普通高等教育“十一五”国家级规划教材

HZ BOOKS
华章教育

程序设计教程

用C++语言编程

第2版

陈家骏　郑滔　编著
南京大学

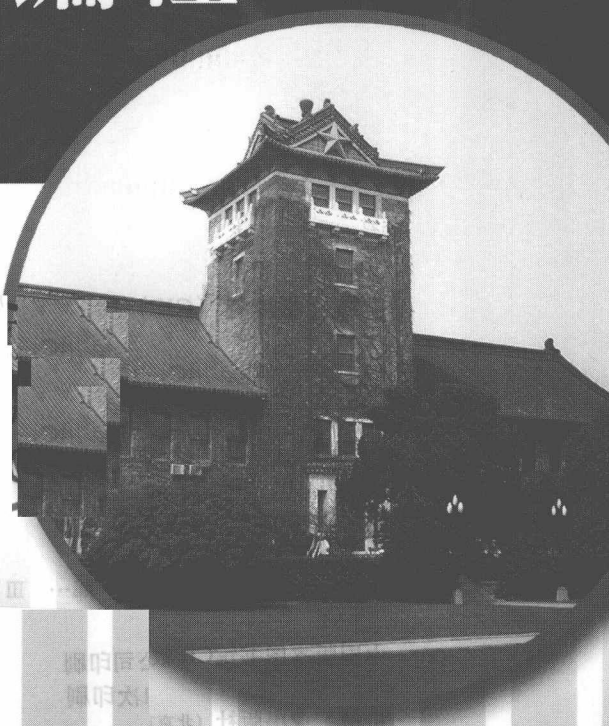
机械工业出版社
China Machine Press

普通高等教育“十一五”国家级规划教材

程序设计教程

用C++语言编程

第2版



陈家骏　郑滔　编著

南京大学

机械工业出版社
China Machine Press

本书是以C++作为实现语言的第一门程序设计课程的教材。以介绍基本的程序设计思想、概念和技术为中心，强调了数据结构、算法、过程抽象以及数据抽象等重要的程序设计思想。全书共12章，主要内容包括：数据类型、表达式、流程控制、子程序、递归、类/对象、继承、类属（泛型）、输入/输出以及异常处理等。内容相对完整，概念力求精确。

本书在第1版的基础上，相应地增加了例子、代码注释和习题，便于读者轻松且牢固地掌握程序设计的技巧。可作为高等院校本科生第一门程序设计课程的教材，也可供程序设计的初学者参考。

图书在版编目（CIP）数据

程序设计教程：用C++语言编程 / 陈家骏，郑滔编著. 2版. —北京：机械工业出版社，2009.4
（普通高等教育“十一五”国家级规划教材）

ISBN 978-7-111-26801-7

Ⅰ.程… Ⅱ.① 陈… ② 郑… Ⅲ.C语言－程序设计－高等学校－教材 Ⅳ.TP312

中国版本图书馆CIP数据核字（2009）第067658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：李俊竹
北京京北印刷有限公司印刷
2009年4月第2版第1次印刷
184mm×260mm · 20.25印张
标准书号：ISBN 978-7-111-26801-7
定价：35.00元

凡购本书，如有倒页、脱页、缺页，由本社发行部调换
本社购书热线：（010）68326294

第2版前言

本书第1版自2004年出版以来，得到了广大读者的热情关注和支持，很多读者还提出了宝贵的建议，我们深表感谢。

在近几年的教学中，我们也发现了本书的一些不足之处。首先，编写该教材的初衷是介绍程序设计的基本思想、概念和技术，C++语言是作为编程实现语言的角色出现，然而，在教材某些内容的表述上违背了这个初衷，教材的一些地方出现了C++语言“喧宾夺主”的情况。其次，教材在一些内容的表达上过于“精炼”，使初学者有“看天书”的感觉。再次，教材对现在比较流行的C++标准模板库（STL）以及它所支持的泛型程序设计没有给出足够的介绍，从而给读者学习使用STL带来了困难。此外，教材中还存在少量的错误。

针对上述问题，我们对教材进行了修订。第2版的变动主要体现在以下几个方面：

1) 重新组织了一些章节的内容，并调整了相应章节（主要是节）的标题和次序，进一步突出了程序设计的主流思想、概念和技术。

2) 对教材的文字进行了润色，补充了例子，并为例子中的程序代码增加了注释，使之更加容易理解。

3) 补充了对STL的介绍，包括一些常用的容器和算法以及它们的使用实例，有利于读者更好地进行泛型程序设计。

4) 增加了对计算机内部信息表示的介绍，使得读者能更好地理解程序设计中涉及的二进制。

5) 补充了一些习题，使读者有更多的机会进行有针对性的训练。

6) 对一些重要的程序设计术语用不同的字体加以突出的标注并给出了它们的英文对照，突出了对程序设计重要概念的介绍。

7) 修正了上一版中的一些错误。

在教材的修订过程中，得到了很多人帮助，在教材第2版出版之际向他们表示感谢，并希望继续得到大家的支持，使教材进一步得到完善。

作者于南京大学

2009年2月

第1版前言

随着计算机应用领域的不断扩大、应用层次的不断加深，社会对计算机软件的需求急剧增长，这就导致了软件的规模不断扩大、复杂程度不断提高。如何设计出大量的满足用户需求的高质量软件是软件工作者所面临的严峻挑战。

作为计算机软件的主要表现形式——计算机程序不同于其他程序（如音乐会程序），它是由计算机来执行的。这就使得计算机程序的编制（程序设计）不能完全以人的思维模式和习惯来进行，它往往要受到计算机解决问题的方式和特点的限制。除此之外，要设计出解决各种问题的程序，程序设计者往往还需要了解与问题领域有关的知识。这些都给程序设计带来一定的难度。

从程序设计的发展历史来看，程序设计经历了从低级语言到高级语言、从以编码为中心到面向软件生存周期的软件工程、从过程式到面向对象的发展过程。这一过程体现了人们对程序设计活动的不断认识和改进的过程，特别是从过程式程序设计到面向对象程序设计的发展，体现了人们对以自然的方式来描述和解决问题的需求，它使得解题过程更接近于人的思维方式。

有人认为程序设计是一门艺术，而艺术基于人的灵感和天赋。对于一些小型程序的设计而言，上述的说法可能有一些道理。但是，对于大型、复杂的程序设计问题，灵感和天赋是不能很好地解决问题的，几十年的程序设计实践已证明了这一点。不可否认，程序设计需要灵感和天赋，它们往往在程序的一些局部设计上发挥着作用。但从总体上讲，程序设计是一门科学，它是有规律和步骤可循的。通过对程序设计的基本思想、概念和技术的学习，再加上必要的训练和实践，程序设计的规律和步骤是可以掌握的，这正是本书的主旨所在。本书强调准确的程序设计基本概念、良好的程序设计风格和实际动手能力的训练与培养。

关于C++语言是否适合作为介绍程序设计时的编程实现语言，目前存在不同的看法。持反对意见的人认为C++太灵活，以至于会使初学者感到无所适从。本书之所以选择C++语言作为实现语言，首先，因为C++语言是一种流行的高级语言，很多人都在用C++编写实际的程序；其次，C++支持大部分基本的程序设计思想、概念和技术，其中包括对过程式及面向对象两种程序设计范型的支持。再次，与其他高级语言相比，C++语言具有灵活和高效等特点，这使得一些程序设计思想、概念和技术能够更好地实现。本书是以介绍基本的程序思想、概念和技术为主旨，C++服务于这个主旨，而不是相反，这样，初学者在使用C++时能够做到有的放矢。因此，本书对C++的一些特殊的、用于解决非主流的程序设计问题的成分和技巧不予重点介绍。特别地，本书对一些属于C++语言“文化”范畴的内容不予过分强调。

本书的内容分成两大部分：第1章至第5章为第一部分，主要对一些基本的程序设计思想、概念和技术以及过程式程序设计的基本内容进行介绍，其中包括：数据类型、表达式、流程控制、子程序（函数）、递归等；第6章至第12章为第二部分，重点介绍面向对象程序设计的基本内容，其中包括：类/对象、继承、操作符重载、类属（模板）、输入/输出、异常处理以及面向对象的Windows应用程序基本框架等。

本书既适合于程序设计的初学者使用，同时，对具有一些程序设计经验的人也有一定的参考价值。本书可以作为一学年的程序设计课程使用，其中，第一学期介绍第1章到第5章的全部内容以及第10章的部分内容；第二学期介绍第6章到第9章全部内容、第10章的部分内容以及第11章到12章的全部内容。如果读者已学过过程式程序设计（如C语言程序设计等）的基本内容，则本书也

可作为一学期的面向对象程序设计课程使用，重点介绍第6章到12章的内容。书中加“*”标记的节在初次阅读时可以跳过。

本书的编写和完成与很多人的帮助是分不开的。首先，要感谢郑国梁教授对本书编写工作的精心指导。在内容的选取、安排、用语的规范性等方面，郑老师都事无巨细地给予了考虑，并检查了全文（包括每个例子程序）。值得一提的是，作者编写本书所必备的专业知识和专业素质是在郑老师的长期熏陶下获得的，这些知识和素质使得作者能够完成本书的编写。其次，非常感谢尹存燕老师和戴新宇在本书习题的设计和文字易读性方面所做的大量工作；非常感谢孙明欣和周明对书中内容所做的检查工作，特别是对本书初稿中一些概念上的模糊与谬误、内容安排的合理性与易读性以及在遵守C++标准规范方面所提出的建议；感谢胡昊和徐锋，作者对一些基本概念的理解是在与他们就相关问题的讨论中获得的。另外，还要感谢机械工业出版社的温丽芳总编辑对本书编写工作的鼓励和支持。

最后，要感谢我们的家人对本书编写工作的理解和支持，本书的编写占用了大量的本应属于与他们在一起共度家庭欢乐的时间。感谢所有支持和帮助过本书编写工作的人们。

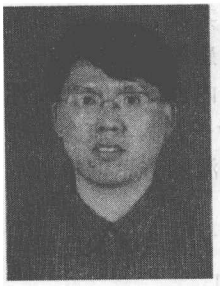
由于作者水平有限，加之时间仓促，错误和疏漏在所难免，恳请广大读者不吝指教，以便于我们在今后的版本中进行改进。

陈家骏
南京大学计算机科学与技术系
chenjj@nju.edu.cn

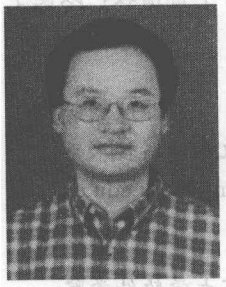
郑滔
南京大学软件学院
zt@nju.edu.cn

2004年4月

作者简介

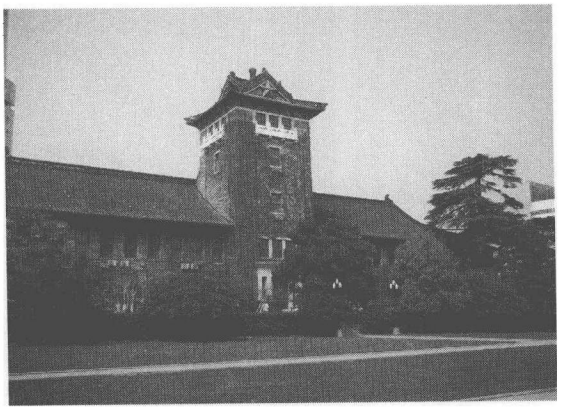


陈家骏，男，1963年生。获南京大学博士学位，现为南京大学计算机科学与技术系教授，博士生导师。主要从事自然语言处理和软件工程领域的研究工作，曾多次主持国家及省级科研项目的研究与开发，其中包括：863“基于语义和多策略融合的日汉机器翻译关键技术研究”、国家自然科学基金“基于统计关系学习的汉语指代消解研究”以及江苏省自然科学基金“基于条件随机场和核集成的自适应中文信息抽取技术研究”等项目，研究成果多次获得省部级科研奖励。具有多年的软件开发和程序设计课程教学的经历。



郑滔，男，1966年生。获南京大学硕士学位，现为南京大学软件学院教授。主要从事软件工程和嵌入式系统领域的研究工作。曾多次参加国家/省科技攻关项目、国家/省自然科学基金项目和国家863高科技项目的研究与开发，研究成果两次获得部级科技成果二等奖。具有多年软件开发和程序设计以及编译技术课程教学的经历。

封面图片介绍



南京大学座落在钟灵毓秀、虎踞龙盘的六朝古都南京，是直属国家教育部的重点综合性大学。南京大学治学严谨，具有多学科的综合优势和高素质的教师队伍，办学水平在全国高校中名列前茅。“诚朴雄伟，励学敦行”的校训继承和反映了南大百年办学的优良传统，同时体现了办学理念的更高追求。

南京大学的标志是位于鼓楼校区北园最北端的北大楼，她的特异之处在于布满全身的爬山虎，每至春夏之交，悠久的北大楼一片绿叶绵延，迎着入口处的石阶拾级而上，仰望她的宁静致远，俯看南大的建筑群体，静谧之气悄然而至。

教学建议

教学内容	学习要点及教学要求	课时安排	
		两学期	一学期
第1章　概述	• 了解冯·诺依曼（von Neumann）体系结构计算机的硬件和软件构成。 • 了解计算机内的信息表示（二进制）。 • *初步认识过程式及面向对象等程序设计范型。 • 清楚程序设计的基本步骤。 • 了解程序设计语言及其翻译（编译和解释）。 • *了解C++语言的特点以及C++程序的构成，掌握C++程序的运行步骤。 • 掌握C++语言的词法。	4	1
第2章　数据描述（I）——基本数据类型和表达式	• *了解数据类型的概念。 • 掌握C++的各种基本数据类型。 • 初步掌握常量和变量的使用以及变量值的输入。 • 初步掌握C++操作符的使用，对操作数的类型转换有一定的认识。 • 初步掌握表达式的使用以及表达式的输出。 • 了解表达式的副作用问题。	8	1
第3章　流程控制——语句	• 了解程序流程控制的基本思想。 • 掌握C++语言的顺序控制、选择控制和循环控制语句的使用。 • 熟练掌握基于计数循环和事件循环的程序设计。 • 了解无条件转移控制。 • *了解程序设计风格和结构化程序设计。	10	1
第4章　过程抽象——函数	• *了解基于过程抽象的程序设计。 • 掌握C++函数的定义、调用以及值参数传递。 • 掌握C++程序的多模块结构。 • *弄清标识符的作用域与变量的生存期概念。 • 掌握“分而治之”的程序设计思想和递归函数的使用。 • *了解内联函数、带默认值的形式参数以及函数名重载的基本内容。 • 了解C++的标准函数库。 • 了解基于条件编译的多环境程序编制以及程序调试技术。	12	3
第5章　数据描述（II）——构造数据类型	• 掌握枚举类型及其应用。 • 掌握一、二维数组以及字符串类型及其应用。 • 掌握结构类型及其应用。 • *理解联合类型的作用。 • 掌握指针类型的基本操作以及把指针应用于参数传递和动态变量。 • 了解指针与数组的配合使用。 • 了解函数指针的运用。 • 了解多级指针。 • *掌握引用类型的使用。	18	2

（续）

教学内容	学习要点及教学要求	课时安排	
		两学期	一学期
第6章　数据抽象——对象与类	• 理解基于数据抽象的程序设计思想。 • 掌握C++类的定义、类成员的访问控制、对象的创建和操作、对象的初始化以及消亡处理。 • 弄清this指针的作用。 • 掌握成员对象的初始化以及拷贝构造函数的使用。 • 弄清对常量对象的访问（常量对象）、对象之间的数据共享（静态成员）以及提高对私有成员访问效率（友元）等问题。 • 了解程序的基于类的模块结构。	12	12
第7章　操作符重载	• 了解操作符重载的基本思想。 • 掌握操作符重载的实现方法。 • 掌握C++几种特殊操作符重载的实现。 • 通过实例掌握操作符重载的具体应用。	8	6
第8章　类的继承——派生类	• 理解类继承作为软件复用的手段。 • 掌握C++单继承的定义和使用。 • 弄清C++的protected访问控制的作用。 • 理解C++继承方式的作用。 • 掌握派生类对象的初始化和赋值等操作。 • 了解聚集的概念以及与继承的比较。 • 掌握通过虚函数实现消息处理的动态绑定机制。 • 弄清多继承及其问题。	12	12
第9章　类属（泛型）机制——模板	• 理解类属（泛型）程序设计的基本思想。 • 弄清模板的基本思想。 • 掌握函数模板和类模板的基本使用。 • 理解模板的复用原理。 • 了解C++标准模板库STL的基本思想。学会使用STL的容器、算法以及迭代器来解决一些程序设计问题。	12	10
第10章　输入/输出	• 了解C++中输入/输出的基本做法。 • 了解面向控制台的输入/输出。掌握基于cin和cout的控制台输入/输出。 • 弄清文件的两种组织方式（文本文件和二进制文件）。掌握面向文件的输入/输出。 • 了解面向字符串变量的输入/输出。 • 掌握抽取操作符“>>”和插入操作符“<<”的重载。	8	4
第11章　异常处理	• 理解异常处理的基本思想。 • 掌握C++异常处理机制。	2	2
第12章　实例——面向对象的Windows应用程序框架	• 了解Windows应用程序的基本结构。 • 弄清面向对象的Windows应用程序的对象组成。 • 学会利用MFC进行Windows应用程序的开发。	10	10
	教学总学时建议	116	64

说明：

① 本教材可作为一学年（两学期）的程序设计课程使用，也可作为一学期的面向对象程序设计课程使用。对于后者，选择第1~5章中加“*”的部分进行重点讲解。

② 本教材授课学时包含理论讲授、课堂讨论、习题讲解以及课堂练习等必要的课内教学环节。

目　录

第1章 概 述

自1946年第一台数字电子计算机（ENIAC）问世以来，计算机在理论、技术以及应用等方面发展迅速。特别是计算机的应用，它已从早期的数值计算应用拓广到现在的大量的非数值计算应用，如管理信息系统、文字处理系统、基于Internet的Web浏览器以及嵌入式应用系统（如家电的电脑控制）等。现在，计算机已经渗透到人类社会活动的各个领域并发挥着巨大的作用。

一台计算机由硬件和软件两部分构成。硬件是指计算机的物理构成，软件主要是指计算机程序（指令序列）。硬件是计算机的物质基础，软件是计算机的灵魂。没有硬件就没有计算机；但是，如果只有硬件没有软件，可以说计算机几乎什么事情也做不了，要想用计算机来解决各种问题，必须要有相应的软件。从某种意义上讲，一台计算机的性能主要由硬件决定，而它的功能则主要由软件来提供。

随着计算机应用领域不断扩大、应用层次不断加深，社会对计算机软件的需求急剧增长，从而导致软件规模不断扩大、复杂程度不断提高。如何设计出大量的满足用户需求的高质量软件是软件工作者所面临的严峻挑战。

1.1 计算机的工作模型

计算机程序不同于其他程序（如音乐会程序），它是由计算机来执行的，编制计算机程序（程序设计）时通常要考虑到计算机解决问题的方式和特点。因此，要进行程序设计，就有必要对计算机的工作模型有一定的了解。下面将分别从计算机的硬件、软件以及计算机内部的信息表示几个方面来介绍计算机的工作模型。

1.1.1 硬件

硬件（hardware）是指构成计算机的元器件和设备。计算机元器件的发展经历了电子管、晶体管、集成电路以及超大规模集成电路等几个阶段，其集成度和速度在不断提高。虽然现在计算机的计算能力与早期的计算机相比已有了很大的提升，但是，目前的计算机基本上采用的还是传统的*冯·诺依曼体系结构*（von Neumann architecture）。冯·诺依曼结构的计算机由存储、算术/逻辑运算、控制、输入以及输出五个单元构成。*存储单元*用于存储程序（指令）和数据[⊖]；*算术/逻辑运算单元*用于进行计算；*控制单元*用于向其他单元发出控制信号，实现对整个系统的控制；*输入单元*和*输出单元*作为计算机与外界的接口，用于实现系统的输入和输出功能。冯·诺依曼计算机的本质是通过不断地改变程序的状态来实现计算，程序的状态由存储单元中的数据构成，状态的转换则通过程序中的指令来实现。

目前，计算机的硬件一般按中央处理器、内存以及外围设备三个部分来组织，它们之间通过用于传递数据、地址和控制信号的总线来连接（如图1-1所示）。

1. 中央处理器

中央处理器（central processing unit，简称CPU）是计算机的核心部件，它用于执行指令以完成计算任务。CPU由控制器、运算器以及寄存器等构成。*控制器*负责从内存中取指令并根据指令发出控制信号以引起其他部件的动作。*运算器*执行运算指令所规定的算术和逻辑运算。*寄存器*主

⊖ 冯·诺依曼体系结构的计算机又称为存储程序式计算机，程序和数据存储在同一个存储器中。在冯·诺依曼体系结构出现之前的计算机中，存储单元只存储数据，而程序则是以一种外插的形式接入系统。

要用于暂时存放当前指令所需要的数据和计算结果（供后续指令使用）、记录当前指令的执行状态以及下一条指令的内存地址等，其作用主要是减少访问（存/取）内存中内容的次数，提高指令的执行效率。

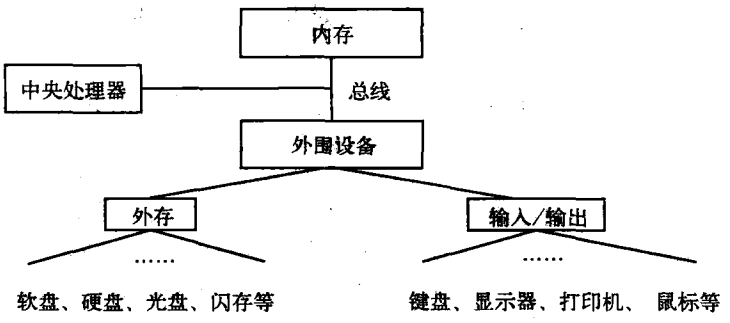


图1-1 典型的计算机硬件组成

CPU所能执行的指令通常有以下几种：

- 算术指令：实现加、减、乘、除等运算。
- 比较指令：比较两个数据的大小。
- 数据传输指令：实现CPU的寄存器、内存以及外设之间的数据传输。
- 执行流程控制指令：用于确定下一条指令的内存地址，包括转移、循环以及子程序调用/返回等指令。通常情况下，CPU从某个内存地址开始依次取指令来执行（顺序执行），执行流程控制指令可以改变指令顺序执行这个默认行为。

2. 内存

内存（memory）是内部存储器（或主存储器）的简称，它用于存储正在运行的程序和正在使用的数据。内存由许多存储单元构成，每个存储单元都有一个地址，对存储单元的访问是通过其地址来进行的。与CPU内的寄存器相比，内存的容量（内存单元的数目）要大得多，但指令访问内存单元的速度比访问寄存器要慢得多。

内存可分为只读存储器和随机访问存储器两部分。*只读存储器*（read only memory，简称ROM）中的内容只能读取不能修改，用于存储固定的程序（如开机引导程序）；*随机访问存储器*（random access memory，简称RAM）中的内容可以读取也可以修改，用于存储可变的程序和数据。内存中大部分是RAM，ROM只占很少一部分。另外，一旦断掉电源（如关机），RAM中的内容就丢失了，而ROM中的内容则一直保持着。

3. 外围设备

外围设备（peripheral device，简称外设）提供了计算机与外界的接口，主要用于计算机的输入/输出以及为计算机提供大容量的信息存储。外设分为输入/输出设备以及外部存储器两类。键盘和鼠标等属于*输入设备*（input device）；显示器和打印机等属于*输出设备*（output device）。

外部存储器（external storage，简称外存）是大容量的低速存储部件（与内存相比），用于永久性地存储程序、数据以及各种文档等信息。外存包括软盘、硬盘、光盘、闪存（U盘）、磁带等。存储在外存中的信息通常以文件形式进行组织，通过文件名来访问它们。外存与内存除了在容量和速度上不同外，它们的另一个区别在于：内存中存储的是正在运行的程序和正在使用的数据，而外存中存储的则是大量的、当前不使用的程序和数据。

计算机的工作过程是：把待执行的程序从外存装入到内存中，CPU从内存中逐条地取程序中的指令执行；程序执行中从外设或内存中获得所需要的数据，程序执行产生的临时结果保存在内存中，程序的最终执行结果通过外设输出。

由于构成计算机的各个部件存在速度上的差异，快速部件往往要花费大量的时间等待慢速部件的操作，因此，在冯·诺依曼结构计算机中存在着几个影响程序执行效率的瓶颈，这主要体现在CPU、内存以及外设之间的数据传输。现在的计算机中往往利用程序执行以及程序对数据的访问所具有的局部性特征（某段时间通常在一个范围内进行），通过*高速缓存*（cache）来解决部件之间速度不匹配问题，从而提高计算机的整体性能。缓存中存储的是近期用过的、今后可能还要用到的一些内容。例如，现代的计算机大都在CPU中为内存提供*内存高速缓存*（memory cache），以减少CPU访问内存的次数；在内存中为外存提供*磁盘高速缓存*（disk cache），以减少CPU访问外存的次数。

1.1.2 软件

软件（software）是计算机系统中的程序以及有关的文档。*程序*（program）是对计算任务的处理对象（数据）与处理规则（指令序列或算法）的描述，它由计算机来执行；*文档*（document）是为方便人理解程序而准备的资料和说明，它供程序开发与维护使用。

软件一般可以分为系统软件、支撑软件和应用软件。*系统软件*（system software）是计算机系统中直接让硬件发挥作用的软件，它与具体的应用领域无关，其他软件一般要通过系统软件发挥作用，如操作系统（Windows、UNIX、Linux等）就属于系统软件。*支撑软件*（supporting software）是指支持软件开发与维护的软件，一般由软件开发人员使用，如软件集成开发环境Visual C++等就是典型的支撑软件。*应用软件*（application software）是指用于特定领域的专用软件，如人口普查软件、财务软件等。有时，支撑软件也被归入到系统软件中。图1-2给出了各类计算机软件之间以及它们与计算机硬件之间的关系。

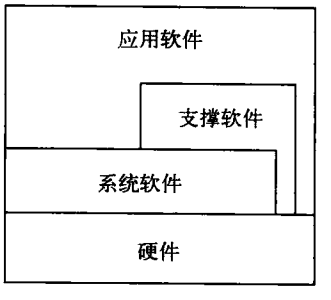


图1-2 各类软件之间以及它们与硬件之间的关系

从图1-2可以看出，与硬件“打交道”最多的是系统软件，它直接对硬件进行操作，其中包括系统资源的分配（如内存的分配）以及外设的驱动等。其他软件一般通过系统软件来操作硬件，其中，应用软件也会通过支撑软件来与系统软件打交道。当然，为了提高效率和灵活性，其他软件有时也能直接对硬件进行操作。

由硬件构成的计算机常常被称为“裸机”，在它之上，每加上一层软件就得到了一个比它功能更强的“*虚拟机*”（virtual machine）。例如，硬件加上操作系统就构成了最基本的虚拟机，我们在计算机上的大部分操作是在这个虚拟机上进行的。再例如，硬件构成的裸机只能识别用机器语言表示的指令，但如果在它上面加上C++的编译程序，则这个虚拟机就能执行由C++语言所表示的指令（语句）了。

1.1.3 机内信息表示

在计算机的内部，任何信息（包括指令、数据和地址）都是用一系列的“0”和“1”来表示的，它们分别对应着电气设备的两个稳定状态，如开关的开/关、电压的高/低、电流的有/无等。

对于基于“0”和“1”表示的信息，其常用的计量单位有：

- *二进制位*（bit）：由一个“0”或“1”构成。
- *字节*（byte）：由8个二进制位构成。
- *千字节*（kilobyte，简称KB）：由1024个字节构成。
- *兆字节*（megabyte，简称MB）：由1024个千字节构成。
- *吉字节*（gigabyte，简称GB）：由1024个兆字节构成。
- *太字节*（terabyte，简称TB）：由1024个吉字节构成。

在内存与外存中，通常用字节作为基本单位来组织信息。内存与外存的容量常常也是以字节

数来计算的。例如，内存的容量通常为512MB、1GB、2GB等；硬盘的容量通常为40GB、80GB、160GB等。

在计算机的信息表示中，数据的表示占有很重要的地位。下面简单介绍数的几种常见的进制表示以及它们之间的转换规则。

1. 数的进制表示

在日常生活中，数一般采用十进制来表示；而在计算机中，数则是以二进制来表示的。除此之外，常用的进制表示还有八进制和十六进制。表1-1给出了一些数的几种进制表示。

表1-1 一些数的几种进制表示

十进制	二进制	八进制	十六进制
0	0	0	0
1	1	1	1
2	10	2	2
3	11	3	3
4	100	4	4
5	101	5	5
6	110	6	6
7	111	7	7
8	1000	10	8
9	1001	11	9
10	1010	12	A
11	1011	13	B
12	1100	14	C
13	1101	15	D
14	1110	16	E
15	1111	17	F
16	10000	20	10

数的不同进制表示的特征如下：

- 十进制（decimal）：由0～9组成，计数时逢十进一。
- 八进制（octal）：由0～7组成，计数时逢八进一。
- 十六进制（hexadecimal）：由0～9以及A～F组成，计数时逢十六进一。
- 二进制（binary）：由0和1组成，计数时逢二进一。

例如，对于十进制数13，它的八进制表示为15，十六进制表示为D，二进制表示为1101。再例如，对于十进制数13加上3，各种进制数的加法运算如下：

$$\begin{array}{llll} (1\,3)_{10} & (1\,1\,0\,1)_2 & (1\ 5)_8 & (\ D)_{16} \\ +(\ 3)_{10} & +(\ _1{}_1{}_1{}_1)_2 & +(\ _1 3)_8 & +(\ _1 3)_{16} \\ \hline (1\,6)_{10} & (10\,0\,0\,0)_2 & (2\ 0)_8 & (1\,0)_{16} \end{array}$$

2. 十进制数与二进制数之间的转换

下面分别针对整数和小数来介绍十进制数与二进制数之间的转换。

(1) 对于整数，十进制数与二进制数之间的转换规则

十进制转成二进制

把十进制数连续除以基数2，直到商为0，所得的各个余数的倒序即为对应的二进制数。例如，对于十进制数29，其二进制表示的计算过程如下（结果为11101）：