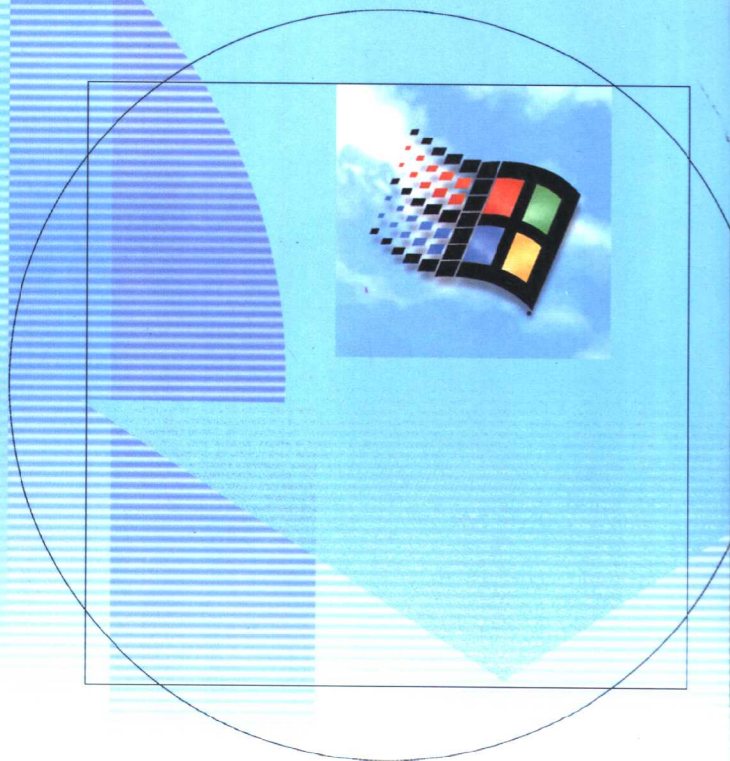


普通高等学校非计算机专业计算机基础系列教材



计 算 机 软 件 技 术

——语言、算法与程序设计方法

◆ 主 编 曾 一
◆ 副主编 李绍全
应 宏

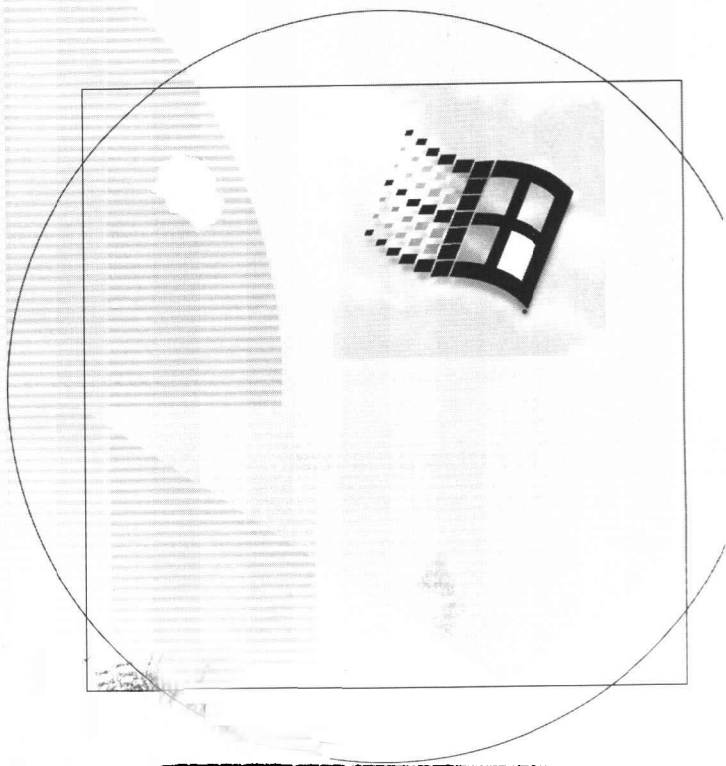
重庆大学出版社



普通高等

计算机基础系列教材

0



江苏工业学院图书馆

藏书章

计

算机软件技术

——语言、算法与程序设计方法

主 编 曾 一

副主编 李绍全 应 宏

参 编 (以姓氏笔画为序)

刘晏兵 应 宏 何 进

李绍全 陈 莉 杨芳明

聂永平 盛明兰 曾 一

重庆大学出版社

内容提要

本书从软件的角度出发,以程序设计方法为主要目标,以程序设计语言 C/C++ 为编程工具,介绍软件的基本概念、分类及其作用,构造软件的基本方法、技术和实现途径,如结构化程序设计的基本思想、模块化程序设计方法、数据组织和算法设计、面向对象的程序设计思想和方法、程序设计语言 C/C++ 等,通过典型的例题把程序设计方法和程序设计语言相结合,来进一步展现程序设计的全过程。

与本书配套编写的《计算机软件技术实验教程》同时由重庆大学出版社出版。本书可作为高校理工类非计算机本科、专科专业学生学习计算机程序设计语言、程序设计方法等方面的教材,也可作为高等教育自学考试教材和高等职业技术教育教材,以及从事计算机软件应用开发技术人员的参考书。

图书在版编目(CIP)数据

计算机软件技术/曾一主编. —重庆:重庆大学出版社,2003.2

普通高等学校非计算机专业计算机基础系列教材

ISBN 7-5624-2652-X

I. 计... II. 曾... III. 软件—高等学校—教材 IV. TP31

中国版本图书馆 CIP 数据核字(2003)第 007406 号

普通高等学校非计算机专业计算机基础系列教材

计算机软件技术

——语言、算法与程序设计方法

主 编 曾 一

副主编 李绍全 应 宏

责任编辑:肖顺杰 陈 其 张 彬 版式设计:吴庆渝

责任校对:任卓惠 责任印制:张永洋

*

重庆大学出版社出版发行

出版人:张鸽盛

社址:重庆市沙坪坝正街 174 号重庆大学(A 区)内

邮编:400044

电话:(023) 65102378 65105781

传真:(023) 65103686 65105565

网址: <http://www.cqup.com.cn>

邮箱: fxk@cqup.com.cn (市场营销部)

全国新华书店经销

万州日报印刷厂印刷

*

开本:787×1092 1/16 印张:19.5 字数:199 千

2003 年 2 月第 1 版 2003 年 2 月第 1 次印刷

印数:1—6 000

ISBN 7-5624-2652-X/TP·372 定价:26.00 元

本书如有印刷、装订等质量问题,本社负责调换

版权所有 翻印必究

编委会

顾 问 吴中福 邱玉辉

主 任 陈流汀

副主任 杨天怡 严欣平 张鸽盛

委 员 (以姓氏笔画为序)

王世迪 邓亚平 程小平

杨国才 范幸义 洪汝渝

郭松涛 黄 勤 曾 一

谭世语 熊 壮 莫 堃



序言

计算机技术的飞速发展,加快了人类进入信息社会的步伐,改变了世界,改变了人们的工作、学习和生活,对社会发展产生了广泛而深远的影响。计算机技术在其他各学科中的应用,极大地促进了各学科的发展。不掌握计算机技术,就无法掌握最先进、最有效的研究开发手段,将影响到其所从事学科的发展。因此,计算机技术基础是 21 世纪高校非计算机专业大学生必须掌握的、最重要的基础之一。

经过多年的探索和实践,按“计算机文化基础”、“计算机技术基础”、“计算机应用基础”三个层次组织教学已被公认为高校非计算机专业计算机基础教学的基本模式。第一层次开设“计算机文化基础”课程,教学的主要任务是使学生掌握计算机基础知识和基本操作能力;第二层次开设“计算机软件技术基础”和“计算机硬件技术基础”课程,教学的主要任务是使学生掌握计算机软、硬件技术的基本知识和基本开发技术;第三层次按专业群开设“计算机信息管理基础”、“计算机辅助设计基础”、“计算机网络技术基础”、“计算机控制技术基础”等不同课程,教学的主要任务是培养学生应用计算机技术分析解决本学科及相关领域问题的能力。

为了适应计算机技术的飞速发展和广泛应用对高校非计算机专业人才培养提出的新要求,我们组织一批



前言

本书是为适应计算机软件技术的发展和计算机基础教学改革的要求而编写的。

随着计算机技术的迅速发展,计算机软件技术的概念、方法和应用范围也在迅速变化。但是,在用计算机软件技术解决实际问题的時候,最基本的方法和技术就是程序设计。其中,结构化程序设计方法和面向对象程序设计方法以及能够支持这两种程序设计方法的程序设计语言 C 和 C++ 具有极大的影响力。因此,本书的组织将紧紧围绕程序设计方法和程序设计语言这两个主题逐步介绍相关的内容。第 1 章作为全书的概论,介绍了计算机系统中软件的概念、分类及其作用,程序、程序设计语言和程序设计方法,标准和规范、过程和管理等软件工程中的一些基本概念;第 2 章介绍程序设计的基础,从结构化程序设计方法的角度出发,首先介绍构成程序的 C 语言基本成分,然后介绍程序的基本结构,包括顺序结构、选择结构和循环结构,并通过程序设计举例进一步说明 C 语言对结构化程序设计方法的强有力的支持,为了加强程序的可读性和可理解性,本章还引入了程序设计风格的概念;第 3 章是企图从解决大问题的角度出发,进一步介绍一种体现结构化程序设计基本思想的方法即模块化程序设计方法的基本思想和概念,利用 C 语言对模块化程序设

计方法提供支持的特征,如函数、存储类、作用域、预编译等,通过典型例题说明在解决大问题时运用模块化程序设计方法的基本策略和一般步骤;第4章主要是结合C语言本身的特点,在引入如数组、指针、结构、共用体、文件等复杂数据类型的基础上,介绍如何利用这些丰富的数据类型、前述各章所介绍的C语言基础和程序设计方法等来解决实际问题,并通过大量例题予以说明;第5章实际上是程序设计的高级部分,为使初学计算机语言和程序设计的读者,尤其是非计算机专业的学生,不至于陷入讨论数据结构和算法设计本身,这里仅仅选择一些常用的基础数据结构和相应的算法如线性表、栈、队列、二叉树、分类和查找等内容,试图阐明“数据结构+算法=程序”的观点;第6章所介绍的内容是本书的另一个意图,那就是让学生初步建立面向对象程序设计的基本概念,并通过C++语言和简单例题让学生有一个认识、了解和体会面向对象程序设计方法的途径,这也为学生能够开展结构化程序设计方法和面向对象程序设计方法、以及支持结构化程序设计方法的C语言和支持面向对象程序设计方法的C++语言之间的比较研究提供了基础。

教材的每章还附有与内容紧密结合的习题,以促进学生的学习、巩固和提高,部分习题的参考答案、以及C语言常用函数、C++流类库、ASCII码对照表等在与本书配套编写的《计算机软件技术实验教程》中给出。

本书由曾一任主编,李绍全、应宏任副主编。各章编写分工如下:曾一编写第1章,杨芳明编写第2章,刘宴宾、何进、李绍全、聂永平编写第3章和第4章,应宏、盛明兰编写第5章,陈莉编写第6章。

编 者

2002年6月



目 录

1 软件技术初步

1

1.1 计算机系统	1
1.1.1 计算的需要及其局限性	1
1.1.2 计算机系统及其发展	4
1.1.3 计算机软件分类及其功能	6
1.1.4 计算机系统的应用模式	11
1.2 程序和程序设计	13
1.2.1 数据和数据类型	13
1.2.2 数据结构	14
1.2.3 算法和算法描述	14
1.2.4 程序和程序设计	17
1.3 程序设计语言	21
1.3.1 程序设计语言的基本概念	21
1.3.2 程序设计语言及其发展	22
1.4 程序设计方法	27
1.4.1 结构化程序设计方法	27
1.4.2 面向对象程序设计方法	30
1.5 软件工程	32
1.5.1 软件工程	32
1.5.2 软件的生命周期	34
1.5.3 软件过程与软件过程改进	36

习题 1	38
------------	----

2 程序设计基础

2.1 一个简单的程序	42
2.2 常量与变量	43
2.2.1 数据类型的概念	43
2.2.2 常量与变量	44
2.2.3 运算符与表达式	47
2.3 C 程序的结构	54
2.3.1 C 语言中的语句	54
2.3.2 C 程序的组成	55
2.4 程序的顺序结构	56
2.4.1 字符的输入与输出	56
2.4.2 格式化输入与输出	57
2.4.3 简单程序设计举例	60
2.5 程序的选择结构	62
2.5.1 选择结构的概念	62
2.5.2 C 语言中选择结构的实现	63
2.5.3 选择结构程序设计举例	72
2.6 程序的循环结构	74
2.6.1 循环结构的概念	74
2.6.2 C 语言中循环结构的实现	75
2.6.3 循环结构程序设计举例	81
2.7 程序设计风格	82
2.7.1 源程序文档化	82
2.7.2 数据说明	84
2.7.3 语句结构	85
2.7.4 输入和输出(I/O)	86
2.8 控制结构程序设计举例	86
习题 2	88

3 模块化程序设计

3.1 模块化程序设计的基本概念	93
3.1.1 模块化的基本思想	93
3.1.2 C 程序的模块结构	94



3.2 函数的定义和调用	95
3.2.1 函数的定义	95
3.2.2 函数的说明和调用	97
3.2.3 函数的参数传递	99
3.2.4 函数的返回方式	100
3.2.5 函数的嵌套调用和递归调用	102
3.3 作用域和存储类别	108
3.3.1 变量的作用域	108
3.3.2 变量的存储类别	114
3.3.3 内部函数和外部函数	118
3.4 编译预处理	119
3.4.1 “文件包含”处理	119
3.4.2 宏定义	120
3.5 系统函数	121
3.6 模块化程序设计举例	124
习题 3	127

4 复杂数据类型和程序设计

3

4.1 指针类型和程序设计	131
4.1.1 指针的概念	131
4.1.2 函数和指针	135
4.2 数组类型和程序设计	139
4.2.1 一维数组	140
4.2.2 二维数组	143
4.2.3 字符数组和字符串	146
4.2.4 指针和数组	152
4.3 结构类型和程序设计	161
4.3.1 结构类型变量的定义	162
4.3.2 结构类型变量的引用	164
4.3.3 结构变量的初始化	165
4.3.4 结构和函数	165
4.3.5 结构和数组	167
4.3.6 结构的嵌套	169
4.3.7 指向结构的指针	170
4.4 共用体和程序设计	175
4.5 位运算及位段	176
4.5.1 二进制位运算符及位运算	177
4.5.2 位段	181

4.6 文 件	183
4.6.1 文件的概念	184
4.6.2 文件的操作	186
4.7 复杂数据类型程序设计举例	195
习题 4	201

5 基础数据结构及算法

5.1 数据结构的基本概念	206
5.1.1 什么是数据结构	206
5.1.2 数据的逻辑结构	208
5.1.3 数据结构的存储方式	210
5.1.4 数据结构的基本运算	212
5.1.5 抽象数据类型和数据结构的 C 语言描述	212
5.2 线性数据结构	214
5.2.1 线性表及其基本运算	214
5.2.2 线性表的链接存储结构	218
5.2.3 栈和队列	223
5.2.4 C 语言实现线性数据结构程序设计应用举例	231
5.3 树型数据结构	233
5.3.1 树的基本概念	233
5.3.2 二叉树	235
5.4 查找与排序	239
5.4.1 查找	240
5.4.2 排序	247
习题 5	251

6 面向对象的程序设计和 C++ 语言

6.1 C++ 语言面向对象的特点	257
6.2 类	259
6.2.1 从结构到类	259
6.2.2 类与对象	260
6.2.3 访问类成员	266
6.2.4 成员函数的特性	272
6.3 继承	274
6.3.1 继承的概念及分类	274
6.3.2 单继承	275



6.3.3 多重继承	282
6.4 多态性	288
6.4.1 虚函数	289
6.4.2 抽象类和纯虚函数	292
6.4.3 由抽象类派生具体类	293
习题 6	295
主要参考书目	298



1

软件技术初步

在一个计算机系统中,不仅需要硬件,更需要软件。人们可以利用计算机语言,通过某种程序设计方法来编制小型应用软件,也可以利用软件工程的原理、技术、方法、规范、标准开发大型的应用软件。因此,了解有关软件技术的基本概念就显得非常重要。本章从软件技术的角度,介绍了计算机系统、计算机语言、程序设计方法和软件工程等方面的基本概念。

1.1 计算机系统

1.1.1 计算的需要及其局限性

几乎在整个历史进程中,人类都主要是依靠大脑来进行计算的。分析一下采用纸和笔进行手工计算的过程,如图 1.1 所示。纸相当于存储部件,其基本目的是信息存储。在纸上的信息,可以包括进行计算时应当遵循的一系列指令(即算法和程序),以及所用的数据,计算过程的中间结果和最终结果都记录在纸上。必要的计算过程在大脑中进行,大脑可以被称为处理器,完成两种主要功能:控制功能(即解释指令并保证以正确的顺序执行)和执行功能(即进行诸如加法、减法、乘法和除法等具体的计算)。但是,随着计算的规模和复杂程度的不断增长,这种计算存在着严重的不足:计算速度有限;计算容易出错。因此,人类借助于计算机来完成更快、更精确、更复杂的计算,其计算过程如图 1.2 所示。这里,存储部件(M)相当于手工计算所用的纸,用

来存放指令和数据;中央处理部件(CPU)由程序控制部件(PCU)和算术逻辑部件(ALU)构成,相当于人的大脑,程序控制部件解释指令并按顺序排好,算术逻辑部件执行指令。比较手工计算和借助于计算机计算的过程,不难发现,其最显著的差别是信息的表示方式不同。人类使用的是有着大量符号的正常语言,通常又以十进制的形式表示数。而在计算机中,信息的存储和处理均以二进制形式进行,只用0和1两个符号表示。为了在机器和用户之间提供通信,就需要在计算机语言与人类语言之间提供信息转换的手段,这就是图1.2中输入输出设备(I/O)的主要功能。

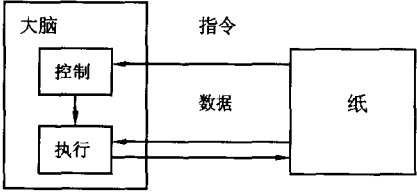


图 1.1 人工计算过程

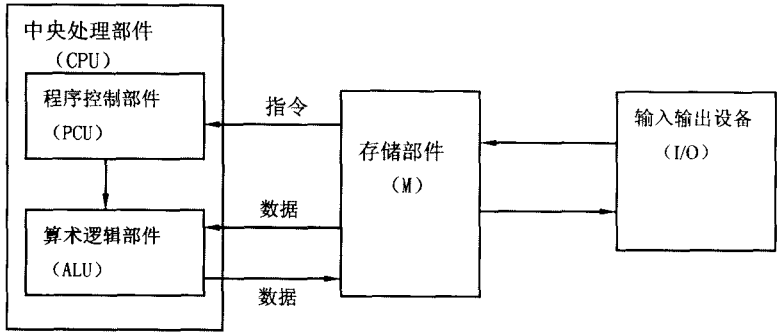


图 1.2 计算机计算过程

通过上述分析,可以把计算理解为求某个函数 $f(x)$ 的值,这里 x 是给予的输入数据, $z=f(x)$ 是要求输出的数据。 x, z, f 均可给予广泛的解释, x, z 可以代表数、词句、信息文件等等, f 可以是数值计算、定理证明、文件更新规程等等。为了用一特定计算机求 $f(x)$, 必须能将 f 表达成一串函数 $f_1, f_2, \dots, f_n$, 这些函数可以由计算机指令系统来确定。指令系统就是计算机能够执行基本功能的集合。 $f_1, f_2, \dots, f_n$ 就可以理解为对 $f(x)$ 求值的程序。基本操作的序列:

$$\begin{aligned} y_1 &= f_1(x) \\ y_2 &= f_2(y_1) \\ &\dots \\ y_{n-1} &= f_{n-1}(y_{n-2}) \\ z &= f_n(y_{n-1}) \end{aligned}$$

可以作为计算的一个形式定义。

那么,所有的问题是否都可以计算呢?

1936 年,英国数学家 Alan Turing 提出了计算机抽象模型——Turing 机器。人们在对 Turing 机研究的基础上,得到了如下的概念和认识:

①任何给定的 x , 如果 $f(x)$ 可以由 Turing 机在有限步内完成计算, 则称函数是



有效的或可计算的。实际上,很多计算机应用问题都是可计算的。

②存在不可计算的函数。例如,没有经验的程序员可能会编出一个在一定输入条件下无法停止的程序。又如,编写一个能测定任何程序是否停止的调试程序,这几乎是不可能的,因为只有当这个程序应用到全部可能的程序类时,这个说法才成立。这两个问题都属于不可解问题或不可计算的。

③即使问题是可解的,但也有难处理的或不现实的。例如,计算机不能实现任意大的二进制数的乘法计算。有大量的“困难”问题在实际中很重要,但在计算过程中其需要的存储空间太大以及计算时间太长,以致没有一台实际的计算机能够解它们。这样的结论,实际上是基于下面两个理由:一是计算机不可能存放所有可能的问题的答案;二是计算机处理信息的速度是有限的。

例如,Hanoi 塔问题。在一个基座上竖立着 3 根轴 A,B 和 C,轴 A 上套着 $n(n=64)$ 个圆盘,按直径的递减顺序由下而上叠放在一起,如图 1.3 所示。要求一次一个地把这些圆盘从 A 移到 B,但在任何情况下都不允许把大的圆盘叠放在小的圆盘上面,在移动过程中可以借助 C 轴来暂时存放圆盘。

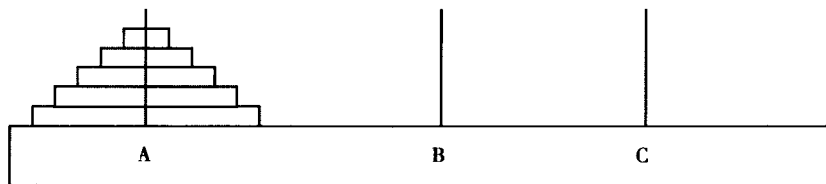


图 1.3 Hanoi 塔

这个移动过程是一个规模为 n 个圆盘的移动,可以进一步分解为规模为两次 $n-1$ 的移动和一次规模为 1 个圆盘的移动,而每个规模为 $n-1$ 的移动规则与规模为 n 的移动规则相同。移动过程可以描述为如下算法:

```
Hanoi(n,A,B,C)  /* 过程 Hanoi 把 A 顶上的 n 个圆盘借助 C 移到 B */
{
    如果 n=1 则
        把它从 A 轴移到 B 轴; /* 如果盘数 n=1 即只有一个圆盘 */
    否则{
        Hanoi(n-1,A,C,B); /* 先把 A 顶上的 n-1 个圆盘借助 B 移到 C */
        把 A 剩下的最大的圆盘一次移到 B;
        Hanoi(n-1,C,B,A) /* 同样把 C 中的 n-1 个圆盘借助 A 移到 B */
    }
}
```

这个过程实际上是一个递归算法,用数学归纳法不难证明这个算法的正确性。但这里关心的是计算机是否能够解决一个任意大小的 n 的 Hanoi 塔问题。为了回答这个问题,不妨把上面的移动过程中的移动次数表示为如下的递归方程:

$$T(n) = \begin{cases} 1 & (n=1) \\ 2T(n-1)+1 & (n>1) \end{cases}$$

然后解这个递归方程,可以得到移动次数为:

$$\begin{aligned}
T(n) &= 2 * T(n-1) + 1 \\
&= 2 * (2 * T(n-2) + 1) + 1 \\
&= 2^2 * (2 * T(n-3) + 1) + 2^1 + 2^0 \\
&= 2^3 * T(n-3) + 2^2 + 2^1 + 2^0 \\
&\dots \\
&= 2^{n-1} + 2^{n-2} + \dots + 2^2 + 2^1 + 2^0 \\
&= 2^n - 1
\end{aligned}$$

通过计算可得,64 个盘的移动次数为: $2^{64}-1=18\ 466\ 744\ 073\ 709\ 511\ 615$ 次。假设一台计算机 $1\mu\text{s}(1\text{s}=10^6\mu\text{s})$ 可移动一次圆盘,那么把这个算法转换成计算机程序来完成移动 64 个圆盘所需的时间将近 60 万年。

显然,Hanoi 塔问题虽然有解决算法,但是属于难解问题,即使计算机的速度由于技术的改进有所增长,解决类似问题仍然存在着局限性。

这就给出一个启示,单纯依靠提高计算机的速度是不能从根本上解决这类问题。也就是说,在运用计算机解决实际问题的过程中,不仅要考虑计算机硬件技术的应用,更重要的是要考虑计算机软件方面的技术、方法和过程的应用,应该从系统的角度考虑问题解决的途径。

1.1.2 计算机系统及其发展

所谓系统,通俗地说就是有组织、有秩序地达到某种目的的集合。一只手表、一台电视机是一个系统;一个人、一个足球队、一个商店、一个医院、一个港口、一个发电厂是一个系统;一个国家的工业、农业也是一个系统;甚至一个国家或一个世界都可以认为是一个大系统。一个系统应该具有整体性、层次性和适应性。

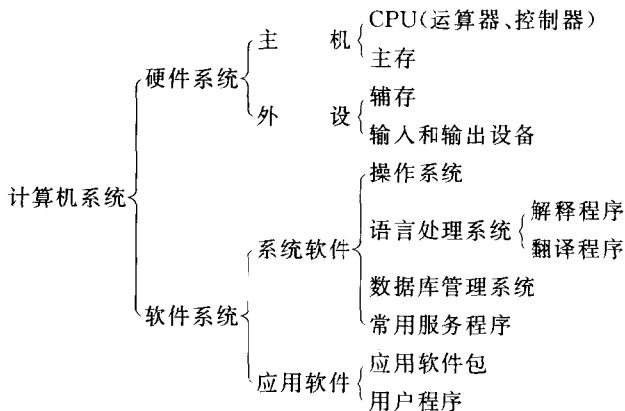


图 1.4 计算机系统

一个计算机系统(如图 1.4 所示),是指由计算机的硬件系统和软件系统组成的统一整体。软件系统是计算机系统不可缺少的一部分,是计算机系统能否发挥效能的主要成分,但其效能的发挥又必须通过计算机系统的硬件部分表现出来。

从第一台计算机 ENIAC(Electronic Numerical Integrator And Calculator)到当



今的 Internet 的应用,计算机科学与技术的发展突飞猛进。表 1.1 给出了计算机系统发展的大致过程。

表 1.1 计算机系统发展的大致过程

年代 项目	第 1 代 1946—1956	第 2 代 1957—1963	第 3 代 1964—1981	第 4 代 1984—2002
硬件	电子管 磁鼓	晶体管 磁芯存储器	集成电路 半导体存储器 磁盘	大规模/超大规模集成电路 微处理器/多处理器 光盘 分布式系统、嵌入式系统
软件	机器代码的 存储程序 机器语言 汇编语言	高级语言 Fortran 语言 Algol 语言 Cobol 语言 Lisp 语言	结构化程序设计 操作系统 软件工程 数据库系统 Simula 语言 Pascal 语言 Smalltalk 语言 C 语言 Prolog 语言	面向对象程序设计 Ada、C++、Java、VC 面向逻辑程序设计 面向函数程序设计 面向对象数据库系统 专家系统/知识库系统 应用软件包(MRP II /ERP) 视窗操作系统 第 4 代语言 4GL PC 软件工具(字处理软件、 多媒体软件、集成软件包) 超文本/可扩展标记语言 HTML/XML Web 浏览器(Netscape/IE) 软件质量认证(ISO 9000) 软件过程改进(CMM/CM- MI)

硬件(Hardware)是构成计算机的物理装置,是看得见、摸得着的一些实实在在的有形实体。计算机硬件系统是计算机运行的物质基础,计算机的性能如运算速度、存储容量、计算精度、可靠性等,在很大程度上取决于硬件的配置。

软件(Software)是指使计算机运行需要的程序、数据和有关的技术文档资料。没有软件,计算机是不能执行交给它的任务。概括地说,软件的功能包括:

①管理企业或组织内部计算机的资源。

②为人们利用这些资源提供工具。

③在企业或组织与所存储的信息之间提供中间服务。软件是计算机的灵魂,是发挥计算机功能的关键。有了软件,人们可以不必过多地去了解计算机本身的结构与原理,方便、灵活地使用计算机。软件的功能越强,使用起来就越方便。