



中国计算机学会会刊

ISSN 1001-3695

# 计算机应用研究

APPLICATION RESEARCH OF COMPUTERS

1997·2



- 可用于制作多媒体电子出版物
- 可用于生成多媒体信息管理系统
- 可用于制出各种信息查询系统



ISSN 1001-3695



03 >



9 771001 369007

广西计算中心海蓝集团

地址: 广西南宁市星湖路32号 邮编: 53002

电话: (0771) 5851945 5860210 58516

传真: 5852684





中国计算机学会会刊

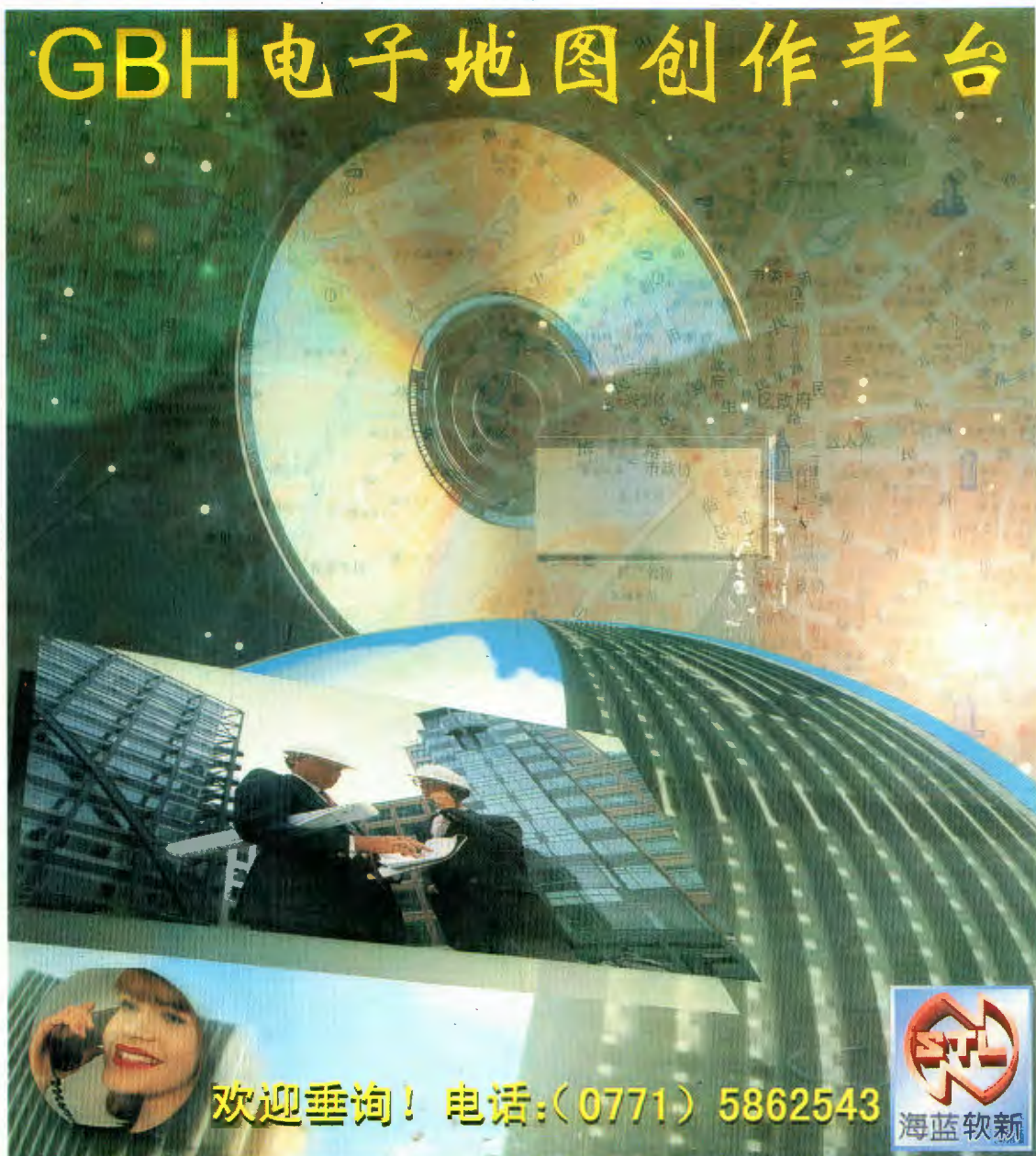
ISSN 1001-3695

# 计算机应用研究

APPLICATION RESEARCH OF COMPUTERS

1997·2

## GBH电子地图创作平台



欢迎垂询！电话：(0771) 5862543



ISSN 1001-3695



03 >



**广西计算中心海蓝集团**

地址：广西南宁市星湖路32号 邮编：530022  
电话：(0771) 5851945 5860210 5851627  
传真：5852684

# 中国计算机学会 1997 年活动计划

会议名称	时间	地点	主办单位	承办单位	联系人 通信地址
NOI'96 冬令营	1. 27	南京	普及工委	南京金陵中学	汪笑梅 210002 南京金陵中学
专委、地方计算机学会工作会议	4. 25	深圳	学会	深圳大学等	蒋似柳 100080 北京 2704 信箱
全国现场总线会议	6 月	北京	学会、机电一体化办、 工控机专委、工业 PC 联合开发委、过程控制 仪表专委	北京华 技有限公 司	张广川 郝丕荣 100011 北京德外关厢 塔院胡同甲 12 号
第 12 届全国计算机安全技术交流会	6 月	桂林	安全专委	总参 51 所	薛晋康付增少 北京长辛店 7227 信箱 15 分箱
全国第 10 届微机电路学术交流会	6 月	大同	微机电路学组	陕西临潼 771 所	王潼 710600
全国第 3 届微机医药应用技术交流会	6 月	待定	微机医药应用学组	国家医药管理局情报所	李士久 100810 北京市北礼士路中 38 号
全国计算理论暨形式语言与自动化学术研讨会	8 月	上海	理论专委	复旦大学计算机系	朱洪 200433
全国青少年计算机竞赛 NOI'97	8 月	香港	学会	香港电脑学会	蒋似柳 6256 2503
NOI'97 深圳夏令营	8 月	深圳	学会	深圳大学等	蒋似柳
全国第 14 届数据库学术会议	9 月	成都	软件专委数据库学组等	四川联合大学计算机系	唐常杰 610064 四川联合大学计算机系
全国金融科技学术交流会	9 月	待定	金融专委等	中国人行支付与科技司	王微玮 100054 金融电子化公司
'97 全国计算机新技术研究会	9 月	南阳	微机软件学组	南阳市人民银行	辛小红 临潼 771 所 开发部
全国开放式分布与并行计算机学术会议	10 月	长沙	开放系统专委	国防科技大学计算机学院	金士尧 王志英 410073 (0731)4220089
全国第 6 届多媒体技术学术会议	10 月	西安	多媒体专委等	西安交大计算机系	钟玉琢 清华大学计算机系
第 17 届计算机维护与管理学术会议	10 月	成都	维护专委	中软总公司	任公越 100081 北京学院南路 55 号
'97 全国计算机体系结构年会	10 月	待定	体系结构专委	李佐文 100854	北京 142 信箱 406 分箱
'97 全国计算机信息保密学术年会	10 月	昆明	信息保密专委	国家保密技术研究所	杜虹 100083 北京学院路 40 号 6 信箱—02
第 5 届全国计算机工程与工艺学术年会	11 月	无锡	工程与工艺专委	江南计算所	吕春阳 214083 无锡 35 信箱
全国团体会员信息交流会议	下半年		组织工委		张学孝 王彪 100083 北京 619 信箱 24 分箱
'97 全国计算机科学理论学术年会	11 月	云南	理论专委	贵州大学计算机理论所	李祥 550025 贵州大学校长办
第 7 届全国容错计算学术会议	12 月	广州	容错专委、基金委信息科学部	中科院广州电子所	吕立义 510070 广州黄花岗
'97 全国多微计算机系统与网络学术研讨会	四季度	待定	微计算机系统与网络学组	西安交大计算机系	郑庆华 710049
第 7 届全国青年计算机工作者会议 (NCYCS'98)	10 月	上海	学会	上海交通大学计算机系	傅育熙 200030 上海华山路 1954 号 电话 6281 2330



根据以上分析,我们可以由下列几步来获得间隙软指纹保护的软盘。

1)获得间隙软指纹:可分两步

①格式化特殊磁道:选定特定的磁道作特殊格式化,为避免和正常磁道相冲突,选用 28H 或 29H 磁道。格式化以间隙软指纹程序进行,也就是在设定 C、H、S、N 参数时,设法让磁盘软指纹判别程序读到此特殊磁道的首尾相接处。可以把 N 设为 6。

②记录软指纹:检查此磁道首尾相接处的数据即间隙软指纹;记录它的起始地址值及软指纹数据;放到指定的数据区,注意此数据有时不稳定,多读几次试试,如不稳定,重新格式化。

2)编写磁盘软指纹判别程序。利用此程序读放有用户程序的加密盘的特殊磁道。读出软指纹数据和第一步取得的放在指定位置的软指纹数据相比较。如果相同,判别通过;否则显示错误信息。

3)把磁盘软指纹判别程序 and 用户程序合并。用户程序可以是 .EXE 或 .COM 文件。注意 .EXE 和 .COM 文件的结构不同要分别对待。合并后的程序一般是把判别程序放在用户程序后面。对于 .COM 文件修改 100H 处的转移指令,转移到判别程序,判别通过后再转回原处执行即可。对 .EXE 文件,因有文件头与重定位问题,增加一段程序的处理就比较复杂,要作仔细的考虑。如原首指令位置的确定,增加程序后文件长度的改变,首指令位置的改变等,此处不再详述。

以上间隙软指纹加密技术具有很强的防拷贝能力,但用户程序和保护程序,很容易利用 DEBUG.COM 程序进行静态分析和动态跟踪,所以应该再加上反静态分析及反动态跟踪措施,以保护判别程序,否则一旦检测程序被人找到,软指纹就没有作用了。当然保护程序往往不只应用一种,而多种保护程序联合应用才更有效。

《计算机应用研究》杂志联(协)办单位

中国计算机学会  
四川省电子计算机应用研究中心  
贵州省科委计算中心  
安徽省计算中心  
吉林省计算中心  
福建省计算中心

内蒙古电子计算中心  
云南省电子计算中心  
新疆电子计算中心  
甘肃省计算中心  
广西计算中心  
新潮计算机产业集团公司

山东省计算中心  
河南省计算中心  
青海省测试计算中心  
福建林学院计算机科学系  
武汉汽车工业大学电子系

《计算机应用研究》杂志社董事会

董事长:周赛渝  
董事:管小宝 张国栋 徐雅君 王升亮  
徐世坤 叶大卫 乔中南 胡耀萍  
冯德成 朱 华 王俊杰

《计算机应用研究》杂志编辑委员会

主任委员:张执谦 副主任委员:李泽民  
委 员:贾洪钧 李天健 龚宇清 罗海鹏  
张湘金 倪凯民 范德元 李文华  
王义章 崔振远 刘铁军 毕常青  
周荣春 刘启茂 宁正元 赵桂林  
黄安南

1997 年第 2 期(总第 76 期)

出版日期:1997 年 3 月 31 日

责任编辑:唐大利

计算机应用研究(双月刊)  
JI SUAN JI YING YONG YAN JIU  
(公开发行)

刊 号:国际标准刊号:ISSN 1001-3695  
国内统一刊号:CN51-1196/TP

邮发代号:BM4408(国外)  
62-68.(国内)

主 编:张执谦 副主编:李泽民

主管单位:国家科委、四川省科委

主办单位:四川省电子计算机应用研究中心

编辑出版:《计算机应用研究》杂志社

通讯地址:成都市人民南路 4 段 11 号附 1 号

邮政编码:610041 传真:5221946

电 话:(028)5210177 或 5220666 转 2055

印 刷:成都市新都华兴印务有限公司

订 阅 处:全国各地邮政局

国内总发行:成都市邮政局

国外总发行:中国国际图书贸易总公司

广告经营许可证:川蓉工商广字 065 号

每册定价:4.00 元



中国计算机学会会刊  
中国科技论文统计源期刊



计算机应用研究 第14卷 第2期 (总第76期)

目次

综述评论

EMIS 开发应用如何适应校园网的发展 .....	罗伟其	朱育雄(1)
软件测试是软件质量保证的重要手段 .....		牟光灿(3)

研究探讨

EXPRESS 的对象特征 .....	唐贵川	董金祥	何志均(6)		
面向对象思想方法及其在地理信息系统开发中的应用 .....	童小华	岳秀平	杨东援	陈龙飞(8)	
面向对象技术在 MIS 建模中应用的探讨 .....				倪凯民(11)	
用于辨识建模中的神经网络 .....	侯媛彬	樊重俊	韩崇昭(13)		
一种快速而低复杂的调色板色彩组合选择算法 .....	黎越	王锁萍	秦军(16)		
工作站环境中电路的蒙特卡洛分析 .....				张文明(18)	
基于 PC 总线的设备驱动式固态硬盘的设计 .....	马骏	孔振宇	赵万生	赵德开	初佃辉(20)
图形用户界面系统的层次结构模型 .....					贺鹏(23)
FOXPRO 2.6FOR WINDOWS Wizard 生成工具功能浅析 .....				陈凤祥	贺茂林(25)
用 VB 编程实现 Windows 下屏幕的快速截取 .....				方锐	朱秋萍(28)
优化散列排序算法研究 .....				邢为民	宁正元(30)
同屏显示多幅彩色图像技术新探 .....					陆遥(32)
多任务生产计算机动态优化调度方法 MP-PERT 之探讨 .....	王首安	杜玉桥	杜习英(35)		
固态 ROM 盘初始固化文件的生成 .....	马骏	孔振宇	赵万生	赵德开	初佃辉(39)

应用开发

面向对象程序设计的分解与归纳方法					姚世军(41)
Windows 下 PC 机与 8098 单片机的串行通信				杨 军	王宏山(43)
异种机互连的实现					谢 彬(44)
基于模式识别的大罐排污自控系统					喻宗泉(47)
抚顺乙烯化工有限公司计算机网络系统设计与实现		卢 苇	徐晓玉		张志洲(49)
医用眼底图象分析系统的研制			谢晓恒		李卫华(51)
蓄电池快速充电模糊控制的初步探讨			童晓阳		童晓霞(52)
嵌入式计算机在提升机状态监控中的应用			任子晖		马小平(55)
玻璃钢罐管道纤维缠绕自动控制系统的开发		曹二堂	刘玉林		田德福(56)
塔钟多媒体自动报时系统		许 伟	余立建		蔡 淮(58)
多媒体技术在制导兵器武器系统总体技术中的应用		何衡湘	闵庆东		安景晖(60)
C 语言实现 EGA/VGA 图形方式下特殊画面的设计		王天芹	罗文保		王天泽(62)
实现 .DBF 文件结构动态调整的方法					雷 霆(63)
用于继电保护装置的智能检测分析系统研究与实现		刘 青	文劲宇	刘 沛	程时杰(66)
利用扩充内存实现多幅图象的快速存储				谢晓恒	李卫华(68)
改进的 8259A 扩展 8031 中断的接口电路及编程实现		王文涛	蔡朝晖		杨喜敏(69)
一种用单片机组成的高精度 A/D、D/A 转换电路					钟志良(71)
Foxpro 下一个功能更强的“屏幕程序生成器”的设计与实现方法					俞 坚(74)
Windows 画笔在软件界面设计中的应用研究					谭祖胜(76)
利用 Windows 环境下 DDE 通讯实现 MIS 图表的动态打印					王爱学(78)
Windows 环境下的图象处理——图象旋转		李为民	黄 瑜		梁济仁(80)
基于超文本技术的查询系统		崔大勇	聂发如	薛 智	毛静涛(82)
通用辅助录入技术的实现					唐开山(84)
DOS 系统中文件的存盘、删除与恢复技术				罗淑英	杨云江(86)

维护维修

间歇软指纹加密技术 .....	李京民	王柏盛	刘瑞英(89)
-----------------	-----	-----	---------

动态简讯

启事 .....			(19,48,61)
----------	--	--	------------

# Soul Journal of Chinese Science & Technology Articles

## APPLICATION RESEARCH OF COMPUTERS

VOL. 14 NO. 2(Total 76)

### CONTENTS

#### SURVEY

- How does the Development and Application of EMIS Suit the Development of Campus Network ... Luo Wei qi et al. (1)  
Software Test is a Key Method for Pledging Quality of Software ..... Mu Guang can(3)

#### RESEARCH

- Object Characteristics of EXPRESS ..... Tang Guichuan et al. (6)  
Idea and Method of the Object—Oriented and its Application in Developing GIS ..... Tong Xiao hua et al. (8)  
Applications of Object—Oriented Modeling of MIS ..... Ni Kai min(11)  
A Neural Network is Used for Identification System's Model ..... Hou Yuanbin et al. (13)  
A High—speed and Low Complexity Algorithm to Select Composing Colour of Palette ..... Li Yue et al. (16)  
Analysis of Monte Carlo on Circuit in Working Station ..... Zhang Wenming(18)  
Design of Solid state Disk Based on Device Drive Type of PC Bus ..... Ma Jun et al. (20)  
Hierarchical Structure Model of Graphics User Interface System ..... He Peng(23)  
Elementary Analysis to Function of Generate Tool of FOXPRO2. 6 FOR WINDOWS ..... Chen Feng xiang et al. (25)  
Realizing Rapidly Truncating and Getting on Screen Under Windows by VB Programming ..... Fang Rui et al. (28)  
Algorithm Research to a New Scatter Sorting ..... Xing Weimin et al. (30)  
A New Approach to the Technique for Showing Two or More Color Images All on the Screen ..... Lu Yao(32)  
Approaching Dispatch Method of MP—PERT of Multi—Mission Production on Dynamic Optimality Computer .....  
..... Wang Shou an et al. (35)  
Generation of the Initial Programming Files for Solid State ROM Disk ..... Ma Jun et al. (39)

#### APPLICATION & DEVELOPMENT

- Methods of Resolve and Induction for Designing Object—Oriented Program ..... Yao Shi jun(41)  
Windows Programmed Communication between PC and 8098 Microcontroller ..... Yang Jun et al. (43)  
Realizing Linking up Different Computers Each Other ..... Xie Bin(44)  
An Auto Control System Based on Pattern Recognition for Drain off Polluted Water from Petroleum Tank .....  
..... Yu Zong quan(47)  
Designing and Realizing Computer Netware System of Fushun Limited Company of Ethylenc Chemical .....  
..... Lu Wei et al. (49)  
Developing Analysis System of Medical Eyeground Image ..... Xie Xiaoheng et al. (51)  
The Rudimentary Research of Battery Quick Charging System with Fuzzy Logic Controller .....  
..... Tong Xiao yang et al. (52)  
Application of Flush Bonding Computer in Mode Monitor of Elevator ..... Ren Zihui et al. (55)  
The Microcomputer Control System of the Fibreglass Winding in Production of Tank and Pipeline .....  
..... Cao Ertang et al. (56)  
A Multimedia Turret Clock for the Automatic Time Signal ..... Xu Wei et al. (58)  
The Application of the Multi—media in the Overall Technology of Weapon System of Guided Enginery .....  
..... He Hengxiang et al. (60)  
Graphics Design of EGA/VGA by C Language ..... Wang Tianqin et al. (62)  
Method of Realizing Dynamic Alignment of .DBF File Structure ..... Lei Ting(63)  
An Intelligence Testing and Analysing System For Substation Protection System ..... Liu Qing et al. (66)  
Realizing Quick Memory of Multi Images by Extended Internal Storage ..... Xie Xiaoheng et al. (68)  
Expanding Interface Circuit of 8031 Interrupt by 8259A and Realization of Programming ..... Wang Wentao et al. (69)  
A High Precision A/D,D/A Converter Used by Single Chip Microcomputer ..... Zhong Zhiliang(71)  
The Design and Fulfillment of Screen Program Builder with More Function under Foxpro ..... Yu Jian(74)  
On Use of Windows Paintbrush in Software Interface Design ..... Tan Zusheng(76)  
Realizing Dynamic Printing MIS Diagram by DDE Communication under Windows Enviroment ..... Wang Aixue(78)  
Image Rotating under Windows Enviroment ..... Li Weimin et al. (80)  
Interrogating System Based on Technology of Over Text ..... Cui Dayong et al. (82)  
Realizing Technology of Common Aided Recording ..... Tang Keishan(84)  
The store Delete and Resume Technology of File in Disk Operating System ..... Luo Shuying et al. (86)

#### MAINTENANCE

- The Technique of Soft Finger—print Secret—setting ..... Li Jingmin et al. (89)

#### INFORMATION

- Notice ..... (19,48,61)



# EMIS 开发应用如何适应校园网的发展

罗伟其 朱育雄

(暨南大学信息网络工程研究中心 广州 510632)

**摘 要** 校园网(Campus Network)作为学校的信息基础设施,它的建立和发展,为学校的计算机应用开辟了广阔的前景。校园网 EMIS(Education Management Information System)是学校管理现代化的标志。本文从系统的总体规划、开发、实现等方面讨论了校园网 EMIS 的开发应用如何适应校园网的发展。

**关键词** 校园网 EMIS 总体规划 Client/Server 计算模式 面向对象方法 Web

## 1 前言

近年来,由于计算机技术和通信技术的迅猛发展,全球范围内兴起了联网热潮。我们国内也不例外,许多高校和科研机构,纷纷建起了校园网络,并把校园网联入了国际网(Internet)。校园网的建成将大大地促进了学校的教学、科研和管理水平的提高。

校园网是作为学校的信息基础设施来重点建设的,它的建成无疑给校园网 EMIS 的重建或改造提供了新的环境。然而,校园网 EMIS 毕竟和各个部门单独开发的 MIS 存在着很大的差异,它的开发应用有着自己的特点。我们学校进行校园网 EMIS 建设已有一年多的时间,本文根据我们在工作中的经验和体会,从总体规划、系统开发、计算模式等方面就 EMIS 开发应用如何适应校园网发展问题提出一些看法。

## 2 校园网 EMIS 的总体规划

校园网 EMIS 联系着整个学校的活动。这些活动涉及到人员管理、资金管理、资产管理、实验器材管理等,这些管理都可以用 EMIS 中的信息来反映、调节和支配,以维持正常的、有秩序的活动。校园网 EMIS 遍布全校,是一个整体,虽然它可以分部门、分层次,但各个部门、各个层次是相互关联着的。因此,必须以系统工程的思想和方法为指导。校园网 EMIS 的设计,既要考虑到学校管理工作的现在和未来,又要考虑校园网技术的发展。

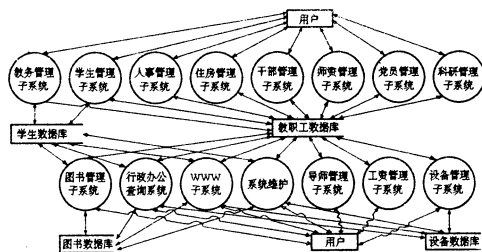
目前,国外提倡按组织机构的发展战略来制订 EMIS 的发展战略。要规划校园网 EMIS,要对学校的整体状况、办学方向、目标和战略以及校外环境等进行详尽的调研分析和把握,根据调研分析和研究评价的结果,确定要开发的校园网 EMIS 的目的、战略策略和开发规划;总体框架及系统体系结构;管理基本模型、数据基本模型、业务处理模型、技术规范、各个系统的开发次序;所需资金、资源、人员、总体计划进度等,并对学校各职能部门提出要求和作出适应性调整。

在对系统进行总体规划时,同时还应考虑数据的标准化和规范化,以便实现学校之间的数据资源共享和数据交换。国家教委于 1992 年颁布的《高等学校学生管理基本信息集》和《高等学校教职工管理基本信息集》,以及 1994 年颁布的《高等学校仪器设置管理基本信息集》都是开发校园网 EMIS 应遵循的标准。

这样,对系统就有了一个全局的、统一的考虑。有了这个思想来开发 EMIS,才能做到胸有全局、主次分明、能够统筹安排。

值得一提的是,教职工尤其是学校各级领导干部是否具有现代化信息管理意识,是校园网 EMIS 开发成败的一个关键因素。EMIS 的开发是为了应用,如果大家都没有良好的这种意识,对现代化信息管理的作用和特点缺乏了解,就不会去重视和关心校园网 EMIS,这将给系统的开发和应用带来不可逾越的障碍。因此,做好教职工和各级领导干部的现代化信息管理意识的培养至关重要,直接关系到校园网的顺利发展。

附图 1 是暨南大学校园网 EMIS 的数据流图总图。该图清楚地反映了系统的各个组成成份以及各个子系统之间的关系。



附图 1 暨南大学校园网 EMIS 示意图

## 3 系统开发的总体考虑

系统开发的首要问题是选择适宜的开发方法。目前已有许多成型的系统开发方法,常用的有生命周期法、原型法、结构化生命周期法、结构化与原型法相结合的方法、面向对象方法等。这些方法都有自己适用和局限的方面。面向对象方法比其它方法在提高系统的

可靠性、可重用性、可扩充性、减少软件复杂度等方面要有很大的优势。

校园网 EMIS 应采用分步实施、坚持边开发边应用边完善的原则。由于系统已有了统一的规划,各部分之间的联系与接口已经预先考虑好了,所以分步实现不致产生大的矛盾、冲突和不协调。

系统的开发不只是开发人员的事情,而且需要职能部门的用户、领导与开发人员的通力合作。系统功能和用户的业务过程有着十分紧密的联系,没有用户的参与,开发将一事无成。开发工作人员的组织也是一个影响成败的因素。对于校园网 EMIS 这样一个大型系统来说,开发人员应有不同的工作分工,不同分工的工作人员需要互相理解和支持。开发力量不足或人员之间沟通不够都会影响开发工作进程和系统质量。

学校的某些职能部门在微机或小型网络上已经开发了一些支持单一业务的 MIS 系统,使用时间比较长,存在着认识和工作习惯上的一些惯性,而且在系统内已经建立了许多有用的数据信息,现在要对它们进行更新改造,无论在技术上、经济上,还是在人的观念和习惯上都会遇到许多的问题。实现新旧系统交替的方式不外乎有四种:(1)是新旧系统并行工作一段时间,证明新系统是可行时再切换到新系统;(2)是先进进行试点再推广;(3)是逐步以新代旧;(4)由全部切换。究竟选择什么方式,不仅决定于技术因素,还决定于业务性质以及工作人员的心理状态。为了保护原有信息资源和投资,需要把旧系统中有用的数据信息进行选择 and 变换,然后装入到新系统中。

由于各个高校的组织结构和业务情况十分类似,它们使用的校园网 EMIS 系统必然会有很大的相似性。我们认为把别的高校开发的比较完善和成熟的系统,根据本校的特点对某些部分做些修改后集成到本校的校园网 EMIS 中去,或高校之间联合开发通用的校园网 EMIS 系统都是可行的。这不仅缩短了开发周期,而且大大地提高了 EMIS 的质量。

#### 4 校园网 EMIS 计算模式

校园网 EMIS 的实现有三种计算模式可供选择:主机模式、文件服务器模式和 Client/Server 模式。

主机模式将 DBMS 安放在主机上,数据处理和数据库应用程序全部集中在主机上,PC 机则通过仿真终端方式与主机进行数据通讯。因此当主机不堪重负时便产生数据处理瓶颈。

文件服务器模式主要将 DBMS 安放在文件服务器上,但数据处理和应用程序实际上全部分散在各个 PC 机上。PC 机与文件服务器之间互相传输的是整个数据文件,而不能达到数据记录级的互操作,因此网络负担很重。当网络用户增加而超出网络并发响应能力时,便产生数据的瓶颈,整个网络性能会严重下降。

在 Client/Server 计算模式中,处理被分割为客户

机处理的部分和服务器处理的部分。处理由客户机启动并部分控制,由服务器和客户机一起协同执行一个应用处理直至完成。客户机主要完成屏幕交互和输入/输出处理等的前端任务,服务器运行 DBMS,完成大量的数据处理及存储管理等后端任务。网上传输的不是整个数据文件,而是客户请求命令和服务响应及数据记录,因此网络上的信息流量大大减少。这就说明,Client/Server 模式非常有效地解决数据处理和数据传输的瓶颈问题,并摆脱文件服务器模式那种文件管理方式,实现真正的数据库管理。

学校的组织机构布局在地理上一般是分散的,各种业务处理也是按职能分工而分散进行的,但在数据管理和控制上往往需要集中,形成一种“集中一分散”的作业过程,这实际上就是一种主从式分布处理环境。因此,分布式处理模式比集中式处理模式更符合应用本身的发展规律,采用主机模式和文件服务器模式不能满足日益发展的分布数据处理和分布事务处理和要求,校园网 EMIS 只有采用 Client/Server 模式,才能更好地适应学校各种业务处理的要求。

校园网 EMIS 采用 Client/Server 计算模式也是和学校的经济投资能力和计算机应用系统的普及应用水平相适应的。校园网一般是以专用服务器或超级微机作为数据库服务器,用户使用的机器是普通的 PC 机,采用 Client/Server 计算模式,可以充分发挥服务器和 PC 机的性能,提高业务处理的效率。

#### 5 Web 与校园网 EMIS 数据库的连接

校园网是 Internet 的组成部分。Internet 是联结了世界上许多计算机网络的网络,是全球共享资源的集合。Web 是 Internet 上应用最为广泛的服务之一,它提供了对全球超文本和超媒体信息的灵活存取和浏览。为了将校园网 EMIS 的某些信息服务于全球用户,建立 Web 与校园网 EMIS 数据库的连接十分有意义。

目前有两种方式实现 Web 对数据库的访问。较早出现的一种是通过 Web 服务器访问,这种方式是 Web 服务器通过 CGI 机制调用外部程序实现对数据库的访问,这个外部程序称为 Web/Database 中间件。为了提高处理效率,Microsoft 和 Netscape 公司分别为它们的 Web 服务器开发了 ISAPI 和 NSAPI, W3 Consortium 也在研制一种称为 FastCGI 的机制,试图解决 CGI 和 ISAPI/NSAPI 所存在效率低和可移植性差等方面的缺陷。

实现 Web 访问数据库最新的一种方法是利用 Java 语言直接通过 Web/Database 中间件对数据库进行访问,即不象第一种方法那样要通过 Web 服务器。这两种方法所采用的中间件是不一样的。

目前实现使用 CGI 方式访问数据库的例子已是数不胜数,但无论是在国外还是在国内,使用 NSAPI/ISAPI 和 JDBC 访问数据库 (下转第 77 页)



# 软件测试是软件质量保证的重要手段

牟光灿

(中航总成都飞机设计所 610041)

**摘 要** 软件测试是软件工程的重要环节,它是软件质量的重要保证。基于此因,该文从测试原则,常用的测试方法,测试过程以及人工评测等方面对软件测试进行了介绍。

**关键词** 软件 测试

## 1 前言

软件测试是一项十分复杂多样化的工作。它在软件的整个生存期中占据重要的位置。如果在测试阶段未能很好地把握质量关,就会对整个软件系统造成十分严重的后果。因此,软件测试是软件质量的重要手段。本文就软件测试的基本概念、目标和方法进行简述与研讨。

## 2 软件测试的基本概念

### 2.1 什么是测试

统计资料表明,即使是程序设计工作进行得很周密、细微,也仍然存在有错误。所以,应在测试阶段找出并排除这些错误,测试工作是指为发现错误所进行的活动。找出软件错误不只是找出程序中的错误,所有与软件开发有关的文书资料都应是测试的对象。如软件系统分析、需求分析、数据结构分析、程序流程分析、程序结构分析、代码编写等。

### 2.2 测试的目的和目标

测试目的不是随便找几个数据,把程序走通就行了。个别测试数据走得通并不意味着程序里没问题。

因此,我们强调测试的目的是要寻找错误。在测试活动中,要始终把目标对准未被发现的隐藏错误。建立了正确的测试概念,就有了评价测试工作的标准。一个好的测试数据应该有发现未知错误的较高概率。成功的测试是能够发现一隐藏错误的测试。当然,软件的测试着重解决的是找出错误,而消除和改正这些错误则属于调试(debug)的任务。

### 2.3 测试原则

#### 2.3.1 测试与各阶段的关系

测试工作实际上涉及到软件开发的各个阶段。我们从图1可以看出它与各开发阶段的关系。对于比较小的系统,测试层次可能不需要这么多。

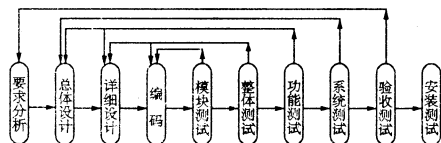


图1 测试与前面各阶段的关系

#### 2.3.2 测试的原则

- 1) 预先估计输出结果;
- 2) 编程人员应避免测试其本人的程序;
- 3) 一个程序设计组织,在条件允许时应避免测试本单位编写的程序;
- 4) 测试例题的设计必须兼顾有效与无效输入,正确与错误输入;
- 5) 检查一个程序是否完成了所有的工作只是测试工作的一半。另一半是要检查程序是否还有预料之外的副作用;
- 6) 应在系统有错的假定下进行测试,绝对不要假定程序是没错的。

#### 2.4 错误类型

就错误发生的环境而言,错误类型可分类如下:

- 1) 功能错误:说明书不完整、模糊、自相矛盾,从而导致对系统功能的误解。
- 2) 系统错误:与外部接口打交道的协议错误,无效计时,外部输出输入格式错,对错误输入容忍性不强。参数调用错,子程序调用错。分页功能的误解,I/O 设备操作错误,I/O 地址错,设备状态错,错误的中断处理等。以及操作系统接口错误,控制序列错误,资源管理问题等。
- 3) 过程错误:运算错,初始化错,逻辑错等。
- 4) 数据错误:动态数据与静态数据混淆。信息、参数与控制数据混淆。数据格式、内容、结构与属性错误等。
- 5) 编码错误:语法错,变量名错,局部与全局变量混淆,程序逻辑错误。

下面给出一个通用系统,上述错误所占比例的大概情况。

错误类型	功能错	系统错	过程错	数据错	编码错	其它
百分比	27%	16%	27%	10%	4%	16%

## 2.5 信息流与数据

### 2.5.1 测试信息流

测试过程中的信息流按图2的规定运行。

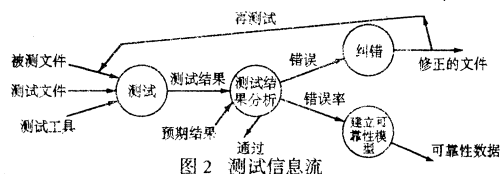


图2 测试信息流

其中,测试需要三种信息:

1) 被测试文件:如上所述可以是软件需求说明书、设计任务书、源程序等;

2) 测试文件应包括:测试计划,说明测试工作如何进行;测试用例,通常指的是一些测试数据;测试程序,测试的工作流程;

3) 测试工具:指测试过程中为便于测试而使用的软件工具。

把测试结果和预期结果进行比较后可以得出是否有错的信息,从而进入纠错处理。另一方面,根据出错的情况,可以建立可靠性模型,从而取得可靠性数据。修正后的文件还需再测试。

### 2.5.2 测试数据

为了完成测试任务,需要给出一定数量的初始数据,亦可称为测试用例(test case)。

测试数据的设计要考虑到系统运行的各种情况。它既要能检查程序是否正确地完成了它该做的事,功能和性能都符合设计要求;还要能检查出是否做了不该由它做的事。

测试数据要能体现出合理的输入数据,同时也应能体现出不合理的输入。若以不合理的数据输入时,程序应给出相应的信息,指出问题所在。

凡在测试中使用过的测试数据和相应的测试结果,都应在测试报告中得到保留,以备维护使用。

### 2.6 确认(Validation)与开发结合

如图3所示,对开发阶段的其它文件(需求分析、设计说明书等)的检验(亦称确认),传统的做法是在需求分析、设计、编码后集中进行检验确认。这样的做法会使前面几个阶段出现的问题积累起来,在最后集中检验确认又往往使问题难以解决而造成资源的浪费。

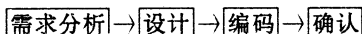


图3 确认在软件开发中的传统位置

较好的做法如图4所示。好处是问题能及时发现,及时解决。比传统做法容易发现问题。

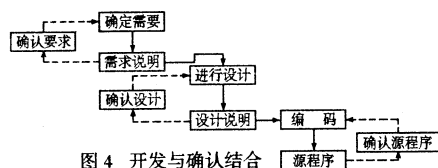


图4 开发与确认结合

### 3.1 静态分析

静态分析是检验确认软件系统的一种方法。它的主要特点是无需运行软件系统本身,而对需求分析、设计和编码阶段得到的文件进行检验,以保证软件质量。以下就这三方面的检验分别进行讨论:

#### 1) 针对需求说明的静态分析

在需求说明书里,通常要给出软件的功能、性能以及 I/O 条件。对这些要求进行静态分析,首先要列出检查表(check list),然后利用它逐项地做人工检查。检查表中应列出许多可能出现问题的项目。总的来说,这些项目应该反映以下特性:相容性、必要性、充分性、可行性、可测性。

#### 2) 针对设计文件的静态分析

在设计阶段遇到的设计环节有数学方程、算法、模块的功能描述、模块的接口、数据流程图、逻辑流程图、抽象数据结构等。同样也要列出检查表,对上述有关设计环节进行分析。它们应体现:相容性、必要性、充分性、正确性。

#### 3) 针对源程序的静态分析

对程序做静态分析有两种做法:

A: 程序特性信息分析. 在得到这些静态信息之后,再用人工方法来查找错误。例如,静态分析提供被测程序中函数(或子程序)的调用关系,变量或其它符号名的交叉引用情况,程序的循环特性,参数特性等。

B: 程序正确性分析. 查找程序中特定类型错误或发现程序结构上异常现象的功能。例如,参数分析可发现形参和实参不匹配的现象;表达式分析可发现数组下标越界,以零做除数的运算;变量引用分析可发现未赋值而引用或只赋值不引用等情况。使用静态分析来检查程序是十分有利的。

### 3.2 动态分析

对被测程序进行动态分析是要在运行被测程序的条件下,取得程序的动态特性信息。动态分析通常要经历的步骤是:

#### 1) 针对需求说明书的程序动态分析

典型方法是功能测试(即黑盒测试)。在这种测试中,人们根据功能要求来设计测试数据,并通过测试结果来检验功能。

#### 2) 针对设计文件的程序动态分析

根据设计过程中所涉及到的计算公式、算法、模块功能及接口等设计环节,提出测试数据,进行动态测试。应从各个不同的设计方面,提出不同的测试数据,使各个设计环节均能得到相应的验证。

#### 3) 针对源程序的动态测试

根据程序结构,包括程序中的语句、分支、路径,或是根据数据流、表达式、设计测试数据所完成的动态测试,常用的是:

A: 分支测试:要求在测试时,对程序中所含有的



每个分支至少执行一次。

B: 路径测试: 要求在测试时, 对程序中所含的每一路径至少执行一次。

#### 4 测试过程

##### 4.1 测试步骤

就软件开发而论, 测试工作可分为三个步骤:

###### 1) 模块测试:

模块测试主要对软件产品的每个模块单独地进行测试。其目的是力图解决编码阶段的错误。常常使用结构测试方法。

模块测试主要完成下列几项任务: 指定的模块功能执行; 测试程序逻辑与数据流路径; 输入数据, 产生输出信号并预测比较; 给出错误数据供排错用。

###### 2) 集成测试:

集成测试或称整体测试。在各个模块完成测试的基础上, 把各模块联接起来进行测试, 力图发现设计阶段的隐患, 通常使用功能测试方法。

集成测试主要完成下列任务: 系统的所有功能和特性应包括在测试计划中, 并进行测试; 数据库的装载等方面的测试; 系统接口, 包括内部接口与外部接口的测试; 错误状态处理的测试; 与特定资源分布的一致性测试; 完全装载的系统的实际或模拟测试, 检查系统的安全性。

###### 3) 验收测试:

其任务是决定软件产品是否合格, 是否符合要求, 有否在需求分析阶段的问题暴露出来。通过了验收测试的软件, 才能认为是合格的软件产品。

##### 4.2 增式与非增式测试

怎样合理地组织模块测试和集成测试, 传统的有两种方法: 增式与非增式测试。

###### 1) 非增式测试

假定已设计好的模块结构如图5中子图(1), 它表示六个模块之间的调用关系。在模块测试时, 它们应分别独立地进行测试。根据在子图(1)中的位置, 有的需要一个模拟调用它的驱动模块(R), 有的只需要一个模拟被它调用的桩模块(S), 有的两种都需要。

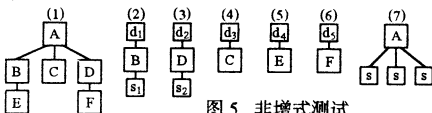


图5 非增式测试

图5中子图(2)~(7)表示模块B、D、C、E、F和A的模块测试情况。按这样的方式将每个模块测试完后, 再把各模块按子图(1)的原样连接起来进行集成测试, 这就是非增式测试方法。

###### 2) 增式测试

我们也可以按增式方法进行。自顶向下地逐步连接起来称为自顶向下增式测试, 如图6所示。自底向上

逐步连接称为自底向上增式测试, 如图7所示。

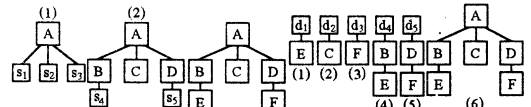


图6 自顶向下增式测试

图7 自底向上增式测试

对增式测试和非增式测试方法比较来说, 增式测试工作量少, 节省辅助测试模块, 能较快地发现排除错误。非增式测试只有在最后才能发现错误。总的来说, 增式具有较多的优越性。

##### 4.3 测试例题的设计

测试例题的设计, 根据笔者的经验, 应从用户应用环境和目的出发, 使其具有查出错误的最大可能性。要做到这点, 比较常用的方法是首先利用黑盒法建立测试例题, 对功能进行测试, 然后再利用白盒法产生补充的测试例题, 进行逻辑覆盖测试。

##### 5 人工评测

为了提高测试工作的效率, 以计算机辅助进行自动测试, 开发一些软件工具是完全必要的。但另一方面, 人工测试仍被广泛使用着。由于它确是一种高效的查错方法, 至今仍受到重视, 决不能认为是落后的笨办法。据美国IBM公司统计说明, 人工测试找出的错误可占所有发现错误的80%。

人工评测有以下几种形式:

###### 5.1 评审会(Inspection)

由有经验的程序人员作主持人(他不应是软件的设计者或程序的编写者), 吸收程序的作者或参加过设计的人员参加, 也可邀请有关专家和技术人员参加。

会前, 参加评审会者应研究过被评审软件的有关资料。会议上由程序作者介绍任务来源和程序的编写与调试情况。参加者提出问题, 作者回答。也可能要使用错误检查表, 逐步进行检查, 其中涉及到数据引用、数据说明、计算、比较、控制流程、接口、输入和输出等方面。会议时间不要太长, 以保持审查的效率。评审会只是为了发现问题。

###### 5.2 遍查会(Walkthrough)

作法与评审会大体相同, 只是会前由测试人员准备好测试数据和预期结果, 会上使用测试数据, 运行被测试软件, 以取得测试结果, 并进行比较分析。

#### 参考文献

- 1 何克清. 计算机软工工程. 武汉: 武汉大学出版社 1983
- 2 李伟华, 康继昌. 实时软件测试用例构造的充要策略研究.

西安: 航空计算技术, 1996, 第2期

# EXPRESS 的对象特征

唐贵川 董金祥 何志均

(浙江大学 CAD&CG 国家重点实验室、计算机系人工智能所 CIMS 实验室 杭州 310027)

**摘 要** EXPRESS 是 STEP 资源和应用协议的定义基础。本文主要介绍 EXPRESS 语言比较特殊的类和对象特征:子类和超类,属性和规则的继承以及子类超类约束关系。

**关键词** EXPRESS 对象 继承

## 1 引言

STEP(the Standard for the Exchange of Product model data)<sup>[1,2]</sup>是 CAD/CAM 领域的一个国际标准(ISO 10303),它提供了一种不依赖于具体机器和环境的中性机制,用以描述产品模型数据,进行产品数据的中性文件交换和产品数据库共享存档。为了有效地描述复杂的 CAD/CAM 内容,STEP 提供了形式化的模型语言 EXPRESS<sup>[3]</sup>。

EXPRESS 作为产品模型的规范语言,用以描述集成资源和应用协议,同时它还是整个 STEP 中数据模型的形式化描述工具。为了增强其描述能力,EXPRESS 吸收了许多语言的功能和特点,特别是 Ada, Algol, C, C++, Euler, Modula-2, Pascal, PL/1, SOL。除此之外,EXPRESS 还增加了一些新功能,以更适宜于表达信息模型。

## 2 面向对象特征

### 2.1 实体描述

实体(ENTITY)是一类对象,表现为一些公共属性和约束,在 STEP 中说明为一个实体数据类型,与常规语言(如 C 语言)不同,EXPRESS 不仅描述一个实体的属性,而且描述实体上的约束;属性值的大小范围;给定实体属性值之间的关系(值域规则);一类实体属性值间的唯一性,逆关系;以及多类实体间实例的关系。可以想象这是在 C 的数据结构之上加进一些约束。

一个实体说明结构如下:

```
ENTITY entity_id /* 实体类型标识符 */
[ subsuper ]; /* 子类/超类说明 */
[ explicit_attr ]; /* 显式属性说明 */
[ derive_clause ]; /* 导出属性说明 */
[ inverse_clause ]; /* 逆向属性说明 */
[ unique_clause ]; /* 唯一性规则 */
[ where_clause ]; /* 值域规则 */
END_ENTITY;
```

### 2.2 对象的特征

EXPRESS 允许某些实体定义为另一些实体的子类,子类继承其超类的特性(即属性和规则),这就创建了一种子类/超类的继承关系。

#### 2.2.1 子类/超类

一个实体通过一个 Subtype 描述说明为子类,通过显式 supertype 或隐式描述说明为超类。当隐式说明为超类时,是指它至少存在它的一个子类的实体说明中,即至少另一个实体中出现 subtype 说明为它的子类。

1)属性继承(Attribute Inheritance)和重说明(Re-declaration)

超类属性在子类的范围中是可见的,即子类继承其超类的所有属性。EXPRESS 支持多重继承(multiple inheritance)即继承其所有超类的所有属性。当多类继承出现同名属性时,通过施加超类实体名于继承的属性前以解决同名歧义性。

除此之外,EXPRESS 还支持属性重说明,即超类属性可以在子类中重新说明以进一步提高描述能力。有三种重新说明方式,超类中属性类型可以改为其一个值域的特殊子集(约束其范围,如数可以进一步约束为整数);超类中的可选(optional)属性可以更改为强制(mandatory)属性(即强化其表现作用);超类中的显式属性可以更改子类中的导出属性(即减弱其表现作用)。显然通过子类对超类的属性重说明,进一步地明确了实际需要,增加了描述事物的能力。

2)规则继承(rule inheritance)

每个应用于超类实体的局部或全域规则同样应用于子类实体。这样子类继承所有超类的所有规则。不能改变或删除任何继承的规则。但是可以添加新的规则以进一步约束子类。

#### 2.2.2 子类/超类约束

EXPRESS 规定,一个子类实体的实例需要其所有超类的实例,一个超类实体需要零个或一个子类实例。子类/超类约束可以放在子类/超类实例的组合中,通过 supertype declaration 说明。

1)抽象超类

\* 本文得到 863 集成化 CAD/CAPP/CAM 系统重大项目基金资助。

收稿日期:1996 年 6 月 24 日



EXPRESS 通过 abstract supertype 定义不可直接实例化 (be instantiated) 的超类, 但抽象超类可以和至少一个子类一起被实例化。这就是说, 抽象超类实例描述的信息不完备, 必须进一步和其子类联合起来方能表达一确切的概念, 这一点和 C++ 中的虚拟对象说明类同。

## 2) ONEOF 关系

如果超类实体将要和一组可能的子类实例的一个一起实例化, 那么 ONEOF 约束用来指定这种选择, 这种选择可以是一个单个子类或其他子类/超类约束的表达式, 即 ONEOF (...) 括号内的项可以是子类/超类约束的递归定义项。例如,

```
ENTITY pet
  ABSTRACT SUPERTYPE OF (ONEOF (cat, rabbit,
  dog, ...));
```

```
  name: pet_name;
END_ENTITY;
ENTITY cat
  SUBTYPE OF (pet)
  ...
END_ENTITY;
```

## 3) ANDOR 关系

如果一个超类的多个子类不是异或 (mutually exclusive) 的关系, 即不是相互排斥的, 那么该超类实例可以包括其多个子类的多个实例, 子类间的这种关系用 ANDOR 约束指定。如

```
ENTITY person
  SUPERTYPE OF (employee ANDOR student);
  ...
END_ENTITY;
```

## 4) AND 关系

如果超类被分成几组互不相容的子类 (即多个 ONEOF 组), 就是说有不只一种方式分类超类, 那么组间关系用 AND 约束指定。

```
ENTITY person
  SUPERTYPE OF (ONEOF (male, female) AND ONEOF (citizen, alien));
  ...
```

```
END_ENTITY;
```

## 5) 子类间的缺省约束

如果超类约束没有在超类实体中说明, 那么其子类 (如果有) 应该是彼此互相相容的, 即是说, 好象所有子类已经隐式说明为 ANDOR 关系。

如果超类约束在其某些子类上, 那么其他子类仍然缺省为 ANDOR 关系。

## 3 结论

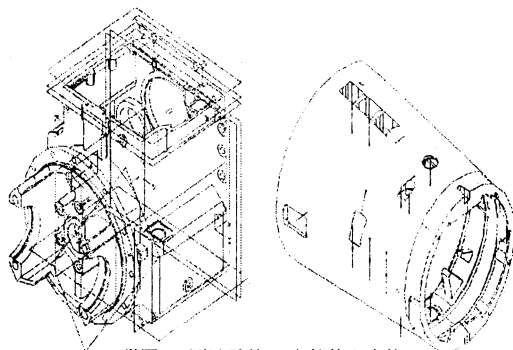
EXPRESS 是 STEP 的功能强大的面向对象的描

述语言, 和其它的对象式语言有所不同, 如在类的继承上不仅有属性的继承, 而且有规则的继承; 在超类的实例化时, 超类需要零个或多个子类的实例; 深入理解 EXPRESS 是很有必要的, 因为甚至可以借用 EXPRESS 进行其它非 STEP 领域的描述。

我们在分析 EXPRESS 的基础上, 设计了 EXPRESS 的转化器。并成功地运用于 STEP PART 203 的前后处理器的设计与实现之中。同时还发现, EXPRESS 的许多方面对开展其它工作都有实际的参考价值。

补充说明:

根据上述思想, 通过把 STEP PART 203 的 EXPRESS 面向对象的描述转化为 C 的描述后, 在 SUN sparc 工作站上 UNIX 环境下用 C 语言实现了 STEP PART 203 的前后处理器。选取了有代表性的数十个零件, 如叉车, 杭齿、杭汽、主轴, 仓体等箱体与回转体零件, 进行测试, 全部通过。该实现满足 STEP PART203 一致性分类 1, 5, 6 的要求, 即一致于 STEP203 的构件, Facetted — Brep 和 Advanced — Brep。叉车箱体零件是杭州叉车厂迄今设计最复杂的零件, 几乎用到了所有的几何描述, 并仓体列于附图 1。



附图 1 测试零件: 叉车箱体和仓体

## 参考文献

- 1 董金祥, 杨小虎. 产品数据表达与交换标准 STEP 及其应用. 北京: 机械工业出版社, 1993. 7
- 2 杨小虎. 基于 STEP 的 CAD/CAPP/CAM 系统集成技术的研究 [博士学位论文], 浙江大学, 1993. 11
- 3 ISO 10303-11: Industrial automation systems and integration — Product data representation and exchange — Part 11: Description methods; the EXPRESS language reference manual.
- 4 ISO 10303-203: Industrial automation systems and integration — Product data representation and exchange — Part 1: Part 203: Application protocol: Configuration Control Design.

# 面向对象思想方法及其在地理信息系统开发中的应用

童小华 岳秀平 杨东援 陈龙飞

(同济大学道路与交通工程系 上海 200092)

**摘 要** 本文主要论述采用面向对象的认知思想和方法来表达现实世界,包括面向对象的发展,面向对象的核心概念,并阐述其在地理信息系统中的应用;地理信息系统的空间抽象数据类型、面向对象分析、面向对象设计以及面向对象的实现等。最后简要介绍其在城镇地籍信息系统开发实例中的应用,指出面向对象地理信息系统将是今后的重要发展方向。

**关键词** 对象 面向对象 地理信息系统 城镇地籍信息系统

## 1 面向对象思想方法简介

“对象”这个概念在现实生活中无时、无处不在。可以说,我们身边的一切事物,小至一根针,一本书,大至一个人,一所学校,都是对象。一个对象,有其自身的特点和状态,并具有与外界联系沟通的渠道和能力。这里,我们给出对象 OBJ 的定义如下:

OBJ::= $\langle$ ID,MS,DS,MI $\rangle$

其中 ID 是对象标识(名字);

MS 是对象的操作集合;

DS 是对象的数据集合;

MI 是对象的消息集合,也称对外接口或协议。

“面向对象”(Object Oriented)一词的使用只有 10 来年的历史。和人们认识世界的规律一样,面向对象的方法学认为:客观世界是由许多各种各样的对象所组成,每种对象都有各自的内部状况和运动规律,不同对象间的相互作用和联系构成了各种不同的系统,构成了我们所面对的客观世界<sup>[1]</sup>。因此,我们在设计和实现一个客观系统时,应按照人们在现实世界中对外观事物的思维方式,设计尽可能直接、自然地表现问题的求解。面向对象方法所追求的是现实问题空间与软件系统解空间的近似和直接模拟,它希望人们用最小的气力、最大限度地利用软件系统来求解问题,将客观世界的对象映射到面向对象系统的解空间是面向对象方法的核心内容。

软件系统设计开发的历史经历了一个较长的发展道路,大体上可以划分为三个阶段:混沌时代,结构化时代和对象化时代。如图 1:

1950—1960 混沌时代	1970—1980 结构化时代	1990— 对象时代
jumps, gotos 非结构化变量 由程序弥散的变量	if-then-else 程序模块记录 while 循环	对象、消息 方法、继承

图 1 软件系统设计的阶段

40 年代第一台电子计算机问世,无论在运行速度

和存储容量,都相当低,而且是“0”、“1”等面向机器代码的编程,其软件规模不可能庞大。70 年代荷兰学者 Dijkstra 提出了“结构化程序设计”,将程序设计抽象为三种基本程序结构:顺序结构、选择结构和重复结构,并提出了自顶向下、逐步求精、模块化程序等设计原则,它是一种面向过程的思想。但是,这种方法的致命弱点是随着软件规模和复杂性的增长,整个软件系统的结构须重写而变得难以维护。“面向对象”程序设计起源于 1972 年美国 Xerox 公司为快速处理各种信息在 Alto 个人机上研制的 Smalltalk 语言,这不仅仅是语言上的进步,更重要的,是思想方法上的一场深刻的变革,因而被人们称为是一种回归自然的设计思想和方法。

## 2 面向对象思想方法的核心概念

面向对象方法的基本出发点是尽可能按照人类认识世界的方法和思维方式来分析和解决问题,它与现实世界之间有着自然而直接的对应关系,能很好地仿真人工系统,模拟现实世界,因此被广泛地引入到许多领域。面向对象的基本概念包括数据与操作的封装性(Encapsulation),四个抽象概念:分类(Classification)、概括(Generalization)、聚集(Aggregation)和联合(Association),以及继承(Inheritance)和传播(Propagation)两个语义模型工具。

### 2.1 类和实例

类是对一组客观对象的抽象,它将该组对象所具有的共同特色集中起来,以说明该组对象的能力和性质。因此,由一大批相同或相似的对象可以归纳构造出类。如同济大学北教学楼,它是一个对象,同济大学南教学楼,它也是一个对象,这两个对象除了地理位置不同外,其它特征完全一样,这样,我们可以构造“建筑物”这样一个类,建筑物类中包括建筑物的座落、结构、层次、面积、用途、建造时间等属性,以及显示其外观、查询等操作方法。而南楼和北楼都是这个类的实例。它们之间的关系是抽象和具体的关系(Instance of),实例是类的具体事物,类是多个实例的综合抽象。面向对

象很适合这种认识方式的组织。

## 2.2 封装和信息

封装将数据结构及其处理操作的描述连接在一个根下,对象状态的描述以及状态的转移只能通过这个对象内部定义的各种操作访问和改变。这样,对象的数据避免了暴露在对象之外,数据的存储和实现对于其他对象来说是透明的。其他对象毋需知道其具体实现细节,而只需知道它能做什么。对象与对象之间只能通过消息进行通讯,而且,这种通讯必须建立一种协议机制。所谓协议,是一个对象对外服务的说明,它告知一个对象可以为外界做什么,外界对象能够并且只能够利用该对象发送协议中所提供的消息,请求该对象服务。这种封装与消息机制,将对象的使用者和对象的设计者分开,实现了信息隐蔽,提高了系统的重用性、可维护性、灵活性等。

## 2.3 继承和派生

继承是对象之间独特的关系,它使得某类对象(派生类)可以继承另外一类对象(基类)的特征和能力。继承使得新的子类不但自然取得其父类的所有方法,而且可以定义本身所特有状态和方法,并且可以用同一名字的不同方法或属性来抑制它的父类中的方法或属性(称为多态性,Polymorphism)。继承的种类很多,像多重继承,单重继承,取代继承,包含继承,受限继承,特化继承等。引入类的继承,就出现了类的层次结构,也就有了超类、子类的概念。类的层次关系是用于描述和组织世界事物的一种工具,实践证明,是一个很有效的建模工具。继承机制大大提高了系统的可重用性,面向对象系统可以通过继承机制不断扩充功能,而不影响原有系统的运行。

## 2.4 联合和聚集

联合是将一组具有部分相同属性值的对象组合起来,形成一个更高层次的对象类型。它与概括不同之处在于概括是对类型进行抽象概括,而联合是对对象进行抽象联合。聚集与联合类似,但聚集是将几个不同特征的对象组合成一个更高水平的复合对象(如同济大学由建筑物、道路、运动场、学生和教师等组成)。传播是作用于联合和聚集的工具,它为复杂对象提供了从各成员对象抽取所需属性信息的方法。

## 3 面向对象方法在地理信息系统中的应用

地理信息系统(Geographic Information System, 简称GIS)是集计算机科学、地理学、测绘遥感学、环境科学、城市科学、空间科学、信息科学和管理科学为一体的新兴边缘学科。属于信息科学和信息产业的一部分<sup>[2]</sup>。GIS的外观为计算机系统,内涵则是由计算机程序与地学空间数据组成的地理空间信息模型。GIS的研究对象是整个地理空间,而世界上80%的信息与地理位置有关,因而GIS发展受到了世界范围内的普遍重视,应用遍及环境、资源、交通、商业、城市规划、经济

咨询、灾害损失预测和军事等领域,渗透到社会和经济的各个方面。

在GIS传统数据模型中,空间数据模型主要有矢量模型和栅格模型,而非空间信息主要采用关系数据模型。传统的数据模型对空间对象进行分解,即由点构成线,线构成面,面构成地理空间,它是对点、线、面的管理和操作,缺乏对一个完整的对象进行明确且自然的表达的能力。传统模型将数据的定义和操纵分离,对复杂的结构化或非结构化的数据信息的存储和运算繁琐,而且数据结构的可扩充性弱,难以对空间实体进行快速查询,也很难将某些地理实体作为整体来进行分析。传统的关系数据模型实质是一张二维表,难以表达空间实体复杂的层次和网络结构,为了能用规范化的关系描述,对实体进行了人为的分解。因此关系模型在效率、数据语义、模型扩充、程序交互和目标标识等方面,特别是在处理空间数据库所涉及的复杂对象方面,显得难以适应<sup>[7]</sup>。而面向对象数据模型和方法的提出和面向对象GIS(OOGIS)的建立,将为目前GIS所面临的问题提供解决途径,也是GIS能尽快进入决策应用阶段的关键技术。

### 3.1 面向对象的数据模型

考察空间数据中的各种地物,在几何性质上有以下几种类型:点状地物、线状地物、面状地物和由这三种地物组合起来的复杂地物。如图2。

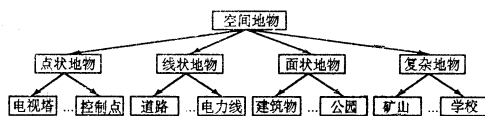


图2 空间地物的分类

可以将上述4种类型定义为GIS中地物类型的抽象类(Abstract Class),实现空间地物的封装对象,并通过多层次的派生关系衍生出多种具体对象。

### 3.2 面向对象的开发方法

面向对象的开发是基于面向对象分析(OOA),面向对象设计(OOD)和面向对象的实现(OOS)的系统开发技术。

面向对象分析过程是提取一个系统的需求的过程,即是为了满足用户的需求,系统必须做什么,而不是指系统将如何实现。面向对象分析首先确定类—对象,明确类与类或对象与对象之间的关系,然后对它们的属性和方法进行描述,并按照它们之间的关系进行组织,以得到类或对象结构。在这个过程中,需要应用诸如数据抽象、过程抽象、分类、封装、组织方法等系统分析的方法。

面向对象设计采用建模观点。它允许设计者从问题域开始并自然过渡到求解域,采用问题模型构造应用程序。OOD包括类的设计和应用设计两个部分,并且被融合于应用开发中,在类的设计过程中,将确定的

类组成基础类库,用以支持重用。应用设计则包括任务管理和数据管理、人机交互等部分的设计。OOA 与 OOD 并不是分裂开来的,而是交叉在一起。

面向对象的实现是在 OOA 和 OOD 基础上,运用面向对象程序设计语言(OOP)进行系统开发,实现具体的功能和任务。

#### 4 实例介绍

城镇地籍信息系统(County Cadastral Information System,下称 CCIS)是自行开发的基于 Windows 环境的专题性 GIS 系统。地籍信息系统,以计算机为支持,宗地为对象,对各种地籍信息进行采集、加工、存储、分析和表达,为土地管理和社会服务的信息管理系统。CCIS 采用 Borland C++3.1 中提供的面向对象的类库 ObjectWindow Library(简称 OWL)进行开发。OWL 具有 C++ 面向对象的特征,实现窗口消息的封装,对 Windows API 函数的抽象,自动消息响应等,是一种 OOP 语言。在系统开发的整个过程中,采用 OOA、OOD 和 OOS 方法和技术进行系统的分析,设计和编程。下面简要介绍 OO 方法在 CCIS 的数据模型和界面设计中的应用。

##### 4.1 面向地籍实体的数据结构

4.1.1 CCIS 空间数据结构的设计上,采用面向地籍实体的数据结构。这里地籍实体可抽象为点状实体、线状实体、面状实体以及由这三种基本形状组成的复杂实体。例如,一个宗地,宗地界址地址线,宗地内的建筑物(面状),围墙,栅栏,电力线,道路,河流(线状),控制点,门牌,窖井,消防栓(点状)等均属于上述四种超类的派生对象。以这些地籍实体及其属性、关系为处理对象的数据结构,称之为面向地籍实体的数据结构。实现对这些地籍实体进行属性和操作的封装,并以为之处理的最小单元。以下是基本类的部分定义:

```
class Object{
protected:
    char code;           //实体代码
    char describe[9];    //实体参数
    BOOL DelFlag;       //删除标志
    HDC PaintDC;
    int Layer;
    int Color;
    ....
public:
    virtual void Draw();
    virtual void Change();
    virtual void Add();
    ....
};
class LineObject:public Object{
public:
    构造函数及析构函数...
    List *x,*y;        //定位点链
    BOOL MainLine ParelLine,SubLine;
    void Draw();
    void Change();
    void Add();
    ....
};
```

#### 4.2 CCIS 的界面设计

CCIS 界面系统采用 MDI(Multiple Document Interface,多文档界面)标准,充分利用面向对象中的封装、继承、聚集和传播等技术。具有多窗口、多任务、用户友好等特点。以下是 CCIS 界面设计的窗口对象图(见图 3)和部分用户界面(见图 4)。

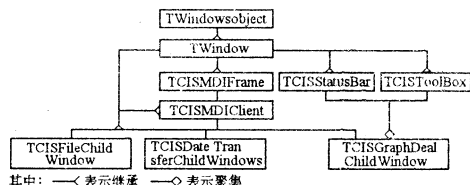


图 3 CCIS 界面对象图

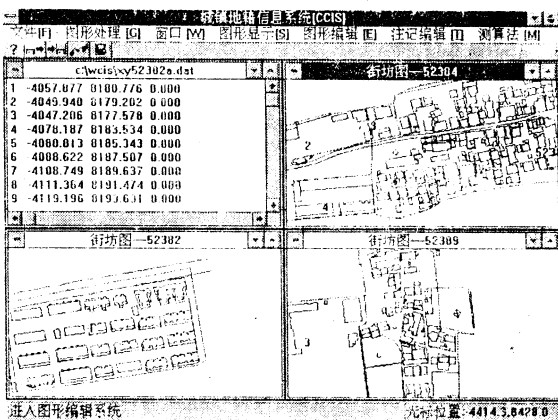


图 4 CCIS 部分界面

#### 5 结语

面向对象是一种认知方法学,它既提供从一般到特殊的演绎方法,又提供了从特殊到一般的归纳方法。面向对象也是一种设计方法学,它基于信息隐蔽和抽象数据类型概念,把系统中所有资源,如数据、模块以及系统看成“对象”,作为“研究高新技术的好方法”之一,面向对象方法已经被广泛地引入到许多领域。在面向对象的地理信息系统中,将面向对象的思想和方法应用于地理信息系统的设计和开发,如面向对象的空间数据结构,面向对象的数据库管理系统,面向对象的决策支持系统等,将是地理信息系统今后发展的方向。

##### 参考文献

- 汪成为等编著. 面向对象分析,设计及应用. 北京:国防工业出版社,1992
- 李德仁等编著. 地理信息系统导论. 北京:测绘出版社,1993
- 龚健雅. GIS 矢量栅格一体化数据结构与面向目标数据模型的研究. 武汉:[博士学位论文]. 1993
- 杨笑清等编著. 面向对象程序设计——一种循序渐进的系统构造方法. 北京:北京大学出版社,1992
- '94 地理信息系统学术讨论会论文集. 北京:中国科学技术发展基金会资助出版,1994
- 董小华. 基于 Windows 环境的城镇地籍信息系统的开发. 上海:[硕士学位论文]. 1996
- Lee, Y. C. A comparison of relational and object oriented mode is for spatial data. Report in the Symposium of IS-PRS. Wuhan. 1990



# 面向对象技术在 MIS 建模中应用的探讨

倪凯民

(安徽省计算中心 合肥 230031)

**摘要** 管理信息系统(MIS)是计算机应用的一个极其广泛的领域,而建立 MIS 系统的信息模型又是其中最关键的一步。传统的建模方法是面向处理的功能分解法,本文先分析了该方法的有限性,然后介绍了现在流行的面向对象技术的概念,并把它应用于 MIS 系统的建模当中。

**关键词** 面向对象 管理信息系统 MIS 建模

## 1 引言

面向对象(Object-Oriented)最初是用来描述 Smalltalk 程序设计语言的概念。现在面向对象的研究遍及计算机系统的各个领域,如面向对象的程序设计语言、面向对象的程序设计方法、面向对象的操作系统、面向对象的数据库管理系统以及面向对象的系统分析方法等等。在大型软件开发项目中,无论是实时过程控制系统、数据库系统、专家系统、人工智能系统,还是具体到自动代码生成,或扩展到一个工厂的计算机集成制造系统,都存在着一个共同的需求,这就是在系统分析阶段建立系统的信息模型。管理信息系统(MIS)是计算机应用的一个极其广泛的领域,而建立 MIS 的信息处理模型又是其中最关键的一步。传统的建模方法是面向信息处理的功能分解法,本文先分析了该方法的局限性,然后介绍了现在流行的面向对象技术的概念,并把它应用于 MIS 系统的建模当中。

## 2 建立 MIS 系统信息处理模型的传统方法

### 2.1 MIS 系统的结构特征

由于各个企业或组织的生产和管理方式、组织形式、运行机制及信息结构图不相同,所以使得信息系统也呈现不同的形式,但是构成信息系统的实体都是类似的或相同的。这里的实体是指存在于问题空间的客观事物(即对象),包括物理实体(如:顾客、零件)和概念实体(如:订单)。这些实体并不依赖于企业生产管理方式。组织形式和运行机制,具有相对的独立性,是可以标准化的实体。存在于问题空间的信息流将有关实体联结起来,就构成特定企业或组织的信息系统。所以说 MIS 系统的结构特征是由相对稳定的实体和随具体情况变化的信息结构组成的。

### 2.2 MIS 系统处理过程建模及其要求

根据国外比较一致的定义,一个处理就是特定用来达到某个目标的若干步骤的集合,其中的每一个基本元素就是一个处理单元(PROCESS ELEMENT),它是处理的原子活动,从外部看,它没有明显的内部结

构。

一个处理模型是一个实际的或假想的具有若干事务处理单元的事务处理的抽象描述。显然对一个事务处理来说,为它而建立的模型可能会因人而异,因方法而异,但是不管建成什么样的模型,都应能表示以下四个方面,即:

①功能方面(表示正在执行的处理单元及与其相关的物质流);

②行为方面(表示什么时间执行处理单元);

③组织方面(表示在哪里、又由谁来执行处理单元);

④信息方面(表示由一个处理过程产生或操作的结果或信息)。

实际上,这些功能方面和信息方面的表示是高度相关的。

从信息系统角度看,处理过程建模应该包括新的技术,而且建模的结果应该成为开发新系统的基础。因此,一种建模方法必须具有如下特征:

①它必须能描述前述的事务处理的四个方面;

②它应简单、易懂、规范;

③有助于计算机辅助系统分析,即其结果有助于新系统的开发。

### 2.3 数据流图(DFD)及其局限性

信息系统建模的常规方法是侧重于分析数据的流动和变换,数据流图(DFD)就是最常用到的方法。然而它在 MIS 系统事务处理过程的建模中有局限性:

①传统的结构化方法和数据流图是面向功能的。功能表示在结构化分析中起着中心作用,DFD 描述静态环境中事务处理非常有效。可是在一个动态的处理环境中,功能上的一点点变化会导致整个系统的 DFD 发生很大变化。

②DFD 缺少能力表示时间关系,它没有提供工具表示一系列处理过程的前后时间对应关系,即它无法表示前述的行为方面要求,这一点是 DFD 方法在 MIS 事务处理过程的建模中最致命的缺陷。

③DFD 也不能确定由谁在哪里执行处理单元。

DFD 的四个基本要素:数据流、实体、文件和处理都无法描述在哪里由谁来进行事务处理,即 DFD 缺少表示组织方面的功能。

总之,DFD 方法用于 MIS 系统的建模不是十分合适的,应该引入新的方法,而面向对象(OO)技术正日趋成熟,下面就把该方法用于 MIS 系统事务处理的建模中去。

### 3 面向对象(OO)的系统建模方法

#### 3.1 面向对象(OO)基本概念

面向对象分析(OOA)系统中的基本实体是对象(OBJECT),对象具有内部数据描述即属性(ATTRIBUTES)及其上的操作(OPERATIONS),有两种基本操作,一种是操纵内部数据称为方法,另一种是发送消息给别的对象请求服务。消息和数据流相连表示两个对象进行数据交换。对象分组成为类(CLASS),一类对象具有一些共同的属性。类下面分为子类(SUBCLASS),子类可从超类(SUPCLASS)那里继承属性(包括数据定义和操作)。

图 1 表示了对象、继承性及服务请求等概念。

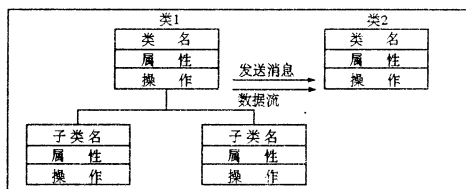


图 1 对象类、继承性及消息发送

#### 3.2 面向对象的建模方法

在面向对象的模型中,对象之间通过传递消息进行联系的,对象类之间的动态时间关系可以放在传递的消息中定义,没有必要在模型外部对处理控制进行静态的描述。同样地,面向对象方法具有数据封装功能,没有必要建立静态的数据库或数据文件模型。所以面向对象方法比 DFD 法更有效。

根据面向对象分析建模要求,MIS 系统中有四类基本对象类。即:

- ① 输出对象类,如:销售报表,订单等;
- ② 自然状态的对象类,即实际存在的实体,如:顾客,机器等;
- ③ 事件对象类,如:日常处理(订货),决策活动(信用确认)等;
- ④ 输入对象类,如:输入系统的信息实体(象税单,订货申请等)。

根据系统的定义,一个对象类可以同时是输出对象类和输入对象类。举例来说,订单对购买系统来说是输出对象类,而对订货系统来说又是输入对象类。

面向对象建模是任务驱动的。其工作过程是以下三个过程的反复:

① 根据系统的输出要求,确定输出对象类及其属性和操作;

② 从输出对象反推到要做的事件对象类、自然状态对象类、甚至输入对象类等;

③ 优化定义的层次关系。

以上面向对象建模方法很适合于 MIS 的系统分析,因为它能为查找和建立对象类提供一种结构化方法。

对于复杂的 MIS 系统,尤其是对动态发展,变化的复杂的系统建立信息模型,是一件难度很大的工作,采用传统的结构化方法(功能分析法,数据流分析法等)都很难清楚而全面地说明系统的需求。然而,面向对象的分析方法已展现出其优越性。面向对象的系统分析的最终结果是建立准确的实体系统的信息模型,即待开发的 MIS 系统的信息结构和基本实体的集合。根据信息模型结构,借助于运行机制,利用实体构造系统是开发 MIS 系统的有效途径。

#### 3.3 实例

下面举一例来说明以上方法的应用。

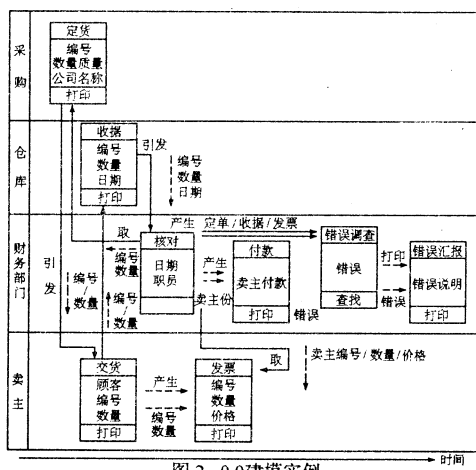


图 2 OO建模实例

某营业单位财务部门的工作过程如下:

- ① 订货部门送来一份订货单据;
- ② 货到以后,财务部门又收到仓库的验收收据;
- ③ 卖主给财务部门送来售货发票;
- ④ 财务部门核对订单、收据和发票;
- ⑤ 如果一切正常,财务部门付款;
- ⑥ 如果有差错,派人调查,并给出出错汇报。

这一过程涉及四个部门:订货部门,仓库,财务部门和卖主。汇款单和出错汇报这样的输出对象类是可以区分的,从它们可以推出要做的事件对象类:财务部门方面的出错调查和验收、卖主方面交货工作,再进一步可推出订单,收据和发票。

由于本文仅强调建模过程,所以包括的仅是一些主要属性和操作,其它一些细节方面就略去了。图 2 为

# 用于辨识建模中的神经网络

侯媛彬 樊重俊 韩崇昭

(西安交通大学系统工程研究所 710049)

**摘要** 本文重点讨论正向辨识建模,提出了一种基于BP网络的改进的BP神经网络(Improved BP Neural Network 简称为IBP)用于正向建模的方法。并以非线性的具有滞后的系统为例进行了仿真,对采用常规的BP网络和IBP网络用于正向辨识建模的情况进行了比较,得出了比较理想的结果。

**关键词** 正向辨识建模 IBP网络 网络的抗干扰性

## 1 正向辨识建模中新的神经网络的提出

在非线性系统的建模方面神经网络显示了强大的潜力<sup>[1]</sup>,并比传统的辨识和建模方法有更多的优点。在系统辨识中有一个重要问题是系统的可辨识性,即给定一个特殊的模型结构,被辨识的系统是否可以在该结构内适当地被表示出来。因此,必须预先给予假设,即所有被研究的系统都属于所选神经网络可以表示的一类里,根据这个假设,对同样的初始条件和任何特定输入,模型和过程应产生同样的输出。因此,辨识的过程就是根据模型和过程的输出误差,利用其算法调节神经网络的参数。直到模型参数收敛到它们的期望值。如果神经网络训练过程要表示系统正向动态,这种建模方法就叫做正向辨识建模。

在对的具有滞后非线性系统正向建模时,设一阶且滞后一个采样周期的数学模型<sup>[2]</sup>为:

$$Y_i(K) = 0.3366Y_i(K-1) + 0.6634u_i(k-2) + d \times \sin\left(\frac{2\pi K}{A}\right) + q \times f((Y_{i+1}(K))) \quad (1)$$

作为(过程)实际系统的模型,这里采用IBP神经网络,

收稿日期:1996年10月2日

图1中的神经网络结构如1示。图中的协调器来完成不同于BP网络的改进部分的工作。该网络分为一个输入层,其神经元(节点)的个数为三;两个隐层,每个隐层的神经元个数均是九;一个输出层,其神经元的个数为1。下面先介绍一般的BP网络用于正向建模的情况,然后探讨改进的BP(IBP)神经网络用于正向建模的方法。

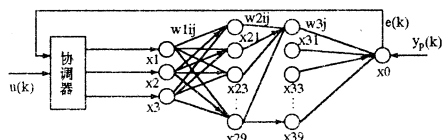


图1 正向建模的神经网络结构

## 2 对传统的BP网络在正向建模中学习情况的分析

BP网络是一种正向的、各层相互全连接的网络。其激活函数选用连续可导的Sigmoid函数 $f(x) = 1/(1 + \exp(-x))$ 。本文用MATLAB软件(在用神经网络作控制器和系统联机调试时是用C语言)。根据BP算法,模型式(1)式中干扰的系数为零,输入760℃时,经过对所有学习因子的配合全面调整,达到了这种算法

以上过程的简单模型。

## 4 结束语

面向对象技术提供了一种描述信息的机制,它是一种独特的、抽象层次较高、思维方法和传统方法不一样的方法。其概念不易掌握,所以从其诞生到引起重视,进而进入应用领域成为计算机工作者的有力工具经历了十几年时间。虽然说现在面向对象技术已成为一个非常热门的课题,但是目前在许多方面仍处于研究和发展阶段,还没有完全成熟。本文正是试图把该技术应用到MIS系统建模中,以期解决传统方法所不能解决的一些问题。

## 参考文献

- 1 蔡希尧,陈平. 面向对象技术. 西安电子科大出版社
- 2 李芳芳,柴跃廷. 计算机软件新技术——面向对象

的系统分析. 清华大学出版社

- 3 B. Curtis, M. I. Kellner, and J. Over. Process Modeling
- 4 S. Wang. Object-Oriented Systems Analysis. A Tool for MIS
- 5 S. Wang. OO Modeling of Business Processes
- 6 Rumbaugh, J. et al. Object-Oriented Modeling and Design
- 7 Sally Shlaer, Stephen J. Object-Oriented Systems Analysis Modeling the World in Data
- 8 Barry D. Kurtz, Donna Ho, and Teresa A. An Object-Oriented Methodology for System Analysis and Specification
- 9 Dimitris N. Chorafas & Heinrich Steinmann. 面向对象数据库. 清华大学出版社

的最佳状态如图 2 示。

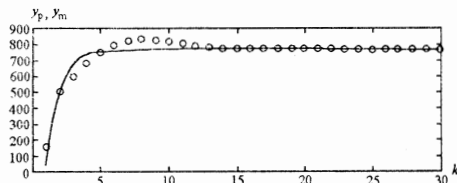


图 2 BP 网络对理想模型跟踪情况

### 3 基于 BP 网络的抗干扰性的 IBP 网络

在正向建模中,IBP 是在分析了 BP 网络的抗干扰性后提出来的。对于上述的 BP 网络,可以根据输入和输出的连接权矩阵的结构定性判断网络对输入的容错能力<sup>[3]</sup>,在 IBP 网络中,仍选用 Sigmoid 函数,即  $f(x) = 1/(1+e^{-x})$ ,其中  $f(x)$  为节点输出,  $f(x) = a$ ;  $x$  作为下层节点输入,表示为:

$$x_i = \sum_{j=1}^p W_{ij} a_j + \theta_i, (i=1, 2 \dots n) \quad (2)$$

其中  $P$  为上层节点数,  $n$  为下层节点数,  $a_j$  为上层节点的输出。由于 BP 网络的训练过程是一种误差修整的算法,它由正、反向传播组成,在正向传播中,输入信号从输入层通过作用函数后,逐层向隐层、输出层传播,每一层神经元状态只影响下一层神经元状态。通过修改各层连接权值和阈值,使得误差信号最小。对三层 BP 网络,其输出可以表示为:

$$Y = f(W_3 a_2 + \theta_3) = f[W_3 f(W_2 a_1 + \theta_2) + \theta_3] \\ = f\{W_3 [f(W_1 x + \theta_1) + \theta_2] + \theta_3\} \quad (3)$$

式中,  $Y$  为输出向量,  $Y = [y_1, y_2 \dots y_n]^T$ ;  $a_1, a_2$  为第一隐层和第二隐层的输出;  $X$  为输入向量,  $X = [x_1, x_2, x_3]^T$ ;  $\theta_{1i}, \theta_{2i}$  为第一隐层和第二隐层  $\theta_{hi} = [\theta_{h1}, \theta_{h2} \dots \theta_{hn}]^T$ , ( $h = 1, 2$ );  $\theta_3$  为输出层的阈值,上式写成分量形式:

$$Y_k = f\left(\sum_{j=1}^n W_{3ij} a_{2j} + \theta_3\right) = f\left[\sum_{j=1}^n W_{3ij} f\left(\sum_{j=1}^p W_{2ij} a_{1j} + \theta_{2i}\right) + \theta_3\right] \quad (4)$$

$$= f\left\{\sum_{j=1}^n W_{3ij} \left[\sum_{j=1}^p W_{2ij} f\left(\sum_{j=1}^m W_{1ij} x_j + \theta_{1i}\right) + \theta_{2i}\right] + \theta_3\right\} \quad (4)$$

式中  $n=9, m=3$ 。

下面分析连接权值矩阵  $W$  及输入  $X$  的变化对网络输出的影响,进而探讨提高网络的抗干扰性的措施。

由于 Sigmoid 函数连续可微,因此有:

$$\frac{\partial Y_k}{\partial W_{1ij}} = Y_k(1-Y_k) \sum_{j=1}^m W_{3ij} a_{2k} (1-a_{2k}) \sum_{j=1}^p W_{2ij} a_{1k} (1-a_{1k}) \quad (5)$$

$$\frac{\partial Y_k}{\partial W_{2ij}} = Y_k(1-Y_k) \sum_{j=1}^p W_{3ij} a_{2k} (1-a_{2k}) a_{1k} \quad (6)$$

$$\frac{\partial Y_k}{\partial W_{3ij}} = Y_k(1-Y_k) a_{2k} \quad (7)$$

$$\frac{\partial Y_k}{\partial x_i} = Y_k(1-Y_k) \sum_{j=1}^p W_{3ij} a_{2k} (1-a_{2k}) \sum_{j=1}^p W_{2ij} a_{1k} (1-a_{1k}) \quad (8)$$

$W_{1ij}$

从而有:

$$\Delta Y_k = Y_k(1-Y_k) \sum_{j=1}^n W_{3kj} a_{2k} (1-a_{2k}) \sum_{j=1}^n W_{2kj} a_{1k} (1-a_{1k}) x_i$$

$$\Delta W_{1ij} \quad (9)$$

$$\Delta Y_k = Y_k(1-Y_k) \sum_{j=1}^n W_{3kj} a_{2k} (1-a_{2k}) a_{1k} \Delta W_{2ij} \quad (10)$$

$$\Delta Y_k = Y_k(1-Y_k) a_{2k} \Delta W_{3ij} \quad (11)$$

$$\Delta Y_k = Y_k(1-Y_k) \sum_{j=1}^n W_{3kj} a_{2k} (1-a_{2k}) \sum_{j=1}^n W_{2kj} a_{1k} (1-a_{1k}) \Delta W_{1ij} \quad (12)$$

$\Delta x_i$

分析(9)~(12)式,得出如下结论:

(a) 由于  $0 < Y_k < 1, 0 < a_{1k} < 1, 0 < a_{2k} < 1$ ,当权值矩阵  $W_{pji}$ , ( $p=1, 2, 3$ ) 发生变化时,网络能减小由于系数变化对输出产生的影响。从(9)式可见,若  $W_{1ij}$  变化一个单位,网络的输出  $Y_k$  的变化最大不超过 0.015625;从(10)式可见,若  $W_{2ij}$  变化一个单位,网络的输出  $Y_k$  的变化最大不超过 0.0625;从(11)式可见,若  $W_{3ij}$  变化一个单位,网络的输出  $Y_k$  的变化最大不超过 0.25,势必对输出的影响最大,  $W_{2ij}$  经过了二级 Sigmoid 函数的作用,它对输出的影响次大;

(b) 比较式(9), (10), (11)可知,网络对连接权系数  $W_{1ij}$  阵的变化比连接权系数  $W_{2ij}$  阵的变化的承受能力强,网络对连接权系数  $W_{2ij}$  阵的变化比连接权系数  $W_{3ij}$  阵的变化的承受能力强,可见,网络对连接权系数  $W_{1ij}$  阵的变化比连接权系数  $W_{3ij}$  阵的变化的承受能力更强。这是由于 Sigmoid 函数有降低参数变化的作用,  $W_{1ij}$  经过了三级 Sigmoid 函数的作用,  $W_{3ij}$  只经过了一级 Sigmoid 函数的作用,当然  $W_{3ij}$  对输出的影响最大。

(c) 从式(12)可推出,输入  $x_i$  的变化对输出的影响不仅取决于神经网络的层数(该输入  $x_i$  经过了三级 Sigmoid 函数的作用,神经网络的层数越多,输入对输出的影响越小),而且取决于权值阵的乘积的结构(该神经网络权值阵的乘积的结构为  $W_{3ij} W_{2ij} W_{1ij}$ )。如果这个乘积(和)的绝对值很小,则  $X_i$  的变化将不会影响输出的变化;如果这个乘积(和)的绝对值很大,则  $x_i$  的变化将会引起输出的变化。例如对于上述的三输入、一个输出的神经网络(网络有一个输入层、两个隐层、一个输出层),设  $W_{3ij} W_{2ij} W_{1ij} = [a_1 a_2 a_3]$ ,当  $a_i$ , ( $i=1, 2, 3$ ) 很小时,输入向量中  $x_i$  ( $i=1, 2, 3$ ) 的变化不会影响输出变化,当  $a_i$  很大时,输入向量中  $x_i$  的变化将会影响输出变化(如果网络是  $M$  个输入  $P$  个输出,则  $W_{3ij} W_{2ij} W_{1ij}$  矩阵是  $P$  行  $M$  列矩阵,  $a_i$  ( $i=1, 2 \dots M$ ) 代表  $W_{3ij} W_{2ij} W_{1ij}$  矩阵中第  $i$  列的  $P$  个元素的和),因此,可根据  $W_{3ij} W_{2ij} W_{1ij}$  矩阵的结构定性判别网络对输入变化的容错能力。

根据以上分析,IBP 在协调器中作了以下工作:

1) 在考虑输出对输入的灵敏度时,采用调节各层权值矩阵元素值的大小来实现。例如,在上述神经网络中,连接权系数阵  $W_{3ij}$  对输出影响比连接权系数阵  $W_{2ij}$  和  $W_{1ij}$  对输出影响都大。因此,在建模过程中,协调器负责对修正  $W_{3ij}$  的学习因子进行切换调整,在程序



中,对式  $W_{3ij}(k+1) = W_{3ij}(k) + a\delta_i O_{pj} + b(W_{3ij}(k) - W_{3ij}(k-1))$  中的学习因子  $a, b$  进行控制,在模型和网络之间的误差  $e > \epsilon$ , ( $\epsilon = 10\% Y_p$ ) 时,  $a, b$  在  $0 \cdots 1$  之间选相对大些的值,让  $W_{3ij}$  按负梯度方向快速增加,使网络的输出快速变化;在模型和网络之间的误差  $e < \epsilon$  时,  $a, b$  在  $0 \cdots 1$  之间选相对小些的值,让  $W_{3ij}$  按负梯度方向缓慢变化;使网络的输出的变化不至于造成过冲。

2) 由于可根据  $W_{3ij} W_{2ij} W_{1ij}$  矩阵的结构定性判别网络对输入变化的容错能力,本文由网络的协调器控制,在修改权值的软件中将各层权值矩阵的学习因子错开,以便使各层权值矩阵的值错开。可让  $W_{3ij}$  的值相对小些,原因有两个:其一既利用它的灵敏度又防止  $W_{3ij}$  的值变化对输出引起过大的变化;其二可使  $W_{3ij} W_{2ij} W_{1ij}$  矩阵的值小些,从而提高网络的抗干扰能力。

正向建模中,IBP 网对干扰系数  $d = 0.05$ , 干扰频率系数  $A$  从  $10-100$  之间随机变化,且根据工艺要求不同,相邻子系统的耦合  $f((Y_{i\pm 1}(k)))$  的系数  $q$  在  $0-0.01$  之间随机取值,输入为  $760^\circ\text{C}$  时,系统的跟踪情况如图 3 所示。

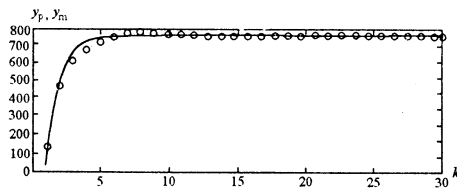


图3 正向建模中,IBP网络对随机干扰非线性系统的跟踪情况

3) BP 网络和改进的 BP 网络在正向建模中跟踪系统的均方差图。如图 4、5 所示。

从图 2 到图 5 可见,BP 网络对理想的线性系统的跟踪有大约 10% 的超调量,而 IBP 网络对随机干扰的非线性系统的跟踪只有 3% 的超调量。显然,IBP 网络对随机干扰的非线性系统的跟踪误差比传统的 BP 网络对理想的线性系统的跟踪误差小。本文还对其它类

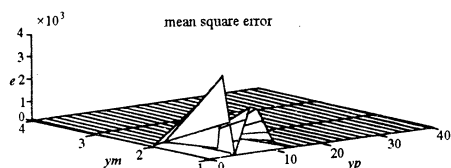


图4 BP网络对理想线性系统的模型跟踪的均方差

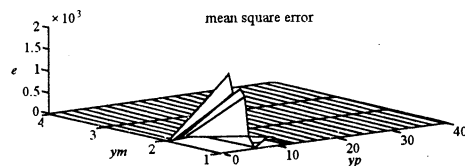


图5 RBP网络对随机干扰非线性系统的模型的跟踪均方差

型的非线性系统进行了仿真,均得到了类似上述的结果。所以说 IBP 网络具有较强的抗干扰性,是一种较理想的辨识建模网络,无论对线性系统的模型还是有随机干扰且输入随机变化的非线性系统模型,均能准确跟踪。

### 参考文献

- 1 Chen. S and Billings, S. A, Neural Networks for Nonlinear Dynamic System Modelling and Identification N. Int. T. Control, 1992, 56: 319-346
- 2 侯媛彬等. 一种能消除混沌现象的神经网络. 1996 年中国智能自动化学术会议, 内蒙, 1996. 8
- 3 孙德保, 高超. BP 网络的抗干扰性研究及误差分析. 华中理工大学学报, no. 8, 1994
- 4 Y. Linde, A. Buzo, R. M. Gray "An algorithm for vector quantizer design" IEEE Trans. Commun. COM-28 pp. 84-95 1980
- 5 A. Netravali, B. Prasada "Adaptive quantization of picture signals using spatial masking" Proc. IEEE 65, pp. 536-548 1977
- 6 S. J. Wan, P. Prusinkiewicz, S. K. M. Wong "Variance-based Color Image Quantization for Frame Buffer Display" COLOR Research and Applications 15(1) pp. 52-58 1990
- 7 M. T. Orchard, C. A. Bouman "Color Quantization of Images" IEEE Trans. Signal processing 39(12) pp. 2677-2690 1991

(上接 17 页)

### 参考文献

- 1 P. Heckbert "Color image quantization for frame buffer display" Computer Graphics 16(3) pp. 297-307 1982
- 2 G. Braudaway "A procedure for optimum choice of a small number of colors from a large color palette for color imaging" Electronic Imaging '87 San Francisco CA 1987
- 3 R. S. Gentile, J. P. Allebach, E. Walowit "Quantization of color images based on uniform color space" Journal of Imaging Technology 16(1) pp. 12-21 1990

# 一种快速而低复杂的调色板色彩组合选择算法

黎 越

(江苏省寻呼局 南京 210003)

王锁萍 秦 军

(南京邮电学院 南京 210003)

**摘 要** 本文介绍了一种快速而低复杂的调色板色彩组合选择算法。由于采用了匹配人眼视觉特性的空间活动性测度和梯度,在计算时间较短的前提下取得了很好的主观视觉质量。

**关键词** 矢量化 调色板 中值裂分

## 1 概述

许多图像显示设备都只能同时显示很小范围内的彩色集合,这部分彩色集合称为调色板,如典型的微机 VGA 只能同时显示 256 种颜色。这种显示设备由于存储量小而相对价廉,但为了显示真彩色图像(若每个彩色分量量化成 8bit,则有  $2^8 \times 3 = 16777216$  种色彩)我们必须选择一种好的映射方法,这种方法包括:选择一种最优的调色板色彩组合并且将每个像素的真彩色映射到调色板中,使得根据这种调色板色彩组合显示的图像和原图像相比具有小的失真度。选择一种优化的调色板色彩组合问题,实质上是一种矢量量化。即生成一种包括有调色板容量(如 256 个)码字的矢量量化表,然后对真彩色图像的每个像素进行矢量量化。图像色彩的量化问题是重建图像的显示质量,算法的复杂程度和压缩率三者之间的折衷。本文介绍一种快速,低复杂度但显示质量较好的算法。

## 2 彩色图像的矢量量化方法

在彩色矢量量化技术中,有一类方法是将初始选择的调色板进行不断叠代,优化比[1]-[3],其中的优化算法[4],就是著名的 Linde-buzo-Gray(LBG)算法。这种算法在每次叠代过程中力求减少原图像和量化后图像之间的方差和(total squared error(TSE0))直到达到局部最小方值为止。这种方法重建图像质量高,但计算量大,这对调色板的初始选取非常敏感。另一类方法是 Equitp 首先在灰度图像的矢量量化中提出的聚类技术。这种技术通过在彩色空间中不断地合并簇对使得图像色彩被归组到逐渐增大的簇中,合并的标准是使得由于合并簇和造成的 TSE 的增加达到最小。还有一类算法称为裂分算法[1][6][7],这类算法通过不断地将彩色空间裂分成越来越小的区域,然后将图像色彩归组到这些区域中。合并的标准是使得由于裂分而造成的 TSE 的减小程度达到最大。本文采用的中值裂分算法也属于裂分算法,但不以 TSE 作为裂分标

准。

值得说明的是,无论是用聚类算法还是用裂分算法得到的调色板都可以用 LBG 算法来作最后的优化。但这些算法都没有考虑人眼的视觉特性,因此,即使进行了不断的优化,得到的重建图像的主观视觉质量不一定是最优的。

## 3 彩色空间的中值裂分算法

彩色空间采用 R,G,B 坐标系,每个基色用 8bit 量化构成  $256 \times 256 \times 256$  彩色立方体。对于彩色立方体中的每个单元(即一种色彩)我们首先统计图像中具有这种色彩的像素数目,然后开始裂分。裂分的步骤如下:

首先选择 3 个坐标轴中的坐标轴  $i$  来分裂彩色空间; $i$  的选取标准是沿着坐标轴  $i$  的数据比沿着其它的坐标轴有更大的方差,然后,计算沿着坐标轴  $i$  的数的中值  $t_i$ ,用通过  $t_i$  且垂直于坐标  $i$  的平面将彩色空间分成两个子区域。上述步骤不断进行,注意每次裂分的区域是在整个彩色空间中包含有最多像素个数的区域,直至最得到 256 个子区域为止。将这 256 个区域的每个核代表的色彩作为安装硬件调色板的输入,然后,我们为原图像中每个像素计算一个指向调色板的指针,根据这个指针,即可用 256 色的调色板显示真彩色图像的每一个像素。

图 1 中显示了三次裂分过程,为图解方便起见,将彩色空间的裂分投影到 G-B 平面来表示。实双线代表分割平面的投影,虚线代表具有最大数据方差方向的投影,圆点代表裂分后各区域的核的投影。

预量化步骤:

由于 VGA 的调色板结构是每个调色板单元的 R,G,B 分量各用 6bit 存储器,因而在算法中对每个分量用 8bit 量化是浪费的。另外,我们注意到,人的视觉系统对两种色彩的绿色分量的差别要比红色分量的差别敏感,对蓝色分量的敏感程度最低。因此,对彩色信号蓝色分量量化成 4bit,红色分量 5bit,绿色分量 6bit,与三种信号都量化成 6bit 相比,这不仅使总的

色彩数目大大减少,减少了算法的执行时间,同时也保持人眼的视觉系统敏感部分色彩的较高分辨率。

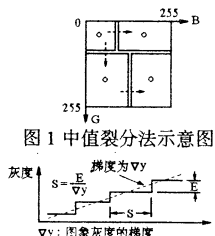


图1 中值裂分法示意图

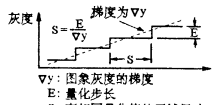


图2 伪轮廓的形成示意图

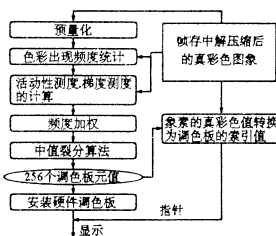


图3 调色板色彩组合选择算法流程图

#### 4 空间活动性测度和梯度测度

人的视觉系统(HVS)对图像平滑部分的量化噪声比对活动性强的部分如边缘要敏感,因此,对于彩色图像的平滑部分必须精确量化。在我们的调色板色彩组合选择算法中必须有足够多的独立的调色板单元来显示这部分色彩。换句话说,在彩色空间中,裂分时必须保证包含这些色彩的区域被分裂成尽可能小的区域。而对于图像中活动性强的部分可以较粗略地量化,裂分时可以为这些部分的色彩分配较大的区域,在总区域数不变的情况下,匀出较多区域数给平滑部分的色彩,从而达到总体视觉效果最佳的目的。

##### 活动性测度

对于 $8 \times 8$ 的子块1来说,它的活动性测度 $a_1$ 定义如下:

$$a_1 = \frac{1}{64} \sum_{B=B_1} |G_s - \bar{G}_1|$$

$B_L$ 代表块1内所有象素的位置, $G_s$ 代表象素位置 $s$ 处的灰度值, $\bar{G}_1$ 代表块1内的所有象素的平均灰度值。应该注意的是,一种色彩可能出现在一幅图像的多个不同的块中,因此,对于一种特定色彩 $X$ 来说,它的唯一的活动性测度定义如下:

$$a_x = \min_{1 \leq B \leq B_x} a_1$$

$B_x$ 代表含有色彩 $x$ 所有子块的集合,选择块的活动性测度的最小值的理由是,我们希望算法对平滑部分的色彩比较敏感,即不管这种色彩是否在其它活动性较高的部分出现过,只要它有一次出现在活动性低的部分,就将这个活动性低的块的活动性测度作为该色彩的活动性测度。

在中值裂分算法中,如果色彩 $x$ 在整幅图像中出现的频度为 $Q_x$ ,我们根据这种色彩的活动性测度对它的频度进行加权:

$$\bar{Q}_x = Q_x * W_x$$

$Q_x$ 为加权后的频度, $W_x$ 为加权系数,可定义为:

$$W_x = \begin{cases} W_1 & \bar{a}_x \geq \tau_1 \\ W_2 & \tau_1 > \bar{a}_x \geq \tau_2 \\ W_3 & \bar{a}_x < \tau_2 \end{cases}$$

其中 $W_1 < W_2 < W_3$ , $\tau_1, \tau_2$ 是活动性测度的阈值,分别将强活动性色彩,中等活动性色彩和弱活动性色彩分离。

从上式可以看出,活动性低的色彩加权后的频度比活动性高的色彩频度大,在中值裂分算法后,包含有活动性低的色彩的彩色空间区域裂分的可能性变大,量化矢量减小。

##### 梯度测度

对于象素灰度值缓慢变化的部分(如人的脸部),粗糙的量化会造成明显的伪轮廓,示意图如图2。从图中可以看出,由于表示平缓变化部分(具有灰度梯度 $Dy$ )的量化色彩不够多,原本平缓变化的部分变成了阶梯状,在平面图像上的表现就是伪轮廓。

假设位置 $(i, j)$ 处色彩的灰度值为 $x(i, j)$ ,我们定义该处的梯度为:

$$\nabla x = |x(i, j) - x(i-1, j)| + |x(i, j) - x(i, j-1)|$$

采用这种简化了的计算公式是为了节约计算时间。

设色彩 $x$ 在整幅图像中出现的频度为 $Q_x$ ,我们根据这种色彩的梯度测度对它的频度进行加权,首先设定一个阈值 $\tau$ ,则加权后的色彩的频度为:

$$Q_x = \begin{cases} W_x * Q_x & \nabla x < \tau \\ Q_x & \nabla x > \tau \end{cases}$$

$W_x$ 为大于1的常数。

从上式可以看出,图像平缓变化的部分(灰度梯度小于 $\tau$ )的色彩加权后的频度增大,中值裂分时被裂分的可能性变大,从而达到了量化精细,减小伪轮廓出现和可能性的目的。图3是整个调色板组合选择算法的流程图。

#### 5 实验结果

实验中采用的图像为C L A图尺寸 $352 \times 288$ 真彩色。为了评价系统的性能,我们定义均方误差(MSE)如下:

$$MSE = \frac{1}{N} \sum_{s \in A} (X_s - \bar{X}_s)^2$$

其中 $X_s$ 为位置 $s$ 处的真彩色象素的色彩矢量, $\bar{X}_s$ 为对应的量化矢量,即调色板的显示色, $A$ 为具有一定特性的图像分区, $N$ 为分区 $A$ 内的象素数目总和。

表1 两种算法的均方误差比较

A	未考虑活动性测度	考虑活动性测度
整幅图象	614.22	619.35
强活动性分区	979.30	2070.59
中等活动性分区	1216.06	1270.92
弱活动性分区	80.667	74.30

从表1可以看出,考虑活动性测度后,弱活动性分区的均方误差减小,这是以强和中等活动性分区的均方误差的增大为代价的,主观视觉效果大大改善,图片略。

(下转15页)

# 工作站环境中电路的蒙特卡洛分析

张文明

(长沙国防科大 406 教研室 410073)

**摘 要** 本文详细介绍了工作站环境电路蒙特卡洛分析的内容及基本步骤,并在 SUN 工作站上用 SPICE 给出了实例说明。

**关键词** MONTE CARLO PDF CDF EDA

## 1 引言

随着电路的复杂度和频率的不断提高,单纯靠人工设计、人工调试电路的工作量过大,周期过长,已不适应现代系统设计、电路设计分析的需要,逐渐被 EDA (Electronic Design Automation, 电子设计自动化) 手段所代替。在电路的设计和模拟中,通常是从已知的激励源及给定电路中各元器件的值,来分析和计算电路的工作状态及其响应特性,但将这些计算结果与实际电路的测量值相比较,结果往往不符合,其中除了测量设备的测量误差外,主要是因为实际电路中的元器件参数值往往与设计计算的值不一致造成的。设计计算的值称为标称值,而实际使用的元器件的值是在一定容差范围内的随机数值,由于容差的存在,电路的容差分析成为现代电路设计中不可缺少的关键工作之一。所谓容差分析,是为电路设计方案确定电路中元器件的容差变化范围,即元件的容差,其分析可分为两类:一是分析问题,即给定元器件、电源及温度等参数的容差,计算电路特性的容差,以验证是否符合设计要求;二是设计问题,是给定电路特性指标的范围,求出所用元器件及电源等的容差,验证设计方案、工艺实现及成本是否适宜等,因为容差越大,元器件价格越低,但电路的可靠性及成品率也将降低,所以应在保证电路性能指标范围内尽量扩大元器件容差范围。由于容差设计问题没有唯一解,在电路模拟中要解决容差设计问题,通常通过容差分析问题进行反求。蒙特卡洛 (MONTE CARLO) 分析是容差分析方法中最基本的一种。

## 2 工作站环境电路的蒙特卡洛分析

现在公认的电路分析程序是 SPICE,下面以 SUN 工作平台上的 EDA 工具 Cadence 中的 SPICE 为例进行说明。

电路的蒙特卡洛分析法是使用随机数发生器按具有容差的参数值的概率分布来选择各参数,通过大量的模拟次数对电路进行分析,所以蒙特卡洛分析可在元器件模型参数赋给的容差范围内,进行各种复杂的

分析,包括直流、交流及瞬态特性分析。这些分析结果可用于预测电路生产时的成品率及成本等。

电路的蒙特卡洛分析基本由以下几个方面组成:

- 电路图的输入
- 激活输入输出设备及仿真控制器
- 额定值仿真分析
- 激活蒙特卡洛分析仪及进行蒙特卡洛分析

以下对几个方面进行详细描述。

### 2.1 电路图的输入

电路图的输入一般为逻辑电路的图形输入,而较少进行文件编辑输入。在对电路图进行准确的输入后,需对各元器件及变量的容差分布进行选择及说明。容差分布描述分为 PDF (Probability Density Function, 概率密度函数) 和 CDF (Cumulative Density Function, 累积密度函数) 两种说明方式,SPICE 程序提供的标准容差概率分布可分为如图 1 所示的以下四种 PDF。

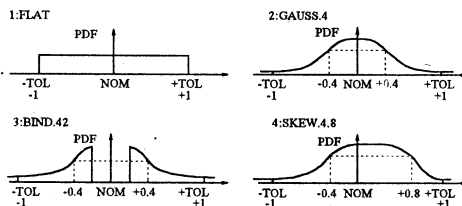


图 1 四种 PDF 情况

图 1 是由程序提供的四种标准 PDF,如一电阻为 100 欧姆,其容差为  $\pm 10\%$ ,则电阻值分布为 90—110 欧姆之间,如其 PDF 为 FLAT 型,则表示在 90—110 欧姆之间取值的可能性是一样的。通常元器件及变量概率密度接近于高斯型 (Guass) 分布。如元器件及变量概率密度与上述几种不一样,设计者可以自己任意设计 PDF,程序提供三种方式设计 PDF:

- Curve Generator
- Trace converter
- Raw Data Filter

容差可以进行全局说明,也可对元器件进行单个说明。全局说明包括缺省变量 (Default Variables) 和用户变量 (User Variables),缺省变量说明全局电子、电



容、电感等全局容差,用户变量说明温度等容差;对元器件的单个说明包括容差开关(即仿真时是否允许本元件存在容差)、概率密度函数及元件的容差。

## 2.2 激活输入输出设备及仿真控制器

SPICE 利用软件模拟各种仪器设备。在进行直流或瞬态仿真前,需调用 Funtion Generator 产生各种输入信号(可调用各种波形或绘制各种波形作为输入信号)并设置仿真周期,把输入信号通过各个通道连接到电路图的信号源上,调用直流表和示波器等输出设备观测感兴趣的输出波形,必须将输出信号连接到输出设备的各个通道上才能进行观测,可同时连接 16 路输入信号及 16 路输出信号,在输出设备上还可测量输出信号各点的输出值;在进行交流仿真前,需调用 Freuency Sweeper 产生各种扫频信号,确定扫频范围,通过各通道连接到电路图的扫频源上,调用网络分析仪观测并测量交流仿真的输出。仿真控制器设置各种仿真条件如迭代次数等,控制仿真开始或结束,在仿真开始时对电路进行编译,提示编译错误,在仿真开始后提示仿真进度。

## 2.3 额定值仿真分析

在设置好输入输出设备后,就可利用仿真控制器进行额定值仿真分析,即按照元器件及变量的 NORMAL 值进行仿真。在仿真开始时先对电路进行编译,如有错误,必须返回第一步对电路图进行修改直到编译通过,并注意对仿真的各种参数的设置。在仿真完成后,在输出设备上可以看到输出波形,虽然输出设备只有 16 路通道,但可以随时修改各通道所连接的信号,所以可以看电路上任意一点的输出波形。

## 2.4 激活蒙特卡洛分析仪及进行蒙特卡洛分析

在完成额定值仿真分析后,就可以进行蒙特卡洛分析。首先激活蒙特卡洛分析仪,以控制仿真次数和提示仿真进度。仿真次数当然是越多越精确,但要考虑到时间的因素。

蒙特卡洛仿真按概率密度随机取元器件及变量值进行仿真,所取的值与以下几个方面有关:

- 元器件及变量的 NORMAL 值大小
- 正负容差大小
- 随机数产生器产生的随机数
- 概率密度函数的设计及选取

随机参数值获得后,一次次进行仿真。输出设备随仿真的进行不断的改变,以描述当前仿真的输出波形。仿真完成后,通过对蒙特卡洛仿真器与输出设备的通道的连接,就可以得到输出设备上各通道信号的蒙特卡洛结果,以 CDF 与 PDF 两种形式表示。

## 3 蒙特卡洛分析实例

仿真的低通滤波器的电路图如图 2 所示。

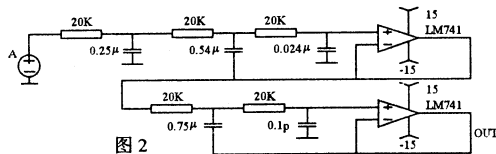


图 2

将 A 连接至扫频范围为 50Hz 至 2KHz 的扫频仪,设置电阻、电容容差为 10%,温度为 27℃,容差为 ±10%,概率密度函数设为 FLAT 或 BIND 型观测输出信号 OUT 的 3db 带宽=1KHz,增益=-2.889db。利用蒙特卡洛分析仪对电路进行 500 次仿真,得到结果如图 3 所示。

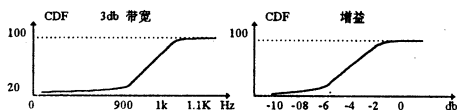


图 3

## 4 结束语

随着 EDA 技术的广泛应用,电路的软件分析逐渐代替了大部分实验室的硬件调试内容,而成为系统设计人员所必须掌握的一门技术。蒙特卡洛分析是电路软件分析的一项重要内容,它的主要优缺点如下:

优点:

- 能取得硬件测量所无法得到的数据,提高可靠性与成品率。
- 在很复杂的条件下,该法是得到一个现实解的唯一方法。
- 此法对各种广泛的问题均适用且具有相当的灵活性。

缺点:

- 因无法完全精确的描述元器件及变量容差的概率密度函数,计算存在固有误差。
- 当电路模型不精确时,结果有相当的误差。
- 计算量很大,所以花费较大。

## 本刊启事

为更快地推动、促进我国计算机产业的迅猛发展,及时地为计算机科研、开发、生产、管理等部门提供计算机行业的最新发展动向,加速高新尖端技术、最新软硬件的开发、移植、引进,及时地为广大读者及计算机爱好者奉献更多更新的计算机专业技术资料,并使众多作者、译者脱颖而出,《计算机应用研究》杂志社现正开展优惠出版各类计算机技术专著业务,欢迎广大作、译者踊跃赐稿。具体出版业务欢迎来函与我刊张钢编辑联系(邮编:610041 通讯地址:四川省成都市人民南路四段 11 号附 1 号《计算机应用研究》杂志社)

《计算机应用研究》杂志社启

# 基于 PC 总线的设备驱动式固态盘的设计

马 骏 孔振宇 赵万生

(哈尔滨工业大学机电工程学院 150001)

赵德开 初佃辉

(哈尔滨工业大学威海分校计算机教研室)

**摘 要** 本文介绍了一种基于设备驱动程序的固态盘的实现方法,并给出了普遍意义上的硬件原理及设备驱动程序框图。该实现方法具有灵活性和可移植性的优点,能够胜任工业环境下对程序及数据存储设备的可靠性及速度等方面的要求。

**关键词** 固态盘 存储器 设备驱动程序 工业控制

## 0 引言

工业环境下,控制系统对其程序和数据外存的性能提出了很高的要求,主要表现在速度和可靠性两个方面。但现有的外存设备软硬盘装置在这两个方面都比较欠缺。首先,盘驱动器是精密机械电子装置,是微机中最容易磨损的部分,可靠性和耐用性都比较较差。其次,磁盘的访问是一个低速的机械寻道过程,而在个人机中应用最广泛的 IDE(Integrated Device Electronics)接口其最大的数据传输率为 5Mbps,亦即其访问每字节的时间最小需要  $1\mu\text{s}$ ,而访问内存的速度是在 ns 级(80~120ns),这样就造成了低速 IO 和高速 CPU 处理之间的“瓶颈”,影响着整个系统的速度。如何做到工业环境下控制程序的可靠运行以及保证数据的高速可靠传输,一直是人们探索的焦点。

## 1 设备驱动式固态盘的提出

针对上面所提出的问题,如果用半导体存储器代替磁性材料作为存储介质,来模拟软硬盘的操作,则可解决上述问题,我们称这种卡板为固态盘。我们知道半导体存储器的数据存取是简单易行的,是高速度的,把半导体存储器芯片组合成存储介质,再配以读写电路逻辑,由 PC 扩展槽的地址总线、数据总线和控制信号线就可以选通控制存储器单元进行数据的存取,因为没有了机械部件,而且因为半导体存储器是比磁性存储介质快得多的存储介质,所以固态盘的存取速度要比磁盘快得多,而且还具有可靠性高、维修简单、实现容易的特点。

在微机中,如果用户要使用一个特定的设备,一般说来,需要做一个复杂的软件控制子程序。不过一般情况下这个操作可以借助两种方法来完成,一是直接使用 ROM-BIOS 的功能调用,二是使用 DOS 中的设备服务程序。大家都知道,DOS 是建立在系统 ROM-BIOS 基础上的一种操作系统,它由三层模块组成:界面层-DOS-Shell,核心层-DOS-Kernel 和接口层

-DOS-BIOS。DOS-BIOS 是基本输入输出系统的管理接口模块,其中包括一系列标准的字符设备和块设备的驱动程序,DOS 的核心模块可以用一个特定的数据结构——请求标题来调用这些驱动程序,而且这些驱动程序本身又可以通过控制命令的形式调用 ROM-BIOS 中的硬件设备驱动子程序。DOS 本身管理着很多设备,当 DOS 服务于一个需要访问设备的请求时,DOS 就按照一组标准规则翻译此请求,这些规则对所有的设备都是一样的,不管对于字符设备还是块设备,DOS 把所请求的服务功能变成一条特殊的命令,然后把它传递给一组处理该命令的子程序,这些子程序就是 DOS 的设备驱动程序。利用 DOS 编写的设备驱动程序具有灵活性和可移植性。基于这种想法,把固态盘看作是微机的一个外部设备,并编写与其相应的设备驱动程序,就可利用 DOS 实现对固态盘的灵活管理。

## 2 硬件原理简介

固态盘是用存储器芯片作为存储介质来模拟软硬盘操作的一种硬件卡板。通常可分为 ROM 盘和 RAM 盘两种。由只读存储器组成的固态盘叫 ROM 固态盘,盘只供读取而不可改写,断电后里面的信息不丢失,就象一个带有写保护的磁盘。可用于存放系统的控制程序。由随机存储器组成的固态盘称作 RAM 盘,它可以读写,但在断电以后其中的数据会丢失,因此一般用于暂存或缓存一些中间数据。如果采用带有后备电池的或闪速存储器 flash RAM 来构成 RAM 盘,则可保证信

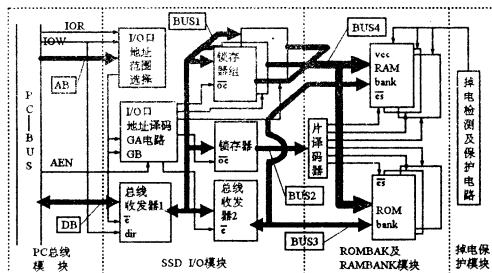


图1 固态盘硬件原理框图

息在断电后的不易失性。

固态硬盘的硬件原理图如图1所示,整个硬件部分由四大功能模块构成:PC总线模块,固态硬盘I/O模块,ROM/RAM存储器模块和掉电检测及保护模块。下面分别介绍这几个模块的功能。

(1)PC总线模块是连接固态硬盘和微机信息的桥梁,负责固态硬盘和微机之间的数据的传送。若只需进行8位的数据传输,则只需要采用PC/XT总线,若希望更高的数据传输率,则可采用16位带宽的PC/AT总线结构,这里只讨论8位的PC/XT总线结构。

(2)固态硬盘I/O模块。这个模块主要负责固态硬盘内部存储器地址的选择,以及数据的输入和输出。由PC总线模块传来的地址和数据信息,经过这个模块以后,就可以自动的分离成对存储器模块进行操作的地址和数据信息。BUS1为分时复用的地址和数据总线,BUS2为固态硬盘内部的高位地址总线,BUS3为内部的数据总线,BUS4为内部低位地址总线。

(3)RAM/ROM模块。这是固态硬盘中最重要的模块,由ROMBANK和RAMBANK,片选择码器和固态硬盘的内部地址总线和数据总线构成。它接收锁存器组和片选择码器来的地址和片选信息并将数据信息通过一定的读写逻辑输入输出到总线收发器2。

(4)RAM盘掉电检测及保护模块。本模块的主要作用在于实时的检测系统的电压,在系统掉电时能够保护RAM盘中的数据不至于丢失。

本硬件结构的显著特点是采用了模块化设计,且保证了固态硬盘不占用任何的系统内存地址,真正作到了零系统内存。只占用了少量的几个I/O口地址,使占用系统机的资源达到了最低程度,大大增强了其互换性。

### 3 固态硬盘设备驱动程序及其编写

#### 3.1 DOS设备驱动程序

每一个连接到PC机上的设备都必须建立其符合Microsoft设计规范的设备驱动程序,并在DOS的启动过程中进行加载,这样才能完成DOS对设备的协调管理。每一个服务请求,无论多复杂,最终由DOS将其转换成为一系列简单的驱动程序命令并传递给相应的设备驱动程序。DOS管理的设备有两种:字符设备和块设备。一次传送一个字符的设备称为字符设备,以字符组方式传送数据的设备称为块设备。一个标准的设备驱动程序由五部分组成:设备标题(Device Header)、数据存储和局部过程、策略过程(Stratgy)、中断过程(Interrupt),以及命令处理程序。

设备标题部分位于驱动程序的首部,其中包括有关设备驱动程序本身的信息,例如设备名称、指向下有关驱动程序的指针等。驱动程序的第二部分用于存储局部数据变量、局部子程序以及过程。策略过程和中断过程组成一体,用来处理DOS传送给设备驱动程序的

每一个命令,使DOS能够将控制传递给驱动程序。设备驱动程序的最后一部分是每一条命令的实际处理程序,而这些命令就是DOS传送给设备驱动程序的,它是设备驱动程序的核心部分。

#### 3.2 固态硬盘的设备驱动程序

所有的磁盘,不管是软盘、硬盘,都属于DOS的块设备。我们现在所说的固态硬盘,也应属于DOS的块设备,因为它是以字符组的方式进行数据传输的,因此它们的驱动程序也应符合Microsoft为块设备驱动程序所定义的规范。固态硬盘设备驱动程序是在一定硬件的基础上模拟一个硬件磁盘的功能,它也应该包括DOS设备驱动程序的五大部分。它的设备驱动程序框图如图2所示。限于篇幅,下面着重介绍其中的重点部分。

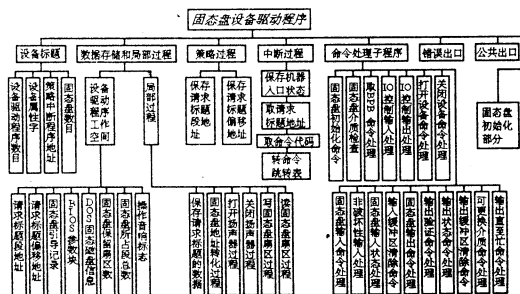


图2 固态硬盘设备驱动程序框图

DOS要求的设备请求标题定义设备驱动程序数目,设备的属性,策略程序的地址,中断程序的地址以及固态硬盘设备的数目。设备请求标题的第一个字段定义了应该指向下一个设备驱动程序的双字指针,对于有ROM盘和RAM盘两部分的固态硬盘来说,其驱动程序也应分为ROM盘和RAM盘两部分,将RAM盘驱动程序的段和偏移地址放在第一字段中;对于只有ROM盘或只有RAM盘的固态硬盘,则这一字段用-1来代替,表示它的后面没有其它的设备驱动程序。

数据存储和局部过程包括两部分:一是设备驱动程序的工作空间,它定义了固态硬盘设备驱动程序中使用的变量,其中最重要的是固态硬盘的引导记录和BIOS参数块BPB,固态硬盘要作到兼容DOS的磁盘操作,其数据的组织形式与磁性磁盘必须是相符的,由四部分组成:开始是引导区,紧接着是1到2个文件分配表(FAT),再就是文件目录,最后才是固态硬盘的实际数据存储区,如图3所示。对于ROM盘,它的格式是通过分割磁盘得到的,和某种磁盘的类型是相符的,因此在设备驱动程序中不用重新定义。对于RAM盘,应根据其实际容量确定这些参数,并用这些参数进行RAM盘的格式化。固态硬盘参数块BPB分为8个域,占用13个字节。这8个域分别是:每扇区字节数、每簇字节数、引导扇区数、FAT表个数、文件目录项数、总扇区数、介质描

述符、每个 FAT 占用扇区数。这些参数的选取应根据实际固态盘的容量进行；第二部分定义了六个局部过程：即保存请求标题数据过程(save)、固态盘地址转换过程(add-cvt)、打开扬声器过程(bell-on)、关闭扬声器过程(bell-off)、写固态盘过程(ssd-in)、读固态盘过程(ssd-out)，save 过程用来保存来自请求标题的数据，以备命令处理子程序使用；add-cvt 过程将扇区的信息转换为固态盘内存地址，并计算出要传输的字节数，送到 ssd-in 或 ssd-out 过程；ssd-in 过程将 add-cvt 过程中得到的地址和字节数通过 SSD-IO 模块分离成存储器的片选信号和片内地址，把微机内存数据缓冲区的内容写到固态盘相应的“扇区”中；ssd-out 过程与 ssd-in 过程的作用正好相反，它是把固态盘“扇区”的内容传送到数据缓冲区。为了在固态盘的操作中产生一个与硬盘或软盘一样的可闻可见的感觉，过程 bell-on 和 bell-off 分别负责在对固态盘进行操作时开启和关闭扬声器。

DOS 要求设备驱动程序处理一个命令时，DOS 要两次调用设备驱动程序，第一次 DOS 将控制传送给策略过程，策略过程将 es:bx 中请求标题的段和偏移地址分别保存在两个变量中，为下一次 DOS 调用作好准备。

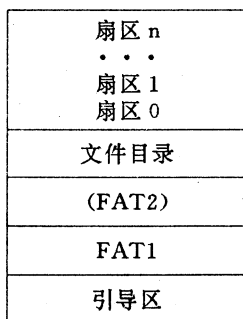


图 3 固态盘数据组织形式

DOS 在第二次调用设备驱动程序时，进入中断过程。它首先利用策略过程保存的请求标题的地址，从请求标题中取出命令代码，利用公式：

$$\text{命令表首地址} + 2 \times \text{命令代码} = \text{表地址}$$

计算出该命令在命令处理表 CMDTAB 中的地址，然后查表找到该命令处理子程序的入口地址并跳转到

该命令处理子程序。

DOS 的命令处理程序是设备驱动程序中最重要的部分，DOS 对设备的请求最终要转移到各个命令处理子程序去调用 BIOS 中断来完成。固态盘命令处理程序包括了初始化命令处理、介质检查命令处理、取 BPB 命令处理、I/O 控制输入命令处理、输入命令处理、输出命令处理、I/O 控制输出命令处理、输出验证命令处理、固态盘输入命令处理、固态盘输出命令处理等共 17 个命令处理子程序，参见图 2，它负责了固态盘的全部操作。

固态盘设备驱动程序经过汇编、连接并转换成 COM 格式的文件 SSD.SYS 后，将其挂到 CONFIG.SYS 文件，即在 CONFIG 文件中加入如下命令行：

DEVICE=SSD.SYS

就可完成对固态盘的安装，DOS 将自动给固态盘分配盘号，紧接着系统现有的最大盘号。

#### 4 结论

本设计方法可以实现固态盘对磁盘工作的仿真，安装简易，只需把盘卡插到微机的扩展槽中，在 CONFIG 文件中加入固态盘的设备驱动程序，开机即可运行。不占用系统内存，只占用系统的几个 I/O 口地址。使占用系统资源达到最低程度，具有很好的互换性和可移植性。其容量在卡板的面积允许的情况下可以任意地扩充。很适合在某些恶劣的应用 PC 总线的工业控制环境下运行，具有高的可靠性及快的运行速度。也适合在一些仪器、仪表的无盘工作站，PC 总线商用收款机及 POS 系统，军用的车载、舰载、机载的计算机系统中应用。具有良好的应用前景。

#### 参考文献

- 1 张昆藏. IBM PC/XT 微型计算机接口技术. 清华大学出版社, 1992. 5
- 2 李志源, 郑南京, 沈金发. IBM-PC 长城微机基本输入输出系统. 清华大学出版社, 1993. 6
- 3 戴水贵. DOS/BIOS 功能调用及程序例. 海洋出版社, 1992. 8
- 4 马骏, 宋颖慧, 赵万生等. 工业环境下高性能固态盘的设计与实现. 机械与电子, 1995

(上接 38 页)

#### 5 结束语

本文所提出的思想与方法，是在为安徽新华印刷厂开发计算机生产周期管理系统的过程中摸索并总结出来的。考虑到此方法具有一定的适用性，既适合于印刷行业，也适用于机械行业，所以在此作了适当抽象介绍，希望能通过争鸣引出更好的方法。在具体开发过程中，引用了 FOXBASE 和 PROLOG 解释系统，综合利用 DBMS 的灵活性。由于多任务流模型包括了以销定

产和以产定销两种企业类型，所以本方法可应用于绝大多数中小型企业生产计划管理。本方法不仅可以处理工序顺序的加工方式，也能处理局部工序倒序的加工方式，而这是一般排序方法难以解决的。此外，经过有限扩充，还可以将批量处理方式纳入 MP-PERT 的处理能力之中。我们希望通过安徽新华印刷厂管理系统的实现，为进一步研制中小企业生产周期管理系统的开发环境奠定坚实的基础。



# 图形用户界面系统的层次结构模型

贺 鹏

(葛洲坝水电工程学院电气工程系 宜昌 443002)

**摘 要** 基于各种窗口系统的图形用户界面 GUI(Graphics User Interface)已经成为人们研究和开发用户界面的重要手段。本文结合图形用户界面的基本特征,提出了图形用户界面系统的层次结构模型并分析了实例,叙述了图形用户界面系统的模型标准,简要介绍了图形用户界面系统的开发工具。

**关键词** 图形用户界面系统 结构模型 应用程序接口 模型标准 开发工具

## 0 图形用户界面的基本特征

随着计算机平台技术的迅速发展和图形用户界面技术的迅速普及,图形用户界面(GUI)已经成为广大用户和计算机厂商一致关注的热点,到目前为止,虽然业界还没有对图形用户界面这一概念形成一个严格的定义,但图形用户界面的一些主要的基本特征已逐渐被人们所接受:

(1)直接操作 图形用户界面采用的是位映象图形显示技术,用户对应用程序的控制主要是通过操纵显示在屏幕上的图形对象来完成,这些图形对象如图符、窗口、菜单和按钮等,都是软件控制下的位映象图形实现。

(2)用户控制 应用程序的运行不再由编程安排好的过程来驱动,而是由用户通过图形用户界面引入的输入设备来移动光标或点触图形对象,实现对应用程序的直接操纵,这是一种事件驱动方式,它体现了人在控制应用程序运行中的中心地位。

(3)界面定制 图形用户界面一方面提供了可视的(如窗口、滚动条、控制面板等)和不可视的(如屏幕、颜色、字体等)界面对象,另一方面又允许用户根据需要对应用程序的界面进行剪裁和定制,如移动、缩小或放大窗口,设置颜色,字体或鼠标键等。

(4)信息反馈 图形用户界面的帮助机制能为用户操作界面提供导航,而它的保护机制又为系统的可靠运行提供容错排错。这两者都有大量的信息反馈。

(5)界面一致 图形用户界面系统作为一个完整的运行环境,应该提供一个不依赖于具体问题的界面设计标准,使得在该环境下运行的各种应用程序的界面风格及与用户交互的方式都具有良好的一致性。

(6)开发工具 开发工具是图形用户界面系统的一个重要组成部分,是用户在图形用户界面环境下开发应用程序的重要手段。

图形用户界面(GUI)是图形用户界面系统实质内容的主要概括,图形用户界面作为一个系统,其基本特征有些与图形用户界面系统各层次的模型相关,如界

面定制与窗口模型有关;有些与相应层次的模型的应用程序接口有关,如开发工具与用户模型的应用程序接口有关。为更清楚地认识这些特征,有效地使用图形用户界面技术,有必要对图形用户界面系统的结构模型有一个深入地了解。

## 1 图形用户界面系统的层次结构模型

图形用户界面系统通常是指由相应硬件平台和操作系统支持的显示模型,窗口模型和用户模型,以及由这三个模型的应用程序接口组成的应用程序接口 API(Application Program Interface),如图1所示。其中计算机硬件平台和操作系统是图形用户界面系统的基础。只有当硬件平台技术和操作系统功能发展到一定程度,用户才有条件使用图形用户界面。一般而言,某一种图形用户界面系统仅依赖于某种特定的硬件平台和操作系统,但 Motif 是可以在多种硬件平台和操作系统上运行的界面系统。

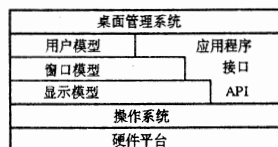


图1 GUI系统结构模型

面定制与窗口模型有关;有些与相应层次的模型的应用程序接口有关,如开发工具与用户模型的应用程序接口有关。为更清楚地认识这些特征,有效地使用图形用户界面技术,有必要对图形用户界面系统的结构模型有一个深入地了解。

图形用户界面的显示模型决定了图形在屏幕上的基本显示方式,即用位映象图形显示技术显示各种图形对象的方式。如定义在屏幕上创建字体(font)的字样(typeface)和尺寸,定义绘图程序中的线或曲线的产生方式,包括各种处理,像维数(dimension),浓淡处理(shading)等。不同的图形用户界面系统采用的显示模型各不相同。例如目前大多数在UNIX系统之上运行的图形用户界面都采用X窗口系统作显示模型,而MS Windows 3.1则采用Microsoft公司自己设计的图形设备接口(GDI)作显示模型。

窗口模型确定了在屏幕上显示的多个应用程序的形象,体现在程序设计上可表示为构造可移动(movable),可伸缩(resizable)的窗口、菜单、滚动条、对话框等。它通常包括两部分:一是编程工具,如函数集;二是对如何移动,输出和读取屏幕显示信息的说明。有一些

图形用户界面系统包含有独特的窗口系统,如 Apple Macintosh,另一些图形用户界面系统则使用公共的窗口系统,如 MIT 的 X 窗口系统。X 窗口系统不但可以充当图形用户界面系统的显示模型,也可以充当它的窗口模型。

图形用户界面的用户模型也称图形用户界面的外观与视觉(look and feel),它也包括两部分:一是构造用户界面的工具,如工具箱和框架集,包括对高层界面构件(Widget)或对象元的数据结构的定义和说明;二是定义在屏幕上组织各种图形对象以及这些对象之间的行为规范和协议。每个图形用户界面用户模型都应当说明它支持什么样的窗口和什么样的显示方式,因为定义规范和建立工具集必须针对具体的显示模型和窗口模型。

图形用户界面系统的应用程序接口 API 由显示模型、窗口模型和用户模型的应用程序接口共同组成,各模型的应用程序接口主要是该模型提供给开发者的开发工具。详细叙述见第 5 部分。

桌面管理系统是人机交互的图形化管理系统,它使得人机交互更加简便灵活,它通常包括以下几个组成部分:图形化的文件管理系统,供用户使用的图标库,桌面管理机构,图标库管理机构等。

## 2 层次结构模型实例

图 2 所示为两个基于 UNIX 操作系统的 GUI 产品的层次结构模型实例。

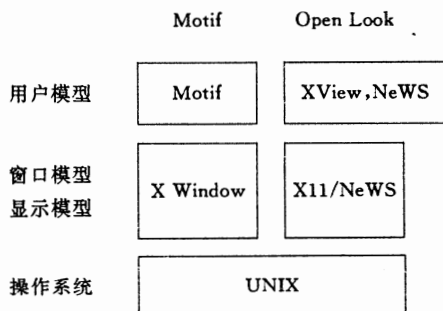


图 2 层次结构模型实例

Motif 是 OSF (Open Software Foundation) 于 1989 年推出的图形用户界面系统,它以 X 窗口系统作显示模型和窗口模型, Motif 本身位于用户模型层。Motif 由工具箱 (Motif Toolkit), 用户界面语言 UIL (User Interface Language), 窗口管理程序 (MWM (Motif Window Manager) 和风格指南 (Style Guide) 四部分组成。

Motif 的应用程序接口由它的显示模型与窗口模型的应用程序接口 Xlib 和用户模型的应用程序接口 Xt Intrinsics 及 Motif Toolkit 共同组成。见图 3 所示。Xlib 和 Xt Intrinsics 都是 X 窗口系统的组成部分,它们是 X 窗口系统提供给应用程序开发者的应用程序接口。其中 Xlib 是一个包含 300 多个 C 函数的程序

库,这些 C 函数提供了发送、接收和处理 X 协议请求的功能。而 Xt Intrinsics 是在 Xlib 之上的开发工具箱,它监控与最终用户交互作用相关的事件并分发正确的程序去处理这些事件,这是一种基于 Client/Server 的工作方式。它有两方面的功能:一是提供一组高级实用函数,以简化编写程序的难度,二是提供一套构造更高层界面构件的数据结构和方法。由于 Xt Intrinsics 仍然遵从 X“只提供机制不提供风格”的原则,所以应用程序界面的风格还需由它之上的对象元类集合 (工具箱和框架集) 来决定。一个具体的 X Toolkit 产品应该由 Xt Intrinsics 和它自己定义的对象元类集合组成,不同的 X Toolkit 产品有不同的对象元素集合。Motif Toolkit 就是这样一个产品,这除了 Xt Intrinsics 之外,还定义了自己的对象元类集合,以及操纵这个对象元类集合的简便函数。

与 Motif 一样,Open Look 也是一个描述用户界面的规范说明。基于此种规范的 Open Look 产品的显示模型和窗口模型是 X11/NeWS,它是一个窗口服务器和页描述技术的综合系统,能处理遵从 X11 或者 NeWS 协议的客户端消息。X11 是 X 窗口系统的第 11 版本,它是一个静态可扩展的窗口系统,也就是说,服务器的供应商可能提供扩展,但用户无法加载自己的扩展。NeWS (Network extensible Window System),是一个基于 POSTSCRIPT 页面描述语言的系统,用户可以定义自己的功能函数并加载到服务器中。处在用户模型层的工具箱和框架集除了如图所示的 XView Toolkit 和 NeWS Toolkit 之外,还有 Open Look Intrinsics Toolkit (OLIT)。XView 提供一个面向对象的函数集和开发 X11 应用程序的框架结构,NeWS 由一套服务器构件和一套客户构件组成,其中服务器构件用扩展 NeWS POSTSCRIPT 语言写成,而客户构件用 C 语言写成;OLIT 是基于 X Toolkit 工具箱的,它向用户提供了一套预定义界面构件和其他用户接口抽象。

## 3 图形用户界面系统的模型标准

最早的有关图形用户界面的研究工作是 Xerox Palo Alto 研究中心 (PARC) 于七十年代开始进行的,在此基础上,Apple 公司于 1984 年推出第一个带图形用户界面的 Macintosh 系统。之后,图形用户界面 (GUI) 技术被越来越多的软硬件生产厂商所采用,并已开发出数以百计的图形用户界面产品,这些产品可分为以 UNIX 为操作系统平台的系统和个人计算机的系统两类。前一类产品的主要代表有 OSF 的 Motif, Sun 公司和 AT&T 公司共同开发的 Open Look 以及 SCO 公司的 Open DeskTop 等;后一类的代表有 Microsoft 公司的 Windows 3.1, Apple 公司的 Macintosh 以及 HP 公司的 NewWave 等。

在图形用户界面领域中,为了规范众多图形用户

界面产品的应用和发展,必须要建立一个统一的能适合图形用户界面系统广泛应用的国际标准。按照国际惯例,确定一项软件技术标准,一般要遵循两个基本原则:一是通用性,即软件的可移植性或是与设备的无关性;二是能够被市场接受的程度。由于X窗口系统始终遵循“只提供机制不提供风格”的原则,并强调与设备的无关性,使得越来越多的UNIX操作系统平台的开发商都采纳X为窗口系统的标准,在市场上也占有越来越大的份额,所以X窗口系统已被X/Open和IEEE P1201.1确定为图形用户界面系统标准中的显示模型标准和窗口模型标准。

就目前而言,不同产品之间的区别除了显示模型和窗口模型的不同,还体现在用户模型的不同,主要表现在定义对象及对象元之间的行为规范或协议,用于构造高层界面构件的数据结构和方法,以及提供给应用程序开发者的工具箱和框架集等。如前所述,定义规范和建立工具集必须针对具体的显示模型和窗口模型,显然只有在已有标准的显示模型和窗口模型之上确定的用户模型才更有利发展。诸多产品,精彩纷呈,但就发展趋势来看,Motif已占优势,很有可能成为图形用户界面系统用户模型的标准。

#### 4 图形用户界面系统的开发工具

图形用户界面系统的开发工具是开发在图形用户界面环境中运行的应用程序的应用程序接口。不同层次的开发者使用图形用户界面产品所提供的不同层次的应用程序接口。根据层次结构模型提供的各层次的API,以Motif为例,可将开发工具分为如下几种。

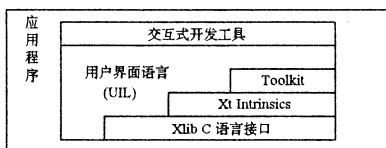


图3 开发工具的层次分类

(1)基本开发工具,如上图所示有Xlib,Xt Intrinsics,Motif Toolkit等,它们都是以高级语言的库函

数形式提供,面向专业开发人员的。这类工具的功能很强,应用形式灵活,许多高层的工具就是以它们为基础开发出来的。但编程量大,复杂度较高。

(2)原型开发工具 利用Motif开发的应用程序通常可分为界面代码和功能代码两部分,Motif的用户界面语言UIL特别适用于应用程序界面的原型开发阶段。其特点是只要用少量的描述性的语句即可编制出应用程序的图形用户界面,如果要对界面进行修改,也只需更改描述界面后原文件即可。减轻了界面设计与开发的工作量,有助于实现应用程序的界面代码与功能代码的分离。

(3)交互式开发工具 利用交互式开发工具开发界面的过程是全交互式的,不要求开发者编程,只要用鼠标或键盘对系统的提问作必要的回答和选择就可以设计出开发者所需的界面。这类工具也可以实现界面代码与功能代码的分离,使用面向对象的回调(callback)机制可将应用程序的功能代码与相应的对象连接起来,这类工具的典型代表有Sun公司的Development Guide,HP公司的Interface Architect等。

需要注意的是,上述几种类型的开发工具有内在的层次性,高层次的开发工具是用低层次的工具写成的,用户在用高层次工具的同时,可以使用低层次的工具。但用户开发应用程序的灵活性要比单纯使用底层工具差得多。

除了以上几种类型外,还有其他形式的工具,如可用于在多个图形用户界面系统上开发的工具和用于将在字符终端上开发的应用程序转换为可在图形用户界面环境下运行的转换工具等。

#### 参考文献

- 1 汪成为等. 面向对象分析,设计及应用. 国防工业出版社
- 2 刘英姿等. 用户界面的研究及设计方法. 计算机应用研究,1995.2
- 3 张倪等. Motif与图形用户界面开发工具. 清华大学出版社

## FOXPRO2.6 FOR WINDOWS Wizard 生成工具功能浅析

陈凤祥

(安徽师范大学计算中心 芜湖 241000)

贺茂林

(安徽师范大学数学系 芜湖 241000)

**摘要** 本文对FoxPro2.6 FOR WINDOWS下Wizard生成工具的功能作了一些简介,重点分析了用Wizard生成的屏幕文件中的有关代码片断,并对其中的不足给出了改进的办法。

**关键词** FoxPro Wizard 生成文件 屏幕文件

软件用户界面友好,功能强,系统可靠且便于维护。如果在 FOXPRO2.6 FOR WINDOWS 下开发软件使用其提供的生成工具 Wizard,可以省去不少代码的编写,缩短软件开发周期并提高软件的质量

### 1 Wizard 功能简介

Wizard 是 Microsoft 在一些较新版本的应用软件(如 Access, Excel, Foxpro 等)中新增的一个功能,它提供一种以图形交互的操作接口,让用户通过它所提供的各种格式去选择一些配件来组合,就可以自动迅速地完成:

- (1)数据库文件结构的定义
- (2)关系式查询文件(SQL)的建立
- (3)标签及报表文件的建立
- (4)信件合并
- (5)数据维护画面的设计
- (6)建立简易型的应用系统(含有增、删、修改、浏览查询、报表打印等)

Microsoft Foxpro2.6 提供的工具 Wizard 如下所列:

Clients-Server Query Wizard

Group/total report Wizard

label Wizard

Mail Merge Wizard

multi-column Report Wizard

Report Wizard

Screen Wizard

SQL QUery Wizard

Table Wizard

Updatable Query Wizard

利用这些工具生成文件时,使用起来很方便,而且也容易掌握,生成后的文件可根据需要修改,以符合应用程序的要求,这里不再多述。

我们重点讨论一下利用 Screen Wizard 工具生成的屏幕文件及有关代码。自动生成的屏幕文件由系统自动管理数据文件的打开与关闭,窗口的定义与释放,各控制按钮的功能与作用的时效性。

### 2 Screen Wizard 工具生成的屏幕文件中各函数与过程及相关代码简介

我们知道在屏幕生成器的代码层,包含有这样一些代码片段:

Screen Setup Code

Cleanup and Procedures

On Screen Entry(When)...

On Screen Exit(Valid)...

On Window Activate...

On Window Deactivate...

On Refresh(Show gets)...

用 Screen Wizard 工具生成的屏幕文件,不但在生成的屏幕上放置了所选的 GET 对象,还生成了这些代码片段中的大部分。在 Screen Setup Code, Cleanup

and Procedures, On Window Activate... 及 On Window Deactivate... 中生成了相关代码。这些代码包括了最常用的数据库操作功能,如增加、删除、编辑、报表打印、浏览记录和移动记录指针的操作。而且每个 GET 对象是用对应于字段名的内存变量,用户在其上作任何操作都需经认可后方可写入数据文件中,不必担心无意中破坏了数据。所以我们可以用 Wizard 生成屏幕及大部分代码片段的框架,然后调整屏幕并将根据需要编写的代码加到屏幕文件中的有关代码片段中去,使之符合应用程序的要求。比如,可以将每次执行 READ 后需刷新的内容放在 On refresh(show gets)的代码片段中,这些内容是不能通过执行 SHOW GETS 语句自动刷新的。不管什么时候执行一个 SHOW GETS,用这个方法可指定和 GET 刷新一起采取的动作。因此软件开发人员可以利用 Wizard 工具生成一些通用过程与函数并把其作为更复杂程序设计的起点。

在生成的 SETUP CODE 代码段中作了许多预处理,与整个屏幕相关的代码片段也都放置在 Screen Setup Code 中。如果还需增加一些与该屏幕有关的代码,请在该代码片段中设置。注意,在 Screen Setup Code 中不能有函数和过程。

在生成的 Cleanup and Procedures 代码段中,除了有关的清除代码外,还有如下一些过程与函数,它们的名字与作用如下所述:

PROCEDURE readdeac && 使当前 READ 下的窗口不活动时,执行该过程;

PROCEDURE readact && 激活一个 READ 窗口时,执行该过程,可以使用附加代码来处理 READ 窗口的初始活动;

PROCEDURE wizerrorhandler && 程序运行中出现错误后,此过程给出错误信息;

PROCEDURE printrec && 打印输出时调用的过程;

PROCEDURE BTN-VAL && 控制按钮的 VALID 校验调用此过程;

PROCEDURE REFRESH && 该过程控制 GET 对象是否有效;

PROCEDURE edithnd && 增加记录时将调用此过程进行有关操作的处理;

PROCEDURE fox-alert && 弹出一个对话框,返回逻辑值;

PROCEDURE pdialog && 打印预览与打印选择的对话框;

PROCEDURE loc-dlog && 浏览数据文件时调用的过程。

### 3 对 Wizard 生成的屏幕文件的一点改进

生成的上述标准函数与过程,有可能与应用程序要求有些不符,可以将自己写的代码,嵌在其中。

比如,打印输出时,系统默认的报表文件名为"alias()"+"frx",为能多次调用 PROCEDURE printrec 输出报表,我们可以将其改为一个有参过程,每次调用时带报表格式文件调用 PROCEDURE printrec。在 WINDOWS 环境下另外还可以将语句

REPORT FORM(ALIAS()) &pr-out &pr-record

改为:

REPORT FORM(ALIAS()) prompt &pr-out &pr-record

这样将弹出一个打印对话框,在打印输出报表时,可以对打印范围、打印质量及打印机等有关选项进行设置。

用 Wizard 生成的屏幕文件,用屏幕生成器将其生成源代码后运行,此时发生了一件使人困惑的事情。当数据文件为空时,运行该文件的过程中,屏幕上出现一个对话框,告诉你此时数据文件为空,是否要增加一条记录,倘若此时回答<NO>(不增加记录),进入用户界面后执行<ADD>操作,怎么也不能激活 GET 对象,此时屏幕上的按钮也只有<ADD>与<EXIT>有效,别的都无效。不管怎么选择<ADD>都无法激活 GET 对象,除非选择<EXIT>退出,然后在运行程序的过程中,当告之数据文件为空后,回答<YES>(增加一条记录),才可以在选择<ADD>按钮后激活 GET 对象,但对于刚才增加的那条记录,实际上是增加了一条空记录。而进入用户界面后,系统并未告之第一条记录为空记录,除非在选择<ADD>按钮前选择<EDIT>按钮,在第一条空记录处输入所需数据,否则,时间长了就会忘记在数据文件中存放了一条空记录。如果这样的程序交给不了解此种情况的人用,程序运行到后来,很可能出现难以预料。出现此种情况,原因何在呢?

我们知道运行一个由屏幕生成器生成的文件时,首先运行的代码片断是 Screen Setup Code 中的代码,在此代码片断中作了内存变量初始化的设置,其中设置了:m.isediting=.F. m.isadding=.F.

然后执行当前 READ 层次的 ACTIVATE 子句,即执行 On Window Activate... 中的代码,而在 On Window Activate... 中的代码是调用函数 Readact(),恰是调用了 PROCEDURE readact,仔细分析一下过程 PROCEDURE readact.

```
PROCEDURE readact
  IF ! isediting
    SELECT(m.wzalias)
    SHOW GETS
  ENDIF
  DO REFRESH
RETURN
```

此过程又调用了过程 REFRESH,请看 REFRESH 过程:

```
PROCEDURE REFRESH
DO CASE
CASE m.isreadonly AND RECCOUNT()=0 && 在只读且
                                     数据文件为空时
  SHOW GETS DISABLE && 所有 GET 对象均无效
  SHOW GET exit-btn ENABLE
                                     && 只有<EXIT>按钮有效
CASE m.isreadonly && 在只读状态
  SHOW GET add-btn DISABLE && 不能增加
  SHOW GET del-btn DISABLE && 不能删除
  SHOW GET edit-btn DISABLE && 不能编辑
CASE (RECCOUNT()=0 OR EOF()) AND ! m.isediting
                                     && 非编辑状态
  SHOW GET DISABLE && 所有 GET 对象无效
  SHOW GET add-btn ENABLE && <ADD>按钮可选
  SHOW GET exit-btn ENABLE && <EXIT>按钮可选
```

```
CASE m.isediting && 在编辑状态
  SHOW GET find-drop DISABLE && 改变按钮的状态
  SHOW GET top-btn DISABLE && 未改变其它 GET 对象的状态
  SHOW GET prev-btn DISABLE
  SHOW GET loc-btn DISABLE
  SHOW GET next-btn DISABLE
  SHOW GET end-btn DISABLE
  SHOW GET add-btn DISABLE
  SHOW GET prnt-btn DISABLE
  SHOW GET exit-btn DISABLE
  SHOW GET edit-btn,1 PROMPT "\<Save"
  SHOW GET del-btn,1 PROMPT "\<Cancel"
  ON KEY LABEL ESCAPE DO BTN-VAL WITH '
DELETE'
  RETURN
OTHERWISE
  SHOW GET edit-btn,1 PROMPT "Ed\<it"
  SHOW GET del-btn,1 PROMPT "\<Delete"
  SHOW GETS NABLE
ENDCASE
IF m.is2table
  SHOW GET add-btn DISABLE
ENDIF
ON KEY LABEL ESCAPE
RETURN
```

我们通过分析源代码,发现问题出在过程 REFRESH 上。看看当数据文件为空,激活屏幕时,m.isediting=.F.,此时满足 DO CASE... ENDCASE 结构的第三种情况,即

```
CASE (RECCOUNT()=0 OR EOF()) AND ! m.isediting
  SHOW GETS DISABLE && 所有 GET 对象无效
  SHOW GET add-btn ENABLE && <ADD>按钮有效
  SHOW GET exit-btn ENABLE && <EXIT>按钮有效
```

这样一来,所有 GET 对象除了<ADD>按钮和<EXIT>按钮外均无效了,而当选择了<ADD>按钮后,将执行过程 PROCEDURE BTN-VAL,此时满足过程 BTN-VAL 中的条件:

```
CASE m.btnname='ADD' AND ! isediting && add record
  isediting=.T.
  isadding=.T.
  =edithand('ADD')
  -curobj=1
  DO refresh
  SHOW GETS
  RETURN
```

其中调用的函数 edithand 如下:

```
PROCEDURE edithand
  PARAMETER m.paction
  * procedure handles edits
DO CASE
CASE m.paction='ADD'
  SCATTER MEMVAR MEMO BLNK
CASE m.paction='SAVE'
  INSERT INTO (ALIAS()) FROM MEMVAR
CASE m.paction='CANCEL'
  * nothing here
ENDCASE
RETURN
```

在这里未对 GET 对象的状态做任何改变,紧接



着再次调用 PROCEDURE REFRESH, 这次满足 PROCEDURE REFRESH 中的 CASE m. isediting 条件, 我们注意到在此条件下, 仍未对 GET 对象的状态做任何改变。因此, 这时的 GET 对象除部分按钮外, 仍是处于无效的状态, 因而出现前面所讲的情况, 即执行 (ADD) 操作后 GET 对象没被激活。为了解决这个问题, 在 CASE m. isediting 后, 有关控制按钮的控制操作语句前加上语句: show gets enable 问题也就得到了解决。

为了使问题简化, 在对 PROCEDURE REFRESH

作了上述修改后, 将 Screen Setup Code 片段中判断数据文件为空, 并提示用户增加一条空记录的语句删除。因为作了这样的改进后, 当数据文件为空时, 激活屏幕文件, 只有 (ADD) 和 (EXIT) 按钮有效, 别的按钮都无效。

Foxpro 2.6 for DOS 环境下 Wizard 生成工具的使用法与 Windows 环境下的用法基本一样。生成的屏幕文件的代码也基本一致, 除了有关 Windows 环境的设置外, 过程与函数的代码基本一样, 如果要修改生成文件的不足, 方法也是一样的, 这里不再多述。

## 用 VB 编程实现 Windows 下屏幕的快速截取

方 锐 朱秋萍

(武汉大学电子信息学院 430072)

接触过 Windows 的人往往想把一些精美的图形、图像保存下来, 以作它用。但是, 使用 Windows 本身自带的屏幕截取功能则存在如下两方面的问题。一是操作步骤多, 比较麻烦。首先得用热键 Alt+PrintScreen 把当前活动窗口的内容拷贝到剪贴板 (Clipboard) 上, 然后在画笔 (Paintbrush) 中把图形或图像从剪贴板上粘贴过来, 最后以位图格式存盘保存。二是不论用户截取的窗口有多大, 画笔总是按满屏的大小生成 BMP 文件, 于是截取一个小小窗口的内容以位图格式存盘后, 就要浪费许多存储空间, 这是很不合算的。

鉴于此, 本文尝试利用 Visual Basic 强大的图形功能编制了一个屏幕截取程序, 能方便、灵活地对屏上的内容按用户的需要进行截取, 并直接以 BMP 文件格式存盘, 而且该位图文件的大小依实际截取窗口大小而定, 从而避免了不必要的存储空间的浪费。

该程序操作简单, 屏幕截取快, 效果甚佳, 使 Windows 的屏幕截取功能得到了进一步的改进和完善。下面将详细介绍该程序的设计思想和具体实现方法。

### 1 操作过程

本程序第一次执行后, 若按下热键——空格键 (SpaceBar), 光标便变成十字形, 表明已进入截取模式, 此时便可按下鼠标左键拖动鼠标在屏幕上选择所需内容直到松开左键为止, 然后单击回车键 (Enter) 便可将所选内容以位图格式存盘, 若按下 Esc 键则退出截取模式。

### 2 程序简介

Windows 内部提供了丰富的应用程序接口 (API) 函数, 供专业开发人员用来制作应用程序。Visual Basic 作为一个开发工具具有直接调用 Windows 应用程

序接口函数的特性, 故本文选用 VB 编制该屏幕截取程序。主程序为:

程序 1:

```
Sub Main ()
    Do
        DoEvents
        Loop Until (GetKeyState(32) And &H8000) = &H8000
        Load Form1
        capture Form1
        Form1.Show
        Loop
End Sub
```

#### 2.1 检测热键

程序中设置了一个热键, 为简单起见, 采用了空格键, 用户可根据具体情况自己设定一个 Windows 中不常使用的键。为了确定该热键是否被按下, 程序则重复查询键盘状态, 为此调用了 Windows API 中的 GetKeyState(nVirtKey) 函数, 该函数检测到该指定的热键被按下后, 程序则跳离循环体, 且函数返回整数值 1 的最高位为 1。

一旦检测到空格键被按下, 程序便调用屏幕截取过程 Sub Capture (程序 2), 然后将屏幕内容以位图形式装入 Form1 并显示, 供用户选择截取用。

#### 2.2 屏幕截取过程

在程序 Sub Capture 中, 调用 Windows API 中的 CreateDC 函数, 用 "DISPLAY" 做为设备程序名, 这样创建了一个设备描述表 SourceDC, 其内容就是屏幕内容的位图形式。为了将位图拷贝到 Form1 中, 本程序使用了 Windows API 中的 BitBlt 函数。该函数把 SourceDC 中的位图以屏幕大小拷贝到 Form1 中, 所以 Form1 必须是最大化的, 且 AutoRedraw 要设置为真 (true)。

程序 2:

```
Sub capture (f As Form)
```

```

Dim ,SourceDC
Dim scrnw As Integer,scrnh As Integer
DoEvents
scrnw = CInt(Screen.Width\Screen.TwipsPerPixelX)'
将 Twips 模式转为
scrnh = CInt(Screen.Height\Screen.TwipsPerPixelY)'
Pixel 模式
SourceDC = CreateDC("DISPLAY", "0", "0", "0")
BitBlt f. hDC, 1, 1, scrnw, scrnh, SourceDC, 0, 0,
&HCC0020
DeleteDC SourceDC '删除设备描述表 SourceDC
End Sub

```

Form1 被显示后,就需要在其中按住鼠标左键,移动鼠标在所需内容的周边画上矩形边框。一旦松开鼠标左键,框便固定显示在屏上。

程序设计中将下列变量声明放入 Form1 中的 General Declare 段:

```

Dim x0 As Single,y0 As Single,x1 As Single,y1 As Single

```

其中(x0,y0)为鼠标左键按下时的坐标位置,即矩形框的左上角坐标,(x1,y1)为该矩形框的右下角坐标。放置程序 3 到 \_\_MouseDown 事件中去,以便设定起始坐标,并清除前一次画的边框。

程序 3:

```

Sub Form __MouseDown(button As Integer,Shift As Integer,x As Single,y As Single)
If button=1 Then
Form1.Line(x0,y0)-(x1,y1),,B
x0=x
y0=y
x1=x
y1=y
End If
End Sub

```

放置程序 4 到 \_\_MouseMove 事件中,当按住鼠标左键并移动时,需要不断的清除上一次画的边框,并再画出新的边框。所以同一矩形须画两次,且第一次可见,第二次被清除,则画图模式(DrawMode)须为 Invert(反转)。

程序 4:

```

Sub Form __MouseMove(button As Integer,Shift As Integer,x As Single,y As Single)
If button=1 Then
Form1.Line(x0,y0)-(x1,y1),,B
Form1.Line(x0,y0)-(x,y),,B
x1=x
y1=y
End If
End Sub

```

内容选定后,按下回车键(Enter)就将该矩形区域之位图内容拷贝到 Form2 的图片框 Picture 中,这样用 SavePicture 命令便能将 Form2. Picture. Image 以 BMP 形式存盘,该文件就是用户所需要的文件。此过程见程序 5,且置于 \_\_KeyDown 事件中。

程序 5:

```

Sub Form __KeyDown(keycode As Integer,Shift As Integer)
Dim xm As Integer,ym As Integer
Dim ax As Integer,ay As Integer
If keycode=27 Then
Unload Form1
ElseIf(keycode=13)And ((x0<>x1) Or (y0<>y1))Then
xm=CInt((x1-x0)\Screen.TwipsPerPixelX)-1
b_ym=CInt((y1-y0)\Screen.TwipsPerPixelY)-1
ax=CInt(x0\Screen.TwipsPerPixelX)+1
ay=CInt(y0\Screen.TwipsPerPixelY)+1
Load Form2
Form2. Picture. Width=x1-x0-1'调整 PictureBox 的大小
Form2. Picture. Height=y1-y0-1
BitBlt Form2. Picture. hDC,0,0,xm,ym,Form1. hDC,
ax,ay,&HC0020
SavePicture Form2. Picture. Image, InputBox("File Name")
Unload Form2
End If
End Sub

```

注意 Form2 及 Form2. Picture 的 AutoRedraw 均应设置为真。

### 3 结束语

以上程序已用 VB3.0 版本软件在 486 微机上调试通过,程序运行稳定。操作极为简单,既方便又节省了操作时间,为用户在 Windows 下截取屏幕时提供了得心应手的工具。

用户若因工作需要,还可将本程序作少许改动,将矩形框改为椭圆框,使所截画面更具有艺术风格。

附录:

其余程序代码如下:

1)将下列声明放置于本程序的总声明段内。

```

Declare Sub BitBlt Lib "GDI" (ByVal hDestDC As Integer,
ByVal X As Integer,ByVal Y As Integer,ByVal nWidth As Integer,
ByVal nHeight As Integer,ByVal hSrcDC As Integer,ByVal XSrc As Integer,
ByVal YSrc As Integer,ByVal dwRop As Long)
Declare Function CreateDC Lib "GDI" (ByVal lpDriverName As String,
ByVal lpDeviceName As String,ByVal lpOutput As String,ByVal lpInitData As String)As Integer
Declare Sub DeleteDC Lib "GDI" (ByVal hDC As Integer)
Declare Function GetKeyState Lib "User" (ByVal nVirtKey As Integer)As Integer

```

2)将下列代码放置在 Form1 中的相应位置处。

```

Sub Form __Load()
Form1.DrawMode=6'设置为反转模式
End Sub
Sub Form __Unload(Cancel As Integer)
x0=0
y0=0
x1=0
y1=0
End Sub

```

# 优化散列排序算法研究

邢为民

(西北大学计算中心 西安 710069)

宁正元

(福建林学院计算机科学系 南平 353001)

**摘 要** 本文介绍一种优化了的散列排序新算法。适当增加存储空间,子序列元素相对一次到位,有效地减少了地址冲突,大大地提高了排序效率。

**关键词** 散列排序 散列变换 冲突 装载率

## 1 引言

用建立待排序码与其目标序列的相应位置关系来排序的思想实质,是将待排序码的数据元素按其数据特征进行递增,递减或其它要求的顺序进行排序,待排序码的数据元素在目标序列中的位置应是该数据元素值的一个映射。基于这种思想,文[1]构造出了具体的散列排序法,排序效果较为理想,得到了美国佛吉尼亚 ODU 著名教授 KURTMALY 的赞赏,他认为:“这个方法是 BUCKET 排序和线性散列技术的巧妙结合。在搜索数据或建立一个排序数据时,开拓  $O(1)$  平均情况的散列复杂性。”在实际应用中,散列排序法远优于冒泡排序法和快速排序法。

然而,要成功地构造散列排序算法,关键是要解决好排序过程中数据元素的地址冲突问题。传统的做法是增大存储空间以减少冲突,这样做使装载率大为降低。文[1]提出了把冲突元素插入已成的子序列中的解决冲突思路,本文将给出略增加存储空间,子序列元素相对一次到位的解决冲突措施,实践证明可以减少冲突 58%。当冲突元素插入已成序列的子序列中时,比较移动的次数大幅度减少,排序效率明显提高。

## 2 排序算法

文[1]给出了这样的映射函数定义:给定映射函数  $J(A)$ ,它产生唯一的一组整数,均有  $J(A(i)) \neq J(A(l))$ 。其中  $i \neq l$ ,且当  $A(i) < A(l)$  有  $J(A(i)) < J(A(l))$ 。

根据  $J$  的定义,给定一个  $A(i)$  必须唯一产生一个  $J(A(i))$ ,于是  $N$  个元素的数据集合  $A$  通过函数  $J$  的变换可以产生一组  $J_i$ ,其中  $i=1,2,3,\dots,N$ ,使  $A(i)$  对应落位于一个递增或递减序列之中, $J$  是散列变换函数且是一一对应的映射关系,这是构造散列变换所希望的,也是无冲突事件发生的理想情况。

然而,在实际应用中冲突事件是不可避免的。对给定的排序码集合  $A$ ,会有如下两种情况存在:

①当  $A(i)=A(l)$ , $i \neq l$ ,有  $J(A(i))=J(A(l))$ ;

②当  $A(i) \neq A(l)$ ,有  $J(A(i))=J(A(l))$ 。

这就形成了非一一对应映射关系,称之为“冲突”事件。要提高排序速度,就得设法减少冲突事件的发生次数。而减少冲突事件的发生次数。关键在于构造良好的散列变换函数  $J$ 。

用  $A_{ik}^j$  ( $j=1,2,\dots,k_i$ ) 表示含有  $k_i$  个元素的第  $i$  个子集  $A_{ik}$  中的第  $j$  个元素,并设  $A_{ik}^{j1} \in A_{ik}$ , (若  $A_{ik}^j = A_{ik}^{j1}$ ,则认为它们是集合  $A_i$  中的两个元素)。若  $A_{ik}^j < A_{ik}^{j1}$ ,有  $J(A_{ik}^j) = J(A_{ik}^{j1})$ ; 若  $A_{ik}^j < A_{(i+1)k(i+1)}^{j1}$ ,有  $J(A_{ik}^j) < J(A_{(i+1)k(i+1)}^{j1})$ 。

于是对排序码集合  $A$  中所有元素的排序结果可以表示为:  $A_{1k_1}^1, A_{1k_1}^2, \dots, A_{1k_1}^{k_1}, A_{2k_2}^1, \dots, A_{2k_2}^{k_2}, \dots, A_{mk_m}^1, \dots, A_{mk_m}^{k_m}$

在第一个子集  $A_{1k_1}$  中,所有的散列函数值相等,在第二个子集  $A_{2k_2}$  中所有元素散列函数也相等,……,在第  $m$  个子集  $A_{mk_m}$  中所有元素的散列函数值也相等。由映射函数  $J$  的定义知,当  $A_{1k_1}^1 < A_{2k_2}^1$  有  $J(A_{1k_1}^1) < J(A_{2k_2}^1)$ ,所以  $A_{2k_2}$  集合中的所有元素的排列位置都处于集合  $A_{1k_1}$  中所有元素的右侧, $A_{3k_3}$  集合中的所有元素的排列位置都处于集合  $A_{2k_2}$  中所有元素的右侧,……, $A_{mk_m}$  集合中的所有元素的排列位置都处于集合  $A_{(m-1)k(m-1)}$  中所有元素的右侧。 $A_{1k_1}^1$  元素排列在序列的第 1 个位置上, $A_{2k_2}^1$  元素排列在序列的第  $k_1+1$  个位置上, $A_{3k_3}^1$  元素排列在序列的第  $k_1+k_2+1$  个位置上,……, $A_{mk_m}^1$  元素排列在序列的第  $\sum_{i=1}^{m-1} k_i+1$  个位置上。

由于每个子集中元素通过映射函数映射到目标序列中的位置区段是确定的,所以冲突事件只可能发生在子集内部的诸元素之间,发生地址冲突需要移动元素的最大次数为该子集元素个数减一。即  $A_{ik}$  中发生冲突时的最大移动次数为  $k_i-1$ 。

基于以上分析,可给出如下的优化散列排序算法:

1) 将所有的存储单元  $A(i)$  ( $i=1,\dots,n$ ,  $n+1,\dots,2n$ ) 置初始态;

2) 将统计子集元素个数单元  $B(i)$  ( $i=1,2,\dots,n$ ) 初始化为 0;

3) 将待排序码数据存于  $A(i) (i=n+1, \dots, 2n)$ ;  
 4) 求数据集的最大值  $\max A$  和最小值  $\min A$ ;  
 5) 统计每个子集元素的个数,  $B(J(A(i))) = B(J(A(i))) + 1, i-1$

6) 确定每个子集元素的上界(或下界)在序列中的位置,  $B(i) = \sum_{l=0}^{i-1} B(l) + 1$

7)  $L$  由  $N+1$  到  $2N$  步长为 1 执行下面的循环:

(1) 逐个计算  $J1 = \text{INT}(1A(L) / (\max A - \min A) / n1)$ ; (2)  $J = B(J1)$ ; (3) 若  $A(J)$  未曾被赋值, 则  $A(J) = A(L)$ , 并继续下一个循环; (4) 若  $A(J) > A(L)$ , 则  $C = A(J)$ ,  $A(J) = A(L)$ ,  $A(L) = C$ ,  $J = J + 1$ , 转(3); (5) 若  $A(J) < A(L)$ , 则  $J = J + 1$ , 转(3)。

### 3 算法分析

若满足散列函数  $J$  的定义,  $A(i) > A(k)$  且  $J(A(i)) > J(A(k))$ , 按上述算法产生目标序列的过程中,  $A(i)$  与  $A(k)$  均置于所指定的位置而无冲突事件发生。若产生冲突, 即当  $J(A(i)) = J(A(k))$  时, 经  $J = B(J1)$  从第  $J$  个位置开始向右有限个元素(最多为所有  $J(A(i)) = J(A(k))$  的那些  $A(i)$ ,  $A(k)$  所形成的子集中的元素个数减 1) 比较, 按符合递增要求将  $A(i)$  插入到目标序列中去。

我们对随机产生的 100 组(每组  $N$  个)数据元素分别运用优化散列排序算法和一般散列排序算法进行排序试验, 试验结果(见表 1)表明, 优化散列排序算法的排序效率是很理想的, 并且随排序码个数  $N$  的趋大, 排序效率更趋显著。

表 1: 优化散列排序耗时对比分析

排序码样本数 $N$	一般散列所需时间(")	优化散列所需时间(")
100	0.016	0.00505
200	0.07074	0.01533
500	0.19169	0.02560
1000	0.598	0.05108
2000	1.9383	0.093
3000	3.7733	0.13924
4000	7.19	0.184
8000	29.1	0.72

应用优化散列法和一般散列法排序, 在排序过程中会出现冲突或再冲突事件。我们通过对 100 组不同

数据排序过程中的冲突和再冲突次数的测试统计。优化散列排序中冲突次数最高的为 50.32%, 最低的为 47.5%。平均产生冲突率(冲突率为冲突次数与待排序数据元素个数  $N$  之比)为 49.43%(见表 2)。

表 2: 优化散列排序算法冲突率测试分析

待排序数据个数	冲突次数	冲突率(%)
100	50	50
200	95	47.5
300	148	49.3
400	197	49.25
500	250	50
1000	497	49.7
5000	2516	50.32
8000	3959	49.49

在算法构造中, 适当增加存储单元是降低冲突率的重要措施之一。由文[2]知, 装载率降低, 产生冲突次数将会减少。由文[2]的定义分析上面给出的算法, 其装载率  $= N / (3 * N) = 30\%$ , 因此理论计算冲突量应为  $1.2N$ , 而优化散列排序算法实测结果为  $0.5N$ 。比原理值降低了 58%。

综合以上分析, 优化排序算法最好的情况是冲突次数为零, 一般情况下在排序过程中产生  $0.5N$  次冲突。在一般情况下, 排序过程中需做  $N$  次求子集元素个数,  $N$  次求  $J$ ,  $N$  次判定  $A(j)$  是否曾被赋值,  $0.5N$  次处理冲突的判别, 故该算法的时间复杂度为  $T(n) = O(n)$ ; 其空间复杂度为  $S(n) = O(n)$ 。

### 4 结语

优化散列排序是一种排序效率极高的排序方法, 其时间复杂度是问题规模的线性函数, 值得在实际应用中大力推广使用。

#### 参考文献

- 1 张正铤等. 散列排序算法. 广西科学院学报, 1982, VOL. 1, NO. 1;
- 2 曹东启等译. 数字计算机的编译程序构造. 科学出版社, 1976;
- 3 宁正元. 栈式排序的几个算法. 计算机应用研究, 1995, NO. 5.

(上接 75 页)

```
do key_up
case 向下键
do key_down
case 向下滚页键
do key_pgdn
case 向上滚页键
do key_pgup
case 增加
do zj_cz
case 插入
do cr_cz
```

```
case 删除
do su_cz
case 修改
do xg_cz
...
case lk=27 &&ESC 退出
exit
endc
endd
```

⑤ 向外界提供单一的控制接口的函数

### 4 结束语

本人深深地感到, 在开发应用程序时先开发一个多记录全屏幕操作的“屏幕程序生成器”是十分必要的, 因为它能带给您一劳永逸的效果。

#### 参考文献

- 1 章立民. FoxPro 2.5 for Dos 程序设计提高篇. 人民邮电出版社, 1994, 7

# 同屏显示多幅彩色图像技术新探

陆 遥

(广西师范大学数学与计算机科学系 桂林 541004)

**摘 要** 本文以能提供真彩色显示模式的各种系列的 SVGA 显示卡为硬件基础,结合适于 24 位真彩色图像显示的 TIFF 图像文件格式,提出了一种能彻底解决同屏显示多幅彩色图像的技术方案,并给出了实现该方案的汇编语言源程序。

**关键词** 彩色图像 图像处理 真彩色显示模式 VESA 标准 TIFF 图像格式

同屏显示多幅彩色图像是图像处理领域令人感兴趣的问题。在 VGA 主导微机显示系统时,由于只能同屏显示 256 种颜色,不同的彩色图像均有自己的 256 色颜色集,且互不相容,故同屏显示多幅彩色图像时,其难点体现在:如何构造这样一个 256 色颜色集,使得它能尽量满足每幅图像的显示要求。关于这方面工作的研究成果,已见诸于一些专业刊物。但有一点可以断言,无论采用怎样好的算法去构造这个颜色集,抛开算法的时间开销不论,其最终的显示效果也是难以尽如人意的。当然,造成这个问题无法完美解决的主要原因,是由于显示系统硬件的不完善。

本文在能提供真彩色显示功能的 SVGA 显示系统的硬件基础上,提出一种能彻底解决上述问题的技术方案,供有兴趣的读者参考。

## 1 硬件基础

各种 SVGA 显示卡均能提供一些真彩色图形显示模式。在这些模式下,同屏显示颜色数达到 16M,每幅彩色图像所需的颜色均已包含在这 16M 种颜色中,无需各自配备自己的颜色集,也不存在颜色集不相容的问题。此外,真彩色显示模式下,图像的显示直接通过图像像点的像素值在 16M 颜色中进行选择,所以每个像素的长度为 24 位,其中红、绿、蓝三种成分(简称 RGB 分量)各占 8 位。由于各种系列的 SVGA 显示卡所设定的显示模式号和内部寄存器端口不尽一致,若针对某种卡编程将失去通用性,好在各种 SVGA 卡均支持 VESA 标准,所以编程时可采用 VESA 标准显示模式号和标准扩展 BIOS 功能调用以提高软件兼容性。

## 2 实现方案

1) 利用图像截取程序将图像截下,这样可免于对各种格式图像文件的解压。设这样截获的图像文件格式如表 1(详见参考文献[1])。

2) 将截获的图像文件转换为一种适合于真彩色显示模式的图像文件格式。本方案采用的是无压缩的

表 1

文件标志:0a0dh
显示模式(1 字节)
256 组颜色寄存器数据(共 768 字节)
图像像素数据(长度随显示模式变化)

TIFF 图像格式。并于 TIFF 格式的详细内容请见参考文献[2],在此仅将其中的重要部分作一简介。

(1) 关于 IFD——图像文件目录。一般只有一个 IFD,且其中一般包含 15 个标记指针。这些标记指针中,大部分可以是不变的,对本文所要求的图像显示意义也不大,但下述四个标记指针非常重要:①标记码 100H:图像宽度(每行像素数);②标记码 101H:图像长度(扫描行数);③标记码 116H:每条(strip)包含的扫描行数。一幅图像一般只做成一条,故该项的值一般等于图像长度值;④标记码 117H:条中字节计数。因每像素占 3 字节,故该项的值为像素数的 3 倍。

(2) 关于像素数据。像素数据一般从文件内偏移 0C8H 处开始存放,每像素占 3 字节,按 RGBRGB……的顺序存放。

3) 以 VESA 标准显标模式 112H(640×480×16M)对多幅 TIFF 格式彩色图像进行同屏显示。

## 3 技术关键

1) 将截获的彩色图像文件转换为 TIFF 格式图像文件。

关键是构造 IFD 和生成 TIFF 像素数据。构造 IFD 时需根据原图像的显示模式,确定图像的宽度和长度,并计算出每条的扫描行数和条中字节数。生成像素数据时,首先根据原图像的 8 位像素值,到 256 组颜色数据中找到对应的一组 3 字节 RGB 数据,然后将原 RGB 分量(均为 6 位)各左移两位,即得到相应的一个 TIFF 像素(24 位)。

2) TIFF 格式彩色图像的显示。

①调用 VESA BIOS 扩展功能完成多项重要操作。其中:调用 4F02H 号功能调置 112H 号显示模式;调用 4F01H 号功能返回每线字节数、窗口 A 起始段及粒度等重要参数;调用 4F05H 号功能实现内存页切换。(详见参考文献[3])

②VESA 扩展显示模式 112H 下的 VRAM 组织方式为线性方式,但在写像素时,三种颜色分量需按



文后所附源程序 1 用于将截获的 256 色彩图像文件转换为 TIFF 格式图像文件;源程序 2 用于同屏显示两幅非压缩 TIFF 真彩色图像。

；源程序 1-TIFGEN  
；用法：TIFGEN〈截获图像文件〉〈TIFF 文件〉

```

        je err1
        jmp f11
f11end:mov byte ptr es:[di],0
f102:lodsb
        cmp al,20h
        je f102
        cmp al,0dh
        je err1
        lea di,es,tiffile
f12:stosb
        lodsb
        cmp al,0dh
        jne f12
f12__end:mov byte ptr es:[di],0
        push es
        pop ds
open__file:
        lea dx,imgfile
        mov ax,3d00h
        int 21h
        jc err1
        mov i__handle,ax
        lea dx,tiffile
        mov cx,00h
        mov ah,3ch
        int 21h
        jc err1
        mov t__handle,ax
        mov ah,3fh
        mov bx,i__handle
        mov cx,303h
        lea dx,img__id
        int 21h
        jc err2
        mov ax,word ptr img__id
        cmp ax,0d0a1
        jne err3
        call make__ifd
        call img__to__tif
close__file:
        mov ah,3eh
        mov bx,i__handle
        int 21h
        mov ah,3eh
        mov bx,t__handle
        int 21h
        jmp done
err1:push es
        pop ds
        mov ah,40h
        mov bx,01h
        lea dx,err__msg1
        mov cx,11h
        int 21h
        jmp done
err3:mov ah,40h
        mov bx,01h
        lea dx,err__msg3
        mov cx,14h
        int 21h
        jmp close__file
err2:mov ah,40h
        mov bx,01h
        mov cx,14h
        lea dx,err__msg2
        int 21h
        mov ah,42h
        mov bx,t__handle
        mov al,00h

```

```

xor cx,cx
mov dx,cx
int 21h
mov ah,40h
int 21h
jmp close __file
done:mov ah,4ch
int 21h
make __ifd proc near
lea di,tif __ifd
mov al,orgmode
cmp al,13h
je v13
cmp al,5ch
je v5c
cmp al,5dh
jne err3
v5d:lea si,v5dh __info
jmp modi __ifd
v5c:lea si,v5ch __info
jmp modi __ifd
v13:lea si,v13h __info
modi __ifd,
add di,leh
movsw
add di,0ah
movsw
add di,46h
movsw
add di,0ah
movsw
movsw
add di,08h
movsw
movsw
add di,08h
movsw
movsw
ret
make __ifd endp
img __to __tif proc near
mov ah,40h
mov bx,t __handle
mov cx,0c8h
lea dx,tif __ifd
int 21h
jc err2
lea si,tif __ifd+7eh
mov ax,[si]
mov dx,[si+2]
mov bx,300h
div bx
cmp dx,0
je RGBStripGen
inc ax
jmp RGBStripGen
err2:jmp err2
RGBStripGen:
r,push ax
lea di,tif __buff
mov bx,t __handle
mov cx,100h
lea dx,img __buff
mov ah,3fh
int 21h
jc err21
mov si,dx
mov cx,ax
push ax

```

```

    lea bx,imgcolor
convert,xor ax,ax
    lodsb
    mov di,3h
    mul di
    push bx
    add bx,ax
    mov ax,[bx]
    shl ax,1
    shl ax,1
    mov [di],ax
    mov al,[bx+2]
    shl al,1
    shl al,1
    mov [di+2],al
    add di,3h
    pop bx
    loop convert
W, pop ax
    mov bx,3h
    mul bx
    mov cx,ax
    mov bx,t__handle
    lea dx,tif__buff
    mov ah,40h
    int 21h
    jc err21
    pop ax
    dec ax
    jne r
w__Res,mov ah,40h
    mov bx,t__handle
    mov cx,10h
    lea dx,X__Y__Res
    int 21h
    jc err21
w__end,ret
img__to__tif endp
code ends
end start
;源程序 2—TIFSHOW
;用法,TIFSHOW(TIFF 文件 1)<TIFF 文件
2>
stack segment stack
dw 32 dup(?)
stack ends
data segment
tiff1 db 30 dup(0)
tiff2 db 30 dup(0)
handle1 dw ?
handle2 dw ?
handle dw ?
open__fs db 00h
tif__head db 8 dup(?)
tif__ifd db 192 dup(?)
orgvmode dw ?
tif__id db 49h,49h,2ah,0,8h,0,0,0
v112h__inf db 256 dup(?)
tif__buff db 3000 dup(?)
winAplace dw ?,?
err__msg1 db 'File not found.',0dh,0ah
err__msg2 db 'File access error.',0dh,0ah
err__msg3 db 'VESA not support.',0dh,0ah
err__msg4 db 'Unknown file type.',0dh,0ah
data ends
code segment
assume cs,code,ds,data,ss,stack
start,mov ax,data
    mov es,ax
    mov si,81h
    lea di,es:tiff1
    lea bx,es:tiff2
f101,lodsb
    cmp al,20h
    je f101
    cmp al,0dh
    je err1
f11,mov es:[bx],al
    inc bx
    stosb
    lodsb
    cmp al,0dh
    je f11__end
    cmp al,20h
    jne f11
f11__end,mov bl,al
    mov al,00h
    mov es:[bx],al
    stosb
    cmp bl,0dh
    je f1__end
    lea di,es:tiff2
f102,lodsb
    cmp al,20h
    je f102
    cmp al,0dh
    je f1__end
f12,stosb
    lodsb
    cmp al,0dh
    je f12__end
    cmp al,20h
    jne f12
f12__end,mov al,00h
    stosb
f1__end,push es
    pop ds
    jmp save__curvmode
err1,push es
    pop ds
    mov ah,40h
    mov bx,01h
    lea dx,err__msg1
    mov cx,11h
    int 21h
    cmp open__fs,00h
    jne close__file
    jmp done
save__curvmode,
    mov ax,4f03h
    int 10h
    cmp ax,4fh
    jne err3
    mov orgvmode,bx
;set vedio mode 112h
    mov ax,4f02h
    mov bx,112h
    int 10h
    cmp ax,4fh
    jne err3
    lea dx,tiff1
    lea bx,handle1
    call open__file
    mov winAplace,00h
    mov winAplace+2,00h
    call show__image
    lea dx,tiff2
    lea bx,handle2
    call open__file
    mov winAplace,04h
    mov winAplace+2,le0h
    call show__image
close__file,
    mov h,3eh
    mov bx,handle1
    int 21h
    dec open__fs
    je done
    mov ah,3eh
    mov bx,handle2
    int 21h
    jmp done
err4,mov ah,40h
    mov bx,01h
    lea dx,err__msg4
    mov cx,14h
    int 21h
    jmp close__file
err2,mov ah,40h
    mov bx,01h
    mov cx,18h
    lea dx,err__msg2
    int 21h
    jmp close__file
err3,mov ah,40h
    mov bx,01h
    lea dx,err__msg3
    mov cx,13h
    int 21h
    jmp close__file
done,mov ah,01h
    int 21h
    mov ax,4f02h
    mov bx,orgvmode
    int 10h
    mov ah,4ch
    int 21h
err11,jmp err1
open__file proc near
    mov ax,3d00h
    int 21h
    jc err11
    inc open__fs
    mov [bx],ax
    mov handle,ax
    mov bx,ax
    mov ah,3fh
    mov cx,0c8h
    lea dx,tif__head
    int 21h
    jc err2
check__file__format,
    lea si,tif__head
    lea di,tif__id
    mov cx,8h
    repe cmpsb
    cmp cx,00h
    jne err4
    ret
open__file endp
show__image proc near
get__v112h__info,
    mov ax,4f01h
    mov cx,112h
    lea di,v112h__inf
    int 10h
    cmp ax,4fh
    jne err3
    mov al,v112h__inf+2
    cmp al,07h
    jne err3
get__winAseg,
    mov ax,word ptr v112h__inf+8
    mov es,ax
    push winAplace+2
    mov ax,word ptr tif__ifd+16h
    mov bx,3h
    mul bx
    push ax;每线字节数入栈
    mov bx,word ptr v112h__inf+10h

```

```

sub bx,winAplace+2
cmp ax,bx
jbe line__size
mov ax,bx
line__size:
push ax;每线实际使用字节数入栈
mov ax,word ptr v112h__inf+4
mov bx,400h
mul bx
mov bx,word ptr v112h__inf+10h
div bx
push dx
push ax
mov bp,sp
set__winA:
mov ax,4f05h
mov bx,00h
mov dx,winAplace
int 10h
cmp al,4fh
jne err31
inc winAplace
mov dx,[bp]
xor di,di
add di,[bp+8]
write__image:
call read__image
lea si,tif__buff
mov cx,[bp+4]
rep movsb
dec word ptr tif__ifd+22h
je w__end
add di,word ptr v112h__inf+10h
sub di,[bp+4]
dec dx
jne write__image
mov cx,[bp+2]
cmp cx,00h
je set__winA
mov ax,word ptr v112h__inf+10h
sub ax,cx
mov[bp+8],ax
mov ax,word ptr v112h__inf+4
mov bx,400h
mul bx
sub ax,[bp+8]
sbb dx,0
mov bx,word ptr v112h__inf+10h
div bx
mov [bp],ax
mov[bp+2],dx
call read__image
cmp cx,[bp+4]
jae write__a__line
jmp write__part
err21: jmp err2
err31: jmp err3
write__part:
push cx
lea si,tif__buff
rep movsb
mov ax,4f05h
mov bx,00h
mov dx,winAplace
int 10h
cmp al,4fh
jne err31
xor di,di
mov cx,[bp+4]
pop ax
sub cx,ax
rep movsb
dec word ptr tif__ifd+22h
je write__end
jmp set__winA
w__end: jmp write__end
write__a__line:
lea si,tif__buff
rep movsb
dec word ptr tif__ifd+22h
je write__end
jmp set__winA
write__end:
push ds
pop es
pop ax
pop ax
pop ax
pop ax
ret
show__image endp
read__image proc near
push ax
push bx
push cx
push dx
mov ah,3fh
mov bx,handle
mov cx,[bp+6]
lea dx,tif__buff
int 21h
jc err22
mov cx,word ptr tif__ifd+16h
mov bx,dx
data__p,mov dl,[bx]
xchg dl,[bx+2]
mov [bx],dl
add bx,3h
loop data__p
jmp read__end
err22: jmp err2
read__end:
pop dx
pop cx
pop bx
pop ax
ret
read__image endp
code ends
end start

```

## 参考文献

- 1 易明. VGA 图形拷贝程序. 计算机世界(周刊), 1993 年 6 月 23 日; 第 133 版
- 2 侯阳. 微机图形文件格式集粹. 北京: 学苑出版社, 1993. 12
- 3 施威铭研究室. SVGA 显示原理和绘图技巧. 北京: 学苑出版社, 1995. 11

## 多任务生产计算机动态优化调度方法 MP—PERT 之探讨

王首安 杜玉桥 杜习英

(合肥工业大学 619 信箱 230009)

**摘 要** PERT(Program Evaluation and Review Technique)技术在工程计划和企业生产管理中得到日益广泛的重视和应用,但它主要是针对单项工程任务提出的。本文提出一种 MP—PERT(Multi-Projects PERT)技术和以加工完成时间为主约束的优化排序方法,设计了实现所需的系统结构,以期解决多任务加工企业的生产计划编制与作业计划实施监督系统的计算机问题。

作者深信,在本方法基础上完全可以构造出一个中小企业生产周期管理软件的开发环境,其作用将远远超过 Multiplan 和 Locut 系列等开发工具。

## 1 引言

自从泰勒开创了动作行为研究学科以来,企业管理界就兴起了关于动作时间的研究和工时定额等一系

列研究热潮,并推动管理进入了量化时代。在此背景下,从五十年开始先后提出了 CPM(Critical-Path Method)和 PERT(Program Evaluation and Review Technique)等新技术,以指导解决量化基础上的工

程计划制定与实施的问题。

CPM 和 PERT 都要求用网络图的形式来表示工程作业中各步工序间的先后衔接关系及每一工序的时间等参数值。但 CPM 仅提供如何寻找关键路线的方法,而 PERT 则不仅要找出关键路线,而且更重要的是解决在时间,成本,资源等约束条件下关键工序、人、物、时的保证和赶超计划的综合优化问题。因而,PERT 技术在需要统筹安排的大中型工程计划编制中得到了广泛的应用,并分别派生了成本估算法和物资分配法等方法。

但是,PERT 技术并不能直接指导解决多项任务的加工计划安排与实施控制问题,这是因为其方法本身有两大缺陷。首先,PERT 网络图上每一分支所标的时间值往往只包括估计的加工时间和传递时间,而没有考虑到在加工过程中因受其它任务牵挂所必须花费的排队等待时间,并且由于所在分支上各任务加工次序难以事先固定下来,因而这一等待时间也就难以静态确定,所以“关键路线”概念在此就不适用了。其实,在计划安排过程中,无法就单个分支所表示的工序为某一任务孤立地做出优化安排。因每一工序上的加工能力总是有限的,为此一任务作出的优化安排可能会冲击另一些更急迫任务的安排从而损失全局的优化。为了有效地解决这些问题,并探讨出一套系统的方法,本文作者在分析研究了几个行业若干中小企业的生产布局与加工过程的基础上,通过本文提出了多任务企业加工过程的一般模型,并用“网络图标”概念来取代“网络图”概念,用“关键工序”概念来取代“关键路线”概念,从而进一步提出 MP—PERT (MUL tiProject PERT) 技术,介绍了基于 MP—PERT 的动态模拟优化排序方法。本文所提出的尚属一些初步设想,其效用将在后续计算机生产周期管理系统的开发中得到验证,同时欢迎广大同仁批评指正。

## 2 多任务流企业的加工模型

工业企业的生产部门通常是由车间组成的,而车间又可划分成若干加工工序组,每一生产任务常常由工序组协调完成。工序组是工厂最小的组织实体,工序组之间只有并行或串行关系,组与组之间所完成的类型是不同的。生产任务是指外界输入给企业的作业请求,企业必须在指定日期前输出产品或劳务。企业的职责就在于分析外界请求,指定任务的工艺路线,估算各工序的工时定额,综合编制生产计划并监督实施,以保证按质按量如期完成。

“多任务流”指的是一种加工类型,其任务来源广,到达时间随机,各任务所走的工艺及工时定额可能有较大差异,此外各任务对完工时间的要求又互不相同。对于这种生产类型,一方面因随时可能有急件要求加工而打破原计划安排,另一方面又无法预测后续到达的任务及其对加工工序组的要求有何侧重,所以难以

制定长远规划,更无法事先编制具体的年、季、月度生产计划。对于多任务流企业,其任务的到达和完成状态见图 1 所示。其中  $N \leq M$ 。I1, I2, ..., In 是 1...m 的不完全排列,以反映出任务的各种完工次序。

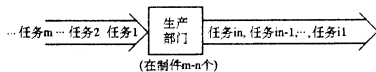


图 1 多任务流状态图

从生产服务角度来看,企业生产部门可以视为工序组的有序结构。对于一个企业,其可提供的服务或加工类型总是有限的。每一服务必定是以某种次序流经有限工序。将所有工序及其之间可能的流经路线综合起来,构成一个总的工序结构关系图,就形成了企业的加工模型。多任务流企业的加工模型见图 2 所示。

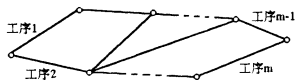


图 2 多任务流加工模型

## 3 PERT 林与 MP—PERT 介绍

在加工模型基础上,可将企业生产部门看成由 M 个逻辑工序组成。每一任务的到达,首先估算其在 1—M 个工序上的加工与传递时间  $T_{ij}$  (对于任务 i,  $1 \leq j \leq m, T_{ij} \geq 0$ ),并将  $T_{ij}$  标注在与加工模型同构的网络图上,就形成了该任务的 PERT 时序图。

对于所有刚到达的或正在加工中的任务,均可用其 PERT 时序图来反映它们在每一工序上的额定或剩余工作时数。所有这些网络图形成了一个 PERT 林,见图 3 所示。PERT 林中各工序对应的分支间用虚线连起来,以表示其相关性。

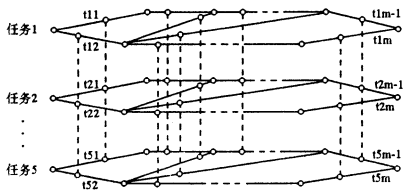


图 3 PERT 林示意图

MP—PERT 技术的作用,就是要在图 4 所示的三维空间上,针对每一工序安排各任务的加工次序,这些次序在各工序上一般是不相同的。其目的就是提高每一任务在工序间的配套性以及所有任务的按期完工

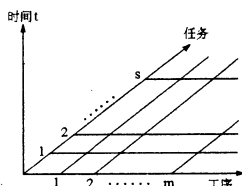


图 4 三维决策空间

任务 工序工时	任务 1	任务 2	任务 3
P1	1	1	3
P2	2	3	2

表 1

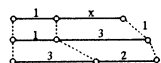


图 5

率,同时争取提高工序生产任务安排的均衡性和连续性。

以二个工序,三个任务为例,其工时定额如表1所示,PERT林见图5。按照MP—PERT方法在图4所示三维空间上安排的结果列于图6。据此不仅可以得出各工序上任务加工时间安排的甘特图,如图7;而且还可以进一步得出所有任务的计划进度表,示于图8。

MP—PERT方法的应用,首先要求对具体企业作大量的生产方式、加工工艺路线及工序生产能力的调查统计工作,形成定量定额标定,然后才能进入以下五个步骤:

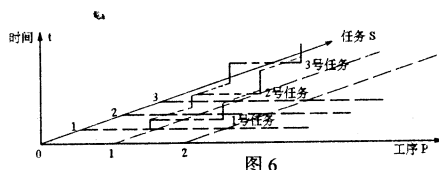


图6

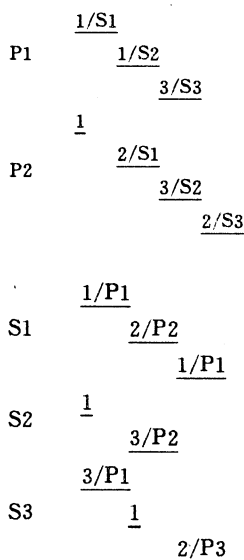


图7

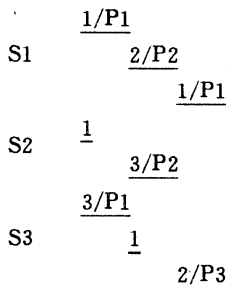


图8

(1)新任务的到达,首先落实工艺路线,并计算其工时定额。

(2)对单项任务,根据指定的交货日期从后向前计算各道工序上的最迟完工时间LFT(LAST FINISH TIME)和最迟必须开工时间LST(LAST START TIME),并将其头道工序的最早开工时间EST(EARLY START TIME)置成任务到达时间。

(3)统筹分析所有新到达的和已在加工的任务:从加工模型中最前工序开始,依次向后每次选一个工序 $I(I=1..M)$ 按EST从前到后和LST从小到大的次序将任务在工序I上排序,得出一组加工序列 $I_1, I_2, \dots, I_n$ ,然后分别计算各任务的实际可能完工期RFT(REAL FINISH TIME) IF  $RFT_{ik} \leq LFT_{ik}$  ( $R=1..N$ )则将 $IK$ 任务工序 $I$ 之后续工序的EST置为 $RFT_{ik}$  否则工序 $i$ 很可能会成为关键工序,这时需对

任务 $ik$ 的工时定额表进行分析,寻找出能进行并发压缩的工序 $j$ ,然后再从可压缩的工序 $j$ 开始重新向后进行动态排序。至于某一关键工序上任务饱满和无法压缩的则可通过安排加班甚至修改交货期来完成。

(4)将各工序上已排好序的任务序列列表输出,作为指挥生产加工次序的优选方案供决策参考。

(5)将各车间的日报表各实际生产进度表输入系统,在PERT林中更新各任务在对应工序上的(待)加工时间,为下次排序做好准备。

#### 4 电子计算机生产周期管理系统的数据库结构和MP—PERT的算法实现

计算机生产周期管理系统在企业中所处的地位可由下图说明。它起着指导,记录生产进行情况和动态修改、生成作业计划的作用。整个系统的工作流程见图10。

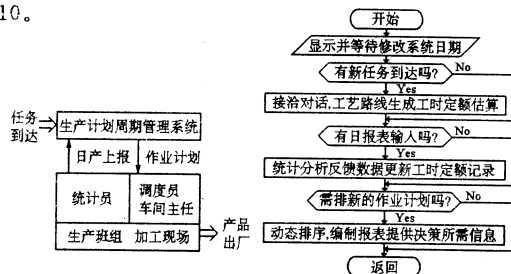


图9 管理结构层次图

图10 系统流程

为了在计算机中方便地表示PERT林及MP—PERT算法,必须设计出一套相应的文件和数据结构。为此,我们将管理系统建立在五类文件和六个算法的基础上。

##### 文件1、生产能力库

车间名称	工序名称	类别	小时定额	定额单位
------	------	----	------	------

##### 文件2、扩展的加工模型

工序号	工序组有效日工时	允许并发数	前序工序号	后继工序表
-----	----------	-------	-------	-------

##### 文件3、业务到达记录文件

业务号	业务有关情况
-----	--------

##### 文件4、工时定额文件

任务号	工序1	工序2	...	工序M	交货时间
-----	-----	-----	-----	-----	------

##### 算法1、ESTIMATE(任务号)

根据文件3和文件1,估算新到达任务的工序加工时间,将结果存入文件4。

##### 文件5、对每一工序,产生一个库文件结构:

任务号	加工持续	并发数	最早可	最迟开工	最迟完工	实际完工
i	时间	CUN	开工时间	时间	时间	时间
	LAST	( $\rangle=1$ )	EST	LST	LFT	RFT

##### 算法2、EVALUATE(任务号)

(根据文件4的内容,对单个任务估算各道工序上的最迟必须完成时间LFT和最迟必须开工时间LST)

## EVALUATE(i)

从文件 4 中读入加工时间非零的 K 个工序时间  
置成 P——LAST(工序 il, til)

.....

P——Last(工序 ik, tik)(工序号及其加工时间关系)

读入交货时间 T

置成 P——LFT(工序 ik, T)(最迟必须完工时间记录)

P——LST(工序 ik, T——tik)(最迟必须开工时间记录)

调用 P——SET(K) (从工序 ik 开始计算)

RETURN

P——SET(J);

if j=0 then Exit

elst if 工序 ij 的后续工序集合为 X  
and X 中各工序的 LST 之最小值为 Y  
and 置工序 ij 的 LFT 为 Y  
and 工序 ij 加工时间设为 Z  
and 置工序 ij 的 LST 为 Y-Z  
and 调用 P——SET(J-1).

endif

算法 3、P——SORT(J) J=1, 2, ..., M

(顺序对工序 1, ..., 工序 M 进行排序)

P——SORT(J)

STEP1: CREATE REPORT——FILE(J)(对工序 J 创立一个报告文件)

SIMU——DATE(——CUR——DATE(用系统当前报告的时间来初始化排序动态模拟时钟)

STEP2: 在第 5 类文件工序 J 的各记录中,

找满足  $EST \leq SIMU - DATE$  的任务号集合置入 X,

if X 为空集, 则  $SIMU - DATE(——$  所有记录中 EST 最小者, 重新进入 STEP2

ELSE 从 X 中找出 LST 最小者, 其任务号记为 Y,

ENDIF

STEP3: 计算  $RFTy < --- SIMU - DATE + LASTy$

if  $RFTy > LFTy$

then 调用算法 5 作特殊处理。

else  $simu - date < --- simu - date + 并发数 * LASTy / 工序组的日有效工时$

ENDIF

将 Y, LASTY 并发数 Y, ESTY, LSTY 置入 REPORT—FILE(J) 并将其从老文件中注销(重排序时可以收回)

STEP4: 转 2、直至所有记录的排序完毕。

STEP5: 对工序 J 中所有参加的任务, 计算其后续工序的 EST 以便为以后的排序打下基础。

算法 4、DAY—FEEDBACK(J) (J=1, 2, ..., M)

(对每一工序进行日进度汇报, 以便系统动态掌握生产情况)

DO WHILE 报表未输入完

INPUT 任务号, 完成百分比 K。

从文件 4 中取出任务 I, 工序 J 上的时间 TIJ 更新文件 5:

$NEW - LAST = TI * J(1 - K) * 保险率 / 并发数$

IF K=1

THEN 说明该任务在工序 I 上已完成, 从文件 5 中删去该行, 并将后继工序的 EST 置为 CUR—DATE。

ENDIF

IF 有返工事故, 则转算法 6。

ENDDO

END——ALGORITHM——4

算法 5(并发处理)

DO CASE

CASE——1, 从当前工序 J 向前继工序回溯, 找到可压缩的工序, 则重新进行排序。

CASE——2, 从当前工序 J 开始, 向后继工序找压缩点, 排序仍然按原次序进行, 排到最后一步工序再看结果是否满意。

CASE——3, 对任务进行全局分析, 看其所有工序的工时定额中是否有最可压缩的点, 进行压缩后再重新排序。

CASE——4, 安排加班, 突击完成关键工序上的任务。

ENDCASE

算法 6, REACTION

REPEAT

INPUT 任务号, 工序号, 返工估算工时

REPLACE 文件 4 中对应工序工时定额值

CALL EVALUATE(任务号)

REPLACE 文件 4 中对应工序工时定额值

CALL EVALUATE(任务号)

UNTIL 返工情况处理完毕

END——REACTION

本系统所介绍的文件结构, 其空间耗用量为  $O(M \cdot N)$ , 算法所需比较次数为  $O(M \cdot N^3)$ , 乘除运算次数为  $O(M \cdot N)$ ; 其中 N 表示在制件数, M 表示工序组数目。以手头 600 件, 80 项工序为例, 空间耗用量估为 4M 字节, 比较次数为  $8640 \cdot 10^6$  次, 乘除次数约为  $1 \cdot 10^5$  次。 (下转 22 页)



# 固态 ROM 盘初始固化文件的生成

马 骏 孔振宇 赵万生

(哈尔滨工业大学机电工程学院 150001)

赵德开 初佃辉

(哈尔滨工业大学威海分校计算机教研室 150001)

**摘 要** 本文通过对软磁盘物理及逻辑结构的分析,提出了一种磁盘数据分割的方法。利用这种方法可以实现磁盘数据的基于逻辑扇区的任意组合,可应用于固态盘系统的二进制初始固化文件的生成。

**关键词** 软盘 分割 固态盘 扇区 磁道

## 0 引言

固态盘作为一种高可靠性、高速度的数据存取装置,越来越被广大的用户所接受。广泛地应用于 PC 总线及 STD 总线的工业控制机及其它环境恶劣的控制系统中。常用固态盘一般由两部分组成,ROM 盘和 RAM 盘。ROM 盘的存储介质一般采用的是 EPROM 半导体存储芯片,由于通常 EPROM 芯片本身不具有单字节擦除的特性,所以对 EPROM 芯片操作时必须整片进行。ROM 盘中的存储单元一般是由一片或几片 EPROM 芯片组成。采用 DOS 兼容的数据组织格式,主要用于存放操作系统及用户的控制程序,采用的是扩展 BIOS 或设备驱动程序的管理方法。在固态 ROM 盘使用之前,必须将系统或控制程序按一定的数据格式固化到 EPROM 芯片中。DOS 系统是基于逻辑扇区对磁盘进行操作的,对固态 ROM 盘的操作也是如此。一般情况下,ROM 固态盘上的数据组织形式是与软磁盘相对应的,它是将软磁盘上的数据按逻辑扇区的顺序以二进制文件格式一一传送到存储器的相应“扇区”中来完成固态 ROM 盘的初始化工作的。因此我们先来分析一下软盘的数据组织形式。

## 1 软磁盘结构及数据的组织形式

软磁盘是计算机系统中一种重要的数据外存,在使用之前,这必须经 DOS 的 FORMAT 命令格式化。FORMAT 命令主要完成两件事情:(1)建立磁盘的标准磁道记录格式;(2)建立 DOS 使用的文件目录表、分配表和磁盘参数表。前者称为磁盘的“物理格式化”,后者称为磁盘的“逻辑格式化”。前者在磁盘的表面上划分了若干个同心圆——记录信息的磁道(track),每个磁道又被分成若干个扇段——扇区(sector)。盘的记录面数、每磁道的位置是由磁盘及驱动器的硬件特性决定的,是固定不变的。但每个磁道中的扇区的位置,扇区的数目以及每扇区记录的字节数却是由软件控制的。DOS 标准的格式化命令确定了每张磁盘的磁道数、磁面数、每道扇区数和扇区字节数。大多数 5.25" 的软盘划分为 40 或 80 个磁道,每道为 9 个或 15 个扇

区,容量为 360KB 或 1.2MB,3.5" 软盘划分为 80 个磁道,每道有 18 个扇区,总容量为 1.44MB,DOS 标准扇区大小是 512 字节。经过物理格式化的软盘的扇区安排是顺序的。

DOS 的逻辑格式化对软盘上所有的扇区以逻辑扇区顺序编号,DOS 就利用这些逻辑扇区号有效地组织对文件的访问。所谓逻辑扇区是指在一张软盘片上以 0 头 0 道的第 1 扇区作为逻辑扇区的 0 扇区,以后所有的磁道和磁头上的扇区都按一定的顺序连续编号的相对扇区。相对编号的顺序是:对某一磁道,先编最低磁头号下的所有扇区,然后再编下一磁头号的所有扇区号,直到该磁道所有磁头号下的所有扇区都编完,再进到下一个磁道,依次类推,最后到最高磁道上最高号磁头下的最后一个扇区为止。简言之,先变扇区,然后变磁头,最后变磁道。如双面双密度的 360K 磁盘,0 头 0 道上的相对扇区号为 0 到 8,而相对 9 扇区号是 1 磁头 0 磁道上的 1 扇区。整个磁盘的结构如图 1 所示。

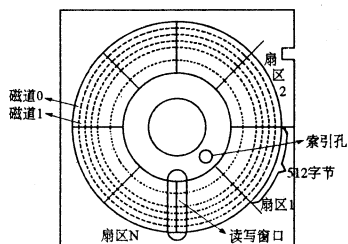


图 1 磁盘结构形式

DOS 为了能够有效地组织对不同类型磁盘的管理,必须在磁盘的一个特定的区域存放说明磁盘的具体存储信息,以便磁盘对于 DOS 透明。DOS 的磁盘信息大致可以分为四部分,第一部分是保留区,通常也叫引导区;第二部分是文件分配表,它记录磁盘空间的使用分配情况;第三部分是文件的目录区,它用来存放磁盘中文件的长度、地址、建立日期和时间等信息;第四部分才是用户的数据区也就是真正数据的存放区。

## 2 文件的访问和固态 ROM 盘初始化文件的生成

磁盘文件的访问过程是先通过 DOS 系统调用程序,进行文件的逻辑定位,并将文件访问转换为对某个

逻辑扇区的请求,然后通过公式:

物理扇区(R)=逻辑扇区 MOD 每道扇区数+1

物理磁头(H)=(逻辑扇区÷每道扇区数)MOD 磁头数

物理磁道(C)=逻辑扇区÷每道扇区数×磁头数注:MOD 指取模,而除法的商应取整。

将逻辑扇区的转换成实际的物理扇区,磁道号、磁头号、扇区号,作为进入盘驱动程序 INT13H 的入口参数,然后由 INT13H 盘处理子程序按照物理磁头、磁道、扇区号调用相应的硬件资源,完成数据的传输。

固态 ROM 盘作为一种高速、高可靠性的计算机外存,其存储介质采用的是半导体存储器,而并非是组成软盘的磁性材料。所以它实际的数据组织也就有其特殊性。ROM 盘是由存储器芯片按照地址顺序划分为一个个 512 字节大小的‘逻辑扇区’,这些扇区与软盘的逻辑扇区是一一对应的。固态 ROM 盘在使用之前,必须经过初始化过程,也就是要将所需要的用户程序固化到相应的 EPROM 存储芯片中。这就需要将软盘逻辑扇区中的内容——‘映射’到 ROM 盘相应的‘逻辑扇区’中。这些按地址顺序由小到大排列的‘扇区’就相当于软盘的逻辑扇区。ROM 盘数据的管理有两种方法,扩展 BIOS 方法和设备驱动程序法,这两种方法的实质是完成了 ROM 盘文件和‘逻辑扇区’之间的转换。最终交给 DOS 去管理的只是固态 ROM 盘的这些‘逻辑扇区’。为了使 ROM 盘‘逻辑扇区’的数据与软盘的逻辑扇区的数据一一对应,一般的做法是要固化的程序拷贝到软盘上(也可以包括操作系统),然后根据固态 ROM 盘的构成——由几片 EPROM 组成,每片容量多大,通过分割程序将磁盘数据分割成与 ROM 芯片容量相应的几个二进制小文件,然后再通过外部或内部的 EPROM 写入电路将这些二进制小文件顺序地固化到相应的 EPROM 芯片中,以完成 ROM 盘的初始化安装。分割程序的主要目的是按磁盘逻辑扇区的顺序将软盘文件分割成一个个小的二进制文件,以备固化时用。

下面以 1.2M 固态 ROM 盘为例,分析一下这个分割程序的编写。

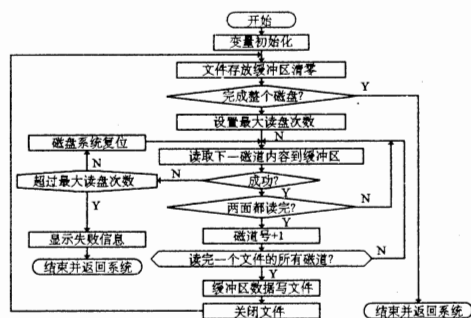


图2 文件划分程序流程图

以 1.2M 软盘为例,假设有一 1.2M 的固态 ROM 盘由 10 片 27010 EPROM 组成。在软盘中保存有我们所需要的应用程序和操作系统,现在要把软盘中的数据按逻辑扇区顺序分割成 10 个小的二进制文件固化到相应的 EPROM 芯片中。分割程序的流程图如图 2 所示。

程序一开始先要进行一些变量的初始化,以确定每个小文件所在的磁道、扇区。1.2M 的软盘每面由 80 个磁道,每道由 15 个扇区构成。要将 1.2M 容量的软盘分割为 10 个小的二进制文件,则每个小文件的容量应为:  $1.2 \times 1000 \times 1024 \div 10 = 122880$  字节,按 DOS 标准每扇区 512 字节计算,每个小文件应占用:

$122880 \div 512 = 240$  扇区,进一步换算为磁道应为:

$240 \div 15 \div 2 = 8$  个磁道,也就是说每个小文件占用 8 个磁道 240 个逻辑扇区,共 122880 个字节。第一个文件占用 1 到 8 磁道,第二个文件占用 9 到 16 磁道,……,第十个文件占用第 73 到 80 磁道。完成变量的初始化后,接下来要把 DOS 文件存放缓冲区清零,以备存放磁道内容之用。注意 DOS 的文件存放缓冲区一般设定为小于 64K,而我们的一个文件就有 120K,所以每个文件要分两次存取,生次保存 60K。在进行磁道的读取时要应用 ROM—BIOS 的功能调用,即 INT13H 盘驱动程序,应用其 02H 号功能将制定扇区的内容读到内存缓冲区。每次读取一个磁道共 15 个扇区的内容。由于 INT13H 的入口参数为物理扇区,即磁头、磁道和扇区号,故应将逻辑扇区的顺序转换为相应的物理扇区去进行操作。因为不能保证每次磁盘读取时都能成功进行,为了确保文件分割的连续性,通常设定最大读盘次数为 5 次,若 5 次读取仍不成功,则宣告磁盘的分割失败。注意,在第二次保存文件时,应改变前一半文件的读写指针位置,将其移动到二进制文件的末尾,以使两次保存的缓冲区内容能够正确连接。按照这个思想就可很容易地完成各种类型固态 ROM 盘固化二进制文件的准备工作。

### 3 结论及应用

利用这个磁盘数据分割程序能够完成固态 ROM 盘二进制初始化固化文件的生成,这些二进制文件可以是用户的控制程序,也可以是系统程序,还可以是其它任何 DOS 兼容文件与数据。这些文件通过在线或离线的编程器能够很容易的固化到固态 ROM 盘相应的 EPROM 芯片中,以完成 ROM 盘的文件初始化准备工作。

### 参考文献

- 戴水贵编. DOS/BIOS 功能调用及程序例. 海洋出版社, 1992. 8
- 白为民编译. DOS/OS/2 操作系统高级使用技巧. 中国科学院希望高级电脑技术公司, 1990. 4
- 马骏, 宋颖慧, 赵万生等. 工业环境下高性能固态盘的设计与实现. 机械与电子, 1995(4)

# 面向对象程序设计的分解与归纳方法

姚世军

(解放军测绘学院计算中心 郑州 450052)

**摘要** 面向对象开发方法多数是基于自底向上方法。本文介绍了自顶向下方法的变型—分解和归纳方法。在实际中把二者有机地结合起来会使系统更加可靠和可读。

**关键词** 面向对象(OO) 分解和归纳

## 1 引言

面向对象(Object-Oriented)是一种与传统软件工程功能方法完全不同的方法,它以对象为中心。OO的方法和技術是目前软件研究和应用开发中最活跃的一个领域,被认为是90年代的核心技术之一。目前有许多新的方法学来支持面向对象的软件开发。用面向对象方法写的程序由对象组成,而对象是模拟现实世界的事物,因此它们具有与模拟的现实世界相似的结构,并有高度可理解性;这些特性有利于软件维护和软件重用。

绝大多数的面向对象分析方法均是基于自底向上的方法构造系统,而忽略自顶向下方法的分解问题,其实软件系统并非简单的对象堆砌就能得到。自底向上方法有两个基本步骤:一步是由较小的对象建立较大的对象(组合),另一步是从一般化对象建立更具体的对象(专业化)。

与自底向上方法相对的是自顶向下方法。它是基于分解和归纳。分解是将大的更复杂的对象分解为更小的更简单的对象;归纳是专业化的反面,即把几个专用对象的共同特性抽象出来并用一般方法总结。分解就象现实中分配任务一样,即把大任务分成小任务,小任务又直接分配给每个人。自顶向下方法使用类似的过程:大任务分解成小任务,然后小任务分配给对象,并由对象来完成单个小任务。

在实际应用中,有些系统适用于自底向上,另一些系统适用于自顶向下方法,但大多数情况下使用这两种方法的结合是最合适的。

OO PASCAL, C++等其它面向对象语言都更有利于自底向上的开发方法。本文的目的是介绍用自顶向下方法来建立系统也象自底向上方法一样好用和自然。

在研究自顶向下方法中的分解和归纳方法时不把代码实现与设计分开。因此,它对应于生存周期的“开发”阶段,设计和编码是组合在一起的。

## 2 分解和归纳方法

用面向对象的分解和归纳方法(简称为OODG方法)开发程序时,有两种对象类:

• 完成类 即完全定义好的类。

• 派生类 即类已存在并知道类的某些特性,但没有它们的编码。

OODG开发方法分两步。第一步是分解,它把类分成更小更简单的派生类或是把它分解成在基本程序设计语言结构中已完全定义的类;第二步是归纳,它建立派生类之间的继承关系。

就以C++为例,程序的开发是从MAIN()开始,它控制整个程序。第一步是分解MAIN(),此时会生成几个派生类,下一步是选择其中一个派生类进行分解或归纳,反复进行,直到再没有派生类为止。

### 3 分解

类的分解是由几个活动组成,下面以发生顺序介绍:

#### 3.1 选择一个派生类

在派生类集中有多个派生类,其中一个选作将要分解的类。这个类的选择既影响到最终程序的结构也影响到程序开发的过程,因此必须仔细选择。选择期间应主要考虑的最重要的三个因素是:

##### (1) 风险控制

先选最有风险的类。软件开发本身是个冒险的过程。在对系统进行太多努力之前判定风险是合适的。例如响应时间就是应该考虑确定的,因为响应时间可能是系统的关键参数,它可能决定系统的成功或失败。

##### (2) 过程相容

所谓过程相容就是要求开发方法不强迫开发者修改完成类。换言之,它优化开发过程使每个类只有一次开发访问。如果没有过程相容,类就可能反复被访问,而每次访问都可能修改类,这样的过程更难于控制并可能在项目中引起混乱。

每当要把过程相容作为考虑的因素时,就应该选择那些在实现其它派生类时最不可能被改变的派生类,通常这种派生类处在使用关系层次中的最高层。这样它的定义可能有助于更好地了解其它派生类。例如:可往其它派生类加入新成员。

##### (3) 易于选择

当考虑到这个因素时,就是要选择容易编程实现的类,并且它们的分解不会产生新的派生类。当过载(Override)作为简化程序测试的方法时,这个选择是

合适的。

显然在具体应用中这三个考虑的因素可能相互矛盾,此时必须做合理的权衡。

### 3.2 选择类的逐步求精

一旦选择了要进行分解的类,类的数据和成员函数就要分解和定义。数据成员的定义可能会引用其它派生类,成员函数的代码可能调用派生函数(这叫临时代码)。这一方法与传统的逐步求精方法相似。

### 3.3 派生函数的类成员

在这个活动中,要确定所有无类成员的派生函数的成员。首先必须建立派生类函数和类之间所有匹配关系。如果函数 F 和类 C 的对象 O 之间有信息交换,则派生函数 F 与派生类 C 有匹配关系。在逐步求精之后所有匹配关系组成一个图。每个匹配关系必须转换成下列情况之一:

① 类成员;② 返回值;③ 形式参数。

对每个函数,最复杂(或最强)的关系是类成员。原理是如果所有复杂关系都成功地隐藏在类里面,那么整个结构显得清晰并且增加了代码的可读性和可重用性。实际中把关系的强度定义为:分析函数和类之间交换信息的数量。信息交换的数量小表明关系弱,数量大表明关系强。函数与类的接口应容易解释,用简单一句来解释称为弱关系,而复杂的解释表明强关系。大多数情况都有明显的强弱关系。

对于其它弱关系,至多可以把一个关系变成函数的返回值,并且必须是从函数中传出的值;其余的关系将成为形式参数。

### 3.4 代码更新

在这个阶段要更新临时代码并将其转换成永久代码。所有函数调用都必须更新,在语法上这就意味着在函数名前面必须插入对象名作为前缀,实参表也要放在函数名后面。

所有新的派生类和新的匹配关系加入到派生类集合中。这一步使定义类的状态由“派生”变成“已定义”。

### 3.5 验证

验证有新的已定义类的程序。为了验证所有派生类,用近似于最终实现的功能的程序段来暂时实现所有派生类。

### 4 归纳

归纳是 OODG 方法的另一步,它有下列活动:

#### 4.1 建立继承关系

先选两个派生类,在它们之间建立继承关系,其中一个做为基类,而另一个做为派生类。在这个活动时,建立的新派生类可以为另一个或几个派生类的基类。

#### 4.2 调整基类的函数成员

作为归纳的另一部分,有些成员函数可能被传到基类,因为它们处理基类中定义的更通用的数据结构。这正是开发者的目标之一:尽可能使函数成员传到继承树的更下级,这样就可以建立更加通用和可重用的基类。

### 5 结束语

在面向对象研究中,提出许多有价值的开发方法。各种方法都有自己的优点或适用范围。面向对象的自顶向下方法它除了有通常面向对象分析的特点外,还引入了 ensembles 概念。ensemble 是与对象不同的子系统,它包含了一组基本对象,并且有内部的并发行为。面向对象的分解和归纳方法实际上是自顶向下方法的一种变型,是基于逐步求精技术。OODG 方法在国外已被用到多个系统中,并取得较好效果。在应用中把它与自底向上的开发方法有机地结合起来,将使开发过程性、程序可靠性更强。

### 参考文献

- 1 P. Coad, and E. Yourdon, Object-Oriented Analysis, Prentice-Hall, 1990
- 2 V. Rajlich, Stepwise Refinement Revisited, J. Syst. Software 5, 1985
- 3 蔡希尧,陈平,面向对象技术. 西安电子科技大学出版社, 1993

(上接第 62 页) \*y+=xlate\_y;

```
* x *=scale_f;
* y *=scale_f;
}
ont.h 程序清单:
unsigned st[10]={
    0x0005, 0x0516, 0x1626, 0x2635, 0x3532, 0x0161, 0x0262, 0x6063, 0xffff,
    0xffff
};
xidef.h 程序清单:
#define WIDTH 80L
#define XMAX 639
#define YMAX 479
#define XMIN 0
#define YMIN 0
#define TRUE 1
#define FALSE 0
#define ON 1
#define OFF 0
#define SOLID 0xfffff
#define NORMIT 0
#define XMETRIC 0.237546468
#define YMETRIC 0.096601942
```

```
#define PI 3.141592654
union
```

```
{
    struct
    {
        unsigned col2,4;
        unsigned row2,4;
        unsigned col1,4;
        unsigned row1,4;
    }part;
    struct
    {
        unsigned word;
    } whole;
}font;
```

本程序在 AST 486 及兼容机上运行通过。

说明:1. 用户可根据需要自行造字,存入笔画图形库 font.h 中。

2. 对源程序稍加修改,就可命令行参数调用程序使图形对象在指定位置以指定高度、宽度、倾斜度、旋转角、比例因子而显示。

# Windows 下 PC 机与 8098 单片机的串行通信

杨 军

(第四军医大学物理教研室 西安 710032)

王宏山

(第四军医大学唐都医院设备科 西安 710038)

**摘 要** 本文介绍了 PC 机与 8098 单片机串行通信的数据接收、硬件连接及 PC 机对 8098 单片机的控制。

**关键词** 通信 WINDOWS 消息 单片机

## 1 概述

8098 单片机是一种片内集成有十位 A/D 转换的性能价格比极高的微处理器。以 PC 机和 8098 单片机构成的控制和数据采集系统不仅控制简便、灵活,而且人机界面友好,进行大量的数据处理极为方便。

对于 DOS 操作系统下的 PC 机,上述功能是用汇编语言或 C 语言编制中断服务程序实现的。而在 WINDOWS 操作系统下,WINDOWS 的设备无关性要求避免用中断方式对内存或硬件设备进行直接的存取控制。利用 Borland C++3.1 的 Windows.h 提供的通信函数可以实现 PC 机对单片机的主从控制。

## 2 硬件构成

PC 机和 8098 单片机的连接如图 1。

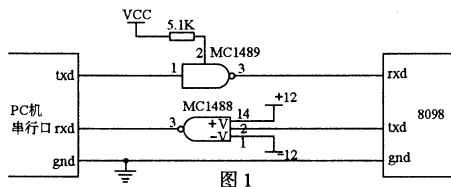


图 1

标准的 RS232C 上的高电平信号在 +5V~+15V 之间代表逻辑 0,与 TTL 电平不同,而 8098 的串行口输入和输出均是 TTL 电平,故 8098 的串行口 TXD、RXD 须进行电平转换才能发送和接收。该连接的最大传输距离为 30 米。

## 3 8098 单片机的串行通信程序的编制

8098 单片机串行口有三种异步方式。当工作于异步方式 1 时可全双工操作,即能同时收发。将 8098 设定为通信方式 1,即通信方式为每 10 位构成一串行帧,1 位起始位,8 位数据位(最低有效位在前),中断工作方式。为便于叙述,8098 的 A/D 转换结果暂取高 8 位数据。8098 单片机串行通信程序编制如下:

在主程序中,首先完成串行口的初始化工作:

```
1db IOC1, #20h          ;p2.0 口设为 TXD
1db baudrate, #09h      ;波特率寄存器中置
```

09,波特率为 9600 比特

.....

8098 单片机的波特率计算公式如下:

$$\text{Baudrate} = \frac{\text{Xtal1}}{64(B+1)}$$

其中:B 为波特率寄存器中的值

完成初始化后,中断服务程序中就可以接收和发送数据了。

```
org 4000h
sioint:pushf
rdcycl:1db sstatu, #00h    ;sstatu 为片内寄存器
1db stemp0, spstat
orb sstatu, stemp0        ;保存串口状态寄存器的值
andb stemp0, #60h
jne rdcycl                ;读取后应自动清零
getb: jbc sstatu, 6, snrb ;不是接收中断,跳转
.....
snrb: jbc sstatu, 7, sret ;不是发送中断,返回
.....
sret: popf
ret
```

## 4 PC 机通信程序

PC 机的通信程序用 Borland C++3.1 编译。

WINDOWS 程序是事件驱动的,当 PC 机串行口接收到 8098 单片机发送的数据后,产生 WM\_COMMNOTIFY 消息。WINDOWS 程序对该消息进行处理,完成数据的存储,图形显示,打印,对图形的局部放大显示等处理。

以下介绍 PC 机的数据接收程序:

### 4.1 通信

WINDOWS 系统函数库包含通信中断功能。当有通信动作时,产生一硬件中断,暂停程序运行,把输入的数据存入内存缓冲区的“接收数据队列”中,然后将控制权交给 WINDOWS 系统。类似的,要发送的数据的字节也要写入“发送数据队列”中。建立串行通信接口首先要用 OpenComm() 函数打开一个通信道路,并建立函数中定义的发送和接收数据队,并用 SetCommState() 函数定义通讯设备的设置,如波特率,数据位数,停止位数,校验方式,使 PCB 数据结构不

效。以下是实现上述功能的程序:

```
case WM_CREATE:
.....
nComID=OpenComm("COM2",1024,128);/* 打开串口 com2,建立 1024 字节的输入队列 */
if(nComID<0)
    MessageBox(hWnd,"Can not open COM2","WARNING",MB_OK);
else
    FlushComm(nComID,0);/* 清除发送队列 */
    FlushComm(nComID,1);/* 清除接收队列 */
    GetCommState(nComID,&dcb);
    dcb.BaudRate=CBR_9600;
    dcb.ByteSize=8;
    dcb.Parity=NOPARITY;
    dcb.StopBits=ONESTOPBIT;
    dcb.fChEvt=FALSE;
    SetCommState(&dcb);
    SetCommEventMask(nComID,EV_RXFLAG);/* 当接收队列字符并放入接收队列时,使通信事件发生 */
    EnableCommNotification(nComID,hWnd,200,-1);/* 当通信设备向输入队列写入 200 字节时,WM_COMMNOTIFY 消息,而输出队不规定必须多少字节 */
    MessageBox(hWnd,"COM2 is open","Message",MB_OK
.....
```

#### 4.2 PC 机对 8098 单片机的控制

PC 机可通过串行通信口向单片机发送 ASC II 码指令,以达到控制单片机系统运行程序的目的。如发送字符串"START"启动单片机,进而控制步进电机以规定的转速转动。

#### 4.3 数据接收

通信设置中定义,当接收数据队列中接收到 200 个数时,发送 WM\_COMMNOTIFY 消息,并可在对此消息的处理中进行数据读取:

```
.....
else if (CN_RECEIVE&LOWORD(1Param));
CN_COMMNOTIFY 的低字节为 CN_RECEIVE
{
    if ((nReadChars = ReadComm(nComID, bByte, 200))>0)
    {
        for(i=0;i<nReadChars;i++)
            bInBuf[i]=bByte[i];
    }
..... }
```

此程序中将一接收数据写入了 bInBuf[]中,如多次中断,可在此数组中依次写入。数据量大时,可以定义全局块。

#### 5 结束语

在医疗仪器的研制中,用本文方法实现的 PC 机对 8098 单片机的控制和数据采集系统在多台 386, 486 微机上实验,使用非常方便。软件中加入串口选择,可避免和鼠标冲突。

#### 参考文献

- 1 方建淳编著. 8098 单片机原理及应用技术. 天津科技出版社, 1990
- 2 王旭等译. Microsoft Windows 3.1 程序员参考大全. 清华大学出版社, 1993

## 异种机互连的实现

谢 彬

(中国工程物理研究院流体物理研究所 成都 610003)

**摘 要** 本文介绍了不同操作系统、不同类型的计算机之间进行通信的一种方法。文中对硬件的连接、软件的安装、实现通信的实用程序等作了详细的介绍。

**关键词** 计算机 操作系统 通信

#### 1 引言

随着社会的进步,人们对信息的需求日益增多。尽管单机已经具有很高的性能和较广泛用途,但如果仅仅是一台单机,它的容量和存储的信息毕竟有限,特别是软件及信息资源更有限,在国外被视为“信息孤岛”。网络的发展,为信息处理、信息共享提供了可能。人们可以通过计算机网络使用更多的计算机软件、硬件资

源。然而,计算机的种类很多,除人们常用的 PC 机,还有大型机、小型机、工作站等。在这些计算机上使用的操作系统也不同,有 OPENVMS, IRIX, OSF/1, WINDOWS, DOS 等。有的大型应用软件,例如 DYTRAN 只能安装在工作站上,有的又只能安装在小型机上。这就提出了如何实现使用不同操作系统的不同类型的计算机之间通信的问题。我们采用总线型加星型的结构,使用 DIGITAL 公司的网络软件,实现了异种机之间的通信。



## 2 硬件连接

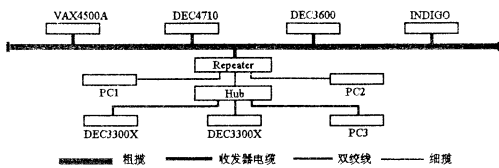


图 1 网络拓扑结构图

在我们的网络环境中,有这样一些机器:DIGITAL 公司的 VAX4500A 和 DEC4710 小型机各一台, DIGITAL 公司的 DEC3300X 工作站二台, DEC3600 工作站一台, SGI 公司的 INDIGO 工作站一台, 微机三台。采用总线型加星型拓扑结构, 用粗缆作主干, 将二台小型机、二台工作站、一台中继器(Repeater)通过收发器电缆连在主干上, 将二台工作站和一台微机用双绞线连在集线器(Hub)上, 集线器用细缆连在中继器(Repeater)上。另二台微机用细缆连在中继器上。使用的网卡有 DIGITAL 公司的 EtherWORKS 和 NOV-ELL 公司的 NE2000 及其兼容卡。DIGITAL 公司的网络软件提供了这些网络的驱动程序。图 1 是网络拓扑结构图。

## 3 网络软件安装及配置

### 3.1 使用 OPENVMS 操作系统的异种机的网络安装及配置

VAX4500A、DEC4710 和一台 DEC3300X, DEC3600 使用的是 DIGITAL 公司的 OPENVMS 操作系统。在安装好操作系统之后, 用操作系统提供的安装命令过程, 将随机提供的光盘上的网络介质装入系统盘, 进行网络配置, 设置本机数据库和网络上其它节点的信息(节点名, 节点地址)。如 VAX4500A, 定义其节点名 VAX450, 节点地址 1.1; DEC4710, 节点名为 DEC471, 节点地址为 1.2。网络软件的安装, 配置命令过程如下:

```
$ SET DEFAULT SYS $ UPDATE 安装网络软件
$ @VMSINSTAL
$ EDIT SYS $ MANAGER; SYSTARTUP -
VMS.COM 编辑启动文件加入命令行
$ @SYS $ STARTUP:DSV $ STARTUP.COM
$ SYS $ COMMOM:[DECSEVER] DSV $ CON-
FIGURE.COM 配置本机数据库
$ MC NCP 配置其它节点信息
NCP>SHOW NODE
NCP>SET NODE DEC471 ADDRESS 1.2
NCP>DEFINE NODE DEC471 ADDRESS 1.2
```

凡是需要与 VAX450 通信的节点均用 NCP 命令过程来设置。至此, 网络上使用 OPENVMS 操作系统的机器可互相通信、传递数据或作仿真终端等。为实现与微机的通信, DIGITAL 公司提供了一套网络软件 PATHWORKS V5, 它分为服务器(SERVER)方和客

户(CLIENTS)方软件。因此, 在 VAX450 上还要安装 PATHWORKS 的 SERVER 方软件, 安装过程同上。若计划将 CLIENTS 方软件通过微机安装在服务器上, 还需调用 ADMIN 来建立一个文件服务 PWV50, 作为 CLIENTS 方软件安装时的系统服务。

\$ SET DEFAULT SYS \$ UPDATE 安装 SERV-  
ER 方软件

\$ @VMSINSTAL

\$ @PWRC \$ CONFIG 配置 PATHWORKS

\$ EDIT SYS \$ MANAGER; SYSTARTUP -

VMS.COM 编辑启动文件

加入命令行, 使系统启动时, 自动启动 PATHWORKS。

\$ @SYS \$ STARTUP: PWRC \$ START-  
UP.COM

\$ ADMIN 建文件服务

### 3.2 使用 OSF/1, IRIX 操作系统的异种机网络安装及配置

另一台 DEC3300X 使用的是 OSF/1 操作系统, INDIGO 使用的是 IRIX 操作系统。而 OSF/1, IRIX 操作系统已将网络软件作在其内核, 无须另外安装网络软件。在操作系统安装完成以后, 只需要以 ROOT 注册, 编辑/vi/hosts 文件, 将与本机通信的机器的 IP 地址和节点名增加即可。具体命令过程如下:

>>>B

Login Username: root

Password: \* \* \* \* \*

# vi /et/hosts

编辑 hosts 文件

192.1.1.1 VAX450

增加其它节点的

192.1.1.2 DEC471

IP 地址, 节点名。

192.1.1.5 DEC332

192.1.1.6 INDIGO

192.1.1.10 PC10

### 3.3 微机方的网络软件安装及设置

PATHWORKS 的 CLIENTS 方软件为 18 张 3 寸软盘, 其中二张为 SETUP 盘, 若计划将 CLIENTS 方软件装在服务器上, 就需要 SETUP 盘。在实际使用中, 我们发现, 将 CLIENTS 方软件装在服务器上受 PATHWORKS 的 LICENSE 限制, 只要微机的硬盘空间充足(全部安装需要 52MB), 装在微机上更好。

在安装软件之前, 先建立一个子目录, 作为系统服务。CLIENTS 方软件将装在该目录下。用 SUBST 命令(DOS6.0 以上均有 SUBST 命令), 使该子目录与一个未使用的驱动器符对应。插入 1 号盘在 A 驱中, 启动安装过程, 按提示依次插入 2—16 号盘, 直到屏幕提示安装完成。转至安装目录下的 PCAPP 子目录, 启动配置过程。配置过程自动建立一个子目录 PW, 微机与其它节点通信所需要的文件安装在此。配置完成后, 回到 DOS 状态。查看 C:\PW 目录下, 是否有以下文件; 没有, 就需要将以下文件从 C:\PWV50\PCAPP 目录

拷贝到 C:\PW 目录下,它们是

```
MEM.BAT  FAL.EXE  DTR.EXE  NML.EXE
TNCPL.*  NCP.*    SPAWNER.*  NFT.*
SETHOST.*  FTP.*  TFTP.*    INETNAME.*
ARP.*      NETSTAT.*  RCP.*
```

这些文件是微机与异种机通信所需要的。

配置完成后,调用 DECnet 实用软件 NCP 来增加目标和节点名,节点地址。实现微机与使用 OPEN-VMS 操作系统的其它异种机通信。编辑 C:\PW 下 HOSTS 文件,增加其它节点的 IP 地址和节点名,通过 TCP/IP 协议,实现微机与使用 OSF/1,IRIX 系统的异种机的通信。在 C:\PW 下,用 STARTNET.BAT 命令启动网络,进行通信。通信完成后,在 C:\PW 下,用 STOPNET.BAT 命令,终止网络连接,释放内存,使微机可进行其它需要大量占用内存的工作。具体命令过程如下:

```
C:\>MD PWV50          建立子目录
C:\>SUBST E: C:\PWV50  使该子目录与一未使用的
                        驱动器对应

C:\>E:
E:\>A:PWINSTA          插入 1 号盘在 A:中,启动
                        安装过程,直到安装完成

E:\>CD PCAPP
E:\PCAPP>PWSETUP       启动配置过程
C:\PW>NCP              定义目标
NCP > DEFINE OBJECT FAL NUMBER 17 FILE
FAL.EXE
NCP > DEFINE OBJECT MEMMAN NUMBER 0 FILE
FILE MEM.BAT
NCP>DEFINE NODE 1.1 NAME VAX450  定义节点名,
节点地址
NCP>DEFINE NODE 1.2 NAME DEC471
NCP>DEFINE NODE 1.10 NAME PC10
NCP>EX
C:\PW>EDIT HOSTS        增加其它节点的
192.1.1.1      VAX450    IP 地址,节点名
192.1.1.2      DEC471
192.1.1.5      DEC332
192.1.1.6      INDIGO
192.1.1.10     PC10
C:\PW>STARTNET          启动网络。
C:\PW>STOPNET           终止网络连接。
```

#### 4 实现通信的实用程序

DIGITAL 公司的 PATHWORKS 是一种成熟的网络软件,通过它,PC 机与异种机可以互相通信。了解 PATHWORKS 所能提供的通信实用程序,有助于灵活地使用网络,为我们的工作提供更多方便。

PATHWORKS 提供以下实用程序:

- Mail:发送和接收存储在本地硬盘或邮件服务器相应目录内的信息。
- BCAST:发送信息到网络其它节点。
- RCV:接收系统管理员或网络其它节点用户发送的信息,包括邮件通知。

——TRCV:接收由 TCP/IP 网关发送的信息。

——SEDT:全屏幕编辑器,提供许多文本编辑和文本控制功能。

——USE:连接到服务器上的服务和应用程序。

——Terminal emulation:SETHOST 和 VT320 窗口终端仿真器允许个人计算机仿真 VT320 串行终端连到一个网络系统主机。

——DECnet utilities

——Network Control Program(NCP):分配、管理、配置网络节点。

——Network File Transfer(NFT):访问本地和网络其它节点处文件,用于访问运行 FAL 实用程序的 DECnet 系统上的文件。

——File Access Lister(FAL):网络上运行 FAL 实用程序的其它用户访问本地机。此时,本地机作为一台服务器,监听和接收远程节点文件访问请求。

——TCP/IP utilities

——Address Resolution Protocol(ARP):增加、删除、显示 IP 地址和硬件地址。

——File Transfer Protocol(FTP):拷入(拷出)二进制或 ASCII 码文件到网络上一台支持 TCP/IP 协议的主机。

——NETSTAT:显示网络和配置的信息。

——PING,显示指定主机是否在网络上,是否开机。

——INETNAME:建立一种连接,使本机可以使用 NETBIOS 应用(如一台 PATHWORKS 服务器)的资源。增加节点名,IP 地址。

#### 5 结束语

我们采用总线型加星型的拓扑结构,利用美国 DIGITAL 公司提供的网络软件,实现了 DIGITAL 公司的四种机型(VAX4500A、DEC4710、DEC3300X、DEC3600)与 SGI 公司的 INDIGO 及微机之间的互连,保证了各站点之间数据的可靠传递及对网络昂贵硬件资源(如大容量硬盘存储设备、网络外部设备等)、软件资源的充分共享和快速访问。实践证明,这一连接是成功的、可靠的。

#### 参考文献

- 1 胡道元编著. 计算机网络工程指南. 电子工业出版社,1993 年 4 月
- 2 胡道元编. 计算机局域网. 清华大学出版社,1990 年 9 月
- 3 PATHWORKS V5 for DOS and Windows TCP/IP User's Guide, Digital Equipment Corporation Maynard, Massachusetts
- 4 PATHWORKS V5 for DOS and Windows DECnet User's Guide, Digital Equipment Corporation Maynard, Massachusetts

# 基于模式识别的大罐排污自控系统

喻宗泉

(华中理工大学汉口分校 武汉 430012)

**摘要** 本文讨论油水界面各模糊模式隶属函数及模式样本的归属,给出了系统硬件电路及主程序流程。

**关键词** 大罐排污 油水界面 模糊模式 样本 隶属函数 贴适度

## 0 引言

从油井自喷或泵吸到达地面的油、气、水混合物,经集输管线、计量站、泵站、中间站,汇集到集油站的大罐中暂时存储。三相介质在静止的大罐内按比重分层,污水在下,原油在上,油水之间为油水界面,天然气逸出充溢罐顶。为了保证进入长输管线或外运装车装船的是原油,需要定期将积于罐下部的污水排出。早期的大罐排污是依靠人工进行的。工人在开放位于罐底的排污闸阀后,一直守候在出水口沟边,看到水中有油的乳化物或散块小油团时赶紧关闭闸门,否则放跑了原油就是事故。由于大罐内污水的高度无法测量,因此排污所需要的时间往往心里没底。自动排污势在必行。

## 1 模式识别

原油和污水除了比重不同外,还有其它一系列物理性质和化学性质的差异。原油粘度高,含腊量高,常温下呈半固态,为绝缘介质。污水为液态,因富含盐类为良导体。在油水分层时,油水界面往往不清晰,随油水温度不同呈现不同状态。温度高时原油融化在水中为乳状,温度低时污水中有悬浮的油团油块,体积悬殊很大,大的有几个立方厘米,小的肉眼未必能分清。油水界面不是指一个平面,而是有一定厚度的中间层。

### 1.1 模糊模式的隶属函数

待识别对象油水界面不是确定的单个元素,它的每一样本是论域  $U$  上的一个模糊子集  $\tilde{B}$ 。 $\tilde{B}$  的特征指标是由测量转换电路提供的样本温度、电阻率、电容量,由这些参数计算出来的比重值不是一个确定值,而是一个样本区间  $[X, Y]$ 。模式识别的任务是使用逼近原则确定对象集合  $\tilde{B}$  与  $\tilde{A}_i$  的贴适度  $\sigma(\tilde{B}, \tilde{A}_i)$ ,指出  $\tilde{B}$  归属于那一类模式,决定是否排污。

考虑比重  $u$  的变化域作为论域  $U$ ,按照不同重量作为变量指标条件下,将油水界面划分成 4 个模糊模式:  $A_1$ ——含水的油

$A_2$ ——油水各半的混合物

$A_3$ ——含较多油的水;以上不能排放

$A_4$ ——含微量油的水;可以排出

每一个模糊模式之间没有截然的分界线,例如三种样本的油含量分别为 51%, 50%, 49%, 我们绝不能

贸然把第一个样本称为含水的油,归于  $A_1$ ;也不能把第三个样本称为含油的水,归于  $A_3$ 。这是因为各模糊模式之间没有清晰的界限,它们的边界是模糊的,需要借助隶属函数描述。鉴于  $\tilde{A}_i$  按重量划分,隶属函数理应为线性。比重  $u$  对于各个模式的隶属函数可记为:

$$\begin{aligned}\tilde{\mu}_{A_1}(u) &= \begin{cases} 0, & \text{当 } u \leq e \\ (u-e)/(f-e), & \text{当 } e < u < f \\ 1, & \text{当 } u \geq e \end{cases} \\ \tilde{\mu}_{A_2}(u) &= \begin{cases} 0, & \text{当 } u \leq 1 \text{ 或 } u \geq f \\ -(u-e)/(f-e), & \text{当 } e < u < f \\ 1, & \text{当 } d \leq u \leq e \\ (u-c)/(d-c), & \text{当 } c < u < d \end{cases} \\ \tilde{\mu}_{A_3}(u) &= \begin{cases} 0, & \text{当 } u \leq a \text{ 或 } u \geq d \\ -(u-c)/(d-c), & \text{当 } c < u < d \\ 1, & \text{当 } b \leq u \leq e \\ (u-a)/(b-a), & \text{当 } a < u < b \end{cases} \\ \tilde{\mu}_{A_4}(u) &= \begin{cases} 0, & \text{当 } u \geq b \\ -(u-a)/(b-a), & \text{当 } a < u < b \\ 1, & \text{当 } u \geq a \end{cases}\end{aligned}$$

由工作经验,式中系数取值为  $a=0.02, b=0.05, c=0.35, d=0.45, e=0.55, f=0.65$ 。

### 1.2 模式样本的贴适度

在确定模式样本的隶属函数时,在模式样本区间  $[X, Y]$  内使用线性内插方法插入  $n$  个分点,把  $[x, y]$  分成  $n+1$  等分,这些分点的横坐标依次为  $X_0, X_1, \dots, X_n, X_{n+1}$ , 且区间端点  $X=X_0, Y=X_{n+1}$ , 设在小间区  $[X_{i-1}, X_i]$  内样品共试验  $m_i$  次,则由模糊统计规律:

$$\tilde{\mu}_B((X_{i-1}+X_i)/2) = m_i/m, i=1, 2, \dots, n+1$$

式中  $m$  为  $m_i$  中最大值。由此可以确定模糊子集  $B$  的隶属函数为:

$$\tilde{\mu}_B(u) = \begin{cases} 0, & \text{当 } u \leq x \text{ 或 } u \geq y \\ (u-x)/(p-x), & \text{当 } x < u < p \\ 1, & \text{当 } p < u < a \\ -(u-q)/(y-q), & \text{当 } q < u < y \end{cases}$$

模式样本的贴适度计算公式:  $\sigma(\tilde{B}, \tilde{A}_i) = 1 - (\lambda_i - \lambda_i) + (\tilde{A}_i \otimes \tilde{B} - \tilde{A}_i \oslash \tilde{B})$ .  $\lambda_i = \bigvee_{u \in U} \tilde{\mu}_{A_i}(u)$ ,  $\lambda_i = \bigwedge_{u \in U} \tilde{\mu}_{A_i}(u)$ ,  $\tilde{A}_i \otimes \tilde{B} = \bigvee_{u \in U} (\tilde{\mu}_{A_i}(u) \wedge \tilde{\mu}_B(u))$ ,  $\tilde{A}_i \oslash \tilde{B} = \bigwedge_{u \in U} (\tilde{\mu}_{A_i}(u) \vee \tilde{\mu}_B(u))$ .

例如模糊子集  $\tilde{B}$  的样本区间为  $[0.01, 0.045]$  时,得出  $\sigma(\tilde{B}, \tilde{A}_1) = 0, \sigma(\tilde{B}, \tilde{A}_2) = 0, \sigma(\tilde{B}, \tilde{A}_3) = 0.83, \sigma(\tilde{B}, \tilde{A}_4) = 1$ .  
 $\max(\sigma(\tilde{B}, \tilde{A}_1), \sigma(\tilde{B}, \tilde{A}_2), \sigma(\tilde{B}, \tilde{A}_3), \sigma(\tilde{B}, \tilde{A}_4))$   
 $= \max(0, 0, 0.83, 1) = 1 = \sigma(\tilde{B}, \tilde{A}_4)$

则此时样本属  $A_4$ , 应开启闸阀继续排污。与此同

时,  $\sigma(B, A_3) = 0.83$ , 告诉我们, 闸阀开度应当小些。

油水界面划分成 4 个模糊模式, 而不是 2 个 ( $A_1$ ——含水的油及含较多油的水,  $A_2$ ——含微量油的水), 是由现场实际情况决定的。进入大罐的油水温度通常在  $40^\circ\text{C}$  以上, 一段时间后降到常温, 水中的油就会凝固, 处于  $A_2, A_3$  的样本会进入  $A_4$  模式, 因此划分成 4 个模式有利于排污的进行。

## 2 系统硬件

硬件电路由测量转换电路、8098 单片机系统、外围接口电路三部分组成。

### 2.1 测量转换电路

功能是测量温度、电阻率、电容量三个物理量, 将它们转换为模拟电压, 经 A/D 转换后分时进入 8098 的 P3 口。被测模拟量不从 P0 口直接输入而选择 P3 口, 是因为阻容量的值小, 变化范围小, 需要在外电路或用程序对运放的零点漂移和 A/D 转换非线性进行补偿。现以电阻率测量电路为例说明原理和调试过程, 温测和容测电路略。见图 1, 运放  $A_1$ 、T、外围电阻构成压控恒流源, 输出电流如图 1。

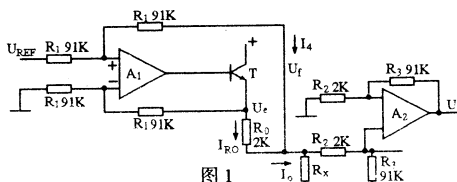


图 1

$$I_{RO} = (U_c - U_T) / R_0 = U_{REF} / R_0 = \text{constant}$$

因被测电阻  $R_X$  为  $0 \sim 30\Omega$ , 相对于  $R_1$ , 可忽略不计, 因此

$$I_O = I_{RO} + I_c = I_{RO} + U_{REF} / 2 / R_1 = \text{Constant}$$

$A_2$  为测量和 R/V 转换电路, 输出电压  $0 \sim 5V$ :

$$U = R_3 I_O R_X / R_2 = U_{REF} R_3 R_X / R_2^2.$$

为了提高测量和转换精度, 在电路设计和调试中应当注意以下一些问题:

- 选用受温度影响较小的金属膜电阻。同一标称系列的阻值予以筛选, 做到阻值在个位数上一致。

- 使用高增益低失调漂移的运算放大器, 例如 HA4620/5604 或 LM163/LM363。

- 必要时增加一级隔离驱动器, 消除输入线之间对地电容失配造成的影响。

- 罐区因天然气存在, 禁止有电火花, 因此被测电阻联线较长, 电路虽然在原理上能克服联线电阻的影响, 但需要反复校准零点, 且连接要牢靠。

### 2.2 单片机系统和外围电路

单片机系统由 8098、74373、2764、6164 组成, 接收被测信号信息经模式识别确认是油、是水、是油水界面中哪一模式。外围接口电路将单片机输出信号分别送电磁阀控制绕组、排出液量显示及介质显示。键盘的功能是整个排污系统实施人机对话及手动控制。

## 3 程序及流程图

由主程序和 2 个子程序组成, 主程序功能是接口芯片初始化, 调参数检测子程序, 调贴度计算子程序, 显示样本模式、温度、流量, 由人机对话选择是否需要排污, 如果选择要排则连通阀, 自动断定排污结束后关机。

## 4 结束语

实验结果表明: 识别准确无误, 动作可靠。另一方面, 在集油站集散控制系统的设计中, 大罐排污属基层管理层次的过程控制级。大罐排污系统引入人工智能领域里的模式识别并实验成功, 实施对现场变送器、执行机构、显示仪表的监测控制, 无疑对与上层计算机相连, 接受上层的管理信息提供了基础, 充分体现了集中管理、分散控制的设计指导思想。

### 参考文献

- 1 汪培庄. 模糊集合论及其应用, 上海科技出版社, 1984
- 2 Lee, E. T. Application of fuzzy languages to pattern recognition, kybernetik 6:3, 1977
- 3 方建淳编著. 8098 单片机原理与应用技术, 天津科学技术出版社, 1990
- 4 张跃, 邹寿平, 宿芬编著. 模糊数学方法及其应用, 煤炭工业出版社, 1992

## 1997 年在国内召开的国际学术会议和组团派出计划

会议名称	日期	地点	联系人、电话、通讯地址
参加香港国际计算机会议(HKICC' 97)	10. 23	香港	中国计算机学会组团 杜子德
北京香港国际计算机会议(BJHK ICC)	10. 29	北京	杜子德 (010)62562503, 100080 北京 2704 信箱
参加第 9 届国际信息学奥林匹克(IOI' 97)	11. 30	南非	吴文虎(010) 62784141, 100084 清华大学
第 5 届国际计算机辅助设计和图形学会议(CAD/Graphics' 97)	11. 30	深圳	李华 (010)62565533—5658, 100080 CAD 开放实验室
中国国际计算机通信与网络博览会(Com-Net' 97)	11 月	上海	徐桂珍 (021)63276055, 200020 上海南昌路 47 号

# 抚顺乙烯化工有限公司 计算机网络系统设计与实现

卢 苇 徐晓玉

(四川联合大学 成都 610064)

张志洲

(抚顺乙烯化工有限公司信息中心 113004)

**摘 要** 该厂计算机网络系统包括了生产控制管理系统 TCS 和企业管理信息系统 MIS. 本文对这两个网络系统的设计思想及系统结构和系统特点以及它们的一体化结构都做了较为详细的阐述。

**关键词** 生产控制管理系统 管理信息系统 一体化

## 1 前言

抚顺乙烯是由中国石化总公司和辽宁省政府共同投资兴建,委托抚顺石化公司管理的有限责任制企业,企业现有职工 3200 人,企业建设总投资 21.5 亿元。公司拥有六套生产装置,分别从加拿大、法国、意大利、日本等国家成套引进,其中包括 11.5 万吨/年的乙烯装置,8 万吨/年的聚乙烯装置,6 万吨/年的聚丙烯装置,5 万吨/年的环氧乙烷装置,4 万吨/年的乙二醇装置,0.4 万吨/年的树脂后加工装置。各装置均采用集散型计算机控制系统。

根据该公司的生产管理情况,计算机网络系统应该包括两套网络管理系统,TCS 生产控制管理系统和 MIS 企业管理信息系统。这两套系统都有一定的特点和功能,形成一个完整的计算机网络系统,真正地集管理、控制为一体。

## 2 系统设计思想及实现方案

### 2.1 系统设计思想

将生产控制管理系统与企业管理信息系统连成一个网络系统,真正实现管理控制一体化的结构。根据公司的业务要求以及现有的实际情况,我们提出了以下的系统设计思想。

- 将 TCS 系统和 MIS 系统合并成一个完整的全厂计算机网络系统。

- 系统具有良好的开放性和可移植性。
- 系统具有良好的人机界面操作简单方便。
- 根据各部门的业务特点,各自形成一个独立的网络体系。

- 系统具有良好的安全性及保密性。

- 从整体上看全厂具有一个完整的计算机网络一体化的结构。从各个部门上看又有各自独立的部门管理系统。

### 2.2 系统实现方案

根据设计思想及公司的实际情况,我们采取了以

下的实现方案。

- 将各个生产装置的 DCS 系统连接形成一个 TCS 系统,用 VAX 小型机作为主机,用来显示各自生产装置的运行情况,同时为 MIS 系统提供生产方面的数据。

- 根据各个处室的业务开发编制各自独立的管理系统,再将各个子系统连接成一个主系统形成 MIS 系统。

- 用 HP9000/800 G50 小型机作为全厂计算机网络系统主服务器,用 Sybase 大型关系型数据库为核心的 Client/server 结构的网络体系。

- TCS 系统中 VAX3300 与主服务器 HP9000/800 G50 通过各自的以太网接口连接,采用的通信协议是 TCP/IP。

- MIS 系统中的各个子网与主网通过网桥连接。

- 用 C 语言编写转换程序将 TCS 系统中的数据传送到主服务器中供主网共享。

- MIS 系统中的子系统运用 Foxpro 作为客户端的开发工具。

## 3 生产控制管理系统 TCS 和计算机网络系统构成

### 3.1 生产控制管理系统 TCS 构成

各个生产装置的 DCS 系统类型分别为:

乙烯装置:MOD300;丁烯装置:TDC300

聚乙烯装置:MOD300;聚丙烯装置:NET90

环氧乙烷/乙二醇装置:FoxBORO 的 I/A, Series.

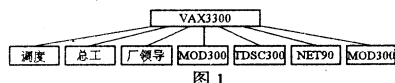


图 1

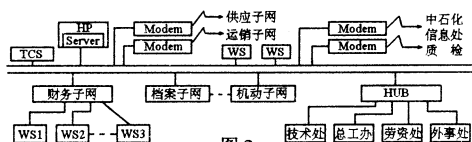


图 2

用 VAX 小型机 VAX3300 作为主机与各个装置的 DCS 系统连结起来组成一个网络系统。即生产管理

系统 TCS,该系统通过图形和数据为厂领导及有关部门实时显示各个生产装置的生产情况及运行情况。图 1 给出了 TCS 系统构成图。

### 3.2 全厂计算机网络系统的构成

用 HP 900/800 小型机为主数据库服务器, Sybase 大型关系型数据库为核心的 Client/server 网络体系。将全厂各管理部门按照业务和地理位置来组成各个子网。再将各个子网组成全厂管理信息网。图 2 给出了全厂计算机网络系统的构成图(办公楼内局部图)。

各个子网根据自己的业务用 Foxpro 作为软件开发工具来开发应用程序,再将公共信息送到主服务器中,供各部门共享。

## 4 系统特点

用 Sybase SQL Server 作为数据库一端的数据库管理系统,主要有以下特征:

(1)联机事务处理能力。针对系统各个子系统的特点,要求服务器一端的 DBMS 有着较快的联机事务处理能力。SQL Server 采用多线索技术降低了操作系统的开销,它能优化诸如业务调度、切换、磁盘高速缓存及封锁之类的联机处理功能。

(2)由服务器实现安全性和完整性。安全性是指授权并控制用户对数据的访问,而完整性是指数据的正确性和一致性,它主要包括了字段级完整性(数据类型及取值的合理值,即实体完整性),参照完整性(主键值和外部键值表的一致性),以及规则(用户定义的完整性)。在小型数据库管理系统如 Foxpro 中,把维护数据的安全性及完整性也完全交给用户在应用程序中实现,这样势必大大增加应用系统的开发难度,安全性和完整性得不到很好的控制。在 Sybase SQL Server 中,提供了多种机制,如用户定义数据类型,给缺省建立规则,提前将 SQL 语句编译为存储过程,以及触发器技术等来解决安全性和完整性的要求。

(3)对整个系统进行监控和性能调优比较方便。

Sybase 可对系统的性能进行监控,并利用硬件软件的容错功能保证整个系统的健壮性和可用性。系统管理员可随时监视整个系统的运行状况、及早发现问题,及早解决。另外,还可利用所得的性能信息,对数据库的设计和存储进行修改,使得整个系统始终处于最佳的运行状态。

(4)优秀的分布式数据管理功能。

在本系统中,一个应用横跨多个部门,多台计算机,只有这些节点协调一致地对数据进行存取,一个事务才可能完整的得以实现。在 Foxpro 中,由于不存在事务处理功能,保证事务处理过程中的数据一致性只能由应用程序实现,这不仅加大软件开发难度,而且往往使得一个事务不能顺利得以实现,整个系统处于停滞状态。

在客户端使用 Foxpro,主要有以下优点:

Foxpro 易编程、易维护、易修改。

在我国,大多数工矿企业的计算机应用人员对 XBase 系列数据库较为熟悉,有着丰富的使用经验。Foxpro 作为 XBase 数据库的升级产品,不仅全兼容以前的系统,而且为用户提供了更加易学、易用的功能和界面,拥有广大的用户群。鉴于这种状况,为最终用户提供一个他们比较熟悉的 DBMS 势在必行。Foxpro 将大型数据库的各种复杂概念屏蔽起来,用户看到的只是传统的微机数据库,而把这种不同数据库间的转换交由软件开发人员来完成。这样,在系统开发的初期,最终用户就可参与到软件开发过程中来,从数据库的设计,到人机界面的确定,到程序编码,离不开他们紧密合作,只有这样,才能开发出高质量的、真正实用的应用软件。如果使用其它数据库开发工具,用户面临的不只是陌生的开发环境,而且还有一大堆表、事务、规则、触发器等大型数据库中的名词,使得他们很难真正配合好开发人员。这样从数据库的前期设计开始就埋下了隐患,注定应用系统失败的命运。另外,软件的开发只是暂时的,将来的维护与升级则是长期、持久的工作,如果用户对开发工具和整个系统不熟悉,势必增加他们维护和升级的难度。这个应用软件的寿命便可想而知了。

## 5 结束语

抚顺乙烯化工有限公司的计算机网络系统真正实现了管理和控制一体化的结构,做到了与生产有关的管理信息系统的数据库直接从 TCS 系统中传递过来,进而实现了全厂数据信息共享,使厂领导能够及时掌握全厂的生产、供应、销售及及管理情况。

### 参考文献

- 1 李志蜀,银涛编著. NetWare 3. xx~4. xx 实用技术,电子科技大学出版社,1996 年
- 2 王利强,王强编著. Visual C++4.0 易学活用. 成都出版社,1996 年

(上接第 83 页)系统,介绍了基于超文本技术查询系统的开发。讨论了开发 Windows 帮助系统的几个关键问题,提出了解决办法,并给出了相应的程序。通过以上讨论我们发现,利用 Windows 本身特点,特别是进行充分开发有着相当重要的意义。另外,基于超文本技术开发查询系统相当简单和方便,具有很强的实用性。

### 参考文献

- 1 梁晋等. 中西文 Windows 3.1 使用指南. 西安电子科技大学出版社,1994
- 2 陈海东等. 多媒体技术及应用系统制作. 北京师范出版社,1994



# 医用眼底图象分析系统的研制

谢晓恒

(合肥市电子技术研究所 230027)

李卫华

(中国科技大学力学与机械工程系 合肥 230026)

**摘 要** 本文研制的眼底图象分析系统可以快速保存患者的眼底图象,同时又可以减少医生仅凭对患者眼底的观察诊断可能带来的主观误差。本文利用 CCD 摄像机把眼底图象取出并通过图象板把眼底图象转化为图象数据,送入计算机进行处理、分析。本系统利用扩充内存(EMS)技术完成了同时对 30 幅图象(每幅 256KB 数据)共 8MB 数据的实时存储,可以较好地满足医学研究的需要。另外,为了数据保存和传输的需要,根据眼底图象的特征,采用行程编码技术和二维编码技术对图象进行压缩,效果很好。

**关键词** 眼底图象 EMS 技术 行程编码 二维编码

## 1 引言

眼疾是人类的一种常见疾病,治疗眼疾常用的方法是医生通过对患者眼底的观察、分析,凭经验确定病情。因而,医师的主观因素(主要指经验)在诊断中占较大的比重,诊误率也可能增加,尤其对于刚参加工作不久的年轻医师更是如此。为了尽量克服主观因素引起的误诊,增加客观判断因素,本课题采用计算机处理方法开发了图象分析系统,可以方便地诊断眼疾。具体而言,利用 CCD 摄像机连续采集若干幅图象(一般 20~30 幅)实时存入计算机,通过计算机图象处理给出眼底图象的重要特征,作为眼疾的判断依据。

根据医学上对眼疾图象的实际需要,本系统可以达到快速性、真实性及存储大量数据的要求。具体而言,拍取患者的眼底图象时,为了不使其眼睛长时期受光的刺激,在极短的时间内(20 秒内)连续拍取 30 幅眼底图。实际上利用本系统平均每 0.6 秒可以取一幅图象,采集 30 幅图象大约 20 秒,可以满足要求;另外,每采集一次图象(30 幅)数据量极大(8MB 左右),这么大的数据不管保存还是传输都很不方便,故要对图象数据进行压缩。同时要求解压缩时图象不能失真太大。针对本系统,我们设计利用了行程编码和二维编码方法既实现了较大压缩比(1:10),又保证了图象的质量,较好地满足了眼底图象分析系统的要求。

另外,虽然目前国际上已经研制并推广了眼底照相机,但每台价格在 5 万美元左右,国内只有少数大医院才有能力购买使用。国内中小医院常用裂隙灯诊断眼疾,裂隙灯的价格只有两三千元。我们正是在裂隙灯的基础上设计了一套光学系统,把患者的眼底图象通过裂隙灯和光学系统成象在 CCD 象面上,而后通过图象采集软件把图象存入计算机,通过一系列图象处理手段得到眼底图象的特征作为诊断判据。

更重要的,本系统成本较低,是对国外眼底照相机

的一种有益补充,适合国内中小医院的推广应用。该系统是和安徽医学大学附属医院眼科联合开发并进行了临床试验,收到了较好的效果。另外,随着研究工作的进一步加深,该系统必会得到更好的完善和发展。

## 2 系统结构

系统的结构简图如图 1 所示:

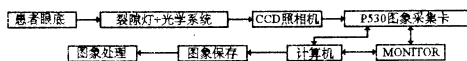


图 1 系统结构图

光学成象系统把眼底图象成象在 CCD 象面上,利用 P530 图象采集卡把图象数据采集存入计算机内,利用图象处理技术可以得到眼底图象的特征。

## 3 扩充内存(EMS)技术

为了满足医学上快速采集眼底图象的要求,本文利用了 EMS 技术进行数据存储。本系统使用 PC486 内存,其中常规内存(640KB)、上位内存(UMB)、高内存区(HMA)共占 1M,其余的为扩大内存(EMS)。在扩充内存卡的支持下,可以通过 640KB 以外的窗口来使用扩充内存,窗口尺寸为 64KB,以 16KB 为一页,扩充内存的任何部分都必须通过窗口才能使用,故同一时间只能访问 64KB。但通过指定不同的页号,可以寻址扩充内存中其它的任何物理页面。窗口的一页与扩充内存页面的对应可通过扩充内存管理程序 EMM386.EXE 自动进行,只要启动计算机时装入 HIMEM.SYS 后就可使用,使用时用 INT67H 激发其工作,本程序全部用汇编语言写成,速度完全满足医学上的需求。

## 4 图象压缩

如前所述,一次拍摄 30 幅图象共有约 8MB 数量,这么大的数据量不仅存储不方便,即使对硬盘也是难以忍受的,因此应解决图象压缩问题。

图象压缩也就是图象编码的问题,图象压缩要同时兼顾两个方面:一要达到一定的压缩比,二是保证图

象失真不是太大。综合这两方面的因素,本系统采用的行程编码及二维编码技术可以完全满足需求。

#### (1)行程编码

行程编码是较简单的一种,对某些图象的压缩效率也较高。图象的灰度值总有其相关性,在一定大小的区域内灰度值不会有太大的改变,利用一行内灰度值的相关就产生了行程编程方法。它是一种以行为单位的压缩方法,从一行的第一列开始顺序搜索,以图象灰度改变的幅度不超过某一最大值的最长序列为一个单位进行保存。每一单元的保留信息有定位信息、大小信息、灰度信息,这种方法的优点是位利用率高,压缩比高,失真度小,缺点是压缩文件的组织存储复杂,需要有许多位组合的计算。

#### (2)二维编码

二维编码利用图象的二维相关性。因为实际的图象不仅行具有相关性,而且列也具有相关性,利用二维相关可以更好地兼顾图象的空间相关信息。二维编码利用矩形块来划分子区域,它要保存的信息有起始行坐标,起始列坐标,终止行坐标,终止列坐标及子区域的灰度信息,一般情况下,行数与列数的不同并不能使子区域的面积有多大的增加,故子区域常取为  $N \times N$  的方块。实际结果表明,二维编码由于利用了图象的空

间相关性,具有比行程编码好得多的效果,是图象压缩中常用的方法。

#### 5 结论

本文根据医学上对眼底分析的要求,讨论了眼底图象分析系统的结构、组成,并实现了图象快速采集、分析处理、压缩等一系列功能,使系统具有快速性和实时性,基本上满足了医学上的需求。随着研究的进一步加深,该系统必然会得到更好的完善和发展,也希望本系统能尽快通过鉴定。

#### 参考文献

- 1 吴建康著. 数字图象分析. 人民邮电出版社, 1991. 11
- 2 周明德著. 微型计算机 IBM/XT(0520 系列)系统原理及应用. 清华大学出版社, 1990. 9
- 3 沈美明, 温冬婵著. IBM-PC 汇编语言程序设计, 清华大学出版社, 1991. 6
- 4 CA5300 图象采集卡用户手册. 大洋公司出版
- 5 Porilidis, T. Structural Pattern Recognition, Berlin, Heidelberg, New York; Springer Verlag, 1977
- 6 Chang, S. K., Pictorial Information Systems, Letore Notes, 1985

## 蓄电池快速充电模糊控制的初步探讨

童晓阳

(西南交通大学模拟中心 成都 610031)

童晓霞

(江苏理工大学计算机系 镇江 212023)

**摘要** 对于大容量铁路用蓄电池快速充电问题,针对其具有非线性、时变的、有纯滞后的特点,本文初步研究了采用模糊控制器对蓄电池快速充电进行控制,分析了其模糊控制器的组成,并讨论了模糊控制在蓄电池快速充电中存在的问题。

**关键词** 蓄电池快速充电 模糊控制

#### 1 问题的提出

蓄电池的快速充电为什么要引入模糊控制?

蓄电池的充电过程是一个复杂的过程,要实现对蓄电池快速充电的控制,是有相当难度的,原因在于蓄电池的充电控制系统是非线性的、时变的、有干扰的、具有纯滞后的控制系统。在充放电过程中,它的许多参数如类型、充电率、最大允许充电电流、内阻、出气点电压、温度、寿命等,对于每一种蓄电池都是不同的,有些参数在充电过程中随时发生变化。这些参数一般都不能直接获得,而且参数之间相互影响,即使这些参数都获得,那么所设计的控制器结构就非常复杂。这类系统

建模的困难使现代控制理论中的控制难以实现。采用传统的 PID 控制在整定参数时也比较困难。对某一种蓄电池为最佳的参数,对另一种就不是最佳。

传统的控制系统是建立在被控对象精确模型的基础上,如果当被控对象的数学模型很复杂或者数学模型无法建立时,控制系统就较难控制。蓄电池快速充电正是属于这种情况,由于蓄电池的充电过程有自己独特的物理化学规律。因此考虑采用模糊控制来进行蓄电池的快速充电。

建立在计算技术基础上的基于规则的控制方法,能较圆满地解决这些矛盾。模糊控制是这类方法中比较简单易行的一种。模仿人手控充电的操作经验,克服参数多变的情况,提高充电接受率,减小出气率。国外

已于八十年代末和九十年代开展了蓄电池的模糊控制,取得了一定的进展,新加坡的 Y.C. LINAG 和 T.K. NG 等已研究了应用在蓄电池快速充电中的模糊控制器的设计,通过试验证明了模糊控制增大了充电电流,提高了充电效率,对不同参数的蓄电池有较好的适应性。而国内尚无人作此方面的研究。本文结合蓄电池快速充电的规律,初步研究了运用模糊控制器来控制充放电过程的方法。

## 2 蓄电池快速充电模糊控制的依据

蓄电池快速充电一般采用三级充电和脉冲充电相结合的方法。三级充电的原理由图1所示,一级恒流和二级恒流都是以恒定的电流给蓄电池充电,二级恒压是以恒定的电压供给蓄电池,使得通过蓄电池的充电电流由一级的大电流逐渐减小为三级的小电流。脉冲充电是指充电电流或电压以脉冲形式加在蓄电池两端,实现的方法可以是调节充电电压的可控硅导通角,也可以调整充放电脉冲的宽度或充放电周期的大小。

充电过程主要受充电率和温度影响。图1中实线为充电高温时充电曲线,虚线为低温时的充电曲线。采取不同充电率,蓄电池的端电压随时间变化的规律由图2所示(4节酸性蓄电池)。蓄电池的充电率不同,蓄电池的温度不同,蓄电池的种类不同,充电时到达出气点G(指蓄电池充电后期出现气体的时刻)的时间就不同,温度越高,充电率越大,出气点就越提前,并且出气点电压曲线的斜率值变得很大。充电过程一般检测蓄电池两端的端电压,作为控制反馈信息。在本文的模糊控制中,考虑选用端电压误差和端电压误差的变化率作为输入变量确定充电阶段,进行模糊推理,决定控制量。人手工控制充电理想的过程是,在出气点以前,根据端电压误差和端电压误差的变化率的变化情况,随时增大和减小蓄电池的充电电流,保证尽可能地以较大的电流充,同时又不会有太大的电压误差变化率,即避免很快到达出气点,到达出气点后,采取较小的充电率充电,直至电量充足。这样就突破了三级充电的曲线,实时调整充电电流,缩短了充电时间,又延长了蓄电池使用寿命,这恰好符合理想的蓄电池充电规律。这些为蓄电池充电的模糊控制提供了依据。

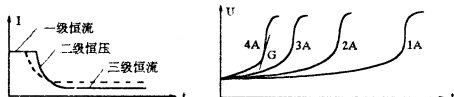


图1 蓄电池的三级充电法

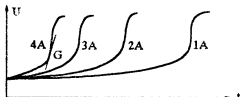


图2 不同充电率的电压曲线

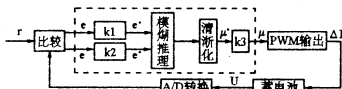


图3 蓄电池模糊控制的结构框图

## 3 蓄电池快速充电模糊控制器的组成

模糊自动控制是以模糊集合论、模糊语言变量及模糊逻辑推理为基础的一种计算机数字控制。蓄电池

模糊控制的结构框图可由图3表示,它的核心部分为模糊控制器,如图中虚线框中部分所示。

模糊控制的过程是这样的,实时地采集蓄电池的端电压,计算出其误差和误差变化率,判断其所处的状态,进行模糊推理,再经清晰化,获得控制量 $\Delta I$ ,驱动PWM电路(周期性信号的定频调宽电路),调整外加充电电源的电压脉冲宽度,即可调整充电电流的大小。其设计过程可分为三部分:输入信号的变换、模糊推理、控制量的清晰化。

### 3.1 模糊化

在模糊控制中,输入变量的数字量需要转化为模糊量,才能用模糊控制算法。输入变量为端电压的误差 $e$ 和它的误差变化率 $\dot{e}$ , $\Delta I$ 作为控制的输出变量。由于误差和误差变化率实际变化范围很大,需将它们转换到固定区域内。端电压在充电过程中总处于上升阶段,因此 $e$ 的变化范围只选正半部分,另外,误差变化率 $\dot{e}$ 在快到达出气点时数值很大,模糊语言变量再增加PVL(正向非常大的变化)。因此 $e$ 的语言变量选PS、PM、PL、PVL, $\dot{e}$ 的语言变量选NS、ZE、PS、PM、PL、PVL。如果 $e$ 的精确值 $x$ 实际范围是 $[a, b]$ ,通过公式 $y = 6(x-a)/(b-a)$ ,转换为 $[0, 6]$ 间变化的量 $y$ 。 $\dot{e}$ 的精确值 $x$ 实际范围是 $[c, d]$ ,通过公式 $y = 12(x-(c+d)/2)/(d-c)$ 转化为 $[-6, +6]$ 间变化的量 $y$ 。

在建立隶属函数时,需要通过收集各种蓄电池的多样充电曲线,来确定蓄电池端电压误差和误差变化率参考变化范围,进而确定它们的隶属函数。要分别确定出在出气点前后 $e$ 和 $\dot{e}$ 合理的变化范围,初步确定后,还要通过实验来进一步地调整参数。例如对于4节的蓄电池电压变化情况作以下的考虑:

(1)在充电初期,端电压通常为12V左右。

(2)从开始充电到出气点前,端电压变化范围为12~13.5V。

(3)当端电压接近于13.5~14.5V。误差变化率突然变大时,出气现象发生。

(4)要求的最终电压约为14.5~15.2V。

基于以上的考虑,可粗略地画出此时端电压误差的隶属函数曲线,如图4所示。

端电压误差变化率的隶属函数的建立与上同理,只是还需要做大量的实验,考察各种充电曲线从中总结出合理的变化规律。

### 3.2 模糊推理

模糊推理是将固定的控制规则的前件与后件中语言变量所对应的模糊集进行相应的模糊运算。控制规则来源于手控的操作经验和已取得的充电实验数据。为获得较理想的模糊控制量,还需要在模糊控制器设计完成后的实验情况作进一步的参数调整。由操作经验和实验数据可初步确定出24条控制规则如图4所示。

U \ e	e	PS	PM	PL	PVL
e					
NS		PL	PM	ZE	NM
ZE		PL	PS	NS	NL
PS		PL	ZE	NM	NL
PM		PL	NS	NL	NL
PL		PM	NM	NL	NL
PVL		PS	NL	NL	NL

这里控制规则可用以下条件语句表示:

if  $E=A_i$  and  $EC=B_j$ , then  $U=C_{ij}$

其中  $A_i$  表示控制系统端电压的误差值,  $B_j$  表示与  $A_i$  对应的误差变化率, 而  $C_{ij}$  则表示合成规则计算出的控制量  $\Delta I$ 。

对于某一条规则的激发强度  $C_{fr}$  (firing strength)

$$C_{fr} = m_A(x_0) \wedge m_B(y_0) \quad x_0 \in A, y_0 \in B \quad (1)$$

其中  $x_0, y_0$  分别为  $e$  和  $\dot{e}$  的精确值,  $m_A, m_B$  分别为  $e$  和  $\dot{e}$  的隶属函数。

这里模糊推理采用 Mamdani-minimum 操作规则

$$m'_{c_r}(z) = C_{fr} \wedge m_{c_r}(z) \quad z \in C \quad (2)$$

其中  $z$  指的是电流变化量  $\Delta I$ ,  $m_{c_r}$  是  $\Delta I$  的隶属函数,  $m'_{c_r}$  是此条规则对整个控制规则的贡献。那么总的控制规则的隶属函数为

$$m_c(z) = \bigvee_{r=1}^n m'_{c_r}(z) \quad z \in C \quad (3)$$

输出变量  $\Delta I$  的模糊量根据模糊推理合成规则可得

$$C = (A \times B) \cdot R \quad (4)$$

### 3.3 清晰化

模糊控制器的输出是一个模糊量, 这个模糊量不能直接控制被控对象, 还必须对它进行清晰化处理。常用的清晰化方法有三种: (1) 最大隶属度法, 即选取模糊子集中隶属度最大的元素作为控制量; (2) 加权平均判决法, 以隶属度为系数求出加权平均值, 以此作为执行量; (3) 取中位数法 (COA), 即选取求模糊子集的隶属函数曲线和横坐标所围成区域的面积平分分为两部分的数, 作为输出的控制量。这里选用第二种方法, 它能比较充分地利用了模糊子集提供的信息量。

$$u^* = \frac{\sum_{i=1}^n \mu(u_i) \cdot u_i}{\sum_{i=1}^n \mu(u_i)} \quad (5)$$

求出  $u^*$  后再乘以比例因子, 即可得到作用于被控对象的实际控制量。

### 4 总结

模糊控制器设计完成后, 可将软件固化到单片机的 EPROM 中, 软件的任务是根据输入变量值, 通过查询控制表, 得到控制量, 然后驱动脉宽可调电路 PWM, 改变充电电流的脉冲宽度, 使充电电流沿着合适的趋势变化, 使充电过程中尽可能地以较高的充电

率充电, 达到较理想的蓄电池充电状态。

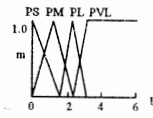


图4 电压误差的隶属函数

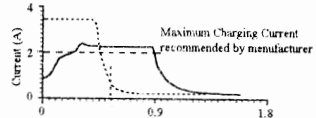


图5 蓄电池充电的模糊控制和常规控制比较

模糊控制设计的主要环节在于误差和误差变化率隶属函数的确立、控制规则的确定及控制表的响应数值。这需要观察各种蓄电池在不同的充电率、不同温度下的充电曲线, 从而总结出较好的变量变化范围及对应的隶属度。

蓄电池模糊控制系统的优点在于它模仿手工充电操作经验, 有很大的适应性, 符合蓄电池快速充电的规律。由图 5 所示的  $I-t$  曲线 (实线为模糊控制的充电曲线, 虚线为常规的三级充电), 可看出模糊控制的充电电流比较接近于厂家所允许的充电电流, 提高了充电效率, 延长了蓄电池使用寿命。

另外蓄电池充电节数有时是不确定的, 这也成为模糊控制应考虑的一个因素。节数不同, 各变量的实际变化范围及隶属函数都发生变化。作者认为, 可将电池的节数也作为一个模糊变量来考虑, 将节数分成几段, 每段分别确定各自的隶属函数和控制表。控制时, 每次键盘输入一个节数变量值, 或者测量蓄电池组的总电压数值, 将节数或总电压数值模糊化到相应的段, 调用相应的控制表, 如同“点菜谱”那样较好地实现不同节数蓄电池的充电。

由于蓄电池快速充电的模糊控制仍处于研究阶段, 还有许多问题值得研究, 要想获得一个高效的蓄电池模糊控制系统, 需要依靠大量的实验, 总结出模糊控制蓄电池充放电时的规律, 模糊控制理论和实践的结合同才能将蓄电池快速充电技术提高到一个新的高度。

本课题研究期间得到了西南交大模糊控制中心徐扬教授的悉心指导, 在此表示衷心的感谢。

### 参考文献

- 1 李士勇. 模糊控制和智能控制理论与应用. 哈尔滨工业大学出版社, 1990. 12, 78~112 页
- 2 陈永义. 模糊控制技术及其实例. 北京师范大学出版社, 1993. 5, 140~146 页
- 3 姜绍信. 铅酸蓄电池快速充电. 天津科学技术出版社, 1984. 1
- 4 Y. C. LIANG and T. K. NG. Design of battery charging system with fuzzy logic controller. Int. J. Electron(UK), Vol. 75, No1
- 5 张福民, 徐红兵, 孙增圻, 孙富春. 飞航导弹模糊滑动模糊控制系统. 中国智能自动化学术会议暨智能自动化专业委员会成立大会论文集, 1995. 8, 484

# 嵌入式计算机在提升机状态监控中的应用

任子晖 马小平

(中国矿业大学信息与电气工程学院 徐州 221008)

**摘 要** 本文介绍一种以 PC/104 嵌入式计算机为核心的新型提升机状态监控系统,介绍了系统的组成、各部分的功能和特点。

**关键词** 嵌入式计算机 提升机 监控

## 1 引言

矿井提升机是采矿行业的大型机电生产设备,其运行的安全性、可靠性和经济性,对矿井生产及安全有着重要影响。为满足采矿业生产的不断发展,各国提升机生产制造企业都在努力提高其产品的技术装备水平,尤其是在对提升机实际运行中的设备状态进行实时监测和保护方面,一直是提升机技术发展中的关键课题。

为了满足我国煤矿生产安全的需要,针对当前的提升机运行状态监护方面存在的不足,根据我们多年在提升机安全技术方面的研究成果,并在学习和总结国内外有关先进技术的基础上,利用先进的计算机技术,研制开发了新一代具有先进技术水平提升机状态监测与保护系统。

## 2 系统的组成与原理

系统是由 PC/104 嵌入式计算机系统、油压传感器、光电编码器、自行研制的盘式制动器贴闸信号传感器、输入电路、控制继电器以及显示器和打印机组成。系统原理图如图 1 所示。

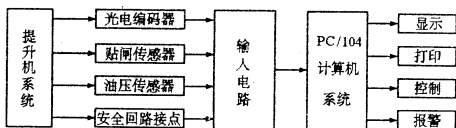


图1 系统原理框图

### 2.1 嵌入式计算机

嵌入式计算机是八十年代末出现的一种新型的工业控制计算机,采用自层叠结构,因其总线之间的互连使用了 104 个信号线,因此嵌入式计算机又称为 PC/104 计算机。它的主要技术特点是:

- 1) 可靠性高,产品稳定,使用寿命长。
- 2) 使用 CMOS 器件,功耗低,不存在散热问题。
- 3) 结构紧凑,各种模块尺寸均为  $90 \times 96\text{mm}$ ,配置灵活,便于功能扩展。
- 4) 与 PC/AT 等完全兼容,开发软件工具丰富,极大缩短了产品的开发时间,节约开发费用。

鉴于以上因素,并考虑到将来进一步扩展功能,我们选用盛博公司的 PC/104 产品组成本系统的核心,

以三块模块,即 CPU 模块、扩展功能模块、数据采集与控制模块,构成满足我们对提升机运行工况进行监测与保护需要的工业控制计算机系统。下面简单介绍各模块的功能:

CPU 模块提供了 PC/AT 母板所具有的全部功能和一些附加卡的功能,具有与 PC/AT 和 MS-DOS 全兼容的标准软、硬件资源;支持固态硬盘(Solid State Disk),以替代磁盘驱动器,通过使用 SSD 支持软件,开发的应用程序可不做任何修改即可从 SSD 调出运行;另外,应用软件可选择使用板上的看门狗定时器。以上两项设置进一步提高了系统的可靠性。用固态硬盘取代软盘驱动器,还可以避免计算机病毒的侵入。板上 CPU 为 386SX, 25MHz, 内存 1M。

扩展功能模块的作用是:提供串行接口、打印接口及软盘和硬盘接口(IDE),在板有 512KB 显示存储器,支持  $1024 \times 768$  分辨率、16 种颜色以及 VGA 等多种显示方式,与通用的 Trident 控制器全兼容。

数据采集与控制模块提供 24 路开关量输入/输出、三个 16 位减法计数器和 8 路模拟量输入功能。该模块由一片可编程外设接口 71055 提供三组各 8 路的 TTL CMOS 数字 I/O,其中 8 路可独立地定义为输入或输出,我们将两组口定义为输入口,用于监测每个制动器的贴合状态,另一组定义为输出口,用于报警和控制;板上的 71054 提供三个 16 位减法计数器,它有 5 种工作方式,可由软件设定,我们将其设定为软件触发计数方式,用于统计光电编码器输出的脉冲数;模拟量输入口用于制动油压大小的测量,板上的 A/D 转换器为 AD574,转换精度 12 位,转换速度  $20\mu\text{s}$ ,并具有输入过压保护功能。

### 2.2 输入电路

自行设计的输入电路完成对各种输入信号的预处理,将外部信号与 PC/104 隔离,保障系统的正常工作和 PC/104 模块的安全。贴闸信号隔离采用 TLP521 四光耦,速度脉冲信号隔离采用高速光耦 6N139;同时,由于油压传感器输出的信号为毫伏级,输入电路需对该信号进行放大,然后才能进行 A/D 转换。由于放大倍数较高(约 800 倍)。我们选用高精度测量放大器 AD521,它工作稳定,增益容易调整。为了使放大器工作稳定,所有电阻和电位器都选用精密电阻和精密电

位器。

油压信号经压力传感器输入 AD521, 经放大后再送 A/D 转换。制动器的贴闸状态是通过安装在每副闸上的贴闸传感器反映的, 经光电隔离后送入数据采集与控制模块的开关量输入口。安装在与滚筒(或深度指示器)同轴的增量式光电编码器用于测量提升高度和提升速度, 光电编码器每转发出 1024 个脉冲, 因此, 提升高度和速度测量的精度是相当高的。为避免干扰, 光电编码器选用集电极开路输出型。利用光电编码器提供的相位相差 90° 的两路脉冲信号, 可以判断出提升方向。

### 3.2 报警与控制输出

报警电路由 MC1413 和蜂鸣器组成, 用于减速点提示, 过速、过卷及运行方向错误报警提示。

由 PC/104 数字输出口直接驱动固态继电器(SSR), 将固态继电器的接点接入控制回路即可实现预定的控制功能。

系统应用软件用 Turbo C++ 语言编写, 在 UC DOS 汉字操作系统下运行。

### 3 系统的功能

提升机状态监护系统主要实现盘式制动器状态监测和提升机运行参数实时监测两方面功能。

#### 3.1 盘式制动器状态监测:

1) 实现对盘式制动器的总制动正压力和各闸瓦单项制动正压力的在线监测以及制动和松闸过程中的正压力变化显示。

2) 实现对各制动油缸阻力、油压的在线监测及显示。

3) 实现对各制动闸开合状态的监测、显示, 当闸瓦间隙磨损超限时报警。

4) 实现对各闸瓦空动时间, 二级制动时间的监测、显示。

#### 3.2 提升机运行参数实时监测

1) 实现对运行参数合理计算和设计速度图的构成及屏幕显示。

2) 通过彩显屏幕在已构成的设计速度图上, 对提升过程中实际运行速度参数进行实时跟踪显示, 提升司机可根据设计速度图和实际运行跟踪速度图的变化、对比, 对提升机实际运行参数进行操作控制, 以保证实际运行参数按设计参数变化, 保证提升机的合理运行及安全。

3) 具有屏幕图形深度指示, 运动参数显示, 过卷显示, 提升次数显示, 减速点提示, 过速、过卷报警。

4) 对提升机实际运行速度图及重要参数按时间顺序进行存储记忆, 并可通过屏幕和打印机对历史曲线及参数进行再现, 以便进行事故分析和数据查询。

### 4 系统的主要特点

1) 抗干扰能力强, 可靠性高, 实时性好。

2) 通过彩显对监测参数、汉字图表、性能曲线进行实时跟踪显示。

3) 可对事故过程和重要数据进行记录保存和再现。

4) 可通过打印机将记录参数和文字图表进行打印输出。

5) 中文下拉式菜单和在线帮助, 便于操作。

### 5 结束语

采用 PC/104 嵌入式计算机构成本系统, 由于它与 PC/AT 兼容, 软件丰富, 缩短了开发周期。本系统已在邢台矿务局、大同矿务局和徐州矿务局的多个矿投入使用, 效果良好。我们将利用 PC/104 便于扩展的特性, 进一步完善和扩展系统功能, 增加监测信号的数量, 如: 提升打点信号和轴承温度信号等, 并对保护功能加以扩充。

## 玻璃钢罐管道纤维缠绕自动控制系统的开发

曹二堂 刘玉林 田德福

(河北经贸大学计算机中心 石家庄 050061)

**摘 要** 纤维缠绕过程中纱片搭接和导丝头的返向处理是两个关键性问题, 本文阐述了微机控制系统及采用位置跟踪和虚拟长度解决这两个问题的方法。在实际生产中使用表明该系统性能稳定可靠。

**关键词** 玻璃钢 纤维缠绕 工控机

玻璃钢产品具有耐腐蚀、强度高等优点, 广泛用于石油、化工及输水工程等。纤维缠绕在玻璃钢罐、管道生产中是一种重要成型工艺。以自动控制设备对缠绕过程进行自动控制, 可以实现产品连续成型, 大幅度降

低生产成本, 提高产品质量, 改善劳动环境。我们根据纤维缠绕规律, 结合自动控制理论和计算机技术研制出性能优越、自动化程度高的 CR-4000 型玻璃纤维缠绕自动控制系统。

### 1 缠绕原理

纤维缠绕是指将浸过树脂胶液的纤维纱或带, 按



照一定规律及设计要求缠绕在芯模上,待固化后脱模,根据导丝头和芯模相对运动之间的定量关系,缠绕线型可以分为两类:环向缠绕和螺旋缠绕两类。如图1、图2所示,本文只讨论螺旋缠绕。根据缠绕规律,缠绕过程要满足两点要求:第一,导丝头的移动速度和芯模的转速按照一定的比例,以实现给定的缠绕角;第二,导丝头每个单程内芯模要多转或少转一个角度,两条“标准”线搭接完好,既不重叠也不离缝,均匀连续布满芯模表面。

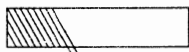


图1 环向缠绕



图2 螺旋缠绕

芯模绕自轴转动,导丝头按特定速度沿芯模轴线方向往复运动。于是在芯模的筒身和封头上便实现了螺旋缠绕,其缠绕角一般为 $12^{\circ}-70^{\circ}$ 。

螺旋缠绕中,纤维在芯模表面的排布方式,即线型,直接和纤维在极孔圆周上切点位置有关,不同切点下芯模转角与线型的关系如下:

单切点时

$$\theta_1 = (1+N) \times 360^{\circ} \pm \Delta\theta$$

两切点时

$$\theta_2 = (1/2+N) \times 360^{\circ} \pm \Delta\theta/2$$

三切点时

$$\theta_3 = (1/3+N) \times 360^{\circ} \pm \Delta\theta/3$$

n 切点时

$$\theta_n = (K/n+N) \times 360^{\circ} \pm \Delta\theta/n$$

式中: $\Delta\theta$ 为使位置相邻的两个切点所对应的纱片在筒身段错开一个纱片宽度而设定的一个微小增量,其值由纱片宽度决定。

$\theta_n$ 为导丝头往返一次,芯模的转角

n为切点数

N为正整数

K为使 $K/n$ 为最简分数的取值,即 $n \geq 3$ 时各切点位置排列顺序和时序并不一致。

从上述关系可以看出,切点数决定了缠绕线型,要实现预定的缠绕线型,需要芯模转动与导丝头的运动在整个缠绕过程中按照缠绕规律严格配合。利用微机控制系统,可以自动检测缠绕状态,并控制芯模的转动和导丝头的运动。

## 2 系统组成

系统以IPC-610型工业控制计算机为控制核心,采用无级变速调速器,不仅可以减小缠绕误差,还可以在一定范围内根据需要调节缠绕速度。系统组成如图3所示。

由控制台发出运行/停止、自动/手动、高速/低速、正转/反转、系统加速/减速、系统复位等信号通过数字通道送入IPC-610工业控制计算机。D/A通道1输出电压信号1控制直流调速系统以控制导丝头速度;

D/A通道2输出电压信号2控制交流调速系统以控制芯模的转速;两个计数器分别记录导丝头移动的距离和芯模转过的角度。

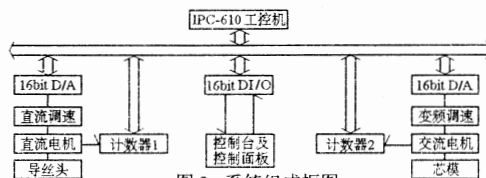


图3 系统组成框图

## 3 控制原理及其实现

### 3.1 控制原理

根据芯模直径D、长度L、缠绕角 $\alpha$ 、切点数n计算出导丝头和芯模的速比,由计算机通过D/A通道1和D/A通道2分别向直流调速系统和交流调速系统发出转速控制信号,控制导丝头和芯模的相对运动;同时根据两个计数器返回的信号修正导丝头和芯模的速度,以实现预定的线型。计数器2另一个作用是检测导丝头的位置,当到达芯模长度时,计算机通过数字通道向直流调速系统发出导丝头反向信号。

### 3.2 位置跟踪

导丝头在运动过程中由于各种原因造成速度的变化,必然造成导丝头和芯模的相对位的变化,从而造成线型的异变,使纱片搭接不好,因此,在控制过程中采用位置跟踪,以确保纱片搭接良好。位置跟踪采用PID算法,其方程如下。

$$c(n) = K_p e(n) + K_i \sum_{j=0}^n e(j) + K_d [\Delta e(n)]$$

其中: $c(n)$ 是全输出量

$e(n)$ 是第n个采样时刻的偏差值

$[\Delta e(n)] = e(n) - e(n-1)$ 是本次测得的偏差值与上次测得的偏差值的差值

### 3.3 导丝头返向处理

根据缠绕原理理想化的线型如图4所示。导丝头到达芯模长度时应立即以原速度返回,由于电机拖动系统的惯性等因素,导丝头不可能以原速度立即返回。且由于张力的作用,纤维会在芯模上打滑。因此,在芯模的两端不能按照理想线型缠绕,这样给整个缠绕过程带来困难,为了保证在芯模筒身区间的线型不变,我们采用虚拟长度法解决了这个问题,具体作法如下:在参数计算中,将芯模的实际长度附加一个虚拟量,然后,按预定线型计算控制参数,而实际运行过程中导丝头只在筒身的实际长度区间运动,导丝头接近芯模两端时发出返向信号。由于导丝头的惯性,发出返向信号后导丝头开始减速→停止→返向加速,并需要一定的时间。使这一段时间正好是缠绕虚拟长度所需的时间。这样可以使导丝头在芯模端头很快到达预定线型位置,从而保证了芯模筒身区间的线型不变。采用虚拟长度后的线型如图5所示。



图4 理想线型展开图

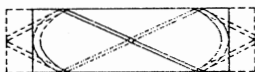


图5 实际线型展开图

#### 4 软件设计

本系统采用C语言编成,具有良好的用户界面。用户只需输入基本缠绕参数,系统根据这些参数确定相应的线型并等待运行,用户在控制台发布运行命令,并可以随时调整系统运行速度。

##### 4.1 参数输入和计算

输入参数包括芯模几何参数:长度 $L$ ,直径 $D$ ,及工艺参数:缠绕角 $\alpha$ ,纱片宽度 $b$ ,切点数 $n$ 。

螺旋缠绕中,一个完整的循环后缠绕纱片必须回到起始点并错过一个纱片,这样,才能做到连续两个完整循环缠绕的纱片在芯模表面相接。实际输入参数不一定符合线型设计要求,因此,需要按选定线型,对缠绕角作适当调整。另外,螺距必须是纱片宽度的整数

倍,纱片宽度也需要作适当的调整。才能计算缠绕速度。限于篇幅,其算法和程序在此省略。

##### 4.2 控制流程图

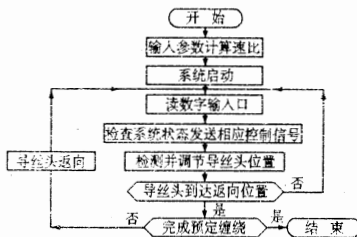


图6 控制流程图

#### 5 结束语

该项研究是河北省科委资助下完成的,其成果已在1995年4月用于河北中意玻璃钢有限公司玻璃钢罐管道生产线中。经过一年多的运行表明,该系统性能稳定可靠。

## 塔钟多媒体自动报时系统

许伟 余立建 蔡淮

(西南交通大学计算机学院 成都 610031)

**摘要** 将多媒体计算机技术应用于塔钟上,使其成为有高保真、多功能、全自动化的报时系统。

**关键词** 多媒体 塔钟 报时

多媒体计算机技术汇集了计算机体系结构、计算机系统软件、视频音频信号的获取、处理、特技以及显示输出等技术。把多媒体计算机技术应用到塔钟自动报时系统上,使其不仅具有平时播出高保真、大功率的音乐和钟声,还可以用于如公安机关在广场(如火车站)召开公捕公判大会,人防部门发布防空紧急警报等一些特殊场合。本系统每天自动开、关计算机,自动开、关前后级功率放大器,自动校时。本系统平时播出的音乐和钟声都是经过数字化采样、处理并储存在计算机硬盘上。所以与其它报时系统相比,具有多用途、智能化、数字化及报时钟声音质好、声音宏亮等特点。

##### 1 系统总体结构

整个系统的组成及结构如图1所示。

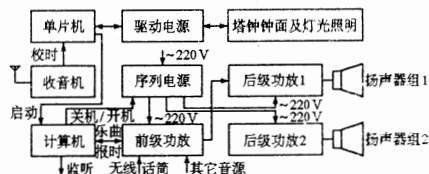


图1 系统的组成及结构

该系统工作原理:每天早上5:55由单片机送来的“启动”信号将计算机电源接通,系统软件自动装入

内存开始运行,5:57计算机向序列电源发出“开机”信号,序列电源按照前级功放→后级功放1→后级功放2的秩序将它们的电源一一接通。6:00前计算机读硬盘数据送声霸卡播送乐曲《东方红》,乐曲完毕立即连续播送报时钟声6次,6:00整打最后一响。声霸卡播送的“乐曲、报时”信号,经过前级功放预放大输出左、右两个声道分别送后级功放1和2。后级功放1和2采用桥接方式,驱动后面总负载阻抗为8ohm的扬声器组1和2,使每组扬声器可以输出功率达1000W的声音信号。6:02计算机向序列电源发出“关机”信号,序列电源按照后级功放2→后级功放1→前级功放的秩序将它们的电源一一关闭。以后系统软件继续运行,6:55由单片机送来的“启动”信号送入计算机,由系统软件检测到,立即将计算机系统时钟校正到6:55,并向序列电源发出“开机”信号,重复前面的过程。该系统每一小时播送报时钟声,12:00和18:00还分别播送乐曲《五星红旗》和《一路平安》。22:00播送完报时钟声后在22:02关闭功放,22:05计算机自动关机。收音机接收中央人民广播电台的报时信号,送入单片机进行处理,再自动校正塔钟系统的时钟。单片机输出步进电机控制信号给驱动电源,经其放大以后给塔钟钟面背后的步进电机,控制其带动钟面指针的转动。另外钟面背后装有传感器,钟面指针的位置可以由这

些传感器获得,将其转化为电信号输入给单片机。单片机可以显示北京时间和塔钟钟面时间,并且都可以进行人工设置。塔钟钟面的灯光由单片机控制,白天关,晚上开。

## 2 计算机 I/O 接口卡

该系统计算机 I/O 接口卡主要电路如图 2 所示。

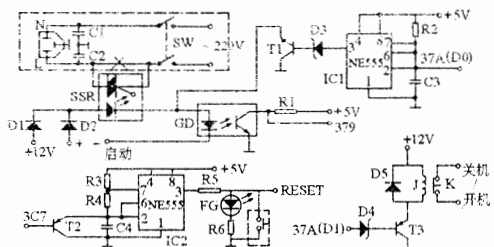


图2 计算机I/O接口卡电路图

图中虚线框内为原计算机电源交流 220V 输入端部分电路,在标有“×”处将电源线断开,引出两条线接固态继电器 SSR 的输出端,平时 SW 处于接通状态。

SSR 输出端不导通时,计算机没有交流 220V 输入,不工作。当塔钟控制仪送来“启动”信号,通过二极管 D2,加到 SSR 的输入端和光耦 GD 输入端,使 SSR 输出端导通,计算机电源获得交流 220V 输入,立即启动工作。由于“启动”信号有效时间较长(约 5 秒),计算机电源可以输出稳定的 +5V、+12V 等电源。+5V 加到单稳触发器 IC1 上,使其 3 脚输出低电平,通过稳压管 D3 接到三极管 T1 的基极;+12V 加到 D1、SSR 输入端、T1 发射极与集电极之间,使 D1、SSR 输入端、T1 导通,这样就维持了 SSR 输出导通,计算机电源继续工作。

当计算机要停止工作时,只需向 IC1 的 2 脚发送一个低电平,使其触发翻转,3 脚输出一定宽度的高电平,通过稳压管 D3 接到 T1 的基极,这样 T1 就截止, D1、SSR 输入端没有电流通过,SSR 输出端也关断,计算机电源没有交流 220V 输入,计算机停止工作。

“启动”信号也用于校时,当塔钟控制仪送来“启动”信号,通过二极管 D2,加到 SSR 的输入端和光耦 GD 输入端,使 GD 输入端导通,+5V 通过电阻 R1 使其输出端导通,输出一个低电平,系统软件通过查询检测到此信号以后,校正系统时钟。

图 2 左下脚是 IC2 和其外围元件组成的“看门狗”电路。由于计算机长时期的每天连续工作十几个小时,有时可能会出现“死机”的情况,此时需要进行系统复位——RESET。本系统采用自动复位方式,无需人为干预。IC2 实际上是一个多谐振荡器,振荡周期是  $tm1 + tm2 = (R3 + R4)C4 \times \ln 2 + R4C4 \times \ln 2$ ,其中  $tm1 = (R3 + R4)C4 \times \ln 2$  为第 3 脚输出高电平时间, $tm2 = R4C4 \times \ln 2$  为第 3 脚输出低电平时间, $tm1 \gg tm2$ ,  $tm1$  为 5 分钟左右, $tm2$  为 1 秒左右。平时系统软件定时对 T2 的基极送低电平,使其导通,C4 两端的电荷

被放掉,IC2 的第 2 脚始终为低电平,第 3 脚始终为高电平,RESET 无效,计算机正常工作。当出现“死机”时,系统软件也终止执行,无法对 T2 的基极送低电平,+5V 通过 R3、R4 向 C4 充电,当其电压大于三分之二(+5V)时,IC2 的第 3 脚输出低电平,RESET 有效,系统复位。由于第 3 脚输出低电平时间较短,系统复位后立即变为高电平,计算机重新启动,而第 3 脚输出高电平时间较长,能确保系统软件的重新装入。

图 2 右下脚为输出给序列电源“开机/关机”控制信号的电路。该序列电源控制开关接通时,电源不工作,断开时,电源工作,所以我们采用一个继电器 J 的常闭点 K。当需要打开前、后级功放时,系统软件通过稳压管 D4 向三极管 T3 的基极送一低电平,+12V 通过 J 加到 T3 的发射极和集电极,使其导通,电流通过 J,常闭点 K 断开,序列电源工作,将前级和后级功放顺序开启。当需要关闭前、后级功放时,系统软件通过稳压管 D4 向三极管 T3 的基极送一高电平,T3 截止,J 中没有电流通过,常闭点 K 导通,序列电源不工作,将前级和后级功放逆顺序关闭。

## 3 系统软件的设计

本系统软件采用 C 语言编程,界面及时钟显示采用 Windows 软件编制。用 C 语言调用 Windows 绘制的图形文件,读取单片机输出的“启动”信号,通过 I/O 接口卡控制序列电源及功放的开、关,完成校时,播放乐曲和报时功能,整个系统由主控软件、驱动软件、图形文件和声音文件构成,约占 8M 字节存储容量。为了提高系统运行速度,提高硬盘使用寿命,把常用的图形文件存入虚拟盘中。

整个系统正常工作时,由系统完成自动开机,自动装入配置文件及批处理文件,进入 C:\B\\_ADSP16 子目录,运行主控软件 HCB.EXE,自动校时,自动报时,自动关机,不需人工干预。每小时第 55 分时根据单片机给出的“启动”信号进行自动校时(在校时范围内),若因特殊情况超出校时范围(50 分~59 分),可进行人工校时,按键盘左上角“Esc”键,退出系统,键入 DOS 下修改日期命令“DATE”,根据格式输入正确日期或键入修改时钟命令“TIME”,屏幕显示当前时钟,用户可输入现在标准时钟,然后回车,系统回到 DOS 状态,再输入执行文件“HCB.EXE”,进入正常工作状态。

系统软件及所有图形文件,声音文件装在 C:\B\\_ADSP16 目录下,同时在 C:\HCB 目录下装有备份系统,需要时可将备份文件拷贝到工作目录 C:\B\\_ADSP16 下,再重新运行系统。

## 4 结束语

该系统具有自动报时,自动校时,自动开、关序列电源,开、关功放,开、关计算机的特点;根据不同日期及时间播放不同的前奏乐曲,人防部门发布防空紧急警报等功能。系统已在成都火车站广场投入运行,收到了良好的社会效果,得到了社会各界的好评。

# 多媒体技术在制导兵器武器系统总体技术中的应用

何衡湘 闵庆东 安景晖

(中国兵器工业第 209 所 成都 238 信箱 610041)

**摘 要** 如何应用多媒体技术为制导兵器的研制服务已引起了专家学者的广泛关注。本文根据多媒体技术的特点,概述了它在某型号制导兵器总体技术研究中的应用情况,对制导兵器行业开展多媒体技术应用的研究具有参考意义。

**关键词** 总体技术 多媒体 系统仿真

## 1 概述

制导兵器系统总体工作具有广泛的内涵,包括诸如项目组织、状态跟踪和阶段控制等管理工作及总体综合、系统分析、方案确定、兼容性分析等技术工作。高新技术的发展为武器系统总体工作提供了更有效、便捷的研制手段,制导兵器研制工作的向前发展一方面是各种成熟技术应用的结果,同时也推动了高新技术的发展。

八十年代发展起来的多媒体技术综合了计算机声音处理技术、计算机图形处理技术、计算机图像处理技术、计算机通信技术、计算机存储技术、集成电路技术等。这些技术的有机集结,对科技界、产业界、教育界、娱乐界乃至军事指挥等领域产生了强劲的冲击。

在制导兵器系统总体技术研究方面,国内部分专家敏锐地看到了多媒体技术的发展前景,提出了要在制导兵器系统总体工作中开展多媒体技术应用研究的建议。本文结合某型号制导兵器武器系统总体工作,从以下三方面讨论了多媒体技术的应用。

## 2 多媒体型号数据库及信息系统

多媒体武器系统型号数据库及信息系统为武器系统管理决策者和研制者随时提供直观的信息,为其作系统分析及管理决策提供依据。因此,开展该项研究工作首先受到人们的关注。

### 2.1 多媒体武器系统型号数据库

武器系统型号数据库包含了广泛的内容:战术技术指标、系统组成、精度分配、地形地貌、气象条件、载机特征、敌我制导兵器及有关武器装备情况、目标特性、武器系统和各分系统技术参数、目标和背景的散射及辐射特性、大气传输特性、电磁对抗和光电对抗数据、武器系统及分系统的可靠性和可维修性数据、后勤保障数据等。

应用关系型数据库管理系统的理论算法和模型来建立上述的数据库是件简单的事情,但该数据库中的数据不完全等同于多媒体数据,因为多媒体数据常常相伴而生,占用资源较多,要求设备档次较高,针对不同的对象拥有特定的处理方式和数据形式。

参照成功的理论与经验,采用通用的关系型数据和适宜于多媒体信息的、与超媒体相结合的数据模型,以现有成熟的关系型数据库系统为基础,通过一专门的通信模块联结在外壳型应用界面下,通过外壳实现对数据信息和多媒体信息的协调管理(概念模式如图 1 所示)。

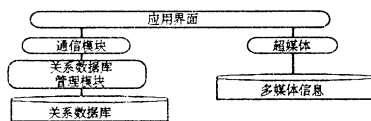


图 1 关系型数据库和超媒体的共存方式

### 2.2 多媒体信息系统

建立了多媒体数据库模型之后,即开始武器系统总体多媒体信息系统的设计工作。在设计中将应用相关技术手段解决如下主要问题。

#### 2.2.1 项目确立

在项目确立时,需要明确产品的应用领域,确定使用对象,为较好地达到目的,还需要进行综合媒体的个数、组成开发设计班子等工作。

#### 2.2.2 系统分析

在这一阶段主要确定系统需要依赖的硬件平台、软件平台、构造信息的结构、目录主题、层次结构、浏览顺序,确定交叉跳转关系等,这是设计过程中最重要的一个环节,也是最富创造力的阶段。

#### 2.2.3 数据的组织管理

本阶段要进一步完善数据库、理顺数据间相互关系,建立有效的数据结构,以便于在系统实现过程中快速、方便地调用各类数据。

#### 2.2.4 编程实现

经过以上数据采集管理和系统分析等过程后,即开始信息系统的编程实现过程。本阶段的任务大体是以下两个方面:一是用户界面设计,它要有实现菜单选择、鼠标驱动、肖像系统和信息提示等流行的人机交互接口功能;另一个是多媒体演示部分,它需要将检索到的多媒体信息按要求播放。目前,完成这一阶段的任务是靠用 C 语言和调用相关功能库函数编程的方式实现。

#### 2.2.5 试播、修改、生成运行版本

经过多次的试播、修改,即可生成正式运行版本,提交用户使用。

### 3 多媒体网络技术

多媒体技术对视频、音频、文本、图像的同时处理,加上网络技术取消了地域限制,使多媒体协同工作成为多媒体技术在武器系统总体工作中的具体应用。

#### 3.1 多媒体协同设计

多媒体协同设计起源于计算机协同工作(Computer Supported Cooperative Work)。在型号研制中主要用于问题讨论、设计方案共享、协同编辑设计文件等方面。目前拟采用分布式多媒体协同工作方案。

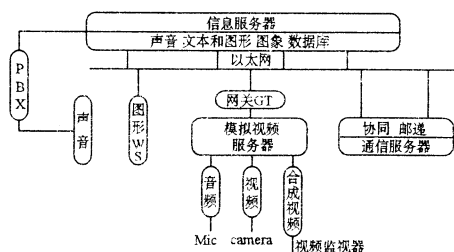


图2 协同工作体系结构图

图2所示为协同工作的简要体系结构。该系统使用三种分立的网络:以太网用于数据传输、PBX用于声音传输、宽带网用于视频传输。服务器由信息服务器和通信服务器组成。

#### 3.2 多媒体会议

为适应型号研制的需要,拟制定分布式多媒体会议系统方案,由会议管理、电子黑板、多媒体文档制作和多媒体数据库四部分组成。系统由如图3所示的三个层次组成:网络服务、网络控制中枢、高层界面。

会议管理模块是本系统的核心部分,起到系统总控和中转的作用,负责整个会议的管理、会议成员的安全性检查、数据帧的识别,处理会议间的意外情况等。电子黑板类似课堂上的讲台。文档制作要求节点多媒体化,节点间采用信息链结构,从而使整个系统有良好的导航性,并且文档还有深度和层次感。数据库用于记录与会人员的情况。

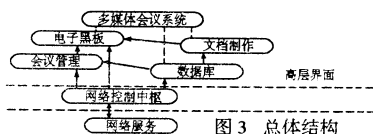


图3 总体结构

该系统选用成熟的NOVELL网络,运行于WINDOWS环境下。

### 4 多媒体作战模拟系统

借助于多媒体技术把实战的场面和音响效果制作成多媒体场景,经过编辑、裁剪、重叠方法,根据敌我双方武器性能和可能性作战的战场,制作成能模拟敌我对抗的场景。应用这些场景,指战员只需坐在模拟器前,随着荧屏展示的模拟实战场面,学习对战争的应付,模拟进行对敌军各种火力的抵御,掌握该型号武器的正确使用。

#### 4.1 系统构成

硬件:制导武器火控计算机;目标信号转换器。

软件:载机航路模型;制导武器弹体飞行动力学模型;导引头捕捉目标模型;命中目标圆概率模型;制导武器射击效果评判模型;目标特性模型库等。

#### 4.2 功能介绍

- 1)能够集中显示飞行大队协调的命令和过程;
- 2)能够集中显示战斗全过程;
- 3)能显示各环节的操作情况及重要参数;
- 4)能显示制导武器对目标的攻击过程;
- 5)能显示制导武器的飞行过程;
- 6)能显示射击效果评判过程;
- 7)要求操作简单、显示直观。

#### 4.3 操作流程

操作流程见图4。

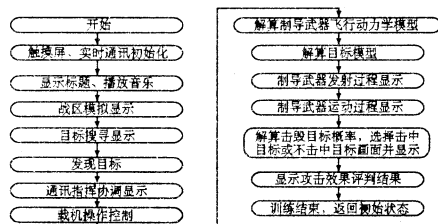


图4 多媒体作战模拟操作流程图

### 5 结束语

多媒体技术在制导兵器武器系统总体工作中的应用远不只以上几点。在平常的工作与研讨中,还可应用多媒体技术将计算结果图像化,测试结果可视化等。本文所述仅为开展多媒体技术的应用研究作一探索,其包含的理论支持和系统模型还有待更进一步的研究。

#### 参考文献

- 1 军事对抗模拟系统中多媒体技术的应用。多媒体世界。月刊,1996年3期
- 2 陈鸣编著。多媒体技术。人民邮电出版社,1994年2月

欢迎订阅一九九七年《计算机应用研究》杂志(双月刊)

● 邮局订阅代号: 62—68 (国内)  
BM—4408(国外)

● 1997年全年订价: ¥24(国内)  
\$15(国外)

# C 语言实现 EGA/VGA 图形方式下特殊画面的设计

王天芹 罗文保

(贵州安顺双阳飞机制造厂科技处计算机室 561018)

王天泽

(河南大学 开封市 475001)

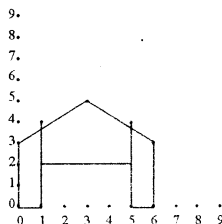
**摘 要** 本文简述在 EGA/VGA 图形方式下不规则对象的存储及处理方法,并给出一示例程序。

在 EGA/VGA 图形方式下,对于直线、圆、弧这样的对象,可以由算法来生成,并可对其进行旋转、平移、比例变换等操作。而对于字符或现实世界中存在的其他不规则实体,则没有算法可寻。在计算机中要对这些对象进行处理,首要任务是解决它们在机内如何存储的问题。

目前,大多数系统都采用点阵信息存储字符,这种方法对一般应用来说已经足够,但若要对这些对象进行变换处理以产生特殊显示效果,用笔画字形库方式存储更为方便。

所谓笔画字形库实际上就是直线数据库,每一元素存储直线的两个端点信息,代表一个笔画,点的坐标是相对于对象原点的偏移量。这样对实体的处理就转移为对其每一笔画直线进行处理。

这里,笔画字形库设计举例如下:



对象左下角为原点,相应笔画字形表为:

```
{ 0x5330, 0x3000, 0x0001, 0x0141, 0x5336, 0x3606,
  0x0506, 0x0545, 0x2125, 0xFFFF }
```

取出表中每一元素,将其分解为直线的两个端点坐标,对每一条直线进行处理,从而实现对整个对象的处理,使复杂问题简单化。

通过自动定义字形表,可设计出各种不同效果的画面,达到特殊显示的目的。

附: TURBO C 源程序清单:

```
pic.c 程序清单:
#include <picdef.h>
#include <font.h>
#include <conio.h>
#include <graphics.h>
#include <math.h>
double xlate_x, xlate_y;
double rot_x, rot_y;
double xform_a, scale_f;
void playchar(int, double, double, int, int, int, int);
```

```
void xform(double *, double *);
void main(void)
{
    double x, y;
    int height, width, slant, color, p;
    int gdriver = DETECT, gmode;
    initgraph(&gdriver, &gmode, "...\\");
    x = 1000.0; y = 1000.0;
    rot_x = x; rot_y = y;
    xlate_x = xlate_y = 0;
    scale_f = 1;
    xform_a = 0;
    height = 3500;
    width = 1000;
    slant = 0;
    color = YELLOW;
    p = 0;
    playchar(p, x, y, height, width, slant, color);
    getch();
    closegraph();
}

void playchar(p, x, y, h, w, s, c)
int p, h, w, s, c;
double x, y;
{
    double x1, y1, x2, y2;
    double w_f, h_f;
    int x1_out, y1_out, x2_out, y2_out;
    int i = 0;
    while((font.whole.word = st[p][i++]) != 0xffff)
    {
        w_f = ((double)(w - 1) / 7.0);
        h_f = ((double)(h - 1) / 7.0);
        y1 = font.part.row1 * h_f;
        x1 = font.part.col1 * w_f;
        x1 += s * w_f * (y1 / (h - 1));
        y2 = font.part.row2 * h_f;
        x2 = font.part.col2 * w_f;
        x2 += s * w_f * (y2 / (h - 1));
        x1 += x;
        y1 += y;
        x2 += x;
        y2 += y;
        xform(&x1, &y1);
        xform(&x2, &y2);
        x1_out = x1 * XMETRIC;
        y1_out = YMAX - y1 * YMETRIC;
        x2_out = x2 * XMETRIC;
        y2_out = YMAX - y2 * YMETRIC;
        setcolor(c);
        line(x1_out, y1_out, x2_out, y2_out);
    }
}

void xform(x, y)
double *x, *y;
{
    double x_temp;
    x_temp = *x - rot_x;
    *y -= rot_y;
    *x = x_temp * cos(xform_a) - (*y) * sin(xform_a);
    *y = x_temp * sin(xform_a) + (*y) * cos(xform_a);
    *x += rot_x;
    *y += rot_y;
    *x += xlate_x;
```

# 实现.DBF文件结构动态调整的方法

雷 霆

(焦作工学院图书馆 454159)

**摘 要** 本文提出了实现.DBF文件结构动态调整的设计目标,给出设计思路和实现的方法。

**关键词** 文件结构 调整

## 1 引言

ioxpro 广泛应用于数据库管理领域,它所面临的绝大多数问题都是管理领域内的数据存储、检索、输入和输出等问题。通常的应用是在库文件结构一定的前提下实现管理功能。而在实际应用中有时要求库文件结构因需要能做相应调整才能满足工作需要。

开发通用管理软件,实现.DBF结构动态调整尤为显得重要。作者在设计通用工资管理软件中,实现对工资库文件结构动态调整,可以满足用户因需要可任意对工资栏目调整的功能,并且保证原有其它栏目的数据不受破坏及系统中对应的输入输出模块与之保持相应的调整。

## 2 设计目标

此模块要完成以下功能:

(1)在保护原有有效数据的前提下实现库结构的调整;

(2)实现数据输入(增加、修改等)部分相关界面的变动;

(3)实现运算部分相关程序的变动;

(4)实现数据输出(表格打印)部分相关程序的变动。

## 3 基本思想和设计思路

解决问题的基本思想是首先将存放实际数据的.DBF文件(例GRGZ.DBF)的结构用.DBF文件管理(例YFX.DBF、YKX.DBF、GZJG.DBF)。通过对存放库结构信息的YFX.DBF、YKX.DBF文件进行数据处理(增加、删除记录或修改字段)后,生成GZJG.DBF。再由GZJG.DBF的数据完成对GRGZ.DBF结构的调整。最后以程序生成程序的方式实现数据录入、运算、输出部分的变动。

其设计思路是:

(1)工资库的数据除了基本信息外可分为应发项和应扣项两类,为调整的方便建立两个相应的辅助库文件YFX.DBF和YKX.DBF,分别存放应发项和应扣项字段的信息。调整后的结构数据综合到GZJG.DBF中。此处关键在于YFX.DBF、YKX.DBF

中的字段名称要代码化。

(2)根据修改后的GZJG.DBF建立临时的数据文件TEMPGZ.DBF并将原数据拷贝到相应的数组MGZ中。

(3)用GZJG.DBF建立新的GRGZ.DBF,并将数据转录过来。

(4)用程序生成程序的方法修改界面提示程序。

(5)表格的输出也以程序通过运算生成表格的每一行,并将其作为记录存放入库文件中,然后调此库文件的记录输出打印表格即可。

## 4 结束语

用此方法设计的工资软件已在多个单位运行一年多,反映良好。若将此方法用于其它软件中也会收到好的效果。需要通用工资管理程序或软件者可同作者联系,也可附寄20元工本费向本刊函索。

附一 YFX.DBF结构及记录例

Record #	FIELD	NAME	FIELD	TYPE	FIELD	LEN	FIELD	DEC	FIELD
MC									
1	F1	N	9	2	基职工资				
2	F2	N	7	2	提10%				
3	F3	N	9	2	津贴				
4	F4	N	5	0	护龄				
5	F5	N	9	2	职务津贴				
6	F6	N	9	2	地方补				
7	F7	N	7	2	水电补				
8	F8	N	8	2	卫生津贴				
9	F9	N	7	2	医生费				
10	F10	N	7	2	女工费				
11	F11	N	4	2	电话费				
12	F12	N	7	2	退休金				
13	F13	N	7	2	其它				

附二 YKX.DBF结构及记录例

Record #	FIELD	NAME	FIELD	TYPE	FIELD	LEN	FIELD	DEC	FIELD
MC									
1	K1	N	8	2	房公积				
2	K2	N	7	2	房费				
3	K3	N	7	2	待业金				
4	K4	N	7	2	医保金				
5	K5	N	7	2	工资				
6	K6	N	8	2	定存				
7	K7	N	7	2	其它				
8	K8	N	9	2	收帐				

附三 程序清单

```
*****
* 程序名:GZ55.PRG
* 功 能:修改工资库结构
* 编 程:雷 霆
* 最后修改:94.12.15
*****
defi wind WIN1 from 3,8 to 24,70 title"修改工资项目"
SHADOW
acti wind WIN1
store "N" to NY
@2,5 say "修改工作将会调整原有数据,请慎重!"
@4,5 say "继续吗? (Y/N)"get NY PICT 'Y'
READ
```



```

if NY="N"
  close all
  deac wind WIN1
  return
endif
sele A
use YFX
go top
sele B
use YKX
go top
sele C
use GZJG
if recc()>5
  go6
  dele next 40
  pack
endif
sele D
use KSJG
if recc()>7
  go8
  dele next 40
  pack
endif
sele E
use FLJG
if recc()>7
  go8
  dele next 40
  pack
endif
*****
* 对 YFX.DBF,YKX.DBF 中的记录进行修改
*****
sele A
for I=1 to 20
  appe blan
endif
go top
brow fields field_mc,H="应发项名称",field_len,H="宽度",field_dec,H="小数位"save color,,W/GR,,,,,
dele all for FIELD_MC=" "
pack
go top
sele B
for I=1 to 20
  appe blan
endif
go top
brow fields field_mc,H="应扣项名称",field_len,H="宽度",field_dec,H="小数位"save color,,W/GR,,,,,
dele all for FIELD_MC=" "
pack
sele C
appe from YFX
appe from YKX
use
sele D
appe from YFX
appe from YKX
use
sele E
appe from YFX
appe from YKX
use
*****
* 以下根据保留字段建立临时的工资,科室,合计库(字段代
* 码是原来的)并将数据拷贝到相应的数组中
*****
crea TEMPGZ from GZJG
appe from GRGZ
if recc()<>0
  copy to MGZ
endif

```

```

crea TEMPKZ from KSJG
appe from KS
if recc()<>0
  copy to MKS
endif
crea TEMPHJ from FLJG
appe from FLHJ
if recc()<>0
  copy to MHJ
endif
use
*****
* 以下修改 YFX,YKX,GZJG 库中的字段代码,成为序号连
* 续的代码
*****
sele A
go top
do while NOT EOF()
  repl FIELD_NAME with "F"+ltrim(str<recc())
  skip
enddo
sele B
go top
do while NOT EOF()
  repl FIELD_NAME with "K"+ltrim(str<recc())
  skip
enddo
sele C
use GZJG
if recc()>5
  go 6
  dele next 40
  pack
endif
appe from YFX
appe from YKX
use
sele D
use KSJG
if recc()>7
  go 8
  dele next 40
  pack
endif
appe from YFX
appe from YKX
use
sele E
use FLJG
if recc()>7
  go 8
  dele next 40
  pack
endif
appe from YFX
appe from YKX
use
*****
* 用修改过的字段代码修改 GRGZ,KS,FLHJ 库结构,并将
* 数据转录过来
*****
create GRGZ from GZJG
if file('MGZ.DBF')=.t.
  appe from MGZ
endif
create KS from KSJG
if file('MKS.DBF')=.t.
  appe from MKS
endif
create FLHJ from FLJG
if file('MHJ.DBF')=.t.
  appe from MHJ
endif
close all
! ERASE TEMP*. *

```

```

1 erase M*.DBF
do gzgs
close all
return
proc GZGS
*****
* GZGS. PRG
* 用于产生工资结构调整后的屏幕格式文件 GZTS. PRG
*****
set talk off
set device to file GZTS. PRG
* 以下生成项目提示
@1,1 say "procedure TS0"
use YFX
go top
do while not eof()
    @1,1 say "@"+ltrim(str(recn()+2))+", 5 say ""+field
    _MC+""
    skip
enddo
use YKX
go top
do while not eof()
    @1,1 say "@"+ltrim(str(recn()+2))+", 30 say ""+
    field _MC+""
    skip
enddo
@1,1 say "return"
@1,1 say ""
* 以下生成项目提示(带数字)
@1,1 say "procedure TS00"
use YFX
stor recn() to m. FXS
go top
do while not eof()
    @1,1 say "@"+ltrim(str(recn()+2))+", 5 say ""+ltrim
    (str(recn()))+""+field _MC+""
    skip
enddo
use YKX
go top
do while not eof()
    @1,1 say "@"+ltrim(str(recn()+2))+", 30 say ""+
    ltrim(str(recn()+m. FXS))+""+field _MC+""
    skip
enddo
use YKX
go top
do while not eof()
    @1,1 say "@"+ltrim(str(recn()+2))+", 30 say ""+
    ltrim(str(recn()+m. FXS))+""+field _MC+""
    skip
enddo
@1,1 say "return"
@1,1 say ""
* 以下生成带 GET 的屏幕格式
@1,1 say "procedure TS1"
use YFX
go top
do while not eof()
    @1,1 say "@"+ltrim(str(recn()+2))+", 5 say ""+field
    _MC+""+get m. "+FIELD _NAME
    skip
enddo
use YKX
go top
do while not eof()
    @1,1 say "@"+ltrim(str(recn()+2))+", 30 say ""+
    field _MC+""+get m. "+FIELD _NAME
    skip
enddo
@1,1 say "return"
@1,1 say ""
* 以下生成带+STR()的屏幕格式

```

```

@1,1 say "procedure TS2"
use YFX
go top
do while not eof()
    @1,1 say "@"+ltrim(str(recn()+2))+", 5 say ""+field
    _MC+""+STR("+rtrim(FIELD _NAME)+", 7,
    2)""
    skip
enddo
use YKX
go top
do while not eof()
    @1,1 say "@"+ltrim(str(recn()+2))+", 30 say ""+
    field _MC+""+STR("+rtrim(FIELD _NAME)+",
    7, 2)""
    skip
enddo
@1,1 say "return"
@1,1 say ""
* 以下生成由 m. 变量转换字段变量程序
@1,1 say "procedure ZH0"
store '' to FX, FXH0, FXH1, KX, KXH0, KXH1
use YFX
go top
do while not eof()
    FX = FX + rtrim(FIELD _NAME) + 'with m. ' + rtrim
    (FIELD _NAME) + ', '
    FXH0 = FXH0 + 'm. ' + rtrim(FIELD _NAME) + ' + '
    FXH1 = FXH1 + rtrim(FIELD _NAME) + ' + '
    skip
enddo
use YKX
go top
do while not eof()
    KX = KX + rtrim(FIELD _NAME) + 'with m. ' + rtrim
    (FIELD _NAME) + ', '
    KXH0 = KXH0 + 'm. ' + rtrim(FIELD _NAME) + ' + '
    KXH1 = KXH1 + rtrim(FIELD _NAME) + ' + '
    skip
enddo
@1,1 say "repl XM with m. XM, KS with m. KS"
@1,1 say "repl "+subs(FX, 1, len(FX)-1)
@1,1 say "repl "+subs(KX, 1, len(KX)-1)
@1,1 say "repl YFHJ with "+subs(FXH0, 1, len(FXH0)-
1)
@1,1 say "repl YKHJ with "+subs(FXH0, 1, len(KXH0)-
1)
@1,1 say "repl SF with YFHJ-YKHJ"
@1,1 say ""
* 以下专用于转换 YFHJ, YKHJ, SF
@1,1 say "procedure ZH1"
@1,1 say "repl all YFHJ with "+subs(FXH1, 1, len(FXH1)-
1)
@1,1 say "repl all YKHJ with "+subs(KXH1, 1, len(KXH1)-
1)
@1,1 say "repl all SF with YFHJ-YKHJ"
@1,1 say ""
* 以下生成由 m. 变量转换 KS. DBF 字段变量程序
@1,1 say "procedure ZHKS"
@1,1 say "repl RS with m. KS, YFHJ with m. YFHJ, YKHJ
with m. YKHJ, SF with m. SF"
@1,1 say "repl "+subs(FX, 1, len(FX)-1)
@1,1 say "repl "+subs(KX, 1, len(KX)-1)
@1,1 say ""
* 以下生成由 m. 变量转换 FLHJ. DBF 字段变量程序
@1,1 say "procedure ZHFL"
@1,1 say "repl RS with m. RS, YFHJ with m. YFHJ, YKHJ
with m. YKHJ, SF with m. SF"
@1,1 say "repl "+subs(FX, 1, len(FX)-1)
@1,1 say "repl "+subs(KX, 1, len(KX)-1)
@1,1 say ""
set device to screen
return

```

# 用于继电保护装置的智能检测分析系统研究与实现

刘 青 文劲宇 刘 沛 程时杰

(华中理工大学 电力系 武汉 430074)

**摘 要** 本文设计与实现了一个用于检测分析变电站继电保护装置的智能化系统,详细介绍了该系统的知识表示、数据库、分析推理机制以及良好的解释机制与人机接口。该系统已完成对某变电站所有保护装置的检测试验,实践表明系统达到高度智能化与自动化,有效性强,可靠性高。

**关键词** 检测分析 智能系统

## 1 引言

继电保护装置是电力系统自动化的重要组成部分,是保证电力系统安全可靠运行、保障人身安全、防止设备损坏的主要措施之一。由于继电保护装置起着监视保护电力系统的作用,其自身的可靠性、选择性、快速性和灵敏性等性能至关重要。

长期装在电力系统内的继电器,其实际动作的机会非常少,如果电力系统发生故障,保护系统能否正确动作?动作时间是否符合标准?这就需要维护人员定期进行试验,人工产生模拟故障检测继电器的性能。因此,研究对安装在各变电站运行现场的继电器和保护回路进行检测的智能化方法与技术,是一项十分重要和有意义的工作。

近十年来,计算机在各领域的作用日益显著,随着数据库理论、人工智能理论和图形技术的日趋成熟,越来越多的智能软件应用于电力系统,国内外开发了许多用于电力系统的专家系统[1][2],而把人工智能技术用于继电保护检测分析的文献报导较少,本文将介绍作者研制的一个智能化面向变电站保护系统的检测分析系统。

## 2 智能系统功能要求

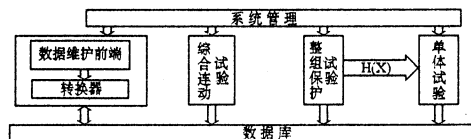


图1 系统功能图

由于电力系统输电网具有规模大、地理位置分布广的特点,继电保护维护人员必须按一定时间间隔在各变电所间流动试验,其检测时间有限而项目繁多,既要检测电压、电流、阻抗等各种类型继电器的动作特性,又要分析整套保护装置的动作情况,进行大量数据记录、报表分析,此项工作十分耗时与繁琐,并要求维

护人员做到准确无误,为了减轻工作人员劳动强度和减少工作失误,实现电网运行维护自动化与智能化,我们设计开发了一个用于继电保护装置的智能检测分析系统,该系统功能可分为如下几部分(见图1):

**系统管理:**提供人机界面,实现对数据维护和三种试验功能的调度与管理。

**数据维护:**通过数据维护前端可根据每一变电站实际情况往数据库中建立或撤销保护装置,并随保护运行方式的变化修改、插入或删除保护装置的各项参数。这些接近维护人员语言描述的信息经转换器转变为内部索引与编码形式存入数据库。

**综合连动试验:**从交流输入到断路器动作,测试保护回路动作正确与否。

**整组保护试验:**从交流输入到保护装置出口,模拟各种故障如单相、两相、三相短路,瞬时、永久、转换故障等,根据保护装置动作情况将故障点定位到继电器元件级,并记载到启发信息函数  $H(x)$  中,以便单体试验对此继电器作进一步分析测试。

**单体试验:**相当于各继电器的静态试验,可在保护屏上检测各继电器的动作值、返回值、动作时间、返回时间、电源波动影响、谐波影响、冲击试验、最小精工电流、最小精工电压、特性曲线等。

**数据库:**存储各变电所所有保护装置的各项参数和继电器初始状态、正确动作后应达到的状态等信息。

## 3 系统总体结构

该智能检测分析系统由一台工业 PC 机 Intel 386 故障模拟器、智能检测分析软件、通信电缆、接线插头组成,安装在保护屏内的被测装置通过接线插头与正在运行的电力系统断开,与检测系统形成回路,智能系统的总体结构如图2所示。

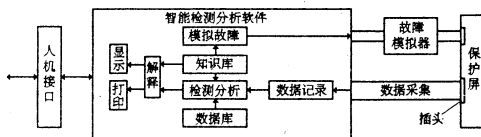


图2 检测分析系统总体结构图

#### 4 智能检测分析系统的知识表示

知识是智能系统的核心,本系统根据维护人员操作规程和分析结果时所运用的经验与规则,形成领域知识存储在知识库中,以便模拟故障模块和检测分析模块调用。合适的知识表示形式是系统有效工作的关键,本系统采用产生式规则和框架表示两种方式,二者都具有模块性强、清晰度高的优点。

故障模拟过程是以产生式规则描述的,例如:

R01 IF  $\times \times \times$ 继电器为过值型继电器

THEN 按  $vv$  步长递增  $uu$  输出

模拟量  $I_A, I_B, I_C$  或  $U_A, U_B, U_C$

P02 IF  $\times \times \times$ 继电器为欠值型继电器

THEN 按  $vv$  步长递减  $uu$  输出

故障知识和保护装置的信息则以框架结构描述,见表1~3。

表2 谐波子框架

表1 模拟故障主框架

框架名	主框架
模拟量号	模拟量号
模拟量幅值	模拟量幅值
模拟量初相角	模拟量初相角
谐波子框架	谐波子框架

框架名	子框架
1次谐波幅值	1次谐波幅值
1次谐波初相角	1次谐波初相角
2次谐波幅值	2次谐波幅值
...	...
7次谐波幅值	7次谐波幅值
7次谐波初相角	7次谐波初相角

表3 保护装置主框架

框架名	主框架
被测装置接线图	被测装置接线图
保护装置内继电器数	保护装置内继电器数
继电器参数子框架	继电器参数子框架

知识表示的模块化,有利于知识的重用和系统的扩充,便于建立知识之间的对应关系,降低存储容量和系统的冗余度,增强系统的灵活性,图3表示了知识库中模拟故障与保护的对应关系。

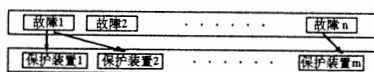


图3 模拟故障与保护对应关系

#### 5 检测分析

选择被测保护装置后,智能系统中的模拟故障模块在知识库中搜索到检测此保护装置所对应的故障情况,同时数据库管理系统根据由(变电站·屏号·位置)组成的索引代码,快速检索到有关该装置的静态数据,如继电器的整定值、各节点初始状态(0或1)等,见图4,将数据装入内存,然后根据操作规则,生成数字化故障波形,故障模拟器按接收的信号生成电流和电压,把模拟量加载到被测对象上,数据记录模块则实时采集保护动作信息(见图1),检测分析模块借助知识库中该保护的知识和内存中的初始数据,从第一项动作信息开始匹配分析,若该节点从初始态到动作态与系统期望的一致则动作正确。

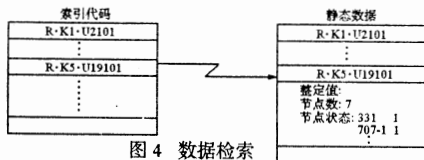


图4 数据检索

模拟故障模块和数据记录模块用实时中断程序实现,保证了系统的并发性和实时性。

解释功能是智能系统区别于传统程序的另一特征,如果一个检测系统对试验的处理只能输出简单的结论,如“距离保护装置完好”而不能给出必要的分析过程,则容易被人们视为“黑箱”中的预言难以接受,本系统在分析过程中,搜索对应的元件描述,结合判断结论形成解释文件,整组保护和综合连动试验过程与结论以实验报告形式显示在屏幕上,见表4,单体试验数据自动填写在软件生成的表格中。

表4 整组试验报告

所亭名称:江岸西变电站

试验日期:03/11/95 时间:15:23:01

继电器位置:R·K5·U19101

试验名称:馈线·距离保护

保护对象:ZZB

#### 试验过程与结论

- 1)距离保护各继电器初始状态正确
- 2)加模拟故障
- 3)阻抗继电器 331 正确动作
- 4)信号继电器 707—1 正确动作
- 5)继电器 131(二段切换)正确动作
- 6)继电器 513(三段切换)正确动作
- 7)继电器 707—2 正确动作
- 8)继电器 125 正确动作
- 9)继电器 719 正确动作
- 10)I段动作时间为105.6 ms,正确

...

#### 6 结束语

本文介绍了一个智能化检测分析系统,从模拟故障产生、数据采集分析到形成结论,然后自动生成测试文档,以磁盘或硬拷贝形式存档,实现高度自动化、集成化。系统为全中文环境,具有良好的人机界面,维护人员只需通过点菜单即可完成全部检测工作,操作简便,过程清晰,结果准确。系统为模块结构,便于今后功能扩充,知识的模块表示有利于通过组装、裁剪、拼接等方法形成多种试验方案。本系统已在某变电站进行一个多月的现场测试,效果良好,证实该系统的有效性、可靠性、智能软件用C语言实现。今后将对智能检测系统的通用性、复杂检测知识表示等方面做进一步的研究。

#### 参考文献

- 1 朱永利等.电力系统故障判断专家系统.华北电力学院学报, No. 1 1992
- 2 Taka fumi Kimura et. al., "Development of An Expert System For Estimating Fault Section in Control Center Based On Protective System Simulation" IEEE PWRD Vol. 7, No. 1 1992
- 3 J. L. 布列克勃恩. 继电保护的应用. 水利电力出版社
- 4 王永庆. 人工智能原理·方法·应用. 西安交通大学出版社

# 利用扩充内存实现多幅图象的快速存储

谢晓恒

(合肥市电子技术研究所 230027)

李卫华

(合肥中国科技大学五系 230026)

**摘 要** 本文讨论了 80286 以上微机扩充内存的管理及使用方法,并通过对 EMM386.EXE 的调用,实现了多幅图象的快速存储。

## 1 引言

随着图象处理技术的发展,图象的采集、存储变得越来越重要。在许多应用领域,往往会涉及到多幅图象的快速保存。比如,通过天空云图分析天气趋势,单幅图象无法给出天气的全部信息,而天气情况又变化得较快,这就需要保存不同地域、不同时间的多幅图象;再如,医院眼科在诊断眼疾时,需要把眼疾患者的眼底图象保存下来,考虑到仪器长时间照射患者的眼睛会对患者造成一定的损害,因此也需要对眼底图象实行快速存储。针对这些问题,除了在图象采集硬件上做工作外,我们可以利用计算机扩充内存技术来实现多幅图象的快速存储。

## 2 计算机内存结构

计算机内存由以下四部分组成:

常规内存(640KB)、上位内存(UMB)、高内存区(HMA)、扩充内存(EMS)。计算机驱动时装入 HIMEM.SYS 就可获得这些内存块。其内存结构如图 1 所示。

地址	内存块	大小
...	EMS	1M 以上内存
10FFFFH	HMA	64KB—16KB
100000H	UMB	384KB
0A0000H	常规内存	640KB
000000H		

图 1 计算机内存分配示意图

我们知道,80286 以上微机 CPU 有 20 根地址线,可以直接访问 1M 字节以内的内存,1M 以内的内存又可分为两大不同的内存块,其中底层部分为 640KB 的常规内存,一般应用程序只能使用这一部分;上层的 384KB 为上位内存,大部分为视频显示和 ROM 所占用,但仍有大量的自由空间没有用上。另外,8086/8088 只有 20 根地址线,超过 1M 字节的内存部分没有实际的物理空间与之对应,要使用这部分内存,必须通过与 1MB 以下自由空间进行映射后才可使用。

## 3 扩充内存 EMS 的使用

扩充内存原是为增加 DOS 内存容量而增加的内存插卡,DOS 的内存管理器 EMM386.EXE 可以实现这个任务。DOS 无法直接访问扩充内存。使用时需将扩充内存分成若干页,每页为 16KB。同时在系统保留区内取出一段 64KB 的区域作为扩充内存页框,分为 4 页,每页为 16KB。当 DOS 要存取扩充内存数据时,扩充内存管理器先将扩充内存数据以页为单位调入扩充内存页框中,DOS 对页框数据进行读写,操作完毕再由管理器将页框数据写回扩充内存中。利用这种方法可访问的扩充内存最大可达 32MB。计算机启动时只要在 HIMEM.SYS 中装入 EMM386.EXE,使用时利用 INT67H 激发其工作。图 2 为扩充内存的使用框图。

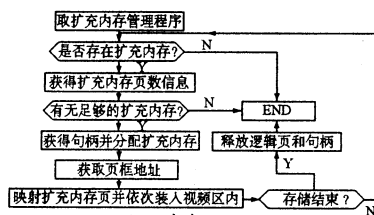


图 2 扩充内存使用框图

## 4 利用扩充内存存储数字图象

我们实际存储的每一幅图象数据为  $512 \times 512 = 256\text{KB}$ ,而 EMS 每一次只能映射 64KB,这也就意味着至少需要映射 4 次才能存储一幅图象数据。实际操作时我们把一幅图象分成  $256 \times 256 = 64\text{KB}$  的四块,每一块按行优先顺序存放在 EMS 的 64KB 中(图 3)。

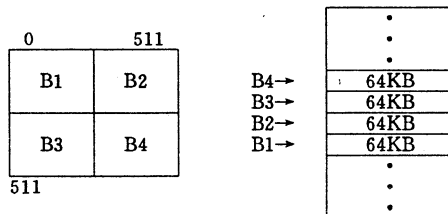


图 3 分块图象数据与扩充内存对应关系示意图

由于图象数据文件是  $512 \times 512$  按行优先顺序存放的,故不能把数据按块存入文件,必须先对数据进行

# 改进的 8259A 扩展 8031 中断的接口电路及编程实现

王文涛

(中南民族学院计科系 武汉 430074)

蔡朝晖

(华中师范大学多媒体中心 武汉 430070)

杨喜敏

(中南民族学院计科系 武汉 430074)

**摘 要** 本文详细分析了在利用中断控制器 8259A 扩展单片机 8031 中断的接口电路及驱动程序设计时所面临的问题,并从新的控制角度,利用软/硬配合,给出了一套完整的设计方案。

**关键词** 中断 中断响应 时序 接口电路

在进行实时控制系统开发时,不可避免要用到中断方式,由于 8031 的外部中断数目有限(只有 2 个),因此常常需要进行中断扩展,中断控制器 8259A 以其能够提供诸如中断优先级管理、中断向量地址、灵活的可编程控制等优越的性能,而被普遍采用。然而 8259A 与 8031 管脚并不完全兼容,要用它进行中断扩展就必须进行信号转换,然后才能与 8031 连接。尽管这方面的知识已有文章和书籍作过一定的介绍,但在我们实际应用中发现,有些书刊所介绍的硬件电路或多或少地存在问题,导致存在一个共同的问题——在采用中断方式工作时,CPU 读不到中断矢量。本文针对该问题,从软、硬两方面着手,提出一种改进的解决思路,经过实际调试和应用,证明该思路切实合理,所设计的接口电路及驱动程序运行正常。

## 1 提出问题

8259A 与 8031 不完全兼容,一方面表现在中断请求 INT 信号(该信号较易变换),另一重要方面在于 8259A 向 CPU 发出中断请求信号 INT 后,要求 CPU 必须返回中断响应信号  $\overline{\text{INTA}}$ ,可是 8031 没有提供专门的  $\overline{\text{INTA}}$  应答信号线。这样须先对接口两端进行分析,然后对 8031 及 8259A 的控制信号进行变换。

1) 首先着重来了解 8259A 的中断工作过程、中断

响应时序和操作控制。

\* 8259A 的中断请求触发方式:

8259A 为了判断外部设备是否提出中断请求,设有电平触发和边沿触发两种方式。

当 8259A 工作于电平触发方式时,若 IR 输入端上检测到一个高电平,并且在第一个  $\overline{\text{INTA}}$  脉冲到来以后维持高电平,就认为有外设提出中断请求,并使 IRR 相应位置 1。

当 8259A 工作于边沿触发方式时,若 IR 输入端上检测到一个由低电平到高电平的跳变时,并且高电平保持到第一个  $\overline{\text{INTA}}$  到来以后,8259A 就认为有外设提出中断请求。

\* 8259A 按下列顺序管理外围设备的中断申请:

a、当在 IR0—IR7 的中断申请输入端上有一个或多个输入出现高电平时,IRR 中的各对应位便被置 1,表明已经有外围设备提出中断申请;

b、8259A 在接受这些中断申请、并分辨优先级的同时,向 CPU 发出中断申请 INT 信号;

c、若 CPU 处于中断允许的情况下,在收到 INT 信号后应向 8259A 发出 INTA 脉冲作为应答;

d、当 8259A 接收到来自 CPU 的第一个  $\overline{\text{INTA}}$  脉冲时,便使 ISR 中的最高优先级相应位置 1,而将 IRR 中与之对应的位置 0,并送一条 CALL 指令码(11001101)至数据总线;

收稿日期:1996 年 7 月 10 日

排序。

## 5 结论与讨论

实验结果表明,利用扩充内存技术存储一幅  $512 \times 512 = 256\text{KB}$  的图象数据大约需要 0.6~0.7 秒,可以在 20 秒内连续存储 30 幅图象。另外,可以把 30 幅图象一次性写盘处理,存盘速度约为 6 幅/分。若中间不需要图象的显示,六分钟内可以把 30 幅图象全部写入硬盘。为提高图象的存储速度,所有的采集、存储软件全部用汇编语言写成。

## 参考文献

- 1 求伯君主编. 新编深入 DOS 编程. 学苑出版社 1994,5
- 2 沈美明,温冬婵著. IBM—PC 汇编语言程序设计. 清华大学出版社 1991,6
- 3 李兆南. 微机内存分析及其配置要点. 微型计算机. Vol. 116,2,1996,pp. 75—76
- 4 吴健康著. 数字图象分析. 人民邮电出版社. 1991,11

e、当 CPU 读到这个 CALL 指令后,便再发出两个  $\overline{\text{INTA}}$  脉冲至 8259A。这两个  $\overline{\text{INTA}}$  脉冲促使 8259 把一个预先编程的 16 位地址送到数据总线上(分两次,先低 8 位后高 8 位)。这个地址就是中断服务程序的入口地址;

f、当执行完上述的 3 字节调用指令后,便转至执行外设中断服务子程序,在子程序执行期间,其相应的 ISR 位一直保持为 1。只有在子程序的末尾,8259A 收到一个 EOI(中断结束)命令时,才使相应的 ISR 复位。

\* 图 1 给出 8259A 中断应答时序:

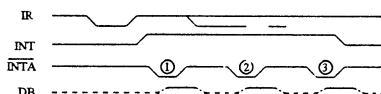


图 1

表 1 给出 8259A 的  $\text{A0}$ 、 $\overline{\text{WR}}$ 、 $\overline{\text{RD}}$ 、 $\overline{\text{CS}}$  的控制作用:

表 1

$\text{A0}$	$\text{D4}$	$\text{D3}$	$\overline{\text{RD}}$	$\overline{\text{WR}}$	$\overline{\text{CS}}$	输出操作
0			0	1	0	IRR、ISR 或中断级 BCD 码→数据总线
1			0	1	0	IMR→数据总线
						输入操作
0	0	0	1	0	0	数据总线→OCW2 寄存器
0	0	1	1	0	0	数据总线→OCW3 寄存器
0	1	×	1	0	0	数据总线→ICW2
1	×	×	1	0	0	数据总线→OCW1、ICW2、ICW3

从上可知,在设计接口电路及驱动程序时必须解决以下几个问题:

① CPU (8031) 需要模拟产生三次中断应答信号  $\overline{\text{INTA}}$ ;

② 8259A 接收中断应答信号后要送数据至数据总线,这易产生数据总线上的数据冲突;

③ 8259A 的中断应答信号  $\overline{\text{INTA}}$  对中断信号 INT 有时序上的要求。一些书刊所介绍的方案就因为忽视或没有解决好这个问题,从而达不到预想的效果。

## 2) 解决方案:

对于①,可在 8259A 提出中断请求 INT,且当 CPU (8031) 响应后,在中断服务子程序中用三条 MOVX A, @DPTR 指令来模拟产生 8259A 所需的三个中断应答  $\overline{\text{INTA}}$  信号,同时获得中断矢量;

对于②,应用组合逻辑来控制 CPU 根据 8259A 是否产生中断请求,来决定是读入中断矢量还是访问 8259A 的寄存器,即 8031 的  $\overline{\text{RD}}$  与 8259A 的  $\overline{\text{RD}}$  不是直接相连;

对于③,比如可选用 P2.6,通过在驱动程序中对其值进行控制,来确保实现中断应答信号  $\overline{\text{INTA}}$  对中断请求信号 INT 在时序上的要求。

## 2 接口电路设计

依据图 2 所示 8031 的 P2.6、 $\overline{\text{RD}}$  与 8259A 的  $\overline{\text{RD}}$ 、 $\overline{\text{INTA}}$  有如下的逻辑关系:

## a、逻辑表达式

$$\overline{\text{INT0}}_{(8031)} = \overline{\text{INT}}_{(8259A)}$$

$$\overline{\text{INTA}} = \text{P2.6} + \overline{\text{RD}}_{(8031)}$$

$$\overline{\text{RD}}_{(8259A)} = \overline{\text{RD}}_{(8031)} + \overline{\text{INTA}}_{(8259A)}$$

## b、逻辑关系

表 2

名称	非中断状态		中断状态			
P2.6(8031)	1		0	1	0	1
$\overline{\text{RD}}_{(8031)}$	0	1	0	1	0	1
$\overline{\text{RD}}_{(8259A)}$	0	1	1	1	0	1
$\overline{\text{INTA}}_{(8259)}$	1	1	0	1	1	1

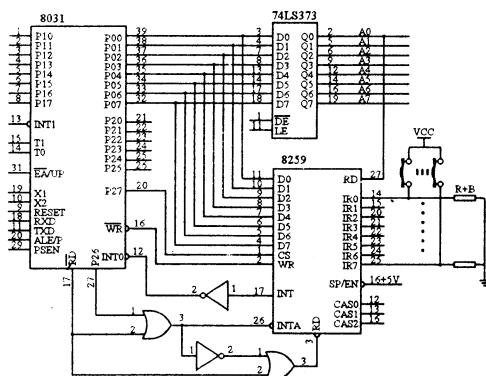


图 2

图 2 中,  $\overline{\text{CS}}$  与 P2.7 相连, P2.6 主要控制中断应答信号  $\overline{\text{INTA}}$  对中断请求信号 INT 的时序要求,平时 P2.6 的值为 1,使中断应答信号  $\overline{\text{INTA}}$  保持为高电平(即屏蔽掉中断应答信号),则 8259A 的  $\overline{\text{RD}}$  信号受 8031 的  $\overline{\text{RD}}$  信号的直接控制,实现正常读/写,故这时 8259A 的入口地址为 7FFEh—7FFFh;若 CPU 要读取中断矢量,在中断服务子程序中,先将 P2.6 置为 0,则此时的中断应答信号  $\overline{\text{INTA}}$  随 8031 的  $\overline{\text{RD}}$  信号的变化而变化,当 8031 的  $\overline{\text{RD}}$  有效时(即其值为 0), $\overline{\text{INTA}}$  也变成有效,从而模拟出 CPU 向 8259A 发送了一个中断应答信号,而此时 8259A 的  $\overline{\text{RD}}$  信号则为无效(避免了数据总线上的数据冲突),这样通过三条 MOVX A, @DPTR 产生三个中断应答信号  $\overline{\text{INTA}}$  给 8259A,使 CPU 获得中断矢量,故这时的入口地址为 3FFEh—3FFFh。CPU 获得中断矢量后,紧接着将 P2.6 置为 1,结束中断响应,从而完成中断矢量的获取过程,使 CPU 转入正常读/写状态。由于图中只有一片 8259A,所以  $\overline{\text{SP/EN}}$  接高电平。

## 3 驱动程序设计

### 3.1 主程序

```
org 0000h
ljmp main
org 0003h
ljmp int0
.....
main: mov sp, #60h
      clr ea
```



# 一种用单片机组成的高精度 A/D、D/A 转换电路

钟志良

(解放军福州医高专微机教研室 350003)

**摘 要** 本文介绍用单片机定时器、中断接口等软硬件资源实现高精度多路 A/D、D/A 转换。该电路具有自动零点调校,自动量程转换等特点。

**关键词** 单片机 A/D 转换 D/A 转换

## 1 引言

在实时测控和智能化仪器仪表等应用系统中,现场的温度、压力、流量等非电模拟量经传感器转换成连续的电信号后,还需要经模数转换变成离散的数字量;离散的数字量也需经数模转换变成连续的电信号才能

控制相应的设备。随着集成电路的发展,目前已有大量现成的 A/D、D/A 芯片,但由于受许多客观条件的限制,很多应用中特别在智能化仪器仪表中,常常需要单独设计既简单精度又高的 A/D、D/A 转换电路。

目前的 A/D 电路不外乎直接 A/D 和间接 A/D 两大类,直接 A/D 主要是逼近式 A/D,而间接 A/D 则以双积分 A/D 较为普及。本文介绍的 A/D、D/A 转换

收稿日期:1996 年 7 月 29 日

```
.....
;8259A 的初始化
setb p2.6;屏蔽 8259A 的中断应答信号  $\overline{INTA}$ 
mov dptr, #7ffh
mov a, #17h;边沿触发(若用 MOV A, #1FH;则可实现
电平触发)
movx @dptr, a
inc dptr
mov a, #01h
movx @dptr, a
mov a, @00h
movx @dptr, a
movx @dptr, a
setb ex0
setb it0
setb ea
.....
```

## 3.2 中断服务子程序

```
int0: push psw
      push a
      clr p2.6;屏蔽 8259A 的  $\overline{RD}$ ,使  $\overline{INTA}$  有效
      mov dptr, #3ffh
      movx a, @dptr
      cjne a, #0cdh, endi;不是 CALL 命令码,则中断
      返回
      movx a, @dptr
      mov dpl, a
      movx a, @dptr
      setb p2.6;完成中断矢量的读取过程,屏蔽
      8259A 的中断应答信号  $\overline{INTA}$ 
      mov dph, a
      clr a
      jmp @a+dptr
.....
```

```
org 0100h
i0:ljmp ir0
nop
i1:ljmp ir1
nop
i2:ljmp ir2
nop
i3:ljmp ir3
nop
i4:ljmp ir4
```

```
nop
i5:ljmp ir5
nop
i6:ljmp ir6
nop
i7:ljmp ir7
ir0:mov pl, #09ah
ljmp endi
.....
ir0:.....
      ljmp endi
ir1:.....
      ljmp endi
ir2:.....
      ljmp endi
ir3:.....
      ljmp endi
ir4:.....
      ljmp endi
ir5:.....
      ljmp endi
ir6:.....
      ljmp endi
ir7:.....
.....
```

```
endi:mov dptr, #7ffh
      mov a, #20h
      movx @dptr, a
      pop a
      pop psw
      reti
```

这里所提供的接口电路设计,通过驱动程序的设计,既适用于边沿触发方式,也可实用于电平触发方式,经过实际应用证明,整套设计合理实用,工作稳定可靠。

## 参考文献

- 1 何利民. MSC—51 系列单片机应用系统设计. 北京航空航天大学出版社
- 2 刘善乐等. 微型计算机接口技术及应用. 华中理工大学出版社
- 3 有关 8259A 中断控制器应用的问题解答. 电子技术应用, 1996. 1

电路则充分利用了单片机的软硬件资源,具有精度高,输入通道多,零点自动调校,量程自动变换等突出优点,有兴趣者不妨一试。

## 2 工作原理

### 2.1 简单的积分电路

图 1.1 为简单的 RC 积分电路,图中电容 C 两端的电压  $V_c$  与电流  $i$  的关系满足积分规律:

$$V_c = \frac{1}{C} \int i dt \dots\dots\dots ①$$

式中,电流  $i$  为非常数,  $V_c$  与输入电压  $V_i$  在积分的开始部分呈近视线性关系,随着时间的增长,积分误差逐渐加大,如图 1.2 所示。

可以看出,理想的积分运算,电流  $i$  必须维持恒定。

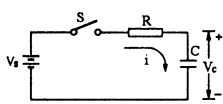


图 1.1 简单的积分电路

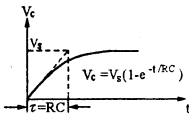


图 1.2 输出电压特性

### 2.2 标准积分器

利用集成运算放大器组成的标准积分电路如图 2.1 所示,其等效电路如图 2.2 所示,从运算放大器的特性可以得出:

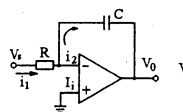


图 2.1 标准积分器

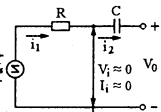


图 2.2 等效电路

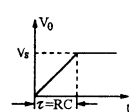


图 2.3 输出特性

$$V_i \approx 0, I_i \approx 0;$$

从而

$$i_1 \approx i_2 \approx i = \frac{V_i}{R}$$

这样,电容 C 就以恒流  $i$  充电,即

$$V_i - V_o = \frac{1}{C} \int \frac{V_i}{R} dt$$

因此

$$V_o = -\frac{1}{C} \int \frac{V_i}{R} dt \dots\dots\dots ②$$

②式中负号表示输出电压与输入电压相位相反,其输出电压特性如图 2.3 所示。当积分初期,即  $0 < t < RC$  时,输出电压与输入电压呈线性关系,即

$$V_o = -\frac{V_i}{RC} \times t$$

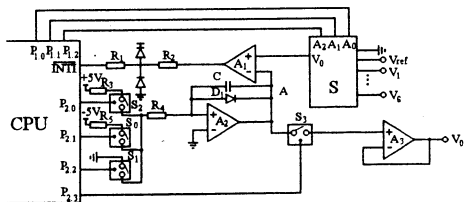


图 3.1 电原理图

当  $t \geq RC$  时,  $V_o = -V_i$ 。

### 2.3 A/D、D/A 转换电路

#### 2.3.1 电原理图

A/D、D/A 转换电路的电原理图如图 3.1 所示。

图中微处理器采用 MCS—51 系列单片机,选用了其内部定时/计数器 T1, MCS—51 单片机 T1 的内部结构框图如图 3.2(工作于方式 1)所示。置 MCS—51 方式控制字 TMOD 的最高位即 Gatel 为“1”时,定时器的启动与停止就完全由外部 INT1 引脚控制:  $\overline{INT1} = “0”$ , 计时器 T1 即停止计时;当  $\overline{INT1} = “1”$  时, T1 即从当前值 X 开始对内部定时信号加“1”计时,当 T1 从 0FFFFH 加“1”产生溢出时,溢出标志 TF1 置“1”,并向 CPU 发中断申请。

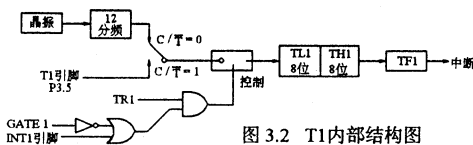


图 3.2 T1 内部结构图

图 3.1 中其它部分电路为:

- A1——电压比较器,当 A1 正端(+)输入大于负端输入时,比较器输出高电平,当正端(+)输入小于负端输入时,比较器输出低电平;
- A2——积分器,通过电容 C 对输入电压进行积分;
- A3——电压保持器(电压跟随器);
- S0——充电选择开关。选择 -5V 通过电阻  $R_5, R_4$  对电容 C 充电;
- S1——充电暂停开关;
- S2——放电选择开关。选择 +5V 通过电阻  $R_3, R_4$  对电容 C 放电;
- S3——采样保持开关;
- S——输入选择开关。通过 A0, A1, A2 可选择 8 路中的某一路  $V_i$  输入。

#### 2.3.2 A/D、D/A 过程

##### (1) A/D 过程

首先输入选择器 S 通过 CPU 的  $P_{1.0}, P_{1.1}, P_{1.2}$  引脚选择某一路输入信号  $V_i$  加到比较器的正端,接通 S0,断开 S1, S2, S3, 同时启动 T1 计时(从 0 开始)。这时,积分器 A2 对固定电压(-5V)进行积分。A 点电压随时间以固定斜率(由  $R_5, R_4, C$  决定)从 -0.7v(箝位二极管 D1 作用)上升,当 A 点电压  $V_a$  与输入电压  $V_i$  相等或超过时,比较器 A1 输出低电平,通知 CPU 并停止 T1 计时。此时,读出 T1 计时值,作为该路 A/D 值,并断开 S0,接通 S2,让积分器 A2 充分放电(反向充电),A 点电压再次降为 -0.7v,准备下次 A/D 或 D/A 转换。

##### (2) D/A 过程

首先将相应的电压输出值转换成对应的计时值送入 T1 计时器;接通 S0,断开 S1,S2,S3,启动 A2 正向积分,同时启动 T1 计时,并允许 T1 中断;这时 A 点电压从 -0.7V 上升,当 T1 计时溢出时,断开 S0,接通 S1,S3,采入积分电压 V<sub>i</sub>,延时一段时间后,断开 S1,S3 接通 S2,让积分器 A2 充分放电(反向充电),准备下次 A/D 或 D/A 转换。此时采样保持器 A3 的输出即为欲输出的电压。

### 2.3.3 特点

从上面的介绍可以看出,图 3.1 的电路具有许多独特的优点,它可以对多路输入进行 A/D 转换,占用 CPU 引脚少,转换精度高;还可以抑制零漂和自动量程转换等,下面着重对以下几个特点加以说明:

#### (1)抑制零点漂移

图 3.1 的电路中,由于输入选择开关 S 的输入端之一接地(GND),因此,在采样实际的输入电压 V<sub>i</sub>~V<sub>6</sub> 前,可先采样零点,通过输入电压即③式的计算,就可以排除零点漂移的影响。

#### (2)自动量程转换

图 3.1 的电路中,由于输入选择开关 S 的输入端接有输入量程参考电压(V<sub>ref</sub>),因此,在采样实际的输入电压 V<sub>i</sub>~V<sub>6</sub> 前,可先采样输入量程参考电压(V<sub>ref</sub>),通过③式的计算,就可以实现自动量程转换。输入电压 V<sub>i</sub> 的计算如下:

$$V_i = \frac{AD_i - AD_0}{AD_{ref} - AD_0} \times V_{ref} \dots\dots\dots (3)$$

其中:

- V<sub>i</sub> —— 第 i 路输入电压;
- AD<sub>i</sub> —— 第 i 路输入电压的采样值;
- AD<sub>ref</sub> —— 量程参考电压的采样值;
- V<sub>ref</sub> —— 量程参考电压;
- AD<sub>0</sub> —— 对地电压采样值。

#### (3)精度分析

由于 MCS-51 定时/计数器 T1 的计数器为 16 位,因此图 3.1 电路的 A/D、D/A 转换精度为 16 位。当然,若软件设置某一寄存器作为累计 T1 计时溢出,理论上,A/D 转换精度可达任意位,但由于此时 A/D 转换时间过长,失去实际意义。

### 3 软件设计

A/D、D/A 转换的软件设计主要考虑五个部分:

(1)主程序(Main)部分完成各个 I/O 引脚的设置,定时/计数器 T1 的初始化,中断的设置以及对其它部分的调用;

(2)A/D 转换启动模块(Atod)完成模/数转换前的开关设置等工作;

(3)D/A 转换启动模块(Dtoa)完成数/模转换前 T1 时间常数的设置以及开关设置等工作;

(4)中断服务程序(Intx1)完成读入计时值作为相应输入 A/D 值,并准备下次 A/D 或 D/A 转换等工作;

(5)中断服务程序(Intt1)完成采样保持及准备下次转换等工作。

其它部分由于与具体应用有关,不再详细说明。下面是部分程序清单:

```

ORG 0000H
LJMP Main
ORG 000BH
LJMP Intt1;T1 计时中断向量
ORG 0013H
LJMP Intx1;外部中断 1 中断向量
Main;MOV P1,#0 ;
MOV P2,#11110001B ;让 A2 充分放电
LCALL Delay ;延时等待复位正常
MOV SP,#65H
MOV TMOD,#00010001B;T1 工作于方式 1
MOV TCON,#0
MOV IE,#0 ;关中断
Loop;LCALL Select ;选择某一路输入
LCALL Atod
LCALL Liner ;计算输入输出电压
LCALL Dtoa
LJMP Loop
Atod;MOV TMOD,#10010001B;Gate1=1
MOV TH1,#0
MOV TL1,#0
MOV P2,#11110010B ;A2 充电
SETB TR1 ;启动 T1 计时
SETB EX1 ;外部中断 1 开放
SETB EA ;总中断开放
RET
Dtoa;MOV TMOD,#00010001B;Gate1=0
MOV TH1,OUT ;置输出电压
MOV TL1,OUT+1 ;时间常数
SETB TR1
MOV P2,#11110010B ;A2 充电
SETB ET1
SETB EA
RET
Intx1;CLR TR1 ;停止 T1 计时
CLR ET1
CLR EX1
MOV CH,TH1 ;读入计时值
MOV CH+1,TL1
MOV P2,#11110001B ;A2 放电
RETI
Intt1;CLR TR1
CLR ET1
CLR EX1
MOV P2,#11111101B ;A2 保持
LCALL Delay ;延时
MOV P2,#11110001B ;A2 放电
RETI

```

### 参考文献

- 1 徐爱卿,孙涵芳等. 单片微型计算机应用和开发系统. 北京航空航天大学出版社,1992.8

# Foxpro 下一个功能更强的“屏幕程序生成器”的设计与实现方法

俞 坚

(杭州应用工程技术学院电机系 310012)

**摘 要** 本文介绍了在 Foxpro 下一个功能更强的多记录全屏操作的“屏幕程序生成器”的设计与实现方法。此工具的开发与使用,对于进一步提高编程效率,缩短软件开发周期,具有十分重要的意义。

**关键词** Foxpro 屏幕程序生成器 数据库

## 1 前言

在编写一个大型的数据库管理程序时,人们总希望找到一个好的编程工具,以便尽可能地减少编程工作量,提高编程效率,缩短软件开发周期,新的开发语言,如 Foxpro 第四代语言顺应软件开发者的要求。提供了许多快速有效的编程工具,如“屏幕生成器”“菜单生成器”“报表生成器”等。

然而 Foxpro 的“屏幕生成器”有其明显的欠缺,因为它只有在屏幕上处理单记录时才让用户感到得心应手,而在处理多记录时,却并没有为用户带来多少方便。然而在实际应用开发中,总是会非常频繁地遇到一屏处理多记录的情况,如多条记录以某种表的形式进行全屏输入、修改、删除,多记录表的查询,还有许多其它操作等等,并且这些工作所占的比重相当大。如果能有效地实现多记录全屏操作的屏幕程序自动生成,那么编程效率将大大提高,软件开发周期将大大缩短。

几年以来,本人在应用开发实践的同时,实现了一个多记录全屏操作的“屏幕程序自动生成器”,并在开发实践中,不断地得到了发展与完善。利用此工具,在生成一屏处理多记录的屏幕程序时,将十分方便快捷。有了此工具,程序开发效率得到了极大的提高。几分钟内便可以得到原先常需几天或更长时间才能完成的程序。

## 2 多记录全屏操作的“屏幕程序生成器”的设计要求

单位往来目录录入.....

往来单位	单位全称	电话
北京旅行车厂	北京旅行车厂股份有限公司	7312465
北京轻型车厂	北京轻型汽车厂制造有限公司	2319670

↑表头  
.....  
↑表体  
↓表尾

图 1 帐务处理中单位往来目录全屏录入界面

### 1) 多记录全屏操作的“屏幕程序生成器”所生成

## 程序具有的功能

根据用户需要的表格式进行多记录全屏操作。灵活方便地进行记录的插入、增加、修改、删除。输入内容能方便地以屏幕表的格式打印。当数据库记录数太多无法一次全部显示在屏幕表中时,可利用方向键、PgUp 键、PgDn 键来上下或左右滚动画面以便能编辑操作数据库中的所有记录。在上述功能的基础上,可方便地增加其它控制操作功能。

2) 使用多记录全屏操作的“屏幕程序生成器”生成程序时,只要在入口处定义如下的内容,便可马上得到用户所需的程序,可谓所见即所得。

① 打开被操作的数据库。

② 定义表:表头、表尾内容直接存入相应数组中。表体定义,只需将表体的一行单元存入格式单元变量 b\_unit 便可。

③ 定义表体行数、栏数及起始位置。

④ 定义表各栏的显示格式。

如图 1 所示的帐务处理中单位往来目录全屏录入程序,只要在入口处做如下的工作,程序便可即刻自动生成。

① 打开单位往来目录数据库。

② 将 4 行表头内容存入一数组,将 1 行表尾内容存入另一数组。

③ 表体格式单元存入变量 b\_unit

即: b\_unit = ' | | | | '。

④ 置表体行数、栏数及起始行变量值,即 max\_h = 10, max\_l = 3, h\_first = 5。

⑤ 根据需要定义表各栏的显示格式,即 f\_1 = 'xxxxxxxxxxxx',...。此项工作也可不做。

## 3 多记录全屏操作的“屏幕程序生成器”的具体实现

屏幕程序生成器主要由四部分组成:①条件定义。②条件识别。③自适应地生成符合条件的屏幕程序。④向外界提供单一的控制接口。

### 1) 条件定义

通过条件定义,用户把自己设计要求告诉“屏幕程序生成器”。对用户来说条件定义越简单越好。本多记

录全屏幕操作的“屏幕程序生成器”条件定义部分已设计得非常简单,如上所述用户只需要给出很少的条件:如操作库、表头、表尾、表体单元、表体行数、起始行及各栏的显示格式便可。

## 2) 条件识别

用户定义的条件,有些可直接供后续过程使用,如表头、表尾、表体单元、表体行数、起始行及各栏的显示格式( $f_1, f_2, \dots, f_{\max\_1}$ )。有些需经过分析加工处理后形成后续过程所需的参数,如以下几方面工作:

① 求出表体的屏幕区域:根据表体行数( $\max\_h$ )、起始行( $h\_first$ )及表体格式单元( $b\_unit$ ),求出表体屏幕区域左上角坐标及右下角坐标,分别存入( $h\_first, l\_first$ )与( $h\_last, l\_last$ )中。

② 求表体各栏的显示列位置:根据表体栏数( $\max\_1$ )及表体格式单元,经分析运算求出表体各栏的显示列位置,并将它们存入变量  $l_1, l_2, \dots, l_n$  中。

## 3) 自适应地生成符合条件的屏幕程序

### (1) 屏幕表的自动生成

根据表头、表尾、表体、行数及表体格式单元等参数,在屏幕上自动生成一张符合用户要求的空表。

(2) 屏幕程序生成器中的各功能模块具有自己适应外界变化的能力

屏幕程序生成器在生成程序时,不是象编译程序将高级语言生成机器语言那样,依一定的词法及语法规则进行从无到有地生成程序,而是先将这些功能模块预先编好,只是这些预先编好的功能模块具有自己适应外界变化的能力。如尽管数据库及表的行列数变了而处理模块无需改变。下面讲述一些与实现这方面功能有关的技术。

① 处理程序与所操作的数据库通过数组作为统一接口。

处理程序不直接读写数据库,它所处理的记录首先由 scatter 读到数组  $cur\_a$  中,处理好后如需要写回由 gather 实现。这样对于不同的数据库,在处理程序看来就没有任何的不同。

② 通过宏替换及 for 语句来实现对每一记录各域(表各栏)进行 say 或 get 操作,以便灵活地适应不同数据库不同域数的不同处理要求。如编辑或修改当前行,可由下面程序实现。

```
for k=1 to max_1
  &&max_1 为入口处定义的表栏数(数据库域数)
  kk=str(k,1)
  @h_first+cur_h-1,l_&kk;
  get cur_a[&kk];
  pict(f_&kk)
endf
read
```

③ 在进行上下移动及上下滚页或插入、增加、删除等操作时,表体内容常需要作滚屏、重显及屏幕表格式修复等处理。这些操作应依赖条件识别后的参数,以便

当屏幕表格式及大小变化时,能自动适应这些变化。下面以表体向下滚  $hs$  行为例,说明滚屏与表格式修复及滚入新数据显示的过程。

```
proce gp_xs &&滚屏与表格式修复
para hs
scroll h_first,l_first,h_last,l_last,hs
for vc=1 to hs
  @h_first+(vc-1),l_first say b_unit
endf

proce d_dis &&滚入新数据显示
para cc
priv cc,vc,m,c2
for vc=1 to cc
  if eof()
    exit
  endf
  scatter to cur_a &&读当前行
  c2=vc-1 &&位置
  do dqh_xs with c2 &&当前行内容显示
  skip
endf
proc dqh_xs &&显示当前行全部记录域
para hen
for m=1 to max_1
  mm=str(m,1)
  @h_first+hen,l_&mm say t[&mm] pict(f_&mm)
endf
```

## 4) 向外界提供单一的控制接口

在进行记录域的编辑及修改操作时,根据实际情况常需作编辑前的 when 控制及编辑后的 valid 控制。对于不同域  $i$ ,可以简单地以函数名  $w\_i, v\_i$  向外提供接口,用户如需要控制某域,只要在相应原先空的函数中增加内容便可。程序中只要在 get 语句中加入对 when 及 valid 函数的宏替换引用便可。

## 5) 主体程序框架

### ① 定义操作数据库记录的数组:

dimen  $cur\_a[\max\_1]$  当前记录

dimen  $last\_a[\max\_1]$  上一记录

② 初始化与屏幕显示位置及数据库记录位置有关的变量:

$fn=recco()$  数据库文件记录数

$rn=1$  当前记录数

$cur\_h=1$  当前行数

③ 根据参数:  $\max\_h, \max\_1, l_1, l_2, \dots, l_n, f_1, f_2, \dots, f_n$  等显示第一屏内容,并让亮条显示在表的第一行显示内容上。

### ④ 主体循环

do while .t.

scatter to  $cur\_a$  读当前记录到数组中

if 屏幕当前行发生改变且原记录行还在屏幕上

清原先行亮条处理

endf

显示当前记录行

显示当前记录域

do case &&操作功能键

case 左键

do key\_left

case 右键

do key\_right

case 向上键

(下转第 31 页)

# Windows 画笔在软件界面设计中的应用研究

谭祖胜

(武汉交通科技大学 173# 430063)

**摘 要** 本文结合实际,阐述了 Windows 画笔在软件用户界面设计中的应用,以期提高软件开发者的工作效率。

**关键词** Windows 软件 用户界面

## 1 前言

软件的用户界面是用户与软件交流的直接语言。一个软件开发得成功与否,从某种程度讲,主要取决于该软件的用户界面开发得是否成功。目前市场上的软件,大都追求类 Windows 风格,都能在 Windows 下运行。比如,Visual Basic, Visual C++ 以及 Foxpro for Windows 等软件开发工具,备受软件开发者的青睐,因为它们本身具备 Windows 风格,且提供了用户界面开发工具,给软件开发者减轻了很多负担。

但任何一种软件,不论它具备多么强大的功能,这些功能也只能是最基本的。软件开发者可以随心所欲地应用它开发自己的产品,但不同程度的产品,单凭这种工具软件,开发者仍然要付出不同程度的努力才能使自己的软件产品拥有令人满意的用户界面。

然而,当你应用 Windows 的画笔工具配合用户界面的开发,你便会取得事半功倍的效果。本文以 Foxpro for Windows 下软件的开发为例,结合实践经验,阐述 Windows 画笔在软件用户界面设计中的应用。

## 2 封面或永久背景的制作

封面是一个软件启动后的第一幕画面,它在整个软件中的地位是相当重要的。一般地,软件封面主要包括以下几个部分:软件名称、版本号、注册商标、软件开发者或软件使用者以及口令设置等。你也可以将口令设置以外的部分定制为整个软件自始至终的永久背景。

虽然 Foxpro for Windows 的屏幕生成器有强大的图形和文字编辑功能,但未免太烦琐,既费时且效果欠佳。比如阴影字的做法一般都是通过如下方法实现的:

- 1) 在指定位置用欲显示的字体、大小和颜色写出文字;
- 2) 重复 1, 但将颜色置黑;
- 3) 用“Object/Send to back”将黑字置后,即将欲显示的字置前;
- 4) 用光标键调整两种颜色的字的位置,以形成阴

影字的效果。

但是,应用 Windows 中的画笔工具,只需选定前景色和背景色,将写的文字用“文本/阴影”处理,瞬时即可达到目的。

Windows 的画笔工具之所以能使封面制作变得简单易行,主要是因为它对图形和文字的编辑处理能力更强劲,且能生成 Foxpro for Windows 能直接调用的 BMP 文件。在应用 Windows 的画笔配合 Foxpro for Windows 定制封面之前,先作好总体规划,即选用什么样的图案作背景及各部分的位置等应事先确定,然后按如下步骤进行:

- ① 启动画笔工具,建立自己的 BMP 文件;
- ② 在“选项/图像属性”中按像素定好画面的高和宽,确定封面的大小;
- ③ 用“编辑/从...粘贴”贴入所选的图像,可以是专门的图片库中拷贝来的,也可以是从 VCD 影片中“抓”来的;
- ④ 按照总体规划,利用画笔中的各种工具,对③贴入的图像进行编辑,如绘制凸凹线,凸凹面、写上各相应的文字内容并作美术加工以及绘制公司的徽标等,完成整个画面;
- ⑤ 存入自己的 BMP 文件,退出画笔;
- ⑥ 启动 Foxpro for Windows,启动屏幕生成器,建立自己的封面文件;
- ⑦ 用“Screen/Layout/Color”中的 Wallpaper 功能,选择①—⑤所制作好的画面为背景;
- ⑧ 把窗体的大小定为②中的大小并居中(选上 Center),把 Window Style 中 Attributes 的全部选项取消,Border 选为 None;
- ⑨ 用弹出式按钮工具或程序段对封面进行控制设计;
- ⑩ 保存自己的封面文件,应用“Program/Generate”进行编译,退出屏幕生成器。

这样,一个完美的封面便完成了。如果要使上述画面成为软件的永久背景,那么画面的大小最好定为充满整个屏幕。另外,该画面不能被激活而使前台屏幕消失(置后)。解决这一问题可以采用如下方法:

打开整个软件的 Project 文件,选择不能置后的文件名称,选“Edit”,在“Edit Screen Set”窗口中选“More >>”,选中“Modal Screens”选项即可。

### 3 输入输出界面的制作

翻过精美的封面,即进入了所使用的软件,用户便可以进行自己的工作,输入各种数据,软件处理后的信息或结果输出到屏幕或打印机。所以,与用户打交道时间最长的便是输入输出界面。也正是这些输入输出界面决定了软件的生命力和市场。

我在为某公司开发人才管理软件时,对输入输出界面的定制工作作了许多探讨,最后从多方面进行总结,认为用 Windows 的画笔配合 Foxpro for Windows 进行编制,不仅效果很好,且节约了宝贵的时间。

比如,个人履历表的输入输出界面,为了方便用户,简单直观,其整体结构最好设计成与打印机输出结果完全相同的表格形式,也就是说输入输出界面的总体背景是一样的。但是,Foxpro for Windows 中输入输出界面的屏幕文件编制工作能不能把其中之一拷贝,然后将各字段的 Field 由 Input Field 改为 Output Field(或由 Output Field 改为 Input Field)而成为另一个呢?当然不行。因为屏幕文件建立在相应数据库之上,而输入输出数据库一般是不同的。

用 Windows 的画笔配合 Foxpro for Windows 制作输入输出界面的步骤如下:

①启动画笔工具,建立自己的 BMP 文件;

②在“选项/图像属性”中按像素定好画面的高和宽,确定界面的大小;

③按照规划好的表格形式,利用画笔中的各种工具,制作画面。其总体背景以灰色为宜,这样显得庄重、严肃且凸凹线和凸凹面易于表达。将所有不属于用户输入的信息,如表格线、表格名称及各栏目名称等,在此全部完成;

④存入自己的 BMP 文件,退出画笔;

⑤启动 Foxpro for Windows,启动屏幕生成器,分别以输入或输出数据库建立自己的输入或输出屏幕文件;

⑥在建立输入屏幕文件时,用“Screen/Layout/Color”中的 Wallpaper 功能,选择①~④所制作好的画面为背景。在背景(即表格)上各处需要用户填写的地方,利用屏幕生成器的各种工具,比如年月输入可用数字增减工具,部门可用弹出式字段工具,来完成各项输入的设计工作。而输出屏幕文件的建立,仍然选用①—④所建立的画面为背景,各输出项一般只需用字段工具或显示式字段工具即可完成;

⑦保存自己的输入或输出屏幕文件,应用“Program/Generate”加入各自的控制屏幕文件,以便对其进行操纵,最后进行编译,退出屏幕生成器。

如果是信息窗或提示窗,同样可以在 Windows 的画笔中将要显示的信息或提示,包括?或!等的图案都制作好,然后在 Foxpro for Windows 的屏幕生成器中加入按钮即可。

### 4 结论

一般,开发一个非常友好的应用软件,其中 70% 以上的时间都用在用户界面的开发上。所以,选择一个好的用户界面开发途径和方法,不仅可以提高软件开发的效率,更重要的是可以把宝贵的时间用在改善软件的性能上,使得开发出的软件更便于使用和维护,以提高软件的水准。

### 参考文献

- 1 孙燕唐,徐玲,戴月明. Foxpro 2.5 基础教程. 学苑出版社,1994 年 11 月
- 2 苏天凤,林毓玲. Windows 3.1 中文版——最佳专辑. 人民邮电出版社,1995 年 4 月

(上接第 2 页)

的例子是极少的,特别是后者。

### 6 结束语

多媒体技术的应用一直受到计算机硬件速度和网络带宽的影响。随着校园网的进一步发展,网络性能的逐步提高,可以考虑将多媒体技术运用到校园网 EMIS 的开发中去,使信息的表达更加方便和直观。

校园网的建设和发展,给学校 EMIS 的开发应用带来了新的挑战。要开发出真正有实用价值,并能促进学校现代化管理的校园网 EMIS,必须有正确的系统规划和良好的人员组织,同时充分利用当今各种成熟的,先进的计算机技术、思想和方法。

### 参考文献

- 1 邝孔武. 我国企业信息系统建设的回顾与展望. 计算机世界 1995.11.1
- 2 赖剑煌. 完善 MIS 开发理论体系. 计算机世界, 95.11.1
- 3 王众托. 计算机在经营管理中的应用——新的系统结构. 大连理工大学出版社,1994.12
- 4 王贝李. Internet 与 C/S 应用有机结合的实例. 国际电子报,96.8.5
- 5 <http://www.w3.org/pub/WWW/CGI/>
- 6 <http://www.javasoft.com/products/JDBC/index.html>



# 利用 Windows 环境下 DDE 通讯实现 MIS 图表的动态打印

王爱学

(大庆石油学院计算机科学系 黑龙江安达 151400)

**摘 要** 本文介绍了利用 Windows 环境下 DDE 通讯实现 MIS 图表的动态打印方法,用 Foxpro for Windows 编制了 DDE 通讯程序,用 Excel 中的 Visual basic 宏语言编制了打印服务程序,介绍了打印图表模板的详细构造方法。

**关键词** MIS DDE 通讯 图表 打印

## 1 引言

Foxpro for Windows 2.5B 以其运行速度快、操作容易、支持网上运行、多窗口用户界面等优点深受广大用户喜爱,该软件还为其应用程序开发者提供了 DDE 通讯接口以实现与其它 Windows 应用程序的数据交换。然而, Foxpro 没有提供自动生成图表打印功能,难以动态实现图表的制作。Microsoft Office 中的电子表格软件 Excel 5.0 图表处理功能强,且能够直接读取 Foxpro 数据库数据。如果利用 Windows 环境下的 DDE 通讯将二者的优点结合起来开发 MIS,将 Excel 作为后台运行的服务器,用 DDE 通讯调用 Excel 打印 MIS 图表,将会获得非常好的使用效果。本文结合作者开发 MIS 中的经验,介绍利用 Windows 环境下 DDE 通讯实现 MIS 图表的动态打印方法。

## 2. 图表打印工作流程

在 MIS 中,首先用 Foxpro 开发出名为 CallPrt.prg 的 DDE 通讯程序,然后用 Excel 制作出打印图表模板,再用 Excel 中的 Visual Basic 宏命令编制出名为 PrintDbf.xls 的图表打印服务程序,最后将这些文件组合在一起,完成 MIS 图表打印工作。图表打印工作流程按以下步骤进行。

1) 用 CallPrt.prg 发送 DDE 命令启动 Excel 服务器在后台运行, MIS 作为客户通过 CallPrt.prg 中的 DDE 命令发送 Excel 宏至服务器,调用 Excel 中的图表打印服务程序 PrintDbf, 图表打印服务开始。

2) 图表打印服务程序发出从 MIS 图表库取数据宏命令。

3) 图表库按 dbf 文件类型打开,以 Excel 工作表的方式读取数据。

4) 图表打印服务程序发出打开图表模板宏命令。

5) 图表模板被打开,按图表库数据在模板上绘制出图表。

6) 图表打印服务程序打印出图表。

MIS 图表打印处理框图如图 1 所示。

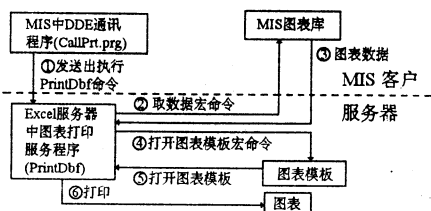


图 1

## 3 DDE 通讯程序

Foxpro 中的 DDE 通讯程序 CallPrt.prg 必须包括如下功能:

1) 启动后台运行的 Excel 服务器,命令服务器打开图表打印宏语言程序文件 PrintDbf.xls。

2) 建立 Excel 服务器对 Foxpro 的服务。

3) 发送出执行宏 PrintDbf 的命令,用 PrintDbf 打印出图表。

4) 打印完成后关闭 Excel 服务器对 Foxpro 的服务。

5) 关闭后台运行的 Excel 服务器。

DDE 通讯源程序清单如下:

```
*** 程序名: CallPrt.prg
*** 后台运行 Excel, 打开 PrintDbf.xls
RUN/N7 EXCEL PRINTDBF.XLS
*** 以下建立 Excel 服务器对 Foxpro 的服务
iChannel=DDEInitiate("Excel","System")
=DDEExecute (iChannel,['APP.ACTIVATE ("FoxPro",
1)'])
*** 发送出执行宏 PrintDbf 的命令, 开始打印报表
=DDEExecute (iChannel,['RUN ("PRINTDBF.XLS!
PRINTDBF",FALSE)'])
*** 关闭 Excel 服务器
=DDEExecute(iChannel,['ERROR(FALSE)'])
=ddeeXECUTE(IcCHANNEL,['qutto'])
=DDETenminate(iChannel)
```

## 4 图表模板的制作

Excel 打印服务程序需要将读取的数据处理后送到图表模板打印输出。图表模板是根据用户要求构造

的 Excel 工作表,用来表达打印的图表格式。制造图表模板需要四个步骤,用 Excel 直接完成,下面通过一个实例说明具体构造方法。

#### 4.1 打开 Foxpro 图表库

Foxpro 图表库是一个普通的 .dbf 库文件,里面存放着待打印的数据。设该文件是某化工厂污水月处理指标数据库,库中已存有三月上旬 PH 数据(10 条记录),每条记录对应一日数据,文件名为 ph.dbf,其库结构如下:

字段名	类型	长度	小数位数
日期	数字型	2	0
PH1	数字型	4	0
PH2	数字型	4	0

设 ph.dbf 在 Excel 中用文件菜单按 dBase 文件类型打开 ph.dbf 图表库,打开库后,各字段对应的单元格如图 2 所示:

	A	B	C
1	日期	PH1	PH2
2	1	9.08	7.03
3	2	8.76	6.7
4	3	11.56	8.97
5	4	10.47	7.08
6	5	8.38	6.04
7	6	9.27	6.89
8	7	8.42	7.02
9	8	7.62	5.03
10	9	8.73	6.47
11	10	9.46	6.08

图 2

#### 4.2 构造图表格式

选取文件菜单建新文件功能打开一个新 Excel 工作簿文件,命名为 ph.xls。选取视图菜单工具栏功能激活图表工具栏,按图表指南按钮后,按照指南中给出的 5 个步骤依据 ph.dbf 表格构造出图表。Excel 共有 15 种不同的图表类型可供选择,每种图表类型有 4—10 种图形格式,其图表形式非常丰富,可满足各种图表打印要求。这里假设使用的是折线图第四种格式,其详细的构造过程请参阅 Excel 帮助文件及有关参考书籍。根据 Excel 中打开的 ph.dbf 图表库数据,构图结果如图 3 所示:

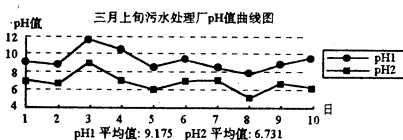


图 3

重要说明:虽然图表模板是根据 ph.dbf 中当前数据得到的,但当 ph.dbf 中的数据变化时,图表模板中的图形和数据可随之动态变化(这是 Excel 的优良特性之一),从而可实现对动态图表数据的处理。在打印每月的报表前,ph.dbf 中的该月数据可在 MIS 客户中使用 .prg 文件得到。

#### 4.3 构造参考数据公式

在图表中,常常需要给出最大值、最小值、平均值等参考数据,如在图 3 中给出了 PH1 和 PH2 的平均值,这些数据不能由图表指南直接得到,而是在图表下面的单元格中运用公式计算得出的,这些数据随着公式中引用的图表库单元格的原始数据而变化,是通过对对打开的 Foxpro 图表库基本数据处理后得到的。Excel 中有很多通用工作表函数用于单元格公式中。例如,计算 PH1 平均值的单元格公式为:

=AVERAGE(ph.dbf! \$B\$2:\$B\$32)

将此公式键入任选的单元格,即可显示出 PH1 平均数据。图表模板中的全部参考数据都可使用这种方法得到。

#### 4.4 保存图表模板

在 Excel 中,图表分为内嵌图表和独立图表。如果图表中不需要给出参考数据,可使用独立图表,否则使用内嵌图表,即直接在工作表中嵌入图表。按所需图表类型建立图表模板后,用文件菜单保存文件功能,按 .xls 文件类型保存文件即可。

#### 5 图表打印服务程序

图表打印服务程序 printdbf() 保存在文件 printdbf.xls 中,以 Visual basic 宏表的形式保存,程序清单如下:

'宏表名:PrintDbf

Sub printdbf()

'以只读方式打开 ph.dbf

Workbooks.Open Filename, = "ph.dbf", ReadOnly, = True

'打开 ph.xls 工作簿

Workbooks.Open Filename, = "ph.xls"

'打印图表,打印份数为 3 份

ActiveWindow.SelectedSheets.PrintOut Copies, = 3

'关闭 ph.xls 工作簿

Windows("ph.xls").Activate

ActiveWorkbook.Close saveChanges, = False

'关闭 ph.dbf

Windows("ph.dbf").Activate

ActiveWorkbook.Close saveChanges, = False

End Sub

#### 6 结束语

本文介绍了利用 Windows 环境下 Foxpro 与 Excel 的 DDE 通讯实现 MIS 系统的图表打印方法,并给出了有关的源程序。利用本文介绍的方法,结合 Excel 卓越的表格处理能力,不仅可用 Excel 服务器实现 MIS 图表打印,而且可实现 MIS 中的全部报表打印。本文的源程序在 Windows 3.2 中文版、Foxpro 2.5B 版、Windows 及 Microsoft Excel 5.0 中文版中运行通过。

#### 参考文献

- 1 美 George F. Goley IV. FoxPro 应用程序开发方法与技巧. 清华大学出版社, 1994. 9
- 2 美 M. Dodge, C. Kinata, C. Stinson Excel5 使用指南. 清华大学出版社, 1995. 1

# Windows 环境下的图像处理——图像旋转

李为民 黄 瑜

梁济仁

(广西软件新技术实验室 南宁 530022)

(广西民族学院物理系)

**摘 要** 本文介绍了在 Windows 环境下,对图像进行 90 度旋转和小于 90 度的任意角度旋转的处理方法,并给出有关程序实例。

**关键词** 图像旋转 设备描述表 全局内存块

## 1 前言

在地理信息等涉及图像处理的系统中,一般都具有图像旋转功能,以便在对图像进行剪裁和拼接前,作一些旋转调整。在对图像进行旋转时,若旋转的角度是 90 度或 90 度的整倍数,如 180 度或 270 度等,可以做到旋转后不产生任何失真(采用附录例程 2 的方法)。但若要对图像作其它任意角度的旋转时,则会产生一定的失真。这里不打算讨论产生失真的原因和补偿方法,本文只是介绍如何将 DOS 环境下一般的图像处理方法,移植到 WINDOWS 环境下,并根据程序实例,分别介绍对图像旋转 90 度和任意角度的编程方法。

为简化问题起见,在下面的讨论中,假定 WINDOWS 的显示方式已设置为 640 \* 480(像点)256 色的模式,要旋转的位图已被调入内存,并且正在屏幕上显示。位图的句柄是 hBitmap,位图所在内存设备描述表句柄为 hMemDC(以下简称“内存 DC”),显示位图的设备描述表句柄为 hDC(以下简称“屏幕 DC”),位图的宽是 bmWidth,位图的高是 bmHeight。

## 2 图像旋转 90 度

这里以图像(以下又称位图)顺时针方向旋转 90 度为例进行讨论。要将位图顺时针方向旋转 90 度,只要将位图每一行的像点,由上到下,依次改为列,并将各列按从右到左的顺序排列,就可以得到一个不失真的顺时针方向旋转 90 度的位图了。要在 WINDOWS 环境下实现这样的操作,可按附录例程 1 的方法处理。对位图旋转 180 度和 270 度(或者逆时针方向 90 度),均可以采用类似的方法处理,或者干脆通过多次旋转 90 度的操作来实现。

在例程 1 的标号 1 的循环中,是逐行逐点采用 GetPixel(hMemDC, x1, y1)来获得内存 DC 上坐标为 (x1, y1)的像点的 RGB 色彩值,再用 SetPixel(hDC, y1, newx - x1, dcolor)来将该 RGB 值置于顺时针旋转 90 度后在屏幕 DC 的对应位置上。这里只是利用屏幕 DC 作为存放旋转位图数据的中间缓冲区,以减少内存开销。该循环完成了整个位图全部像点的旋转工作。

位图旋转 90 度后,其高和宽的值都改变了,原来的高度变成了宽度,原来的宽度则变成了高度(正方形的位图则不变)。故必须如标号 2—6 的语句那样,删除旧的位图,并以新的宽高值重建位图。由于刷新屏幕用的位图是必须放在内存 DC(hMemDC)内的,故还得将旋转后的位图从屏幕 DC 拷贝到内存 DC,这由标号 7 的语句来实现。标号 8 和 9 的语句则清除屏幕的旧内容,再用内存 DC 的内容刷新屏幕。

例程 1 的处理方法程序较简单直观,但所用的函数 GetPixel()和 SetPixel()函数不但处理速度慢,还有颜色匹配的问题。SetPixel(HDC, int, int, COLORREF)函数在设置指定点的像素值时,是把 COLORREF 映射成 DC 中彩色调色板当前最近似的颜色值。当被旋转的位图的颜色只有红、绿、蓝、黄、紫、白、黑等纯色时,用上述方法旋转后的位图颜色仍是正确的。但对于其它由不同层次的丰富颜色构成的彩图(如照片等),旋转后就不能得到正确的颜色(图案的形状是对的,但图案的颜色却变了)。要想在这种情况下得到正确的颜色,可用 MoveTo()和 LineTo()函数来取代 SetPixel()函数,具体方法可参阅文献 1。例程 1 的另一问题是由于采用屏幕 DC 作为旋转图像的中间缓冲区,故所处理的图像尺寸,最大不能大于屏幕的高度。

为较好的解决上述问题,可采用例程 2 的处理方法。在例程 2 中,旋转操作不在设备描述表句柄 hMemDC 和 hDC 上进行,而直接在内存缓冲区操作。这样一来,用指向内存缓冲区的指针 \*lpmem 取得的每个像点数据就是该像点的索引值,不必重新进行颜色匹配,既提高了速度,又保证不变色。

在例程 2 中,标号 1—6 的语句用来取得旋转位图的尺寸和扫描行宽度等有关参数,并以此来计算出新位图所需的存储空间(注意:位图所需空间不是简单的“宽 \* 高”,而应是“扫描行宽 \* 高”,而扫描行宽一般应是四字节的倍数),然后分配一个全局内存块 hMem1 作为中间缓冲区并锁定,以作暂时存放旋转后的位图数据用。标号 7—11 的语句取得被旋转位图的有关参数,据此分配另一个全局内存块 hmem0 并锁定,然后

将要旋转位的图像点数据拷贝到内存块 hmem0 中。标号 12 的循环,则将 hmem0 内的位图数据作顺时针旋转 90 度的变换后,放到中间缓冲区 hmem1 中,标号 13 及以后的语句则删除旧位图,以新的尺寸重建位图,并将新位图的位设置成旋转后的位值(见标号 14 语句),然后将新位图选入内存 DC(即 hmemDC),最后解锁并释放内存,刷新屏幕。

读者试验上述程序时将会看到,用例程 1 的方法来旋转位图时,可能会看到屏幕被一行一行刷新的过程,处理速度很慢。用例程 2 的方法来处理,则可以在瞬间完成整幅图的旋转操作,速度比例程 1 快数十倍。

### 3 图像旋转任意角度

图像旋转任意角度(小于 90 度)的方法,可参考例程 3。它采用与例程 2 相同的内存处理方法。例程 3 对每一个像点旋转后的位置,是通过三角函数的投影变换来获得的,这就不可避免地会产生空洞误差。其表现形式是旋转后得到的位图,会有空洞麻点或花点。这对于白色背景的线条图案(如工程图,黑白地形图等),视觉上影响不太大。但对于象照片这一类的彩图,视觉效果就不太好,还必须作插值和平滑处理才行。本例程可能仅适于对位图作几度以下(乃至零点几度)的小旋转调整。幸好这倒适用于拼接图形前,常要用到的对位图的小角度微调校正,弥补了有些产品软件不能做小于 1 度旋转的缺陷。

在例程 3 中,笔者将顺时针方向旋转的角度(用  $q_1$  表示)定为正值,而将逆时针方向旋转的角度定为负值。即用户输入逆时针方向旋转的角度值时,要在数值前加上“-”号。例程 3 标号 1 的语句中的  $q_1$  就是用户要输入的旋转角度值。现以逆时针旋转 15 度为例,介绍例程 3 的工作原理。

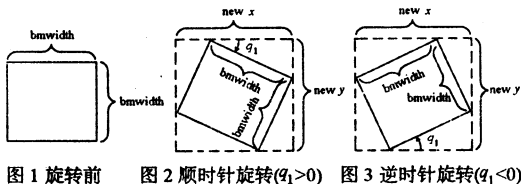


图 1 旋转前 图 2 顺时针旋转( $q_1 > 0$ ) 图 3 逆时针旋转( $q_1 < 0$ )

图 1、图 2 和图 3 分别表示位图旋转前后的各参数所表示的意义。其中  $bmWidth$  和  $bmHeight$  分别为旋转前位图的宽和高,  $newx$  和  $newy$  分别为旋转后新位图的宽和高,  $q_1$  为要旋转的角度。现假定用户要旋转的角度是逆时针方向 15 度,于是标号 1 的语句就成:  $q_1 = (\text{float})(-15)$ 。标号 2 的语句用来将角度换算成弧度(注意到每弧度  $= 360 / (2 * 3.14)$ )。标号 3 和 4 的语句计算顺时针旋转后新位图的宽和高,标号 5 和 6 的语句计算逆时针旋转 15 度后新位图的宽和高。同例程 2 相似,标号 7—10 的语句获得新位图所需的尺寸参数并分配一个全局内存块  $hmem1$  作中间缓冲区。标号 11—15 语句则获得要旋转的位图的一个拷贝,并

放到另一个全局内存块  $hmem0$  中。标号 16 的循环则是以位图中心为旋转轴,对内存块  $hmem0$  中各像点作逆时针旋转 15 的变换后,放到中间缓冲区  $hmem1$  中。标号 17 及以后的语句则是删除旧位图,重建新位图,解锁并释放内存,刷新屏幕,从而完成旋转操作。

读者可将本附录的程序 `rotate.c` 加到自己的 Windows 工程文件(project)中,并在自己的主程序中增加图像旋转的菜单项,再在各菜单项的分支语句中调用 `rotate.c` 内相应的图像旋转函数即可。注意,在调用例程 1 及例程 2 或例程 3 前后,均应分别执行  $hDC = \text{GetDC}(hWnd)$  和  $\text{ReleaseDC}(hWnd, hDC)$  两函数,以获取和释放屏幕 DC。

```

附录(图像旋转源程序)
//rotate.c 96.3.4
#include<windows.h>
extern WORD bmWidth, bmHeight;
extern HDC hMemDC;
extern HBITMAP hBitmap;
//例程 1(顺时针旋转 90 度)
void rotate90R0(HDC hDC)
{
    int y1, x1, newx, newy;
    DWORD dcolor;
    newx = (int)bmWidth;
    newy = (int)bmHeight;
    for(y1 = 0; y1 < newy; y1++) for(x1 = 0; x1 < newx; x1++) [1]
    {
        dcolor = GetPixel(hMemDC, x1, y1);
        SetPixel(hDC, y1, newx - x1, dcolor);
    }
    DeleteObject(hBitmap); [2]
    bmWidth = newx; [3]
    bmHeight = newy; [4]
    hBitmap = CreateCompatibleBitmap(hDC, bmWidth, bmHeight); [5]
    SelectObject(hMemDC, hBitmap); [6]
    BitBlt(hMemDC, 0, 0, bmWidth, bmHeight, hDC, 0, 0, SRCCOPY); [7]
    PatBlt(hDC, 0, 0, 640, 480, WHITENESS); //清屏 [8]
    BitBlt(hDC, 0, 0, bmWidth, bmHeight, hMemDC, 0, 0, SRCCOPY); [9]
}
//例程 2(顺时针旋转 90 度)
void rotate90R1(HDC hDC)
{
    HBITMAP hBitmap1;
    int y1, x1, newx, newy;
    DWORD dsize1, dsize0;
    BITMAP bm0, bm1;
    HANDLE hmem0, hmem1;
    BYTE huge * lpmem0, huge * lpmem1;
    newx = (int)bmWidth;
    newy = (int)bmHeight;
    hBitmap1 = CreateCompatibleBitmap(hDC, newx, newy); [1]
    GetObject(hBitmap1, sizeof(BITMAP), (LPSTR)&bm1); [2]
    DeleteObject(hBitmap); [3]
    dsize1 = (DWORD)bm1.bmWidthBytes * bm1.bmHeight; [4]
    hmem1 = GlobalAlloc(GMEM_MOVEABLE, dsize1); [5]
    lpmem1 = (BYTE huge *) GlobalLock(hmem1); //锁定内存 [6]
    GetObject(hBitmap, sizeof(BITMAP), (LPSTR)&bm0); [7]
    dsize0 = (DWORD)bm0.bmWidthBytes * bm0.bmHeight; [8]
    hmem0 = GlobalAlloc(GMEM_MOVEABLE, dsize0); [9]
    lpmem0 = (BYTE huge *) GlobalLock(hmem0); //锁定内存 [10]
    GetBitmapBits(hBitmap, dsize0, lpmem0); [11]
    for(y1 = 0; y1 < bmHeight; y1++) for(x1 = 0; x1 < bmWidth; x1++) [12]
    {
        * (lpmem1 + (DWORD)(x1 + 1) * bm1.bmWidthBytes - (y1 + 1)) =
        * (lpmem0 + (DWORD)y1 * bm0.bmWidthBytes + x1);
    }
    DeleteObject(hBitmap); [13]
    bmWidth = newx;
    bmHeight = newy;
    hBitmap = CreateCompatibleBitmap(hDC, bmWidth, bmHeight); [14]
    SetBitmapBits(hBitmap, dsize1, lpmem1);
    SelectObject(hMemDC, hBitmap);
    GlobalFree(hmem0); //解锁内存
    GlobalUnlock(hmem1); //释放内存
    GlobalFree(hmem1);
    PatBlt(hDC, 0, 0, 640, 480, WHITENESS); //清屏
    BitBlt(hDC, 0, 0, newx, newy, hMemDC, 0, 0, SRCCOPY); //刷新屏幕
}
//例程 3(任意角度旋转)
int rotateany(HDC hDC)
{
    #include<math.h>
    HBITMAP hBitmap1;
    BITMAP bm0, bm1;
    HANDLE hmem0, hmem1;
    BYTE huge * lpmem0, huge * lpmem1;
    DWORD dsize0, dsize1;
    int newx, newy, dx, dy, x, y, i, j, centerx, centery;
    float q1, a, b, cosq1, sinq1;

```

# 基于超文本技术的查询系统

崔大勇 聂发如 薛 智 毛静涛

(上海电力学院 21 # 200090)

**摘 要** 本文讨论了超文本技术的特点,结合一个校园查询系统,介绍了基于超文本技术查询系统的开发。针对一些关键问题,提出了解决办法,并给出了相应的程序。

**关键词** 超文本 查询系统 开发

## 1 引言

学校校园查询系统是一个集图像,文字,声音为一体的查询系统,由于该系统涉及的信息量大,并且各类信息之间又相互关联,而且关系复杂,又涉及图像及声音,于是采用什么方法进行开发便是首先需要解决的问题。

## 2 超文本技术的概念

当系统涉及的信息量相当庞大时,原有的信息存储和检索方法便已越来越不适应信息的方便应用,这主要是原有的方法不能通过联想来明确信息内部的关联性,而只能单纯地、线性地用字符或者数值式的格式化文本表示。所谓超文本技术就是类似人的联想记忆结构,它采用非线性的网状结构组织块状结构信息,把文本按其内部固有的独立性和相关性划分为不同的信息状,然后,再按它们的自然关系连接成网络。因此,超文本技术就是对信息的非线性表示和检索。

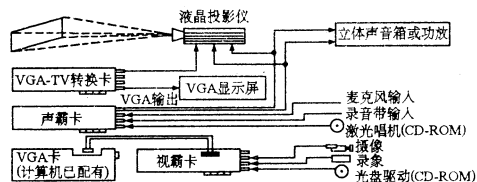


图1 学院介绍多媒体演示系统硬件配置

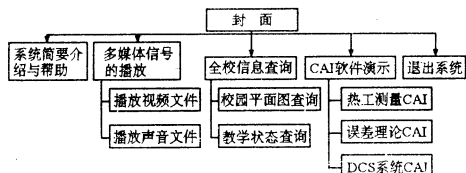


图2 学院介绍多媒体演示系统的主程序框架

Windows 的帮助文件就是一种典型的超文本文件。在 Windows 的帮助文件中有许多所谓的“热点”,而“热点”的作用就是用于进行浏览,当然“热点”中还可以包容“热点”。如果,采用这种办法把各种相互关联的信息组成一条条的链,并且赋予“热点”,那么,便可以实现非线性的浏览。由于,在 Windows 的帮助系统中还可以包含图片,声音和视频信息,因此,便可以利

用 Windows 的帮助系统来实现校园查询系统。

## 3 开发 Windows 帮助系统的几个关键问题

从以上讨论看,基于超文本技术实现校园查询系统可以通过开发 Windows 的帮助系统来完成。以下将讨论开发 Windows 帮助系统的几个关键问题。

### 3.1 步骤专题源文件的编辑

专题源文件是帮助信息的所在之处,是帮助系统的主体,通常由一个个的专题组成。专题内可以包含文本和图形“热点”。文本“热点”由带下划线或者虚线的绿色文字表示,通过对其单击,便可以跳转至别的专题,或者弹出附加的窗口进行说明。以上必须用特殊的编码加以指出,RTF 为编码的格式,编辑可使用支持 RTF 格式的文本编辑器,如 Microsoft Word 进行。

在步骤专题源文件的编辑中将碰到如下的问题:

(1)指定“热点”

“热点”分为跳转和弹出两种。跳转“热点”是指当单击“热点”后,系统便会清除当前的屏幕,并且新开一个窗口以显示与此“热点”相关联的专题内容。弹出“热点”是指当单击“热点”后,系统便会弹出一个附加的窗口以显示与此“热点”相关联的专题内容。跳转“热点”在帮助专题源文件中是带双划线的文本,弹出“热点”在帮助专题源文件中则是带单划线的文本。

编辑“热点”的方法为:首先,把将成为“热点”的字符串高亮度显示;然后,从“格式”菜单中选择“字体”选项,并在字体设置对话框把字符串设定为单或者双划线格式;最后,在该“热点”字符串中键入一个标识字符串(上下文字符串),以标明在该“热点”后将跳转或者弹出到哪一个专题。注意:字符串应设定为隐藏格式;另外,“热点”字符串与上下文字符串之间不能有空格。

每级专题之间采用分页符隔开,在下一级专题前应加入脚注“#”。加入脚注“#”的方法为:从插入菜单中选择“脚注和尾注”选项,然后在“自定义脚注”框中键入“#”。脚注加入后,屏幕上便会出现脚注页面,再在该页面中的“#”脚注后输入上下文字符串(其中不能包含有中文字符),这就形成了脚注与上下文字符串的相关性,即连接上下级专题的链。当然,一旦汇编成帮助系统文件后,脚注“#”和上下文字符串都将不会

出现在屏幕上。

#### (2) 指定专题名,关键字和浏览序号

在帮助系统源文件中指定专题名,关键字和浏览序号的作用是在用户按下帮助系统的“搜索”按钮时,显示关键字和专题。另外,也可用于“历史对话框”,记录已浏览过的专题。

指定专题名的方法是在专题第一行的行首插入脚注“ $\pi$ ”,并在脚注页面内键入专题名;而关键字的指定则是插入脚注“K”。浏览序号用于指定浏览专题的顺序,供用户使用“ $\langle$ ”和“ $\rangle$ ”按钮前后翻页浏览各专题的内容。指定浏览序号的方法是在专题的第一行的行首插入脚注“+”,并在脚注页面内键入浏览序号。

#### (3) 使用图形和图像

可以在帮助系统源文件中使用位图图形文件(.BMP),以及已经“热点”编辑器编辑过的带“热点”的位图文件(.SHG)。在校园平面查询系统中,出现已有“热点”编辑器编辑过的校园平面图,那么,图中每个建筑物都是一个“热点”。这时,只要在需要插入位图的地方加入下列语句之一,便可以在专题中加入位图索引。

bmc 图形文件名:将位图作为一个字符处理,行高由位图决定;

bml 图形文件名:将位图放在一行的左边,正文可在右边卷绕;

bmr 图形文件名:将位图放在一行的右边,正文可在左边卷绕。

使用 Visual Basic 3.0 提供的“热点”编辑器可以编辑位图文件,以使其中的某个部分成为“热点”,方法是将鼠标拖到图形上需要成为“热点”区域的左上角,按住鼠标并将其拖至区域的右下角再放开,这样,该区域便成为了“热点”。接着,在“编辑”菜单中选择“属性”项,并在属性对话框中指定相应的上下文字符串和“热点”类型即可。

#### (4) 创建不可滚动的标题区域

在编辑每一个主题时,第一行通常是专题的标题,可以将其设置为不可滚动区域。当正文滚动时,标题不动,总是保持在第一行;另外,赋予标题与正文区域不同的颜色,以使标题更加醒目。

具体的创建是将光标移至该行行首,然后从“格式”菜单中选择“段落”项,并在段落格式对话框中选择“与下段同页”复选框。这样,编译后文件中的这一行便成为不可滚动的标题区域。

### 3.2 建立帮助项目工程文件

帮助项目源文件提供编译程序所需要的对帮助专题源文件编译的信息,它是一个纯文本文件(扩展名为.HP)。帮助项目源文件中可以包含多个字段,字段被用[]括起来,字段中可以有多个语句。[FILES]字段是必不可少的,它用于指定帮助文件所需的一个或者多

个专题源文件。[OPTION]字段中的 TITLE 选项用于指定帮助窗口的标题,[CONFIG]字段用于指定创建帮助文件中使用的宏,[WINDOWS]字段用于定义帮助系统使用的标题,位置,大小和颜色等。如下为校园平面查询系统中使用的工程文件。

```
[OPTIONS]
WARNING=3
Compress=TRUE
title=上海电力学院校园平面查询
[FILES]
C:\WINWORD\XIAOYUAN.RTF
[WINDOWS]
main="上海电力学院校园平面查询",(0,0,1023,1023),,(192,192,192)
[CONFIG]
CreateButton("btn_Exit","退出查询 &X","Exit()")
SetHelpOnFile("help1.hlp")
AddAccelerator(0x1b,0,"Exit()")
```

其中,[OPTION]字段中 WARNING 语句指明了在编译过程中显示的错误信息等级,WARNING=3 表示报告所有的错误和警告。Compress 语句指定了在编译过程中对数据文件的压缩选项,当 Compress=TRUE 实行压缩,否则不实行压缩。Title 语句设置了当启动帮助系统后帮助窗口标题栏中显示的标题。

[FILES]字段指定了创建帮助文件所需的专题源文件为 XIAOYUAN.RTF。

[WINDOWS]字段指定了主窗口的标题为“上海电力学院校园平面查询”,并且明确了帮助窗口的左上角坐标为(0,0),窗口的高度和宽度均为 1023,标题区的颜色为深灰色。

[CONFIG]字段指定了帮助系统中使用的宏的名称。CreateButton()用于在按钮栏中增加一个新的按钮,该按钮为“退出查询”,作用是执行 Exit()宏指令,即退出帮助系统。SetHelpOnFile 指定了当用户按下 F1 键后调出的帮助文件。AddAccelerator()宏指令把一个 Help 宏指令赋予了一个加速键,即把 Exit()退出帮助系统的宏指令赋予了 ESC 键,因此,当用户按下 ESC 键后就可以退出帮助系统。

### 3.3 编译帮助源文件

编译帮助系统可以使用 Microsoft 公司的帮助文件编译器 HC31。具体使用的方法为,在 DOS 提示符下键入 HC31 帮助项目工程文件名即可。如:

```
HC31 xiaoyuan.hpj
```

编译时,如遇到错误,HC31 会给出各种错误和警告提示,以修改 rtf 或者 hpj 文件。如无错误,就可以生成一个与帮助项目源文件同名,而扩展名为 .hlp 的文件,即 Windows 帮助系统可以调用的二进制数据文件。

## 4 结论

(下转第 50 页)

# 通用辅助录入技术的实现

唐开山

(绍兴文理学院计算机系 312000)

**摘 要** 本文针对数据库应用系统中需要录入大量数据的特点,分析了辅助录入技术的设计思想和实现方法,较好地实现了快速录入数据的目的。

**关键词** 辅助录入 辅助录入信息 代码

## 1 引言

在数据库应用系统中,经常需要录入大量数据,而对每个数据库中的许多字段内容来说,其数据内容在一定范围内是固定的。如档案管理中的性别、民族、政治面貌、婚否、学历、专业、职称等等,这样就可通过选择事先准备好的辅助录入信息来录入这些数据,实现辅助录入的目的,以提高数据录入的速度。

为利用辅助录入信息尽快地录入数据,使在录入过程中尽可能减少以至于不出现与辅助录入信息无直接关系的击键。这就需要在有辅助录入内容的数据录入时,自动出现辅助录入信息,(击一键)选择所需的辅助录入信息而完成相应的数据录入。笔者利用 FoxPro 中的 INKEY() 函数,即时判断录入状态,使辅助录入击键一次予以实现。极大地提高了录入速度。

## 2 设计思想

FoxPro 中的 INKEY() 函数是返回用户所击键的 ASCII 码,用 INKEY() 函数,在录入数据时,根据该函数检测到的值,判断录入数据当时的状态,区分当前录入数据的字段是否有辅助录入内容,当有辅助录入内容时,对自动调入(显示)辅助录入信息,选择对应的辅助录入代码即录入所需的辅助录入信息,这时若需要录入辅助录入信息以外的个别数据,只要录入数据的第一个字符与辅助录入代码不同,则按一般的数据录入处理;若当前录入数据的字段没有辅助录入内容时,则按一般的数据录入处理。这样,不管当前是否有辅助录入内容,既可方便地录入辅助录入信息,也可录入其它数据,不附加任何操作。

## 3 有关数据库结构的设计

为使辅助录入功能具有易修改、易扩充、适应性强等特点,有关数据库结构均按通用性要求设计。

### 1) 主数据库结构

主数据库用来存放需要录入的数据,各字段按应用系统的需要而定。

### 2) 辅助信息库结构

XMM 字段:(C)用来存储主数据库中各字段的

文意义

GS 字段:(C,1)用来存储主数据库中各字段的录入格式

CD 字段:(N,3)用来存储主数据库中各字段的宽度

DMK 字段:(C,4)用来存储主数据库中有辅助录入内容字段的辅助录入信息代码库文件名

该数据库的记录条数与主数据库中字段个数相同。

3) 辅助录入信息代码库结构(文件名由辅助信息库中的 DMK 字段内容决定,数据库个数与主数据库中需要辅助录入数据的字段个数相同)

DM 字段:(C,1)用来存储由 DMK 内容所对应的主数据库中某一字段的辅助录入信息的代码

XM 字段:(C)用来存储对应于 DM 字段的辅助录入信息,其宽度与 DM 字段所指的主数据库中某一字段的宽度相同。

各数据库的记录条数由各所需的辅助录入信息的个数决定。

## 4 实现方法

下面的 FZLR. PRG 是主数据库(RSK. DBF)信息录入主程序,XS. PRG 是辅助录入信息显示模块。

辅助信息库文件名为:RSK0. DBF,辅助录入信息代码库文件名为:DM?? .DBF,??为主数据库字段顺序号。

主程序中用

P1=INKE(0)

来检测数据录入状态,用

P2=INKE()

来获取第二键值,以便处理汉字之类字符的输入。根据 P1 的值通过 KEYBOARD 命令向键盘缓冲区塞入辅助录入信息(需辅助录入时)或键入的字符串,由 GET、READ 命令接收,然后存入主数据库。

在辅助录入时可用 PageUp、PageDown 键对有许多项辅助录入信息时进行翻页,若记得未显示页的辅助录入信息代码,也可直接键入对应的代码录入辅助录入信息。



```

*****
* FZLR.PRG *
*****
SET TALK OFF
SET COLO TO 7+/0
CLEA
SET COLO TO .7/0
@ 0.28 CLEA TO 2.51
SET COLO TO 7+/4
@ 1.33 SAY "基本信息录入"
SELE 1
USE RSK0
D=RECC()
DIME JB(D).S(D.2).JA(D).Y(D.100)
Y=" "
S=0
I=1
DO WHILE I<=D
SELE 1
GO 1
J=RIGH(STR(100+I.3).2)
SET COLO TO 7+/3
@ 1+3.4 SAY RTRI(XMM)
IF GS="D"
JA(I)="/"
X&J="D"
ELSE
JA(I)=SPAC(CD)
X&J=REPL(GS,CD)
ENDI
IF DMK<>" "
DEF=DMK
SELE 2
USE &DEF
D1=RECC()
S(1.1)=D1
S(1.2)=LEN(XM)
J=1
DO WHILE J<=D1
GO J
Y(1.2+J-1)=DM
Y(1.2+J)=XM
J=J+1
ENDD
ENDI
I=I+1
ENDD
SELE 1
USE RSK
I=4
DO WHILE T.
SET COLO TO 7+/1+.7+/1
I=4
J=1
DO WHILE J<=D
JB(J)=JA(J)

```

```

@J+3.24 SAY JB(J)
J=J+1
ENDD
DO XS WITH 1.S(1.2).1
SS=1
DO WHILE T.
P1=0
@ 1.24 SAY " "
DO WHILE P1<1
P1=INKE(0)
P2=INKE(0)
ENDD
DO CASE
CASE P1=24
IF I=D+3
EXIT
ELSE
SS=1
I=I+1
ENDI
CASE P1=5. AND. I=4. OR. P1=27
RETN
CASE P1=5. AND. I>4
I=I-1
SS=1
CASE P1=18
SS=MAX(SS-1.1)
CASE P1=3
SS=MIN(INT((S(I-3.1)-1)/15)+1.SS+1)
OTHE
IF P1>127
KEYB CHR(P1)+CHR(P2)
ELSE
IF S(I-3.1)>0
K=1
DO WHILE CHR(P1)<>Y(I-3.2+K-1)
AND. K<=S(I-3.1)
K=K+1
ENDD
IF K<=S(I-3.1)
KEYB Y(I-3.2+K)
ELSE
KEYB CHR(P1)
ELSE
KEYB CHR(P1)
ENDI
ENDI
C=RIGH(STR(97+I.3).2)
@ 1.24 GET JB(I-3) PICT"@&X&C"
READ
IF I=D+3
EXIT
ELSE
SS=1

```

```

I=I+1
ENDI
ENDC
DO XS WITH I-3.S(I-3.2).SS
ENDD
APPE BLAN
J=1
DO WHILE J<=D
IF JA(J)="/"
JB(J)=CTOD(JB(J))
ENDI
J=J+1
ENDD
GATH FROM JB
ENDD
CLOS ALL
RETN
*****
* XS.PRG *
*****
PARA XX,YY,ZZ
SET COLO TO 7/0
@ 3.50 CLEA TO 24.79
IF S(XX.1)>0
SET COLO TO 7+/2
@ 4.53+INT((YY-8)/2) SAY "辅助录入信息"
ZZ1=MIN(15.S(XX.1)-(ZZ-1)*15)
SET COLO TO .7+/0
@ 6.53.7+ZZ1.57+YY BOX* " "
O=(ZZ-1)*15+1
SET COLO TO .6+/4
DO WHILE O<=ZZ1+(ZZ-1)*15
@ 6+O-(ZZ-1)*15.54 SAY Y(XX.2*O-1)+
"+Y(XX.2*O)
O=O+1
ENDD
ENDI
RETN
5 结束语

```

本辅助录入实现方法操作简单,反应速度较快,通用性强,在FoxBASE下也一样适用。这些技术将会大大提高数据库系统的开发质量。

该程序的运行环境为:UCDOS 3.0, FoxPro 2.5 FOR DOS。在 386、486 微机, FoxPro (2.5, 2.6) FOR DOS, FoxBASE 2.1 下运行通过。

(上接第 81 页)

```

q1=(float)(-15)//要旋转的角度 [1]
q1=(float)q1*3.14/(float)180//转化为弧度 [2]
cosq1=(float)cos(q1);
sinq1=(float)sin(q1);
if(q1>0)//顺时针旋转
{
newx=(int)(bmWidth*cosq1+bmHeight*sinq1); [3]
newy=(int)(bmWidth*sinq1+bmHeight*cosq1); [3]
}
else//逆时针旋转
{
newx=(int)((float)bmWidth*cosq1-(float)bmHeight*sinq1); [5]
newy=(int)((float)bmHeight*cosq1-(float)bmWidth*sinq1); [6]
}
hBitmap1=CreateCompatibleBitmap(hDC,newx,newy); [7]
GetObject(hBitmap1,sizeof(BITMAP),(LPSTR)&bm1); [8]
dwsizel=(DWORD)bm1.bmWidthBytes*bm1.bmHeight; [9]
hmem1=GlobalAlloc(GMEM_MOVEABLE,dwsizel); [10]
lpmem1=(BYTE huge*)GlobalLock(hmem1);
DeleteObject(hBitmap1);
GetObject(hBitmap,sizeof(BITMAP),(LPSTR)&bm0); [11]
dwsizel=(DWORD)bm0.bmWidthBytes*bm0.bmHeight; [12]
hmem0=GlobalAlloc(GMEM_MOVEABLE,dwsizel); [13]
lpmem0=(BYTE huge*)GlobalLock(hmem0); [14]
GetBitmapBits(hBitmap,dwsizel,lpmem0); [15]

```

```

centerx=bmWidth/2;centery=bmHeight/2
a=(float)(newx/2+0.5);b=(float)(newy/2+0.5);
for(i=0;i<(int)bmHeight;i++) [16]
{
dy=i-centery;
for(j=0;j<(int)bmWidth;j++)
{
dx=j-centerx;
x=(int)(dx*cosq1-dy*sinq1+a)
if(x<0||x>newx-1) continue;
y=(int)(dx*sinq1+dy*cosq1+b)
if(y<0||y>newy-1) continue;
*(lpmem1+(DWORD)y*bm1.bmWidthBytes+x)=
*(lpmem0+(DWORD)i*bm0.bmWidthBytes+j);
}
}
DeleteObject(hBitmap); [17]
hBitmap=CreateCompatibleBitmap(hDC,newx,newy);
SelectObject(hMemDC,hBitmap);
bmWidth=newx;
hmHeight=newy;
SetBitmapBits(hBitmap,dwsizel,lpmem1);
GlobalUnlock(hmem0);GlobalFree(hmem0);
GlobalUnlock(hmem1);GlobalFree(hmem1);
BitBlt(hDC,0,0,bmWidth,bmHeight,hMemDC,0,0,SRCCOPY); //刷新屏幕
}

```

### 参考文献

- 1 张军等译. Windows 程序及使用指南(四). 清华大学出版社
- 2 周孝宽编. 实用微机图像处理. 北京航空航天大学出版社

# DOS 系统中文件的存盘、删除与恢复技术

罗淑英

(贵州大学管理科学系 贵阳 550025)

杨云江

(贵州大学计算机网络研究所 贵阳 550025)

**摘 要** 本文系统、深入地分析了 DOS 系统内部的磁盘文件管理方式,详细地介绍了在 DOS 系统中文件的建立与存盘、文件的删除与恢复技术。

**关键词** 系统分析 磁盘结构 文件管理

众所周知,DOS 系统是针对文件及文件在磁盘上的读写方式进行管理的一种微机操作系统。在系统内部,DOS 是怎样管理文件的?文件在磁盘上的存放形式又是如何?这是众多计算机爱好者所关注的问题。本文从 DOS 磁盘布局及文件的内部存放格式入手,对 DOS 系统内部文件的建立与存盘,文件的删除与文件的恢复技术进行了系统、深入的分析。

## 1 磁盘布局

在 DOS 系统下,磁盘由四个部分组成:引导扇区(BOOT 区)、文件分配表 FAT、文件目录区及文件数据区。

BOOT 区	文件分配表 FAT	文件目录区	文件数据区
--------	-----------	-------	-------

其中,BOOT 区(又称引导块),固定存放在磁盘的第 0 面 0 道 1 扇区上。

文件分配表固定从磁盘的逻辑扇区 1 扇区上开始存放,文件目录区及数据区的存放位置则是根据磁盘容量大小的不同而不同,例如:360KB 软盘的文件目录区是从 005H 扇区开始,数据区是从 00CH 扇区开始;而 10MB 硬盘的目录区则从 053H 扇区开始,数据区则是从 073H 扇区开始;33MB 硬盘的目录区则是从 0081H 扇区开始,数据区则是从 00A1H 扇区开始。

文件在磁盘上是“以簇”为单位进行读写操作的,一个文件在磁盘上至少占 1 个簇的空间,而无论该文件的长度有多小,即使仅有 1 个字节长度也是如此。当一个文件的长度超过 1 个簇的容量时,就要用两个簇进行存放,超过两个簇容量时,就要用 3 个簇进行存放…。 “簇”的长度对于不同容量的磁盘是不同的,对于 360KB 软盘 1 个簇占 2 个扇区(每个扇区长度 512 字节,无论对软盘还是硬盘均是如此),其长度为 1024 字节;1.2MB 软盘每簇占 1 个扇区,长度为 512 字节;10MB 硬盘每簇占 8 扇区,长 4096 字节;20MB 及 33MB 硬盘的每个簇占 4 个扇区,长 2048 字节。

文件目录区是用来存放文件名字,文件大小等信息

息的一张登记表,其存放的内容有文件名、扩展名、文件长度(以字节为单位),文件建立或修改的日期和时间以及文件存放的首簇号等信息。

文件名	扩展名	文件长度	首簇号	日期	时间
-----	-----	------	-----	----	----

文件数据区是用来存放文件内容,并以簇为单位进行存放。其存放的位置严格与文件目录区的首簇号及文件分配表一一对应。

在访问某一文件时,先从目录区上找到相应的文件目录,根据文件长度可算出该文件共占多少个簇,并从文件目录区的簇号区上找到文件存放的首簇号,根据其文件分配表的链表关系(其链表关系在后面介绍),可找到该文件所有存放的簇号。

## 2 文件分配表

磁盘上的所有簇号与文件分配表严格一一对应,格式化磁盘时,将文件分配表清空,即以全“0”填写之。在文件分配表上,全“0”表示该簇号为空,为可用簇,非“0”时表示该簇已被占用。所以,磁盘的使用情况,只要查看文件分配表便一目了然。为了清晰起见,在本文中以空格表示空簇,注意在实际应用中应以全“0”代之。

设一张 360KB 软盘使用一段时间后,其文件目录区及文件分配表使用情况如表 1 所示,表 1(A)为文件目录区内容,表 1(B)为文件分配表内容。

表 1(A)

文件名	扩展名	文件长度	首簇号	日期	时间
NAME1	COM	7000	002		
NAME2	EXE	3978	015		
NAME3	TXT	745	010		

表 1(B)

	0	1	2	3	4	5	6	7	8	9
00	标志	003	004	006		007	008	013		
01	FFF		FFF		016	017	018	FFF		

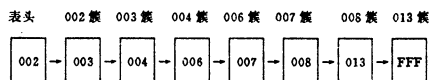
表中“标志”为磁盘标志,360KB 软盘标志为“FD”,1.2MB 软盘标志为“F9”,硬盘标志为“F8”。

文件 NAME1.COM 的文件长度为 7000 字节,共需 7 个簇存放,其簇号依次为 002,003,004,006,007,

008,013;002 为首簇簇号。文件 NAME2.EXE 的文件长度为 3978 字节,共需 4 个簇存放,其簇号依次为 015, 016, 017, 018; 015 为首簇簇号,文件 NAME3.TXT 的长度为 745 字节,需一个簇存放,簇号为 010。

实际上,文件分配表是以链表的形式存放的,该表的每个结点仅由一个指针域组成,用以指出下一簇号位置,文件目录相当于链表表头结点,其表头结点指针指向“首簇簇号”结点。文件 NAME1.EXE 的链表存储形式如表 1(C)所示。

表 1(C)



### 3 文件的存盘

在此,仍以 360KB 软盘上文件的存取为例,先介绍文件在磁盘上的读写操作过程。我们知道,在 360KB 软盘上,每个簇占 2 个扇区,即每个簇长度为 1024 字节。

设有如下两个文件:

文件名(含扩展名)	文件长度	所需扇区数	所需簇数
FILE1.TXT	405 字节	1 个	1 个
FILE2.TXT	4000 字节	8 个	4 个

对于存放长度不大于 1 个簇存储空间的文件 FILE1.TXT,先从文件分配表上找到第一个空簇簇号 005,将该文件的文件名、扩展名、文件长度、当前日期、时间以及首簇簇号 005 一道(共 32 字节长)存放在文件目录区上(若盘上无可用簇号,表示磁盘空间满,转出错处理);再将文件内容存放到文件数据区 005 簇对应的扇区上,最后将 005 簇簇号在文件分配表上对应的位置上填上结束标志“FFF”。

若存放的文件长度大于 1 个簇的存储空间,比如文件 FILE2.TXT 需要 4 个簇,先从文件分配表上找出 4 个可用簇,即 009,011,012,014。(若盘上无 4 个可用簇号,表示磁盘空间满,转出错处理)。先将该文件的文件名、扩展名、文件长度、当前日期、时间以及首簇簇号“009”一道存放在文件目录区上,再将文件内容依次存放到 009,011,012,014 簇对应的扇区上,最后将 009,011,012 三个簇簇号在文件分配表上对应的位置分别填上“011,012,014”,将 014 簇簇号在文件分配表上对应的位置填上簇结束标志“FFF”。如表 2(A)、表 2(B)所示。

表 2(A)

文件名	扩展名	文件长度	首簇号	日期	时间
NAME1	COM	7000	002		
NAME2	EXE	3978	015		
NAME3	TXT	745	010		

FILE1	TXT	475	005		
FILE2	TXT	4000	009		

表 2(B)

	0	1	2	3	4	5	6	7	8	9
00	标志	003	004	006	FFF	007	008	013	011	
01	FFF	012	014	FFF	FFF	016	017	018	FFF	

这里特别值得一提的是,系统在文件分配表上查找空簇时,每次总是从头至尾进行,因此,给每个文件分配的簇号总是从小到大排列的,这一点,也为文件被删除后对其进行恢复提供了方便。

### 4 文件的删除

若欲删除长度不大于 1 个簇存储空间的文件,例如将上述文件 FILE1.TXT 删除,先在文件目录表中将文件名 FILE1.TXT 的第一个字符“F”改为“σ”(ASCII 码 E5),σ 为文件删除标记,表示该文件已被删除,再将 005 簇簇号在文件分配表上对应的位置上填上簇空标志“000”(在此以空格代之)。

若欲删除的文件长度大于 1 个簇的存储空间,例如在删除了文件 FILE1.TXT 后再将文件 NAME2.TXT 删除,先在文件目录表中将文件名 NAME2.EXE 的第一个字符“F”改为“σ”,再将 015, 016, 017, 018 四个簇簇号在文件分配表上对应的位置上填上簇空标志“000”(在此仍以空格代之)。FILE1.TXT, NAME.EXE 两个文件被删除后,目录区及文件分配表的情况如下表:

表 3(A)

文件名	扩展名	文件长度	首簇号	日期	时间
NAME1	COM	7000	002		
σAME2	EXE	3978	015		
NAME3	TXT	745	010		
σILE1	TXT	475	005		
FILE2	TXT	4000	009		

表 3(B)

	0	1	2	3	4	5	6	7	8	9
00	标志	003	004	006		007	008	013	011	
01	FFF	012	014	FFF	FFF					

### 5 文件的恢复

由上述可知,DOS 系统在删除文件时,只是将文件目录的第一个字符改为删除标志“σ”,并将文件分配表对应的链指针数据断开,即将其清空,而文件数据区的数据(文件内容)并没有被删除,还完整地保存在磁盘上,文件长度及文件存放的首簇号也完好地保留在文件目录中,这就为文件的恢复工作提供了有利的条件。当然,并不是所有被删除的文件都能完整的恢复,主要视 DOS 系统的版本及文件的大小和文件存放在磁盘上的位置而定。

对于低版本的 DOS 系统,分三种情况加以讨论。

1)、当一个文件的长度不超过 1 个簇时,该文件被删除后是很容易恢复的,例如,欲恢复上述文件 FILE1.TXT,第一步,恢复文件目录,即将文件目录中“FILE1.TXT”的第一个字符“σ”换成“F”,第二步,恢复文件分配表,从该文件目录中的簇号区中取出其首簇簇号“005”,在文件分配表上,将 005 簇对应的位置填上簇结束标志“FFF”,文件即恢复完毕。

2)、当文件的长度超过 1 个簇,其存放在磁盘上的簇号是连续的,这一类文件被删除后,也是可以完整恢复的。例如,对文件 NAME2.EXE,其文件存放的 4 个簇号是连续的,即 015,016,017,018,恢复时,先恢复文件名,即将文件目录中“NAME1.EXE”的第一个字符“σ”换成“N”,再从该文件目录中的簇号区上取出其首簇簇号“015”,在文件分配表上,从 015 开始的连续 4 个簇上依次填上 016,017,018,FFF,文件即恢复完毕。如下表所示:

表 4(B)

	0	1	2	3	4	5	6	7	8	9
00	标志		003	004	006	FFF	007	008	013	011
01	FFF	012	014	FFF	FFF	016	017	018	FFF	

3)、当文件的长度超过 1 个簇,其存放在磁盘上的簇号是不连续的,例如文件 NAME1.COM,其文件长度占 7 个簇,依次为 002,003,004,006,007,008,013 七个簇号,当该文件被删除后,能否进行完整的恢复要视具体情况而定。下面分三种情况讨论:

(1)当盘上只有文件 NAME1.COM 被删除时,可将其进行恢复,先恢复文件名,即将文件目录中“NAME1.COM”的第一个字符“σ”换成“N”,再从该文件目录中的簇号区上取出其首簇簇号“002”,从文件分配表上的 002 簇开始找出 7 个空簇,簇号为 002,003,004,006,007,008,013,依次填上 003,004,006,007,008,013,FFF,文件即恢复完毕。

(2)当磁盘上被删除的文件除 NAME1.COM 外,还有其它文件(如 NAME2.EXE)也被同时删除,但后者的存放簇号不与前者的存放簇号交错,在这种情形下,文件 NAME1.COM 也是可以恢复的。其恢复方法同上。

(3)当磁盘上被删除的文件除 NAME1.COM 外,还有其它文件(如 FILE1.TXT)也被同时删除,删除文件后文件分配表情况如下表 5(B):

表 5(B)

	0	1	2	3	4	5	6	7	8	9
00	标志									011
01	FFF	012	014		FFF	016	017	018	FFF	

由于后者的存放簇号与前者的存放簇号是交错的,在这种情形下,文件 NAME1.COM 是不能进行完整的恢复的。若要用 PCTOOLS 的 U 命令对其进行强

行恢复时,则 002,003,004,005,006,007,008 这七个簇号将作为该文件的存放簇号,恢复后的文件分配表如下表 6(B):

表 6(B)

	0	1	2	3	4	5	6	7	8	9
00	标志		003	004	005	006	007	008	FFF	011
01	FFF	012	014		FFF	016	017	018	FFF	

从表 6 可看出,恢复后的 NAME1.COM 文件是不完整的,因其一部分内容已被文件 FILE1.TXT 的内容所代替,而有一部分内容丢失。

注意:在低版本 DOS 系统下,当磁盘文件被删除后,该磁盘没有写入新的数据之前,才有可能对其进行恢复,否则是无法进行恢复的。

在高版本的 DOS 系统(6.2)下,可以使用 UNDELETE 程序来恢复被删除的文件,文件能否被完全恢复,要视 DOS 系统的保护等级设置而定。文件的保护等级用 UNDELETE 程序来设定,由高到低分为三等:①站岗级,这是对文件最高级别的保护,当某一个盘的保护等级设置为站岗级后,删除该盘上的一个文件,系统先将文件目录区中该文件名的第一个字符换成“σ”,同时将该文件的内容保存到隐含目录\SENTRY 中,然后将原来存放该文件内容的簇号链表断开。在这种保护级别下,无论被删除的文件内容是否被覆盖,都能将其完整恢复,站岗级需占用部分内存(UNDELETE.EXE 程序常驻内存)和部分磁盘空间(大小可设置)。②追踪级,追踪级是次高级别的保护,当某个盘被设置为追踪级后,若删除该盘上的某个文件,系统先将文件目录区中该文件名的第一个字符换成“σ”,同时将该文件名及存放该文件内容的簇号保存到 PCTRACKR.DEL 文件中,然后将原来存放该文件内容的簇号链表断开。在追踪级保护下,若被删除的文件内容没有被覆盖,则能将其完整恢复,追踪级亦需部分内存和磁盘空间。③标准级,标准级是最低级别的保护,是系统默认的保护等级,不需要额外的内存和磁盘空间,对文件的恢复情况与低版本 DOS 系统下前面讨论的三种情况相同。

### 参考文献

- 1 乌托什编译:MS-DOS 磁盘结构与分析流程.中国科学院希望高级电脑技术公司,1989 年 12 月
- 2 Thom Hogan.The Programmer's PC Sourcebook.Microsoft Corporation,1988
- 3 亦欧,五月译.MS & PC ODS 内存驻留、中断和磁盘管理.中国科学院希望高级电脑技术公司,1990 年 8 月
- 4 李玉林,钱培德著.PC-DOS 系统汉化和 CC-DOS 分析与使用.测绘出版社,1989 年 4 月
- 5 杨云江,罗淑英.一张磁盘究竟能存放多少个文件.微小型计算机开发与应用,1995 年第 3 期
- 6 朱力,常建军著.中西文 MS-DOS 6.2 实用技术

# 间隙软指纹加密技术

李京民 王柏盛 刘瑞英

(河北煤炭建工学院电子系 邯郸 056001)

**摘要** 本文根据磁道格式和扇区格式阐述了间隙软指纹加密技术的原理,指出了制作软指纹的方法,以及用软指纹加密技术保护用户程序的方法。

**关键词** 密钥 间隙软指纹 索引孔 驱动器操作参数表

随着计算机价格的降低,计算机逐步进入家庭,对各种软件的需求越来越大,软件作为商品在市场上流通,软件的开发者总希望自身的劳动成果得到保护,以维护自身的经济利益。为了做到这点,学习一些软件加密技术是十分必要的。流行的加密工具相当多,现在就比较有效的间隙软指纹加密技术(或称仿激光软加密技术)加以介绍。软指纹加密技术实际上就是产生一个密钥,如果要执行用户文件首先要检查这个密钥(即软指纹),只有通过了检查才能执行用户文件,而这个软指纹制作方便而拷贝困难,所以有很强的实用性。现代加密技术不能只用一种,这就要求在了解软指纹加密技术的基础上再辅助其它一些加密措施,才能更有效。

## 0 软磁盘的格式

首先我们以 5.25 英寸双面双密度的软磁盘为例说明磁盘的格式。5.25 英寸盘有一个索引孔位于磁道的开头,孔的两边是前置区与后置区。前置区中包括间隙 GAP4a、GAP1、同步区 SYNC 和索引地址标志 IAM。后置区包括 GPA4b,另外还有 9 个扇区。如图 1 所示。

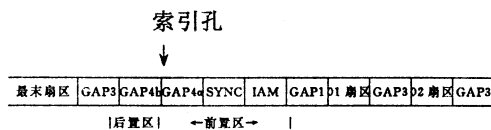


图 1 磁道结构

每个扇区又可分为识别字段区和数据区,如图 2 所示。

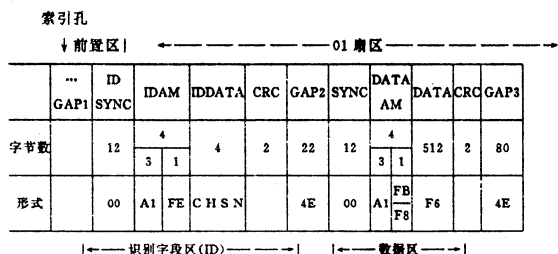


图 2 扇区格式

由图 2 可知识别字段区分为 SYNC、ID、地址标志 IDDATA 和 CRC 四个区。识别字段区的内容是在格式化时一次写入,不能改变,除非重新格式化。

特别值得重视的是 C、H、S、N 四个参数。

C:磁柱面(Cylinder),亦可称磁道,存放该磁道所在编号,其值为 0—38。

H:磁头(Head),存放 0 或 1。

S:扇区(Sector)亦称记录区,存放该扇区的编号,其值为 0—9。

N:数据数目(Number),存放该扇区之数据区中可供使用的数据多少的编号,如 N=1,存放 256 字节数据,N=2 存放 512 字节数据。可用下式计算:

$$\text{每扇区字节数} = 128 * 2^N$$

这四个参数非常重要,它标明了该扇区的识别号码。当驱动器读写数据时,就由此识别字段来辨别是否为该数据存放位置。以上所列出的 C、H、S、N 值均为正常格式化之后所得到的。实际上 C、H、S、N 值可以自由设定,改变 C、H、S、N 值可以格式化一条特殊的磁道。

数据区分为同步区、标志区、数据区和 CRC 四部分。为了适应不同转速,在 ID 区和数据区之间有间隙(GAP2)。对于 5.25 英寸 MFM 方式的磁盘,一个扇区共占 654 字节。

## 1 驱动器操作参数表

利用 ROMBIOS 的 INT 1EH 中断可以指向软驱动器的操作参数表。由 DEBUG 来找到 1EH 的指针,它存放在 0:0078—0:007B 处,指针指向 0000:0522 的存储器位置。如果是 360KB 磁盘,用 DIR 来列软盘上的目录后,它的 11 个参数值为:

—d0:522 52C

0000:7520 DF 02 05 02 09 1B—FF 50 F6 01 08

参数的意义为:

(1)0522H:此位置之参数代表步进马达的速率及磁头卸载时间。

(2)0523H:此位置代表磁头载入时间及数据传输模式。

(3)0524H:此位置之参数指定了马达等待之时间。

(4)0525H:此参数即为 N 值。正常情况下为 2,代表每个扇区个存放 512 字节之数据。

(5)0526H:此位置之参数代表每道扇区数。

(6)0527H:此位置之参数代表读写数据时扇区缓冲区大小。

(7)0528H:此位置之参数为每扇区字节(当 N=0 时)。

(8)0529H:此位置之参数为格式化时扇区缓冲区字节

数。

(9)052AH:此位置之参数为格式化时数据区填充数字。

(10)052BH:此位置之参数为寻道后磁头稳定时间。

(11)052CH:此位置之参数为启动马达的时间。

以上参数有些与机器有关不能随便改动,而有些参数只与操作有关,所以可以改变。(4)、(5)、(6)、(7)、(8)项在格式化时可以改变,作为格式化特殊磁道用。

## 2 INT 13 中断的功能

INT 13 有多种功能调用,其中主要有:

AH=00H	重设驱动器
AH=01H	读驱动器状态
AH=02H	读取扇区用户数据
AH=03H	写数据至扇区
AH=04H	查验扇区是否存在及是否完整
AH=05H	格式化一条磁道
AH=15H	读磁盘类型
AH=16H	读取软盘变化状态
AH=17H	设定软盘类型

有了以上基本知识,我们就可以格式化一条间隙软指纹磁道。为了避免和 DOS 正常磁道相冲突,可选用 28H(40 道)或 29H(41 道)实际的驱动器能读写这两个磁道,但这条磁道一般不写入数据,现在主要是改变 N 参数。

现在我们设 N=06。

用 DEBUG.COM 格式化一条特殊磁道。

A>DEBUG

```
-A100 ;在 A 命令下建立格式化程序
0D4F:0100 MOV AX,1702 ;设软盘类型,用 1.2M 驱动器格式化 360K 磁盘
0D4F:0103 INT13
0D4F:0105 MOV AH,05 ;格式化磁道
0D4F:0107 MOV CH,28 ;CH=磁道号
0D4F:0109 MOV DX,0000;DH=磁头号,DL=驱动器号
0D4F:010C MOV BX,0200;BX 指向 C、H、S、N 起始地址
0D4F 010F INT13
0D4F:0112
-d0000:0522 052C DF 02 25 02 09 1B-FF 50 F6 01 08
-d200 223 ;在 0D4F:0200 设置 ID 信号表:
0D4F:0200 28 00 01 06 28 00 02 06-28 00 03 06 28 00 04 06
28 00 05 06 28 00 06 06-28 00 07 06 28 00 08 06
28 00 09 06 (N 值全改为 06)
```

-G=100 用 G 命令执行格式化程序

以上程序执行后,因为 0000:0525 单元为 02,0000:0526 单元为 09,所以磁道格式化为 9 扇区,每扇区 512 字节。但是磁盘上识别字段的 N 值却写为 06。所以在读取扇区数据时,如果用 INT13 AH=02H 功能来读取数据,0000:0525 单元的 02 必须改为 06,否则读不出数据。利用 DEBUG 读出数据。

```
-A 100
0D4F:0100 MOV AX,0201 ;AH=02,AL=要读的扇区数
0D4F:0103 MOV BX,3000 ;BX 内存输入缓冲区首址
0D4F:0106 MOV CX,2801 ;CX=磁道号,CL=扇区号
0D4F:0109 MOV DX,0000 ;DH=磁头号,DL=驱动器号
0D4F:010C INT 13
0D4F:010E INT 3
0D4F:010F ;此值改为 06
-d 0000:0522 052C
0000:0520 DF 02 25 06 09 1B-FF 50 F6 01 08
```

-g=100;

```
AX=1010 BX=1000 CX=2801 DX=0000 SP=FFEE BP=
0000 DI=0000
DS=0D4F ES=0D4F SS=0D4F CS=0D4F IP=010E NV
UP EI PL NZ NA PO CY
CS:010E CC INT 3 ↑
```

出现读错误

由以上分析可知,因进位位为 CY,说明出现读错误,AH=10,说明出现 CRC 校验错误,虽然如此但仍然能读出来。由此可以知道出现 CY 且 AH=10 是特殊磁道出现读出错误的标志,这种错误不影响读出。

因为 N=6 时一个扇区应为 8192 字节。因此它读出了 8192 个字节。这超过了一条磁道所能容纳的字节数,所以它把磁道尾端及磁道首尾相接处及磁道的引导部分等也读出来了。并存放在以 CS:3000 开始的缓冲区。

让我们看一看读出在缓冲区的部分

```
-d3000
0D4F:3000 F6 F6 F6 F6 F6 F6 F6-F6 F6 F6 F6 F6 F6 F6
```

```
0D4F:3070 F6 F6 F6 F6 F6 F6 F6-F6 F6 F6 F6 F6 F6 F6
```

显示磁道首尾相接部分(即索引孔前后部分)的数据。

因为每扇区 654 字节,因此 9 扇区占 5886(约 1700H)。

```
-d4700
```

```
0D4F:4700 4E 4E 4E 4E 4E 4E 4E-4E 4E 4E 4E 4E 4E 4E
NNNNNNNNNNNNNNNNNNNN
0D4F:4710 4E 4E 4E 4E 4E 4E 4E-4E 4E 4E 4E 4E 4E 4E
NNNNNNNNNNNNNNNNNNNN
```

```
0D4F:4770 4E 4E 4E 4E 4E 4E 4E-4E 4E 4E 4E 4E 4E 4E
NNNNNNNNNNNNNNNNNNNN
```

```
-d
```

```
0D4F:4780 4E 4E 4E 4E 4E 4E 4E-4E 4E 4E 4E 4E 4E 4E
NNNNNNNNNNNNNNNNNNNN
0D4F:4790 4E 4E 4E 4E 4E 4E 4E-4E 4E 4E 4E 4E 4E 4E
NNNNNNNNNNNNNNNNNNNN
```

```
0D4F:47A0 4E 4E 4E 4E 4E 4E 4E-4E 4E 4E 4E 4E 4E 4E
NNNNNNNNNNNNNNNNNNNN
```

```
0D4F:47B0 9C 9C 9C 9C 9C 9C 9C-9C 9C 9C 9C 9C 9C 9C
NNNNNNNNNNNNNNNNNNNN
```

```
0D4F:47C0 9C 9C 9C 9C 9C 9C 9C-9C 9C 9C 9C 9C 9C 9C
NNNNNNNNNNNNNNNNNNNN
```

```
0D4F:47D0 9C 9C 9C 9C 9C 9C 9C-9C 9C 9C 9C 9C 9C 9C
NNNNNNNNNNNNNNNNNNNN
```

```
0D4F:47E0 9C 9C 9C 9C 9C 9C 9C-9C 9C 9C 9C 9C 9C 9C
NNNNNNNNNNNNNNNNNNNN
```

```
0D4F:47F0 9C 9C 9C 9C 9C 00 00-00 00 00 00 00 00 00
NNNNNNNNNNNNNNNNNNNN
```

索引孔位置

由以上看出,3C 为磁道的头尾相接处,其后为磁道的开始处。其原理如图(3)所示。由此图可以看出,在磁道结尾处无法将一个字节连接得非常完整。因此,读出时在索引孔位置,即 47A6H 处的 3C 有几位属磁道尾部,有几位属磁道的开头,这样以后的部分全被偏移了。对于每个磁道,不同软盘或同一软盘两次格式化后,这些数据因驱动器转速不稳定而有变化,故可用首尾相接处的数据作为间隙软指纹。将此索引孔前后几个字节的数据存入指定的数据区。

在磁盘保护程序中,写一段判别程序,此判别程序读出首尾相接的数据与写入数据区的数据比较判别,只有通过此判别,才能运行被加密程序,否则退出运行。

格式化后间隙 GAP4b 的情况。

索引孔

↓磁道开头

```
01001110010011100100111001001110010011100
```

磁道结尾

```
0100111001001110010
```

```
4E 4E
```

读出首尾相接处的情况:

```
010011100100111001001110010011100100111001001110010011100
```

```
4E 4E 49 c9 c9 c9
```

图(3)