

邹繁伟 张川 编著

实用 微机 软件 手册

电子科技大学出版社

- DOS 6.0
- Windows 3.1
- C 和 C++ 语言
- Foxpro 2.5 数据库
- Novell 网络

实用微机软件手册

邹繁伟 张 川 编著

电子科技大学出版社

• 1994 •

[川]新登字 016 号

实用微机软件手册

邹繁伟 张 川 编著

*

电子科技大学出版社出版

(成都建设北路二段四号)邮编 610054

郫县印刷厂印刷

新华书店经销

*

开本 787×1092 1/16 印张 20 字数 500 千字

版次 1994 年 12 月第一版 印次 1994 年 12 月第一次印刷

印数 1—5000 册

中国标准书号 ISBN 7-81016-721-9/TP·54

定价: 18.00 元

内 容 简 介

这是一本精心整理编撰的计算机工具书,它取材于计算机基础知识、大量最新高级软件的原文资料和最新软件产品测评。内容丰富、精练、实用性强。

全书紧紧围绕计算机的基础知识、实际应用和发展方向,在介绍微型计算机软件基础知识的基础上,着重对目前广泛使用和新近推出的高级版本软件和实用工具进行了介绍。如:MS-DOS6.0操作系统;Microsoft Windows 3.1基本操作技能;文字处理:WS、DFEDIT、PE、vi、WPS、CCED文字处理软件的操作方法;关系数据库原理,数据库设计,FoxPro 2.5 for Windows数据库的主要技术特点、每条函数和命令参考;计算机网络原理及其具体操作方法;ORACLE和SYBASE数据库介绍;C和Borland C++3.1、PCTOOLS 8版操作导航;LOTUS 1-2-3、SC3、OFFICE主要命令和函数等。还有丰富的附录备查,帮助读者提高效率。

本书既注意了应用的广度,又具有一定的深度,可作为大专院校师生、培训班、从事计算机程序设计及计算机应用的技术人员学习的参考书和工具书。

前 言

当今世界,计算机无处不在,学习使用计算机的人越来越多,人们花了大量的金钱和时间投入到计算机的学习、应用和开发之中,却苦于计算机所涉及的知识太多、发展速度太快,无形之中产生了一种学不赢、赶不上的感觉。为帮助用户消除这些苦恼与麻烦,我们根据自己多年学习、工作、教学的经验和收集的资料,经过精心整理编撰了本手册。

本手册收集了目前广泛使用和刚推出的高级新版系统软件和实用工具。由浅入深地系统介绍了应用个人系列计算机的基本技术知识,既能使初学者在较短时间内掌握基本操作技能,学习原理、了解软件发展趋势,又能为有经验的朋友提供查询。

本手册分十三个部分描述。

一、软件基础知识:对于初学者来说,这似乎不太重要,但只要入了门,接着而来的问题就是怎样使用或设计程序,怎样维护和优化系统,这时用户可根据自己的实际选择相应的内容进行学习,为发展自己奠定基础。

二、DOS 操作系统:每个计算机用户都离不开使用系统资源,学会和使用 DOS 操作系统有着重要的实际意义。DOS5.0 是目前广泛使用的版本,为便于学习和查阅,本书按命令的字母顺序列出其条目、格式和功能。DOS6.0 是 1993 年新推出的最优版本,功能丰富,本书特将其命令文件分类描述,并将其新功能做了具体介绍。只要您借助于 HELP 命令,试用感兴趣的命令,一定会有收获。

三、Microsoft Windows 简介和基本操作技能:窗口、菜单、对话框给用户提供了易于接受的界面,Microsoft Windows 极为人性化的操作方法最为引人注目。其窗口环境具有风格一致的界面,所有在 Windows 环境下开发的应用软件也都具有相同的风格。目前广泛使用的诸如:MS-DOS5.0 以上版本的磁盘操作系统、PCTOOLS 实用工具、FoxPro 系列数据库、Borland 和 Microsoft C++ 语言等高级通用软件,都普遍采用了与 Microsoft Windows 键盘和鼠标兼容的操作规范。学会了一种软件的操作,其它软件也很容易学会,大大缩短了用户学习软件所需要的时间。

四、计算机网络:共享资源是衡量计算机应用水平的一个重要指标。计算机网络就像电话一样给人们的生活带来极大的方便,因此掌握计算机网络的一些最基本、最常用的概念是十分有益的。本章将介绍如层、协议、连接和无连接服务、网际等概念。重点介绍了 OSI 的七层标准体系结构,局域网以及局域网的实例——Novell 386 V3.11。

五、UNIX System V/386 Release 4 系统构成基本软件命令:人们的求知欲是无止境的,很想看看 DOS 以外的天地。UNIX 操作系统的强功能、灵活性和精致性使得它成为最受欢迎的开放系统。本章列出用户与程序开发环境命令和功能以飨读者。

六、文字处理:这是用户使用频率最高的领域,不管是应用还是开发几乎天天都要和文字处理软件打交道。丰富多彩的排版系统使人眼花缭乱,纵观博览,总觉得还是 WS 用起来顺手,DFEDIT、PE、vi、WPS、CCED 这些字处理软件也各有自己的长处,用户可任意选用。

七、关系数据库原理:这是一个比较难做,也是用户最有兴趣的课题。数据库广泛用于

信息处理,为了解决业务中的实际问题,经常需要应用和开发数据库管理程序(或系统)许多“半路出家”的用户一般以 dBASE III 引路,在取得大量的实践经验后再学习其他高级数据库系统,这种思路是行之有效的。但他们往往忽视数据库原理的学习,结果在洋洋万言之后,才发现在动手之初就犯了大错误,真是痛心疾首。

本书从数据库管理的目标出发,系统地描述了数据库的基本原理,最后以 E. F. Codd 的关系型 DBMS 的判别准则结束。

八、数据库开发技术:软件是一种非实体物,软件的开发又是一个思维的过程。用户的业务在不断地发生变化,市场上很难买到这种“可以应变”的管理信息系统,确实是硬件易买,软件难求。为了解决这个问题,往往求助于科研单位或大专院校。

怎样向专家表达自己的意图呢?一种有效的办法就是亲自了解一些软件开发技术,用专家易懂的表达方式表达自己的意图,有了共同语言,合作起来高效、愉快!

九、FoxPro 数据库: dBASE III 曾为数据库的普及应用立下了汗马功劳,但现在它已被 FoxPro 彻底取代了。遗憾的是有关 FoxPro 2.5 的完整资料目前还不多见,细心的读者在本章中可以看到,反映 FoxPro 2.5 的高级功能得到了充分的体现,这对应用和开发人员来说,是一份不可多得的宝贵资料。

由于 FoxPro 2.5 极大地丰富了 xBASE 语言,所以其命令和函数显得非常庞大,用户应根据自己的实际,先选用简单的功能,待熟悉之后,再选用复杂的部份,使之能充分展现自己的聪明才智。

十、ORACLE 和 SYBASE 数据库系统介绍:这是两种很有代表性的高性能关系数据库系统,能运行在多种工作平台。在计算机硬件高速发展,价格快速持续下降的今天,在个人计算机上应用、开发 ORACLE 和 SYBASE 资源已达到实用阶段。适时地引入高性能关系数据库系统对提高企业经营手段有着重要的意义。

十一、C 和 C++ 语言:程序设计方法,已从“结构”进化到“面向对象”的程序设计。在所有的高级语言中,C 易于表达且比较自由,C++ 更易于表达,是目前最强大的面向对象的通用程序设计语言。越来越多的用户从不同的层次进(转)入 C 或 C++ 的学习和开发,亲身体验尖端技术的真谛,从中吸取新思想、新方法,编写出比传统的(非面向对象的)程序更加结构化、更可重用、更可扩充、更易于维护的程序。

十二、PCTOOLS:这是大家都感兴趣的工具集,它功能强大、完备,使用简便、高效,受到广大用户的青睐。许多中外独(和合)资企业在招聘条件中有一条“能熟练使用 PCTOOLS”,这足以说明其在业务活动中的重要性。

十三、LOTUS1-2-3、SC3、OFFICE 主要命令和函数:这些是使用最简单、有较长历史的表格软件,在初学计算机时不访用一用,为将来使用 and 开发更高级软件开拓思路。

本手册还有丰富的附录备查使您得心应手。

限于水平,书中难免出现缺点和疏漏,恳请读者批评指正。

作者

一九九四年六月

目 录

第一章 软件基础知识

§ 1.1 数据结构	1
§ 1.1.1 基本概念和术语	1
§ 1.1.2 基本的数据结构	2
§ 1.2 操作系统	3
§ 1.2.1 操作系统概述	3
§ 1.2.2 DOS 操作系统	5
§ 1.3 程序设计语言	5
§ 1.3.1 程序语言的组成	5
§ 1.3.2 语言和程序设计的发展	6
§ 1.3.3 程序语言的使用	6
§ 1.4 程序设计基础	6
§ 1.4.1 程序设计概念	6
§ 1.4.2 问题分析	7
§ 1.4.3 算法设计	7
§ 1.5 结构化程序设计与程序设计风格	8
§ 1.5.1 结构化程序	8
§ 1.5.2 程序设计风格	8
§ 1.6 计算机病毒	9
§ 1.7 C 和 C++ 简介	10
§ 1.7.1 面向程序员的 C	10
§ 1.7.2 面向对象的 C++	10
§ 1.7.3 怎样入手学习面向对象的 C++	11
§ 1.8 文件分类	12

第二章 DOS 操作系统

§ 2.1 DOS5.0 操作系统基本命令	14
§ 2.2 DOS6.0 操作系统的文件及其注释	17
§ 2.3 DOS6.0 操作系统的新功能介绍	20

第三章 Microsoft Windows 简介和基本操作技能

§ 3.1 Microsoft Windows 简介	24
§ 3.2 Microsoft Windows3.1 的基本组成	26

§ 3.2.1 窗口模式.....	27
§ 3.2.2 窗口的组成部分.....	27
§ 3.2.3 肖像类型.....	28
§ 3.3 基本鼠标和键盘操作.....	29
§ 3.4 菜单、窗口、对话框、操作表参考信息	29
§ 3.5 对话框操作表参考信息.....	31
§ 3.6 Windows Help 操作表参考信息.....	32
§ 3.7 Windows 3.1 操作表	33

第四章 计算机网络

§ 4.1 预备知识.....	39
§ 4.2 计算机网络的体系结构.....	40
§ 4.3 计算机网络的拓扑结构.....	44
§ 4.4 计算机网络的传输媒质.....	44
§ 4.5 局域网标准.....	45
§ 4.6 Novell 网	48
§ 4.6.1 Novell 技术关键	48
§ 4.6.2 NETWARE 网络系统分析.....	49
§ 4.7 Novell 网络安装	52
§ 4.7.1 设置硬件.....	52
§ 4.7.2 文件服务器引导.....	55
§ 4.7.3 装入磁盘驱动程序的可装入模块.....	57
§ 4.7.4 装入 LAN 驱动程序和别的可装入模块	59
§ 4.7.5 安装 DOS 工作站	62
§ 4.8 网络互联.....	66
§ 4.9 网络互联设备.....	68
§ 4.10 网络管理	70

第五章 UNIX System V/386 Release 4 命令表

§ 5.1 用户命令.....	72
§ 5.2 程序开发环境命令.....	77

第六章 文字处理

§ 6.1 Wordstar	79
§ 6.1.1 功能键.....	79
§ 6.1.2 命令菜单.....	79
§ 6.2 DFEDIT(Directory File Editor)目录文件编辑器	82
§ 6.3 PE(Personal Editor)个人(专用)编辑器	83
§ 6.4 vi(visual)直观正文编辑程序基本命令	84

§ 6.5 WPS(北大方正 Super V 型汉卡)操作指南	86
§ 6.5.1 预备知识	86
§ 6.5.2 基本操作方法	87
§ 6.5.3 北大方正 Super V 型汉卡 WPS 命令速查表	88
§ 6.6 CCED	92

第七章 数据库原理

§ 7.1 基本概念	95
§ 7.2 关系模型系统	97
§ 7.2.1 关系模型综述	97
§ 7.2.2 关系模型的基本概念	98
§ 7.2.3 关系模型的操纵语言	99
§ 7.3 关系数据库设计	101
§ 7.4 数据库保护	102
§ 7.5 数据库系统	103
§ 7.6 关系型 DBMS 的判别准则及其简单评价方法	103
§ 7.6.1 关系型 DBMS 的十二条判别准则	103
§ 7.6.2 一种简单评价方法	104

第八章 数据库开发技术

§ 8.1 项目确定与工作计划制定	106
§ 8.2 数据库设计的需求分析	106
§ 8.3 数据库的概念设计	107
§ 8.4 数据库的逻辑设计	108
§ 8.5 数据库的分布设计	110
§ 8.6 数据库的物理设计	111

第九章 FoxPro 关系数据库

§ 9.1 FoxPro 简史	113
§ 9.2 预备知识	114
§ 9.2.1 一般问题	114
§ 9.2.2 FoxPro 2.5 的新特点	116
§ 9.2.3 FoxPro 的配套产品	117
§ 9.3 FoxPro2.5 提供的工具	118
§ 9.3.1 最终用户工具(9个)	118
§ 9.3.2 开发者工具(10个)	120
§ 9.4 FoxPro2.5 的查询技术	122
§ 9.4.1 RQBE(关系型按例查询)	122
§ 9.4.2 Rushmore Technology	123

§ 9.4.3	SELECT	126
§ 9.4.4	SQL Cursors	126
§ 9.5	FoxPro2.5 数据语言中使用的缩写与主要子句	127
§ 9.5.1	缩写	127
§ 9.5.2	主要子句	128
§ 9.6	FoxPro 2.5 数据库语言分类	131
§ 9.6.1	概述	131
§ 9.6.2	分类描述	132
§ 9.7	FoxPro 2.5 数据库语言	156
§ 9.7.1	FoxPro 2.5 函数(289)	157
§ 9.7.2	FoxPro 2.5 命令(320)	169
§ 9.7.3	系统内存变量	203

第十章 ORACLE 和 SYBASE 数据库介绍

§ 10.1	ORACLE RDBMS 的特点	206
§ 10.2	ORACLE 分布式处理环境	207
§ 10.3	ORACLE 体系结构	208
§ 10.4	ORACLE 软件产品	209
§ 10.4.1	ORACLE 软件产品	209
§ 10.4.2	ORACLE 软件配置	213
§ 10.4.3	ORACLE 软件产品结构	214
§ 10.5	ORACLE 软件的历史、现状和发展	214
§ 10.5.1	ORACLE 软件的历史	214
§ 10.5.2	ORACLE 7.0 版的新特性	215
§ 10.6	SYBASE 数据库简介	219
§ 10.6.1	SYBASE SQL Server	220
§ 10.6.2	SYBASE SQL Toolset	221
§ 10.6.3	满足在线(On-Line)事务处理的高性能	223

第十一章 C 和 C++ 语言

§ 11.1	C 语言的历史与现状	225
§ 11.2	C 和 C++ 语言程序的格式、特点和开发过程	225
§ 11.2.1	格式	225
§ 11.2.2	特点	226
§ 11.2.3	开发过程	226
§ 11.2.4	C 的内存图	227
§ 11.3	数据与运算	227
§ 11.3.1	标点和特殊符	227
§ 11.3.2	转义符序列	228

§ 11.3.3 基本数据类型和值域(ANSI 标准中的所有数据类型)	228
§ 11.3.4 变量	229
§ 11.3.5 表达式	230
§ 11.3.6 运算符和运算优先级	230
§ 11.4 修饰符	232
§ 11.5 程序控制语句	232
§ 11.6 C++	234
§ 11.7 关于进一步学习 C++ 的建议	235
§ 11.8 Borland C++3.1 的 IDE(集成开发环境)	235
 第十二章 PCTOOLS 实用工具	
§ 12.1 PC Tools 7 版 PC Shell 命令	241
§ 12.2 PC Tools 8 版 Desktop (桌面)	244
§ 12.3 PC Tools 8 版 Desktop Accessories (桌面辅助)	249
§ 12.3.1 Notepads (便笺)	249
§ 12.3.2 Outline (提纲)	251
§ 12.3.3 Clipboard (裁剪板)	252
§ 12.3.4 Appointment (约会计划)	252
§ 12.4 PC Tools 8 版 Central Point Anti-Virus (集成反病毒)	253
§ 12.5 PC Tools 8 版 CP Backup (备份)	255
§ 12.6 DiskFix (磁盘修理)	257
 第十三章 LOTUS1-2-3、SC3、OFFICE 主要命令和函数	
§ 13.1 LOTUS1-2-3	258
§ 13.2 C-SuperCalc 电子报表	262
§ 13.3 OFFICE 命令集	264
附录 1 ASCII 码对照表	266
附录 2 区位码表	267
附录 3 DOS5.0 键盘功能表	268
附录 4 Microsoft Windows 3.1 键盘功能表	271
附录 5 FoxPro 2.5 参考表	280
附录 6 ORACLE 数据库管理系统数据库设计	292
附录 7 2.13H 汉字系统参考表	305
参考文献	307

第一章 软件基础知识

§ 1.1 数据结构

§ 1.1.1 基本概念和术语

数据(Data):对客观事物的符号表示,在计算机科学中,是指所有能输入到计算机并被计算机程序处理的符号的总称。数据的含义极为广泛,如图象声音等都可以通过编码而归之于数据的范畴。

数据元素(Data element):数据的基本单位,在计算机程序中通常作为一个整体进行考虑和处理。如:一个记录。有时一个数据元素可由若干个数据项组成,数据项是数据的最小单位

数据对象(Data object):性质相同的数据元素的集合,是数据的一个子集。

数据结构(Data structure):相互之间存在一种或多种特定关系的数据元素的集合。

“关系”描述的是元素之间的逻辑关系,因此又称为数据的逻辑结构。数据结构在计算机中的表示(也称映象)叫数据的物理结构,又称存储结构。

数据类型(Data type):程序语言中所允许使用的变量或表达式的种类。换句话说,是变量或表达式可能取值的范围和能作运算的集合,即是数据结构及其运算的总称。

抽象数据类型(Abstract data type):是指一组模型以及定义在该模型上的一组操作。

一个软件系统的框架应建立在数据之上,而不是建立在操作之上。即在构成软件系统的每个相对独立的模块上,定义一组数据和施于这些数据上的一组操作,并在模块内部给出这些数据的表示及其操作细节,而在模块外部使用的只是抽象的数据和抽象的操作。显然,所定义的数据类型的抽象层次越高,含有该抽象数据类型的软件模块的复用程度也就越高。

对象(Object):对象是指一个实体,是对现实世界的直接映射。从存储角度看,对象是一片私有存储,其中有数据也有方法(操作)。其他对象的方法不能直接操纵该对象的私有数据,只有对象私有的方法才可操纵它。所有对对象的处理都要通过对象向外部提供方法(操作)进行,对方法(操作)的引用是通过所谓的消息传递来进行的。这样,对象的概念就很好地提供了信息隐蔽机制,实现了模块化。

类(Class):定义的是一种对象类型,它描述了属于该类型的所有对象的性质。每一个对象都属于一个特定的类,称为该类的实例。

“类型”的概念与“操作”是密切相关的,同一种数据结构和不同的操作组将构成不同的数据类型。在面向对象的程序设计语言中,结构说明和过程说明被统一在一个整体——“对象”之中。其中数据结构的定义为对象的属性域,过程或函数定义在对象中称之为方法(method),是对象的性能描述(即能进行任何操作)。

多态数据类型(Polymorphic data type):值的成分不确定的数据类型。

数据的基本操作:

- 插入: 在数据结构中的指定位置上增添新的数据元素;
- 删除: 删去数据结构中某个指定的数据元素;
- 更新: 改变数据结构中某个数据元素的值, 在概念上等价于删除和插入操作的组合;
- 查找: 在数据结构中寻找满足某个特定要求的数据元素(的位置的和值);
- 排序: (在线性结构中)重新安排数据元素之间的逻辑顺序关系, 使之按值由小到大或由大到小的次序排列。

算法(algorithm): 对特定问题求解步骤的一种描述, 它是指令的有限序列, 其中每一条指令表示一个或多个操作。算法具有以下五个重要特性:

- 有穷性: 一个算法必须是(对任何合法的输入值)在执行有穷步之后结束, 且每一步都可在有穷时间内完成;
- 确定性: 算法中的每一条指令必须有确切的含义, 读者理解时不会产生二义性。并在任何条件下, 算法只有唯一的一条执行路径, 即对相同的输入只能得出相同的输出;
- 可行性: 一个算法是可行的。即算法中描述的操作都是可以通过已经实现的基本运算执行有限次来实现的;
- 输入: 一个算法有零(或多)个输入, 它(们)取自于特定的对象的集合;
- 输出: 一个算法有零(或多)个输出, 它(们)同输入有某个特定关系的量。

§ 1.1.2 基本的数据结构

● **线性表(Linear list):** n 个数据元素的有限序列。在复杂的线性表中, 一个数据元素可以由若干个数据项(Item)组成。在这种情况下, 常把数据元素称为记录(Record), 含有大量记录的线性表又称为文件(File)。

线性表的基本操作: 初始化、求长度、取元素、求前驱、求后继、定位前插、删除、判空、置空。

● **栈(Stack):** 限定仅在表尾进行插入或删除操作的线性表(满足后进先出)。表尾端称为栈顶(Top), 表头端称为栈底(Bottom)。不含元素的空表称为空栈。

栈的基本操作: 初始化、判空、入、出、取顶元素、置空、元素个数。

● **队列(Queue):** 与栈相反, 队列是一种先进先出(First in first out)的线性表。它只允许在表的一端进行插入, 而在另一端删除元素。在队列中, 允许插入的一端叫做队尾(Rear), 允许删除一端则称为队头(Front)。

双端队列是限定插入和删除操作在表的两端进行的线性表。

用链表表示的队列叫链队列

队列的基本操作: 判断、入、出、取头元素、置空。

● **串(String):** 是由零(或多)个字符组成的有限序列。

一般记为: $s = 'a_1a_2 \dots a_n'$ ($n \geq 0$)

其中, s 是串名, 单引号括起来的字符序列是串的值, n 称为串的长度, 零个字符的串叫空串。串中任意个连续的字符组成的子序列成为该串的子串。包含子串的串相应地叫主串。通常称字符在序列中的序号为该字符在串中的位置。子串在主串中的位置则以子串的第一个字符在主串中的位置来表示。当两个串的长度相等, 且各个对应位置的字符都相等时, 称

这两个串相等。

串的基本操作：判断、长度、联接、子串、定位、替换、插入、删除。

● 数组(Array)：数组是一种数据结构。一维数组就是线性表；二维数组就是数学中的矩阵，它的每个数据元素由一个值和一对下标组成，元素之间同时存在着两种关系——行关系和列关系。

数组的基本操作：给定一组下标，存取相应的数据元素；给定一组下标，修改相应数据元素中的某一或几个数据项。

数组是一种较复杂的数据结构，一般不作插入或删除操作。也就是说，一旦建立了数组，则结构中的数据元素个数和元素之间的关系就不再发生变动。因此，采用顺序存储结构表示数组是自然的事了。

● 树(Tree)：是 $n(n \geq 0)$ 个结点的有限集。有且仅有一个特定的称为根(Root)的结点；当 $n > 1$ 时，其余结点可分为 $m(m > 0)$ 个不相交的有限集 $T_1, T_2, \dots, T_m$ ，其中每个集合本身又是一棵树，并称为根的子树(Subtree)。

树的结点包括一个数据元素及若干指向其子树的分支。结点拥有的子树数称为结点的度(Degree)，度为零的结点称为叶子(Leaf)或终端结点。度不为零的结点称为分支结点或非终端结点。

结点的层次(Level)从根开始定义起，根为第一层，根的孩子为第二层……其双亲在同一层的结点互为堂兄弟，树中结点的最大层次称为树的深度(Depth)。如果将树中各子树看成从左至右是有序的(即不能互换)，则称该树为有序树，否则称该树为无序树。

森林(Forest)：是 $m(m \geq 0)$ 棵互不相交的树的集合。对树中每个结点而言，其子树的集合即为森林。

树的基本操作：初始化、求根、求双亲、求孩子、求右兄弟、建树、插入子树、删除子树、遍历、清除结构(即置为空树)。

树的应用广泛，在不同的软件系统中树的操作不尽相同，因此树的抽象数据类型可随需要而设定。

二叉树(Binary tree)：是一特定的树型结构，其特点是每个结点至多只有二棵子树，并且，二叉树的子树有左右之分，其次是不能任意颠倒。

● 图(Graph)：是一种较线性表和树更为复杂的数据结构。在线性表中，数据元素之间仅有线性关系，每个数据元素只有一个直接前驱和一个直接后继；在树形结构中，数据元素之间有着明显的层次关系，并且每一层上的数据元素可能和下一层中多个元素相关，但只能和上一层中一个元素相关；而在图形结构中，结点之间的关系可以是任意的，图中任意两个数据元素之间都可能相关。因此图的应用极为广泛，已渗入到诸如语言学、逻辑学、计算机科学和人工智能中。

§ 1.2 操作系统

§ 1.2.1 操作系统概述

操作系统是电子计算机发展的基础上产生的，形成于 20 世纪 60 年代，今天已发展成为

比较完善的一门学科。

● 操作系统(Operating System)

控制和管理计算机硬件和软件资源,合理而有效地组织计算机工作流程以及方便用户的程序集合。

- 资源:处理机、主存储器、外设和信息(程序和数据)。
- 功能:作业的管理、组织、控制、状态、调度。
- 分类:批处理、分时、实时系统。
- 使用:熟悉系统主要配置,学会启动/关闭计算机,正确使用系统键盘命令。

作业(Operating):用户程序及其所需的数据和命令的集合。用户要求计算机系统所作的既有联系,又相对独立的一项工作叫做作业步。

批处理系统:用户将一批作业有序地排在一起形成一个作业流,计算机系统自动按顺序执行作业流,以节省人工操作的时间和改善机器的使用情况。

分时系统:多道程序分时共享硬件和软件资源。通常按时间片轮转,每道程序一次运行一个时间片。用户通过终端与计算机发生交互作用,他们彼此并不感到有别的用户存在,好象整个系统为其所独占。同时、独立、及时、交互是分时操作系统的基本特征。

实时系统:实时控制和处理系统的统称。

进程:是一种活动。它由一个动作系列组成,每个动作是在某个数据集上执行一段程序。整个活动的结果是提供一种系统或用户功能。进程由程序、数据集、进程控制(描述)块三部分组成。

睡眠:是进程为了等待某种事件发生而被迫暂停前进时所处的一种状态。

并发:指在操作系统中存在许多同时或并行的活动。

● 文件(File)

有名字的一组相关联元素的有序序列。

文件系统:操作系统中与文件管理有关的那部分软件、被管理的所有文件以及管理所需要的一些数据基的总体。逻辑结构和物理结构是两个重要概念。

逻辑结构:文件元素的呈现方式,是用户对数据的表示和存取方式。常有无结构的流式文件(即有序字符的组合)和记录式文件(即数据记录的集合)两种。

物理结构:文件元素在物理存储器上存储的方式,是元素的物理表示和组织,它涉及到索引、指针、链表等技术。通常有如下三种结构:

- 顺序结构:逻辑文件的信息存放在外存的连续编号的物理块中。
- 链结构:存放文件的每一盘块中有一指针指向下一盘块。
- 索引结构:它通过文件索引表(由逻辑号和对应的物理号所组成)建立文件的逻辑号和物理号之间对应的映射关系。

文件操作:列目录、屏幕/打印输出、复制、链接、改名、删除、文本编辑。

文件调用:创建、激活、读、写、关闭、删除。

文件名的组成:文件名、扩展名。

文件属性:文件的特性及特征。如:读写、只读、隐函等特征。

§ 1.2.2 DOS 操作系统

● 磁盘操作系统(DOS; Disk Operating System)的组成

- 引导程序(BOOT RECOD)
- 命令处理程序(COMMAND.COM)
- 输入/输出处理程序(IBMIO.COM)
- 文件和“功能调用”处理程序(IBMDOS.COM)

● 资源命令类型

- 外部命令(E):这类命令的处理程序存在磁盘上,当用户打入外部命令时 DOS 把相应的处理程序调入内存并执行,完后将其释放(覆盖)。
- 内部命令(I):这类命令的处理程序在 DOS 生成时调入内存而且常驻内存。

命令格式:命令名 参数 参数…参数

命令参数:包括[d:]; [path]; filename. [ext]; filespec.

指令:控制信号。

软资源:系统运行所需的所有信息(程序及资料)。

硬资源:系统中所有的实际装置。

根目录:系统目录(在格式化盘时自动生成)。

子目录:目录中的目录(用 MD 命令生成)。

当前目录:当前正在使用的目录。

路径:一组有序的目录名(目录名之间以“\”相隔)。

路径名:[d:] [path] filename [.ext] (指定某个文件)

[d:] [path] (指定某个文件)

源文件:输入信息文件。

目标文件:输出信息文件。

编辑:修改数据的形式或格式。

任务:一个程序的一次执行。

§ 1.3 程序设计语言

§ 1.3.1 程序语言的组成

程序(Program):计算机处理对象和计算规则的描述,即数据结构加算法。

程序设计语言(Program Language):用来书写计算机程序的语言。其基础是一组记号和规则。根据规则由记号构成的记号串的总称就是语言。成分如下:

- 数据:用以描述程序中所涉及的数据。
- 运算:用以描述程序中所包含的运算。
- 控制:用以描述程序中的控制构造。
- 传输:用以表达程序中数据的传输。

在学习程序语言应注意它的三个方面:

- 语法(Syntax): 表示程序的结构或形式,即表示构成语言的各个记号之间的组合规律,但不涉及这些记号的特定含义。

- 语义(Semantics): 表示程序的含义,亦即表示按照各种方法所表示的各种记号的特定含义,但不涉及使用者。

- 语用(Pragmatics): 表示程序和使用者的关系。

§ 1.3.2 语言和程序设计的发展

计算机语言的发展是从低级语言走向高级语言,由过程化发展到非过程化。

低级语言一般指机器语言和与之接近的汇编语言。

高级语言一般指 BASIC、FORTRAN、Ada、C 等

语言发展中 Ada 语言可谓过程化语言的“顶峰”。

非过程化语言: 只需要描述要做什么或需要什么,而无需描述怎样做或如何满足这种需要。伴随着数据库的广泛应用,非过程化语言得到了迅速发展。

程序设计的发展是:

- 为摆脱冯·诺依曼思想的束缚,出现了函数程序设计。
- 为使计算技术更有效地为人工智能服务,逻辑程序设计出现了新的面貌。
- 随着计算机走进办公室,面向客体(对象)的程序设计日益受到重视。

§ 1.3.3 程序语言的使用

用高级语言编写的程序计算机不能立即执行。必须把它翻译成等价的机器语言程序。我们把用某种高级语言编写的程序正文称为源程序;与源程序等价的机器语言程序称为目标程序;把源程序翻译成目标程序的程序称为编译程序。

一般程序语言使用过程:

录入编辑源程序→经编译程序产生目标程序→执行目标程序→获得结果

§ 1.4 程序设计基础

§ 1.4.1 程序设计概念

程序设计,根据所提出的计算任务,编制一组能正确完成该任务的程序。

- 定义问题: 把问题、所涉及的输入数据以及需要的结果,用自然语言尽可能清晰、完整地表达出来。

- 算法设计: 确定怎样解决问题,并把任务分成计算机能够执行的几个特定操作。即: 确定解决问题的合适的算法(数值的或非数值的计算方法);

- 流程图设计: 用形象的、适于编写程序的方式表达算法;

- 编写程序: 用选定的语言,按流程图提供的步骤写出程序;

- 调试程序: 即排错,查出在程序执行过程中出现的错误位置并予以纠正;

- 测试程序: 确认程序在各种可能情况下均可正确执行,输出结果准确无误;

- 文档资料编制: 编写程序的使用和维护说明书;

• 6 •