

软件技术 基础教程

臧 浏 王珊珊 潘梅园 尤彬彬 张定会 编著



TP311.5
605

014057559

软件技术 基础教程

臧浏 王珊珊 潘梅园 尤彬彬 张定会 编著



TP311.5
605



北航

C1745983



南京大学出版社

014027223

图书在版编目(CIP)数据

软件技术基础教程 / 臧浏等编著. —南京: 南京大学出版社, 2014. 7

ISBN 978-7-305-13041-0

I. ①软… II. ①臧… III. ①软件—高等学校—教材
IV. ①TP31

中国版本图书馆 CIP 数据核字(2014)第 068832 号

内容简介

教材面向高等院校非计算机专业的学生,书中系统地介绍了计算机软件方面的相关知识,内容涵盖数据结构、软件工程、数据库原理和操作系统的基础知识。尽量用通俗、简洁的语言来描述与计算机软件相关的基本概念,基本原理和软件设计的基本方法。此外,每章均配有习题,以帮助读者理解和掌握书中的内容。为任课教师免费提供电子课件及习题解答。

本书内容丰富,结构合理,循序渐进、实用性强,便于教学和自学,既可作为全国普通高等院校非计算机专业学习计算机软件的通用教材,也可作为参加计算机等级考试的参考用书。

出版发行 南京大学出版社

社 址 南京市汉口路 22 号

邮 编 210093

出版人 金鑫荣

书 名 软件技术基础教程

编 著 臧 浏 王珊珊 潘梅园 尤彬彬 张定会

责任编辑 邓海琴 单 宁

编辑热线 025-83596923

照 排 江苏南大印刷厂

印 刷 盐城市华光印刷厂

开 本 787×1092 1/16 印张 18.5 字数 461 千

版 次 2014 年 7 月第 1 版 2014 年 7 月第 1 次印刷

ISBN 978-7-305-13041-0

定 价 39.00 元

网 址: <http://www.njupco.com>

官方微博: <http://weibo.com/njupco>

官方微信号: njupress

销售咨询热线: (025)83594756

* 版权所有,侵权必究

* 凡购买南大版图书,如有印装质量问题,请与所购
图书销售部门联系调换

前 言

随着信息技术日益深入到人们生活和工作的各个环节,作为信息技术的核心——计算机应用技术,已经不只是计算机专业人员的专门技术。现在,越来越多的软件需要由非计算机专业的人员开发,专业人员开发本专业的软件有着天然的优势,而专业人员又缺乏计算机专业方面的知识。因此对非计算机专业的在校大学生来说,掌握计算机基础知识,能够以专业背景为依托,充分有效地运用计算机技术解决未来工作中的专业问题,显得越来越重要。而提高计算机的应用能力,尤其是计算机软件的开发和应用能力,对计算机软件系统的相关知识和原理有一个系统的学习和掌握是非常必要的。

作为以计算机应用为主要目的未来工程技术人员,除了要掌握现有计算机的基本使用方法外,还必须要掌握软件设计与开发的基本知识和有关技术。这些技术包括以算法为核心的算法与数据结构,以数据库为基础的信息系统的开发和组织,以操作系统为基础的计算机资源的利用和管理,以软件开发工程化为主线的软件工程技术等内容。

本书面向高校非计算机专业学生,旨在提高其计算机应用能力特别是算法设计能力、计算机系统资源管理能力、数据库系统应用能力和软件开发能力。书中的内容力求使读者了解计算机软件知识的全貌,内容丰富,涉及面广。

根据我们多年来的教学实践以及计算机等级考试的相关要求,全书内容由五章组成。第一章是概述,主要介绍了计算机软件的发展特点。第二章是算法与数据结构的相关内容,介绍数据结构和算法的基本概念、线性结构、树结构、图结构以及查找与排序算法。第三章是软件工程的相关内容,主要介绍软件危机及软件工程方法学、结构化软件开发方法、面向对象软件开发等内容。第四章是数据库系统的相关内容,主要包括数据模型及相关概念、结构化查询语言、数据库设计与应用等内容。第五章是操作系统的相关内容,包括操作系统总论、处理器管理、存储管理、设备管理及文件管理等内容。每章后面都配备了相应的习题供读者练习,以检查学习掌握的情况。

在编写过程中,我们力求做到概念清晰,取材合理,深入浅出,突出应用。

本书第一、第二章的 2.1~2.6 节由臧浏编写;第二章的 2.7~2.9 节由尤彬彬编写;第三章由王珊珊编写,第四章由张定会编写;第五章由潘梅园编写。全书由臧浏负责统稿。感谢皮德常教授百忙之中抽空对本书进行了审阅,并提出宝贵意见。

本教材在学校已经作为自编教材使用三年,但由于作者水平有限,书中欠妥之处,敬请广大读者和专家批评指正。

zangliwen@nuaa.edu.cn

编 者

2014. 4

目 录

04	第三套题参考答案	0.0.0
84	第四套题	0.0.0
82	第五套题	0.0.0
22	第六套题	0.0.0
72	第七套题	0.0.0
02	第八套题	0.0.0
50	第九套题	0.0.0
50	第十套题	0.0.0
第 1 章	概述	1
1.1	数学预备知识	1
1.1.1	集合	1
1.1.2	常用数学术语	2
1.1.3	对数	3
1.1.4	递归	3
1.1.5	级数求和	3
1.1.6	数学证明方法	4
1.2	软件技术的发展	4
1.2.1	程序设计语言的发展	4
1.2.2	软件工程的发展	5
1.2.3	数据库技术的发展	6
1.2.4	操作系统的发展	7
习题 1		8
第 2 章	基本数据结构及算法	9
2.1	概述	9
2.1.1	数据结构的概念	9
2.1.2	算法及算法评价	13
2.2	线性表	23
2.2.1	线性表的逻辑结构	23
2.2.2	线性表的物理结构	24
2.2.3	线性表的基本运算	27
2.2.4	线性表的应用	33
2.3	栈和队列	44
2.3.1	栈的基本概念	44
2.3.2	栈的顺序存储及其运算	44

2.3.3	栈的链式存储及其运算	46
2.3.4	栈的应用	48
2.3.5	队列的基本概念	53
2.3.6	循环队列及其运算	55
2.3.7	队列的链式存储	57
2.3.8	队列的应用	59
2.4	索引存储结构	62
2.4.1	索引存储结构的概念	62
2.4.2	“顺序-索引-顺序”存储方式	63
2.4.3	“顺序-索引-链接”存储方式	64
2.5	数组	65
2.5.1	数组的定义	66
2.5.2	数组的顺序存储	66
2.5.3	稀疏矩阵	69
2.6	树与二叉树	73
2.6.1	树的基本概念	74
2.6.2	二叉树及其基本性质	75
2.6.3	二叉树的遍历	79
2.6.4	树转换成二叉树	83
2.6.5	二叉树的应用	84
2.7	图	89
2.7.1	图的基本概念	89
2.7.2	图的存储结构	91
2.7.3	图的遍历	94
2.8	查找	99
2.8.1	查找的基本概念	99
2.8.2	线性表的查找	99
2.8.3	树型查找	106
2.8.4	哈希 (Hash) 表技术	107
2.9	排序	118
2.9.1	冒泡排序	118
2.9.2	快速排序	119
2.9.3	直接插入排序	122
2.9.4	希尔排序	123
2.9.5	简单选择排序	124

2.9.6 堆排序	126
2.9.7 归并排序	130
2.9.8 排序算法总结	132
习题2	133
第3章 软件工程基础	136
3.1 软件与软件工程	136
3.1.1 软件	136
3.1.2 软件工程	136
3.1.3 软件生命周期	137
3.1.4 软件工程的目标与原则	139
3.2 需求分析	140
3.2.1 需求分析的任务和内容	140
3.2.2 需求分析方法——结构化分析	141
3.2.3 结构化分析常用工具	142
3.3 软件设计	147
3.3.1 软件设计的任务和目标	147
3.3.2 软件设计方法——结构化设计	147
3.3.3 用户界面设计	156
3.4 程序编码	159
3.4.1 程序设计语言	159
3.4.2 程序设计风格	161
3.5 测试与调试的基本技术	162
3.5.1 测试	163
3.5.2 调试	170
3.6 软件维护	171
3.7 软件开发新技术	172
3.7.1 快速原型方法	172
3.7.2 面向对象方法	173
习题3	178
第4章 数据库设计基础	179
4.1 数据库基础知识	179
4.1.1 数据库系统的由来	179
4.1.2 数据库系统结构	180

4.1.3	数据库系统的组成	181
4.1.4	数据库管理系统	182
4.1.5	数据模型	183
4.2	关系数据库	186
4.2.1	关系模型的数据结构及其形式化定义	186
4.2.2	关系代数	189
4.3	关系数据库规范化理论	194
4.3.1	关系模式的设计问题	194
4.3.2	函数依赖	196
4.3.3	范式	196
4.4	关系数据库标准语言 SQL	199
4.4.1	SQL 概述	199
4.4.2	SQL 的数据定义	200
4.4.3	SQL 的数据查询	204
4.4.4	SQL 的数据更新	209
4.4.5	视图	210
4.5	数据库设计与管理	211
4.5.1	数据库设计概述	212
4.5.2	需求分析阶段	212
4.5.3	概念结构设计	213
4.5.4	逻辑结构设计	218
4.5.5	数据库的完整性	219
4.5.6	并发处理与控制	222
习题 4		226
第 5 章	操作系统基础	229
5.1	概述	229
5.1.1	操作系统的概念	229
5.1.2	操作系统的基本类型	232
5.1.3	操作系统的功能	235
5.1.4	流行操作系统简介	237
5.2	处理器管理	238
5.2.1	中央处理器	238
5.2.2	中断技术	240
5.2.3	进程与线程	241

5.2.4	处理器调度	244
5.3	存储管理	247
5.3.1	存储器	247
5.3.2	存储管理方法	252
5.3.3	虚拟存储管理	264
5.4	设备管理	266
5.4.1	设备管理的任务和功能	266
5.4.2	设备类型	268
5.4.3	通道	268
5.4.4	缓冲技术	270
5.4.5	虚拟设备(SPOOLing 技术)	273
5.5	文件管理	274
5.5.1	文件的概念	274
5.5.2	文件目录	277
5.5.3	文件的组织	279
5.5.4	文件的使用	281
习题 5		282
参考文献		283

第1章 概述

1.1 数学预备知识

1.1.1 集合

数学意义上的集合概念在计算机科学领域有广泛的用途。

定义 1.1 集合(Set)是由互不相同的成员(Member)或者元素(Element)构成的一个整体。成员取自一个更大的范围,称为基类型(Base Type)。集合的每个成员或者是基类型的一个基本元素(Primitive Element),或者它本身也是一个集合。集合中没有重复的概念,取自基类型的每个值要么在集合内要么不在集合内。

例如,集合 R 由整数 1、2、3 组成,此时 R 的成员是 1、2、3, R 的基类型是整型。依赖集合的基类型,它的成员经常有一个线性顺序。

定义 1.2 线性顺序(Linear Order)有如下性质:

1. 对于集合 S 中的两个元素 a 和 b ,三个表达式 $a < b$, $a = b$, $a > b$ 中有且仅有一个为真;

2. 对于集合 S 中的任意三个元素 a 、 b 、 c ,如果 $a < b$ 、 $b < c$,则有 $a < c$,这称为传递性。

有线性顺序的基类型实例包括整型、字符型和实型。但是,如果一个集合是由各种颜色的球组成的,在这些不同颜色的球之间并不存在一个相对顺序的标准。如果需要人为地给它们规定一个顺序,可以使用有些程序设计语言中提供的枚举类型。表 1-1 给出了表示集合和它们之间关系的常用符号。下面给出使用这种表示方法的实例。首先定义两个集合 R 和 S :

$$R = \{1, 2, 3\}, S = \{3, 5\}$$

则有 $|R| = 3$ (因为 R 有 3 个成员), $|S| = 2$ (因为 S 有 2 个成员)。 R 和 S 的并是由 R 和 S 中的元素组成的集合,记为 $R \cup S$,上例 $R \cup S$ 为 $\{1, 2, 3, 5\}$ 。 R 和 S 的交是由既在 R 中又在 S 中的元素组成的集合,记为 $R \cap S$,上例 $R \cap S$ 为 $\{3\}$ 。 R 和 S 的差是由在 R 中但不在 S 中的元素组成的集合,记为 $R - S$,上例 $R - S$ 为 $\{1, 2\}$ 。注意:总是有 $R \cup S = S \cup R$, $R \cap S = S \cap R$,但通常 $R - S \neq S - R$,本例中 $S - R = \{5\}$ 。

表 1-1 集合的表示方法

$\{1, 8\}$	由数字 1 和 8 组成的集合
$\{x x \text{ 是正整数}\}$	使用规范形式定义的集合,例:所有正整数的集合

续表

$x \in R$	x 是集合 R 中的一个元素
$x \notin R$	x 不是集合 R 中的一个元素
$\emptyset$	空集
$ R $	基数: 集合 R 的大小或者元素数目
$R \subseteq S, S \supseteq R$	集合 R 包含在集合 S 中, 集合 R 是集合 S 的一个子集, 集合 S 是集合 R 的一个超集
$R \cup S$	集合的并: 所有属于集合 R 或者集合 S 的元素
$R \cap S$	集合的交: 所有属于集合 R 并且属于集合 S 的元素
$R - S$	集合的差: 所有属于集合 R 而不属于集合 S 的元素

一个与集合有关的概念是序列。

定义 1.3 长度为 n 的一个有限序列是定义域为集合 $\{0, 1, \dots, n-1\}$ 的一个函数 $f(x)$ ($0 \leq x \leq n-1$)。这个定义的含义与集合的不同之处为:

- (1) 序列的元素有一个顺序(第 1 个成员, 第 2 个成员等等)。
- (2) 成员可以重复, 但是它们是序列的不同成员(因为 $f(i)$ 可能与 $f(j)$ 相等)。

1.1.2 常用数学术语

1. 阶乘函数: 阶乘函数 $n!$ 是指从 1 到 n 之间所有整数的连乘, 其中 n 为大于 0 的整数。因此, $6! = 1 \times 2 \times 3 \times 4 \times 5 \times 6 = 720$ 。特别地, $0! = 1$ 。阶乘函数随着 n 的增大迅速增长。由于直接计算阶乘函数非常耗时, 所以有时使用一个公式做近似计算是非常有用的。Stirling 近似公式为:

$$n! \approx \sqrt{2\pi n} \left(\frac{n}{e}\right)^n$$

其中 $e \approx 2.71828$ (e 是自然对数的底数)。

2. 排列: 一个序列的排列就是把这个序列的成员按照一定顺序组织起来。例如, 数字 1 到 n 的排列可以把这些数字按照任意顺序放置。如果一个序列有 n 个不同的成员, 那么这个序列有 $n!$ 种不同的排列。因为排列中的第一个成员有 n 种选择方法; 对于每个选定的第一个成员, 第二个成员有 $n-1$ 种选择方法; 依此类推。有时需要得到某个序列的一个随机排列, 即 $n!$ 种可能排列中任何一种被选中的概率是相同的。

3. 布尔变量: 布尔变量是一个只能取值为真或假的变量。这两个值常常分别与值 1 和 0 对应, 这样规定并没有任何特殊的理由。

4. 取下整与取上整: 实数 x 的取下整(记为 $\lfloor x \rfloor$) 返回不超过 x 的最大整数。例如, $\lfloor 6.4 \rfloor = 6$, 与 $\lfloor 6.0 \rfloor$ 的结果一样。实数 x 的取上整(记为 $\lceil x \rceil$) 返回不小于 x 的最小整数。例如, $\lceil 6.4 \rceil = 7$, 与 $\lceil 7.0 \rceil$ 的结果一样。

5. 取模操作符: 取模返回整除后的余数。有时在数学表达式中用 $n \bmod m$ 表示, 在 C 或 C++ 语言中取模操作符的表示为 $n \% m$ 。从余数的定义可知, $n \bmod m$ 得到一个整数 r , 满足 $n = qm + r$, 其中 q 为一个整数, 且 $0 \leq r < m$ 。也可将取模结果表示为 $n - m \lfloor n/m \rfloor$ 。 $n \bmod m$ 的结果一定在 0 到 $m-1$ 之间。例如: $5 \bmod 3 = 2$, $25 \bmod 3 = 1$, $5 \bmod 7 = 5$, $5 \bmod 5 = 0$,

$-3 \bmod 5 = 2$ 。最后一个实例所给出的结果是以数学上取模的定义为基础得到的。在某些程序设计语言中(比如C++),取模操作符的任何一个操作数为负,运算结果将随着编译器的不同而不同。

1.1.3 对数

以 b 为底 y 的对数定义为使得 b 的某次幂等于 y 的那个指数,记为 $\log_b y = x$ 。从而可得:

$$\log_b y = x \Leftrightarrow b^x = y \Leftrightarrow b^{\log_b y} = y$$

其中符号“ $\Leftrightarrow$ ”表示“等价于”。

在算法分析时经常使用对数。例如,在一个线性表中查找指定值所使用的二分法查找算法是一个把问题分解为更小问题的算法。二分法查找首先与中间元素进行比较,以确定下一步是在上半部分进行查找还是在下半部分进行查找,然后继续将适当的子表分半,直到找到指定的值(二分法查找算法在2.8.2节中介绍)。一个长度为 n 的线性表被逐次分半,直到最后的子表中只有一个元素,一共需要分 $\log_2 n$ 次。

1.1.4 递归

如果一个算法用自己来完成它的部分工作,就称这个算法是递归的。这种方法要想获得成功,必须在比问题小的问题上调用自己。总而言之,一个算法必须有两个部分:初始情况和递归部分。初始情况只处理可以直接解决而不需要再次递归调用的简单输入。递归部分包含对算法的一次或者多次递归调用,每一次的调用参数都在某种程度上比原始调用参数更接近初始情况。

递归算法的设计总能够使用下面的方法实现。首先写出初始情况,然后考虑通过组合一个或多个较小但类似子问题的结果来解决问题。如果编写的算法是正确的,那么当然可以依靠它来递归地解决更小的问题。这种方法非常简单。

递归方法在日常生活的问题求解中没有类似的概念。由于它需要使用一种新的思维方式来考虑问题,因此这个概念很难掌握。为了有效地使用递归,必须强制自己尽量不用非递归方法来思考问题。只要在不超过递归调用的范围内分析递归过程,子问题就会迎刃而解。一个实例基于著名的“汉诺(Hanoi)塔”问题,它的C++实现有多个递归调用,它不是那么容易就能够使用 while 循环改写的,具体实现将在2.1.2节中介绍。

把递归作为一种主要用于设计和描述算法的工具,对于不熟悉它的编程人员是很难接受的。递归算法通常不是解决问题最有效的计算机程序,因为递归包含函数调用,比其他替代选择诸如 while 循环等,所花费的代价更大。但是,递归有时提供了一种能合理有效地解决问题的算法。

1.1.5 级数求和

大多数程序都具有循环结构,当分析带有循环的程序运行时间时,需要把每次循环执行的时间累加起来,这就是一个级数求和的例子。级数求和,简单地说,就是把函数在一定范围内所取的值累加起来,一般表示为:

$$\sum_{i=1}^n f(i)$$

这个记号表示对作用于某个(整数)范围内的值 i 的函数 $f(i)$ 之值求和,表达式的参数和它的初值写在 $\sum$ 符号的下面。这里,记号 $i=1$ 表明参数是 i ,它的初值是 1。 $\sum$ 符号的上面是表达式 n ,表示参数 i 的最大值。因此,这种表示法表示当 i 从 1 变到 n 时对 $f(i)$ 的值求和。例如 $\sum_{i=1}^n i = \frac{n(n+1)}{2}$ 。

1.1.6 数学证明方法

这里简单介绍一种证明方法:数学归纳法。

数学归纳法通常被用于证明某个给定命题在整个(或者局部)自然数范围内成立。该方法促使人们从简单的子问题入手考虑问题的求解。

设 T 是一个要证明的定理,用一个正整数参数 n 来表达 T ,数学归纳法表明,如果下面两个条件为真,那么对于参数 $n(n \geq c, c$ 是一个较小的常量)的任何值, T 都是正确的:

(1) 初始情况: $n = c$ 时 T 成立;

(2) 归纳步骤: 如果 $n = k$ 时 T 成立,则对于 $n = k + 1$ 时 T 也成立。

【例 1.1】用数学归纳法证明前 n 个自然数的和为 $n(n+1)/2$ 。

证明: (1) 当 $n = 1$ 时, $1 \times (1+1)/2 = 1$ 成立;

(2) 假设 $n = k$ 时成立,即: $\sum_{i=1}^k i = \frac{k(k+1)}{2}$;

(3) 当 $n = k + 1$ 时, $\sum_{i=1}^{k+1} i = \sum_{i=1}^k i + k + 1 = \frac{k(k+1)}{2} + k + 1 = \frac{(k+1)(k+1+1)}{2}$

结论成立;

所以,由数学归纳法得出 $\sum_{i=1}^n i = \frac{n(n+1)}{2}$ 。

1.2 软件技术的发展

当今社会正由工业社会向信息化社会迈进,信息技术的核心技术是软件技术,软件产业已经成为信息产业中的龙头产业。本节对软件技术中相关的程序设计语言的发展、软件工程的发展、操作系统的发展及数据库技术的发展作一简单介绍。

1.2.1 程序设计语言的发展

程序设计语言经历了一个从低级到高级的发展过程,从最初的机器语言发展到今天的面向对象语言,语言的抽象程度越来越高,程序设计语言的风格也越来越接近人类自然语言的风格,其发展主要经历了 4 个阶段。

1. 从机器语言到汇编语言

最初的计算机操作是用 0 和 1 的数字串(即机器语言)直接和计算机硬件打交道的。这种方式难读、难记、难理解。后来用助记符简化机器语言指令(即汇编语言),初步解决了难

读、难记、难用、难理解的问题。

2. 从汇编语言到高级语言

汇编语言执行速度快,适合编写直接和计算机打交道的程序。有了汇编语言,减轻了人们的编程工作,但由于它缺少对数据类型和程序控制结构的表示手段,用起来还是很吃力。由于汇编源程序与机器语言程序有简单的对应关系,即一条汇编指令对应一条二进制的机器指令,不同种类的机器有不同的机器语言,不同类型的机器之间,“不懂”对方的语言,所以用汇编语言编写的程序缺乏通用性和可移植性。于是人们又设计出了“高级程序设计语言”,相对地,机器语言和汇编语言称为低级语言,它们是依赖于机器的。

3. 从高级语言到第四代语言

高级语言的代表语言是 Basic、Pascal、Fortran 和 C,这些语言具有丰富的数据类型和程序控制结构,这比汇编语言前进了一大步,但是用这些语言设计程序仍然要告诉计算机一步一步如何做。能否只告诉计算机下一步要做什么,而不必告诉具体如何做,第4代语言(4GL)达到了该目标,这种语言以数据库操纵语言为代表,如结构化查询语言(SQL)。例如要从数据库中查询满足条件的一些记录,用 SQL 语言只需要一条查询命令即可,若是用高级程序设计语言(如 C++ 语言),需要多条语句才能达到同样的效果。

4. 从第四代语言到面向对象语言

面向对象程序设计技术将对象作为程序的基本结构单元,对象将数据及对该数据的操作封装在一起成为一个相对独立的实体,以简单的接口对外提供服务,面向对象程序设计语言通过提供继承与派生、多态性、模板等技术,使开发者最大程度地重用已有的程序代码,大大提高程序的开发效率。目前常见的面向对象程序设计语言有 C++ 和 Java, C++ 是由 C 发展而来的,它是一种混合的面向对象的编程语言,因为它支持 C 语言的过程程序设计;而 Java 是纯面向对象的编程语言。面向对象的程序设计语言都提供了相应的集成开发环境 IDE(Integrated Development Environment),方便了用户程序的开发。

1.2.2 软件工程的发展

随着计算机硬件性能的飞速提高,计算机在各行各业都得到了广泛的应用,但计算机硬件一旦离开软件的支撑,就不能发挥它的巨大作用,软件已成为计算机系统的重要组成部分,软件开发也已成为当今社会的主要产业。

软件开发经历了3个时期,第1个时期是程序时期(20世纪50年代初~20世纪50年代末),该时期程序作为机器运行时必须进行的准备工作。程序设计全凭设计者个人经验和技巧独立进行,是一种典型的手工智力劳动。第2个时期是“软件=程序+说明”时期(20世纪50年代末~20世纪70年代初),程序规模较大,需要多人协作才能完成,是“作坊式”的生产;程序的设计与运行维护不能由一个人来承担;程序不再是计算机硬件的附属部分,而是计算机系统中与硬件相互依存不可缺少的部分。第3个时期是“软件=程序+文档”时期(20世纪70年代至今,即软件工程时期),用“工程化”的思想来指导软件研究和开发。

经过几十年软件工程的实践,虽然解决了软件开发中的部分问题,但软件行业的生产效率依然没有很大的提高。虽然许多大型的应用软件采取了大规模的生产和协作,但这种软件往往开发时间长、效率低、无法动态调整,无法由僵硬变得灵活和敏捷。软件工程较开始时已经有了很大的进步,如软件开发过程、开发方法、质量保证、项目管理及工具环境等方面都变得成

熟起来。但要成为一个完全成熟的学科还要做大量的工作。

软件工程化是一项庞大的系统工程,必须借助一个应用工具平台才能有效地实施,比如 Rational Rose、国内的北大青鸟系统等。

具体对于程序设计人员而言,软件的开发必须以工程化的思想为指导,运用标准和规范的方法来进行。程序编码,即通常所说的“编程”,是软件开发过程中的一个关键阶段。程序是软件的主体,软件的质量主要是通过程序的质量来体现的,因此程序设计在软件开发中占有十分重要的地位。程序设计人员既需要有严格的思维能力,又需要有创意及一定的艺术修养。

1.2.3 数据库技术的发展

数据库技术是一项理论成熟,应用极广的数据管理技术,它从诞生到现在,在不到半个世纪的时间里,形成了坚实的理论基础、成熟的商业产品和广泛的应用领域。数据库的诞生和发展给计算机信息管理带来了一场巨大的革命。四十多年来,国内外已经开发建设了成千上万的数据库,它已成为人们日常工作和生活不可缺少的一部分。同时,随着应用的扩展与深入,数据库的数量和规模越来越大,数据库的研究领域也已经大大地拓广和深化。

数据管理技术经历了手工管理、文件管理和数据库管理 3 个发展阶段。

手工管理数据不能保存大量的数据,没有软件系统对数据进行管理,基本上没有“文件”的概念且一组数据对应一个程序。

文件管理数据比手工管理数据前进了一大步,它可以以文件的形式将数据长久地保存在计算机的外存中,文件系统由专门的数据管理软件提供有关数据存取、查询及维护的功能,文件系统中的数据文件多样化,文件系统的数据存取是以记录为单位的。但文件系统的数据库冗余大,缺乏数据与程序间的独立性。

数据库技术从 20 世纪 60 年代中期开始萌芽,当时计算机开始广泛地应用于数据管理,对数据的共享提出了越来越高的要求。传统的文件系统已经不能满足人们的需要,能够统一管理和共享数据的数据库管理系统(DBMS)应运而生。数据模型是数据库系统的核心和基础,各种 DBMS 软件都是基于某种数据模型的。所以通常也按照数据模型的特点将传统数据库系统分成网状数据库、层次数据库和关系数据库 3 类。

最早出现的是网状 DBMS,是在 1961 年开发成功的 IDS(Integrated DataStore),其奠定了网状数据库的基础,后来重写了整个系统,并将重写后的系统命名为集成数据管理系统(IDMS)。网状数据库模型对于层次和非层次结构的事物都能比较自然的模拟,在关系数据库出现之前网状 DBMS 要比层次 DBMS 用得普遍。在数据库发展史上,网状数据库占有重要地位。

层次型 DBMS 是紧随网状数据库而出现的。最著名典型的层次数据库系统是 IBM 公司在 1968 年开发的 IMS(Information Management System),它是一种适合其主机的层次数据库。这是 IBM 公司研制的最早的大型数据库系统程序产品。至今已经发展到 IMSV6,提供群集、N 路数据共享、消息队列共享等先进特性的支持。这个具有 30 年历史的数据库产品在如今的 WWW 应用连接、商务智能应用中扮演着新的角色。

网状数据库和层次数据库已经很好地解决了数据的集中和共享问题,但是在数据独立性和抽象级别上仍有很大缺陷。用户在对这两种数据库进行存取时,仍然需要明确数据的存储结构,指出存取路径。而后来出现的关系数据库较好地解决了这些问题。

关系数据库用数学理论作基础,采用关系模型作为数据的组织方式。关系型数据库系统以关系代数作为坚实的理论基础,经过几十年的发展和实际应用,技术越来越成熟和完善。其代表产品有 Oracle、IBM 公司的 DB2、微软公司的 MS SQL Server 以及 Informix 等。

关系数据库的标准语言是 SQL (Structured Query Language),它是在 1974 年提出的。SQL 语言的功能包括查询、操作、定义和控制,是一个通用的、功能极强的关系数据库语言,同时又是一种高度非过程化的语言,只要求用户指出做什么而不需要指出怎么做。SQL 集成实现了数据库生命周期中的全部操作。SQL 提供了与关系数据库进行交互的方法,它可以与标准的编程语言一起工作。SQL 语言标准的每一次变更都指引着关系数据库产品的发展方向。

关系型数据库系统虽然技术很成熟,但其局限性也是显而易见的,虽然能很好地处理所谓的“表格型数据”,却对技术界出现的越来越多的复杂类型的数据无能为力。九十年代以后,技术界一直在研究和寻求新型数据库系统。但在什么是新型数据库系统发展方向上的问题,产业界一度是相当困惑的。受当时技术风潮的影响,在相当一段时间内,人们把大量的精力花在研究“面向对象的数据库系统(Object Oriented DataBase)”或简称“OO 数据库系统”上。然而,数年的发展表明,面向对象的关系型数据库系统产品的市场发展的情况并不理想。另外,面向对象的关系型数据库系统使查询语言变得极其复杂,从而使得无论是数据库的开发商家还是应用客户都将其复杂的应用技术视为难题。

二十世纪九十年代,随着基于 PC 的客户/服务器计算模式和企业软件包的广泛采用,数据管理的变革基本完成。数据管理不再仅仅是存储和管理数据,而转变成用户所需要的各种数据管理的方式。Internet 的异军突起以及 XML 语言的出现,给数据库系统的发展开辟了一片新的天地。

1.2.4 操作系统的发展

操作系统使得对计算机的操作成为一件轻而易举的事。而最初的计算机并没有操作系统,人们通过各种操作按钮来控制计算机,后来出现了汇编语言。这些将语言内置的电脑只能由操作人员自己编写程序来运行,不利于设备、程序的共用。为了解决这种问题,就出现了操作系统,它能很好地实现程序的共用,以及对计算机硬件资源的管理。计算机操作系统的发展经历了两个阶段:第 1 个阶段为单用户、单任务的操作系统,以 CP/M、MS-DOS 等磁盘操作系统为代表;第 2 个阶段是多用户多任务和分时系统,以 UNIX、Windows、Linux 等操作系统为代表。

随着计算技术和大规模集成电路的发展,微型计算机迅速发展起来。从 20 世纪 70 年代中期开始出现了计算机操作系统。1976 年,正式推出 8 位操作系统 CP/M (Control Program/Monitor, 控制程序或监控程序)。CP/M 其实就是第一个微机操作系统,该系统允许用户通过控制台的键盘对系统进行控制和管理,其主要功能是对文件信息进行管理,以实现硬盘文件或其他设备文件的自动存取。

继 CP/M 操作系统之后,还出现了许多磁盘操作系统(DOS)。其中最成功的是微软的 MS-DOS,它是在 IBM-PC 及其兼容机上运行的操作系统,是单用户的操作系统。从 1981 年问世至今,MS-DOS 经历了 7 次大的版本升级,从 1.0 版到 7.0 版,不断地改进和完善。但是,MS-DOS 系统的单用户、单任务、字符界面和 16 位的大格局没有变化,因此它对于内存的管理也局限在 640 KB 的范围内。MS-DOS 最初是为 IBM-PC 开发的操作系统,因此它对硬件

平台的要求很低,因此 MS-DOS 系统既适合于高档微机使用,又适合于低档微机使用。MS-DOS 系统最大的一个优势是它支持众多的通用软件,如:各种语言处理程序、数据库管理系统、文字处理软件、电子表格等。

Windows 操作系统是 Microsoft 公司在 1985 年 11 月开始发布的窗口式多任务系统,它使微机进入了图形用户界面时代。Windows 已发表了多个版本,且大部分版本都有相应的中文版。1995 年,微软推出的 Windows 95 是个分水岭。在此之前的 Windows 都是由 DOS 引导的,也就是说它们还不是一个完全独立的系统,而 Windows 95 是一个完全独立的系统,并在很多方面做了进一步的改进,集成了网络功能和即插即用功能,是一个全新的 32 位操作系统。1998 年推出的 Windows 98 把微软的 Internet 浏览器技术整合到了 Windows 里面,使得访问 Internet 资源就像访问本地硬盘一样方便,从而更好地满足了人们越来越多地访问 Internet 资源的需要。之后出现的 Windows 2000、Windows XP、Windows 7.0、Windows 8.1 等版本,也都成功占领了微机市场。Windows 已发展成为一个界面友好,多用户、多任务,网络支持良好,具有出色的多媒体功能和众多的应用程序,硬件支持良好的操作系统。

UNIX 操作系统也是值得一提的操作系统。它并非指单一的操作软件,而是包括一系列的 UNIX 家族。它是一个真正的多用户分时系统,1969 年问世,最初是在中小型计算机上运用。目前已发展成为通用的操作系统,以其为基础形成的开放系统标准(如 POSIX)也是迄今为止唯一的操作系统标准。UNIX 系统的转折点是 1972 年到 1974 年,因为当时 UNIX 用 C 语言写成,把可移植性当成主要的设计目标。1988 年开放软件基金会成立后,UNIX 经历了一个辉煌的历程,成千上万的应用软件在 UNIX 系统上开发并用于几乎每个应用领域,UNIX 从此成为世界上用途最广的通用操作系统。UNIX 大大推动了计算机系统及软件技术的发展。UNIX 系统主要用于小型机、工作站和服务器,很少用于微型机,因为 UNIX 提供的文件共享方式涉及到不支持任何 Windows 或 Macintosh 操作系统的 NFS 或 DFS。

除此之外,还有 Linux 操作系统它是 1991 年,芬兰人 Linus Torvalds 开发出基于 UNIX 的网络操作系统 Linux,因此,Linux 也可以看作 UNIX 家族的成员。Linux 与 UNIX 在用户界面上完全兼容,而且在英特网上开放源代码,用户可根据自己的需要进行修改。Linux 操作系统是目前全球最大的一个自由软件,它是一个可与 UNIX 和 Windows 相媲美的操作系统。虽然,现在说 Linux 会取代 UNIX 和 Windows 还为时过早,但一个稳定性、灵活性和易用性都非常好的软件,肯定会得到越来越广泛的应用。

习题 1

1. 软件工程的发展经历了哪几个时期?
2. 数据库技术的发展经历了哪几个阶段?
3. 操作系统的发展历史中有哪些主要的操作系统?