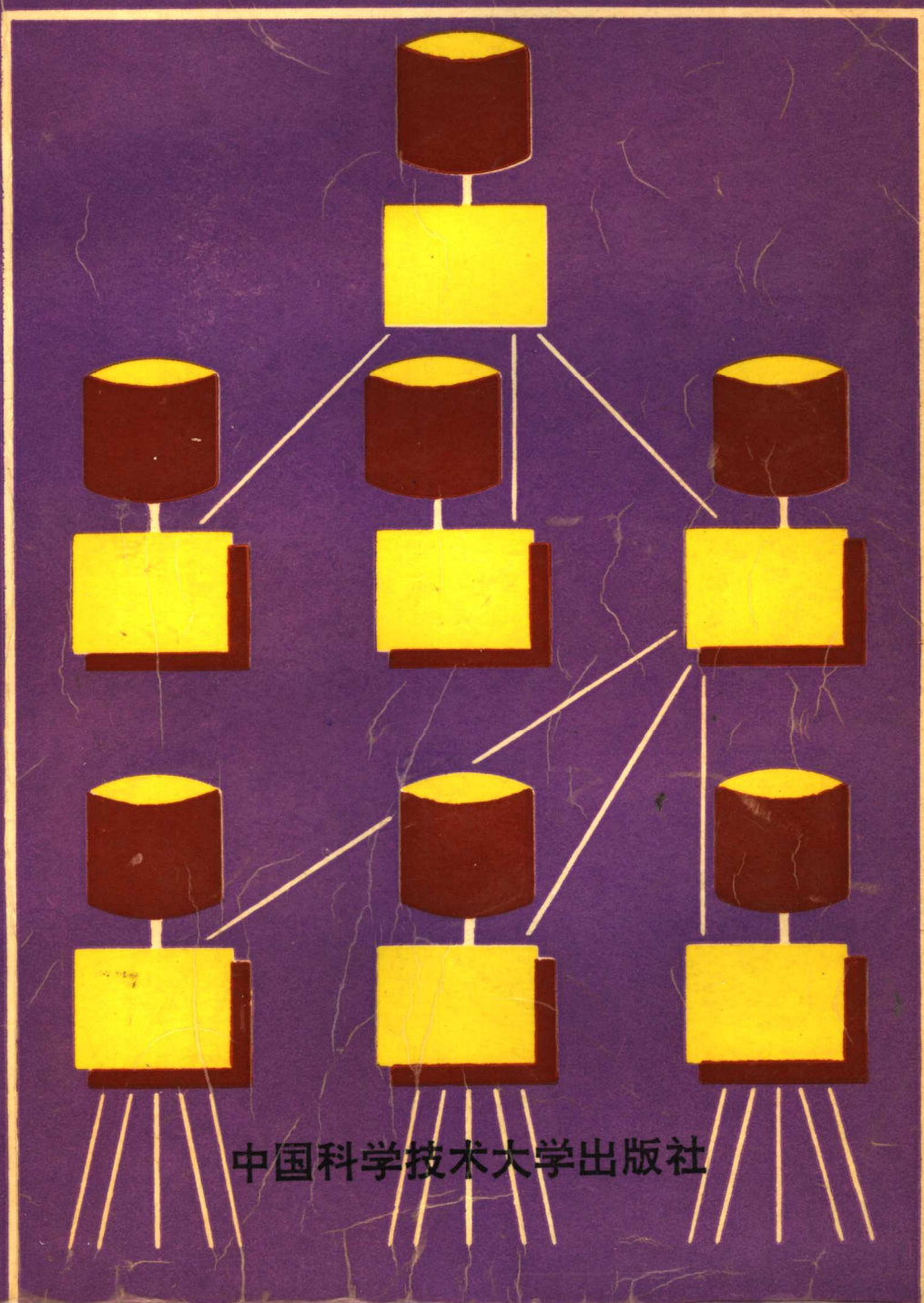


软件技术基础

■ 岳丽华 纪金龙 黄刘生 编著



中国科学技术大学出版社

软件技术基础

岳丽华 纪金龙 黄刘生 编著

中国科学技术大学出版社

1994·合肥

内 容 简 介

本书选择数据结构、操作系统、计算机局域网、数据库系统和软件工程等五门计算机软件课程内容作为主要内容,系统地介绍了软件技术的基础知识。

为适合非计算机专业人员的学习,各有关篇章中都删除了要求计算机专业知识较多且难于理解的有关实现方面的内容,但保留了各篇章的系统性和主要精髓。各篇章都提供了一定数量的习题,以便学习者加深对有关内容的理解。

本书既可以作为大学本科非计算机专业的教材,也可以作为各类专科院校计算机软件专业或各类软件培训的教材,还可以作为有中等文化程度以上读者的自学教材,对于广大计算机应用开发技术人员也有一定的参考价值。

(皖)新登字 08 号

图书在版编目(CIP)数据

软件技术基础/岳丽华等编著. —合肥:中国科学技术大学出版社,1994年9月
ISBN 7-312-00576-4

- I 软件技术基础
- II 岳丽华等编著
- III ①计算机 ②软件 ③基础 ④大学教学
- IV TP

凡购买中国科大版图书,如有白页、缺页、倒页者,由印刷厂负责调换

中国科学技术大学出版社出版发行

(安徽省合肥市金寨路96号,230026)

中国科学技术大学印刷厂印刷

全国新华书店经销

开本:787×1092/16 印张:30.75 字数:787千

1994年9月第1版 1994年9月第1次印刷

印数:1—5000册

ISBN 7-312-00576-4/TP·73 定价:14.50元

前 言

随着计算机的应用越来越多地渗入到国民生产的各个领域,计算机应用再也不只是计算机专业人员的独门技术。非计算机专业人员除学习计算机程序设计语言外,更多地学习计算机专业课程已是刻不容缓的事情。

计算机软件方面的课程很多,怎样使非计算机专业人员既能学到计算机软件的精髓,又不致于陷入计算机软件专业繁多的课程学习之中呢?选择哪些课程作为非计算机专业的计算机软件主干课程就是一件比较重要的事情。根据中国科学技术大学多年来开设计算机辅修专业的教学实践,我们感到,除学习一般程序设计语言外,把数据结构、操作系统、计算机网络、数据库系统和软件工程作为非计算机专业人员的软件基础课是比较合适的。这些内容的学习将使他们对计算机软件的发展和内涵有更深入的了解,对他们今后开发自己所需的软件系统将有很大的帮助。

为了编好这本教材,中国科学技术大学教务处专门组织了全校各系有关人员一起讨论,制定了本课程的教学大纲,最后委托计算机系组织有多年教学实践经验的教师执笔编写。

本书共分五篇。第一篇数据结构篇,介绍了一些最常用的数据结构,阐明了各种数据结构内在的逻辑关系及其在计算机中的存储表示,并讨论了对这些数据结构的各种运算。第二篇操作系统篇,从资源管理的角度出发,全面介绍了操作系统的基本概念、设计原理、实现技术及典型系统(如 UNIX)的使用。第三篇计算机局域网络篇,从网络使用者的角度介绍了计算机网络的基本概念、拓扑结构与通信协议,并通过对以太网和 Novell Netware 网软件的介绍,使读者了解计算机网络的实现原理。第四篇数据库系统篇,以关系型数据库系统为例,介绍了数据库系统的基本概念、结构和功能,并较详细地介绍了目前广泛使用的微机数据库系统 FOXBASE⁺;此外,还对分布式数据库的概念和网络数据库作了简要介绍。第五篇软件工程篇,从软件生命周期的角度出发,全面讨论了软件设计的方法问题。

本书的绪言、第四篇数据库系统和第五篇软件工程由岳丽华编写,第一篇数据结构由黄刘生编写,第二篇操作系统和第三篇计算机局域网络由纪金龙编写。

在本书的编写过程中,得到了中国科学技术大学朱滨教务长的全力支持,陈意云副教授和唐策善教授都给予过很多有益的指导与帮助;全书定稿前,陈意云副教授审阅了全部书稿;另外,在编写中我们也参考了许多同类书籍,在此都一并表示最诚意的感谢。

由于我们水平有限,编写时间仓促,书中难免有不确切甚或错误之处,敬请读者原谅并不吝指正。

编 著 者

1993 年 10 月 7 日

于中国科学技术大学

目 次

前言	(I)
绪论	(1)
0.1 计算机语言的发展	(1)
0.1.1 机器语言	(1)
0.1.2 汇编语言	(2)
0.1.3 高级程序设计语言	(3)
0.2 计算机软件的发展	(4)
0.2.1 数据结构	(4)
0.2.2 操作系统和计算机网络	(4)
0.2.3 数据库管理系统	(5)
0.2.4 软件工程	(5)

第一篇 数据结构

第一章 引论	(9)
1.1 什么是数据结构	(9)
1.2 算法描述	(11)
1.3 算法分析	(15)
习题	(18)
第二章 线性结构	(20)
2.1 线性表	(20)
2.1.1 线性表的定义和运算	(20)
2.1.2 线性表的顺序存储结构	(21)
2.1.3 线性表的链式存储结构	(25)
2.2 栈	(34)
2.2.1 栈的定义和运算	(34)
2.2.2 栈的存储结构	(35)
2.3 队列	(39)
2.3.1 队列的定义和运算	(39)
2.3.2 队列的存储结构	(40)
习题	(46)

第三章 树结构	(47)
3.1 树的定义和术语	(47)
3.2 二叉树	(50)
3.2.1 二叉树的定义和基本性质	(50)
3.2.2 几种特殊形态的二叉树	(52)
3.2.3 二叉树的存储结构	(53)
3.2.4 二叉树的遍历	(56)
3.3 树的存储结构和遍历	(59)
3.3.1 树的存储结构	(59)
3.3.2 树的遍历	(63)
习题	(63)
第四章 图	(66)
4.1 图的定义和术语	(66)
4.2 图的存储结构	(69)
4.2.1 邻接矩阵表示法	(69)
4.2.2 邻接表表示法	(71)
4.3 图的遍历	(73)
4.3.1 深度优先搜索	(74)
4.3.2 广度优先搜索	(77)
习题	(79)
第五章 查找	(80)
5.1 线性表的查找	(80)
5.1.1 顺序查找	(80)
5.1.2 二分查找	(82)
5.1.3 分块查找	(84)
5.2 二叉排序树的查找	(86)
5.2.1 二叉排序树的插入和生成	(87)
5.2.2 二叉排序树的删除	(89)
5.2.3 二叉排序树的查找	(90)
5.3 散列表的查找	(92)
5.3.1 散列表的概念	(92)
5.3.2 散列函数的构造方法	(95)
5.3.3 处理冲突的方法	(96)
5.3.4 散列表的查找及分析	(99)
习题	(102)
第六章 排序	(104)
6.1 插入排序	(104)
6.2 交换排序	(107)
6.2.1 起泡排序	(107)
6.2.2 快速排序	(109)

6.3 选择排序	(112)
6.3.1 直接选择排序	(112)
6.3.2 堆排序	(113)
6.4 归并排序	(118)
习题	(120)

第二篇 操作系统

第一章 引论	(125)
1.1 计算机系统的硬件组织	(126)
1.1.1 小型机与微型机的典型结构	(126)
1.1.2 大、中型计算机的典型结构	(127)
1.2 操作系统的概念	(127)
1.2.1 什么是操作系统	(127)
1.2.2 操作系统运行的环境	(128)
1.2.3 多道程序设计技术	(130)
1.2.4 操作系统与用户的接口	(131)
1.3 操作系统的发展	(133)
1.4 操作系统的分类	(134)
1.4.1 多道批处理系统	(134)
1.4.2 分时系统	(135)
1.4.3 实时系统	(135)
1.5 操作系统的结构	(136)
1.5.1 模块接口	(136)
1.5.2 分层结构	(136)
1.5.3 虚拟机	(137)
1.5.4 客户-服务器模式	(138)
习题	(138)
第二章 进程与处理机管理	(139)
2.1 进程的概念	(139)
2.1.1 进程的引入	(139)
2.1.2 进程的定义	(140)
2.1.3 进程的基本状态	(140)
2.1.4 进程的组织与控制	(141)
2.2 进程间的通信	(142)
2.2.1 临界区问题	(143)
2.2.2 通过“忙等待”实现互斥	(144)
2.2.3 信号量	(146)

2.2.4 生产者和消费者问题	(148)
2.2.5 消息传送	(149)
2.3 处理机调度	(151)
2.3.1 调度的层次	(151)
2.3.2 作业调度	(151)
2.3.3 进程调度	(152)
2.3.4 调度算法	(152)
习题	(154)
第三章 输入输出管理	(155)
3.1 引言	(155)
3.1.1 输入输出组织	(155)
3.1.2 输入输出管理的任务和功能	(156)
3.2 I/O 硬件的工作原理	(156)
3.2.1 I/O 设备	(156)
3.2.2 设备控制器	(158)
3.2.3 I/O 操作的实现技术	(158)
3.3 I/O 软件的结构与工作原理	(160)
3.3.1 I/O 软件的目标	(160)
3.3.2 中断处理程序	(161)
3.3.3 设备驱动程序	(161)
3.3.4 独立于设备的 I/O 软件	(162)
3.3.5 用户空间的 I/O 软件	(162)
3.3.6 I/O 处理的基本过程	(163)
3.4 死锁	(163)
3.4.1 死锁的产生	(163)
3.4.2 产生死锁的必要条件	(164)
3.4.3 处理死锁的策略	(165)
习题	(166)
第四章 存储管理	(167)
4.1 引言	(167)
4.1.1 存储器的物理组织	(167)
4.1.2 存储管理的功能	(167)
4.1.3 存储分配与地址重定位	(168)
4.2 实存储器的管理	(169)
4.2.1 分区式存储管理	(169)
4.2.2 分页式存储管理	(175)
4.2.3 分段式存储管理	(178)
4.3 虚拟存储器管理	(181)
4.3.1 虚拟存储器的概念	(181)
4.3.2 分页虚拟存储管理	(181)

习题	(182)
第五章 文件系统	(184)
5.1 引言	(184)
5.1.1 文件和文件系统	(184)
5.1.2 文件系统的功能	(185)
5.2 文件的组织与存储方式	(185)
5.2.1 文件的逻辑组织	(185)
5.2.2 文件的物理组织	(186)
5.3 文件辅存空间的管理	(187)
5.3.1 空白文件目录	(187)
5.3.2 空白块链	(187)
5.3.3 位示图	(187)
5.4 文件的目录结构	(188)
5.4.1 文件目录和文件描述符	(188)
5.4.2 一级目录结构	(188)
5.4.3 多级目录结构	(188)
5.5 文件的共享与保护	(190)
5.5.1 文件的共享	(190)
5.5.2 文件的存取控制	(191)
5.6 文件的使用	(192)
5.6.1 文件的建立	(192)
5.6.2 文件的打开	(193)
5.6.3 读文件	(193)
5.7 文件系统的层次模型	(193)
习题	(195)
第六章 UNIX 操作系统简介	(196)
6.1 引言	(196)
6.2 UNIX 系统的基本结构	(197)
6.2.1 系统结构	(197)
6.2.2 进程管理	(198)
6.2.3 存储管理	(199)
6.2.4 设备(输入输出)管理	(200)
6.2.5 文件系统	(200)
6.3 UNIX 系统使用入门	(203)
6.3.1 基本概念	(203)
6.3.2 目录和文件管理	(206)
6.3.3 DOS 格式的操作命令	(213)
6.3.4 Shell 语言简介	(215)

第三篇 计算机局域网

第一章 引论	(221)
1.1 计算机网络的形成与发展	(221)
1.2 计算机网络的组成	(222)
1.3 数据通信的基本概念	(223)
1.3.1 模拟数据通信和数字数据通信	(223)
1.3.2 波特率和比特率	(224)
1.3.3 同步通信和异步通信	(225)
1.3.4 单工通信和双工通信	(225)
1.3.5 通信介质的共享	(226)
1.4 计算机局域网	(228)
1.4.1 局域网的定义	(229)
1.4.2 局域网的分类	(229)
习题	(230)
第二章 局域网的实现技术	(231)
2.1 局域网的传输介质	(231)
2.1.1 双绞线	(231)
2.1.2 同轴电缆	(232)
2.1.3 光纤电缆	(232)
2.2 局域网的拓扑结构	(233)
2.2.1 星形结构	(233)
2.2.2 环形结构	(233)
2.2.3 总线结构	(234)
2.2.4 树形结构	(234)
2.2.5 网络拓扑结构的选择	(235)
2.3 局域网的访问控制方式	(236)
2.3.1 CSMA/CD 方式	(236)
2.3.2 令牌环方式	(237)
2.3.3 令牌总线方式	(237)
习题	(238)
第三章 局域网的通信协议	(239)
3.1 计算机网络的 OSI 模型	(239)
3.2 局域网的通信协议模型	(240)
3.3 Novell 网的通信协议	(242)
习题	(242)
第四章 局域网的操作系统	(243)

4.1 局域网操作系统的结构及功能	(243)
4.2 局域网操作系统的分类	(244)
4.3 几种流行的微机局域网操作系统简介	(245)
习题	(246)
第五章 Ethernet 网络与 Novell 网络系统简介	(247)
5.1 Ethernet 局域网	(247)
5.1.1 Ethernet 网络的结构	(247)
5.1.2 Ethernet 网络的控制策略	(249)
5.1.3 Ethernet 网络的通信协议	(249)
5.2 Netware 的特点	(251)
5.3 Netware 的基本结构	(255)
5.3.1 Netware 的逻辑结构	(255)
5.3.2 Netware 的目录结构	(257)
5.3.3 Netware 的分层结构与协议	(257)
5.4 Netware 操作入门	(260)
5.4.1 Novell 网络的基本概念	(260)
5.4.2 启动文件服务器和工作站	(262)
5.4.3 登录到服务器上	(264)
5.4.4 建立或修改口令	(264)
5.4.5 查看文件服务器上的用户	(265)
5.4.6 查看或建立映射驱动器	(265)
5.4.7 退网和关闭文件服务器	(266)
5.5 Netware 命令实用程序	(267)
5.5.1 工作站实用程序	(267)
5.5.2 文件服务器实用程序	(270)
5.6 Novell 菜单实用程序	(272)
5.6.1 系统配置实用程序(SYSICON)	(273)
5.6.2 会话管理实用程序(SESSION)	(275)
5.6.3 文件管理实用程序(FILER)	(276)
习题	(277)

第四篇 数据库系统

第一章 引论	(281)
1.1 什么是数据库系统	(281)
1.1.1 数据库系统的定义	(281)
1.1.2 数据库系统的主要特征	(282)
1.2 数据库体系结构的三级模式	(284)

习题	(285)
第二章 数据模型	(286)
2.1 概述	(286)
2.1.1 数据模型的定义	(286)
2.1.2 数据模型的功能	(286)
2.2 关系模型	(287)
2.3 关系数据库操作语言	(289)
2.3.1 关系代数	(289)
2.3.2 关系演算	(290)
习题	(292)
第三章 数据库管理系统	(293)
3.1 DBMS 的构成	(293)
3.2 数据库语言	(293)
3.2.1 数据定义语言 DDL	(294)
3.2.2 数据操作语言 DML	(294)
3.2.3 数据控制语言 DCL	(294)
第四章 FOXBASE+ 简介	(295)
4.1 FOXBASE+ 的特点	(295)
4.1.1 FOXBASE+ 的文件	(295)
4.1.2 FOXBASE+ 的系统构成及执行方式	(296)
4.1.3 FOXBASE+ 的语句规则	(297)
4.2 数据库操作	(298)
4.2.1 FOXBASE+ 的定义语句	(298)
4.2.2 数据库记录的输入	(302)
4.2.3 数据库记录的查询	(305)
4.2.4 数据库记录的更新	(307)
4.2.5 多库之间的操作	(310)
4.2.6 库函数语句	(313)
4.2.7 数据库记录的显示输出	(314)
4.3 参数设置操作	(317)
4.3.1 输出环境设置	(317)
4.3.2 状态设置	(319)
4.3.3 数据库操作参数设置	(323)
4.3.4 程序操作参数设置	(324)
4.3.5 其他设置语句	(325)
4.4 FOXBASE+ 的函数	(327)
4.4.1 日期和时间函数	(328)
4.4.2 字符函数	(330)
4.4.3 数值函数	(334)
4.4.4 测试函数	(336)

4.4.5 库文件函数	(337)
4.4.6 逻辑函数	(338)
4.5 FOXBASE+ 的程序设计	(341)
4.5.1 程序的建立和执行	(341)
4.5.2 常量和变量	(343)
4.5.3 表达式	(344)
4.5.4 内存变量的操作	(345)
4.5.5 内存环境	(351)
4.5.6 分支程序设计	(354)
4.5.7 循环程序设计	(357)
4.5.8 过程设计及其调用	(361)
4.6 与其他高级程序语言的接口	(368)
4.6.1 RUN 语句	(368)
4.6.2 数据交换	(368)
4.7 应用实例	(369)
习题	(382)
第五章 数据库设计	(384)
5.1 E-R 模型	(384)
5.1.1 E-R 模型设计	(384)
5.1.2 应用实例	(386)
5.2 逻辑数据库设计	(389)
5.2.1 E-R 模型到关系模型的转换	(389)
5.2.2 函数相关	(390)
5.2.3 第二范式	(390)
5.2.4 第三范式	(391)
5.3 物理数据库设计	(392)
习题	(392)
第六章 分布式数据库	(393)
6.1 分布式数据处理	(393)
6.1.1 数据处理的变革	(393)
6.1.2 分布式数据处理与集中式数据处理	(393)
6.2 分布式数据库	(394)
6.2.1 分布式数据库定义	(394)
6.2.2 分布式数据库语言	(395)
6.2.3 数据分布	(397)
6.2.4 并发控制	(398)
6.3 FOXBASE+ 网络数据库功能	(399)
6.3.1 多用户语句	(399)
6.3.2 多用户函数	(401)
6.3.3 应用实例	(402)

习题	(405)
----------	-------

第五篇 软件工程

第一章 引论	(409)
1.1 什么是软件工程	(409)
1.2 软件生命周期	(410)
习题	(413)
第二章 软件计划	(414)
2.1 可行性研究	(414)
2.2 软件计划的内容	(415)
2.2.1 软件范围	(415)
2.2.2 资源	(416)
2.2.3 时间进度表	(417)
2.2.4 软件费用	(417)
2.2.5 软件计划书	(419)
习题	(420)
第三章 软件需求分析	(421)
3.1 需求分析的目标和任务	(421)
3.2 数据流分析技术	(422)
3.2.1 数据流图	(422)
3.2.2 数据词典	(424)
3.2.3 数据流分析实例	(426)
3.2.4 需求分析说明书	(435)
3.3 面向对象分析技术	(435)
3.3.1 对象及其性质	(436)
3.3.2 实例	(439)
习题	(443)
第四章 软件设计	(444)
4.1 软件设计准则	(444)
4.2 结构化设计	(446)
4.2.1 数据流图的类型	(446)
4.2.2 设计步骤	(448)
4.3 面向对象的设计	(450)
习题	(456)
第五章 软件测试	(457)
5.1 软件测试的任务	(457)
5.2 软件测试方法	(457)

5.3 软件测试过程	(459)
5.3.1 模块测试	(459)
5.3.2 整体测试	(460)
5.3.3 有效性测试	(462)
5.3.4 系统测试	(462)
5.4 测试用例设计	(462)
5.4.1 白盒法	(462)
5.4.2 黑盒法	(467)
5.4.3 综合策略	(468)
习题	(468)
第六章 软件维护	(469)
6.1 软件维护的定义	(469)
6.2 软件的易维护性	(470)
6.3 软件维护的任务	(470)
6.3.1 软件维护机构	(470)
6.3.2 编写报告	(471)
6.3.3 软件维护模型	(471)
6.3.4 记录保存	(472)
6.3.5 评价	(473)
习题	(473)
参考文献	(475)

绪 论

自 1946 年第一台通用电子计算机 ENIAC 问世以来,计算机走过了近半个世纪的历程。从早期需一栋大楼才能安装得下的“巨型”机,到如今可以随身携带的便携式计算机,计算机的历史也可以说是一部使计算机越来越方便人们使用的发展史。

计算机的发展,从晶体管到大规模集成电路,从仅有几十个累加器的存储空间到上百兆位的大容量存储空间,从每秒仅几千次的运算速度到千亿次的运算速度,这其中计算机硬件有着其辉煌壮丽的历史篇章,但更使计算机界感到骄傲与自豪的是计算机软件的发展。

早期,计算机软件还需与计算机硬件搭配在一起出售,而如今,无论计算机硬件对计算机有多么重要,实际情况正如第七届软件工程国际会议主席所说:软件是任何系统的头脑、心脏和灵魂。计算机只有在配备了所需的各种软件后,才能展现出令人眩目的多功能的光彩。

“软件”一词是 60 年代才被普遍使用的,通常用来描述计算机的非硬件成分,特别是指计算机程序。虽然计算机软件与各种各样的程序相关,但根据程序的功能可将软件进一步划分为系统软件和应用软件两大类。系统软件通常是指那些不仅仅是为解决某个计算或数据处理问题而设计的程序,系统软件的使用将有利于其他程序的产生、调试、维护和正常运转。应用软件则主要是针对某一特定应用而设计的,它通常必须在一定的系统软件支撑下运行。

由于软件也就是程序,而程序必须用计算机语言完成,因此,我们首先从计算机语言的发展来看计算机软件的发展。然后,我们再讨论与软件设计技术有关的其他软件发展。

0.1 计算机语言的发展

0.1.1 机器语言

机器语言是可以被机器直接执行的程序设计语言,这种语言的语句是由 0,1 两位数字组成的字符串。语句的操作对象是存储地址,语句的操作符部分执行某个算术或逻辑操作。由于机器语言通常是与某个特定机器有关,因此,不同的机器有不同的语句表示法,语句的长度(0,1 二进制数的位数)也可以不同,操作符的编码也不相同,即使是同一组 0,1 串序列,在不同的计算机上也有不同的解释。因此,编写机器语言程序是一件相当繁杂和痛苦的工作。

例如,我们要编制两个数相加的程序。

在大部分计算机中,算术操作是在特殊的累加器寄存器中执行的,数据都存放在以字或字节为单位地址的存储器中。因此,要计算两个数的和,首先要把其中第一个数先放到累加器中,然后将累加器的内容与存放在另一地址中的数相加,最后再将结果送入结果存储器地址中。选

用 36 位长的机器指令,完成上述操作的机器语句如下:

首先是取数,语句是二进制数串:

000101000000000000000000000000001100100

该串的最左边 12 位是指令位,执行的动作是把最右边 15 位表示的地址 000000001100100(十进制数 100)中的内容取出,送累加器中。

要把累加器中的数值与 101 地址的值相加,则需语句:

000100000000000000000000000000001100101

最后,将计算结果送入 102 地址的语句是:

000110000001000000000000000000001100110

将上述三个 0,1 数字串存放在计算机的某一连续存储器中,将要计算的数值分别存入地址 100 和 101 中。然后,告诉机器从存放取数语句的地址开始执行,则完成了将两个数相加的计算工作。显然,编制这样的计算机软件程序是相当困难的事情。目前,机器语言程序几乎已不再被使用。

0.1.2 汇编语言

汇编语言是机器语言的符号表示。例如,对于加法运算,不用二进制串 000100000000 表示,而可以用符号 ADD 表示。这样,既增加了语言的可读性,又为程序的编写带来了方便。上例中两数值相加的程序可以用如下三条汇编语句替换:

CLA 100

ADD 101

STO 102

这里的运算对象仍然是使用存放该数值的存储器地址(十进制数表示)。在汇编语言中,我们还可以用伪操作给算子赋名,然后用名字来存取数值。这种伪操作类似于编辑在书稿上做的编辑注解,用以通知排版人员以什么形式排出此稿,而不是真正要排版人员将此注解也作为稿文输入。通过伪操作赋名,上例的语句可改写为:

CLA Aug1

ADD Aug2

STO Sum

其相应的赋名伪操作是:

Aug1 0

Aug2 1

Sum 0

用汇编语言写出的程序,并不可以被计算机直接执行。而需要一个翻译,其功能是将汇编语句的符号转换为对应的机器语言语句,操作符替换为机器的操作码,运算对象根据伪操作的说明替换为相应的存放地址。完成上述工作的是汇编程序,它是系统软件之一。任何用汇编语言写出的程序都称为汇编源程序,把汇编源程序作为汇编程序软件的输入,经过汇编程序加工,产生的就是可执行的机器语言程序。

虽然汇编语言只是机器语言的符号表示,但是它使程序设计摆脱了对存储器的直接分配。另外,汇编程序还提供了很多功能很强的宏定义语句,使得汇编语言的功能得到增强。直到今