



WUTP

普通高等学校
自动化类专业新编系列教材

Foundations Of Software Technique

软件技术基础

钟珞 杨波 主编

武汉理工大学出版社

普通高等学校自动化类专业新编系列教材

软件技术基础

主编 钟 珞 杨 波

武汉理工大学出版社

· 武汉 ·

内 容 提 要

本书首先从软件应用的基础知识出发,详细介绍了C语言、算法与数据结构、软件工程、操作系统和数据库系统等方面的基础理论知识。其中,第四章用VC++编写了一个实例进行演示,第七章是一个以VC++编写的自动化方面的实例。

本书内容丰富、论述严谨、重点突出,体系结构和内容选取强调基础性和实用性,符合理工科学生的认识规律,可作为电气工程与自动化专业及其有关非计算机专业应试人员的教材,也可供大专院校师生、科技人员及电脑爱好者阅读。

图书在版编目(CIP)数据

软件技术基础/钟珞,杨波主编. —武汉:武汉理工大学出版社,2001.8

普通高等学校自动化类专业新编系列教材

ISBN 7-5629-1719-1

I. 软…

II. ①钟… ②杨…

III. 软件-高等学校-教材

IV. TP31

出版发行:武汉理工大学出版社

武汉市武昌珞狮路122号 邮编:430070

HTTP://www.whut.edu.cn/chubanl

E-mail:wutp@public.wh.hb.cn

经 销 者:各地新华书店

印 刷 者:武汉市科普教育印刷厂

开 本:787×1092 1/16

印 张:23

字 数:568千字

版 次:2001年8月第1版

印 次:2001年8月第1次印刷

印 数:1—5000册

定 价:29.50元

凡购本书,如有缺页、倒页、脱页等印装质量问题,请向出版社发行部调换

本社购书热线电话:(027)87397097 87394412

普通高等学校自动化类专业新编系列教材

出版说明

世纪之交,我国高等学校的人才培养工作正处在一个关键的历史时期。为了适应我国改革开放和社会主义现代化建设特别是社会主义市场经济体制对高等教育人才培养工作的新要求,为了适应世界科学技术发展的新趋势和新特点,原国家教育委员会组织对普通高等学校本科专业目录进行了第四次全面修订,并于1998年7月由教育部正式颁布实施。修订后的专业目录中,自动化类专业的专业面大大拓宽,相应的专业培养目标、业务培养要求、主干学科、主要课程、主要实践性教学环节等都有了不同程度的变化。要适应新的专业培养目标和教学要求,组织一套新的自动化类专业系列教材就成了当务之急。为此,武汉理工大学出版社在广泛调研的基础上,组织国内近30所大学的近100位教授共同编写了这套系列教材。

本套教材定位于普通高等学校自动化类专业本科层次,遵照教育部颁发的《普通高等学校本科专业介绍》中所提出的培养目标和培养要求,依据2000年5月全国23所高等院校的70多位专家教授在武汉共同确定的指导思想和编写大纲进行编写,具有如下特点:

观念新——主动适应教学改革的需要和市场经济对人才培养的要求;

内容新——自动化技术在近20年来进展巨大,并与计算机技术、航空航天技术、建筑工程、生物工程、社会科学(社会系统与经济系统)联系越来越紧密,这套教材尽可能反映了这些内容,以适应21世纪自动化与控制工程人才的培养要求;

体系新——在以前的基础上重构和重组,而非重建。各门课程及内容的组成、顺序、比例更加优化,避免遗漏和不必要的重复;

与国际接轨——自动化类专业教育要面向世界,面向未来,面向区域经济。在借鉴发达国家高等教育的专业模式和课程设置的同时,适当兼顾当前各地区经济文化发展不平衡的现状;

教学手段现代化——本套教材力求具有网络化、电子化、数字化的特色,大力推进电子讲稿和多媒体课件的出版工作。

本系列教材是在21世纪初推出的目前系统优化、品种较全、作者阵容最强的一套普通高等学校自动化类(本科)系列教材。我们将高度重视,兢兢业业,保证质量,恳请选用本套教材的广大师生在使用过程中给我们多提意见和建议,以便我们不断修订、补充、完善全套教材。

21世纪已经到来,知识经济的曙光已经初现。面向新世纪的中国高等教育正在经历前所未有的变革和发展,人文与理工相通,科学与技术相融,教学与研究并重,知识与智慧同尊,以培养社会经济文化发展所需要的复合型人才,这是我国建立知识创新体系的重大挑战和空前机遇。我社愿与各位专家、读者真诚合作,共同努力,为新世纪的中国高等教育事业做出更大的贡献。

武汉理工大学出版社

2001年8月

普通高等学校自动化类专业新编系列教材

编审委员会

顾问：

郑大钟 熊有伦 戴冠中 萧德云 陈伯时 周祖德
项国波 席裕庚 褚 健

主任委员：

萧蕴诗 张崇巍 陈大钦 吴 坚 陈福祥 高鸣涵

委员(按姓氏笔画顺序)：

马建国	王 辉	王孝武	王明阳	王建华	王俊杰
文 方	方康玲	卢京潮	龙 伟	申功璋	叶春生
全书海	吕 锋	刘 泉	刘涤尘	刘京南	李汉强
李磊民	宋靖雁	林 都	林 辉	林锦国	杨 波
杨天怡	杨家本	周泽义	胡 超	赵英凯	赵曾贻
侯朝桢	钟 珞	须文波	翁维勤	夏承铨	郭圣权
徐科军	黄席樾	章卫国	彭容修	程耕国	温阳东
曾庆军	谢克明	熊前兴	黎明森	戴文进	

编委会秘书：

黄 春

总责任编辑：

杨学忠 徐秋林

前言

为尽快实现教育部提出的 21 世纪计算机基础教育要上一个新台阶的宏伟目标,落实高等院校在计算机基础教学上达到三个层次的基础要求,根据教育部颁发的新专业目录及面向 21 世纪的自动化类专业培养规格及培养方案,由多所高校的学科带头人和骨干教师组织编写了本教材,其内容丰富、系统、完整,凝聚了多年教学经验和智慧。

为了满足各种水平读者的需要,本书内容上力求由浅入深,通俗易懂,简明扼要,注重实用技术。从计算机软件应用的基础知识、基本概念入手,介绍了 C 语言、数据结构与算法、软件工程、操作系统和数据库系统等方面的基础理论知识。本书是一本可作为电气工程与自动化专业及有关非计算机专业应用人员的优秀教材,同时也可供高等院校师生、科技人员及电脑爱好者阅读。

全书分七章。第一章是计算机软件概述,详细介绍了计算机软件与软件工程的基本概念和基本原理,同时介绍了计算机开发的原则与方法;第二章是高级程序设计语言基础,以 C 语言的基本应用为基础,介绍了 C 语言的基本组成和基本结构,主要包括变量及常量、数据结构、函数、指针、基本的 C 语言设计方法、基于 C 语言的结构化程序设计方法;第三章是算法与数据结构,主要讲述算法与数据结构的基本概念及常用的典型数据结构与算法,包括链表、队列、堆栈等线性数据结构,二叉树、哈夫曼树等树型数据结构和简单的图型数据结构。在算法方面,结合数据结构讲述了检索与排序算法。最后针对控制中常用的滤波、PID 控制给出了相应的算法;第四章是软件工程基础,介绍了系统分析与软件项目开发计划、需求分析、软件设计、编码、软件测试、面向对象方法和软件维护等内容,并给出了实例演示;第五章是操作系统,主要介绍 Windows 操作系统,其中重点介绍 Windows98 操作系统的五大管理功能:存储管理、进程管理、作业管理、设备管理与文件管理。对 Windows2000 也作了简要的介绍;第六章是数据库系统,主要介绍数据库的基本概念与技术,包括数据库的基础知识、数据库的三个模型、关系数据库的基本理论、实时数据库简介、数据库应用开发的基本技术。重点就关系数据库应用和应用开发进行了介绍,并根据数据库的应用给出了实例;第七章是实验部分(PC 机与全数字变频调速器的 C 语言通信编程),用 VC++ 编程实现了 PC 机与全数字变频调速器之间的通信。本书的七章内容是计算机软件知识的基础部分,通过对本书的学习,读者会对计算机软件有一个清楚的认识。

本书由武汉理工大学钟珞教授、济南大学杨波教授主编,济南大学曲守宁副教授、武汉理工大学袁佑新副教授及武汉理工大学吕品、郑巧仙、徐齐刚、阎京生、李辉等同志参加编写。

由于计算机软件技术发展很快,加之我们的水平和收集的资料有限,难免有不足之处。殷切希望得到广大同仁和读者的批评指正,以使本教材的质量得到进一步提高。

编者

2000 年 12 月

目 录

1 计算机软件概论	(1)
1.1 计算机软件	(1)
1.1.1 计算机系统资源	(1)
1.1.2 软件的概念与特点	(2)
1.1.3 软件的分类	(4)
1.1.4 软件的发展和软件危机	(7)
1.2 软件工程基本概念	(8)
1.2.1 软件工程学概念的提出	(8)
1.2.2 软件工程的定义	(9)
1.2.3 软件工程项目的目标	(10)
1.2.4 软件工程的周期	(10)
1.3 应用软件开发	(12)
1.3.1 应用软件开发的原则和方法	(12)
1.3.2 结构化分析方法	(14)
1.3.3 结构化设计方法	(18)
1.3.4 结构化编程方法	(23)
1.3.5 测试方法	(26)
习题	(28)
2 高级程序设计语言基础	(29)
2.1 高级程序设计语言概述	(29)
2.1.1 程序设计语言的发展	(29)
2.1.2 C语言出现的历史背景	(30)
2.1.3 C语言的特点	(31)
2.1.4 简单的C程序介绍	(31)
2.1.5 C的基本数据类型与运算	(33)
2.1.6 C语言中的语句概述	(40)
2.1.7 终端上的数据输入与输出	(40)
2.2 C语言的基本结构及程序设计	(42)
2.2.1 C程序的控制结构	(42)
2.2.2 模块化程序设计	(50)
2.3 指针与函数	(60)
2.3.1 指针的概念	(60)
2.3.2 指针变量作函数的参数	(61)

2.3.3	函数的指针与函数的调用	(62)
2.3.4	返回指针值的函数	(63)
2.4	数据类型	(63)
2.4.1	构造类型和指针	(63)
2.4.2	结构体和指针	(73)
2.4.3	共用体	(77)
2.5	编译预处理	(78)
2.5.1	宏定义	(78)
2.5.2	参数的宏定义	(79)
2.5.3	“文件包含”处理	(80)
2.5.4	条件编译	(81)
2.6	应用举例	(82)
	习题	(86)
3	算法与数据结构	(87)
3.1	算法与数据结构概述	(87)
3.1.1	数据、数据元素及数据对象	(87)
3.1.2	数据结构	(87)
3.1.3	数据类型	(88)
3.1.4	算法	(88)
3.2	线性数据结构	(89)
3.2.1	线性表定义、特征和运算	(90)
3.2.2	线性表的顺序存储结构	(91)
3.2.3	线性表的链式存储结构	(93)
3.2.4	栈和队列	(99)
3.3	树型数据结构	(105)
3.3.1	树的概念	(105)
3.3.2	二叉树	(106)
3.3.3	哈夫曼树	(111)
3.4	图形数据结构	(113)
3.4.1	图的概念及术语	(113)
3.4.2	图的存储结构	(114)
3.4.3	图的遍历	(117)
3.5	检索与内部排序	(119)
3.5.1	检索	(119)
3.5.2	内部排序	(125)
3.6	应用实例	(136)
3.6.1	离散系统中的数字滤波	(136)
3.6.2	PID 算法	(137)

习题	(139)
4 软件工程基础	(141)
4.1 系统分析和软件项目计划	(141)
4.1.1 系统分析	(141)
4.1.2 系统需求识别	(141)
4.1.3 可行性分析	(142)
4.1.4 编制项目开发计划	(143)
4.2 需求分析	(144)
4.2.1 确定对系统的综合要求	(144)
4.2.2 分析系统的数据要求	(144)
4.2.3 导出系统的数据模型	(145)
4.2.4 修正系统开发计划	(145)
4.2.5 开发原型系统	(145)
4.2.6 分析过程	(146)
4.2.7 概念模型和规范化	(148)
4.2.8 图形工具	(150)
4.3 软件设计	(152)
4.3.1 软件设计概念和原则	(152)
4.3.2 详细设计	(155)
4.3.3 详细设计的工具	(163)
4.4 编码	(167)
4.4.1 程序设计语言分类	(167)
4.4.2 程序设计语言的特点	(169)
4.4.3 选择一种语言	(171)
4.4.4 写程序的风格	(172)
4.5 软件测试	(174)
4.5.1 测试的基本概念	(175)
4.5.2 测试步骤	(176)
4.5.3 白盒测试的测试用例设计	(177)
4.5.4 黑盒测试的测试用例设计	(180)
4.6 面向对象方法	(183)
4.6.1 面向对象的基本概念	(183)
4.6.2 面向对象建模	(185)
4.6.3 面向对象的分析与设计	(194)
4.7 软件维护	(197)
4.7.1 软件维护的定义	(197)
4.7.2 软件维护的特点	(198)
4.7.3 软件维护的过程	(199)

4.7.4 软件的可维护性	(202)
4.8 编程实例	(204)
4.8.1 电力电子变流技术软件	(204)
4.8.2 控制理论基础软件	(209)
4.8.3 感应电动机机械特性曲线	(217)
习题	(223)
5 操作系统	(224)
5.1 操作系统概述	(224)
5.1.1 什么叫操作系统	(224)
5.1.2 操作系统的基本类型	(225)
5.1.3 操作系统的功能	(227)
5.1.4 操作系统的特性	(228)
5.1.5 操作系统的硬件环境	(228)
5.2 Windows 操作系统的组成和功能服务	(230)
5.2.1 Windows 操作系统的简介和发展	(230)
5.2.2 Windows98 的组成	(230)
5.2.3 Windows98 功能服务	(231)
5.2.4 Windows2000 的新特性	(234)
5.3 Windows 操作系统的存储管理	(237)
5.3.1 Windows98 的存储管理简介	(237)
5.3.2 86 系列 CPU 内存工作模式	(237)
5.3.3 常规存储管理、XMS 和 EMS 规范	(241)
5.3.4 Windows98 的虚拟存储管理	(242)
5.3.5 Windows2000 的虚拟内存管理和内存管理	(243)
5.4 Windows 操作系统的进程管理	(244)
5.4.1 Windows98 的进程管理	(244)
5.4.2 Windows98 的多任务原理	(246)
5.4.3 Windows2000 的进程和线程管理	(247)
5.5 Windows 操作系统的作业管理	(248)
5.5.1 Windows98 的作业管理	(248)
5.5.2 Windows98 的应用程序接口	(248)
5.5.3 OLE 组件模型的基本原理	(250)
5.5.4 Windows98 的外壳 Shell	(251)
5.5.5 Windows98 的任务调度	(252)
5.5.6 Windows2000 的 COM 和 OLE 技术	(253)
5.6 Windows 操作系统的设备管理	(253)
5.6.1 Windows98 设备管理的特点	(253)
5.6.2 Windows98 的 32 位驱动程序模型 WDM	(254)

5.6.3	Windows98 的电源管理	(255)
5.6.4	Windows98 的即插即用管理	(255)
5.6.5	Windows2000 的即插即用与电源管理功能	(257)
5.7	Windows 操作系统的文件管理	(259)
5.7.1	Windows98 文件系统的结构	(259)
5.7.2	Windows98 FAT、VFAT、FAT32、CDFS 文件系统	(260)
5.7.3	Windows98 的文件名	(261)
5.7.4	Windows2000 的文件系统	(261)
	习题	(263)
6	数据库系统	(264)
6.1	数据库系统概述	(264)
6.1.1	数据库管理技术的产生和发展	(264)
6.1.2	数据库基本概念	(265)
6.1.3	数据模型	(265)
6.1.4	常用的数据模型	(268)
6.1.5	数据库系统结构	(274)
6.2	关系数据库理论基础	(277)
6.2.1	关系模式的存储异常与数据依赖	(277)
6.2.2	函数依赖的基本概念	(278)
6.2.3	关系的规范化	(280)
6.2.4	多值依赖	(283)
6.2.5	模式分解的优劣	(286)
6.3	实时数据库简介	(287)
6.3.1	实时数据库的发展	(288)
6.3.2	实时数据库与时间	(288)
6.3.3	实时数据库系统的主要技术	(290)
6.3.4	实时数据库应用与研究	(291)
6.4	数据库设计	(291)
6.4.1	规划	(292)
6.4.2	需求分析	(292)
6.4.3	概念结构设计	(294)
6.4.4	逻辑结构设计	(295)
6.4.5	数据库物理设计	(295)
6.4.6	数据库实现	(297)
6.4.7	数据库的运行和维护	(297)
6.5	微机数据库系统简介	(298)
6.6	数据库管理系统	(300)
6.6.1	DBMS 基本功能	(300)

6.6.2 DBMS 的组成	(301)
6.6.3 DBMS 的评价与选择	(303)
6.7 应用实例	(304)
习题	(306)
7 PC 机与全数字变频调速器的 C 语言通信软件	(307)
7.1 C 语言的通信功能调用	(307)
7.1.1 C 语言的 BIOS 功能调用	(307)
7.1.2 用 C 语言直接驱动通信硬件	(308)
7.2 SANKEN 系列变频器的通信协议与传送方式	(308)
7.2.1 ISI 的功能	(308)
7.2.2 电文格式	(310)
7.2.3 电文构成详解	(313)
7.2.4 变频器运转状态数据及控制端子台输入状态数据	(314)
7.2.5 ISI 与计算机间的电文收发方法	(315)
7.3 通信口的初始化	(317)
7.3.1 ISI 通信单元的初始化	(317)
7.3.2 计算机串行口的初始化	(318)
7.4 转速开环变频调速系统编程	(319)
7.4.1 转速开环变频调速系统	(319)
7.4.2 程序设计流程图	(319)
7.4.3 变频器原始数据库的建立	(320)
7.4.4 人机界面程序设计	(324)
7.4.5 底层通讯的编程实现	(342)
附录 A 矢量控制变频调速系统实验	(348)
附录 B 转速闭环变频调速系统实验	(350)
参考文献	(354)

1 计算机软件概论

本章知识要点

1. 计算机软件资源,计算机系统的组成,软件的概念与特点,软件的种类;
2. 软件工程的基本概念,软件工程项目的基本目标,软件工程的生命周期;
3. 应用软件开发的原理和方法,结构化分析方法、设计方法、编程方法与测试方法。

1.1 计算机软件

1.1.1 计算机系统资源

计算机系统资源是指计算机硬件和计算机软件资源。

计算机硬件资源包括中央处理机(简称 CPU)、主存储器 and 显示终端、键盘、打印机、磁带机、磁盘机等外部设备。

计算机软件资源一般认为包括以下几个部分:

- 汇编程序
- 各种高级语言
- 各种语言的编译或解释程序
- 各种标准程序库
- 操作系统
- 数据库系统软件
- 计算机网络软件
- 各种应用程序

计算机硬件和软件组成了计算机使用的统一整体,因而统称为计算机系统。对于使用计算机的人来讲,面对的是以中央处理机和外围设备为物质基础并含有计算机软件的系统。计算机系统使用得好不好,不仅仅指物质基础使用是否正确、运行是否可靠,而且指软件掌握得如何、发挥了多少效用,而后者往往是更为关键的一个方面。

例如以下一些问题没有合适的计算机软件是无法解决的。

1. 多道程序的处理问题。如果内存中保持有多道程序,如何实现交替运行,以便有效地利用 CPU 和外设。软件应解决什么时候把 CPU 分配给某道程序占用,每个程序和其执行的中间结果,最后结果如何存放才不致造成混乱,又如何将各道程序执行结果打印输出。

2. 多个用户使用一台计算机问题。一台计算机挂上若干个终端,每一个终端被一个用户使用,多个用户有效地使用 CPU 和外设的问题,也要依靠计算机软件才能使用。

3. 当某种高级语言编制的应用程序需要占用内存的容量大于内存可用空间,那么这个应用程序是否可以正常运行;如何在系统软件的支撑下,利用计算机的外存来保证应用程序的执行。

4. 多台计算机联成计算机网络,建立通信、扩大功能,提高可靠性,这也是靠计算机硬件和软件共同实现的。

诸如此类的问题很多,因此当讲到计算机资源时应既考虑硬件资源又考虑软件资源。

1.1.2 软件的概念与特点

软件是计算机系统中与硬件相互依存的另一部分,它是包括程序、数据及其相关文档的完整集合。其中,程序是按事先设计的功能和性能要求执行的指令序列;数据是使程序能正确操纵信息的数据结构;文档是与程序开发、维护和使用有关的图文材料。

在 20 世纪五六十年代就曾认为计算机软件即程序,软件就是指包括汇编程序、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高级语言编制的程序等。

这个定义是基于手工方式进行软件开发而提出来的,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。不仅效率低、开发周期长,而且各个模块间的联系和接口很难协调,人的思维也很难胜任以几万条或百万行计数的程序,因而出错率高,维护工作量大,导致成本高,在 60 年代末出现了软件危机。

软件设计者逐渐感到要有一定规范的文档才能保证程序从设计、调试到运行的成功。这样,从 70 年代开始,认为软件不仅是程序,还包括开发、使用、维护这些程序所需的一切文档。

到了 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: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。

为了能全面、正确地理解计算机和软件,必须了解软件的特点。

1. 软件是一种逻辑实体,不是具体的物理实体。它具有抽象性。我们可以把它记录在纸面上,保存在计算机的存储器内部,也可以保留在磁盘、磁带和可重写光盘上,但我们却无法看到软件本身的形态,而必须通过观察、分析、思考、判断,去了解它的功能、性能及其它特性。

2. 软件的生产与硬件不同,在它的开发过程中没有明显的制造过程。也不像硬件那样,一旦研制成功,可以重复制造,在制造过程中进行质量控制,以保证产品的质量。而软件是通过人们的智力活动,把知识与技术转化成信息的一种产品。一旦某一软件项目研制成功,以后就可以大量地复制同一内容的副本。

3. 在软件的运行和使用期间,没有硬件那样的机械磨损、老化问题。任何机械、电子设备在运行和使用中,其失效率大都遵循如图 1.1(a)所示的 U 型曲线(即浴盆曲线)。因为在刚刚投入使用时,各部件尚未做到配合良好、运转灵活,常常容易出现问題。经过一段时间运行,就可以稳定下来。而当设备经历了相当长时间的运转,就会出现磨损、老化等问题,使失效率越来越大。当失效率达到一定程度,就到达了寿命的终点。

而软件的情况与此不同,它没有 U 型曲线的右半翼,因为它不存在磨损和老化问题。然而它存在退化问题。在软件的生存期中,为了它能够克服以前没有发现的故障,使它能够适应硬件、软件环境的变化以及用户新的要求,必须要多次修改(维护)软件,而每次更新必不可免地引入新的错误,这样一次次修改,导致软件失效率升高,如图 1.1(b)所示,从而使软件退化。因此,软件维护比硬件维护要复杂得多,与硬件的维修有着本质的差别。

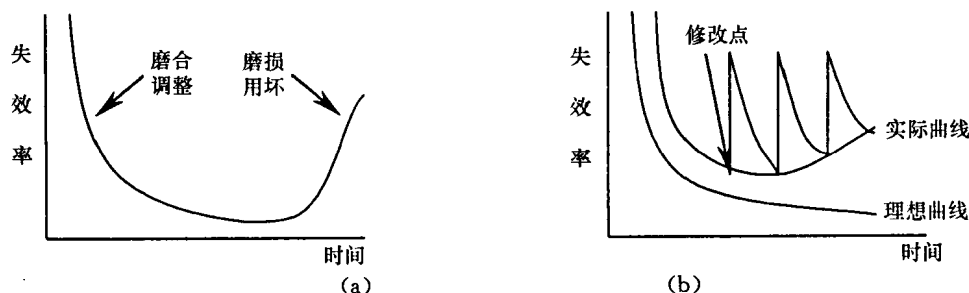


图 1.1 失效率曲线

(a) 硬件失效率曲线; (b) 软件失效率曲线

4. 软件的开发和运行常常受到计算机系统的限制,对计算机系统有着不同程度的依赖性。软件不能完全摆脱硬件单独活动。在开发和运行中必须以硬件提供的条件为依据。有的软件这种依赖性大些,常常为某个型号的计算机所专用,这对使用将带来许多不便。有的软件依赖于某个操作系统。为了解除这种依赖性,在软件开发中提出了软件移植的问题,并且把软件的可移植性作为衡量软件质量的因素之一。

5. 软件的开发至今尚未完全摆脱手工的开发方式。近年来软件技术虽然取得了不少进展,提出了许多新的开发方法,例如充分利用现成软件的复用技术、自动生成技术,也研制了一些有效的软件开发工具或软件开发环境。但在软件项目中采用的比率仍然很低。由于传统的手工开发方式仍然占据统治地位,开发的效率自然受到很大的限制。

6. 软件本身是复杂的。有人认为,人类能够创造的最复杂的产物是计算机软件。软件的复杂性可能来自它所反映的实际问题的复杂性。例如,它所反映的自然规律,或是人类社会的事务,都具有一定的复杂性;另一方面,也可能来自程序逻辑结构的复杂性,例如,一个系统软件要能处理各种可能出现的情况。软件开发,特别是应用软件的开发常常涉及到其它领域的专门知识,这对软件人员提出了很高的要求。软件的复杂性与软件技术的发展不相适应的状况越来越明显。图 1.2 示出软件技术的发展落后于复杂的软件需求,并且随着时间的推移,这个差距日益加大。

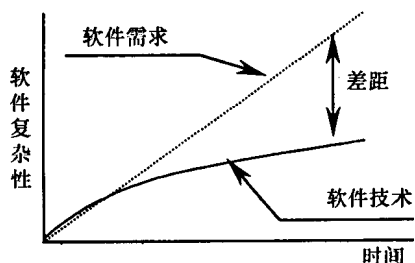


图 1.2 软件技术的发展落后于需求

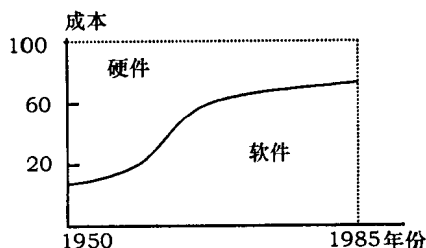


图 1.3 计算机系统硬、软件成本比例的变化

7. 软件成本相当昂贵。软件的研制工作需要投入大量的、复杂的、高强度的脑力劳动,它的成本是比较高的。80 年代软件的开销大大超过硬件的开销,如图 1.3 所示。今天的情况更是这样。美国每年投入软件开发的经费要几百亿美元。然而,也并非在所有软件开发上的花费都能获得成果。

8. 相当多的软件工作涉及到社会因素。类似于企业管理类型的软件自然是不言而喻的。许多软件的开发和运行涉及机构、体制及管理方式等问题,甚至涉及到人的观念和人们的心理。对于这些人的因素重视得不够,常常是软件工作遇到的问题之一。即使是对软件的看法不同也会有很大的影响。例如,由于主管部门对正在开发的软件不够理解,因而软件开发得不到应有的重视和必要的支持,造成人力和资金上的困难,它直接影响到项目的成败。

1.1.3 软件的分类

要给计算机软件做出科学的分类是很难的。鉴于不同类型的工程对象,对其进行开发和维护有着不同的要求和处理方法,我们从不同的方面对软件作出分类。

1. 按软件的功能进行划分

(1) 系统软件:能与计算机硬件紧密配合在一起,使计算机系统各个部件、相关的软件和数据协调、高效地工作的软件。例如,操作系统、数据库管理系统、设备驱动程序以及通信处理程序等。系统软件的工作通常伴随着:频繁地与硬件交往、大量地为用户服务、资源的共享与复杂的进程管理,以及复杂数据结构的处理。系统软件是计算机系统必不可少的一个组成部分。

(2) 支撑软件:是协助用户开发软件的工具性软件,其中包括帮助程序人员开发软件产品的工具,也包括帮助管理人员控制开发的进程的工具。表 1.1 给出了一些支撑软件的实例。

表 1.1 支撑软件举例

一般类型	支持需求分析
文本编辑程序 文件格式化程序 磁盘磁带向数据传输的程序 程序库系统	PSL/PSA 问题描述语言、问题描述分析程序 关系数据库系统 一致性检验程序 CARA 计算机辅助需求分析程序
支持设计	支持实现
图形软件包 结构化流程图绘图程序 设计分析程序 程序结构图编辑程序	编辑程序 交叉编辑程序 预编译程序 连接编辑程序
支持测试	支持管理
静态分析程序 符号执行程序 模拟程序 测试覆盖检验程序	PERT 进度计划评审方法绘图程序 标准检验程序 库管理程序

(3) 应用软件:在特定领域内开发,为特定目的服务的一类软件。现在几乎所有的国民经济领域都使用了计算机,为这些计算机应用领域服务的应用软件种类繁多,包括商业数据处

理软件、工程与科学计算软件、计算机辅助设计/制造(CAD/CAM)软件、系统仿真、智能产品嵌入软件、人工智能软件等,此外,还包括事务管理、办公自动化软件,以及中文信息处理、计算机辅助教学(CAI)等软件。

2. 按软件规模进行划分

按开发软件所需的人力、时间以及完成的源程序行数,可确定 6 种不同规模的软件,如表 1.2 所示。

表 1.2 软件规模的分类

类别	参加人员数	研制期限	产品规模(源程序行数)
微型	1	1~4 周	0.5K
小型	1	1~6 月	1~2K
中型	2~5	1~2 年	5~50K
大型	5~20	2~3 年	50~100K
甚大型	100~1000	4~5 年	1M (=1000K)
极大型	2000~5000	5~10 年	1~10M

(1)微型:一个人在几天之内完成的程序。写出的程序不到 500 行语句,仅供个人专用。通常这种小题目无需做严格的分析,也不需要有一套完整的设计、测试资料。不过这并不是说可以随便地不讲任何方法地做。事实说明,即使这样小的题目,如果经过一定的分析、系统设计、结构化编码以及有步骤地测试,肯定也是非常有益的。

(2)小型:一个人半年之内完成的 2 千行以内的程序。例如,数值计算问题或是数据处理问题就是这种规模的课题。这种程序通常没有与其它程序的接口。但需要按一定的标准化技术、正规的资料书写以及定期的系统审查,只是没有大题目那样严格。

(3)中型:5 个人以内在一年多时间里完成的 5 千到 5 万行的程序。这种课题开始出现软件人员之间、软件人员与用户之间的联系、协调的配合关系问题。因而计划、资料书写以及技术审查需要比较严格地进行。这类软件课题比较普遍,许多应用程序和系统程序就是这样的规模。在开发中使用系统的软件工程方法是完全必要的,这对提高软件产品质量和程序人员的工作效率起着重要的作用。

(4)大型:5~20 个人在两年多时间里完成的 5~10 万行的程序。例如编译程序、小型分时系统、应用软件包、实时控制系统等很可能是这种软件。参加工作的软件人员需要按二级管理,例如划分成若干小组,每组 5 人以下为好。在任务完成过程中,人员调整往往不可避免,因此会出现对新手的培训和逐步熟悉工作的问题。对于这样规模的软件,采用统一的标准,实行严格的审查是绝对必要的。由于软件的规模庞大以及问题的复杂性,往往会在开发的过程中出现一些事先难于做出估计的不测事件。

(5)甚大型:100~1000 人参加用 4~5 年时间完成的具有 100 万行程序的软件项目。这种甚大型项目可能会划分成若干个子项目,每一个子项目都是一个大型软件。子项目之间具有复杂的接口。例如,实时处理系统、远程通信系统、多任务系统、大型操作系统、大型数据库管理系统、军事指挥系统通常都有这样的规模。很显然,这类问题没有软件工程方法的支持,它的开发工作是不可想象的。

(6)极大型:2000~5000 人参加,5~10 年内完成的 1000 万行以内的程序。这类软件很