2.A)_{16}$，即十六进制的 2.A，而不是$(2.5)_{16}$。又如：

$(1100111.10101101)_2=(147.532)_8$

$(1100111.10101101)_2=(67.AD)_{16}$

二进制、八进制和十六进制数之间的对应关系如表 1.2 所示。