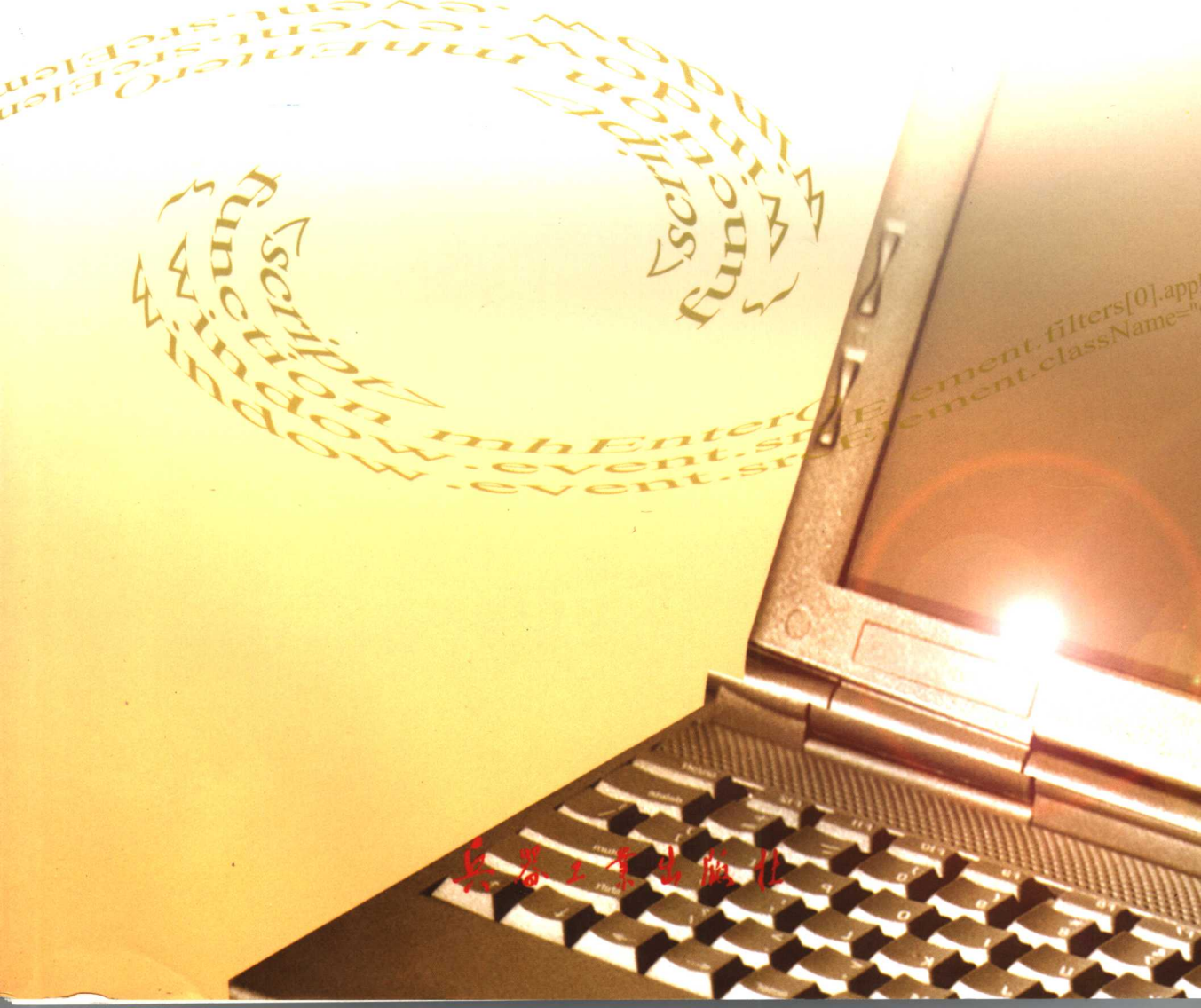
计算机软件技术基础

主编　周晓明　陈湘平



兵器工业出版社

计算机软件技术基础

主编　周晓明　陈湘平

兵器工业出版社

内 容 简 介

本书根据国家教育部颁发的《计算机软件技术基础课程教学大纲》编写。全书共分 16 章：第 1～14 章以 C 语言为主，首先介绍了计算机系统、软件组成、处理算法、C 程序基础等概念和基础知识，阐述了程序控制结构、数组、指针、结构与联合、函数和文件等内容；第 15 章为数据结构，重点介绍了线性表、树和图等常用数据结构，并介绍了数据查找的基本算法；第 16 章为软件工程，简单介绍了软件工程研究内容。书中每一章都配有习题。

本书结构合理，内容新颖，可读性强，适用于本科及以下层次各专业“计算机软件技术基础”、“程序设计基础”等课程的教学，亦可作为参加全国计算机等级考试考生的备考学习资料，还可供从事计算机研究和应用的人员参考。

图书在版编目（CIP）数据

计算机软件技术基础/周晓明，陈湘平主编. —北京：兵器工业出版社，2006.1

ISBN 7-80172-612-X

Ⅰ.计... Ⅱ.①周... ②陈... Ⅲ.软件—高等学校—教材 Ⅳ.TP31

中国版本图书馆 CIP 数据核字（2006）第 001133 号

出版发行：兵器工业出版社
发行电话：010-68962596, 68962591
邮　　编：100089
社　　址：北京市海淀区车道沟 10 号
经　　销：各地新华书店
印　　刷：北京市登峰印刷厂
版　　次：2006 年 1 月第 1 版第 1 次印刷
印　　数：1-2700

责任编辑：王　强
封面设计：天骄兄弟文化传播有限公司
责任校对：郭　芳
责任印制：赵春云
开　　本：787×1092　1/16
印　　张：20.5
字　　数：511 千字
定　　价：33.00 元

前　言

21 世纪是以计算机、网络通信和自动控制为基础的信息时代。以计算机技术为核心的信息技术的发展水平、应用水平和教学水平，已经成为衡量社会进步和文明程度的重要标志。面对挑战与机遇并存的发展形势，全球范围内的多层次、多侧面的计算机教育热潮正在蓬勃兴起。加强计算机基础教育，提高人们的计算机知识水平和操作能力，是培养跨世纪人才所必须具备的科学素质和基本技能。

本书紧紧围绕面向 21 世纪人才培养目标，根据国家教育部提出的在 2000 年前后使大多数高等院校的非计算机专业的计算机基础教学按照三个层次实施的基本要求编写而成。

全书内容包括：第 1~14 章以 C 语言为主，首先介绍了计算机系统、软件组成、处理算法、C 程序基础等概念和基础知识，阐述了程序控制结构、数组、指针、结构与联合、函数和文件等内容；第 15 章为数据结构，介绍了数据结构的基本概念，重点介绍了线性表、树和图等常用数据结构，并介绍了数据查找的基本算法；第 16 章为软件工程，简单介绍了软件工程研究内容。书中每一章都配有习题。

本书结构合理，内容新颖，涉及面广，可读性强。书中内容覆盖了国家教育部颁发的《计算机软件技术基础课程教学大纲》及《计算机二级考试大纲》所规定的要求，某些方面已有所超出。本书适合于各院校本科以下层次各专业的“计算机软件技术基础”、“程序设计基础”等课程的教学，亦可作为参加全国计算机等级考试考生的备考资料，还可供从事计算机研究和应用的人员参考。

本书在编写过程中，广泛征求了有关专家的意见和建议，并由长期从事计算机软件技术基础教学及应用工作的专业人员编写。本书由周晓明、陈湘平担任主编，郑锋、李宏权、魏文斌、陈莉、蔡莉、刘娟、李静参加了编写。编写后组织有关专家进行了审定，最后由主编修改定稿。

本书在编写出版过程中得到了各方面的支持和帮助，在此表示感谢。限于我们的水平，书中一定存在不少不足之处，敬请计算机教学与应用工作者和广大读者提出宝贵意见。

编者

2005 年 9 月

目 录

第 1 章 概 论 …… 1
1.1 计算机软件及其发展 …… 1
1.1.1 计算机软件的定义 …… 1
1.1.2 计算机软件的功能及分类 …… 1
1.1.3 计算机软件的发展 …… 2
1.1.4 软件工程学的基本概念 …… 3
1.2 程序设计 …… 4
1.2.1 程序设计语言 …… 5
1.2.2 语言处理程序 …… 6
1.2.3 程序设计应遵循的基本原则 …… 8
1.2.4 程序设计的基本过程 …… 9
1.2.5 程序设计风格 …… 10
习 题 1 …… 11
第 2 章 C 语言概述 …… 12
2.1 C 语言的发展背景 …… 12
2.2 C 语言的特点 …… 12
2.3 简单 C 语言程序介绍 …… 13
2.3.1 C 语言源程序的构成 …… 15
2.3.2 函数的一般结构 …… 15
2.3.3 C 语言程序的执行 …… 17
2.3.4 源程序书写格式 …… 17
2.4 C 语句概述 …… 18
2.4.1 控制语句 …… 18
2.4.2 函数调用语句 …… 18
2.4.3 表达式语句 …… 18
2.4.4 空语句 …… 18
2.4.5 复合语句 …… 19
2.5 C 程序的上机步骤 …… 19
2.5.1 运行一个 C 语言程序的一般过程 …… 19
2.5.2 TC 的启动、退出与命令菜单 …… 19
2.5.3 编辑并保存一个 C 语言源程序 …… 21
2.5.4 编译、连接——单个源程序文件 …… 21
2.5.5 运行与查看结果 …… 22
2.5.6 编辑下一个新的源程序 …… 22
习 题 2 …… 22
第 3 章 程序的灵魂——算法 …… 23

3.1 算法的概念及特性……23
3.2 算法的描述……25
3.2.1 自然语言表示……25
3.2.2 伪代码表示……25
3.2.3 传统流程图表示……26
3.2.4 N-S 结构化流程图……27
3.2.5 计算机语言表示……28
3.3 算法的设计……29
习 题 3……30
第 4 章 数据类型、运算符与表达式……31
4.1 C 语言的数据类型……31
4.2 常量与变量……32
4.2.1 常量和符号常量……32
4.2.2 变量……33
4.3 整型数据……35
4.3.1 整型常量……35
4.3.2 整型变量……35
4.4 实型数据……37
4.4.1 实型常量……37
4.4.2 实型变量……37
4.5 字符型数据……38
4.5.1 字符常量……38
4.5.2 字符变量……39
4.5.3 字符数据在内存中的存储形式及其使用方法……40
4.5.4 字符串常量……40
4.6 运算符与表达式……41
4.6.1 运算符与表达式概述……41
4.6.2 算术运算符和算术表达式……42
4.6.3 赋值运算……44
4.6.4 关系运算符与关系表达式……45
4.6.5 逻辑运算符与逻辑表达式……46
4.6.6 其他运算……47
4.6.7 不同类型数据间的类型转换……48
习 题 4……49
第 5 章 顺序结构程序设计……52
5.1 顺序结构程序概述……52
5.2 赋值语句……53
5.3 数据输入输出的概念……54
5.4 格式化输出函数 printf……55

5.5 格式化输入函数 scanf……56
5.6 字符输入和输出的函数……57
5.6.1 字符输出函数 putchar……57
5.6.2 字符输入函数 getchar()……58
5.7 顺序程序结构设计……58
习 题 5……59
第 6 章 选择结构程序设计……61
6.1 关系运算符和关系表达式……61
6.1.1 关系运算……61
6.1.2 关系表达式……62
6.2 逻辑运算符和逻辑表达式……62
6.2.1 逻辑运算……62
6.2.2 逻辑表达式……63
6.3 if 语句和条件运算符……64
6.3.1 if 语句……64
6.3.2 if 语句的嵌套……66
6.4 条件表达式构成的选择结构……68
6.4.1 条件表达式……68
6.4.2 条件表达式的运算优先级……69
6.4.3 使用条件表达式的注意事项……69
6.5 switch 语句……70
6.5.1 switch 语句的格式与执行过程……70
6.5.2 使用说明……71
6.6 选择结构程序举例……72
习 题 6……75
第 7 章 循环结构程序设计……77
7.1 循环结构及其执行过程……78
7.2 循环控制语句……78
7.2.1 while 语句……79
7.2.2 do-while 语句……80
7.2.3 for 语句……81
7.2.4 三种循环控制语句比较……82
7.3 循环结构程序设计……83
7.4 循环的嵌套……86
7.5 迭代和穷举算法在循环结构程序中的实现……88
7.5.1 迭代算法……88
7.5.2 穷举算法……88
7.6 break 和 continue 语句……89
7.6.1 break 语句……89

7.6.2 continue 语句 …… 90
习 题 7 …… 90
第 8 章 数 组 …… 93
8.1 一维数组的定义和引用 …… 93
8.1.1 一维数组的定义 …… 93
8.1.2 一维数组的初始化 …… 94
8.1.3 应用举例 …… 94
8.2 二维数组的定义和引用 …… 96
8.2.1 二维数组的定义 …… 96
8.2.2 二维数组的引用 …… 97
8.2.3 二维数组的初始化 …… 97
8.2.4 二维数组程序举例 …… 99
8.3 字符数组 …… 101
8.3.1 字符数组的定义 …… 101
8.3.2 字符数组的初始化 …… 101
8.3.3 字符数组的引用 …… 101
8.3.4 字符串 …… 102
8.3.5 字符数组的输入输出 …… 103
8.3.6 字符串处理函数 …… 104
8.3.7 字符数组应用举例 …… 107
习 题 8 …… 108
第 9 章 函 数 …… 109
9.1 概 述 …… 109
9.2 函数的定义 …… 111
9.2.1 函数定义形式 …… 111
9.2.2 函数定义的有关说明 …… 112
9.3 函数的类型、参数与返回值 …… 112
9.3.1 函数的参数：形式参数和实际参数 …… 112
9.3.2 函数的类型和返回值 …… 113
9.4 函数的调用 …… 115
9.4.1 函数调用的一般形式 …… 115
9.4.2 函数的调用方式 …… 117
9.4.3 函数调用的前提条件 …… 117
9.4.4 函数说明语句 …… 117
9.4.5 函数的嵌套调用 …… 119
9.4.6 函数的递归调用 …… 121
9.5 数组作为函数参数 …… 122
9.5.1 数组元素作为函数的实参 …… 122
9.5.2 一维数组名作为函数的参数 …… 123

9.5.3 多维数组名作为函数的参数 …… 125
9.6 变量及其存储类型 …… 126
9.6.1 变量的有效范围 …… 126
9.6.2 变量的存储类别 …… 129
9.6.3 内部函数和外部函数 …… 134
9.7 多文件程序的运行 …… 136
习 题 9 …… 136
第 10 章 预处理命令 …… 139
10.1 宏定义 …… 139
10.1.1 定义宏命令 …… 139
10.1.2 取消宏的命令 …… 144
10.2 文件包含命令 …… 144
10.3 条件编译命令 …… 146
10.3.1 格式一 …… 146
10.3.2 格式二 …… 148
10.3.3 格式三 …… 148
习 题 10 …… 150
第 11 章 指 针 …… 152
11.1 地址和指针的概念 …… 152
11.2 指针变量的定义及引用 …… 152
11.2.1 指针变量的定义 …… 152
11.2.2 指针变量的引用 …… 153
11.2.3 指针变量作为函数的参数 …… 154
11.3 指针与数组 …… 156
11.3.1 指向数组元素的指针 …… 156
11.3.2 通过指针引用数组元素 …… 156
11.3.3 用数组名（指针）作为函数的参数 …… 156
11.4 字符串与指针 …… 158
11.4.1 字符串的表示形式 …… 158
11.4.2 字符串指针作为函数参数 …… 159
11.5 指向函数的指针 …… 160
11.5.1 指向函数的指针变量的定义和使用 …… 160
11.5.2 指向函数的指针作为函数的参数 …… 161
11.6 返回指针值的函数 …… 163
11.7 main 函数的参数及其返回值 …… 165
习 题 11 …… 166
第 12 章 结构体与共用体 …… 169
12.1 结构体 …… 169
12.1.1 结构体概述 …… 169

12.1.2 结构体类型的定义 …… 169
12.1.3 结构体变量的定义 …… 170
12.1.4 结构体变量的引用 …… 172
12.1.5 结构体变量的初始化 …… 173
12.1.6 结构体数组 …… 174
12.2 指向结构体类型数据的指针 …… 176
12.2.1 指向结构体变量的指针 …… 176
12.2.2 指向结构体数组的指针 …… 177
12.2.3 用结构体变量和指向结构体的指针作为函数的参数 …… 178
12.3 用指针处理链表 …… 179
12.3.1 链表概述 …… 179
12.4 共用体 …… 181
12.4.1 共用体的概念 …… 181
12.4.2 共用体变量引用的特点 …… 181
12.5 枚举类型 …… 184
12.6 用 typedef 定义类型 …… 187
习 题 12 …… 187
第 13 章 位运算 …… 189
13.1 位运算符与位运算 …… 189
13.1.1 按位与运算 …… 189
13.1.2 按位或运算 …… 190
13.1.3 按位异或运算 …… 191
13.1.4 位取反运算 …… 192
13.1.5 位左移运算 …… 193
13.1.6 位右移运算 …… 193
13.1.7 位运算与赋值运算 …… 194
13.1.8 不同长度的数据进行位运算 …… 194
13.1.9 位运算示例 …… 194
13.2 位 段 …… 195
习 题 13 …… 198
第 14 章 文 件 …… 199
14.1 文件的基本概念 …… 199
14.1.1 操作系统的功能和类型简介 …… 199
14.1.2 文件的分类 …… 201
14.1.3 文件类型指针 …… 202
14.2 缓冲文件系统 …… 202
14.2.1 文件的打开、关闭 …… 202
14.2.2 文件的输入/输出操作 …… 205
14.3 非缓冲文件系统 …… 214

14.3.1 文件的打开和关闭 …… 214
14.3.2 读写函数 …… 215
14.3.3 随机定位函数 …… 215
14.3.4 程序举例 …… 216
习 题 14 …… 217
第 15 章 数据结构 …… 218
15.1 基本概念和术语 …… 218
15.2 线性表 …… 220
15.2.1 线性表的定义 …… 220
15.2.2 线性表的基本操作 …… 221
15.2.3 线性表的顺序存储结构 …… 222
15.2.4 线性表的链式表示与实现 …… 226
15.2.5 循环链表 …… 234
15.2.6 双向链表 …… 234
15.3 栈 …… 235
15.3.1 栈的定义 …… 235
15.3.2 栈的基本运算 …… 235
15.3.3 顺序栈 …… 235
15.3.4 链栈 …… 237
15.3.5 栈的应用 …… 238
15.4 队 列 …… 240
15.4.1 队列的定义 …… 240
15.4.2 队列的基本运算 …… 240
15.4.3 队列的链式表示和实现 …… 241
15.5 二叉树 …… 243
15.5.1 二叉树的基本概念 …… 243
15.5.2 二叉树的性质 …… 245
15.5.3 二叉树的基本操作 …… 245
15.5.4 顺序存储结构 …… 245
15.5.5 链式存储结构 …… 246
15.5.6 遍历二叉树 …… 246
15.6 图 …… 248
15.6.1 基本概念和术语 …… 248
15.6.2 图的基本操作 …… 250
15.6.3 图的存储结构 …… 251
15.7 查 找 …… 253
15.7.1 基本概念与术语 …… 253
15.7.2 查找表的存储结构 …… 254
15.7.3 顺序表的查找 …… 254

15.8 排 序 256
15.8.1 直接插入排序 257
15.8.2 冒泡排序(Bubble Sort) 258
15.9.3 简单选择排序 259
习 题 15 260
第 16 章 软件工程 266
16.1 发展简史 266
16.2 软件工程框架 266
16.2.1 软件工程目标 266
16.2.2 软件工程过程 267
16.2.3 软件工程原则 267
16.3 软件工程内容 267
16.3.1 软件开发模型 267
16.3.2 软件开发方法 267
16.3.3 软件过程 268
16.3.4 软件工具 268
16.3.5 软件开发环境 269
习 题 16 270
附录 1 C 语言中的关键字 271
附录 2 运算符和结合性 272
附录 3 Turbo C 的安装使用指南 273
一、Turbo C 的产生与发展 273
二、Turbo C 3.0 集成开发环境的使用 273
三、Turbo C 3.0 的配置文件 280
附录 4 Turbo C 编译错误信息 281
附录 5 Turbo C 库函数 292
附录 6 ASCII 码表 313
参考文献 316

第1章 概 论

1.1 计算机软件及其发展

1.1.1 计算机软件的定义

从功能上可以认为软件是人们利用计算机本身提供的逻辑功能，合理地组织计算机的工作，简化或代替人们在使用计算机过程中的各个环节，提供给用户的一个全球操作的工作环境。因此，不论是支撑计算机工作还是支持用户应用的程序都是软件，换句话说，计算机系统中的程序和有关的文档资料总称为软件。软件是一个计算机系统必不可少的组成部分，它保证计算机系统能有效地运行并提供给用户一个便于操作的工作环境。

在 20 世纪五六十年代曾认为计算机软件就是程序，如汇编程序、编译和解释程序、操作系统和支撑操作系统的各种管理程序、服务程序以及用户用各种程序设计语言编制的程序等。这一定义是基于手工方式进行软件开发而提出来的。即从设计、编程到调试均由个人独立完成，但要开发一个大型软件，特别是实用化、商品化、通用化的软件就碰到不少问题。手工方式不仅效率低、开发周期长，而且各个地区模块间的联系和接口很难协调，人的思维也很难胜任数以几万行甚至是几百万行计的程序，因而出错率高，维护工作量大，导致成本高。因此，在 20 世纪 60 年代末出现了软件危机。当时 IBM 公司 OS/360 系统负责人 Brooks 形象地形容这种手工方式开发软件是："……像巨兽在泥潭中作垂死挣扎，挣扎得越猛，泥浆就沾得越多，最后没有一个野兽能逃脱淹没在泥潭中的命运……程序设计就像是这样一个泥潭……一批批程序员在泥潭中挣扎……没有人料到问题会这样棘手。"软件设计者逐渐感到要有一定规范的文档以保证程序从设计、调试到运行的成功。这样，从 20 世纪 70 年代开始，人们就认为软件不仅仅是程序，而且还包括开发、使用、维护这些程序所需要的一切文档。

到了 20 世纪 80 年代，为了加强工程化、规范化，从软件工程的概念上更为全面地给软件下了定义，认为：计算机程序、实现此程序功能所采用的方法、规则以及与其相关联的文档和在机器上运行它所需要的数据都是计算机软件。计算机软件不断出现新产品，使得应用的灵活性和人机交互能力不断提高，充分体现了软件支撑计算机工作、扩大计算机系统功能的作用。

1.1.2 计算机软件的功能及分类

1.1.2.1 软件的功能

计算机系统中的程序和有关的文档资料总称为软件。软件是一个计算机系统必不可少的组成部分，它保证计算机系统能有效地运行并为用户提供特定的服务。

软件是计算机用户与硬件之间的接口，用户通过软件使用计算机，软件的主要功能有：

(1) 对计算机系统的硬件资源，提高各种资源的利用率；

(2) 在硬件基本功能的基础上，扩充计算机硬件的功能，提高硬件的使用效率；

(3) 提供对计算机硬件进行测试、诊断以及维护所需要的工具；

(4) 为专业人员进行开发应用提供必要的开发工具和开发环境，提高软件的开发效率；

(5) 向用户提供方便、灵活的使用界面；

(6) 提供完成特定应用的专用程序。

1.1.2.2 软件的分类

根据软件与计算机硬件和用户的关系，软件一般可分为系统软件、支持性软件和应用软件三类：

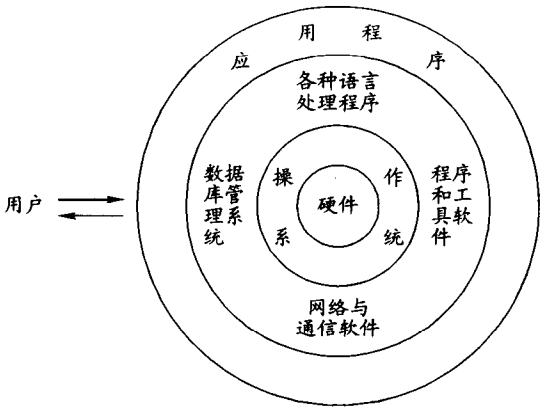


图 1.1 计算机系统中软件的层次

(1) 系统软件泛指为整个计算机系统配置的、不依赖于具体应用的软件。例如，操作系统、各种语言处理程序、数据库管理系统以及一些常用的实用软件等都属于系统软件。

(2) 支持性软件是指支持其他软件的开发和维护的软件。随着计算机应用的发展，软件开发和维护的代价越来越高，因此用于支持其他软件研制的软件显得特别重要。例如，20 世纪 70 年代中期和后期发展起来的程序设计环境和软件工程环境可看做是支持性软件的代表，主要包括环境信息库、各种接口软件和工具软件等。

(3) 应用软件是指用于特定应用领域的专用软件，它又可以分为两类：一类是指为解决某一具体应用，按用户的特定需要而定制的软件；另一类是可以适合多种不同领域的、通用性的应用软件。如文字处理、图形绘制、财务管理、报表处理等方面的软件。

软件是用户与计算机系统中硬件之间的桥梁，在用户到系统硬件之间有不同类型、处于不同层次的软件。软件的层次关系如图 1.1 所示。

1.1.3 计算机软件的发展

自 1946 年世界上出现第一台电子数字计算机以来，计算机系统得到飞速发展。计算机软件随着硬件的发展而发展，计算机硬件随着元器件的发展而发展。人们习惯按组成计算机元件的演变来划分计算机的发展阶段，它大致可以分为四个阶段，相应地软件也可以分为四个阶段。

第一代为电子管计算机时代（1946～1957 年）。这一阶段计算机的主要逻辑元件是电子管，存储器为水银延迟线或静电屏；软件方面，发展了机器语言、汇编语言。

第二代为晶体管计算机时代（1958～1964 年）。这一阶段计算机的主要逻辑元件是晶体管，存储器为由磁芯组成；软件方面，发展了监控程序、高级语言编译等。

第三代为中、小规模集成电路计算机时代（1965～1970 年）。这一阶段计算机的主要逻辑元件是中、小规模集成电路；软件方面，发展了多道程序、实时处理程序和操作系统的形成等。

第四代为大规模和超大规模集成电路计算机时代（自 1971 年开始）。这一阶段计算机的主要特征是以大规模集成电路 LSI(Large Scale Integration)和超大规模集成电路 VLSI(Very Large Scale Integration)为主要功能部件；软件方面，发展了数据库系统、多媒体技术、通信软件、分布式操作系统、网络操作系统、软件产业等。这一时期出现了许多不同类型的大、中、小型计算机和巨型计算机系统，特别是 20 世纪 80 年代以来，个人计算机异军突起。而进入 20 世纪 90 年代以后，计算机网络有了极大的发展，促使计算机应用领域向纵深挺进，使用日益广泛。

尽管人们早已习惯谈论第五代、第六代计算机了，但学术界认为不要再沿用第五代计算机的说法为好，而赞成用新一代计算机或未来型计算机来称呼可能出现的新事物。一些专家认为，新一代的计算机系统主要着眼于机器的智能化，它以知识库为基础，采用智能接口，能进行逻辑推理，完成判断和决策任务。它可以模拟或部分替代人的智能活动，并具有自然的人机通信能力。事实上，对于什么是新一代计算机仍存在着不同的观点和看法。

计算机程序设计语言的发展是随着计算机技术的不断发展及广泛应用而逐步丰富与完善起来的。计算机软件也有标志其发展的 4 个阶段，即：汇编语言的出现、高级语言的出现、操作系统的形成和计算机网络软件的出现。随着计算机应用领域不断向纵深发展，由辅助人们进行科学计算、实时控制、数据处理等发展到辅助设计、信息管理，辅助人们进行思维、决策；由单机系统发展到联机系统、分布计算机系统和计算机网络系统等。

计算机网络软件是计算机技术和通信技术二者高度发展和密切结合的结果。从某种意义上讲，它是更高意义上的操作系统。它利用通信线路把分布在不同地点上的多个独立的计算机系统连接起来，使网上用户可以实现数据传送、共享网络上的所有硬件、软件和数据等资源，并且提高了计算机的可靠性，均衡了网络中各种计算机的负载情况，同时便于系统的扩展。

1.1.4 软件工程学的基本概念

在 20 世纪五六十年代开发大型系统软件要用手工方式进行，其生产效率低、出错率高。例如，IBM 公司的 OS/360 操作系统用了 5000 人年开发，软件产品每个版本均有 1000 个大大小小的错误。这种状态不能满足日益增长的软件生产的需要，产生了下述几个方面的问题。

1. 软件复杂性飞速增长

个体思维不能胜任一个需上千人年甚至几千人年开发的大型软件，这种软件程序量是以几万行或百万行计数的，而且相互关联，造成错误的概率非常高。

2. 成本极高

由于硬件迅速发展，集成度大大提高，出现硬件成本下降，靠落后开发方式编制软件的工作量逐步增长，人工费用也迅速增大。所造成的软件、硬件投资比率的变化，美国有人做了一个统计，如表 1.1 所示。可见软件投资增长之快。

表 1.1 软件投资统计

年份	软件投资占总投资的百分比
1955 年	20%以下
1970 年	60%左右
1980 年	85%以上

3. 开发周期长

手工方式开发的程序，所需人年数是随程序代码行的上升按指数曲线增长的，造成大型

软件开发周期长。

4. 维护工作量大

对软件错误的统计和分析表明，由于设计产生的错误占 70%，而由于编码产生的错误仅占 30%，因此，保证设计过程中软件的正确性且易于检验，是一个亟待解决的问题。此外，当用户需求有变化、硬件设备更新后或操作系统有新的版本出现时，都需要修改程序，以适应新的环境，这使得软件的维护工作耗费相当的人力和资源。20 世纪 50 年代衡量一个软件的质量是看其是否节省存储空间、运行速度是否足够快，而到 20 世纪 80 年代衡量的标准是软件正确、易读、易修改、易测试、有完整的文档资料等。

由于上述原因，出现了“软件危机”，于是在 1968 年北大西洋公约组织的计算机软件学术会议上第一次提出“软件工程”这个词。考虑到研制一个软件系统同研制一台机器或建造一座楼房有许多共同之处，因此参考机械工程、建筑工程中的一些技术来指导软件的研制，像处理其他工程一样来处理软件研制的全过程，形成了软件学科中的一门新学科——软件工程。软件工程强调使用生命周期方法学和各种结构化分析与设计技术，用“系统”的观点来分解问题，然后再分别解决各个子问题。

生命周期方法学是从时间角度对软件开发和维护的复杂问题进行分解，且分成若干个阶段，每个阶段有相对独立的任务，前一个阶段任务的完成是开始进行后一阶段工作的前提和基础，而后一阶段任务的完成是对前一阶段提出的问题给予解决的具体化。每一阶段的开始和结束都有严格标准，因而，前一阶段的结束标准就是后一阶段的开始标准，而且每一阶段结束之前都必须进行正式严格的技术审查和管理审查，审查的一条主要标准就是每个阶段都应该交出高质量的文档资料。文档资料是通信的工具，清楚准确地说明了任务是如何完成的、完成的情况以及下一步工作是什么等等；文档资料也起备忘录的作用。软件生命周期一般分为问题定义和可行性论证时期、系统分析时期、系统设计时期、编码和单元测试时期、系统测试时期与软件维护时期等六个阶段。也有人将其分为三个阶段：软件定义时期（包括问题定义、可行性论证和系统分析）；软件开发时期（包括系统初步设计、系统详细设计、编码、单元测试和综合测试）；软件维护时期（包括修改、更新等活动）。我们将六个阶段的任务及结果文档用表 1.2 列出。

表 1.2 软件生命周期

阶段	任务	结果及文档
问题定义和可行性论证	明确问题是什么，有可行的解否	调研报告、可行性论证报告
系统分析	确定目标系统的功能和信息	目标系统功能模型、信息模型
系统设计	确定系统实现方案及软件模块功能、信息关系	软件系统结构图
编码和单元测试	按规定的语言写出各模块程序并通过单元测试	程序清单及说明、单元测试结果及分析
系统测试	完成符合功能要求的软件产品	测试说明书
软件维护	改进系统	维护记录、修改说明书

1.2 程序设计

程序设计的工作是由程序员完成的。因此，程序员必须了解程序设计环境（程序设计所需的软件与硬件支持）、程序设计的基本过程、程序设计风格以及一些有关程序运行和测试的知识，否则难以独立完成程序设计的全部工作。

1.2.1 程序设计语言

语言是人类用于信息交流的工具，计算机语言是人与计算机交换信息的中间媒介和工具。人们使用计算机语言，告诉计算机“做什么”，“怎么做”——即用计算机可识别的语言编写程序，计算机则根据程序的规定进行处理，并将处理结果告诉人。显然，学习计算机语言就是为了更好地利用计算机为我们服务。

如前所述，人们要利用计算机解决实际问题一般要编制程序。所谓程序就是指令的有序集合。我们通过这些指令告诉计算机“做什么”，“怎么做”。不同的环境、不同的用途要用不同的编程语言。程序设计语言就是指编写程序所使用的语言。它是人们与计算机之间交换信息的工具。实际上它也是人们指挥计算机工作的工具。通常，用户在用程序设计语言编写程序时，必须满足相应语言的文法格式，并且逻辑要正确。只有这样，计算机才能根据程序中的指令作出相应的动作，最后完成用户所要求完成的各项工作。程序设计语言是软件系统的重要组成部分。从计算机的诞生至今，人们已设计出了许多程序设计语言，但只有极少部分得到比较广泛的应用，最常见的分类方法是根据程序设计语言与计算机硬件的联系将其分为机器语言、汇编语言和高级语言三类。

1.2.1.1 机器语言

对计算机来说，它只能理解和执行以“0”和“1”组成的命令。由“0”和“1”组成的二进制编码表示的命令，称为机器指令。机器指令由操作码和操作数组成，其具体的表现形式和功能与计算机系统的结构相关联。

机器语言是直接用这种机器指令的集合作为程序设计手段的语言。机器语言是最底层的计算机语言。用机器语言编写的程序是计算机能直接理解和执行的。因此，它的执行速度比较快，基本上充分发挥了计算机的速度性能。对于不同的计算机硬件（主要是CPU），其机器语言一般是不相同的。每个计算机都有自己的指令集，所谓指令是一种规定CPU执行某种特定操作的命令，也称机器指令。通常一条指令对应于一种基本操作，每台计算机的指令系统就是该机器的机器语言。

由于不同的计算机其机器语言有所不同，因此，针对一种计算机所编写的机器语言程序，一般不能在另一种计算机上运行。用机器语言编写程序的难度较大，修改调试也不方便，容易出错，其程序的直观性比较差，且不易移植。

机器语言的特点：难记忆；编程困难；直接面向机器，其程序的可移植性差。

1.2.1.2 汇编语言

为了便于理解和记忆，人们采用能帮助记忆的英文缩写符号（称为指令助记符）来代替机器语言指令代码中的操作码，用地址符号来代替地址码。用指令助记符和地址符号书写的指令称为汇编指令。

汇编语言就是用这种汇编指令的集合作为程序设计手段的语言。汇编语言又称符号语言。由于汇编语言采用了助记符，因此，它比机器语言直观，容易记忆和理解，用汇编语言编写的程序也比机器语言程序易读、易检查、易修改。汇编语言与机器语言一般是一一对应的。因此，对于不同的计算机，汇编语言源程序是互不通用的。用汇编语言编写的程序执行效率比较高，但通用性与可移植性仍然比较差。

总的来说，汇编语言比机器语言前进了一步。但是，计算机不能直接识别用汇编语言编