

中国计算机软件专业技术水平考试指定用书

中国计算机软件专业技术资格和水平考试中心组织编写

程 序 设 计

王春森 主编 (程序员级)



清华大学出版社

<http://www.tup.tsinghua.edu.cn>



中国计算机软件专业技术水平考试指定用书

程 序 设 计

(程序员级)

中国计算机软件专业技术资格和水平考试中心 组织编写

王春森 主编

清华大学出版社

(京)新登字 158 号

内 容 简 介

本书是中国计算机软件专业技术资格和水平考试中心组织编写的辅导教材之一。编写的依据是考试大纲,对程序员级考试应必备的知识 and 能力作了详尽的讲解。

主要包括:计算机硬件知识、程序语言、操作系统、软件工程、数据库、多媒体和网络等领域的基础知识,与程序设计有关的数据结构和基本算法。

本书可作为参加程序员级资格和水平考试的应考者的主要辅导读物,并可供各部门举办辅导班作为教材,也可作为大专院校师生和广大工程技术人员学习参考。

版权所有,翻印必究。

本书封面贴有清华大学出版社激光防伪标签,无标签者不得销售。

图书在版编目(CIP)数据

程序设计:程序员级/王春森主编. —北京:清华大学出版社,1999.5

中国计算机软件专业技术水平考试指定用书

ISBN 7-302-03557-1

I. 程… II. 王… III. 程序设计-水平考试-自学参考资料 IV. TP311

中国版本图书馆(CIP)数据核字(1999)第16045号

出版者:清华大学出版社(北京清华大学校内,邮编100084)

<http://www.tup.tsinghua.edu.cn>

印刷者:北京昌平环球印刷厂

发行者:新华书店总店北京发行所

开本:787×1092 1/16 印张:21.75 字数:512千字

版次:1999年7月第1版 1999年8月第2次印刷

书号:ISBN 7-302-03557-1/TP·1953

印数:31001~62000

定价:33.00元

序

科学技术的日新月异,信息化时代的来临,使以计算机技术为基础的的信息科学在经济和社会生活各个领域得到了极为广泛的应用,其发展水平成为衡量国家经济与科技实力的重要标志已是不争之事实。各国都把培养大量高水平计算机专业人才作为 21 世纪经济和科技发展的重要战略目标之一。一些经济发达国家通过开展对计算机专业人才的教育培训,尤其是开展不同层次、不同规模的计算机水平测试,吸引、储备了大量计算机高级人才,为迎接日趋激烈的科技竞争奠定了坚实基础。这些国家的成功经验值得我们学习、借鉴。

中国计算机软件专业技术资格和水平考试自 1991 年开始实施至今已走过了 8 年的历程,共有近 40 万人参加考试,在国内外已产生较大影响。特别是 1999 年度中国计算机软件专业技术水平考试将由原来的一个专业扩展为程序设计、软件工程、数据库技术、计算机网络、多媒体技术 5 个专业,这无疑是中国计算机软件专业技术水平考试发展的一个质的飞跃,必将把中国计算机软件专业技术水平考试推向新的阶段。

我相信,新近编写出版的《中国计算机软件专业技术水平考试指定用书》能对广大应试者起到很好的指导作用。

我更加希望,在世纪之交,中国计算机软件专业技术水平考试能够抓住机遇,迎接竞争与挑战,为促进我国科教兴国战略的贯彻实施做出应有的贡献。

赵东亮

编者的话

中国计算机软件专业技术资格和水平考试实施至今已经历了十余年历程,得到了社会的认可。为了满足我国信息技术发展和企业对计算机软件人才的需求,目前已将资格和水平考试的范围和内容扩大为程序设计、软件工程、网络、多媒体和数据库五个方面,每个方面又分为若干级别,以适应社会上对各种软件人才的需求。

编者受中国计算机软件专业技术资格和水平考试中心委托,编写这套适用于程序设计方面考试要求的辅导教材,分为初级程序员级、程序员级和高级程序员级三册,按照考试大纲要求,分别面向能熟练使用指定的常用软件和具有初步程序编制能力的初级程序员;能按照软件设计说明书编制程序的程序员;能按照软件需求规格说明书进行软件设计和编制程序的高级程序员。

按照考试大纲要求,对程序员级或高级程序员级所需的知识结构涉及的面很广,一个条目在大学里往往是一学期的课程,因此编写的难度很高。考虑到参加考试的人员应有一定的基础,所以书中只对考试大纲中所涉及到的知识领域要点加以阐述,请读者谅解。考虑到参加考试的人员一般都已熟悉一种以上的程序设计语言,并已具有基本的编程能力,因此本书以数据结构和基本算法为主线,对读者原有的程序设计知识和编程能力起到总结、拓宽和提高的作用,增强程序的设计和阅读能力,掌握常用数据结构和算法的原理和要点。

全书共分八章。第1章计算机硬件基础知识由丛锁、高传善编写;第2章程序语言知识由张守志编写;第3章操作系统由于玉编写;第4章软件工程由钱乐秋编写;第5章数据库基础知识由丁宝康编写;第6章多媒体基础知识由白雪峰编写;第7章网络基础知识由曹邦伟编写;第8章算法与数据结构由夏宽理编写。张作义、顾天珍、夏玲明和张培玲等同志在全书编写过程中做了大量的辅助工作,在此表示感谢。

因水平有限,书中难免存在错漏和不妥之处,望请读者指正,以利改进和提高。

编者

1999年4月于复旦大学

目 录

| | | | |
|------------------------------|----|-------------------|----|
| 第 1 章 计算机硬件基础知识 | 1 | 3.1.2 什么是操作系统 | 71 |
| 1.1 计算机系统概述 | 1 | 3.1.3 操作系统的功能 | 72 |
| 1.1.1 计算机史概述 | 1 | 3.1.4 操作系统的工作原理 | 73 |
| 1.1.2 计算机系统的结构 | 2 | 3.1.5 操作系统的硬件基础 | 74 |
| 1.2 数据的计算机表示 | 2 | 3.2 并发进程 | 75 |
| 1.2.1 数制 | 2 | 3.2.1 程序、进程和处理器 | 75 |
| 1.2.2 数值数据的表示方法 | 6 | 3.2.2 进程间的通信 | 76 |
| 1.2.3 符号数据的表示方法 | 11 | 3.2.3 信号量与 P、V 操作 | 76 |
| 1.2.4 汉字的处理 | 12 | 3.2.4 高级通信原语 | 77 |
| 1.2.5 数据的校验方法 | 13 | 3.3 系统核心 | 77 |
| 1.3 计算机算术运算和逻辑运算 | 18 | 3.3.1 基本硬件机构 | 78 |
| 1.3.1 二进制数的算术运算 | 18 | 3.3.2 进程 | 78 |
| 1.3.2 逻辑代数及逻辑运算 | 21 | 3.3.3 进程管理 | 79 |
| 1.4 计算机基本结构 | 23 | 3.4 存储管理 | 80 |
| 1.4.1 总线结构 | 23 | 3.4.1 引言 | 80 |
| 1.4.2 中央处理器(CPU) | 26 | 3.4.2 实存管理 | 81 |
| 1.4.3 存储器系统 | 36 | 3.4.3 虚存组织 | 84 |
| 1.4.4 I/O 系统 | 41 | 3.4.4 虚存管理 | 85 |
| 1.5 指令系统 | 45 | 3.5 设备管理 | 86 |
| 1.5.1 机器指令的格式 | 45 | 3.5.1 引言 | 86 |
| 1.5.2 指令的种类和功能 | 47 | 3.5.2 设备的分配 | 88 |
| 1.5.3 指令的寻址方式 | 50 | 3.5.3 磁盘调度算法 | 89 |
| 1.5.4 指令的执行过程 | 51 | 3.5.4 RAID | 89 |
| 第 2 章 程序语言知识 | 52 | 3.6 文件管理 | 89 |
| 2.1 程序语言基础知识 | 52 | 3.6.1 引言 | 89 |
| 2.1.1 概述 | 52 | 3.6.2 文件的结构和组织 | 90 |
| 2.1.2 程序语言的数据类型 | 55 | 3.6.3 文件访问方法 | 92 |
| 2.1.3 程序语言的控制结构 | 60 | 3.6.4 文件存储设备管理 | 92 |
| 2.2 汇编程序基础知识 | 62 | 3.6.5 文件控制块和文件目录 | 93 |
| 2.2.1 汇编语言 | 62 | 3.6.6 文件的使用 | 94 |
| 2.2.2 汇编程序 | 63 | 3.7 作业管理和用户界面 | 95 |
| 2.3 解释程序基础知识 | 65 | 3.7.1 引言 | 95 |
| 2.4 编译程序基础知识 | 67 | 3.7.2 用户作业管理 | 95 |
| 第 3 章 操作系统 | 70 | 3.7.3 作业调度 | 96 |
| 3.1 概述 | 70 | 3.7.4 用户界面 | 97 |
| 3.1.1 引言 | 70 | 3.8 其他管理 | 97 |

| | | | | |
|-----------------|---------------|-----|-----------------------|-----|
| 3.8.1 | 死锁问题 | 97 | 第5章 数据库基础知识 | 141 |
| 3.8.2 | 多处理器系统与线程 | 98 | 5.1 数据管理技术的发展 | 141 |
| 3.9 | 实例 | 98 | 5.1.1 数据管理技术的发展阶段 | 141 |
| 3.9.1 | UNIX 系统 | 98 | 5.1.2 数据库技术的基本概念 | 143 |
| 3.9.2 | Windows NT 系统 | 101 | 5.1.3 数据库系统的进一步发展 | 143 |
| 第4章 软件工程 | | 104 | 5.2 数据描述 | 143 |
| 1.1 | 软件工程概述 | 104 | 5.2.1 数据的描述 | 143 |
| 4.1.1 | 软件生存周期 | 104 | 5.2.2 数据联系的描述 | 145 |
| 4.1.2 | 软件开发模型 | 106 | 5.3 数据模型 | 146 |
| 4.2 | 系统分析和软件项目计划 | 106 | 5.3.1 数据模型的定义 | 146 |
| 4.2.1 | 系统分析 | 106 | 5.3.2 概念数据模型 | 147 |
| 4.2.2 | 可行性分析 | 107 | 5.3.3 基本数据模型 | 148 |
| 4.2.3 | 编制项目开发计划 | 107 | 5.4 数据库系统的全局结构 | 150 |
| 4.3 | 需求分析 | 107 | 5.4.1 数据库的数据体系结构 | 150 |
| 4.3.1 | 需求分析的任务 | 108 | 5.4.2 数据库系统 | 152 |
| 4.3.2 | 结构化分析方法简介 | 108 | 5.4.3 数据库管理系统 | 151 |
| 4.4 | 软件设计 | 109 | 5.4.4 用户访问数据的过程 | 155 |
| 4.4.1 | 软件设计的概念和原则 | 109 | 5.4.5 数据库系统的全局结构 | 156 |
| 4.4.2 | 结构化设计方法简介 | 112 | 5.5 关系模型和关系运算 | 158 |
| 4.4.3 | 详细设计 | 113 | 5.5.1 关系模型的基本概念 | 158 |
| 4.5 | 编码 | 117 | 5.5.2 关系运算 | 159 |
| 4.5.1 | 程序设计语言 | 117 | 5.5.3 关系数据库 DML 的特点 | 164 |
| 4.5.2 | 对源程序的质量要求 | 119 | 5.6 关系数据库 SQL 语言简介 | 164 |
| 4.5.3 | 编码风格 | 119 | 5.6.1 SQL 数据库的数据体系结构 | 165 |
| 4.6 | 软件测试 | 121 | 5.6.2 SQL 语言的组成 | 165 |
| 4.6.1 | 测试的基本概念 | 122 | 5.6.3 SQL DDL | 166 |
| 4.6.2 | 测试步骤 | 123 | 5.6.4 SQL DML 的数据查询语句 | 168 |
| 4.6.3 | 白盒测试的测试用例设计 | 124 | 5.6.5 SQL DML 的数据更新语句 | 173 |
| 4.6.4 | 黑盒测试的测试用例设计简介 | 127 | 5.6.6 SQL 的访问控制 | 175 |
| 4.7 | 面向对象方法的基本概念 | 128 | 5.6.7 嵌入式 SQL | 175 |
| 4.8 | 软件维护 | 130 | 5.7 数据库设计过程 | 175 |
| 4.8.1 | 软件维护的分类 | 131 | 5.7.1 数据库工程 | 176 |
| 4.8.2 | 与软件维护有关的问题 | 131 | 5.7.2 数据库系统生存期 | 176 |
| 4.8.3 | 可维护性 | 132 | 5.7.3 数据库设计的具体过程 | 176 |
| 4.8.4 | 软件维护活动流程 | 132 | 第6章 多媒体基础知识 | 183 |
| 4.9 | 软件管理 | 133 | 6.1 多媒体的概念和特征 | 183 |
| 4.10 | 软件质量保证 | 136 | | |
| 4.10.1 | 软件质量特性 | 136 | | |
| 4.10.2 | 软件质量保证概述 | 139 | | |

| | | | | | |
|--------------|-----------------------------|------------|--------------|----------------------------|------------|
| 6.1.1 | 什么是多媒体 | 184 | | 专用设备 | 217 |
| 6.1.2 | 多媒体是信息系统建造的新范型 | 184 | 7.5.2 | 网络采用的主要传输媒体 | 218 |
| 6.1.3 | 多媒体的关键技术 | 185 | 7.5.3 | LAN 的协议标准和主要的媒体访问控制方式 | 219 |
| 6.1.4 | 多媒体系统的构成 | 186 | 7.5.4 | LAN 的硬件构建示例 | 220 |
| 6.2 | 图像和图形 | 188 | 7.6 | 构建 WAN 网络 | 222 |
| 6.2.1 | 基本概念 | 188 | 7.6.1 | 通信服务和网络接入技术 | 222 |
| 6.2.2 | 图形、图像格式 | 190 | 7.6.2 | 两类重要的网络互联设备 | 223 |
| 6.3 | 声音(音频) | 192 | 7.6.3 | WAN 的几种典型连接方式 | 225 |
| 6.3.1 | 基本概念 | 193 | 7.7 | Internet 的应用 | 226 |
| 6.3.2 | 语音合成与语言合成 | 194 | 7.7.1 | Internet 的服务 | 227 |
| 6.3.3 | 音乐合成和 MIDI | 195 | 7.7.2 | IP 地址和域名 | 228 |
| 6.4 | 视频(动画) | 198 | 7.7.3 | Intranet 初步 | 229 |
| 6.4.1 | 基本概念 | 198 | 7.8 | 网络应用的主要方式 | 230 |
| 6.4.2 | 视频格式 | 199 | 7.8.1 | 文件服务器模型与域模型 | 230 |
| 6.5 | 多媒体创作工具的发展与方向 | 200 | 7.8.2 | 客户机/服务器(Client/Server)使用方式 | 231 |
| 6.5.1 | 多媒体创作工具发展概述 | 200 | 7.8.3 | 网络计算初步 | 231 |
| 6.5.2 | 多媒体创作工具技术发展 | 200 | 7.8.4 | 电子商务介绍 | 232 |
| 第 7 章 | 网络基础知识 | 204 | 7.9 | 网络的安全性初步 | 233 |
| 7.1 | 网络发展简史 | 204 | 7.9.1 | 网络安全性的基本概念 | 233 |
| 7.2 | 网络的主要应用与网络的分类 | 206 | 7.9.2 | 网络的信息安全技术初步 | 234 |
| 7.2.1 | 计算机网络的功能 | 206 | 7.9.3 | 防火墙的基本概念 | 235 |
| 7.2.2 | 若干典型的计算机网络应用系统 | 206 | 第 8 章 | 算法与数据结构 | 236 |
| 7.2.3 | 计算机网络的分类 | 208 | 8.1 | 算法设计技术 | 237 |
| 7.3 | 网络协议和标准 | 210 | 8.1.1 | 迭代法 | 237 |
| 7.3.1 | 网络的标准化 | 210 | 8.1.2 | 穷举搜索法 | 238 |
| 7.3.2 | 网络协议模型 | 211 | 8.1.3 | 递推法 | 240 |
| 7.3.3 | ISO/OSI 网络协议层次模型 | 212 | 8.1.4 | 递归 | 241 |
| 7.3.4 | TCP/IP 协议组 | 213 | 8.1.5 | 回溯法 | 246 |
| 7.4 | 常用的网络操作系统 | 214 | 8.1.6 | 贪婪法 | 255 |
| 7.4.1 | 网络操作系统的基本功能 | 214 | 8.1.7 | 分治法 | 259 |
| 7.4.2 | Novell Netware | 214 | 8.1.8 | 动态规划法 | 262 |
| 7.4.3 | Microsoft Windows NT Server | 215 | 8.2 | 线性表 | 264 |
| 7.4.4 | UNIX | 216 | 8.2.1 | 线性表的基本运算和存储 | 264 |
| 7.5 | 构建 LAN 网络 | 217 | | | |
| 7.5.1 | LAN 涉及的主要网络 | | | | |

| | | | | | |
|-------|--------|-----|-------|------------|-----|
| 8.2.2 | 栈 | 270 | 8.4.3 | 二叉查找树 | 306 |
| 8.2.3 | 队列 | 279 | 8.4.4 | 穿线树 | 310 |
| 8.3 | 数组和字符串 | 284 | 8.4.5 | 树形数据结构程序设计 | |
| 8.3.1 | 数组 | 284 | 实例 | 313 | |
| 8.3.2 | 字符串 | 292 | 8.5 | 排序与查找 | 316 |
| 8.4 | 树和二叉树 | 300 | 8.5.1 | 排序 | 316 |
| 8.4.1 | 树 | 301 | 8.5.2 | 查找 | 329 |
| 8.4.2 | 二叉树 | 304 | | | |

第 1 章 计算机硬件基础知识

1.1 计算机系统概述

1.1.1 计算机史简介

自从 1946 年 2 月现代电子计算机的鼻祖 ENIAC (electronic numerical integrator and computer) 在美国宾夕法尼亚大学问世以后,短短 50 年里,计算机技术经历了巨大的变革。习惯上根据计算机系统所采用的硬件技术来划分计算机的发展阶段:

从 1946 年到 50 年代后期(1946~1957)为电子管计算机时期。计算机的元器件主要由电子管(vacuum tube)组成。其特点是体积庞大、功耗高、运算速度较低。如 ENIAC 占地 170m²,重达 30t,功耗为 140kW,有 18000 多个电子管,每秒钟能进行 5000 次加法计算。这一阶段,计算机主要用于军事、国防等尖端技术领域。除了 ENIAC 以外,1945 年左右,冯·诺依曼等人在研制 EDVAC (electronic discrete variable computer) 时,提出了存储程序(stored-program)概念,奠定了以后计算机发展的基石。IBM 公司 1954 年 12 月推出的 IBM650 是第一代计算机的代表。

从 50 年代后期到 60 年代中期(1958~1964)为晶体管计算机时期。自从 1947 年晶体管(transistor)在贝尔实验室诞生后,引发了一场影响深远的电子革命。体积小、功耗低、价格便宜的晶体管取代了电子管,不仅提高了计算机的性能,也使计算机在科研、商业等领域内广泛地应用。第二代计算机不仅采用了晶体管器件,而且存储器改用速度更快的磁芯存储器,与此同时高级编程语言和系统软件的出现,也大大提高了计算机的性能和拓宽其应用领域。这一时期计算机的代表主要有 DEC 公司 1957 年推出的 PDP-I,IBM 公司于 1962 年推出的 7094,以及 CDC 公司 1964 年研制成功的 CDC6600,1969 年 CDC 公司研制的 CDC7600 平均速度达到每秒千万次浮点运算。

从 60 年代中期到 70 年代初期(1965~1971)为集成电路计算机时代。第一代和第二代计算机均采用分离器件(discrete component)组成。集成电路(integrated circuit)的出现,宣告了第三代计算机时代的来临。由于采用了集成电路,使得计算机的制造成本迅速下降;同时因为逻辑和存储器件集成化的封装,大大提高了运行速度;同时功耗也随之下降;集成电路的使用,使得计算机内各部分的互联更加简单和可靠,计算机的体积也进一步缩小。这一时期的代表为 IBM 的 system/360 和 DEC 的 PDP-8。

从 70 年代初期到 70 年代后期(1972~1978)为大规模集成电路(VLSI)计算机时代。70 年代初半导体存储器的出现,迅速取代了磁芯存储器,计算机的存储器向大容量、高速度的方向飞速发展。存储器芯片从 1kbit,4kbit,16kbit,64kbit,256kbit,1Mbit,4Mbit 发展到 16Mbit(1992 年)

随着技术的日新月异,软件和通信的重要性也逐步上升,成为和硬件一样举足轻重的

因素。同时系统结构的特点也对计算机的性能也有巨大的影响(中断系统、Cache 存储器、流水线技术等等)。实际上在第三代计算机以后,就很难找到一个统一的标准进行划分。

1.1.2 计算机系统的结构

计算机的工作过程一般是由用户(程序员)使用各种编程语言把所需要完成的任务以程序的形式提交给计算机,然后翻译成计算机能直接执行的机器语言程序,在计算机上运行以实现用户需要的功能。

计算机系统可以看作是一个多层的系统,不同的用户(或者应用)与不同的层次打交道。通常把计算机系统分作4层:物理机器(机器语言机器)、操作系统语言机器、汇编语言机器和高级语言机器。如图 1.1 所示。第1层实际上就是由硬件组成的计算机本身,它所能理解和执行的是机器语言。第2~4层其实都是“虚拟”机器,通过软件组成的层次结构,隐藏了机器的硬件实现细节。用户在其上见到的“机器”实际上是在物理机器上运行的软件表示的虚拟机器。

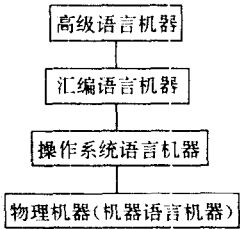


图 1.1 计算机系统的层次结构

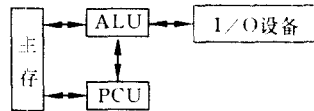


图 1.2 IAS 计算机结构

1945年冯·诺依曼在设计EDVAC时提出了存储程序的概念,后来他和同事合作设计的存储程序计算机IAS的结构被其后几乎所有的通用计算机采用。这种结构也就称为冯·诺依曼机,如图 1.2 所示。

其中主存(main memory)用以存放数据和程序;算术逻辑运算单元ALU(arithmetic-logical unit)完成二进制算术和逻辑运算;程序控制单元PCU(program control unit)负责解释和执行指令;I/O设备由PCU控制,实现和外部环境的数据交换。

1.2 数据的计算机表示

1.2.1 数制

计算机的存储器(memory)和寄存器(register)是两态器件,因此,各种信息是以二进制的方式在计算机里存储和运算。但是为了方便起见,有时也需采用不同的数制(number system)来表示数据,例如十进制、十六进制、八进制等。

1. 数制介绍

一种数制由基数 r (radix)和 r 个不同的数码(symbol)组成。任意一个数用 r 进制可以表示为:

$$\begin{aligned}
 N &= (R_n R_{n-1} \cdots R_1 R_0 R_{-1} \cdots R_{-m}) \\
 &= R_n * r^n + R_{n-1} * r^{n-1} + \cdots + R_1 * r + R_0 + R_{-1} * r^{-1} + \cdots + R_{-m} * r^{-m}
 \end{aligned}
 \tag{1}$$

对于十进制数而言共有 10 个数码:0,1,2,3,4,5,6,7,8,9。十进制数 538.4 即表示 $5 \times 10^2 + 3 \times 10^1 + 8 \times 10^0 + 4 \times 10^{-1}$ 。对于八进制数而言 8 个数码为 0~7;而十六进制的 16 个数码分别为 0~F。二进制的基数为 2,数码为 0 和 1。

公式(1)称为数 N 在 R 进制中的按权(*power*)展开式。

【例 1.1】 把八进制数 1704.25 转化为十进制数。

$$\begin{aligned}
 \text{解: } (1704.25)_8 &= 1 \times 8^3 + 7 \times 8^2 + 0 \times 8^1 + 4 \times 8^0 + 2 \times 8^{-1} + 5 \times 8^{-2} \\
 &= 1 \times 512 + 7 \times 64 + 4 \times 1 + 2 \times 0.125 + 5 \times 0.015625 \\
 &= (964.328125)_{10}
 \end{aligned}$$

【例 1.2】 把十六进制数 2EC.F 转化为十进制数。

$$\begin{aligned}
 \text{解: } (2EC.F)_{16} &= 2 \times 16^2 + 14 \times 16^1 + 12 \times 16^0 + 15 \times 16^{-1} \\
 &= 2 \times 256 + 14 \times 16 + 12 \times 1 + 15 \times 0.625 \\
 &= (748.9375)_{10}
 \end{aligned}$$

表 1.1 列出了常用的十、十六、八和二进制数的对照表。

表 1.1 常用进制对照表

| 十进制 | 十六进制 | 八进制 | 二进制 |
|-----|------|-----|-------|
| 0 | 0 | 0 | 0 |
| 1 | 1 | 1 | 1 |
| 2 | 2 | 2 | 10 |
| 3 | 3 | 3 | 11 |
| 4 | 4 | 4 | 100 |
| 5 | 5 | 5 | 101 |
| 6 | 6 | 6 | 110 |
| 7 | 7 | 7 | 111 |
| 8 | 8 | 10 | 1000 |
| 9 | 9 | 11 | 1001 |
| 10 | A | 12 | 1010 |
| 11 | B | 13 | 1011 |
| 12 | C | 14 | 1100 |
| 13 | D | 15 | 1101 |
| 14 | E | 16 | 1110 |
| 15 | F | 17 | 1111 |
| 16 | 10 | 20 | 10000 |

2. 数制间的转换

(1) 从十进制向 r 进制转换

把 1 个十进制数转换为等价的 r 进制数应分成 2 步进行: 首先把该数的整数部分和小数部分分别转换为对应的 r 进制数; 然后再把这两部分合起来即可。

算法 1: (把十进制整数 I 转换为对应的 r 进制数)

(1) $k := 1$

(2) 把 I 除以 r 得到商 Q 和余数 R_k ; $I := Q$; $k := k + 1$

(3) 若 $I = 0$ 则算法停止, 否则转到(2)

(4) 把 R_k 从高到低排列起来就是所求结果。

【例 1.3】 把 $(307)_{10}$ 转换为对应的二进制数。

解: 商(Q)..... 余数(R_k)

| | |
|-----|---|
| 307 | |
| 153 | 1 |
| 76 | 1 |
| 38 | 0 |
| 19 | 0 |
| 9 | 1 |
| 4 | 1 |
| 2 | 0 |
| 1 | 0 |
| 0 | 1 |

得 $(307)_{10} = (100110011)_2$

【例 1.4】 把 $(307)_{10}$ 转换为八进制数。

解: 商(Q)..... 余数(R_k)

| | |
|-----|---|
| 307 | |
| 38 | 3 |
| 4 | 6 |
| 0 | 4 |

得 $(307)_{10} = (463)_8$

【例 1.5】 把 $(307)_{10}$ 转换为十六进制数。

解: 商(Q)..... 余数(R_k)

| | |
|-----|---|
| 307 | |
| 19 | 3 |
| 1 | 3 |
| 0 | 1 |

得 $(307)_{10} = (133)_{16}$

算法 2: (把十进制小数 F 转换为 r 进制数)

(1) $k := -1$;

- ② F 乘以 r 得到整数部分 R_k 和小数部分 F' ;
 ③ $F := F'$; $k := k + 1$
 ④ 若 $F = 0$ 或者 K 达到规定的精度位数则转到⑤, 否则转至②;
 ⑤ 把 $R_1 R_2 \cdots R_m$ 排列起来, 其中 R_1 是最靠近小数点的位。

【例 1.6】把 $(0.8125)_{10}$ 转换为二进制数。

解:

$$\begin{array}{r}
 0.8125 \\
 \times 2 \\
 \hline
 1 \quad .6250 \\
 \times 2 \\
 \hline
 1 \quad .250 \\
 \times 2 \\
 \hline
 0 \quad .50 \\
 \times 2 \\
 \hline
 1 \quad .0
 \end{array}$$

得 $(0.8125)_{10} = (0.1101)_2$

【例 1.7】把 $(0.8125)_{10}$ 转换为十六进制和八进制数。

解:

$$\begin{array}{r}
 C.8125 \\
 \times 16 \\
 \hline
 D \quad .0
 \end{array}
 \qquad
 \begin{array}{r}
 0.8125 \\
 \times 8 \\
 \hline
 6 \quad .50 \\
 \times 8 \\
 \hline
 4 \quad .0
 \end{array}$$

得 $(0.8125)_{10} = (0.D)_{16} = (0.64)_8$

(2) 二进制数 / 八进制数 / 十六进制数之间的相互转换

因为 $2^3=8, 2^4=16$, 所以每一位八进制数码都对应着一个 3 位二进制数, 每一位十六进制数码则对应着一个 4 位二进制数。如表 1.2 和表 1.3 所示。

表 1.2 八进制数和二进制数对应表

| | | | | | | | |
|-----|-----|-----|-----|-----|-----|-----|-----|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 000 | 001 | 010 | 011 | 100 | 101 | 110 | 111 |

表 1.3 十六进制数和二进制数对应表

| | | | | | | | |
|------|------|------|------|------|------|------|------|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 0000 | 0001 | 0010 | 0011 | 0100 | 0101 | 0110 | 0111 |
| 8 | 9 | A | B | C | D | E | F |
| 1000 | 1001 | 1010 | 1011 | 1100 | 1101 | 1110 | 1111 |

二进制向八进制转换的方法为：从小数点开始，分别向左、右每 3 位二进制数编成一组，若不够 3 位，则小数点最左侧和最右侧用 0 补充；每一组用对应的八进制数码表示即可，八进制向二进制转换的方法为：从小数点开始，把每一位八进制数码转换成对应的 3 位二进制数即可，其小数点最左侧或最右侧的 0（如果有的话）可以省去。二进制和十六进制数之间的转换与此类似，只不过每组的二进制位数是 4 而不是 3。

【例 1.8】 把 $(273.64)_8$ 转换为二进制数。

解： 2 7 3 . 6 4
 010 111 011 . 110 100
 得 $(273.64)_{10} = (10111011.1101)_2$

【例 1.9】 把 $(7CE2.9A)_{16}$ 转换为二进制数。

解： 7 C E 2 . 9 A
 0111 1100 1110 0010 . 1001 1010
 得 $(7CE2.9A)_{16} = (111110011100010.1001101)_2$

在计算机中八进制数、十六进制数通常被认为是二进制数的另一种表示方法。

(3) 特殊方法

① 由于八进制与二进制之间的转换十分简单，因此有时十进制与二进制之间的转换可以以八进制为中介。

$$(273.71875)_{10} = (421.56)_8 = (100010001.10111)_2$$

② 如果一个分数的分母为 2 的整数次幂则可以用以下方法转换。

$$(3/32)_{10} = 3 \times 2^{-5} = (11)_2 \times (0.00001)_2 = (0.00011)_2$$

③ 从十进制向八进制/十六进制转换可以先把十进制转换为二进制，然后再从二进制转换为八进制/十六进制。

$$(27.625)_{10} = (11011.101)_2 = (33.5)_8 = (1B.A)_{16}$$

1.2.2 数值数据的表示方法

因为当前的计算机所采用的存储器件都是两态器件，所以适合于存放二进制数据。尽管为了方便起见，也使用其他进制，如十进制、十六进制；但是在计算机内部数据还是以二进制的形式存放和处理。二进制只使用了两个不同的数字符号，易于用物理器件来实现：在物理世界中具有两个稳定状态的物理器件很多，如晶体管的“截止”与“导通”、电容的“充电”与“放电”、电压信号的“高”与“低”、脉冲的“有”与“无”，电磁单元的“正向磁化”与“反向磁化”等等，只要规定其中一种稳定状态表示“1”，另一种稳定状态表示“0”，就可以用来表示二进制数位了。其次，二进制的运算规则非常简单，易于用电子器件来实现。

计算机里用到的数据类型主要可分为两类：表示数量的数值数据和非数值性的符号数据。所有的数据都是以二进制的形式在计算机里处理和存储。本节主要讲述数值数据在计算机里的表示方法。

1. 符号位的处理

正数和 0 都可以表示为无符号数，而负数通常是用“-”号来表示。但在计算机里，任何符号都是用“1”和“0”来表示的。所以，习惯上计算机里都用最高位（最左位）来表示数的

符号: 0——正数, 1——负数。

例如, $X_1 = (+0010010)_2$ 和 $X_2 = (-0010010)_2$ 在计算机里表示为 00010010 和 10010010。

2. 小数点的处理

计算机在处理数值数据时, 对小数点的处理有两种不同的方法: 定点法和浮点法, 也就对应了两种不同形式的数据表示方法。

(1) 定点数据表示方法

在这种表示方法中, 小数点的位置固定不变。根据其所在位置的不同, 分别用来表示整数和纯小数:

• 定点整数格式: $N = b_s b_m b_{m-1} \cdots b_1 b_0$

小数点位置在最低数据位 b_0 的右边, b_s 为符号位, b_m 为其最高数据位, 其取值范围为: $|N| \leq 2^m - 1$

• 定点小数格式: $N = b_s . b_{-1} b_{-2} \cdots b_{-n}$

小数点位置固定在符号位 b_s 与最高数据位 b_{-1} 之间。其取值范围 $|N| \leq 1 - 2^{-n}$ 。

需要注意的是, 无论是在定点整数还是定点小数表示法中, 小数点作为一个符号并不明确地表示出来, 而是由其表示方法所隐含。例如:

$N = 10111010$, 若表示定点小数则其值为 -0.011101 ; 若表示定点整数, 则其值为 -111010 。

(2) 浮点数据表示方法

这种表示方法把数值数据表示成:

$$N = M \cdot R^E$$

其中, M 被称为 N 的尾数, E 是 N 的指数或称阶码, 而 R 则是该阶码的基数。在这种表示方法下, 同一个数可以有不同浮点表示形式, 阶码的大小可以用来调节数值中小数点的位置。例如: 0.081 可以表示为 $81E-3$, 也可以表示为 $8.1E-2$, 或者 $0.81E-1$ 等等。在这种意义上, 其小数点位置是可以“浮动”的, 所以把该方法称为浮点数据表示法。

需要指出的是在我们上面的表示方法中, 其基数 10 并未出现在表示形式中, 而是隐含规定的。在计算机中, 基数 R 通常由硬件部件(浮点运算部件)隐含规定, 也不明确地出现在表达式中。在计算机里浮点数通常以如下形式表示:

| | | |
|-------|-----|-----|
| M_s | E | M |
|-------|-----|-----|

M_s : 尾数 M 的符号位, 其位数为 1。

M : 用定点小数表示的尾数, 其位数 n 决定了该浮点数的精度。

E : 用定点整数表示的阶码, 其位数 m 则决定该浮点数的取值范围。

IEEE 规定了 3 种常用的浮点数格式: 短浮点数格式, 长浮点数格式以及临时浮点数格式, 见表 1.4。

临时浮点数主要是用来表示运算的中间结果, 以保证运算的精度。

3. 定点数据编码方法

表 1.4 3 种常用的浮点数格式表

| | 符号位 | 阶码 | 尾数 | 总位数 |
|-------|-----|----|----|-----|
| 短浮点数 | 1 | 8 | 23 | 32 |
| 长浮点数 | 1 | 11 | 52 | 64 |
| 临时浮点数 | 1 | 15 | 64 | 80 |

对于数值数据而言,无论以定点形式表示或是以浮点形式表示,都需要通过某种编码方法,以计算机所能处理的“0”和“1”二进制位来表示其数值大小。通常一个数值数据的机内表示形式称为其机器码,而一个机器码所代表的数值称为该机器码的真值。

我们假设机器码的宽度为 n , 最高位为符号位, 下面分别介绍实际应用中常见的 4 种编码方法: 原码、反码、补码和移码(又称增码)。

(1) 原码表示法

• 定点整数原码表示法

定点整数原码的定义如下所示:

$$[X] = \begin{cases} X & 0 \leq X < 2^{n-1} \\ 2^{n-1} - X & -2^{n-1} < X \leq 0 \end{cases}$$

由定义可以看出,正整数的原码就是其自身,而负整数的原码只需把其绝对值的原码的符号位置为 1 即可(0 表示正号,1 表示负号)。原码表示法的取值范围为 $-(2^{n-1}-1) \leq X \leq 2^{n-1}-1$ 。

注意在原码表示法中,零有正零和负零之分。 $[+0]=0000\cdots 0$, $[-0]=1000\cdots 0$ 。

• 定点小数原码表示法

定点小数原码的定义如下所示:

$$[X] = \begin{cases} X & 0 \leq X < 1 \\ 1 - X & -1 < X \leq 0 \end{cases}$$

正的纯小数的原码就是其自身,而负的纯小数的原码可以通过将其绝对值的原码的符号位置 1 来求。其取值范围为 $-(1-2^{-(n-1)}) \leq X \leq (1-2^{-(n-1)})$ 。

综上所述可以看出一个数的原码是由符号位加上数值位绝对值组成,符号位为 0 代表正数,符号位为 1 代表负数。由于 0 有 $+0$ 和 -0 两种表示方法, n 位二进制数可以表示 $2^n - 1$ 个原码。

用原码实现乘除运算规则较简单,但做加减运算不方便。原码表示法的一个主要优点在于其真值和编码表示之间对应关系很直观,容易转换。

(2) 补码表示法

• 定点整数的补码表示

定点整数补码的定义如下所示:

$$[X] = \begin{cases} X & 0 \leq X < 2^{n-1} \\ 2^n + X & -2^{n-1} \leq X < 0 \end{cases}$$

正整数的补码就是其自身,负整数的补码可以通过对其绝对值部分逐位求反,并在最