



中国计算机学会会刊

ISSN 1001-3695

# 计算机应用研究

APPLICATION RESEARCH OF COMPUTERS

1997 • 3



GBH多媒体查询系统制作工具

- 面向对象多媒体创作
- 制作方便快捷

ISSN 1001-3695



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





# 计算机应用研究

APPLICATION RESEARCH OF COMPUTERS

1997 • 3

## GBH多媒体演讲辅助系统

● 交互式多媒体讲稿排版

● 关键字任意跳转

● 三种自动演示方式



ISSN 1001-3695

**广西计算中心海蓝集**

地址：广西南宁市星湖路32号 邮编：530

电话：(0771) 5851945 5860210 585

传真：5852684



# 贵州省科技计算中心简介

贵州省科技计算中心建于1985年,现有职工45人,科技人员占90%,主要以高、中级专业技术人员为主体,平均年龄33岁。享受政府特殊津贴人员2名,是一支较年青的科技队伍。中心主要从事电子信息技术的应用、开发、研究,下设开发一部,开发二部,信息服务部和办公室。

近年来中心与邮电部贵阳通信机械厂合作,共同开发的邮政分拣枢纽“推式悬挂输送机可编程控制器与上位机通信网络系统,已分别在北京西客站、天津、镇江、大连等省市邮局分栋枢纽工程中夺标,并先后投入运行。”包裹分拣机可编程控制器与上位机系统”正在镇江邮局安装调试。独立开发的“电力电量遥测抄表系统”在深圳供电局的支持与合作下,样机试运行半年,取得较满意效果,现正和美国公司洽谈合作定型。“GPS 邮车跟踪系统及车载采集器”已开始在国内试用。在智能卡的开发应用方面,已完成“公交无人售票 IC 卡系统”(含硬、软件)的研制。中心承担了国家科技信息网贵州子网的建设任务,该网97年逐步建成投入运行。

为进一步开拓电子信息产业的应用领域,中心愿与各科研院所,大专院校,企事业单位建立广泛的合作和联系。

中心地址:贵阳市醒狮路11号 邮编:550002 中心主任:朱华 联系电话:(0851)5823599

## 广西计算中心简介

广西计算中心成立于1978年5月,是广西区内成立最早的从事计算机科学研究与应用的专门机构之一,现有职工总数140多人,有专业技术职称的占80%,高中级技术人员占50%以上。中心现有多套超级小型机、有十多套中高档工作站、中高档微机近100套及一批测试仪表和仪器。现从事新技术研究、软件开发、硬件设计、过程控制、辅助设计、电脑销售、技术培训、激光排版、网络工程、通讯工程、电子工程和机房装修等与计算机技术密切相关的业务。到目前为止,已完成科研项目168项,其中国家“863”项目1项。成果获奖121项次,其中获广西科技进步一、二等奖5项。在研的项目中有国家“863”、国家自然科学基金、国家重点攻关项目各1项,并已在国内外学术刊物和会议上发表学术论文约200篇。

广西计算中心所属的海蓝集团是南宁高新技术开发区的骨干高科技企业,下设海蓝电脑公司、CAD公司、网络工程公司、现代办公设备公司、软新技术公司、自动化通讯公司、电子工程公司及多媒体暨工作站业务部、电脑维修中心、技术培训中心等实体。海蓝集团是广西计算中心对外技术服务的窗口,开展设备销售、技术培训、项目开发、设备维修一条龙服务。已向经营多元化、组织集团化方向发展,业务包含电脑范畴各领域。代理经销优质名牌微机、工作站和兼容机及外部设备、软件产品及消耗材料等。竭诚为社会各界服务,欢迎惠顾。

隶属广西计算中心的广西软件新技术实验室,是广西第一个正式挂牌对外开放的实验室。实验室先后承担了国家“863”高科技、国家自然科学基金、广西重点科研、国际合作等类项目15项,其中“基于图形的超媒体系统”获广西科技进步一等奖、广西计算机开发应用成果一等奖。该项目已被列入国家科技成果重点推广计划。实验室的专职研究人员有留美、留日和留学澳大利亚学者,有来自中国科大、清华、复旦等院校的硕士、学士。实验室配有高档工作站和微机多套。近期研究方向为多媒体、超媒体技术、语言文字处理、软件工程工具和计算机辅助设计。实验室从1995年起分别与中国科大研究生院、中科院计算所合作,联合招收计算机软件专业硕士研究生。实验室正在加速建设,争取早日列入国家级对外开放实验室。

广西计算中心1993年获自治区“振兴科技先进单位”称号,广西计算中心党支部从1990年起连续5年荣获区直机关基层先进党支部,有自治区优秀专家2人,有享受政府特殊津贴专家3人,有突出贡献专家2人,自治区十大女杰1人,自治区劳动模范1人,自治区三八红旗手4人。

广西计算中心地址:南宁市星湖路32号 邮编:530022 联系电话:(0771)5851945 5860210 5862543

# 新技术、新产品荟萃

1. 博大(BD)系列电脑电视字幕机  
——用于电视文字字幕特辑、动画/图形创作、数字图像处理特技
2. SMD-6420 三维制鞋辅助测量辅助设计系统  
——用于各种鞋类设计、样片制造及生产管理
3. MD 系列机器设备状态监测与故障诊断系统  
——用于工业生产工艺流程中的各关键设备的运行监测、故障预报、故障分析诊断和指导维修
4. BMP-C 系列彩色脑地形图处理系统  
——用于诊断脑部疾病
5. CBM 系列心电图体表电位标测系统  
——无创性心电图检查, 早期诊断心脏病。

6. KRL 矿热炉微机控制系统  
——用于黄磷、铁合金、电石行业的矿热炉控制
7. WCY 型电网调度微机远动系统  
——用于地方电网的监测、调度指挥
8. DZJ-5G 电能综合计量仪  
——用于电能分时计量、电压缺相检测、电压合格率统计计算
9. 电子防盗、电视监控报警保安系统  
——用于银行、宾馆、机场、车站、商厦、交通枢纽、博物馆、库房等各行业
10. DNX 型电脑成像机  
——用于旅游点电脑画像

对上述系列产品及项目感兴趣的单位及个人, 欢迎来人来函洽谈, 欢迎实地考察, 具体洽谈事宜请同本刊编辑部张钢、哲明同志联系。(联系地址: 成都市人民南路四段 11 号附 1 号, 邮编 610041)

## 《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  
福建林学院计算机科学系

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  
福建省计算中心

山东省计算中心  
河南省计算中心  
青海省测试计算中心  
武汉汽车工业大学电子系  
新潮计算机产业集团公司

## 《计算机应用研究》杂志社董事会

董事长: 周赛渝  
董事: 乔中南 张国栋 徐雅君 王升亮  
徐世坤 叶大卫 管小宝 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: 罗海鹏 龚宇清 张湘金 倪凯民  
王义章 范德元 李文华 毕常青  
崔振远 赵桂林 宁正元 刘铁军  
贾洪钧 刘启茂 黄安南

1997 年第 3 期(总第 77 期)

出版日期: 1997 年 5 月 30 日

责任编辑: 张 钢

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: (028) 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





## 目次

### 综述评论

CAD技术与工业设计的发展.....	吴英飞(1)
全球化电子邮件服务.....	徐友连(4)
面向对象数据库系统的研究和开发.....	孙界平 龚荣武 唐宁九(6)

### 研究探讨

基于工程素描的计算机真实感显示.....	冯雁(9)
基于VR接口的舰艇模拟训练视景仿真原理与方法.....	杨宝民 曾贤勇 朱一宁(11)
静止图像传输系统设计与实现.....	王海云 白德岚 胡建彰(14)
CIMS工程数据库的分布式在线事务处理控制系统.....	俞勇 黄上腾(16)
Oracle 7关系型分布式数据库的两阶段提交机理.....	刘达顷 石冰心(18)
关于集合理论的KDD方法.....	马建军 陈文伟(20)
一种基于推理的自动生成施工网络图模型.....	曹桂琴 郭晖(24)
铁路自动化系统工程中数据通信的研究.....	陈大为 韩艺(26)
基于FPGA的主机与多单片机快速通讯系统的设计.....	党岗 胡华平(28)
Ramsey数 $R(3,3,3,3;2)$ 的下界.....	张正铀 苏文龙 罗海鹏 吴康(30)
$N \times 36$ 表示为两对孪生素数的和.....	尤之述 罗海鹏 黎贞崇 唐贵松(32)
Object Windows与Visual Basic控件资源研究.....	薛少华 陈德新 鲁放(33)
Windows 95的长文件名支持原理剖析.....	刘绪平 赵学胜(36)
多媒体网络技术探讨.....	倪凯民 唐弋清(38)
编译程序开发技术研究.....	钟乐海 刘宴兵(41)

### 应用开发

WINDOWS环境下图形的局部放大.....	张海 潘光斌(43)
WINDOWS顶级窗口技术的实现.....	刘晓梅(46)
WINDOWS环境下多媒体应用程序设计.....	姚青 贾智平(48)
计算机多媒体演示系统的开发.....	崔大勇 聂发如 薛智 毛静涛(50)
在ORACLE*Graphics中使用图象和声音的技术.....	穆斌(52)
RIFF音频文件格式及其波形输出与应用.....	梁雯 刘松林(54)
从C++到Java.....	王舒(55)
Power Builder与ORACLE的接口.....	肖江文 盛翊智(58)
多文档界面在雷达仿真软件中的应用.....	张海 潘光斌(60)
多用户文档管理系统的设计.....	孙正严 张积友 范冰冰(63)
基于客户机/服务器环境下的MIS系统开发.....	倪凯民 唐弋清(66)
电子企业CIMS工程中CAD/CAM的集成.....	金涌 黄沛然 文远保(69)
优化FoxPro菜单管理的一种策略.....	陆遥(71)
怎样正确使用Browse命令防止窗口显示混乱.....	王晓武(73)
如何实现Foxpro和Sybase之间的数据传输.....	成卫青(75)
用FOXBASE播放WINDOWS图象.....	何明昌(77)
向程序管理器增加程序组和程序项的简易方法.....	时丽珉 谢壮宁(78)
一个实用的下拉式菜单设计方法.....	雷霆(79)
双端口RAM实现任意波形发生器.....	袁波 程建溪(81)
实用单片机集散控制系统通信机制的研究.....	刘志勤 陈波(82)
NDPC-386在PC环境下DSP TMS 320C30系统中应用.....	丁国盛(84)
提高8098片内A/D转换器的方法.....	徐创文(86)

### 维护维修

多媒体作品中声音卡的检测技术.....	夏天 王化文 麻中华(88)
微机硬盘不能启动故障原因与维修方法.....	王立伟(89)
SUN工作站IDPROM内容失效后的启动方法.....	杨绍洲(90)

### 动态简讯

简讯.....	(8、68、76、87)
读者意见调查表.....	(65)
启事、广告.....	(10、19、23、31、32、88)

# Soul Journal of Chinese Science & Technology Articles

## APPLICATION RESEARCH OF COMPUTERS

### VOL. 14 NO. 3(Total 77)

### CONTENTS

#### SURVEY

The Development of the Technology of CAD and Industrial Design .....	Wu Ying fei(1)
Serving with The E-mail on The Whole World .....	Xu Youlian(4)
On the Research and Development of Object-Oriented Database Management Systems .....	Sun Jie ping et al. (6)

#### RESEARCH

Stimulating Real Objects Based On The Engineering Sketch .....	Feng Yan(9)
The Principle and Method of Ship View Simulation for Mimic Training Based on VR Interface .....	Yang Baomin et. al. (11)
Design of Still Image Transmising System and It's Implementation .....	Wang Hai yun et al. (14)
Distributed OLTP Transaction Processing Control System for CIMS Engineering Database .....	Yu Yong et al. (16)
Two--Phase Commit Mechanism of Oracle 7 Server .....	Liu Daqing et al. (18)
A Method for KDD with the Base of the Sets Theory .....	MaJianjun et al. (20)
Automated Layout Network Graph Model Based on Reasoning .....	Cao Gui qin et al. (24)
Research on Data Communication in Railway Automatic System Area .....	Chen Da wei et al. (26)
A System of Fast Communication Between Host and Multi-Single-Chip Microcomputer Based on FPGA .....	Dang Gang et al. (28)
Lower Bound for Number Ramsey $R(3,3,3,3,2)$ .....	Zhang Zheng you et al. (30)
The Sum of Two Twin Prime Number by $NX_{36}$ .....	You Zhi shu et al. (32)
Researching With Object Windows between Visual Basic Resources of Control Unit .....	Xue Shaohua et al. (33)
The Principle of Long File Name Supporting of Windows 95 .....	Liu Xuping et al. (36)
The Discussion about the Multimedia Technic in Net .....	Ni Kai min et al. (38)
Researching for the Developing Technology of Compiler Language .....	Zhong Le hai et al. (41)

#### APPLICATION & DEVELOPMENT

The Part Enlargements of Graph on Windows .....	Zhang Hai et al. (43)
The Implementation of Topmost Windows .....	Liu Xiao mei(46)
Multimedia Programming For Windows .....	Yao Qing et al. (48)
Developping for the Demonstrate Programs on Multimedia Computer .....	Cui Da yong et al. (50)
Techniques for Using Image and Sound in ORACLE * Graphics .....	Mu Bin(52)
Applications of RIFF Format and Waveform Restoration of Multimedia Audio File .....	Liang Wen et al. (54)
From C++ to Java .....	Wang Shu(55)
The Interface between Power Builder and ORACLE .....	Xiao Jiangwen et al. (58)
Researching for Files Interface on Emulate Software of Rader .....	Zhang Hai et al. (60)
Designing with Files Interface of Management System .....	Sun Zheng yan et al. (63)
Development of MIS based on Client/Server Enviroment .....	Ni Kai min et al. (66)
Integrated CAD/CAM of CIMS Engineering on Electronic Enterprises .....	Jin Yong et al. (69)
A Method for Optimizing the Menu Management of FoxPro .....	Lu Yao(71)
How Use Browse Command Guard against Confusion of Displaying Windows .....	Wang Xiao wu(73)
The Data Communication between Sybase and Foxpro .....	Cheng Weiqing(75)
Broadcast Windows's Picture with Foxbase .....	He Ming chang(77)
The Simple Method of Increase Programs to Librarian .....	Shi Li min et al. (78)
The Design Method for Push-down Menu .....	Lei Ting(79)
Two Ports RAM Used In Random Waves Generator .....	Yuan Bo et al. (81)
Communication Structure Reach of A Distribution System .....	Liu Zhiqin et al. (82)
Applicatting forNDPC-386 of DSP TMS320C30 System on PC .....	Ding Guosheng(84)
The Method about Raising A/D Converter of 8098 .....	Xu Chuangwen(86)

#### MAINTENANCE

The Detection Technologe for Sound Potent Card on Multimedia .....	Xia Tian et al. (88)
The Fault Cause for Don't actuat Hard-disk on Microcomputer and it's Maintenancing Method .....	Wang Li wei(89)
A Method to Boot SUN Workstation after The IDPROM Contents are Invalid .....	Yang Shaozhou(90)

#### INFORMATION

Information in Bries .....	(8,68,76,87)
The Fact-Findings List of Readers' Views .....	(65)
Notice & Advertising .....	(10,19,23,31,32,88)



# CAD技术与工业设计的发展

吴英飞

(浙江大学计算机系工业设计研究所 杭州 310027)

**摘要** 主要介绍目前的工业设计软件系统对工业设计发展所起的促进作用,并论述了它们的不足之处,提出了发展中国工业设计软件的发展方法和方向。

**关键词** 计算机辅助工业设计 工业设计 平面设计

## 1 CAD技术的发展历史

计算机辅助设计(Computer Aided Design)始于五十年代末期,至今已有近四十年的历史。从七十年代末期以来,CAD产品逐渐成熟,纷纷作为商品进入市场。

CAD技术通过利用计算机进行分析、计算、调整、优化和绘图,把设计人员的思维、综合分析能力与计算机的快速、准确和易于修改的特性综合起来,从而加速了设计的速度,提高了设计质量。

## 2 CAD技术应用于工业设计

工业设计,从广义的角度来分类的话,可以分为二维的平面设计、三维的产品设计和四维的环境设计。

工业设计是一个复杂的以设计师为主导而进行的循环迭代过程。对于产品设计,设计人员根据产品的规格和要求,先进行概念设计,并制定初步的设计方案,然后对方案分析和计算,实现具体的设计,最后对结果进行评估。当要求达到时,就完成设计。当要求未达到时,就需要修改设计方案,重新进行上一次的过程,如此反复,直到满足要求为止,对于平面设计,整个过程也类似。若用传统的设计方法,修改起来麻烦,甚至可能出现重新返工的局面。而用计算机进行辅助设计,就能够充分利用它易于修改的特性来提高效率。

以下我们分类讨论CAD技术在平面设计和产品设计中的应用情况:

### 1) 二维的平面设计

对于二维的平面设计,一般设计人员的最终目的是要将自己的构思精确地表现在印刷品上。整个设计需要进行文字处理、图片处理、整体编排等阶段。若按照传统的手段,每步工作输出时要与不同的单位接触,因此制作周期很长,而且在时间的压力下,修改也不太可能。相比之下,这些过程在计算机中可以一体完成,而且整个作品的效果实时地展现于设计人员的面前,这样极大地改善了设计环境。我们可以感觉到,过去那种使用鸭嘴笔、针笔、喷笔进行手工设计的时代已经过

去,要求简洁、精确及效率的未来趋势要求每一个设计人员进入计算机辅助设计时代。

对于平面设计CAD系统,一般可分为图像类和图形类。作为一名优秀的设计师,必须对这两类平面CAD系统有全面的把握。

图像类的CAD系统用于处理没有结构化的图形的像素来产生新的图形。此类系统有铅笔、刷子、喷枪及橡皮擦等工具,还有许多用于特殊效果表现的工具,其中包括粉彩笔、碳笔、蜡笔等。在微机上典型的软件为:PHOTOSHOP和PHOTO STYLER等,在苹果系列机上有:PIXEL PAINT和MAC PAINT等。

图形类的CAD系统则是通过不同的单元组合而成,而每一个单元都可进行放大缩小、加长减短等变化,并且精度很高。该类系统通常有画线、画几何图形及指定色彩等多种工具。在微机上的常用软件有:COREL DRAW等。在苹果系列机上有:CANVAS及SUPER PAINT等。

由于目前国内应用较多的是国外的平面设计软件,从文字处理来看,英文字母的输入及字体的种类已经相当成熟了,而且编辑时用于变形的方式较为自由,如可将一段文字按照曲线作弯曲变形,也可将它们填充于一个封闭的图形里等等。中文输入相对英文输入就要复杂,需要某些中文软件的支持,通过一些间接的转换过程,这些过程总的可以分为以下两类:

(1)通过某种中文软件,将所需的文字转换为平面CAD系统可以接受的图像格式,而不是类似于英文字体是作为文字来处理。

(2)通过单字节来表示汉字,这种方法要求某一中文平台与平面CAD系统同时运行。

这两种处理手段以第一种为好,因为它的依赖性少,显得较为独立。

设计人员常常要在设计中用到图片,若图片质量不好,就要进行处理。按照传统的方法,要通过暗房技术或用喷笔等对图片进行修改,这样很费时,而且成本颇高。当用计算机进行辅助设计时,可将图片通过扫描仪扫描后,得到计算机能够识别的数字图像信息,然后

再进行各种处理,如可以修补瑕疵,改变图像的对比度或锐利度,改变图像的色调等等。这类系统还提供了多种以往不可能达到或需要花费很大代价才能得到的处理效果,如可以对图像按照设想的模糊程度进行模糊处理,而以前必须借助于印刷技术,而且最后的印刷品是否达到意图还是个未知数。此外系统还可对图像扭曲,动态模糊,块面化等。

此外,设计人员经常要将多幅图片合成。图片的合成是将图片 A 的部分内容或全部内容以某种方式贴到图片 B 中。合成的方式可分为下述两类:

(1)图 A 中要合成的内容完全覆盖了图 B 的内容;

(2)图 A 中要合成的内容以透明的方式贴在图 B 上,两部分图像进行叠加;合成时,边界可以模糊,使两者融于一体。对于这些情况,在计算机中目前都能实现,而且特别是在第二种情况时,能够很好地控制透明的程度,因此图 A 在最后的合成图像中所表现的程度能够随意控制。经过这些处理后的图像会使整个作品显得更有生气。

在一个设计中,常常要绘制插图。简单的卡通式插图可以在图形化 CAD 系统中构出轮廓,然后填上适当的颜色。复杂的需要在图像处理系统下用喷笔,蜡笔,碳笔等工具绘制。计算机很好地模拟了这些工具,使得设计人员能够迅速地掌握并自由地设计。如为了模拟手工操作时喷笔离开画面各种距离的情况,计算机中喷笔的点的大小可以调节;为了模拟手工操作时对出气量大小的控制,计算机中喷笔的疏密程度可调节;还可以方便地改变喷笔的颜色等。另外,为了控制喷涂的范围,按照手工的方法,贴膜是一种不可缺少的材料,但是使用起来非常麻烦。在计算机中,用户可以任意勾勒出发膜的形状或让计算机自动构出边界作为贴膜。

完成上述准备工作后,我们就可以在图形处理系统中,将文字和图形编排在一起了。在完成整个设计后,可以直接制成四色分色胶片为印刷所用。整个过程变得更有设计味了。

## 2) 三维的产品设计

现代社会工业产品日趋多样化和复杂化,产品的生命周期越来越短,而且制造技术日趋密集,因此各部门之间必须相互合作。如果仍用以往的手工方法进行设计,从绘图到制作模型给其他设计成员来探讨,会花费大量的时间,可能已使产品在市场失去了竞争力。而计算机辅助设计的运用能够缩短设计的周期。例如设计一个表面带有细微圆柱体突起的圆柱形把手,模型制作时若用手工制作,非常复杂。但是用计算机来实现,则可先建立一个圆柱体作为把手的基体。然后沿着圆柱体的径向在表面的下部用圆形阵列拷贝方法建立一圈细微的圆柱体。然后再将这圈圆柱体,沿轴向阵列

拷贝上去,并对之进行参数调整,便可迅速完成。

随着计算机硬件技术和软件技术的飞速发展,计算机辅助设计技术已从二维发展到三维,产品的辅助设计已成为可能。针对工业设计这一领域,为了让设计人员在进行辅助设计时有一个直观的印象,产品必须能够在计算机屏幕上真实地模拟出来。这就要求计算机能够建立起精确的立体模型,且能表达出产品表面的色彩,纹理,材质感,亮度,透明度,反光性等等。

这里,我们先讨论以下几个概念:

(A)计算机中对色彩的表示:

一般色彩学上是用色调、彩度与明度三种属性来表示色彩的,这种减色性混合方法对光的表示就不太合适。光是以红、绿、蓝为三基色的增色性混合,由于光的亮度特性无法用明度属性来表达。因此将明度改为亮度来表达,目前在计算机上用色调、彩度与亮度三种属性来表示颜色。

(B)物体表面的光与形:

在实际环境中的物体,物体表面的颜色不仅仅是它本身的颜色,还要随着环境色做出一定的变化。一般,物体没有直接受到光源照射的表面显得灰暗,而面向光源的部分会显得较为明亮。计算机中,反光区(specular),背光区(ambient),受光区(diffuse)的色彩可以设定,使物体在模拟渲染中显得更有立体感和真实感。

(C)计算机对立体表面的明暗模拟:

计算机对立体表面的明暗模拟主要是通过对光与面之间的反射模拟来实现的。光源可分为直接光源与间接光源。间接光源是指其它物体表面的反射光。计算机所要模拟的重点,就是当物体表面在直接光源的照射下,受光区域会均匀地向周围反射光线。光线的计算方法有以下四种:Flat 方法、Gouraud 方法、Phong 方法、Blin 方法。

Flat 方法:

这种方法是较早发展出来的,它将物体看作许多小面块的组合,计算出各面块的法线与光源之间的角度关系,然后计算出各个光亮值。这种方法的优点是速度快,但缺点是表面看起来象鱼鳞,没有强烈的反光效果。

Gouraud 方法:

为了弥补上述方法的缺点,在计算上述各小面块的光亮值后,再以中间插入法在各面块光亮值之间取中间值,求出平滑的光亮变化,这样使得表面看起来圆滑。它的缺点是无法表现出强烈的反光,因此,无法表现出物体表面的闪亮的感觉。

Phong 方法:

在此法中,光源在表面的入射角度与反射角度都被精密地计算,其中高光区域的颜色和光源同色而不是物体表面的颜色,但这也是这种方法的缺陷所在。它



所模拟出的物体表面看起来好象一块塑胶表面。至于金属光泽的表现,需要调整反光区(specular)和受光区(diffuse)的颜色才能模仿出来。另外,此种方法只能模仿平滑光亮的表面,且对低角度的入射光源模拟时反光太集中,不太真实。

Blin 方法:

该方法弥补了 Phong 方法的缺点。它的最大优点是能随着物体表面不同的光滑度产生不同的反光能量,而且从镜面反射到粗糙面的散射都可以模拟。

(D)表面肌理质感模拟:

表面肌理是通过贴图的方式得到的,其中表面贴图和凹凸贴图用得较多。

表面贴图(Texture Mapping)

表面贴图就好像我们用贴皮或喷漆对产品的模型的表面进行处理一样。计算机对构造好的物体的表面进行贴覆时,不需要改变原先建构好的三维实体。贴图的材质可以在一些平面绘图软件中绘出,或用扫描仪扫描图片得到。在贴图的时候,可以对颜色重新进行调整。例如,我们需要将一物体表面覆上黑色的木纹,而手头只有红色木纹的贴图材料,只要对颜色进行适当的调整就能实现。

贴图的时候,需要对不同形状的表面都能进行贴覆工作。在计算机中有三种贴图方式:

(1)平面方式(planar):将材质赋在平面上。

(2)柱状方式(spheral):将材质包裹在一圆柱上。

(3)球状方式(globe):将材质包裹在一个球状的曲面上。

这样,就能够很好地适应各种形体对贴覆材质的要求。

凹凸贴图(Bump Mapping)

物体表面的质感多种多样。许多实际物品的表面都不尽是光滑的,如桔子的表面、水泥地面。凹凸贴图是通过扰乱表面的各部位的法向量,造成表面反射光量的变化来欺骗视觉,让人感觉到具有凹凸质感,它不是以立体高低差变化的方式来表现凹凸,而纯粹是以改变明暗的方式来表现出起伏的感觉。这种技巧,在我们平常的作画中很常见,如在一张平坦的画纸上用碳笔表现有凹凸感的表面,往往在凸出的部位画得较明亮,以产生高反射的感觉,而在凹陷部位加暗使之有低反射感觉。

凹凸贴图通常要配合表面贴图来做的。例如,在一面墙上贴砖块的纹理,凹陷的部位应出现在砖块之间的细缝上。凹凸贴图在进行凹凸感模拟时,首先得判别表面贴图的图案的灰阶差异(对于彩色图案必须转换成灰阶图案),然后依照贴到表面时各部位的黑白差别赋予各部位不同的法线值,使白的部位的反射量增强而黑的部位的反射量减少,从而使表面产生凹凸的质感。

将实际的产品设计问题应用于计算机辅助设计,必须转化为计算机能够接受的形式,但实际的工业产品的几何形状往往非常复杂,很难用一些标准的几何模型(如立方体,圆柱体,球体等)来统一表示。但是无论它们多复杂,事实上都由一些几何体元组成,或由这些几何体元的布尔运算的结果得到。

为了适应工业设计的具体要求,CAD造型系统应具有下述功能:

\* 几何造型方法以堆砌为主,尽量少用布尔运算。这是由于堆砌操作不修改数据结构,所以简单易行且可靠迅速,在实践中证明是行之有效的办法。

\* 对几何模型有完备的各种变换功能:如平移,旋转,二维缩放,三维缩放,阵列布置,镜像变换,删除,对齐,以及布尔运算等系列操作。

\* 完善的二维功能与三维功能的结合。比如在二维的系统下绘出截面的形状,若沿一定的垂直路径进行延伸,则形成平面形零件;若沿一定的轴旋转,则形成旋转体零件。这种方法简单可靠,而且能够构造出人工难以描述的形状,是一种很有效的方法。

\* 给形体表面赋上一定颜色的材质。工业设计中产品的颜色和质感有很高的要求。设计时,我们会用到各种各样的材料,如塑料、木材、大理石、金属等多种材质,而这些材质在计算机中都能正确表现,这样给设计师提供了很大的帮助。

目前的CAD系统较好地实现了上述功能,将设计师从繁重的效果图绘制及模型制作阶段解放出来,而这两个阶段是整个设计过程中最艰苦的阶段。特别是它们的修改也非常困难。CAD则不同,它使设计师更关注于产品的整体效果,并且真实的效果模拟将不断激发设计师的灵感。

目前,在微机中较常用的辅助设计软件有AUTOCAD和3D STUDIO。计算机辅助设计在工作站上的发展也很快。就产品造型真实模拟的需要来说, Alias, Softimage, Wavefront 都是优秀的软件。

### 3 中国开发工业设计CAD系统的发展方向

我们必须注意到,设计电脑化并不能代替设计人员进行设计,而只是作为一种工具而已,但它的时代性与未来性的趋势是不容忽视的。

产品设计的过程,是对一个工业设计人员提出全面挑战的阶段。设计者必须妥善协调各个部门之间的工作,所应考虑的情况包括从产品最初的开发思想,产品的功能设计,产品的外型设计,产品样品的制成,产品设计的修改,产品的定型,产品的营销策略,产品的售后服务等一系列的情况。计算机辅助工业设计系统应该能够系统地帮助设计者完成这些工作。这样,当工程设计人员在用该系统设计产品结构时,产品的结构信息和材料信息应存放于一个公共的库里,以使得工业设计人员在进行外型构思的同时,能够充分考虑到

内部结构,这种系统化地设计,必定是从产品的内部功能到形式的完美结合,而不是流于表面地粗浅地拼凑。

中国在开发自己的计算机辅助工业设计系统时,必须加强对工业设计的理解,了解国内工业设计人员对系统的要求,不要一味地模仿国外的软件。系统开发人员要明确这些软件最终是给国内的设计人员使用的,而实现这个目的的最好办法是与工业设计人员相互合作,利用他们丰富的专业知识和经验,一起讨论整个 CAD 系统的总体发展远景。

就目前的一些西文 CAD 系统来说,有许多系统功能不够完善,如设计人员在进行设计过程中,要对几个物体进行操作 A,操作 B,……,操作 N,按照目前的情况,只能对这几个物体的第一个进行整个操作过程,然后要对第二个物体进行同样的操作过程等等,而不能够将这些操作过程记忆下来,重复作用于每个需要类似操作的物体。这种宏替换工作,在计算机辅助设计过程中是非常必须的。目前的系统还存在这样一个明显的问题,即在平面设计 CAD 系统中,设计者通过电脑进行设计,并将作品通过电脑制作成四色胶片后印刷,会发现印刷品的颜色与设计时在屏幕上模拟时的颜色有差别。这个问题是由于计算机中色彩的表示与一般色彩学的表示不同而产生的。这一点前面我们已经讨论过。在诸多设计系统日益强调“所见即所得”的今天,而颜色上的差别却给忽视掉了。这使得设计人员在设计的过程中,无法精确地把握最后印刷出来的作品的效果。这些问题是系统开发设计人员所不能发现的。这就要求系统开发人员从系统开发的最初阶段,直到每一个具体的研制过程,都和工业设计人员密切地

结合起来,只有这样才能设计出优秀的计算机辅助设计系统来。

CAD 系统的智能化是 CAD 系统的重要发展趋势。由于目前的一些 CAD 系统,只能在方案设计阶段及其以后的几个阶段帮助设计者完成设计任务。而在抽象阶段,如概念设计等阶段则显得无能为力,因为这些阶段的工作需要一些设计者的知识和经验来完成,而这些知识和经验比较模糊,不可能用精确的数学模型来描述。而智能 CAD 将人工智能技术引入 CAD 中,可以综合人们已有的经验和知识,来模仿设计师的思维过程,从而使 CAD 系统具有一定的智能能力,在一定程度上减轻了设计师对一些常规事务的关心。例如:在建筑 CAD 中,系统能够自动在剖面图上指定的各点之间画出楼梯,在平面图各点之间画出停车场。这样,系统可以根据设计师提出的一些约束条件或要求进行优化设计时的自动设计。

我们希望广大的工业设计人员能够更好地应用计算机进行辅助设计,这也是时代对每一个设计人员的要求,同时也期待着国内的 CAD 系统开发人员能够设计出更优秀的系统来。

#### 参考文献

- 1 金庭赞. 计算机图形学. 杭州:浙江大学出版社,1990
- 2 刘慎权. 智能 CAD 的发展与前景. 计算机辅助设计与图形学报,1989(01): 75—79
- 3 许喜华. 工业造型设计. 杭州:浙江大学出版社,1986
- 4 张悟非. 设计电脑化媒介—电脑对环境的真实模拟. 工业设计,1993(22): 201—206

## 全球化电子邮件服务

徐友连

(上海交通大学 上海 200092)

**摘 要** 本文在分析当前电子邮件互联所存在实现全球化电子邮件服务应在 X.400 和 X.500 的基础上,按照 MHS—DS 的规定,建立起中央 Hub 结构的电子邮件系统。

**关键词** 电子邮件 目录 X.400 X.500 MHS—DS

信息爆炸的九十年代,电子邮件已象日常使用的电话机一样开始逐渐普及。随着局域网和广域网的大量建立,特别是 Internet 服务的出现,距离开始逐渐摆脱了和计费的直接关系,更使电子邮件的好处越发明显。近期 Intranet 的兴起,也使电子邮件在企业内部得

到了大量的使用。据世界权威组织 Gartner Group 的统计,电子邮件正以每年 35% 的速度递增。目前,电子邮件产品在市场上种类较多,各具特色,而且技术和功能日新月异。由于现有的电子邮件产品基本上都是各公司独立设计和开发,其报文格式、地址格式、目录系统和管 理方式等都自成一套,相互的兼容性很差,因而导致目前电子邮件的发展面临着两大主要问题:



一用户对电子邮件产品具有较大选择范围的同时,随之而来的问题则是,当用户想要将多个电子邮件产品集成在一起时,面对他们的是电子邮件间的报文(数据)和地址格式的不统一,集成遇到了棘手的问题。

一电子邮件的核心是目录系统,它含有用户的名字、地址等关键信息。电子邮件的集成同样也遇到各电子邮件的目录系统不兼容的问题,导致系统无法建立起一个统一管理下的目录系统。

信息的全球化是发展的必然目标,特别是 Internet 已使全球各原本独立的本地信息系统相连成为可能之时,因而,解决当前电子邮件所存在的问题将是迫在眉睫。针对电子邮件的问题,相应的国际组织已制定了一些标准,也就是 X. 400 和 X. 500,以及 MHS—DS (Message Handling Systems—Directory Services)。其目的就是要实现各电子邮件间的数据格式的转换、地址格式的转换和目录系统的自动同步,集成现有的各种电子邮件,实现全球化电子邮件服务。

### 1 X. 400 实现不同数据和地址格式的电子邮件间的信息互联

解决电子邮件的数据和地址格式的不兼容,可以有四种不同的方式。

#### 1) 在整个系统中强制一种电子邮件产品

这种方式比较适用小型的系统,而当系统较为复杂,且有多种硬件平台存在的情况,将使原系统中已被许多用户所熟悉的电子邮件产品不得不被放弃。

#### 2) 使用点对点信关(Gateway)

这种方式目前使用较多,且有多家计算机厂商能提供这样的信关。但在一个系统中有三个以上电子邮件产品时,大量信关的使用,将导致电子邮件的畸变,以及信关管理上的困难和系统的扩展性差等一系列问题。

#### 3) 采用主干结构(Backbone)

当在系统中建立起一条具有标准化数据和地址格式的公共电子邮件主干道时,就可将各种电子邮件产品仅通过一个信关和主干电子邮件相联,从而实现各种电子邮件间的互联。这种方式的优点是大量减去了信关的数量,对管理和扩展不会产生影响。但其缺点是单纯的仅靠主干上的电子邮件难于实现主干两旁的电子邮件间的目录自动同步。目前,X. 400 和 SMTP 被公认为是最佳的主干电子邮件平台。

#### 4) 采用集中辐状结构(Hub)

这种方式实际上是在综合点对点信关和主干结构的基础上,再在主干上,建立起中央目录系统,使中央 Hub 既具有将各种电子邮件所特有的数据和地址格式与公共电子邮件的数据和地址格式间相互转换的功能,又具有在系统统一管理下实现目录自动同步功能。因而,这种方式可解决各种电子邮件集成中的两个问题,实现全球化的电子邮件服务。

X. 400 目前被公认为是最适合作为 Hub 结构的电子邮件协议,因为它是由国际标准组织制定的国际标准,是所有已知电子邮件协议的超集;X. 500 则通常被选用于建立中央目录系统;而 MHS—DS 则为 X. 400 和 X. 500 的有机结合提供了保证。目前已有多家公司能提供在 X. 400 基础上的 Hub 产品(CDC 的 Mailhub, HP 的 Openmail, DEC 的 Mailbus),它们各具特色,但在系统目录方面,Mailhub 却较为领先。

### 2 X. 500 实现异种电子邮件的目录自动同步

所谓目录自动同步是指系统能自动监测出在某一电子邮件的(本地)目录中的变化,并自动将这些变更的信息从该本地电子邮件目录中取出,同时自动更新系统本身的目录及系统中的其它电子邮件的本地目录。

在 Hub 结构中,通常选择 X. 500 来建立中央目录系统,事实上,X. 500 是目前所能见到的电子邮件产品的目录系统的超集。X. 500 是一种分布式的目录管理模式,它是针对建立全球目录系统而制定的。由于 X. 500 中可建立有关对象属性、路由信息等内容的数据库,并能提供相应的管理功能,所以,X. 500 可容纳并管理各种电子邮件的目录信息,甚至 X. 500 还能管理系统网络环境的数据。通过将完成目录自动同步的过程及功能引入到 X. 500 的 DUA 中,借助于 Hub 中信关的作用,当某个本地电子邮件的目录修改后(如增加一个新用户),系统会立即检测出这一目录的增加变化,并通过信关将一些有关该新用户的目录信息(如用户名,电子邮件地址,数据格式,附件信息等)抽取过来,更新 X. 500 目录,再通过其它信关,实现对系统中其它本地电子邮件的目录更新。因此,多异种电子邮件的目录就可实现自动同步更新,不再需要人工的干预,使系统的管理员能摆脱繁重的目录管理及更新工作。全球化电子邮件服务离不开全球化目录系统,在 X. 500 的发展早期,由于其存取及管理上对操作员有较高的要求,一度成为阳春白雪。近几年在这方面已得到彻底改变,因而其发展迅速,甚至现在已能用 Netscape 这样的 Web 存取工具来访问 X. 500 目录信息。

### 3 MHS—DS 实现全球化电子邮件服务

在 X. 400 结构中,MTA 实现路由、转发和投递服务。电子邮件也就是在 MTA 间传递,直至目的地。在早期的 X. 400 系统中,路由信息是保存在静态文件中的,随着越来越多的 MTA 的建立,静态路由文件不断增大,其更新操作就逐渐成为系统非常大的负担,且逐渐的复杂,相互同步就变得很困难。早在十六年前,当时 Internet(采用 SMTP)也曾面对这样的问题。最初在 Internet 上,所有主机(Host)的名字和地址都由一个中央管理表来集中维护,随着 Internet 的急剧扩展,主机数量大幅度增加,导致中央管理表非常庞大,主机

间的文件传送由于检索中央管理表而变得十分缓慢。针对这种情况,Internet 委员会制定了域名系统(Domain Name System),也就是在 Internet 上建立一个分布式的层次化树结构,来储存和存取有关 Internet 中的主机名和地址等信息。由于引入了 DNS,系统中的每个主机将负责为其下属主机提供有关的地址等信息,且每个主机都保存有所负责的主机名和地址等信息,因而,使得以前集中化的存取操作被分解到各层的主机上,Internet 的性能也得到了大大的提高。同样,对于 X.400 中的问题,IETF 借鉴了包括 Internet 在内的一些经验,特别是借鉴了在欧洲的实验结果,从而制定了 MHS—DS 标准。MHS—DS 定义了 X.400 和 SMTP 怎样使用 X.500 目录来获得路由信息和地址转换信息,同时定义了 X.400 的地址怎样直接映射到 X.500 目录的关键名字上,而 X.500 则可使用该关键名字存取到有关的路由及地址转换信息,使得 X.400 和 X.500 有机的结合在一起。X.500 是一种分布式树结构的目录管理模型,MHS—DS 充分利用了 X.500 的管理机理,使 X.400 原先静态管理的路由等信息,可使用 X.500 的分布式模型,并可直接转由 X.500 进行管理,同时也使 X.400 在获得 X.500 其它功能的基础上,其性能得到了进一步的提高。

MHS—DS 支持面向树结构而定义的电子邮件路由和地址的映射方法,同时也为 X.500 新定义了一些

对象类和属性,以便 X.400 地址中的各基本元能直接层次化的和 X.500 中的目录项相对应,比如,MHS—DS 定义了怎样将 X.400 地址中的国家名字对应到 X.500 中的国家项上;ADMD 怎样和 X.400 中的 ADMD 相对应等。鉴于 Internet 的迅猛发展,MHS—DS 也定义了怎样将 Internet 中的域名和 X.500 中的名字相对应,因而使 MHS—DS 同时也支持 SMTP。在 X.500 中,存在着一种实现电子邮件地址和目录相对应的信息结构,称为路由树。在一个 X.500 目录中,可以有一个公共路由树和一个或多个本地路由树,MHS—DS 完全支持这两种路由树。公共路由树的主要作用是和外界的目录系统相连,一个目录系统的公共路由树通常是全球目录系统中的一个枝叶(比如是 Internet 中 X.500 目录树中的一个枝叶),因而所有目录系统的公共路由树合在一起就形成一个全球目录系统。当在 X.400 和 X.500 的基础上,按照 MHS—DS 的规定,建立起中央 Hub 结构的电子邮件系统,并使各中央 Hub 结构的电子邮件系统加以互联,由此,一个全球化的电子邮件系统就成为可能。

X.400 实现不同数据和地址格式的电子邮件间的信息互联;X.500 实现了异种电子邮件目录的自动同步;而 MHS—DS 则使全球化电子邮件服务在 X.400 和 X.500 的基础上真正成为可能。

## 面向对象数据库系统的研究和开发

孙界平 龚荣武 唐宁九

(四川联合大学计算机系 成都 610065)

**摘 要** 面向对象数据库技术和系统是当前数据库研究和开发的热点之一。本文着眼于国产化产品的研究和开发,综述了国际上的最新研究动态及发展趋势,阐述了在我国进行进一步研究和开发的意义,讨论了值得研究的主要内容和关键技术以及可采用的技术路线。

**关键词** 数据库 面向对象 客户/服务器体系结构

### 1 引言

随着计算机技术应用的推广,传统的数据库技术难以满足新型应用领域的复杂需求。为了克服传统数据库技术的不足,自八十年代中期以来,国际上各大研究机构及工业界有关部门纷纷研究和开发面向对象数据库技术和系统,至今仍方兴未艾。此外,具有客户/服务器体系结构的分布式计算环境已成为各企、事业单位目前采纳或首选的主流环境。因此,客户/服务器体系结构的面向对象数据库管理系统的研究和开发具有重要的意义。

在十多年的时间里,我国广阔的数据库市场被美国各大软件公司瓜分殆尽,极大地限制了我国的国产

化数据库产品的发展。近两年来,新崛起的面向对象数据库软件公司已瞄准了我国的潜在市场,正积极努力地在我国开辟基地。但是,由于产品成熟程度、市场、技术惯性等原因,其努力都收效不大。这种形势为我们提供了一个在面向对象数据库技术和系统方面赶超国际先进水平的契机。一方面,国内市场数据库技术的升级换代乃大势所趋;另一方面,我们在面向对象数据库这一新兴领域的理论和技术方面都基本保持与国际先进水平同步,完全有能力开发实用化的拥有自主知识产权的面向对象数据库管理系统,并且以此为突破口,促进我国系统软件和应用软件的开发,从而促进我国软件行业的进一步的良性发展。除此之外,意义还在于:

(1) 客户/服务器体系结构的面向对象数据库系统有广阔的潜在市场

传统的数据库技术在商务应用等领域取得了很大的成功。但计算机的应用领域在不断扩大,包括CAD/CAM、GIS、CIMS、CASE等等,它们所涉及的数据已从简单的字符、数值型数据发展到图形、图像、声音、文本、空间几何数据等非常规数据类型,还存在继承、引用等复杂的关系,使得关系型数据库难以处理。据统计,现实世界90%的数据为非常规数据,如果把它们看成一个个独立存在而又有联系的对象,使用面向对象的数据库技术来管理已成为顺理成章的事情。

#### (2)我国面向对象数据库市场尚待开拓

改革开放以来,美国许多数据库软件公司蜂拥而至,将我国的数据库市场瓜分殆尽。一方面,这极大地促进了我国信息管理技术的进步,但另一方面,我国的国产化数据库系统的研制和开发却因此而受到了抑制,从而使得我们的数据库技术大多停留在实验室研究阶段,尽管许多技术方面并不落后,但也未能在市场上占有一席之地。

近年来,国外的面向对象数据库厂商,如Object Design Inc., O<sub>2</sub> Technology, UniSQL Inc. 等也试图抢占中国市场,但成效不大。这些厂商在试探中国市场之后,正对其产品作进一步完善,以迎合我国市场的需要。这就给我们提供了用实用化、国产化的面向对象数据库产品占领我国市场的机会。

#### (3)国产化客户/服务器体系结构的面向对象数据库系统的意义重大

客户/服务器体系结构的面向对象数据库管理系统主要是为了满足广泛的新型应用的需求。我们了解国情,能开发更好地满足我国的应用需求的系统,对国内的使用单位来说,国产化的系统比国外的系统更便于使用、维护和改进。因此我们有市场优势,技术上也有一定的特色。此外,开发国产化客户/服务器体系结构的面向对象数据库系统符合我国的软件知识产权保护目标。

### 2 研究和开发现状

#### (1)发展历史

面向对象数据库技术的研究和开发是典型的应用驱动的,初期的发展分为两大方向。一是从扩充面向对象程序设计语言出发,另一是从扩充传统数据库系统的功能出发。最早进入市场的Gemstone, Vbase, Statice, G-Base就是扩充了Smalltalk, C++, LISP等语言而成的系统,目前均已退出市场。这些系统缺乏完善的数据库特征。随着研究和开发的深入,人们普遍认为:

面向对象数据库系统=面向对象系统+数据库系统

目前,这种意义下的面向对象数据库系统的发展呈现三大潮流:

- 面向对象数据库管理系统(OODBMS)

代表性系统有: ObjectStore, Versant, O<sub>2</sub>, Itasca 等

- 对象关系数据库管理系统(O—RDBMS)

代表性系统有: Illustra, Montage, UniSQL

- 扩充的关系数据库管理系统(ERDBMS)

代表性系统有: SYBASE10, ORACLE7, DB2/2 Release 2

这些类型的系统各有优劣,它们面临的不足是,有的对数据库特征支持不够,有的对面向对象特征支持有限。

#### (2)标准化

随着研究和开发的深入,研究者以及开发者逐渐在对象模型、体系结构、接口语言、实现技术等方面达成共识。在此期间,独立地成立了四个与面向对象数据库标准化有关的国际性组织:

- ODMG(对象数据管理组):由面向对象数据库厂商发起成立,颁布了ODMG—93,将颁布ODMG—95。

- SQL家族(包括:ISO JTC1/SC21/WG3, ANSI X3H2):扩充SQL以包含面向对象特征。

- ANSI X3H7:成立OODB任务组,制定对象信息管理(OIM)模型。

- OMG:由IBM, AT&T等十大计算机厂商联合成立,制定OODB的对象查询服务等标准。

这些机构在面向对象数据库标准的许多方面逐渐达成统一。

#### (3)市场情况

在全球范围内,相对于数据库产品的40亿美元的市场而言,面向对象数据库产品的市场只占百分之几,但从市场趋势来看,刚刚起步的面向对象数据库产品发展非常迅速。据IDC估计,其增长率在60%以上。可以预见,数据库的将来一定会发展到面向对象技术。面向对象数据库真正占领市场是从1990年以后才开始的,当前最为畅销的前三位产品分别是: ObjectStore, Versant, O<sub>2</sub>。其中O<sub>2</sub>是欧洲产品,在欧洲市场排在首位。ObjectStore, O<sub>2</sub>以及新近推出的UniSQL都曾于去、今年在我国举行过产品发布会,但我国市场尚未见推广。

#### (4)未来的发展趋势

- 进一步标准化。包括对象数据模型、接口语言、体系结构、实现技术等方面。

- 开放性。与其它系统(包括关系等非面向对象数据库系统)的互连或互操作,目的之一是解决遗产问题。

- 技术突破、性能改善。包括客户/服务器体系结构下的Cache的使用和管理, Replica的一致性管理、对象存储中的存储结构,如索引、簇聚、对象存取方法的设计等。

### 3 对进一步研究和开发工作的考虑



开发适合我国国情、符合国际潮流、技术先进、性能优越的具有客户/服务器体系结构的面向对象数据库管理系统,并使之实用化,进行推广应用,逐步过渡到产品,力争使我国在这一新兴技术领域占有一席之地,以我国为基地开拓市场。这是关于进一步研究和开发工作的现实的目标。

鉴于以上所述,我们的指导思想应该是:

(1)以实际应用为背景

深入调查分析我国各企事业单位的信息系统(特别是 CIMS 工程、多媒体应用系统)对面向对象数据库技术的需求,了解计算机装备情况,确定系统运行平台。

(2)以国际标准为指导

充分消化、吸收有关面向对象数据库的国际标准,了解国际上已有的或正在开发的系统所采用的通用行业标准。在开发我们的系统的过程中,结合国情,尽量遵循国际上的有关标准。

(3)以技术突破为特色

及时了解国际研究动态,尽可能采用国际上成熟的先进技术来实现系统,同时选择与应用密切相关的关键技术(如长事务管理、并发控制、存储结构、Cache 管理等)进行重点突破。

(4)以推广应用为目标

系统开发除注重数据建模能力强、系统性能优越等以外,还注意在方便用户使用、提高用户生产力方面提供强有力的支持,开发面向用户的应用开发工具。

值得进一步研究和开发的主要内容包括:

(1)开发一个客户/服务器体系结构的面向对象数据库管理系统原型,并使之实用化,逐步过渡到产品化。

(2)研究面向对象数据库系统与一般关系数据库系统的 SQL 语言的接口,从而实现与关系数据库系统技术和应用上的兼容。

(3)研究进行面向对象数据库管理系统性能测试的 Benchmark,并测试所开发的系统。

与以上主要内容相关的需要研究的关键技术有:

(1)面向对象数据库管理系统的体系结构

依据有关的国际标准,根据我国国情(包括客户/服务器装备情况,中文信息支持要求)设计系统的体系结构。确定系统的功能结构,设计对象服务器、对象管理器的功能及其在客户、服务器上的功能分配、客户机

上 Cache 的管理等。

(2)数据模型和语言接口

根据国际标准,分析现有系统,并结合目标应用对信息建模能力的需求,考虑系统与现行的数据库系统的互连或互操作的要求,设计系统所支持的数据模型及相应的查询语言,提供用户编程语言接口。

(3)事务管理(包括并发控制和恢复)

事务管理是数据库系统的重要特征之一。传统的事务模型不适合于新型应用领域。客户/服务器体系结构的面向对象数据库管理系统必须提供对长事务、类对象的支持。长事务的研究涉及到对基于锁协议的并发控制和基于日志的恢复机制的改进,以及锁粒度等方面的内容。

(4)存储结构及存取方法

研究对象的簇聚和索引技术、大对象和多媒体信息的压缩存储、操作系统接口、虚拟地址空间的使用及对象指针的调入/调出等内容。

(5)应用开发工具集

基于所选操作系统的图形环境,如 OSF/Motif, X11 Toolkit, Windows 等,开发面向用户的开发工具集,提供浏览、查询等功能。开发特定应用的相应类库,以提高应用编程人员的生产力。

针对以上研究内容和关键技术,可采用的技术路线是,依据国际标准,同国际上的客户/服务器体系结构的面向对象数据库系统主流接轨。注重系统的开放性,从而使系统具有良好的可移植性和易操作性。充分消化、吸收国际上最新技术成果,在较高起点上开展研制工作。对于国际上其它机构已研究过或实现了的成熟技术,我们都采用“拿来主义”,而集中精力研究国际上尚未成熟的、反映自己特色的软件技术,开发相应的软件模块。以市场为导向,使研制成果具有较强的市场竞争力。

### 参考文献

- 1 F. Bancilhon, C. Delobel, and P. Kanellakis, ed., Building an Object-oriented Database System--the Story of O2, Morgan Kaufmann Publishers, San Mateo, CA., 1992.
- 2 W. Kim, Introduction to Object-oriented Databases, The MIT Press, Cambridge, Ma., 1990.
- 3 A. Silberschatz, et. al., Database Systems: Achievements and Opportunities, Communication of Association of Computing Machinery, Vol. 34, No. 10, 1991.
- 4 C. Lamb, et. al., the ObjectStore Database System, Communication of Association of Computing Machinery, Vol. 34, No. 10, 1991.

### 中国计算机市场信息发布会在京举行

〈本刊讯〉由电子工业部计算机与微电子发展研究中心市场信息研究中心分时分地主办的在全国八城市中国计算机市场信息发布会,3月18日在北京新世纪饭店首先召开。

1996—1997年度研究报告分计算机和微电子两部分。计算机市场部分有中国1996年回顾与1997—2000年预测,1996—1997年中国大中小型机、工作站、微机、打印机及耗材、显示终端、网络、软件、信息服务业和进出口贸易年度报告。研究报告有书面和磁盘两种媒介。发布会还在上海、深圳、广州、福州、西安、成都和沈阳分别举行。(恒讯)

# 基于工程素描的计算机真实感显示

冯雁

(浙江大学计算机系 杭州 310027)

**摘要** 本文介绍了一种基于工程素描的真实感显示方法,并把这一方法运用于曲面造型中。在本系统中,建立了基于目标空间和图像空间的曲面线消隐算法,在给定的光照模型下,利用工程素描的方法绘出逼真的真实感图形,算法稳定可靠,具有一定的实用性。

**关键词** 真实感 工程素描 光照模型

## 1 引言

计算机图形与人类为伍近三十年左右,已日趋成为人机交互中最重要的手段,尤其是光栅图形显示器的出现,以其真实的图象、丰富的色彩、低廉的价格。而后来居上为计算机的应用开辟了新的道路。

本文研究的就是如何提高各种形式的真实感。在自然界中光线在许多物体之间经过多次反射,最后照射在物体上,由于物体本身的材料性质和颜色的作用而呈现出不同色彩,亮度和质感,要在屏幕上模拟这一效果,就要建立起能反映实际情况的光照模型。再根据这一模型计算出物体上每一点的光照状况。工程素描的方法,就是用线条或局部打点的方法对一些黑白图形进行真实感处理。用该方法处理的图形在印刷、制版上具有实际应用价值。

## 2 设计思想

工程素描是工业美术中比较实用的基础技术,是用单色描绘各种实体的形象,以获得美而真实的画面。它借助于少量的绘图工具,得到了准确、具体而精细的图象,是工程技术与绘画艺术相结合的新学科。

把工程素描的手法与计算机图形的特点结合起来,在表现图形的真实感方面,一般可以有以下几个步骤,首先要确定物体的明面、暗面、阴面,明面色度最浅,用间隔比较大的描阴线或者完全不画线表示。暗面色度最深。描阴线的间隔应等于或者小于细实体本身的粗细。阴面的色度深浅介于明面和暗面之间,描阴线的间距介于明面和暗面之间,从明面到阴面、再到暗面,这是一个过渡的过程,线的间隔应该是逐渐的、缓慢的,从暗到明,描阴线的间隔是逐渐增大的,打点表示与线条同理,暗面点密集。明面稀疏甚至不打点。确定为明暗面以后,就要选定合适的描阴方法,不同物体的表面质地都有区别。例如,玻璃和铸铁就给人不同的感觉,因此不同的物体表面就要选用不同的描阴方法。表面比较光洁,可选用细线描阴法,表面比较粗糙,可选用打点描阴法,或打点涂黑描阴法。除了进行明暗的刻画外,还必须进行远近的刻画,一般图形总是近大远

小,近露远隐,近深远淡,因此,近边应画得清楚,远边应画得模糊、虚淡、飘渺。

## 3 计算的实现

下面分几点论述:

1) 因为整个曲面的表面是用小面片而产生的,因此在仿真显示时,应尽量避免法向量的突变而造成光强的突变。在确定顶点处的法线时可取交于此顶点的各棱边叉积的平均值,如图1所示:

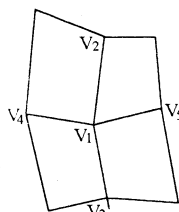


图1

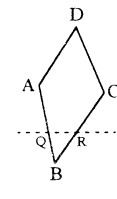


图2

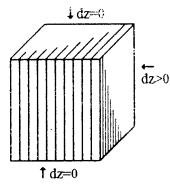


图3

$$N_1 = V_1 V_2 \otimes V_1 V_4 + V_1 V_5 \otimes$$

$$V_1 V_2 + V_1 V_4 \otimes V_1 V_3 + V_1 V_3 \otimes V_1 V_5$$

计算亮度的照明模型可以适当简化,特别是光可以取为平行光,视点亦可取为无穷远处。这样,入射方向 $\vec{L}$ 与视线方向 $\vec{V}$ 均为常矢量,于是,

$$I_P = K_1 \cos(\vec{L}, \vec{N}_P) + K_2 \cos^n(\vec{S}_P, \vec{V})$$

对于平面立体,每个表面的多边形具有一定的法向 $\vec{N}_P$ ,从而 $\vec{S}_P$ 也是固定的,即每一平面的亮度是一样的。

2) 曲面消隐 为了显示曲面,将曲面参数域进行 $u, v$ 分割。目标空间的消隐具有真实感好的特点,但处理时间长,由于曲面的离散线数量很大,消隐速度会很慢。为了改善这一情况,将欲消隐的线段投影到图象空间求交点,结合三维交点分段判断可见性,从而可以较高地提高消隐速度。

下面是平面片近似的曲面进行消隐的步骤:

(1) 取出欲消隐处理的直线段和平面片进行包含性和深度检查,舍弃完全被小平面遮挡的线段。

(2) 在三维空间中计算线段和平面片的交点,将线

段和平面片投影到图像空间,计算图像空间中的交点,根据这些交点将线段分段存储。

(3)对线段的各部分进行可见性判断、合并可见部分,舍弃不可见部分。

(4)对于潜在的相交的曲面片平面化,求解两片小平面之间的交线。

3)用点的稀疏来表示真实感 每一小平面片中点的产生是按扫描线的方式来安排的,因此就必须安排好扫描线之间的间隔分布和每一条线上点和稀疏分布。

为了使光强与点的稀疏联系起来,我们建立了一个子程序用来计算某一光强所对应的点之间的间隔,通过改变这个对照程序的间隔常数,可以达到一个画面的不同效果,从而可以选择一个最佳的常数。

扫描线的间隔产生是按如下方法计算的:首先把小平面片的上面二点的光强取一个平均值,作为上面的光强,把下面的二点光强也取一个平均值,作为下面的光强,然后分别得到它们所对应的间隔  $1d_1, 1d_n$ ,接着利用等差数列求出从  $1d_1$  变化到  $1d_n$  的一个等差常数  $d$ ,这样就得到了扫描线间隔的比较均匀的变化。

设  $dis$  为小平面片上最上面一点和最下面一点  $y$  坐标的距离。

$$dis = \frac{(1d_1 + 1d_n) * n}{2}, n = \frac{2 * dis}{(1d_1 + 1d_n)} \quad (1)$$

$$d = \frac{(1d_n - 1d_1)}{(n-1)} \quad (2)$$

公式(1),(2)用于计算等差常数  $d$ 。

当扫描线的间隔产生以后,同样以等差数列的方法来均匀产生扫描线上的点,扫描线端点的光强是用线性插值的方法产生的。如图(2)所示:

$$I_0 = uI_A + (1-u)I_B, u = \frac{QB}{AB} \quad (3)$$

$$I_R = WI_C + (1-W)I_B, w = \frac{RB}{CB} \quad (4)$$

公式(3),(4)用于计算端点的光强。

4)用线表示图形的真实感 在工程素描中,用线条表示图形的真实感,一般都是指一些比较规则的图形,像柱体、立方体、锥体及复杂的图形如机械零件图等等。

线条的产生是用扫描线方法产生的,当线的方向各不相同,只需把该表面旋转一个角度,使线的方向恰好为水平线,实际在屏幕上画出线条时,再作一个反旋转工作即可。

为了适应不同多面体的情况,线条的产生是用改

进的  $y$  连贯性算法。首先,按边的端点的最大  $y$  值先行桶分类,然后按照边的左右进行插入分类,具有同样最大的边按左右位置插入边记录链中,边记录的信息包括边线的最大  $y$  值和与最大  $y$  值相对应的  $x$  值,扫描线求交点所需的改变  $dx$ ,以及此边与扫描线的交点个数  $dy$ ,从上到下每次用一条扫描线进行填充,对当前的扫描线建立一个有效边线表,把与这条扫描线有交点的所有边线按左右位置链起来。

与  $y$  连贯性稍有不同的是,系统进行填充线条时,每一条线条之间都必须有与光强相对应的间隔,因此每处理好一条扫描线,对当前有效边线表所涉及的边线记录作如下修改:

$$dy = dy - 1d \quad (1d \text{ 为间隔}) \quad \text{当 } dy \geq 0 \text{ 时, } y = y - 1d;$$

$$x = x + 1d * dx$$

当  $dy \leq 0$  时,应把此边线记录删去,把边线最大值等于  $y - dy$  (原  $dy$  值)的新的边线插入到有效边线表中,下一条扫描线显示时必须从  $y = y + 1d - dy$  (原  $dy$  值)开始。

为了表现画面的真实感。近边之线应画得清楚、实在,远边的线应画得模糊、虚淡。实现这点,要计算  $y$  最大扫描线和  $y$  最小扫描线的平均  $z$  坐标之差  $dz$ ,若  $dz > 0$ ,则线条是渐渐远去的。显示时,线条长度按常数  $c$  递减直至没有。若  $dz < 0$ ,线条是从没有到渐渐出现直至显示整个长度。若  $dz = 0$ ,则计算每条扫描线两端的  $z$  坐标,显示靠近  $z$  小的这一部分线段,而不显示整条扫描线。常数  $c$  可以近似处理成  $1d$ 。若  $dz = 0$  且扫描线两端  $z$  坐标相等。则显示该扫描线的整个线段。如图(3)所示。

#### 4 总结

同一造型用不同的方法所产生的真实感显示,其效果是不一样的,即使同样的打点处理,背景色不一样,效果也会相差很多。有的比较透明,有的较粗糙,从屏幕显示及打印结果来看,画面还是较逼真的。但在某些方面还有可改进之处。例如,和手工工程素描相比,有时略显生硬一些,在用线条表示曲面的真实感方面,还可作一些探索。

#### 参考文献

- 1 金廷赞. 计算机图形学,杭州:浙江大学出版社,1988
- 2 明方成. 工程素描,重庆大学出版社,1980
- 3 L. Li. Hidden Line algorithm for curved surfaces. Computer Aided Design, 1988, 20(5)
- 4 苏步青,刘鼎元. 计算几何,上海:上海科技出版社,1982

#### 本刊启事

为了更快地推动、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,欢迎广大作者、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VR接口的舰艇模拟训练视景仿真原理与方法

杨宝民 曾贤勇 朱一宁

(南方图灵VR技术有限公司 成都 610031)

**摘要** 本文根据舰艇综合操纵训练的实际要求,给出了舰艇操纵仿真系统的总体结构,论述了各模块的关键技术例如舰艇综合仿真的层次实现方法。提出了驾驶舱外视景仿真的立体感图像合成方法,针对仿真系统的交互需求,论述了系统设计采用的虚拟现实交互技术,包括适合动态环境的曲面辐射度算法,舰艇航行碰撞检测方法。

**关键词** 真实感图形 虚拟现实 立体图像 动态曲面辐射度方法

## 1 介绍

虚拟现实(VR)技术是真实感图形技术在现代舰艇操纵模拟训练系统应用的关键技术,人工智能技术与图形图像技术相结合,对于海上庞大的场景仿真显得更为迫切,传统的机电式模拟训练系统人机交互接口方式单一,实用性比较差,无法满足现代模拟训练领域的实际要求,影响了模拟训练效果,采用真实感图形技术为使用者提供逼真的虚拟环境,采用虚拟现实人机接口例如三维立体显示装置、力反馈装置、立体声音频设备,实现和谐的人机交互环境,是今后舰艇操纵仿真的发展方向。

美国在设计开发三维超级驾驶座舱用于民航飞行仿真模拟已经取得使用效果,事实上,用于飞行仿真的VR技术有许多共同点,例如座舱外视景仿真,舰艇仿真涉及的视景主要是海面、岛屿、海岸线、天空、码头。飞行仿真涉及天空、飞行物、机场。

舰艇仿真的内容比较广,因此,我们的第一步目标是实现基本功能,完成单舰艇模拟训练的各种科目,第二步应用人工智能技术实现多种战术模拟包括多舰协同训练科目,第三步是实现分布式虚拟现实仿真DIS系统联网,实现战役演习系统规模的训练计划,形成实用的DIS分布式仿真系统。舰艇综合操纵模拟训练系统包括训练管理系统、位置姿态仿真系统、视觉仿真可视化系统、听觉仿真语音合成系统、仿真驾驶控制台、以太网数据通讯系统等组成。解决的关键问题是动态彩色三维对象的交互控制虚拟环境构造。

实现舰艇综合操纵模拟训练模拟系统的下列功能:

- 1) 单舰航行操纵:背离码头、系离码头、叠标航行;
- 2) 单舰特殊操纵:岛礁区航行、夜间航行;
- 3) 多舰互见避碰操纵训练;
- 4) 编队机动训练:机动站位训练、旗号训练;
- 5) 各种战术环境的机动训练。

## 2 系统结构

根据系统实际需求和总体集中设计、模块分散开发的实际条件,在没有现成原型可供参考的实际情况下,独立提出了舰艇模拟训练仿真系统简称(VR SHIP)的软硬件系统结构。

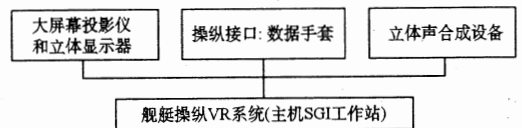


图1 舰艇模拟训练仿真系统系统结构

面向对象的软件方法应用

面向对象的层次结构实现方法

对象构造示例:

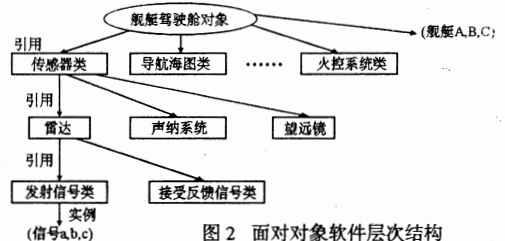


图2 面向对象软件层次结构

## 3 屏幕功能等基本要求

为了方便模拟训练,我们根据实际要求,设计出不同的操作界面,供训练人员使用。具体窗口设计如下:

A. 仪表窗口 舰艇具有各类反映舰艇状态的仪表,显示各类重要参数例如发动机参数等电子仪表。

在仪表窗口可以显示仪表的外形和参数。

B. 雷达窗口 雷达显示窗口包括雷达显示终端,能够模拟雷达扫描线。雷达系统能够显示已经搜索到的目标的主要参数。例如敌我识别,舰艇类型,方向,速度,位置等参数。

C. 敌我舰艇的观察窗口 要求显示敌我舰艇的外形、标志、速度、位置、方向其它操作参数,对于舰艇的三维可以方便完成旋转,移动,不同视点观察等各种操作。



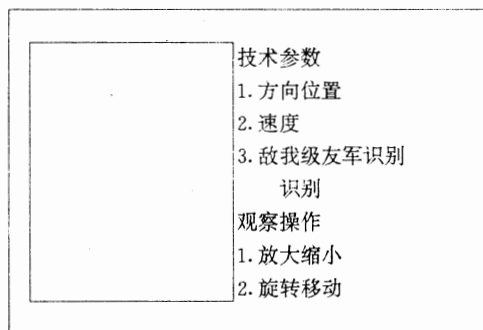


图3 舰艇观察窗口示意图

D. 海图窗口 系统提供的海图允许你在整个区域标记敌我双方舰艇位置,主要海域特征。海图作为导航的重要依据,可以反映舰艇在海域的整体分布特征。

我方舰艇根据总体态势决定航行的方向和速度,例如根据已经发现暗礁,进行简单的单舰避碰训练。

E. 操作窗口 包括雷达、声纳、驾驶舱、导航等图符子窗口,选择不同的子窗口可以进入不同系统。例如选择雷达子窗口,屏幕开始显示雷达系统操作界面。

#### 4 视景仿真的图形图像绘制方法

用于实际训练的舰艇模拟训练环境与游戏环境的主要区别之一在于视景。例如:

Aegis 海战游戏环境已经实现的功能包括:

(1) 战术对抗训练 尤其是具有电子战的内容,反映了现代海战的特点

(2) 电子导航训练 机动站位训练

(3) 战斗训练 包括火控系统

(4) 操纵仪表盘的仿真已经实现

总之,可以看作是基于图符的系统,与海军仿真第一期要求相比,仿真的自然景物不仅包括舰艇和海洋自然环境,而且包括飞机、导弹、潜艇、声纳等景物。

上述的游戏具有很强的专业功能,但是对于实际模拟训练系统要求具有真实感的图形图像模拟视景。其中驾驶舱视景仿真是重点,包括详细的驾驶舱模拟仪表,同时要模拟操作者在驾驶舱内观看周围自然景物。

对于驾驶舱的对象,可以考虑用 ALIAS 软件建立船体曲面造型,对于一般模型采用 AUTOCAD 建立三维模型,然后 Multigen VR 建模软件调用,由于模型非常复杂,必须采取简化措施,具体方法包括:①对 ALIAS 复杂模型或 AUTOCAD 表面进行绘制,生成逼真图象,然后作为纹理在 MULTIGEN 中对简化的几何模型贴图。②采用 ALIAS LINK 软件简化几何模型。

对于复杂场景可以采用辐射度方法,利用该方法与视点无关的特点,可以将真实图象中与视点无关的部分进行预先处理,此外对于动态环境可以采用适合动态环境的曲面辐射度方法。

真实感图形算法今年来主要包括光线跟踪算法和辐射度方法。

光线跟踪技术能够模拟半影、景深、运动模糊等光照效果,但是不能模拟复杂环境中多个物体之间的漫反射效果,例如驾驶座舱内仪表盘与墙面之间的漫发射。

辐射度算法的一般过程如下:

第一步求出所有曲面片的形状因子;若包含镜面,则引进镜面形状因子。

第二步利用辐射度方程,计算所有曲面片的辐射度。

第三步采用光线跟踪技术计算环境中景物表面的光亮度。

传统的辐射度方法将曲面用许多小的平面片,然后用平面辐射度方法逼近,假设每一面片的辐射度为常数,事实上曲面曲率较大的情况下难以成立,为了克服这个缺点,朱一宁 1989 年提出曲面环境的一般辐射度方法。

平面漫射环境的辐射度方程为:

$$B(A_j) = E_j + \rho_j \sum_{i=1}^N B_j F_{ji} \quad (1)$$

适合曲面漫射环境的一般辐射度方程为:

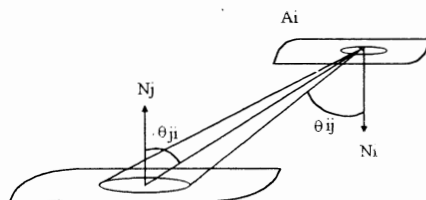


图4

$$B(dA_j) = E_j + \rho_j \sum_{i=1}^N \int_{A_i} B(dA_i) \frac{\cos \theta_{ij} \cos \theta_{ji}}{\pi r_{ij}^2} dA_i \quad (2)$$

$A_j$  与  $A_i$  是辐射面片,  $B(dA_j)$  是微面片  $dA_j$  的辐射度;  $E_j$  为  $dA_j$  自身具有的能量;  $\rho_j$  为的漫反射系数;  $(dA_i)$  是微面片  $dA_i$  的辐射度;  $HID(dA_i, dA_j)$  是  $dA_j$  与  $dA_i$  之间的遮挡函数;  $\theta_{ij}\theta_{ji}$  分别是两面片距离  $r_{ij}$  连线与法矢量之间的夹角;  $F_{ji}$  与  $F_{ij}$  是两个面片之间的形状因子。

对于变化环境的辐射度方程,通过描述外边的时间表示时间决定值。辐射度求解方法中的初始值通过外面的  $t-1$  表示,必须计算新辐射度方法的值,并通过外边的  $t$  表示。外边的  $\Delta t$  用来表示在  $t$  与  $t-1$  之间时间决定值的变化,我们可以将基本辐射度方程写成时间决定的方程:

$$B_i^t = E_i + \rho_i \sum_{j=1}^n F_{ji}^{t-1} B_j^{t-1} \quad (3a)$$

$$\text{和} \quad B_i^{t-1} = E_i + \rho_i \sum_{j=1}^n F_{ji}^{t-2} B_j^{t-2} \quad (3b)$$

$$\text{由于} \quad B_i^t = B_i^{t-1} + \Delta B_i^t \quad \text{和} \quad F_{ji}^t = F_{ji}^{t-1} + \Delta F_{ji}^t$$

我们根据已知的值重写等式 1a, 从先前的时间步骤  $(t-1)$  加上在  $\Delta t$  时刻增加的变化。

$$B_i^{t+1} + B_i^{\Delta} = E_i + \rho_i \Sigma (F_{ij}^{t-1} + F_{ij}^{\Delta}) (B_j^{t+1} + B_j^{\Delta}) \quad (2)$$

扩展了所有项并使用等式 1b 从表达式两边减去  $B_i^{t-1}$ 。

$$B_i^{\Delta} = \rho_i \Sigma F_{ij}^{\Delta} B_j^{t+1} + \rho_i \Sigma F_{ij}^{\Delta} B_j^{\Delta} \quad (4)$$

$R_i$  被定义为重新分布的项:

$$R_i = \rho_i \Sigma F_{ij}^{\Delta} B_j^{t+1} \quad (5)$$

将等式 4 代入等式 3, 给出重新分布的辐射度方程为:  $B_i^{\Delta} = R_i + \rho_i \Sigma F_{ij}^{\Delta} B_j^{\Delta}$  (6)

辐射度重新分布方程(等式 5)同样是基本辐射度方程的形式。只有两个不同点:发射项  $E_i$  已经被重新发射项  $R_i$  取代, 方程未知的变量从  $B_i$  变化到  $B_i^{\Delta}$ 。因此, 我们能够找到  $B_i^{\Delta}$ , 每个面片的辐射度变化分成两步: (1) 计算所有的重新分布项。 (2) 用传统辐射度技术求解  $B_i^{\Delta}$  用来求出  $B_i$ , 为了提高速度, 我们充分考虑了空间连贯性, 因为动态目标在环境中运动, 有些部分光能变化并不明显。如下图所示:

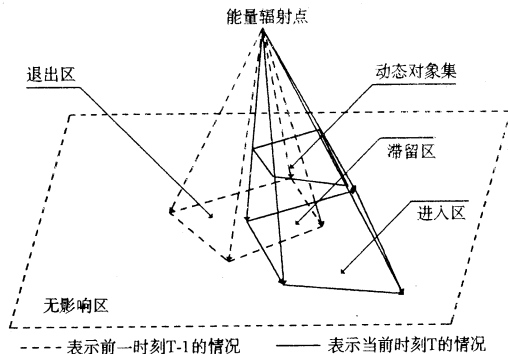


图5 用集合的概念表示动态环境分区空间连贯性

适合曲面非漫射环境的一般辐射度方程为:

$$B_d(dA_j) = E_j + \rho_j \sum_{i=1}^N \left[ \frac{B_d(dA_i) + b}{(dA_i)^3} + \frac{b}{(dA_i)^2} \right] F_{di} dA_j - dA_i$$

第二部分是环境面元直接辐射引起的漫射分量, 用一般逐步求精方法求出; 第三部分是环境面元通过镜面和透射面间接传递引起的能量, 用光线跟踪方法求出。

### (3) 驾驶舱外视景仿真

由于航海仿真系统首先要求场景生成速度满足交互驾驶需要; 其次, 系统要求容纳真实场景, 例如驾驶舱内对象需要采用三维图形技术生成, 舱外天空大海等由图像提供。其次, 系统能够显示与大海视景复杂性无关的高分辨率显示。

采用环形圆柱 360 度全景图像产生立体感, 解决海水、山脉等自然景物图形生成的困难。例如在背离码头, 系离开码头训练中, 在一些关键点拍摄全景图, 这些图像类似环幕电影的画面, 记录在该点 360 度视线方向所见。若视点位于这些关键点, 则任意视线方向的图像可以快速生成。而视点不落在这些关键点时, 可以采用图像插值方法, 或者简单地选取附近的关键点, 效

果比较理想。

具体作法是: 在大海中用五部摄像机提供天空和四周五个方向的图像, 在码头用三部摄影机提供舰艇前面指向码头方向和舰艇船舷两边方向的图像, 将这些图形经过缝合形成全景图。为了降低造价, 可以采取三部高分辨率投影机组合成 180 度环形屏幕作为 SGI 主机显示器, 每个投影机负责 60 度, 图像经过色彩均衡处理后提供给用户。

为了实现典型岛屿地区航行训练, 需要拍摄已经具有航海图的典型的岛屿视景, 例如拍摄同时与现有的测绘数据相结合, 制作典型的海图。或者说, 根据海图, 驾驶员可以观看真实场景。

## 5 核心技术

舰艇视景仿真的核心技术之一是跟踪视点和视线方向的视景生成, 具体分几部分:

- 舰艇活动范围约束
- 船速及方向仿真

以上两点直接决定视点位置, 间接影响视线方向。

视景生成采用图像插值方法, 第一步可以简化, 例如取消 hot spot。

### ——舰艇活动范围约束

舰艇在码头、岛礁区及其它舰艇周围时, 其活动不再自由, 需要进行碰撞检测。具体分两种情况:

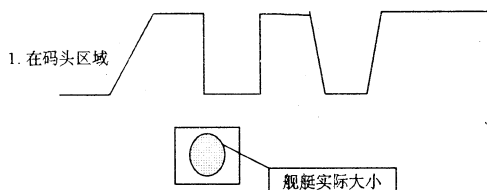


图6 码头及船舶示意图, 本切面图的高度取船坞最高处

方法: 首先用包围盒, 判断这个盒子是否有一部分位于码头陆地部分, 若是则进一步判断, 问题最后简化成直线段和曲线段的碰撞问题。

由于舰艇与岛礁区, 舰艇之间不可以挨得太近, 故而没有必要精确计算, 可用包围盒技术分别为岛礁区和舰船作包围盒, 以包围盒之间的关系取代实体间的关系。

要求: 码头及舰艇模型。

船速及方向公式如下: (实际航行情况)。

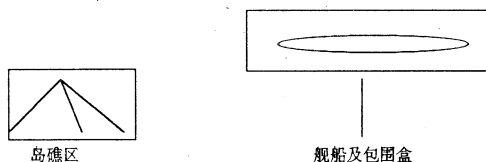


图7 岛礁区与舰艇碰撞艺术示意图

## 6 开发平台

为了专业开发上述软件, 我们根据客户要求从美国引进航海仿真软件平台。

图形仿真:

VR 建模软件 MultigenII:可以满足虚拟现实三维图形模型构造的需要,具有完整的细节控制能力,纹理贴图能力,对象可以定义丰富的物理属性。

航海仿真开发平台:Vega 具体选用模块包括 VegaMarine 具有动态海浪模拟能力,船塔缆绳线,船尾航迹模拟,碇泊浮标,船头波浪,烟雾,船塔等模拟功能。

Vega symbology 符号模块,用于各类船用仪表的仿真。

Audioworks 模块具有实时同步虚拟立体声音模拟能力。

开发平台结构图如下:

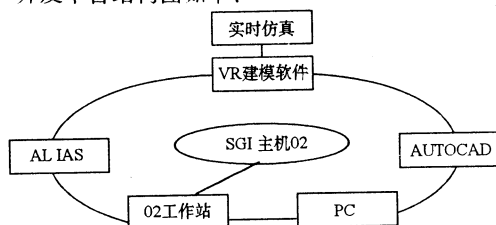


图8 VR开发平台结构图

我们采用先进的 VR 开发平台,采取引进成熟技术与自主开发相结合的方法从事 VR 仿真工作,我们相信新一代仿真系统将会满足舰艇操纵训练的实际需要。

本文写作过程中,曾得到浙江大学硕士朱可伟,南方图灵 VR 公司助理研究员王珏的审阅修改,特致谢意。

### 参考文献

- 1 朱一宁,浙江大学博士论文《真实感图形研究》1989
- 2 David W. George 等 Radiosity Redistribution for Dynamic Environment IEEECG&A1990 July
- 3 Shenchang Eric Chen, Quicktime VR—An Image—based Approach to Virtual Environment Navigation 1995 Siggraphics, Computer Graphics Proceedings, Annual Conference Series
- 4 严成华,舰船视景动态图形显示与仿真系统的研制,1994 计算机工程图学的探索与实践谭建荣主编
- 5 Aegis 海战仿真软件使用手册
- 6 朱一宁,彭群生,梁友栋. peris: A programing Environment for Realistic Image Synthesis, Computer&graphics vol,12, No3 1988

## 静止图像传输系统设计与实现

王海云 白德岚 胡建彰

(南京邮电学院 南京 210003)

**摘要** 本文介绍了一种基于 PSTN 的静止图像传输系统,描述了传输系统的功能和模块结构,并具体介绍了传输系统的实现。

**关键词** 静止图像传输系统 JPEG 压缩 电话 API(TAPI)

### 1 引言

以建立信息高速公路为龙头、Internet 的引入与迅速兴起为标志的全球信息化浪潮把我们带入了信息时代。我国要下世纪中叶赶上中等发达国家水平,就必须顺应国民经济信息化的大趋势。在信息业飞速发展的今天,计算机通信已深入我们的工作和生活;与此同时,人们已不再满足于目前简单的字符终端通信,而对信息质量提出了更高的要求。图像作为一种重要的信息表达、存储形式,正逐渐为人们所接受。所谓“百闻不如一见”,图像所包含的信息量是文字、声音等其它媒体所无法比拟的。与此同时,这种巨大的信息量也给存储、传输带来了很大的问题。

为此,我们提出并设计了一种基于 PSTN 及公用数据网的静止图像传输系统(以下简称“图像传输系统”)。该系统支持用户从视频信源或其他应用程序获

取静止图像,然后采用彩色静止图像压缩标准—JPEG 标准对图像进行压缩处理,再利用公用电话网或数据网把高清晰度图像传输出去。由于采用了图像压缩技术,大大提高了图像传输的速度。该系统可广泛应用于新闻单位图片发送、远程医疗、图像数据库检索以及其它需要对高清晰度图像进行传输的场合。本文主要介绍该系统设计思想与实现方法。

### 2 系统设计思想

1)系统的开发运行环境:由于 Windows95 的流行以及它在通信、多媒体处理上的优良性能,我们采用它作为我们的开发、运行平台。硬件配置上包括 MTV2000 视频采集卡,彩色摄像机、MODEM。采用的开发工具是 Visual C++, WinSocket, TAPI 等。

2)系统结构设计:我们所设计的图像传输系统采用面向对象的设计思想、模块化设计。系统的结构框图如图 1 所示:

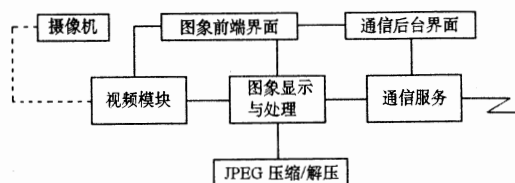


图1 系统模块组成

国际标准化组织(ISO)为所有数据通信网络通信功能建立了一个参考模型,即开放系统互连参考模型(OSI-RM),这是数据通信软件设计的基础。在实际设计时,我们结合图像传输的自身特点,对该分层体系结构作了适当的归并和简化,将系统分为三层:数据传输层,提供面向包的无差错数据传输;应用服务层(API),提供文件传输、视频服务、图像处理与压缩等服务;应用层为用户提供一个基于图形的直观清晰的操作界面,并提供脚本化语言(Script),实现自动批命令方式操作,以提高用户的工作效率,避免重复性工作。

在结构组织上,我们把图像传输系统分为两大部分:图像处理显示前端和后台通信服务器,前端与通信服务器之间通过DDE技术进行通信。图像前端负责采集和输入图像数据并进行处理、压缩,然后将处理完毕的图像数据递交给通信服务器传出去;通信服务器负责完成通信设置、拨号建立连接、发送接收图像文件、拆除连接,在成功接收到图像之后,请求图像前端显示或解压缩。图像前端与通信服务器之间互为客户端/服务器关系,但它们又是相互独立的,可以单独运行完成各自的工作。一般情况下,用户只需工作于图像前端,只在必要时(如诊断通信故障)切换到通信界面。

### 3 系统功能设计与实现

从图1我们可以看到,图像传输系统主要包括以下几个功能:视频控制与截取、图像显示与处理、JPEG压缩/解压、通信服务。

- 视频控制与截取包括视频设置、视频播放、暂停、开启、关闭以及视频截取功能。系统通过视频采集卡与外部摄像机进行连接,并用与设备无关的Windows多媒体控制接口MCI和Video for Windows实现对外部视频信号源的控制。视频捕获是指在视频播放过程中选取任一帧以图像文件形式保存。

- 图像显示处理功能:完成从摄像机、远端主机或其它应用程序中获取的图像存储与显示。由于输入设备的限制(如摄像机的噪声影响),图像在发送之前还要经过处理。所谓图像处理就是在连续色调图像上进行各种操作,最终使图像达到最佳效果。我们提供的图像处理功能主要包括:(1)图像编辑:指对图像进行剪辑、粘贴、文字标注等。(2)图像校正与增强:指对图像的RGB颜色分量调整,亮度、对比度的调整,图像的灰化等。(3)图像格式转换:向各种流行图像格式的转化,以便利用其他页面处理程序对图像进行进一步处理。

- JPEG压缩/解压:编码时采用JPEG推荐的基于DCT的有失真编码系统,再进行Huffman编码后输出图像数据;解码时相反。我们针对用户不同的需求,提供硬件实现和软件实现两种解决方案。

- 通信服务:主要包括终端仿真、文件传输、连接管理三大功能。终端仿真指的是采用仿真软件使PC机模拟主机的屏幕控制和键盘,其目的是为了便于和主机连接并访问各种的资源。文件传输支持多种文件传输协议,以检错重发的方法一次传输一个文件或一批文件。连接管理包括通信硬件设置、初始化、拨号建立连接、拆除连接等功能。

通信服务建立在开放的电话接口标准TAPI(Telephony API)基础上。TAPI是由Microsoft和Intel联合开发的一组应用编程接口(API),它是WOSA(Windows Open Services Architecture)的一个部分,它主要是为Windows平台提供“个人电话”(personal telephony)服务。借助于TAPI,应用程序可以呼叫、被告知入站的呼叫、回答呼叫、设置各种开关等。传统的串行通信编程一般采用Hayes AT命令集,开发者需要自己维护一个由各类Modem的AT命令集组成的数据库。而TAPI为所有通信硬件提供了一个公共的接口,在并不牺牲硬件性能的同时,把程序员从繁琐的、不直观的AT命令中解脱出来,实现了通信设备的硬件无关;另外,由于提供了一组通信设置、硬件设置公共对话框和Modem安装功能,减轻了开发人员的工作量。目前,我们利用TAPI和WinSocket支持三种类型的连接:(1)通过公用电话网(PSTN)进行双向图像传输;(2)通过对Internet ftp的支持,可以从FTP主机上载、下载图像文件;(3)支持Windows消息接口MAPI,可以利用任何一个支持MAPI的服务器程序如Microsoft Mail,将图像以电子邮件形式发给局域网上的用户。

- 用户界面:提供菜单操作界面和图形按钮,集成图像获取、处理、发送于一体;提供脚本化语言,支持通信过程中重复操作的自动化。目前,通信的安全性已经形成共识。为此,我们在参考其它类似系统的做法后提供了安全机制:包括用户口令识别、通信日志跟踪及文件加密机制,以确保系统安全性。

### 4 结束语

通过对图像JPEG压缩的支持,静止图像传输系统实现了高清晰度图像双向高速传输。传输同一图像文件,比其它通信软件快5~10倍以上。图像前端和一体化功能设计,隐藏了大量通信细节,提高了系统的可用性。由于支持TAPI,使得系统能够适应通信设备的变化。

目前,本文所描述的“静止图像传输系统”还处于实验阶段,其功能和界面还有待于进一步扩充完善,下一步我们将在Internet邮件支持和多平台移植上做一



些尝试,并将多媒体数据库技术应用于该系统,使之成为功能更全面、应用范围更广的远程信息系统。

### 参考文献

- 1 宋云烟等《多媒体程序设计技术》,西安交通大学,1995
- 2 Coad P, Yourdon E, Object Oriented Analysis, Prentice

Hall, Eaglewood Cliffs, New Jersey 1990

- 3 麦帆:面向对象程序设计风范与软件开发方法,计算机科学 1992,19(4)
- 4 串行通信 C 程序员指南, Joe Campbell, 清华大学出版社译, 1995

## CIMS 工程数据库的分布式在线事务处理控制系统

俞 勇 黄上腾

(上海交通大学计算机科学与工程系 上海 200030)

**摘 要** 本文介绍一个 CIMS 工程数据库的分布式在线事务处理控制系统 DOTS, 它是一个基于客户/服务器结构的、能与 CIM 全局协调的集成数据分布式在线事务处理控制系统的原型。

**关键词** 面向对象数据库 事务处理 客户/服务器 分布式

### 1 引言

在线事务处理(OLTP)是以数据库为中心,在客户/服务器模式下的一个非常主要的应用。目前,OLTP 系统主要应用于商业事务系统,包括预订票系统、销售点系统、跟踪系统、库存控制、股票经纪人工作站和制造厂车间控制系统等,在这些应用中,可以看出它们都有一个非常明显的特点——在任何情况下必须有快速的响应时间(一般为 1 至 3 秒钟内)。因此,这就要求 OLTP 系统在数据库的安全性和完整性上加以严密的控制,整个系统的可靠性和有效性必须非常高,数据必须保持一致性。同时,由于以数据库为中心的客户/服务器技术的不断成熟和发展,为真正实现实用化的 OLTP 系统提供了良好的基础。另一方面,为了能够支持大规模的 OLTP 系统的稳定的运行环境,在大型机系统里已使用了 TP 监视器,它提供了用于保证 OLTP 系统在任何应用环境中的任何服务,比如反应快、重用性好以及优化的管理,并且,随着 OLTP 系统转移到客户/服务器平台,一批新的 TP 监视器正在涌现,以使得新的环境对任务艰巨而繁重的应用非常友好。因此,对于 OLTP 控制系统的研究与开发是非常紧迫的、急需的,尤其是在计算机集成制造系统 CIMS 中支持并行工程的实现,其应用前景非常广泛。

本文介绍一个 CIMS 工程数据库的分布式在线事务处理控制系统 DOTS,它是一个基于客户/服务器结构的、能与 CIM 全局协调的集成数据分布式在线事务处理控制系统的原型。

从逻辑角度,DOTS 系统主要包括以下两个子系统:

- 以工程数据库为中心、基于客户/服务器结构的分布式数据库管理子系统
- 以 TP 监视器为核心的分布式事务处理控制子

系统

本文的第 2 节讨论有关分布式数据库管理子系统,第 3 节讨论有关分布式事务处理控制子系统,第 4 节从实现角度描述 DOTS 系统的四层体系结构,第 5 节是本文的结束。

### 2 分布式数据库管理子系统

DOTS 系统的数据库语言 OOSQL-II 是一个分布式面向对象的数据库语言。在数据模型的设计中,考虑到 CIMS 的数据集成的需要,我们参考了产品数据交换标准 PDES 中描述产品数据的方法,特别是其逻辑层中的形式化建模语言 EXPRESS,使得用 EXPRESS 描述的产品模型能被方便地翻译成数据库语言 OOSQL 表示的形式,反之亦然(但需要加一些限制)。显然,DOTS 系统的数据库语言 OOSQL-II 同样也继承了 OOSQL 的这一特点。这不仅有利于 CIMS 数据的集成,而且使 DOTS 更接近 CIMS 产品数据管理的要求。

DOTS 系统是基于客户/服务器结构的、并建立在基于标准 SQL 语言的关系式数据库管理系统之上的,我们采用分布式数据目录管理的方法来完成其全局分布数据模型的转换和面向对象数据模型到关系式数据模型的转换以及查询分解和优化,其中,不仅包括对象分布、对象与元组之间的对应关系,还包括对象的行为,如对象的消息(message)和方法(method)等。

### 3 分布式事务处理控制子系统

客户/服务器模式对于开发和实现网络型分布式环境是非常方便的。它通过通信网络把多台计算机连接起来,用户就可以存取和共享所有的分布式资源,以及有效的分布计算工作负载。从客户/服务器结构的度,按照客户和服务器之间的工作负载分布,客户主要用来管理输入、输出、一部分应用逻辑功能以及局部数据存取功能,而服务器是完成数据库存取功能和可共享的应用功能。最重要的是,服务器的作用是实现工作

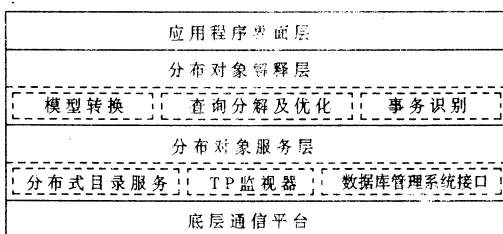
负载分布及在客户请求的情况下对集中式资源存取的管理、同步和控制。传统的事务处理适用于集中式系统。近年来,随着分布式计算结构和开放系统的成熟,事务处理技术得到了快速发展,如事务处理监视器(TP 监视器)的出现。

TP 监视器是一个用于事务处理的管理环境,它能够管理单个或多个服务器上的资源,优化事务的管理,控制事务的运行和平衡系统的负载,从而监督分布式事务的各个方面。DOTS 系统中,我们提供一个事务应用程序接口(API),用于支持分布式事务处理的并发控制和一致性控制。从事务角度,即以事务为基本工作单元,而非传统的以 SQL 语句为基本工作单元,确保系统的完整性和安全性。

为了实现事务分布的透明性,DOTS 系统采用远程过程调用(RPC)概念。RPC 是事务处理环境下的客户/服务器模式的一个延伸。客户/服务器模式将应用分为客户和服务器两个部分,客户和服务器之间得到通信方式是通过消息的请求/应答的方式完成的,即客户发送一个事务消息给服务器,并等待其回答,就像一个应用程序调用一个工程且得到返回一样,RPC 的透明性表现在远程调用的语义十分接近于本地调用的语义,用户在使用时不必了解网络上事务的分布情况。

#### 4 体系结构

从实现角度,DOTS 系统把以工程数据库为中心、基于客户/服务器结构的分布式数据库管理系统和以 TP 监视器为核心的分布式事务处理控制系统进行有机地组合,将其分为四层体系结构,如图所示。



##### 4.1 应用程序界面层

应用程序界面层运用面向对象用户界面(OOUI)技术提供应用开发环境和系统管理环境。应用开发环境包括编程环境和分布式面向对象数据库浏览器,DOTS 系统允许用户在宿主语言 C 或 C++ 中嵌入数据库语言 OOSQL-II 编写复杂的 OLTP 应用程序,也可以数据库浏览器直接对分布式面向对象数据库进行查询。系统管理环境提供系统安装、用户授权、目录操作、出错恢复和数据库管理等,同时还提供 TP 监视器的仿真程序,用于实时观察系统优化和运行情况。

##### 4.2 分布对象解释层

分布对象解释层主要包括模型转换、查询分解及优化和事务识别等三个部分。

由于 DOTS 系统是建立在基于标准 SQL 语言的关系式数据库管理系统之上的,而 DOTS 系统所提供的数据库语言 OOSQL-II 是一个面向对象数据库语言,这两种数据模型——关系模型和面向对象模型——在概念、结构和语义上都存在着很大的差别,尤其是面向对象中的有关类及其相应的一些概念,如消息和方法等,这在关系模型中是无法表示的。分布对象解释层中的模型转换就是提供从面向对象模型到关系模型的转换,以致保证在关系模型上实现面向对象模型中的类、复杂对象、对象标识、封装性、继承性、和重载等概念。

分布式数据库查询的工作分为:查询分解和查询优化。查询分解是把从应用程序对数据库的访问操作——全局查询转换成对类的局部查询,在面向对象模型中,对象间的联系隐含在类层次和类继承中,这有利于查询表述的简化,但是,系统必须具有“自导向”的能力,以确保对象之间联系的正确性。通常按照每一查询的类层次和类继承关系,可以划分若干个局部查询,这种划分主要根据检索条件,确定要进行几个局部查询。一般地,查询分解只是机械地翻译工作,不侧重于效率分析,所以查询处理的第二步需要进行查询优化。一个分布式查询优化可以分成两个独立的问题:结点与结点之间访问策略的分布(主要考虑传输问题)和决定每个结点处的局部访问策略。通常传输的要求更重要,它可以根据成本和响应时间这两方面来估价,最终依赖于网络传送数据量的大小。这就要求增加本地的处理,以减少中间结果的网络传送量。在关系模型下,可以把具有“半连接”的连接操作进行等价变换,它具有减少所需数据传输的特性。而在面向对象模型中,“自导向”的顺序基本上决定了结点之间访问策略的分布,前一个局部查询的结果用于后一个局部查询的“自导向”,必须使第一个局部查询的结果尽可能得小,依据这一原则确定在“自导向”路径上是从父类到子类还是从子类到父类。

有关事务识别是提供事务处理的关键,由于 DOTS 系统是以事务作为最基本的运行单位,这里我们提出事务标识这一概念,它是具有把事务看成自己本身而非依赖于其它值而存在的能力。事务标识由系统自动生成,在整个系统内保持唯一且不变。

##### 4.3 分布对象服务层

分布对象服务层主要包括分布式目录服务管理、TP 监视器和数据库管理系统接口等三个部分。

在 DOTS 系统中,分布式目录服务管理主要在标准 X.500 的基础上实现在分布式计算环境中的面向对象目录服务。其目录服务主要是要完成以下两方面的工作:

- 对象实例在分布式计算环境中的定位与查找
- 对象所属类本身信息的描述,包括类结构和类

之间的关系

由于对象的实例分散在各个不同的节点,因此类的字典信息也被分布式地存放于不同的节点,动态地对当前对象实例的状态进行解释。由于本系统的面向对象数据库(OODB)是建立在关系数据库(RDB)的基础上,所以对象所属类本身信息的描述还担负着从 OODB 到 RDB 的转换任务。

TP 监视器是在在线事务处理的实际要求的引发下产生的一种介于操作系统与用户应用之间的软件环境,它起着均衡工作负载,维持操作系统的正常运行,并且自动维护分布事务的原子性、一致性、隔离性和持久性(ACID 特性),提高程序员开发效率的作用。TP 监视器是基于客户/服务器结构的,在服务器方面,TP 监视器提供运行远程过程调用(RPC)的通用服务器框架,接受客户端的事务请求,并运行相应的存储过程。此外,TP 监视器的运行环境采用两阶段提交协议来支持 ACID 特性的自动维护。

数据库管理系统接口提供一个独立于底层基于标

准 SQL 关系数据库的接口。

#### 4.4 底层通信平台

底层通信平台使用了 OSI-TP 规范,并提供采用两阶段提交协议来保证事务的原子性、一致性、隔离性和持久性(即 ACID 特性)。它可以通过不同的协议(如 IPX、TCP/IP 等)实现不同节点之间的通信。

#### 5 结束语

本文主要对 CIMS 工程数据库的分布式在线事务处理控制系统的体系结构进行分析与设计,目前,DOTS 系统原型已经实现。但是,如何进一步完善系统功能、提高系统性能和优化事务管理等等,还有待于更深入地研究与探讨。

#### 参考文献

- 1 Robert Orfali & Dan Hankey, 《Client/Server Survival Guide with OS/2》, 1993
- 2 Adrian Tang & Sophia Scoggins, 《Open Networking with OSI》, 1992

## Oracle 7 关系型分布式数据库的两阶段提交机理

刘达项 石冰心

(华中理工大学计算机中心 武汉 430074)

**摘要** 本文在简单地介绍了 Oracle 7 之后,引出了两阶段提交机理的概念,详细地阐明了两阶段提交机理的基本原理,扼要地说明了 Oracle 对提交过程中发生的故障的处理方法,接着详细地解释了提交参与节点组成的会话树的概念和方法,最后以一个实例说明两阶段提交机制的工作过程。

**关键词** Oracle 分布式数据库 两阶段提交 分布式事务 会话树 全局协调节点 局部协调节点 提交点现场 提交点强度 回退

#### 1 引言

Oracle 公司是世界上排名仅列于 Microsoft 之后的第二大软件公司,该公司发布的 Oracle 7 关系型分布式数据库是当今最流行的具有高性能、开放式和分布式体系结构的数据库管理系统,它支持多种标准和网络通信协议,并且配有强大的数据库开发工具,继发布 CDE2 协同开发环境之后,又发布了高性能的 Developer/2000。Oracle 7 基于客户/服务器体系结构,拥有高效的 GUI 开发界面。Oracle 7 数据库相当好的实现了关系数据库创始人 E. F. Codd 博士提出的十二条基本准则,为优秀的超完备关系型软件产品。Oracle 7 服务器采用了最新的计算机技术,使之性能更为优秀更安全可靠。Oracle 7 利用两阶段提交方法能够自动的控制和监督分布式事务的提交或回退,保持全局数据库(参与此事务的所有分布数据库)的完整性。

在网络环境下,必须协调所有有关的节点作为一

个整体来提交或回退分布式事务,两阶段提交机理保证这些节点要么一起提交要么一起回退某事务。事务中含有 DML 语句或远程过程调用以及完整性约束、触发器引起的隐含 DML 操作都会受到 Oracle 的两阶段机理的保护。

#### 2 提交的两阶段及故障处理

Oracle 的两阶段提交包括准备和提交两个界限分明的阶段。

##### 1) 准备阶段

全局协调节点向参与节点发出准备请求,参与节点通过发出“准备好”、“只读”、“失败”响应请求,并且每个参与节点完成如下过程:请求后续节点作准备、分配事务提交所需的资源、记录事务引起的数据库的各种改变等。准备好后,参与节点等待“提交”或“回退”信号而不会单方面作出是否提交或回退的决定。如参与节点未能成功地完成“准备”请求,此节点释放当前为事务准备的资源并且回退事务的本地部分再向请求节点发“失败”信号。同时参与此分布事务的其它节点也

回退事务,这就保证了全局数据库的数据完整性。

## 2) 提交阶段

提交某分布式事务的第二阶段即为提交阶段。此分布式事务引用的所有节点必须已经保证有必要的资源来提交事务后才能进入提交阶段。在此阶段,所有的节点参与事务的提交即提交此事务的本地部分。提交完成后分布式数据库的所有有关节点上的数据相互间是一致的。

## 3) 故障处理

在两阶段提交期间总有可能出现故障,典型故障有机器故障、网络故障、软件故障和本地实例失败等。Oracle 的事务两阶段提交机制可以自动的且透明的处理故障,从故障中恢复提交前的全局数据库的状态,无需人工干预。

每个 Oracle 数据库都有一个等待事务表,在分布式事务的提交过程中它存储了所有有关的状态信息,一旦发生故障,Oracle 能自动地从等待事务表中把对应于未提交的事务的有关信息消除。

## 3 会话树

会话树由参与分布式事务提交的所有节点组成,事务中的语句发出提交要求时,Oracle 为事务定义了一会话树,会话树中的所有节点都扮演如下的一至多个角色:客户、数据库服务器、全局协调节点、局部协调节点、提交点现场。具体为何角色,由事务发生地、节点的提交点强度、是否为查询节点等确定。在数据库启动时 Oracle 为 Oracle 数据库系统中的每个数据库分配一个提交点强度。

全局协调节点是会话树的根节点,分布式事务由它产生。全局协调节点把所有的 SQL 语句和远程调用直接发送到相关节点,从而形成会话树。然后它与所有直接引用的节点(提交点现场除外)通信以准备事务的提交。如所有节点准备成功,它再请求提交点现场发出全局提交的信号。

局部协调节点必须引用其它节点的数据以完成事务的相关任务,它负责协调它所引用的节点,把信息中继到这些节点,从这些节点收集信息再把这些信息发回到它的父节点。

两阶段提交机制总要指定会话树中的某一节点为提交点现场。由它决定是提交还是回退分布式事务,而全局协调节点确保所有节点能以与现场提交点相同的方式完成事务的提交。提交点现场不参与提交的一个阶段,以便保护重要数据。

## 4 两阶段提交举例

下面举一个简单的成功提交的例子说明上述的两阶段提交机制。

假设有一个跨国公司的销售数据库(SALES)、仓储数据库(WAREHOUSE)分布两地,它们的域名分别为 SALES.COMPANY.COM、WAREHOUSE.COMPANY.COM,由 Oracle 分布式数据库通过 NOVELL 网络连接,采用 TCP/IP 网络通信协议。

修改销售数据库中的数据时,仓储数据库中相应的数据也必须更新。在销售数据库端某客户机利用 Oracle FORMS 输入订单数据,保存时系统执行了一系列 SQL 语句以更新库存。这些 SQL 语句就组成了一个分布式事务。两阶段提交保证所有这些 SQL 语句作为一个整体执行,防止订单输入到了销售数据库而仓储数据库未作相应的修改,以便维持全局数据库中相关数据的一致性。本事务由销售数据库端的 FORMS 发出,故销售数据库为全局协调节点,订单数据修改了销售数据库和仓储数据库,这两个数据库都被引用,为数据库服务器,并且销售数据库为仓储数据库的客户,也为提交点现场,提交点现场由全局协调器待 FORMS 发出 COMMIT(提交)语句后确定。

本例中,会话树由两个节点组成:销售数据库和仓储数据库,销售数据库为根节点。

建立了会话树后,全局协调节点即销售数据库向参与节点即仓储数据库发“准备”请求信号,仓储数据库发回是否同意请求的信号即“失败”或“准备好”信号,如收到“失败”信号,则执行回退操作,订单数据保存失败,两个数据库中的相应数据均未修改。如收到“准备好”信号,销售数据库服务器提交订单数据,并且通知仓储数据库服务器也执行提交操作,提交后通知销售数据库,至此,订单数据的提交结束,数据提交成功。

## 5 结束语

两阶段提交机制由 Oracle 具体实现时是自动完成的,对用户来说是完全透明的。两阶段提交机制确保了分布式数据库系统的全局数据库的完整性和整个系统的可靠性。了解了 Oracle 的两阶段提交机制,可以利用它的这些特性,有利于更好的开发 Oracle 应用系统和对分布式数据库系统进行优化配置。

## 参考文献

- 1 Oracle 7 Server Concepts Manual, Oracle Corporation
- 2 Oracle Forms Reference Manual V4. 5, Oracle Corporation

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)



# 基于集合理论的 KDD 方法

马建军 陈文伟

(国防科技大学系统工程系 长沙 410073)

**摘 要** 本文运用集合理论对 KDD 进行了描述,介绍了基于集合理论从数据库中发现分类规则的方法,并给出了实现算法和一个应用实例。

**关键词** 集合理论 数据库 知识发现 分类规则

## 1 引言

从大规模数据库中寻找和分析大量的数据,从中识别和抽取隐含的、潜在的有意义有价值的规则知识,即基于数据库中的知识发现(Knowledge Discovery in Database; KDD)近年来在理论模型、发现方法和实际应用等方面都得到了很大的发展。数据库中蕴含的规则知识主要有:分类规则、特征规则、函数依赖、相关关系等。

发现知识的方法主要分为两大类:

- 机器学习方法
- 统计分析方法

前者主要是定性分析,即从知识基表中提取所有或大多数元组满足的规则,后者主要是定量分析,提取属性之间的函数依赖与相关关系等。这些发现方法中所使用的算法主要有:归纳学习算法、神经网络、基于集合理论的方法、基于范例的推理(CBR)、遗传算法和统计算法。

## 2 KDD 的描述

基于集合理论,KDD 可以用一个 5 元组来描述, $S=(R, At, V, f, K)$ 。其中

$R$ :是一个非空集合,表示数据库中所有的记录,用  $e_i$  表示  $R$  中具体的记录对象。

$At$ :是一个非空集合,表示数据库中的全部字段,用  $a_i$  表示  $At$  中具体的字段。

$V$ :字段的  $U$  值组成的集合。 $V=\bigcup(V_a)$ ,其中  $V_a$  是字段  $a$  的定义域, $a \in At$ 。

$f$ :是  $a$  和  $e$  的函数, $f(e, a)$  的值确定记录  $e$  的字段  $a$  的取值。

$K$ :表示发现的规则知识,规则知识的基本表示形式为:

$rule(No, rule-body, cf)$

- $No$  标识发现的规则知识的序号
  - $rule-body$  包含规则的前提和结论,如果一条规则没有前提,那么它表示一个事实
  - $cf$  表示规则的可信度,用  $[0, 1]$  之间的实数表示
- 从表 1 中进行知识发现可以描述成以下形式:

$S=(R, At, V, f, K)$

$R=\{e_1, e_2, e_3, e_4, e_5, e_6, e_7, e_8, e_9, e_{10}\}$

$At=\{a_1, a_2, a_3\}$

$V=\{v_{10}, v_{11}, v_{12}, v_{20}, v_{21}, v_{30}, v_{31}, v_{32}\}$

$f(e, a)$  的值由表 1 给出,

表 1

R	a1	a2	a3
e1	v11	v20	v32
e2	v10	v21	v30
e3	v12	v20	v30
e4	v11	v21	v30
e5	v11	v20	v32
e6	v12	v20	v30
e7	v10	v21	v31
e8	v11	v21	v31
e9	v11	v20	v32
e10	v10	v21	v31

## 3 关系的划分

通常在数据库中,对其中的记录根据某些字段的值是不能够有效地进行区分的,如在表 1 中,根据字段  $a_1$  的值不能够区分记录  $\{e_1, e_4, e_5, e_8, e_9\}$ 、 $\{e_2, e_7, e_{10}\}$  和  $\{e_3, e_6\}$ ;根据字段  $a_1$  和  $a_2$  的值不能够区分记录  $\{e_1, e_5, e_9\}$ 、 $\{e_2, e_7, e_{10}\}$ 、 $\{e_3, e_6\}$  和  $\{e_4, e_8\}$ 。

定义 1:对于  $\forall a \in A$ , ( $A$  中包含一个或多个字段,  $A \subseteq At$ ),  $e_i \in R, e_j \in R, f(e_i, a) = f(e_j, a)$  成立,那么我们称  $e_i$  和  $e_j$  基于字段集  $A$  等价。

定义 2:在数据库记录集合  $R$  中,基于某一个字段集  $A$  的所有等价的记录的集合,我们称之为基于字段集  $A$  的等价类。

定义 3:在数据库记录集合  $R$  中,基于字段集  $A$  的所有等价类的集合,我们称之为  $R$  的等价关系,用  $\tilde{A}$  表示,  $\tilde{A} = \{E_i | E_i \text{ 为 } R \text{ 基于 } A \text{ 的等价类}, i=1, 2, \dots\}$ 。

定义 4:在数据库记录集合  $R$  中,根据等价关系  $\tilde{A}$ ,我们可以将  $R$  中的全部记录按  $\tilde{A}$  中的等价类进行

分类,属于同一等价类的记录归为一类,分类的结果称为R基于字段集A的划分,表示为 $R=\{R_1R_2,\dots\}$ 。

例如表1中的记录:

基于字段 $a_1$ ,可以被划分为三类:

$$E_1=\{e_1,e_4,e_5,e_8,e_9\}; E_2=\{e_2,e_7,e_{10}\}; E_3=\{e_3,e_6\}$$

基于字段集 $\{a_1,a_2\}$ 可以被划分为四类:

$$E_1=\{e_1,e_5,e_9\}; E_2=\{e_2,e_7,e_{10}\}; E_3=\{e_3,e_6\}; E_4=\{e_4,e_8\}$$

等价类的描述:

对一个等价类 $E_i$ ,我们用 $Des(E_i)=(a,v), a \in A, v \in V_a$ 对它的特性进行描述。

如表1基于字段 $a_1$ 的等价类 $E_1=\{e_1,e_4,e_5,e_8,e_9\}$ ,  $Des(E_1)=(a_1=v_{11})$

同理,对基于 $(a_1,a_2)$ 的等价类 $E_1=\{e_1,e_5,e_9\}$ ,  $Des(E_1)=(a_1=v_{11}, a_2=v_{20})$

#### 4 规则生成

在表1中,假设基于 $A=\{a_3\}$ 对其进行划分得

$$Y=\{Y_1,Y_2,Y_3\}$$

$$Y_1=\{e_1,e_5,e_9\}; Y_2=\{e_7,e_8,e_{10}\}; Y_3=\{e_2,e_3,e_4,e_6\}$$

我们又对R基于 $A=\{a_1,a_2\}$ 进行划分得 $E=\{E_1,E_2,E_3,E_4\}$

$$E_1=\{e_1,e_5,e_9\}; E_2=\{e_2,e_7,e_{10}\}; E_3=\{e_3,e_6\}; E_4=\{e_4,e_8\}$$

为了说明对R的两个划分Y和E之间的关系,我们用下面几个概念进行说明:

① $Y_j$ 中至少可能包含E中的等价类有:

$$\overline{AY_j} = \bigcup \{E_i | E_i \in E \text{ 且 } E_i \subseteq Y_j\} \quad \dots\dots(1)$$

$\overline{AY_j}$ 是E中被 $Y_j$ 包含的等价类中的记录的并集。

② $Y_j$ 中最多可能包含E中的等价类有:

$$\overline{AY_j} = \bigcup \{E_i | E_i \in E \text{ 且 } E_i \cap Y_j \neq \emptyset\} \quad \dots\dots(2)$$

$\overline{AY_j}$ 是E中与 $Y_j$ 的交集非空的等价类中的记录的并集。

③显然 $\overline{AY_j} - \overline{AY_j}$ 中的记录根据A中字段的值是不能确定它们是否属于等价类 $Y_j$ 的。

$$\textcircled{4} \text{ 令 } \alpha_A(Y_j) = \frac{|R| - |\overline{AY_j} - \overline{AY_j}|}{|R|}$$

其中 $|R|$ 和 $|\overline{AY_j} - \overline{AY_j}|$ 分别表示集合R、 $(\overline{AY_j} - \overline{AY_j})$ 的记录数。

$\alpha_A(Y_j)$ 的值反映了R中的全部记录能够根据A中字段的值就能确定其属于或不属于 $Y_j$ 的比例,也即对R中的任意一条记录,根据A中字段的值确定它属于或不属于 $Y_j$ 的可信度。

显然 $0 \leq \alpha_A(Y_j) \leq 1$

· 当 $\alpha_A(Y_j)=1$ 时,R中的全部记录根据A中字段的值可以确定其是否属于 $Y_j$ 。

· 当 $0 < \alpha_A(Y_j) < 1$ 时,R中的部分记录根据A中字段的值可以确定其是否属于 $Y_j$ ,而另一部分记录是不能确定其是否属于 $Y_j$ 的。

· 当 $\alpha_A(Y_j)=0$ 时,R中的全部记录都不能根据A中字段的值确定其是否属于 $Y_j$ 。

根据公式(1)得:

$$\overline{AY_1} = \{e_1,e_5,e_9\}; \overline{AY_2} = \{\}; \overline{AY_3} = \{e_3,e_6\}$$

因为 $E_1 \cap Y_1 \neq \emptyset, E_2 \cap Y_1 = \emptyset, E_3 \cap Y_1 = \emptyset, E_4 \cap Y_1 = \emptyset$

所以根据公式(2)得: $\overline{AY_1} = E_1 = \{e_1,e_5,e_9\}$

同理 $\overline{AY_2} = \{e_2,e_4,e_7,e_8,e_{10}\};$

$$\overline{AY_3} = \{e_2,e_3,e_4,e_6,e_7,e_8,e_{10}\};$$

$$\overline{AY} - \overline{AY} = \{\{\}, \{e_2,e_4,e_7,e_8,e_{10}\},$$

$$\{e_2,e_4,e_7,e_8,e_{10}\}\}$$

$$\alpha_A(Y_1) = \frac{10-0}{10} = 1.0; \alpha_A(Y_2) = \frac{10-5}{10} = 0.5;$$

$$\alpha_A(Y_3) = \frac{10-5}{10} = 0.5$$

R中的10条记录,所有记录都可以根据A中字段的值确定其是否属于 $Y_1$ ;50%的记录可以根据A中字段的值确定其是否属于 $Y_2$ ;50%的记录可以根据A中字段的值确定其是否属于 $Y_3$ 。

#### 1) 分类规则的形成

通过分析R中的两个划分E和Y之间的关系,把E视为分类条件,Y视为分类标准,我们可以得到下面的分类规则:

当 $E_i \cap Y_j \neq \emptyset$ 时,则有:

$$r_{ij}: Des(E_i) \rightarrow Des(Y_j)$$

$Des(E_i)$ 和 $Des(Y_j)$ 分别是等价类 $E_i$ 和等价类 $Y_j$ 中的记录的特性描述。

当 $E_i \cap Y_j = E_i$ 时,规则 $r_{ij}$ 成立,即是确定的,规则 $r_{ij}$ 的可信度 $cf=1.0$ 。

当 $E_i \cap Y_j \neq E_i$ 时,规则 $r_{ij}$ 是不确定的,规则 $r_{ij}$ 的可信度 $cf = \frac{|E_i \cap Y_j|}{|E_i|}$ 。

根据R的两个划分E和Y,我们可以得出:

$$Des(E_1) = (a_1=v_{11}, a_2=v_{20})$$

$$Des(E_2) = (a_1=v_{10}, a_2=v_{21})$$

$$Des(E_3) = (a_1=v_{12}, a_2=v_{20})$$

$$Des(E_4) = (a_1=v_{11}, a_2=v_{21})$$

$$Des(Y_1) = (a_3=v_{32}) \quad \alpha_A(Y_1) = 1.0$$

$$Des(Y_2) = (a_3=v_{31}) \quad \alpha_A(Y_2) = 0.5$$

$$Des(Y_3) = (a_3=v_{30}) \quad \alpha_A(Y_3) = 0.5$$

当E作为前提, $Y_1$ 为结论,可以得到下列规则:

$$r_{11}: Des(E_1) \rightarrow (Des(Y_1)); \text{ 即 } (a_1=v_{11}) \wedge (a_2=v_{20}) \rightarrow (a_3=v_{32}) \quad (cf=1.0)$$

$$r_{21}: \text{不存在; 因为 } E_2 \cap Y_1 = \emptyset$$

$$r_{31}: \text{不存在; 因为 } E_3 \cap Y_1 = \emptyset$$

$$r_{41}: \text{不存在; 因为 } E_4 \cap Y_1 = \emptyset$$

因为 $\alpha_A(Y_1)=1.0$ ,所以根据划分E判断R中的记录是否属于 $Y_1$ 类的可信度为1.0。

当E作为前提, $Y_2$ 作为结论,可以得到下列规则:

$$r_{12}: \text{不存在; 因为 } E_1 \cap Y_2 = \emptyset$$

$$r_{22}: Des(E_2) \rightarrow (Des(Y_2)); \text{ 即 } (a_1=v_{10}) \wedge (a_2=v_{21}) \rightarrow (a_3=v_{31}) \quad (cf=0.67)$$

$r_{32}$ : 不存在; 因为  $E_3 \cap Y_2 = \emptyset$

$r_{42}$ :  $\text{Des}(E_4) \rightarrow (\text{Des}(Y_2))$ ; 即  $(a_1 = v_{11}) \wedge (a_2 = v_{21}) \rightarrow (a_3 = v_{31})$  ( $cf = 0.5$ )

因为  $\alpha_A(Y_2) = 0.5$ , 所以根据划分 E 判断 R 中的记录是否属于  $Y_2$  类的可信度为 0.5。

当 E 作为分类条件,  $Y_3$  作为分类的结果, 可以得到下列规则:

$r_{13}$ : 不存在; 因为  $E_1 \cap Y_3 = \emptyset$

$r_{23}$ :  $\text{Des}(E_2) \rightarrow (\text{Des}(Y_3))$ ; 即  $(a_1 = v_{10}) \wedge (a_2 = v_{21}) \rightarrow (a_3 = v_{30})$  ( $cf = 0.33$ )

$r_{33}$ :  $\text{Des}(E_3) \rightarrow (\text{Des}(Y_3))$ ; 即  $(a_1 = v_{12}) \wedge (a_2 = v_{20}) \rightarrow (a_3 = v_{30})$  ( $cf = 1.0$ )

$r_{43}$ :  $\text{Des}(E_4) \rightarrow (\text{Des}(Y_3))$ ; 即  $(a_1 = v_{11}) \wedge (a_2 = v_{21}) \rightarrow (a_3 = v_{30})$  ( $cf = 0.5$ )

因为  $\alpha_A(Y_3) = 0.5$ , 所以根据划分 E 判断 R 中的记录是否属于  $Y_3$  类的可信度为 0.5。

## 2) 实现算法

需要用户输入的信息:

(1) 确定作为前提条件的字段(属性)集 CA 和作为结论的字段(属性)集 DA

(2) 给定  $\alpha_A(Y_j)$  的一个阈值 cp

说明:

$A_i$  表示 CA 中第 i 个字段; A 表示 CA 中实际用于分类的字段集;  $\text{flag}(i)$  表示字段  $A_i$  是否被选择;  $D_i$  表示 D 中第 j 个等价类;  $R'$  表示有待于进一步分类的记录集。

算法:

① 初始化, 基于 DA 对 R 进行划分得  $D = \{D_1, D_2, \dots\}$ , 即将 R 分为  $D_1, D_2, \dots$  等类;

② 初始化,  $\text{flag}(i) = n, A = \emptyset, R' = R$ ;

③ 对 R 分别基于 CA 中的字段  $A_i$  进行划分得  $CA_i = \{C_{i1}, C_{i2}, \dots\}$  并计算  $\alpha_{A_i}(D_j)$  的值:

$$\alpha_{A_i}(D_j) = \frac{|R| - |\bar{A}_i D_j - \underline{A}_i D_j|}{|R|} \quad \text{其中 } i = 1, 2, \dots, |CA|; j = 1, 2, \dots, |D|;$$

④ 选择  $\text{flag}(i) = n$  且  $\alpha_{A_i}(D_j)$  的值为最大的字段  $A_i$ , 令  $A = A \cup A_i$ , 并根据 R 基于  $A_i$  的划分  $CA_i = \{C_{i1}, C_{i2}, \dots\}$  进行分类:

如果  $C_{im} \subseteq D_j$  且  $C_{im} \cap R' \neq \emptyset$ , 则得到确定性规则:

$r_j$ :  $\text{Des}(C_{im}) \rightarrow (\text{Des}(D_j))$ ;  $m = 1, 2, \dots, |CA_i|, j = 1, 2, \dots, |D|$ ; 令  $R' = R' - C_{im}$

⑤ 如果  $R' \neq \emptyset$ , 则对 R 基于 A 进行划分得  $C = \{C_1, C_2, \dots\}$ , 然后对 R 进行分类, 并计算  $\alpha_{A_i}(D_j)$  的值;

⑥ 如果  $R' \neq \emptyset$  且  $\alpha_{A_i}(D_j) < cp$ , 转第 4 步;

⑦ 如果  $R' \neq \emptyset$  则得到不确定的规则:

当  $C_i \cap D_j \neq \emptyset$  且  $C_i \cap D_j \neq C_i$  时, 得不确定的规则:

$r_j$ :  $\text{Des}(C_i) \rightarrow (\text{Des}(D_j))$  (可信度  $cf = \frac{|C_i \cap D_j|}{|C_i|}$ )

⑧ 汇总生成的分类规则, 对同一类规则进行合取;

⑨ 结束。

## 5 应用实例

下面是我们应用上述的理论从一个病例数据库中 发现肺炎和肺结核两种疾病的分类规则的例子。每个病例都含有五种症状: 发烧、咳嗽、X 光所见阴影、血沉、听诊。这五种症状加上病例号、诊断结果分别对应病例数据库表的字段  $a_0, a_1, a_2, a_3, a_4, a_5, a_6$ 。肺炎和肺结核两种疾病的部分病例集见表 2:

表 2 肺炎和肺结核病例集

$a_0$ (病例号)	$a_1$ (发烧)	$a_2$ (咳嗽)	$a_3$ (X 光所见)	$a_4$ (血沉)	$a_5$ (听诊)	$a_6$ (诊断结果)
1	高	剧烈	片状	正常	水泡音	肺炎
2	中度	剧烈	片状	正常	水泡音	肺炎
3	低	轻微	点状	正常	干鸣音	肺炎
4	高	中度	片状	正常	水泡音	肺炎
5	中度	轻微	片状	正常	水泡音	肺炎
6	无	轻微	索条状	正常	正常	肺结核
7	高	剧烈	空洞	快	干鸣音	肺结核
8	低	轻微	索条状	正常	正常	肺结核
9	中	轻微	点状	快	干鸣音	肺结核
10	低	中度	片状	快	正常	肺结核

根据描述:  $S = (R, At, V, f, K)$  有:

$R = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$  (病例号唯一标识数据库中的每条记录)

$At = \{a_0, a_1, a_2, a_3, a_4, a_5, a_6\}$

$V, f$  的值见表 2。

规则形成过程:

1) 选择字段集  $CA = \{a_1, a_2, a_3, a_4, a_5\}$  作为分类的前提条件;  $DA = \{a_6\}$  作为分类结果的字段集;  $\text{flag}(i) = n, i = 1, 2, 3, 4, 5; A = \emptyset, R' = R, cp = 0.9$ 。

2) R 基于  $DA = \{a_6\}$  进行划分得  $D = \{D_1, D_2\}$ , 其中

$D_1 = \{1, 2, 3, 4, 5\}$   $\text{Des}(D_1) = (a_6 = \text{肺炎})$

$D_2 = \{6, 7, 8, 9, 10\}$   $\text{Des}(D_2) = (a_6 = \text{肺结核})$

3) R 基于  $A_1 = \{a_1\}$  进行划分得  $CA_1 = \{C_{11}, C_{12}, C_{13}, C_{14}\}$ , 其中

$C_{11} = \{1, 4, 7\}$   $\text{Des}(C_{11}) = (a_1 = \text{高})$

$C_{12} = \{2, 5, 9\}$   $\text{Des}(C_{12}) = (a_1 = \text{中度})$

$C_{13} = \{3, 8, 10\}$   $\text{Des}(C_{13}) = (a_1 = \text{低})$

$C_{14} = \{6\}$   $\text{Des}(C_{14}) = (a_1 = \text{无})$

$\underline{A}_1 D_1 = \{\}; \underline{A}_2 D_2 = \{6\}; \underline{A} D = \{\{\}, \{6\}\}$

$\bar{A}_1 D_1 = \{1, 2, 3, 4, 5, 7, 8, 9, 10\};$

$\bar{A}_2 D_2 = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$

$\bar{A}_1 D = \{\{1, 2, 3, 4, 5, 7, 8, 9, 10\}, \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}\}$

$\bar{A}1D - \underline{A}1D = \{\{1,2,3,4,5,7,8,9,10\}, \{1,2,3,4,5,7,8,9,10\}\}$

$$\alpha_{A1}(D_1) = \frac{|R| - |\bar{A}1D_1 - \underline{A}1D_1|}{|R|} = \frac{10-9}{10} = 0.1$$

$$\alpha_{A1}(D_2) = \frac{|R| - |\bar{A}1D_2 - \underline{A}1D_2|}{|R|} = \frac{10-9}{10} = 0.1$$

同理可得:

R 基于  $A2 = \{a2\}$  进行划分得  $CA_2 = \{C_{21}, C_{22}, C_{23}\}$

$$\alpha_{A2}(D_1) = 0; \alpha_{A2}(D_2) = 0$$

R 基于  $A3 = \{a3\}$  进行划分得  $CA_3 = \{C_{31}, C_{32}, C_{33}, C_{34}\}$ , 其中

$$\alpha_{A3}(D_1) = 0.3; \alpha_{A3}(D_2) = 0.3$$

R 基于  $A4 = \{a4\}$  进行划分得  $CA_4 = \{C_{41}, C_{42}\}$ , 其中

$$C_{41} = \{1,2,3,4,5,6,8\} \quad \text{Des}(C_{41}) = (a4 = \text{正常})$$

$$C_{42} = \{7,9,10\} \quad \text{Des}(C_{42}) = (a4 = \text{快})$$

$$\alpha_{A4}(D_1) = 0.3; \quad \alpha_{A4}(D_2) = 0.3$$

R 基于  $A5 = \{a5\}$  进行划分得  $CA_5 = \{C_{51}, C_{52}, C_{53}\}$ , 其中

$$C_{51} = \{1,2,4,5\} \quad \text{Des}(C_{51}) = (a5 = \text{水泡音})$$

$$C_{52} = \{3,7,9\} \quad \text{Des}(C_{52}) = (a5 = \text{干鸣音})$$

$$C_{53} = \{6,8,10\} \quad \text{Des}(C_{53}) = (a5 = \text{正常})$$

$$\alpha_{A5}(D_1) = 0.7; \alpha_{A5}(D_2) = 0.7$$

4) 因为在  $\alpha_{Ai}(D_j)$  中  $\alpha_{A5}(D_1) = 0.7$  为最大且  $\text{flag}(5) = n$ , 所以选择字段  $A5$ , 设置  $\text{flag}(5) = y$ ;  $A = A \cup A5 = \{a5\}$ . 并根据 R 基于  $A5$  的划分  $CA_5 = \{C_{51}, C_{52}, C_{53}\}$  进行分类;

因为  $C_{51} \subseteq D_1$  且  $C_{51} \cap R' \neq \emptyset$  所以得确定性规则:  
 $r1: \text{Des}(C_{51}) \rightarrow (\text{Des}(D_1))$  即  $(a5 = \text{水泡音}) \rightarrow (a6 = \text{肺炎})$  (cf=1.0)

$$R' = R' - C_{51} = \{3,6,7,8,9,10\}$$

又因为  $C_{53} \subseteq D_2$  且  $C_{53} \cap R' \neq \emptyset$  所以得确定性规则:

$r2: \text{Des}(C_{53}) \rightarrow (\text{Des}(D_2))$  即  $(a5 = \text{正常}) \rightarrow (a6 = \text{肺结核})$  (cf=1.0)

$$R' = R' - C_{53} = \{3,7,9\}$$

因为  $R' \neq \emptyset$  对 R 基于  $A \{a5\}$  进行划分得  $C = \{C_1, C_2, C_3\}$ , 显然  $C = CA_5$

因为  $C_1 \cap R' = \emptyset, C_2 \cap R' = \emptyset, C_3 \cap R' = \emptyset$ , 所以没有规则生成。

5) 因为  $R' \neq \emptyset$  且  $\alpha_{A5}(D_1) < cp$ , 继续选择  $\alpha_{Ai}(D_j)$  值为最大且  $\text{flag}(i) = n$  的字段  $Ai$ . 字段  $A4$  满足该条件, 选择字段  $A4$ , 设置  $\text{flag}(4) = y$ ,  $A = A \cup A4 = \{a4, a5\}$ . 并根据 R 其于  $A4$  的划分  $CA_4 = \{C_{41}, C_{42}\}$  继续分类:

因为  $C_{42} \subseteq D_2$  且  $C_{42} \cap R' \neq \emptyset$  所以得到确定性规则:

$r2: \text{Des}(C_{42}) \rightarrow (\text{Des}(D_2))$  即  $(a4 = \text{快}) \rightarrow (a6 = \text{肺结核})$  (cf=1.0)

$$R' = R' - C_{42} = \{3\}$$

因为  $R' \neq \emptyset$  对 R 基于  $A \{a4, a5\}$  进行划分得  $C = \{C_1, C_2, C_3, C_4, C_5\}$ , 其中

$$C_1 = \{1,2,4,5\} \quad \text{Des}(C_1) = (a4 = \text{正常} \wedge a5 = \text{水泡音})$$

$$C_2 = \{3\} \quad \text{Des}(C_2) = (a4 = \text{正常} \wedge a5 = \text{干鸣音})$$

$$C_3 = \{6,8\} \quad \text{Des}(C_3) = (a4 = \text{正常} \wedge a5 = \text{正常})$$

$$C_4 = \{7,9\} \quad \text{Des}(C_4) = (a4 = \text{快} \wedge a5 = \text{干鸣音})$$

$$C_5 = \{10\} \quad \text{Des}(C_5) = (a4 = \text{快} \wedge a5 = \text{正常})$$

$$\text{计算得 } \alpha_A(D_1) = 1.0; \alpha_A(D_2) = 1.0$$

因为  $C_1 \cap R' = \emptyset, C_3 \cap R' = \emptyset, C_4 \cap R' = \emptyset, C_5 \cap R' = \emptyset$ , 所以没有规则生成。

因为  $C_2 \subseteq D_1$  且  $C_2 \cap R' \neq \emptyset$  所以得确定性规则:

$r1: \text{Des}(C_2) \rightarrow \text{Des}(D_1)$  即  $(a4 = \text{正常} \wedge a5 = \text{干鸣音}) \rightarrow (a6 = \text{肺炎})$  (cf=1.0)

$$R' = R' - C_2 = \{\}$$

6) 因为  $\alpha_A(D_1) > cp$  且  $R' = \emptyset$ , 汇总生成的规则:  
 $r1: (a5 = \text{水泡音}) \vee (a4 = \text{正常} \wedge a5 = \text{干鸣音}) \rightarrow (a6 = \text{肺炎})$  (cf=1.0)

$r2: (a5 = \text{正常}) \vee (a4 = \text{快}) \rightarrow (a6 = \text{肺结核})$  (cf=1.0)

其中“血沉”对应字段  $a4$ , “听诊”对应诊断  $a5$ , “诊断结果”对应字段  $a6$

7) 程序结束。

## 参考文献

- 1 R. Yasdi, Learning Classification Rules from Database in the Context of Knowledge Acquisition and Representation, IEEE T-KDE VOL. 3, No. 3, Sept 1991
- 2 C. J. Matheus, P. K. Chan and G. P. Shapiro, Systems for Knowledge Discovery in Databases, IEEE T-KDE VOL. 5, No. 7, Dec 1993
- 3 J. W. Han, Y. D. Cai and N. Cercone, Data-Driven Discovery of Quantitative Rules in Relational Databases, IEEE T-KDE VOL. 5, No. 1, Feb 1993

## 征稿启事

### 一、征稿内容:

本刊目前特别欢迎作者撰写投寄以下内容的稿件:

1、具有预见性、策略性、统计性、分析性、整合性、归纳性、技巧性、实用性、新颖性、爆炸性、趣味性、工具性、资料性的综述评论稿;

2、看了即懂、拿来可用、图文并茂、见解深刻、层次分明(含故障现象、故障分析排除等)、新颖实用的维护维修稿;

3、目前计算机应用的新领域、新技术、难点、热点、空白点。

### 二、征稿要求:

稿件宜用激光照排正规打印,标明字数、页码、图号、图位、文种、作者详细地址、电话号码、邮编及英文标题、参考文献;若无条件打印,请正规誊抄于方格稿纸上,图纸要正规描绘,由于人手所限,恕不退稿,请作者自留底稿。来稿要求新颖实用,一稿一投。对优秀来稿,经审核后,本社将择优刊用,稿酬从丰。

热忱欢迎广大作者踊跃赐稿!

《计算机应用研究》杂志社



# 一种基于推理的自动生成施工网络图模型

曹桂琴

(大连理工大学计算机系 116023)

郭 晖

(大连机车车辆厂 116021)

**摘 要** 本文提出一种自动生成施工网络图的模型,采用这种模型开发的系统既适用于制订施工计划,又便于动态地调整施工进度,可提高施工计划现代化管理水平。

**关键词** 人工智能 计算机应用 程序系统 网络计划

## 1 前言

在现今的建筑工程项目管理中,计划管理是非常重要的一个组成部分。计划管理是以系统工程学中的网络技术为理论基础,主要采用关键路线法完成对施工计划的编制,协调各工序间的关系,从而实现对工期最优化控制。其中重要工作之一是绘制网络图,当前较为流行的是时标网络图。示例见图 1。

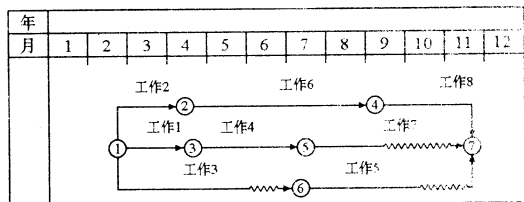


图 1 时标网络图示例

随着建筑工程规模的扩大,手工绘制施工网络图的难度也随之增加。为了更科学、系统地进行施工管理,在微机上通常采用流行的 FOXPRO 关系型数据库管理系统为主要工具进行数据统计、管理。但由于 FOXPRO 绘图功能不强,若与其它语言混合编程又增加了程序的复杂性,所以绘图成为 FOXPRO 应用系统的一大难题,但最近 UC DOS 系列新开发出 UC DOS SDK FOR FOXPRO,成功地解决了这一难题,提供了一整套完全与设备无关的、功能强大的图形、图像、汉字显示、打印控制函数及一个基本的图形用户界面 (GUI) 函数库。用其与 FOXPRO 配合使用可把计划管理与生成网络图有机地结合起来。正是基于这个出发点,我们采用 FOXPRO 2.5 和 UC DOS SDK FOR FOXPRO 为主要工具,在微机研制了一个简化的“时标施工网络图自动生成系统”,若要实现一个实用的系统还有待进一步细化。

## 2 数据库设计

为了便于网络图的自动生成,数据库中除保留有关的施工信息外,还需保存与布局相关的节点位置信息。每一个施工单位的所有工程项目的信息存于工程参数库中,对于每一个工程项目均有各自的节点信息

库和工作信息库。各库的关系模式描述如下:工程参数(工程编号,工程名称,开工日期,单位时间,时标单位,工期,总图数,...);工作信息(始节点序号,终节点序号,工作名称,作业时间,最早开工时间,最晚开工时间,...);节点信息(节点序号,节点入度,节点出度,最早发生时间,最晚发生时间,拓扑序列号,屏幕上的层号,图的编号,...)。

当计算关键路径和推算节点布局时均需要将图中节点排成一个线性序列( $V_1, V_2, \dots, V_n$ ),该序列需满足如下条件:若从节点  $V_i$  到顶点  $V_j$  有一条路径,则序列中节点  $V_i$  必在节点  $V_j$  之前,称满足这个条件的线性序列为拓扑序列,顶点  $V_i$  在拓扑序列中顺序号称为拓扑序列号。

### 知识的表示

施工时标网络图的绘制必须遵循中华人民共和国行业标准中的“工程网络计划技术规程”。例如:时标网络计划中所有符号在时间坐标上的水平位置及水平投影,都必须与其所代表的工作时间值相对应,节点的中心必须对准时标的刻度线。在屏幕上显示网络图,首先必须确定图中节点的位置,根据网络图的复杂程度可把屏幕划分成若干层,节点的行号=节点所在层号;节点的列号=节点事件的最晚发生时间/时间单位。(显示时再转换成像素点坐标)。确定节点位置的规则称为布点规则,主要布点规则如下:

规则 1:如果 节点的出度=1,则直接后继节点的行号=本节点的行号;

规则 2:如果 节点的出度= $n(n>1)$ ,则顺次将其  $n$  直接后继节点分布在从当前行开始的下  $n-1$  行上;

.....

确定箭线走向的规则称为布线规则,主要布线规则如下:

规则 1:如果(节点的出度=1)且(直接后继节点的行号=本节点行号),则箭线水平走向;

规则 2:如果 作业时间=0,则箭线垂直走向;

规则 3:如果 直接后继节点的列号 $\neq$ 本节点列

点,则用户可从三种箭线走向“!”、“?”、“其它”中任选其一;

规则4:如果某些工作长度不足到终节点,则用波形线补足,无波形线的工作就是关键工作,所有关键工作的集合组成关键路径;

.....

本系统采用框架和过程混合知识表示法描述各种规则和知识,把框架和推理结果均存于数据库中。系统采用正向推理方法,从开工节点出发,按拓扑序列,根据布点规则逐个推算每一个节点的直接后继节点的位置;再根据布线规则,逐条显示以每一个节点为始节点的所有工作。

#### 4 系统的实现

本系统的主要功能模块包含录入工程参数、录入工作信息、调整工作信息、添加节点信息、计算关键路径、整体布局、单图布局、调整布局、放大打印屏幕等,时标网络图的自动生成框图见图2。

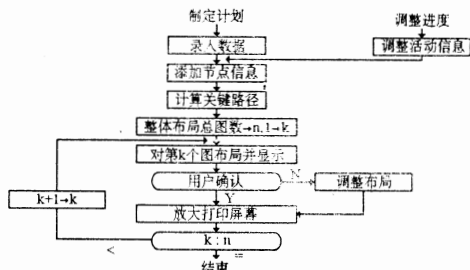


图2 自动生成时标网络图的框图

时标网络图的生成过程按功能可划分为三个阶段,各阶段的功能简述如下:

##### 4.1 数据准备

根据用户需要制订计划或调整进度计划,确定录入工程参数、工作信息还是调整工作信息,然后再向节点库文件添加节点信息。添加的方法是:顺次读工作文件中每一个记录,在节点库中查找与当前工作相关联的节点,添加不存在的节点信息,同时累加该工作始节点的出度和终节点的入度,供拓扑排序所用。我们采用拓扑排序方法计算关键路径,用户可不必考虑终节点序号必须大于始节点序号的问题。至此生成网络图的数据准备工作均已完成。如果将本系统嵌入已有的由FOXPRO编写的“工程计划管理系统”中,则上述这些数据可以共享,数据准备阶段的工作可以简化。

##### 4.2 整体布局

通常,如果工程规模庞大,难于仅在一屏上显示整个网络图,就需要确定应划分的子图个数(n)和每一个节点 $V_i$ 所属的子图号(k)。令t为工期,d为屏幕上显示单位时间的宽度(象素点数),maxx为当前屏幕的最大x坐标, $V1$ 为 $V_i$ 节点事件的最晚发生时间,按下列公式计算n和k:

$$n = \lceil (t * d + b * k) / maxx \rceil$$

$$k = \lceil (V1 * d + b) / maxx \rceil$$

另外每张子图可留有足够的边界,还应考虑子图间的连接问题。

##### 4.3 单图布局

本系统采用正向推理的方法,按拓扑序列,根据布点规则逐个推算每一个节点的直接后继节点的位置并显示。若用户不满意,则调整节点的布局直到满意为止。再根据布线规则,按拓扑序列显示每一个节点为始节点的所有工作。在FOXPRO程序中使用SDK FOR FOXPRO函数实现显示或修改图形的主要几个程序段如下:

###### 4.3.1 调整节点位置

当用户对节点的布局不满意时,就输入节点顺序号及调整后的层号。程序中先用底色刷新原来的节点,然后在新位置上重新显示节点。

```
do loadsdk
```

```
* pmcs 表示屏幕层数;dd 表示节点序号;sh 表示行位;
```

```
sl 表示列位。
```

```
do while .t.
```

```
cs==0
```

```
@23,3say"输入节点:"get dd
```

```
@23,24say"层:"get cs
```

```
read
```

```
@23,0
```

```
if dd=0
```

```
exit
```

```
endi
```

```
sele 3 && 节点信息文件
```

```
go top
```

```
loca for ddhx=dd
```

```
sl=40+zwkg*45
```

```
sh=150+pmcs*50
```

```
* 刷新原节点
```

```
=setcolor(1)
```

```
=setfillstyle(1,1)
```

```
=pie(sl,sh+8,0,361,10,10)
```

```
=sethcolor(1)
```

```
=showhz(sl-10,sh,str(ddhx,2,0))
```

```
* 显示新节点
```

```
repl pmcs with cs
```

```
sh=150+cs*50
```

```
=setcolor(15)
```

```
=sethcolor(15)
```

```
=showhz(sl-10,sh,str(ddhx,2,0))
```

```
=circle(sl+5,sh+8,10)
```

```
endd
```

###### 4.3.2 显示工作

\* 始节点位置为(ls1,cs1),终节点位置为(ls2,cs2);hdm为工作名称

\* 满足布线规则1时,则画水平箭线

```

=drawline(1s1,cs1,1s2,cs1)
=showhz(1s+6,cs-25,hdm)
=showhz(1s2-10,cs1-8,"→")
* 满足布线规则 2,略
* 满足布线规则 3;zd 为终点节点序号
ch=1
@22,2 say"选择"
@22,10 say dd pict"99"
@22,18 say"到"
@22,26 say zd pict"99"
@22,30 say"的画线方式:"
@23,10 prom"→"
@23,15 prom"↖"
@23,40 prom"其它"
menu to ch
do case
case ch=1
1s2=40+45 * pmls-10
=drawline(1s1-11,cs1+10,1s1-11,cs2)
=drawline(1s1-11,cs2,1s2,cs2)
=showhz(1s1+6,cs2-25,hdm)
=showhz(1s2-10,cs1-8,"→")

```

```

case ch=2
1s2=40+45 * pmls-10
=drawline(1s1-11,cs1-10,1s1-11,cs2)
=drawline(1s1-11,cs2,1s2,cs2)
=showhz(1s1+6,cs2-25,hdm)
=showhz(1s2-10,cs1-8,"→")
case ch=3
q=40+45 * pmls+4
p=150+50 * pmcs
=drawline(1s1,cs1,q,cs1)
=showhz(q-12,cs1-8,"→")
=showhz(1s1+15,cs1-25,hdm)
if cs1<p
=drawline(q,cs1,q,p-2)
else
=drawline(q,cs1,q,p+18)
endi
endc
最后用 prnscr() 函数可以放大打印整个屏幕。

```

### 参考文献

- 1 欧震修等. 建筑工程施工监理手册. 北京: 中国建筑工业出版社, 1995. 299~302

## 铁路自动化系统工程中数据通信的研究

陈大为

(北方交通大学自动化系统研究所 北京 100044)

韩 艺

(北方交通大学应用系统分析研究所 北京 100044)

**摘 要** 本文主要从现场数据通信和分布式网络的角度对铁路自动化系统工程中的数据传输和数据通信的体制进行一些探讨, 提出铁路现场网的概念, 并对现场网的发展方向进行一些有益的探索。

**关键词** 铁路自动化系统工程 数据通信 铁路现场网

铁路自动化系统工程是近些年来提出的将系统工程思想和铁路自动化工程结合起来的一种新思路。它将整体的铁路自动化工程分解为互相关联的许多子系统, 既便于从全局方面来调配和控制工程的发展, 又可以及时地将各个子系统中当前最新的技术成果应用到实际的工程中。现主要对数据传输子系统(包括 TMIS、CTC 和铁路现场测控网)中的铁路现场测控网进行一些必要的探讨。

### 1 铁路现场网的定义

改革开放以后, 我国铁路现代化工作进展迅速, 一大批使用智能化的现场设备和控制计算机的新的监测

和控制的自动化项目应运而生。由于多个现场设备分布于不同的地域, 形成了多节点的数字数据传输网络。相对于工业中的现场网, 我们将这种铁路中用于现场设备之间测量和控制信号数据传输和通信的网络定义为铁路现场测控网, 简称现场网。这种底层的现场网与铁路其它网络的关系如图 1 所示。

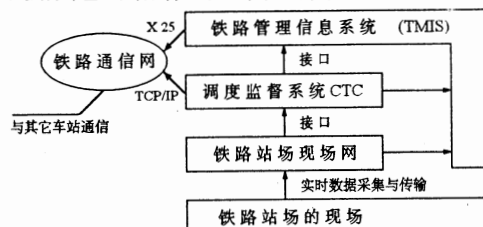


图1 铁路各层信息网络的模型

从理论上说来,铁路现场网类似于局域网 LAN,但是在铁路的特殊条件下,铁路现场网又有现场节点位置分布较远;网络拓扑结构简单;传输的报文短;报文实时性要求高;可靠性要求高;抗恶劣环境能力强等特点。因此铁路现场网不能照搬现有的 LAN 的模式,应根据铁路的实际情况因地制宜地发展具有铁路特色的现场测控网络。

## 2 铁路现场网的研究

### 1) 拓扑结构和传输介质的选择

#### (1) 拓扑结构

在 LAN 应用中常见的网络拓扑结构主要有星型、环型和总线型那么几种。各种结构都有其优缺点。星型结构较为简单,成本低,线路故障只影响对应的节点,便于查错和维护,网络的容量和可扩展性以及可靠性由中心节点的性能决定。环形网络的特点是出错率低,网络总长度比其他形式的结构要长,但其链路容易损坏,任何节点的故障可能使整个网络瘫痪。总线型网络由于无中心节点,因此节点故障不会影响整个网络,通过广播式发送信息性能较好,网络的扩展型较好。

在铁路现场网的设计中,一般将几种拓扑结构综合起来应用:在设备比较密集的现场采用星型拓扑结构,以节省费用;整个网络采用总线型结构,以灵活地适应各种现场条件。

#### (2) 传输介质的选择

在铁路自动化系统中采用的现场数据通信网络,必须要保证的是可靠性和实时性,因为这直接影响到行车和站场作业的安全,关系重大。这就要求现场网络首先要能够抵抗现场恶劣的环境的干扰,降低数据传输过程中的误码率。

在电磁干扰严重的区段特别是电气化铁路区段中,普通传输线上感应的共模干扰电压高达,从十几伏到 1~2 百伏,这使得某些长线驱动器不能正常工作。通信明线由于其抗干扰性较弱、易受天气影响,通信质量不稳定,且衰减大、容量小,因而不适合作为现场网的主干介质。屏蔽双绞线 STP 带宽小,数据传输率低,传输距离近,成本低,比较适合应用在某些采用星型结构的网段。在总线的选择上主要是同轴电缆和光纤的比较。从目前铁路现场的条件来说,采用同轴电缆可以利用现成设备,技术上也比较成熟,因此可以节约投资,较快地投入使用。

随着光纤技术的发展(包括在商用网络中以光纤为媒质的数据传输技术,如 FDDI 等的日臻成熟和完善,并且考虑到铁路现场设备的智能化和数字化的发展,笔者认为进行铁路现场中以光纤为传输媒质的测控数据传输的实验和可行性研究是十分有必要的。

### 2) 链路层通信协议

现场网络的通信协议不但要简单实用,而且严密可靠,保证数据传输的正确性。这些通信协议一般是人

为制定的,根据需要选择协议算法。在某些情况下,面向字符的链路控制规程仍然在实际的一些通信量不太大的数据传输中应用。它的特点是算法和报文格式比较简单,但是通信线路的利用率低,只适用于半双工的工作状态,只能对数据部分进行差错控制,而控制字段出错就无法控制。现在数据通信中一般采用面向比特的链路控制规程如 HDLC 等,它比面向字符的协议有更高的可靠性和更高传输效率。它的特点有二:一是使用固定“封装”式的帧结构和 0 插入、检出技术来传送数据,实现数据的透明传输;二是传输方式是面向位的,即任意比特的任何序列都可以帧的形式传输。

### 3) 数据传输技术

铁路上采取的数据传输方式可根据数据信号的特点分为 3 种方式即:基带信号传输、频带信号传输和数字数据传输。

(1) 基带传输是最基本的数据传输技术,基带信息在介质上被直接编码和传输。它的特点是一个时刻只能传输一个信号,简单可靠,成本相对较低。基带信号直接传输的距离和数据传输率很有限,但采用基带数传机可以将传输距离拓展到几公里到几十公里,采用电流环和长线驱动器也可以低速传输的条件下传输几公里而且抗干扰性较好。基带传输在现场的应用最多,应用也较为灵活和方便。

(2) 频带传输就是将代表数据的二进制“0”和“1”信号,通过 Modem 变换成具有一定频带范围(比如 300Hz~3400Hz 的话音通路频带)的模拟数据信号,使数据可以在普通电话线路上传输,传输距离可以从几公里到几十公里不等。但是通过话路 Modem 数据传输的速率依然不高,例如 V.32 协议的 Modem 传输速率为 9.6kb/s。进行高速数据传输可以使用宽带 Modem,但其价格要比普通的高出数倍。

(3) 数字数据传输方式是利用数字话路传输数据信号的一种方式。它是基于多路复用技术(频分复用 FDM 或时分复用 TDM)的多信道传输方式。当前公用网以及铁路通信网上 PCM 数字电话发展较快。它每一路可以传输的数据速率为 64kb/s,还可加装数据时分复用器,将多路低、中速的数据信号经时分复用合成后进入 64kb/s 的数字话路合成。在京九线的 CTC 调度监督网中就使用了这种数据传输方式,但在现场网中一般不采用,因为这种方式的硬件投入较大,设备较为复杂,适合于广域网通信。

### 4) 通信协议标准的比较

在当今国际上比较流行的工业网络标准有 IEEE802、TCP/IP、Fieldbus 等。与工业自动化网络类似,铁路现场网中也可以考虑采取这些标准。需要指出的是,普通 LAN 采用的 IEEE802 规程并不能完全满足铁路现场网的要求。

#### (1) 介质访问控制 MAC

IEEE802.3 的 CSMA/CD 方法的缺陷是①在重载时冲突频频发生,站点对网络的访问延迟成为不确定,不利于铁路现场网对实时性要求高的特点;②冲突检测要求在电缆上任一点上任何两个站的信号强度的差别足够小,并且辨别是干扰噪声还是其他站的信号较困难,在铁路现场强干扰的环境中适用性较差;③对分组的长度有下限要求,铁路现场网中普遍的交互式的短分组的通信就会浪费信道的带宽。IEEE802.4 令牌总线 和 IEEE802.5 令牌环方式虽然可以解决介质访问时间不确定的问题,在高负荷情况下能实现高吞吐量,但这种方法控制复杂、开销大,难以完全做到固定周期访问,所以也不适应铁路现场测控的蛛网要求。

#### (2) 数据链路服务和应用层服务

IEEE802 委员会规定 3 种链路服务:LLC1(无确认的连接服务)、LLC2(确认的连接服务)、LLC3(确认的无论连接服务)。这 3 种 LLC 服务均不向用户提供周期服务,而周期服务是铁路现场级设备必不可少的通信需求。

在应用层服务方面目前提供的 LAN 应用服务是面向计算机系统的,而不是铁路的各种现场设备(如传感器、调节器等)。

虽然 LAN 为现场设备提供了组网的灵活性和可扩充性,但是作为铁路现场网必须要满足现场设备通信业务的可确定性和周期性,通常概念的 LAN 是不会突出对这种通信需求的处理。因此研究适应铁路现场设备联网的类似于工业中 fieldsbus 体系结构是十分具有应用前景的。

### 3 展望

网络中数据传输和数据通信技术是实现铁路自动化的一个关键技术,它为铁路现场网提供采集信息和控制信息交换的通路,将直接影响到铁路各种自动化系统运行效率和安全。从我国铁路自动化目前的状况来看,现场设备测控信号的数据通信网络还很简陋,还有许多值得探讨和改进的地方。在很多方面可以参考和借鉴目前成熟的工业 CAD/CIMS 技术和办公室自动化高级分布式系统技术,为网络技术在铁路现场中应用做好必要的技术储备。现在主要的研究方向有:

(1) 研究各种现场总线在铁路应用的可能性;

(2) 研究适应铁路现场的网络体系结构;

(3) 加快铁路现场网络的传输线路新技术的应用(如光纤,无线等);

(4) 研究开发各种适用于铁路现场的网管软件。

铁路自动化领域是网络应用的新领域,我们应该及时地对计算机网络通信的各种技术作深入地了解和研究,充分地考虑每一项技术的特征,发展我国铁路专用的现场网络体系,为实现铁路自动化打好坚实的基础。

#### 参考文献

- 1 谢希仁. 计算机网络. 大连理工大学出版社
- 2 王常力. 工业控制计算机系统——设计和应用. 电子工业出版社
- 3 汪润生,周师熊. 数据通信工程. 人民邮电出版社

## 基于 FPGA 的主机与多单片机快速通讯系统的设计

党 岗 胡华平

(国防科技大学计算机系 长沙 410073)

**摘 要** 本文设计了一个基于 FPGA 的主机与多单片机快速通讯的系统,它将多片 MCS—51 系列单片机通过全互连的拓扑结构连接起来,以 PC 486/586 系列微机为主机,构成一个主从分布式控制的多机系统。其中硬件设计在 Xilinx 公司的 CAD 软件上进行了模拟验证并得到了 FPGA 实现,实验表明这种方法可以提高系统的处理效率,提高实时性,对个子系统的编程及调试带来了很大方便。

**关键词** 主机 单片机 互连网络 现场可编程门阵列(FPGA)

### 1 引言

近年来,以 PC 486/586 为代表的各种档次的个人计算机,由于它们优越的性能价格比和丰富的软件资源,已成为计算机应用的主流机种。而 MCS—51 系列单片机在工业控制系统中也越来越得到广泛应用,它以使用方便、价格低、功能强、体积小、抗干扰能力强等特点已渗透到各个领域。特别是在军事工业、航空、航

天、家电、汽车电子、通讯、交通、智能仪表和智能卡等领域都具有极广阔的应用前景。如在工程应用中,常遇到若干任务需要同时执行的情况。因而分布式控制成为一种常见的控制方式,而要实现分布式控制,多采用主从式微机网络形式。采用我国应用最广泛的微机作为主控制单元,以及性能价格比高的 MCS—51 单片机作为从控制单元,从各方面讲无疑都具有很好的优越性。

现场可编程门阵列(FPGA, Field Programmable



Gate Array)。它是标准的大规模集成电路产品,可用于各种数字逻辑系统的设计。FPGA 近年来得到了迅猛发展,它不仅具有传统门阵列的所有特点,而且不需要初始投资,不需提前制造,无库存风险,模拟工作费时较少。最重要的一点是FPGA的“现场可编程性”缩短了设计周期,并且可以立即提供样片和生产的芯片,故许多设计者将其作为开发样机来使用。实际上,研制一片FPGA,比在印刷电路板上使用逻辑器件设计一个等效的系统更为容易和迅速。

## 2 系统设计的总体方案

本系统是以个人计算机为主机,多个单片机为辅助组成的主从分布式系统。其设计总体方案如图1所示。从该图中可以看出主机直接与单片机建立通讯联系,而单片机之间通过互连网络连接起来。为了实现主机与单片机之间准确可靠的交互,系统在主机与单片机之间设置一个缓冲器。同时为了实现数据的快速传送,依据单片机在一般应用中有扩展外存的特点,源单片机对目的单片机内存的访问,可直接简化为通过互连网络对目的单片机扩展外存的访问,而扩展外存同时也起缓冲器的作用。主机和单片机均可对外存的访问,从而保证了主机与多单片机之间准确而快速的交互。并且,互连网络成为单片机与外存之间联系的媒介,互连网络拓扑结构的选择将直接关系到通讯的效率。为了不失一般性,在本设计中,互连网络采用完全互连的 Crossbar 的拓扑结构,即每个结点单片机之间都直接互连进行访问,其示意图见图2<sup>[1]</sup>。

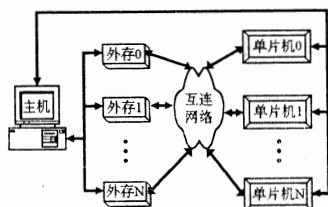


图1 总体方案结构

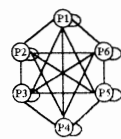


图2 完全连接的互连网络

个人计算机与多单片机之间的通讯包括主机与单片机通讯、单片机与单片机间通讯两大部分。通讯方式有串行通讯和并行通讯两种。串行通讯适合较远距离的通讯,并且需要相应的硬件支持,因其速度慢,易出错,故不利于数据的大量传输;并行通讯适合于近距离的通讯,传输速度快,适合数据的大量传输的情况。本文采用并行通讯的方式,主机选用PC 486/586系列微机,它通过ISA总线对MCS-51系列单片机中多个8051单片机进行控制<sup>[2]</sup>。8051单片机的扩展外存采用2K字节的静态RAM6116,因为本设计的主要目的在于样机调试,故本文选用仅只有2K字节存储空间RAM6116,它足以满足较小规模的应用要求。另外,根据不同的设计需求,从单片机可以设置为几个,几十个甚至更多,在以下的讨论中,为了方便,本文仅考虑四个单片机的情况。

## 3 主机与多单片机并行通讯的软硬件设计

### 1) 硬件结构及功能

主机与单片机通讯的硬件结构如图3所示。主机与多单片机并行通讯的硬件结构主要由以下几部分组成<sup>[3,4]</sup>。

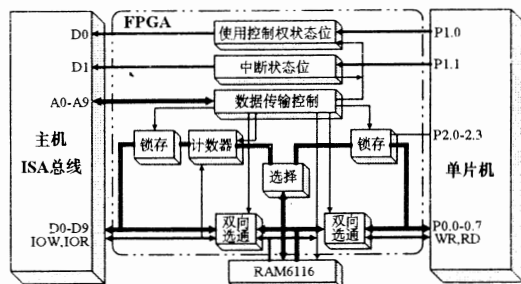


图3 主机与单片机通讯的硬件结构

- 时钟控制:由于微机与单片机晶振频率不同,在主机与单片机通讯时,为了保证数据传送不出错,系统设置了来自微机和单片机两个时钟来源,并且在主机向单片机发送数据的过程中设立了缓冲装置,而每个单片机的扩展外存恰好起到缓冲作用。

- 数据传送控制:在ISA总线中系统的I/O端口均使用低10位地址(A0~A9),对传送的控制也是通过I/O端口传送实现。本系统选用220~22f端口,其中一部分端口用于控制线路。

- 地址计数器:主机访问6116的高位地址和低位地址以及数据都是从D0~D7分时发送的。考虑到速度上的要求,主机对6116的访问设置地址计数器,每读写一个数据,地址自动加一,其加一动作由IOW或IOR信号的上升沿触发,当下一次通过端口送地址时,计数器重新置初始地址。

- 使用权状态标志位:由于6116跨在主机与单片机上,它可被主机和单片机两个来源访问,因而存在着控制使用权问题。为解决这一矛盾,系统设置了一个D触发器标志状态来协调对RAM6116的控制使用权。

- 中断标志位:当某一8051需要将运行结果送给主机时,首先发出中断信号。这一中断信号通过其相应的D触发器保存并形成标志位。主机收到中断信号后,通过软件方法就可查询到是哪个单片机欲向主机发送数据。

此外,当主机向单片机传送完数据后,通过端口发送“发送完”信号,单片机接收到这一信号后,将忙闲状态置“忙”,并开始单片机之间的通讯。

如图所示,这几部分均设计在于FPGA芯片中。

### 2) 软件设计组成及流程

主机与单片机通讯的软件由以下三部分组成:

- 初始化程序,其流程见图4;
- 主机向单片机发送程序,其流程见图5;
- 单片机向主机发送的中断处理程序,其流程见图6。

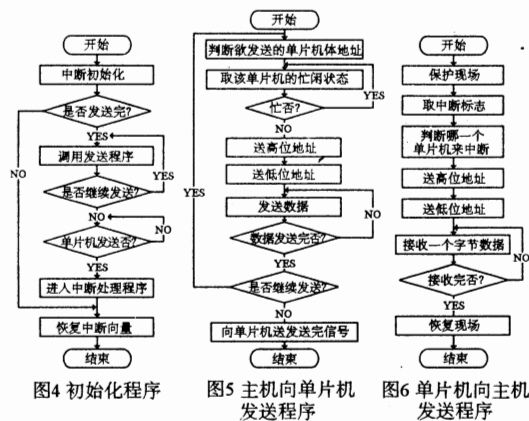


图4 初始化程序

图5 主机向单片机

发送程序

图6 单片机向主机

发送程序

#### 4 结束语

本文介绍了一种基于 FPGA 的主机与多单片机快速通讯系统的设计,其硬件逻辑线路均在 Xilinx

XACT 的 CAD 软件上进行了模拟验证并得到了 FPGA 实现。实验表明这种方法可以提高系统的处理效率,提高实时性,对个子系统的编程及调试带来了很大方便,并进一步提高了系统的可靠性、灵活性及可扩展性,它可广泛应用于单片机间大量传送数据和工业控制领域中大规模并行处理的场合。

#### 参考文献

- 1 Kai Huang, Faye A. Briggs, Computer Architecture And Parallel Processing. McGraw—HM Company. 1984.
- 2 张载鸿. 微型机(PC)系列接口控制教程. 清华大学出版社, 1992.
- 3 何立民. 单片机应用文集(2), 北京航空航天大学出版社, 1993.
- 4 陈光东, 赵性初. 单片微型计算机原理与接口技术. 华中理工大学出版社, 1993.

## Ramsey 数 $R(3, 3, 3, 3, 3; 2)$ 的下界

张正铀 (广西科委 南宁 530012)

苏文龙 (广西梧州一中 梧州 543002)

罗海鹏 (广西科学院 南宁 530031)

吴康 (华南师范大学 广州 510631)

**摘要** 本文用群论和数论的方法研究了素数阶循环图的一些性质, 得到 Ramsey 数  $R(3, 3, 3, 3, 3; 2)$  的新的下界。

**关键词** Ramsey 数 下界 循环图

#### 1 引言

1930 年, 英国科学家 F. P. Ramsey 发现了一个定理<sup>[1]</sup>, 揭示了“不可能有完全的无序”这深刻的自然规律而引起人们浓厚的兴趣, 发展成为图论、组合数学和离散数学的核心内容——Ramsey 理论, 在现代数学中产生了深远的影响, 并在计算机科学的信息检索、噪声通信和决策科学等方面有着广泛的应用。

确定 Ramsey 数是 Ramsey 理论的核心问题, 这是世界著名的数学难题。一般方法是分别确定每个 Ramsey 数的上、下界, 然后逐步逼近它的准确值, 这对于无穷多个 Ramsey 数的计算是非常困难的事情, 因此数学界公认 Ramsey 数研究是可以比拟著名的哥德巴赫猜想的跨世纪的数学难题。经过各国数学家六十多年的努力探索, 迄今为止算出来的 Ramsey 数仅有少数的几个<sup>[2,3]</sup>, 即使寻求 Ramsey 数的较好的下界也极其困难。1955 年 Greenwood 与 Gleason 得到较好的下界估计式<sup>[4]</sup>:

$$R(3, 3, \dots, 3; 2) \geq 2^n + 1$$

n 个 3

60 年代中期以后, 由于电子计算机技术的不断发展, 文献[5—9]研究了阶数较小的循环图, 在电子计算机上得到一些 Ramsey 数较好的下界。但由于这些文献没有充分挖掘循环图的更多的性质, 因而运算效率不高, 当循环图的阶数较大时就要耗费太多计算机运算的时间, 即使近年来电子计算机的运算速度越来越高, 也远远赶不上当阶数增大时循环图的运算量的指数型增长的速度, 因此近年来关于 Ramsey 数研究的进展是非常缓慢的。有鉴于此, 本文运用群论和数论的方法研究素数阶循环图的一些性质, 提高了运算效率, 得到了一个优于上述 Greenwood 与 Gleason 的估计式的新成果:

定理 1.  $R(3, 3, 3, 3, 3; 2) \geq 102$ .

本文引用到的图论、数论和群论的术语见文献[2, 10, 11]。

#### 2 素数阶循环图的线性变换

给定整数  $n \geq 2$ , 素数  $p = 2mn + 1$ , 记  $Z_p = \{0, 1, \dots, 2mn\}$  表示模  $p$  的最小非负剩余系。约定, 以下所有小写英文字母都表示模  $p$  整数, 并且任何若干个整数的加、减、乘、乘方运算的结果(为了简便, 仍用普通等

号“=”写出)都应取模  $p$  同余归结到  $Z_p$ , 除非另外声明. 设  $g$  是  $p$  的原根, 记

$$\bar{Z}_p = \{x | x = g^j, 0 \leq j < 2mn\}.$$

$$\alpha_i = \{x | x = g^{nj+i}, 0 \leq j < 2m, 0 \leq i < n\}.$$

$$\alpha_i \alpha_j = \{x | x = ab, a \in \alpha_i, b \in \alpha_j\}.$$

熟知  $Z_p$  是有限域,  $\bar{Z}_p$  是在模  $p$  同余的乘法运算下的  $2mn$  阶循环群,  $\alpha_0$  是生成元为  $g^n$  的  $2m$  阶循环群, 它是  $\bar{Z}_p$  的正规子群,  $\alpha_i$  是  $\alpha_0$  的陪集,  $\bar{Z}_p/\alpha_0 = \{\alpha_0, \dots, \alpha_{n-1}\}$  是  $\bar{Z}_p$  的商群且:

引理 1.  $\alpha_i \alpha_j = \alpha_{i+j}$ , 其中  $\alpha_{i+j} = \alpha_r$ ,  $r = i+j \pmod{n}$  且  $0 \leq r < n$ . (约定, 以下关于  $\alpha_i$  的下标均仿此, 取模  $n$  同余归结到模  $n$  的最小非负剩余系  $\{0, 1, \dots, n-1\}$ ).

定义 1. 设  $G$  是  $p$  个顶点的完全图, 顶集  $V_G = Z_p$ , 边集  $E_G = \{\alpha_0, \dots, \alpha_{n-1}\}$  (即以命名为  $0, 1, \dots, n-1$  的几种颜色把各边着色); 当且仅当  $x-y \in \alpha_i$  时两个顶点  $x, y$  称为  $\alpha_i$  相邻的 (即顶点  $x, y$  的联边是着色  $i$  的). 象这样命名了各顶点并且规定了各边着色方法的完全图  $G$  称为  $p$  阶循环图.

定义 2. 给定两个  $p$  阶循环图  $G$  与  $G'$ , 如果  $V_G$  与  $V_{G'}$  之间存在 1—1 对应关系  $f$ , 并且  $f$  把图  $G$  中两个  $\alpha_i$  相邻的顶点  $x, y$  映射为图  $G'$  中  $\alpha_j$  相邻的两个顶点  $f(x), f(y)$ , 我们就称这两个图是同构的. 约定, 两个同构的  $p$  阶循环图 (其顶集均为  $Z_p$ ) 看作是同一个图, 上述同构映射  $f$  称为图  $G$  的变换.

引理 2. 设  $a \in \alpha_i, b \in Z_p$ , 则  $f(x) = ax + b (x \in Z_p)$  作成图  $G$  的变换 (线性变换), 它把各边都着色  $i$  的  $k$  阶完全子图变换成各边都着色  $i+j$  的  $k$  阶完全子图.

证明: 由  $a \in \alpha_i$  知  $a \neq 0$ , 对于任意  $x, y \in Z_p$  有  $x=y \Leftrightarrow a(x-y)=0 \Leftrightarrow f(x)=f(y)$ , 即  $f$  作成顶集  $V_G$  的 1—1 变换. 据引理 1 有  $x-y \in \alpha_i \Leftrightarrow f(x)-f(y)=a(x-y) \in \alpha_{i+j}$ , 即图  $G$  中  $\alpha_i$  相邻的两个顶点变换成  $\alpha_{i+j}$  相邻的其他两个顶点, 因此  $f$  把图  $G$  中各边都着色  $i$  的  $k$  阶完全子图变换成各边都着色  $i+j$  的  $k$  阶完全子图. 证毕.

### 3 定理 1 的证明

给定  $n=5$ , 素数  $p=101$ , 则  $g=2$  是  $p$  的最小正原根, 据  $\alpha_0$  的定义有:

$$\begin{aligned} \alpha_0 &= \{x | x \equiv 2^{5i} \pmod{101} \text{ 且 } 0 < x \leq 100, 0 \leq i < 20\} \\ &= \{1, 6, 10, 14, 17, 32, 36, 39, 41, 44, 57, 60, 62, \\ &65, 69, 84, 87, 91, 95, 100\} \end{aligned}$$

据此容易得到第一个结论: 对于任意  $a \in \alpha_0$  恒有  $a-1 \in \alpha_0$ .

据定义 1 我们构造了一个  $p=101$  阶循环图  $G$ . 假设在这图  $G$  中存在某个各边都着色  $i$  的 3 阶完全子图, 其中 3 个顶点为  $x_1, x_2, x_3$ . 据定义 1 知  $x_2-x_1 \in \alpha_i$ , 据引理 1 知  $(x_2-x_1)^{-1} \in \alpha_{-i}$ , 据引理 2 知变换:

$$f(x) = (x_2-x_1)^{-1} \cdot (x-x_1)$$

把上述 3 阶子图变换成一个各边都着色 0 的 3 阶完全

子图, 其顶点  $f(x_1)=0, f(x_2)=1, f(x_3)=a$ . 于是 3 个顶点  $0, 1, a$  中任意两个顶点都是  $\alpha_0$  相邻的, 就有  $a \in \alpha_0$  且  $a-1 \in \alpha_0$ , 这与上述第一个结论矛盾. 因此我们用反证法证明了: 图  $G$  中各边同色的任何 3 阶完全子图都不存在.

众所周知, 在图论中有著名的 Ramsey 定理. 对于任意  $n \geq 2$  个正整数  $k_1, k_2, \dots, k_n \geq 2$ , 存在最小正整数  $R$ , 当  $S \geq R$  时用  $n$  种颜色把  $S$  阶完全图  $G$  的边任意着色, 则一定存在各边同着第  $i$  种颜色的  $k_i$  阶完全子图. 这里  $i$  是  $1, 2, \dots, n$  中的某一个.

上述正整数  $R$  称为 Ramsey 数  $R(k_1, k_2, \dots, k_n; 2)$ . 由于我们已如上述构造了一个  $p=101$  阶循环图  $G$ , 其中各边同色的任何 3 阶完全子图都不存在, 因此据 Ramsey 定理可知  $R(3, 3, 3, 3, 2) \leq 101$  是不可能的, 这就证明了定理 1.

### 参考文献

- 1 F. P. Ramsey, On a Problem of formal Logic, Proc. London Math. Soc., 2nd Ser., 30(1930): 264—286
- 2 J. A. Bondy and U. S. R. Murdy, Graph Theory with applications, Macmillan, London, 1976
- 3 刘任任. 离散数学. 长沙: 湖南科学技术出版社, 1995: 114
- 4 R. E. Greenwood and A. M. Gleason, Combinatorial relations and chromatic graphs, canad. J. Math 7(1955): 1-7
- 5 J. P. Burling and S. W. Reyner, Some lower bounds of the Ramsey numbers  $n(k, k)$ , J. Comb. Theory(B) 13(1972): 168—169
- 6 F. Guldan and P. Tomasta, New lower bounds of some diagonal Ramsey numbers, J. Graph Theory Vol. 7 No. 1 (1983), 149—151
- 7 J. G. Kalbfleisch, Construction of special edge-chromatic graphs, Canad Math. Bull 8(1965): 575-584
- 8 J. E. Graver and J. Yackel, Some graph theoretic results associated with Ramseys theorem, J. Comb. Theory 4(1968): 125-175
- 9 R. G. Stanton and J. G. Kalbfleisch, Introduction to Ramsey theory, Lecture Notes in Math. 829(1979): 1-28
- 10 华罗庚. 数论导引. 北京: 科学出版社, 1957.
- 11 B. L. 范德瓦尔登(丁石孙等译). 代数学. 北京: 科学出版社, 1963.

### 欢迎订阅——软件报

《软件报》最大的特色是融“知识性、趣味性、实用性”为一体. 软硬兼施, 突出实用.

主要栏目有: 热点透视、热门软件、多媒体天地、网络技术、防清病毒、自动化应用、编程技巧、实用程序、家庭电脑、初学者园地、游戏玩家、维修经验、软硬件市场等.

如果你是新订户, 不妨索取二、三期试读, 相信你会爱不释手.

《软件报》为四开八版, 月末为四开十六版.

订阅代号: 61—74, 单价 0.38 元, 月价 2.93 元, 年价 24.32 元.

《软件报》社地址: 成都市金河街 75 号.

邮政编码: 610015 电话: (028) 6248449, 6637880—12

# $N \times 36$ 表示为两对孪生素数的和

尤之述 (湖南长沙交通学院 长沙 410076)

罗海鹏 黎贞崇 (广西科学院 南宁 530031)

唐贵松 (广西计算中心 南宁 530022)

**摘要** 本文给出 C 语言的程序,通过计算,否定了  $N \times 36$  可以表示为两对孪生素数和的猜测,即当  $N = 62$  时,不存在这样的表达式。但当  $N \neq 62$  且  $N \leq 80100$  时,表达式是存在的。

**关键词** 孪生素数和 计算机算法

## 1 引言

文献[1]中第 167 个问题的附加猜测是:任意自然数的 36 倍等于两对孪生素数的和。初始的情形如下:

$1 \times 36 = (5+7) + (11+13);$   
 $2 \times 36 = (5+7) + (29+31);$   
 $3 \times 36 = (11+13) + (41+43);$   
 $4 \times 36 = (29+31) + (41+43).$

随着  $N$  的增大,这个验证工作的计算量增大得很快,我们用 C 语言编了一个程序,来验证这个猜测。

## 2 程序清单

```
#include<stdio.h>
main()
{
    int k,i,j,is_prime;
    long n,p1,p2,p3,p4,s,s1;
    for(n=1;n<=99999;+=n)
    {
        k=0;
        s=n*36;
        for(p1=3;p1<=s/4;p1=p1+2)
        {
            p2=p1+2;
            is_prime=1;
            for(i=2;p2/i>=i;+=i)
                if(p1%i==0||p2%i==0)
                    is_prime=0;
            if(is_prime)
            {
                s1=s-(p1+p2);
                p3=s1/2-1;
                p4=p3+2;
                for(j=2;p4/j>=j;+=j)
                    if(p3%j==0||p4%j==0)
                        is_prime=0;
            }
            if(is_prime)
            {
                printf("%ld*36=(%ld+%ld)+(%ld+%ld)\n",n,p1,p2,p3,p4);
                ++k;
            }
        }
    }
```

```
}
if(k==0)
    printf("%ld*36 no solution\n",n);
printf("%d\n",k);
}
```

## 3 运算结果

### 3.1 上述猜测不成立

当  $N = 62$  时,  $N \times 36$  即  $62 \times 36$  不能表示为两对孪生素数的和。

### 3.2 $N \neq 62$ 时猜测是正确的

用上面的程序已算了  $1 \leq N \leq 80100$ ,除了 62 之外,当  $N$  为其他任何正整数时,均能给出  $N \times 36$  等于两对孪生素数的和。

### 3.3 例子

下面是手算比较难寻找的较大的孪生素数和的例子:

$3852 \times 36 = (10427+10429) + (58907+58909);$   
 $5563 \times 36 = (16901+16903) + (83231+83233);$   
 $44922 \times 36 = (20507+20509) + (788087+788089);$   
 $56593 \times 36 = (16691+16693) + (1001981+1001983).$

### 3.4 多个等式的例子

大多数的等式都不只 1 个,下面是有 24 个等式的例子:

$4567 \times 36 = (197+199) + (82007+82009) = (1427+1429) + (80777+80779) = \dots = (35897+35899) + (46307+46309)$  (共 24 个等式)

## 参考文献

- 1 杨 之,《初等数学研究的问题与课题》(湖南教育出版社,1993 年 5 月)P333.

## 启 事

本刊现有以下最新技术资料可供订阅和邮购:

- 1.《计算机应用研究》1996 年精装合订本,每套 40 元。
- 2.《计算机应用研究》1995 年学术论文专辑 1、2、3、4,约 159 万字,每套 48 元。

以上价格含邮费,数量有限,欲购从速。

通讯地址见版权栏 联系人:邓嘉澍

本刊编辑部

# ObjectWindows 与 Visual Basic 控件资源研究

薛少华 陈德新 鲁放

(郑州大学电子工程系 郑州 450052)

**摘要** 本文研究了 ObjectWindows 与 Visual Basic 控件资源的特点,在此基础上,重点研究了在 ObjectWindows 软件设计过程中使用 Visual Basic 的控件资源的方法,并给出了从 Visual Basic 的控件文件中获得 ObjectWindows 类的方法。

**关键词** 控件 ObjectWindows VBX 类

## 1 ObjectWindows for C++ 的特点

WINDOWS 系统提供了大量的应用程序接口(API)函数供开发 WINDOWS 应用程序时用。程序员可利用这些已有的内部资源去编程,而不用去做最底层的工作,这大大简化了编制程序的难度和复杂性。但由于 API 函数及各开发语言提供了庞大的函数族,在程序设计过程中,寻找合适的函数成为难事。同时,由于 API 函数所需要的参数过多,这给程序设计和将来的软件维护增加了难度。

面向过程的程序设计有代码无法重用,软件测试维护困难等缺陷<sup>[1]</sup>。面向对象的程序设计方法(OOP)以下一些特点<sup>[2]</sup>:数据与操作数据的方法封装在一起,形成一个类,一个类可以有多个实例,这些实例就是类的对象。类有以下特点:封装性,抽象性,继承性和多态性。利用类的这些特性,可以使设计出的程序安全,简单,维护简便。在程序设计过程中,利用类的多态和继承机制,对已存在的类进行继承,从而设计出所需的类及类方法。

基于以上考虑,将 WINDOWS 系统的 API 函数与它所操作的数据封装起来,得到相应的类,我们用这些类的对象,可实现对 WINDOWS 下各种窗口元素的操作,对不满意的类,只要从原类中派生出新类,增加类的方法和数据或对原类的方法重新定义即可。

ObjectWindows Library(OWL)是为 Windows 应用编写的类库,该类库在不断地丰富<sup>[3]</sup>,OWL 把 Windows 中与窗口有关的部分按类的形式描写出来。编程时,可利用 OWL 中的类生成对象来描述一个窗口元素:如窗口,对话框,滚动条等。这样,OWL 就为用户完成了大多数的的工作:如对某一窗口的登录,窗口的创建与维护等。虽然这些工作是必须的,但不需要全部由程序员来完成。程序员从 OWL 类中派生新类,大部分工作在其父类中已完成,若要定义新的数据和方法或改变窗口构造,只要改造新类即可。

图一为 ObjectWindows 的 OWL 类的层次结构一部分。图中可看出各类之间的继承关系:1)从窗口类

TWindow 派生出对话框类和控制类;2)对话框类分别派生出标准对话框类,打印放弃对话框类和输入对话框类;3)控制类派生出各具特点的控制类,主要有滚动条类,静态类,VbxControl 类,列表框类。VbxControl 类是专门为在 ObjectWindows 程序设计中利用 Visual Basic(VB)的控件资源而设计的。

要利用类资源,须对各个类的继承关系,类的内部结构有深刻的理解。

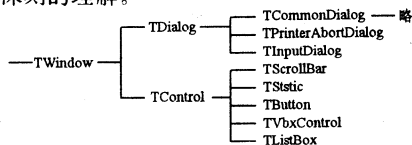


图1 ObjectWindow2.5类层次结构举例

利用类资源可以简化程序的设计过程,例如:一个简单的 ObjectWindows 程序

```

helloapp.cpp
#include <owl\owlpch.h>
#include <owl\application.h>
int OwlMain(int /* argc */, char /* argv */){
    return TApplication("Hello World!").Run();
}
  
```

文件 owlpch.h 定义 owl 的一般类(owlall.h)和 owl 的核心类(owlcore.h),application.h 中定义应用程序类 TApplication;在 TApplication 类中,封装了生成和运行 Windows 应用程序所需要的一切,当生成一个 TApplication 类对象后运行对象方法 Run()时,对象自动完成窗口定义,登记,创建,生成消息循环,消息分发,窗口管理等任务。同样的功能,如用 C++(Not ObjectWindows)来实现,其 whello.cpp 的代码有 200 多行<sup>[4]</sup>,而用 ObjectWindows 实现同样的功能仅需以上几行代码。

利用 OWL 设计程序,可直接从类库中选择所需类,并对所选类作少量的裁剪生成新的类,且类的逻辑层次清晰,容易维护。同时在 Borland C++ 4.5 for Windows(BC45WIN)集成环境中利用 AppExpert 定义程序各模块功能,然后生成 ObjectWindows 的程序,设计者对自动生成的程序进行细微加工或增加模块,可得到所需软件。

## 2 VB 控件及其特点



VB for Windows 给程序设计提供了大量的界面控制部件,这些控件精致实用,利用控件资源,可为程序员节省大量的编程时间,且设计出的程序专业化水平高。

比如:Grid 控件,利用该控件,用户可以生成一个简单表格,并可利用表格的结构特性(Properties)及 VB 提供的方法(Method)消息(Message),可对表格进行操作,以下是对 Grid 的特性,方法,消息的解释<sup>[5]</sup>:

Methods(以 GRID 控件为例):

```
Grid.AddItem item[,index] 向表尾增一行
Grid.RemoveItem index 去除 Grid 表中某行
Grid.Drag 拖动
Grid.Refresh 重画
Grid.Move, left[,top[,width[,height]]] 拖动
Grid.LostFocus[index] 失去焦点
Grid.Zorder[position] 控制在 Z 轴的位置
```

Messages:

```
Click 单击鼠标左键      DblClick 双击鼠标左键
DragDrop 鼠标拖放      DragOver 鼠标坐标变化
GetFocus 获得焦点      LostFocus 失去焦点
KeyDown 键盘释放      KeyDown 键盘按下
KeyPress 按下 ASCII 字符键
MouseMove 鼠标移动      MouseDown 鼠标按下
RowColChange 行列变化  MouseUp 鼠标释放
                          SelChange 选择区行列变化
```

Properties:

BackColor	背景色	Highlight	激活框颜色, True, BLUE, False; 同背景色
ForeColor	前景色	index	序号(主窗口下有多少个表)
Cols	列数	TabStop	是否允许 Tab 键从当前表 index 切换到下一表 index+1
Rows	行数	FixCols	固定列数(纵表头列数)
FixRows	横表头列数	Visible	是否可见
GridlineWidth	线宽	Enable True	True 激活表中任意一个框, False: 不能激活。
...	...	...	...

用以上提供的特性,方法和消息可以实现对 Grid 的操作。如:

Sub Form_Load() ' 初始化 Grid1 行列 Grid1.Rows=9 Grid1.Cols=20 Grid1.AddItem "" ' 向表尾加一行 Grid1.RemoveItem 4 ' 去除第四行 End Sub	' 选择区域变化时产生的消息 SelChange Sub Grid_SelChange() Msg="Now, Select Area Change!" MsgBox Msg End Sub
--	---

VB 中提供了大量的控件,每一个控件对应一个控制文件(\*.VBX)<sup>[5]</sup>,在 ObjectWindows 的编程过程中,利用 VB 控件资源,可提高编程效率和软件质量。

### 3 在 ObjectWindows 编程中利用 VB 控件资源

1) 包含 VBX 控件的 Windows 应用软件的运行机制及有关定义

当一个 VBX 控件发出事件消息,会有下列一系列事件发生:(1)该 VBX 控件发出一个 WM-VBX-FIREEVENT 消息给主窗口 TMyMindow;(2)主窗口 TMyWindow 的 TVbxEventHandler 在它的响应表中发现 WM-VBX-FIREEVENT 消息并且调用 EvVbxDispatch 消息分配函数;(3)如果有子窗口 EvVbxDispatch 发送事件到子窗;(4)如果在子窗对象的消息响应表中有事件处理函数,子窗处理该事件;(5)如果没子窗,或子窗没处理该事件的函数,EvVbxDispatch 将该事件传给主窗口 TMyWindow 的消息响应表。

下面分别对类 TVbxEventHandler 结构和消息 WM-VBX-FIREEVENT 结构作出解释。

TVbxEventHandler 是主窗口的一个子类:	窗口消息 WM_VBXFIREEVENT 的结构:
class TVbxEventHandler:virtual public TEventHandler{ protected: LRESULT EvVbxDispatch ( WPARAM wp, LPARAM lp); LRESULT EvVbxInitForm ( WPARAM wp, LPARAM lp); DECLARE_RESPONSE_TABLE(TVbxEventHandler);	typedef struct VBXEVENT{ HCTL Control; //发出 VBX 消息的 VBX 控件 HWND Window; //处理 VBX 控制的窗口 int ID; //VBX 控制的标识 int EventIndex; //事件序号 LPCSTR EventName; //事件名称 int NumParams; //事件参数数目 LPVOID ParamList; //事件参数指针 VBXEVENT;

在 ObjectWindows 中 VB 消息响应表入口的定义:

有两个功能相同的响应表宏:EV-VBXEVENT-NAME(ControlID,event,EvHandler)和 EV-VBX-EVENTINDEX(ControlID,event,EvHandler),用以标识 VBX 事件的处理函数。其中 ControlID 标识控制,event 事件名称,EvHandler 标识 VBX 消息处理函数。

```
EV_VBXEVENTNAME(IDC_BUTTON1,"MouseMove",Ev-  
MouseMove);
```

```
或:#define Event_Button1_MouseMove 10
```

```
EV_VBXEVENTINDEX(IDC_BUTTON1,Event_Button1,  
MouseMove,EvMouseMove);
```

在 ObjectWindows 中 VBX 消息响应函数的定义,如 VBX Control 中有一个鼠标移动事件:

```
void EvMouseMove(VBXEVENT FAR *event){  
short X=VBX_EVENTARGNUM(event,short,0);  
short Y=VBX_EVENTARGNUM(event,short,1);}
```

其中 event 为事件参数,VBX\_EVENTARGNUM(event,short,index)是宏,它用来分离出 event 中的参数并转化成 C++ 参数类型,short 是分离出的参数的类型,index 是要分离出的参数在 VB 消息函数的参数表中的顺序(从左到右,index 从 0 开始)。

以下是 VB 消息参数类及 C++ 数据类型之间转化的关系:

VB	C++	Macro
Boolean	short	VBX_EVENTARGNUM(event,short,index)
Control	HCTL	VBX_EVENTARGNUM(event,HCTL,index)
Double	double	VBX_EVENTARGNUM(event,double,index)
Enum	short	VBX_EVENTARGNUM(event,short,index)
Integer	short	VBX_EVENTARGNUM(event,short,index)
Long	long	VBX_EVENTARGNUM(event,long,index)
Single	float	VBX_EVENTARGNUM(event,float,index)
String	HILSTR	VBX_EVENTARGSTR(event,index)

以下是个 VB 消息函数中,参数表的顺序 index (从左到右,index 从 0 开始):

Click	无参数	DblClick	无参数
GotFocus	无参数	LostFocus	无参数
KeyDown	Key as Int, Shift as Int	KeyPress	Key as Int, Shift as Int
KeyUp	Key as Int, Shift as Int	MouseDown	X as Int, Y as Int
DragDrop	Source as Control, X as Int, Y as Int		
DragOver	Source as Control, X as Int, Y as Int, State as Int		
MouseMove	X as Int, Y as Int, Shift as Int, Button as Int		

MouseUp X as Int, Y as Int, Shift as Int, Button as Int

在 ObjectWindows 编程时若用 VB 控件, 一般有两种方法: 直接利用 ObjectWindows 提供的 TVbxCtrl 类, 或利用 VBXGEN 生成 VB 控件类。

2) 直接利用 ObjectWindows 中提供的 VBX 控件类编程

在 include \owl\vbxcctl.h 中可查到 ObjectWindows 下与 VB 控件相关的类。VB 的消息响应函数(消息响应表), 方法函数, 特性函数都封装在相关的类中。

vbxcctl.h 中, 有一个 TVbxCtrl 类, 用该类可直接生成 VBX 控件对象。然后可以用类提供的 VBX 方法, 函数对控件对象进行操作, 也可以派生出自己的新类, 重新定义类方法及消息响应表和消息处理函数。

ObjectWindows 中 VB 控件的方法表示:

```
AddItem(int index, const char far * item); Drag(int action); Move
()
Refresh(); ReMoveItem(int index);
ObjectWindows 中 VB 控件的属性设置函数(方法):
GetPro(arg0, arg1, arg2); /* 得到控件的一个特性 */
SetPro(arg0, arg1, arg2); /* 修改控件的一个特性 */
arg0: 参数类型 int proIndex (特性编号), const
char far * name (特性名称); 对 Grid 控件的行, arg0 =
Cols; arg1: 参数类型 int, long, bool, COLORREF,
float, string&, ENUM, HPIC 分别与个特性的性质有
关, Grid 的行数为 10, arg1 = 10, arg3: 参数类型 int,
arg3 = -1。
```

有关直接利用 ObjectWindows 中提供的操作 VBX 控件类的例子见参考文献<sup>[6]</sup>。

3) 利用 VBXGEN 生成特定的 VBX 类, 然后在 ObjectWindows 中对特定的 VBX 控件编程

VBXGEN<sup>[7]</sup> 可以从某一 VBX 控件的库中 (\* .vb) 生成一个头文件, 该头文件中包含 VBX 控件库所支持的类, 类中封装有控件特性, 操作方法, 消息处理函数。由于消息处理函数是空函数, 需从该类中派生出新的类, 定义新的消息处理函数, 然后用新的类生成用户所需的对象。

在 Windows 下运行 VBXGEN.EXE, 然后键入 VBX 库函数和头文件名, 即可生成头文件。

例如, 利用 VBXGEN.EXE 生成 GRID 控件的库 (GRID.VBX) 的头文件 (GRID.H), 该库中有一个 TVbxCtrl 类, 以下是该类的一部分:

```
//特性
....
#define Prop_Grid_Rows 23
#define Prop_Grid_Cols 24
....
//事件
....
#define Event_Grid_MouseMove 10
#define Event_Grid_RowColChange 12
```

```
.....
//默认数据表略
//OWL class
class TVbxCtrl: public TVbxCtrl {
public:
//构造函数
TVbxCtrl(TWindow * parent, int id, LPCSTR title, int x, int y, int w,
int h,
long initLen=0, void far * initData=0, TModule * module=0);
TVbxCtrl (parent, id, " GRID.VBX", " Grid", title, x, y, w, h,
initLen, initData, module) {}
TVbxCtrl(TWindow * Parent, int resourceId, TModule * module =
0);
TVbxCtrl (parent, resourceId, module) {}
//事件处理函数
....
void EvMouseMove (VBXEVENT FAR * ) {}
void EvRowColChange (VBXEVENT FAR * ) {}
//特性操作函数
....
BOOL GetPropRows (int& v) {return GetProp (23, v);}
BOOL SetPropRows (int v) {return SetProp (23, v);}
BOOL GetPropCols (int& v) {return GetProp (24, v);}
BOOL SetPropCols (int v) {return SetProp (24, v);}
...
DECLARE_RESPONSE_TABLE (TVbxCtrl);
};
//OWL 响应表
DEFINE_RESPONSE_TABLE1 (TVbxCtrl, TVbxCtrl)
....
EV_VBXEVENTINDEX (IDC_Grid, Event_Grid_MouseMove,
EvMouseMove),
EV_VBXEVENTINDEX (IDC_Grid, Event_Grid_RowColChange,
EvRowColChange),
....
END_RESPONSE_TABLE;
```

在 ObjectWindows 中使用 VBX 控件时, 要确定 VB 控件是对话框的一部分 (用 WorkShop 预置在对话框中) 还是在程序运行过程中动态的创建, 如果动态的创建就需要一个表文件。ObjectWindows 程序中利用构造函数可动态创建一控制:

```
TVbxCtrl * g = new TVbxCtrl (* window, IDC_GRID " My
Grid", 0, 0, 300, 400,
sizeof (GridData), GridData);
或[7]: HFORMFILE file = VBXCreateFormFile (sizeof (GridData),
GridData); //创建表文件
HCTL c = VBXCreate (... , file); //动态创建控制
VBXDeleteFormFile (file); //删除控制
其中 GridData 必须预先定义。
也可静态创建一控制: TVbxCtrl * g = TVbxCtrl
(* parent, IDC_GRID)。
```

#### 4 结束语

一个软件的设计, 除了针对问题建立好的数学模型和据此建立好的算法外, 提高软件设计的效率和可靠性及良好的用户界面越来越重要, 并且已成为一个商业软件是否成功的决定因素。因此, 要最大限度地利用现有的软件资源, 开发出自己的界面友好, 可靠性高, 可维护性强的软件。用 ObjectWindows 设计软件

并在其中利用 VBX 资源给了我们很好的启示。

### 参考文献

- 1 陈小明等译. PROGRAMMING WINDOWS BORLAND C++4.5 18 电子工业出版社 1995
- 2 蔡希尧. 面向对象技术评述. 计算机应用 1994. 4 2—4
- 3 <http://www.borland.com/>

- 4 Borland C++4.5 for Windows: \bc45\examples\windows\whello
- 5 Microsoft Visual Basic 3.0 for Windows Help
- 6 Borland C++4.5 for Windows: \bc45\examples\owl\owlapp\vbxtcl
- 7 Object Windows 2.5 Help

## Windows 95 的长文件名支持原理剖析

刘绪平 赵学胜

(中国矿业大学北京研究生部 北京 100083)

**摘 要** 本文通过用 Norton Diskedit 对存储了 Windows 95 长文件名的磁盘的目录区进行分析,阐述了长文件名的存储结构,并通过其存储结构解释了长文件名的特点,分析了与 DOS 和 Windows 3.X 的普通文件名兼容的原因,并得出了一些使用长文件名的注意事项和结论。

**关键词** 长文件名 8.3 文件名 目录区 组 ASCII 码 文件分配表

长文件名(LFN: Long Filename)象长变量名一样,有利于对该文件的记忆和理解,但 DOS 和 WINDOWS 3.X 均不支持长文件名。Windows 95 的推出解决了这一问题。

Windows 95 的 LFN 有以下几个特点:

- 文件名最长可达 256 个字符,若包含路径,则路径和文件名的总长可达 260 个字符。

- 在创建长文件名的同时,生成一个 DOS 标准的

8.3 文件名。如:

创建 lxpctest. 1996. 11. 15. txt 的同时自动生成的 8.3 文件名为: lxpctest~1. txt。

- 文件名不区分大小写,但保留大小写。

Windows 95 的 LFN 机制为什么能够具有上述特点,而同时又与 DOS 的 8.3 文件名相兼容呢? 笔者通过用 Norton Diskedit 对存储了 LFN 的磁盘(本文以 1.44M 软盘为例)的文件分配表、目录区、数据区进行分析,基本上搞清了 LFN 的支持原理,利用该原理解释 LFN 的特点,并得出了一个结论,而这些解释和结论在现有资料上是见不到的,现论述如下:

### 1 Windows 95 的 LFN 的存储机制

Windows 95 下磁盘的整体结构未变,文件分配表(FAT)和数据区的结构未变,而目录区的结构则与 DOS 和 Windows 3.X 下有很大区别。一片存储了文件 lxpctest. 1996. 11. 15. txt(其短文件名为 lxpctest~1. txt) dosname. txt 的目录区数据为:

```
11.15.txt(共 59 个 LSI 个 x31 个 p. 其短文件名为 lxpctest~1.txt)的目录区数据为:
```

```
00000000: 42 31 00 31 00 2E 00 31 -- 00 35 00 0F 00 48 2E 00 B1.1.15. H...
00000010: 74 00 78 00 74 00 00 -- FF FF 00 00 FF FF FF FF EXL.....
00000020: 01 6C 00 78 00 70 00 74 -- 00 65 00 0F 00 48 73 00 Lxp1c. Hs...
00000030: 74 00 2E 00 31 00 39 00 -- 39 00 00 00 36 00 2E 00 L.199. 6...
00000040: 4C 58 50 54 45 53 7E 31 -- 54 58 54 20 00 A6 76 51 LXPTEST~1TX...
00000050: 6F 21 6F 21 00 00 76 51 -- 6F 21 02 00 C2 00 00 00 otol. vQol.....
```

与文件名的长度有关。Windows 95 的短文件名和 DOS 的 8.3 文件名存储形式完全一样。

```
00000060: 44 4F 53 4E 41 4D 45 20 -- 54 58 54 20 00 00 00 00 DOSNAME TXT
00000070: 00 00 00 00 00 00 F0 6C -- 6F 21 04 00 62 00 00 00 .....lot. b...
00000080: 4D 2E 00 31 00 35 00 2E -- 00 74 00 0F 00 02 78 00 M.15. L...
00000090: 74 00 00 00 FF FF FF FF -- FF FF 00 00 FF FF FF FF L.....
000000A0: 0C 70 00 70 00 70 00 70 -- 00 70 00 0F 00 02 2E 00 pppppp....
000000B0: 31 00 39 00 39 00 36 00 -- 2E 00 00 00 31 00 31 00 1996.11.
000000C0: 0B 70 00 70 00 70 00 70 -- 00 70 00 0F 00 02 70 00 pppppp~p.
000000D0: 70 00 70 00 70 00 70 00 -- 70 00 00 00 70 00 70 00 pppppp~p.
000000E0: 0A 70 00 70 00 70 00 70 -- 00 70 00 0F 00 02 70 00 pppppp~p.
000000F0: 70 00 70 00 70 00 70 00 -- 70 00 00 00 70 00 70 00 pppppp~p.
00000100: 09 78 00 78 00 78 00 78 -- 00 78 00 0F 00 02 78 00 xxxxxx..x.
00000110: 70 00 70 00 70 00 70 00 -- 70 00 00 00 70 00 70 00 pppppp~p.
00000120: 08 78 00 78 00 78 00 78 -- 00 78 00 0F 00 02 78 00 xxxxxx..x.
00000130: 78 00 78 00 78 00 78 00 -- 78 00 00 00 78 00 78 00 xxxxxx..x.
00000140: 07 78 00 78 00 78 00 78 -- 00 78 00 0F 00 02 78 00 xxxxxx..x.
00000150: 78 00 78 00 78 00 78 00 -- 78 00 00 00 78 00 78 00 xxxxxx..x.
00000160: 06 78 00 78 00 78 00 78 -- 00 78 00 0F 00 02 78 00 xxxxxx..x.
00000170: 78 00 78 00 78 00 78 00 -- 78 00 00 00 78 00 78 00 xxxxxx..x.
00000180: 05 6C 00 6C 00 6C 00 6C -- 00 6C 00 0F 00 02 6C 00 llllll..l.
00000190: 6C 00 78 00 78 00 78 00 -- 78 00 00 00 78 00 78 00 xxxxxx..x.
000001A0: 04 6C 00 6C 00 6C 00 6C -- 00 6C 00 0F 00 02 6C 00 llllll..l.
000001B0: 6C 00 6C 00 6C 00 6C 00 -- 6C 00 00 00 6C 00 6C 00 llllll..l.
000001C0: 03 6C 00 6C 00 6C 00 6C -- 00 6C 00 0F 00 02 6C 00 llllll..l.
000001D0: 6C 00 6C 00 6C 00 6C 00 -- 6C 00 00 00 6C 00 6C 00 llllll..l.
000001E0: 02 6C 00 6C 00 6C 00 6C -- 00 6C 00 0F 00 02 6C 00 llllll..l.
000001F0: 6C 00 6C 00 6C 00 6C 00 -- 6C 00 00 00 6C 00 6C 00 llllll..l.
00000200: 01 6C 00 6C 00 6C 00 6C -- 00 6C 00 0F 00 02 6C 00 llllll..l.
00000210: 6C 00 6C 00 6C 00 6C 00 -- 6C 00 00 00 6C 00 6C 00 llllll..l.
00000220: 4C 4C 4C 4C 4C 4C 7E 31 -- 54 58 54 20 00 83 41 6D LLLLLL~1TX...
00000230: 6F 21 6F 21 00 00 41 6D -- 6F 21 05 00 52 00 00 00 otol. Amol....
```

从以上目录区数据可以看出, DOS 的 8.3 文件名机制中每个文件名占 32 字节(现把 32 的字节定义为一组,以下论述用一组代表 32 个字节),其中有 10 个字节保留。Windows 95 的每个文件名所占用的字节数

Windows 95 的 LFN 所占的存储空间是 32 的倍数,它由两部分组成。第一部分包含一组数据,它存储了创建 LFN 的同时所生成的 8.3 文件名。第二部分包含若干组数据(与文件名的长度有关),存储了完整的 LFN。整个文件名(设共有 n 组数据)在磁盘上的存储顺序为:第二部分的第 n 组, n-1 组, n-2 组, ..., 3 组, 2 组, 1 组, 最后是第一部分。

#### 1.1 第一部分数据

其创建的 8.3 文件名除利用了标准的 DOS 8.3

文件名保留的 10 个字节外,其余的字节定义同 DOS 完全一样。存储时全部使用大写字母。现以 lxpptest. 1996. 11. 15. txt 为例,其 8·3 文件名为 LXPTEST~1. TXT,它在目录区的数据为:

00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	10
4C	58	50	54	45	53	7E	31	54	58	54	20	00	A6	76	51	
6F	21	6F	21	00	00	76	51	6F	21	04	00	62	00	00	00	
10	11	12	13	14	15	16	17	18	19	1A	1B	1C	1D	1E	1F	20

该组各字节的意义为:

- 偏移 00H~07H: 文件名。
- 偏移 08H~0AH: 文件扩展名。
- 偏移 0BH: 属性
- 偏移 0CH: 00H。
- 偏移 0DH: 笔者暂未搞清该字节意义,修改该字节数据并不影响 LFN。

- 偏移 0EH~0FH: 文件创建的时间。
- 偏移 10H~11H: 文件创建的日期。
- 偏移 12H~13H: 文件最后修改的日期。
- 偏移 14H~15H: 均为 00H。
- 偏移 16H~17H: 文件最后修改的时间。
- 偏移 18H~19H: 文件最后修改的日期。
- 偏移 1AH~1BH: 文件的开始簇号。
- 偏移 1CH~1FH: 文件的长度。

文件的属性字节为 20H,读者可以理解为 LFN 的 8.3 文件名为整个 LFN 的档案。

## 1.2 第二部分数据

从第二部分数据的第一组数据开始存储完整的长文件名,写满 32 个字节后,则往前写第二组数据,然后是第三组,第四组……。第二部分的所有组,均有以下规律:

- 每组的偏移 0BH、0CH 处分别为 0F 00
- 每组的偏移 1AH、1BH 处均为 00
- 每组的偏移 0DH 处为一固定数字,不同的文件各不相同,若修改该字节数据,则长文件名丢失(笔者暂未搞清该字节的意义)。
- 每组的偏移 00H 处为本组是第二部分的第几组。如从下往上分别为 01, 02, …… 0A, 0B, 0C, 0D, …… ,最后一组例外,它也是表示该组是第二部分的第几组,但是它不用 01, 02 表示,而是用英文字母表的第 n 个大写字母的 ASCII 码表示,如最后一组为第二部分的第二组,则用第二个字母 B 的 ASCII 码 42H 表示,如最后一组为第二部分的第 13 组,则用第 13 个字母 M 的 ASCII 码 4DH 表示。第二部分最多只能有 20 组(受 256 个字符的限制),所以最大到 T 的 ASCII 码 54H。

- 除以上 6 个字节为固定数字外,文件名按以下方法存储:每存储一个字符(严格按其大小写)的 ASCII 码,后面紧跟一个字节的 00,然后存储下一个字符,再跟一个 00……当遇到特定位置时(即每组偏

移的 00H, 0AH, 0BH, 0CH, 19H, 1AH 处),则跨过去,到非特定位置接着存储。文件名的最后一个字符后仍需跟一个 00。

- 若最后一个组占不满,则先填两个字节的 00,然后充填 FF(遇特定位置除外)。

- 若为长目录名,则第一部分的偏移 0A 处(文件属性字节)为 10H,其余部分与前述完全相同。

## 2 与 DOS 和 Windows 3. X 标准的 8.3 文件名的兼容性分析

Windows 95 的 LFN 之所以能和 DOS 和 Windows 3. X 的 8.3 文件名相兼容,是因为 DOS 和 Windows 3. X 不把 LFN 的第二部分的各组认为是文件名,这样做是必要的,因为:

- 按 DOS 的机制,LFN 的第二部分每一组应为一个文件名。

- 第二部分的各组的 32 个字节与 DOS 8.3 文件的 32 个字节的定义截然不同。

DOS 在列目录名或文件名时,把其目录区(或数据区中有目录或文件名的地方)按 32 字节一组进行显示。为什么 Windows 95 的 LFN 在 DOS 下能够显示其 8.3 文件名而不显示其它的部分?细心的读者肯定已经发现 LFN 的第二部分的各组偏移 0A 处(即文件的属性标志)都为 0F,意为该文件的属性为系统的,隐藏的,只读的,卷标的,所以该组不显示!在此随便告诉读者一个小知识:把文件属性设置为隐藏的,或把文件的起始簇号改写,在某些工具软件如 Norton Command, Petools 中仍能看见,若把属性字节改为 0F,则上述工具软件不能识别该文件,但切记改后不要使用 CHKDSK 或 SCANDISK,它们能把文件毁掉。

LFN 的第二部分的各组偏移 1AH、1BH 处均为 00H,即开始簇号为 0 簇,而 FAT 表的前两个入口点并不用作指出簇状态,而是保留使 FAT 的第一个字节指示磁盘格式的介质描述字节,实际上在数据区并不存在。所以不会发生簇链交叉现象,因此用 DOS 的 CHKDSK 或 SCANDISK 不会破坏 LFN(但不要使用 CHKDSK 或 SCANDISK 对装有 Windows 95 系统文件的硬盘进行操作,那样可能破坏某些有特殊入口的系统文件)。

DOS 的文件名区分大小写,存储时一律转换为大写字母。Windows 95 的 LFN 在存储文件时,它的第一部分(8·3 文件名)全部用大写字母存储,第二部分则严格按读者输入的字母的大小写存储,这就是 LFN 不区分大小写,但保留大小写的原因。

## 3 由 LFN 的存储机制得出的一些结论和注意事项

- Windows 95 推出刚刚一年多,其完全代替 DOS, Windows 3. x 还需要相当长的一段时间,即使装上了 Windows 95,也需要运行一定数量的 16 位应用程序,这些应用程序一般不支持 LFN,它们处理的文

件名不能使用 LFN。例如:Foxpro 2.5 for Windows 虽然能在 Windows 95 中运行,但不能创建一个使用 LFN 的.dbf 文件(若强制创建,系统自动截取前八个字符),也不能打开具有长文件名的数据库。

- 文件拷贝时可能使 LFN 丢失。只有使用 Windows 95 本身(如资源管理器或切换到 DOS 提示符下使用 COPY 命令)才能保证 LFN 完全拷贝,在 DOS 操作系统下或在 Windows 95 中运行 Norton Command 拷贝将使目标文件只具有 8·3 文件名。

- 删除 LFN 时必须使用 Windows 95。使用 Windows 95 删除 LFN 文件时,Windows 95 把 LFN 的第一部分和第二部分的各组的偏移 00H 处均改为 E5,并通过修改 FAT 而释放它所占的数据区。该 LFN 所占的目录区可以用来写其它的文件名,文件的数据区可以用来存放其它的文件。若使用 DOS 或 Windows 3.X 删除 LFN,则操作系统只是把 LFN 的第一部分的第一字节改为 E5,并通过修改 FAT 而释放它所占的数据区。虽然数据区被释放,但 LFN 的第二部分的各组所占的目录区并未释放,除非用 Diskedit、Pctools 等工具软件来修改其第二部分各组的第一字节为 E5,或用 Windows 95 本身带的磁盘整理工具对整个磁盘

进行整理,或对整个磁盘进行格式化。DOS 的 UNDELETE 只能用来恢复 LFN 的 8·3 文件名,而不能恢复整个 LFN,因为它无法恢复 LFN 的第二部分的各组的第一个字符。

- 在目录内使用的 LFN 存放在数据区,而这些数据区可能是不连续的扇区,当使用十六位的磁盘整理程序(如 DOS 的 DEFRAG.EXE 或 Norton 的 SPEEDISK.EXE)时,它们改变了文件的存储扇区,所以 LFN 可能丢失。读者若想整理磁盘,可以使用 Windows 95 本身所带的磁盘管理程序。

- 虽然 LFN 保留大小写,但存储一个名字相同而大小写不同的文件时,原来的文件会被覆盖。

- Windows 95 的 LFN 支持是以不惜存储空间为代价的。DOS 的磁盘管理机制可使 1.44M 软盘在根目录下最多存放 224 个文件或目录。若读者使用长文件名或长目录名,则根目录下最多存放的文件或目录数量会大大降低。所以建议读者尽量少在软盘和根目录下使用 LFN,而应在硬盘和自己的子目录内使用 LFN,这可以使用大量的数据区来存储 LFN。

#### 参考文献

- 1 Supporting Microsoft Windows 95 Microsoft 1995.4

## 多媒体网络技术探讨

倪凯民 唐弋清

(安徽省计算中心 合肥 230031)

**摘 要** 本文从多媒体对网络的要求出发,分析并讨论了多媒体网络技术的结构和技术。

**关键词** 多媒体 网络 OSI ATM FDDI SONET

### 1 引言

多媒体描述了一种新的面向应用程序的技术,该技术基于人的多感官的本性,并演化出计算机存储和处理各种不同类型信息的能力。它将各种不同的数据类型,如文本、图象、声音和视频的数据有机地集成在一起提供了一个单一而统一的信息系统。利用该系统,人们已可以进行一些简单的多媒体应用。如演示、宣传等。但是多媒体技术只有和网络技术相结合,才会有发展潜力。例如,一些比较诱人的多媒体应用:视频会议、多媒体电子邮件,都需要有相应的网络技术支持。由于多媒体技术的特殊性,对网络也有着特殊的要求,本文在分析了多媒体技术对网络的要求后,讨论并分析了与多媒体技术相关的网络技术。

### 2 多媒体对网络的要求

#### 2.1 很高的带宽

一台工作站必须能同时传输数字化的视频、音频、

文本及声音等信息,其传输速率要能分别到达:视频 30MB/S、音频 88Kb/p、文字 12Kb/p 和声音 8Kb/s。如果网络中每个工作站都要有处理多媒体能力的话,则对网络的收集能力要求更是惊人。

#### 2.2 等时服务(Isochronous Services)

多媒体信息传输时对网络延迟特别敏感,它需要等时服务,也即实时通讯,其源和目的地都和时间的有关,比如音频和视频。通过同步实时钟以及建立虚通道可以确保网络有最小的网络延迟。

### 3 多媒体网络模型

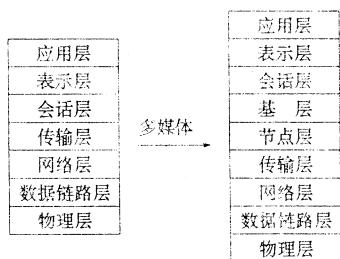
有两种多媒体网络模型:基于 OSI 的分层模型和多媒体服务器模型,它们为传送多媒体信息提供了互补的方法。层次模型强调有组织地在网络上传输数据,通过提供给计算机一个有组织的逻辑通信方法,层次模型可以保证传输的数据独立于系统平台。与其相反,服务器模型检查实际使用和存储的多媒体数据,该模型给出客户工作站怎样使用多媒体及怎样通过服务器存储和存取数据的方法。把这两个方法结合到一起,可



以得到一个结构化的多媒体网络方法,它既适用于人与人交流,又适用于人与信息沟通。

### 3.1 OSI 模型

开放系统互联(OSI)参考模型通过定义一组所有系统都遵循的功能调用来实现互操作性,互操作性可以使不同的系统平台使用同样的方式存取分布在网络中的多媒体信息。标准的 OSI 网络模型如图 1 左边所示。最上边是应用层,允许用户程序调用;应用层下面是表示层,在这一层以用户工作站内部的数据格式传送数据;会话层可把一个在两个不同的工作站上使用的应用程序合并成一个会话;传输层控制数据包的流量;网络层则选择数据传输路径;在网络层下面的是数据链路层,它提供对进入物理层的电信号控制;最下面的是物理层,它包含传输数据的物理元件,如:双绞线、同轴电缆及光纤等。



A. OSI 网络模型

B. 多媒体网络模型

图1 OSI模型和多媒体网络模型

要实现网络多媒体数据库,又称数字化超媒体系统,需对标准的 OSI 模型进行改进。例如,为了协同多媒体信息的网络回放,表示层需要提供对工作站物理元件的控制,多媒体信息的回放需要额外的硬件。音频和视频数据必须传到系统总线上对应的设备上,这就是表示层应该做到的事。

在超媒体系统中,还需要在会话层增加一层——基层(base layer)。以便控制多媒体数据库,因为查询和检索多媒体信息是必须的。常规数据库中存储信息比较简单,一次查询只产生一种数据格式,而多媒体数据库查询必须同时为一个记录项产生多种数据格式,而且产生的结果对客户工作站来说必须是可用的。正由于多媒体数据库的复杂性,OSI 模型中增加一个基层,以便查询系统产生的数据具有相对的独立性。在传输层之上还有节点层(node layer)。它表示包含多媒体信息的各种数据格式的逻辑存储。这一层必须包括从最基本到复杂的多媒体对象,还有对这些对象的标识。一个对象就是包含一个多媒体记录的每种格式,这样多媒体数据库中的一个记录就包含了音频、视频及文本信息。

### 3.2 多媒体服务器模型

和分层模型不同的是,服务器模型强调的是多媒体信息的存储及回放,而不是在网络上传输信息,这一模型的根本是客户/服务器关系。客户工作站接收多媒体

服务器的特定信息或通过服务器给另外一个工作站传送多媒体信息。

在多媒体 OSI 模型的基层及节点控制下,多媒体服务器提供对多媒体信息存储及检索。这些服务器特别考虑到集成到局域网中的多媒体网络运输。例如,视频服务器要对数字化视频信号分帧并压缩以及在局域网上的传送。多媒体服务器的另一个功能是通过用户接口管理器来协调多媒体数据信息的发送,它类似于多媒体 OSI 模型的基层。用户接口管理器也需要协调对包含在音频、图形、视频等服务器的信息查询和检索。

使用多媒体服务器需要增加网络带宽以便服务器传送多媒体信息时不会超载,服务器端的多媒体信息存储也是一个挑战,因为其存储量是很大的。尽管数据压缩技术和高速局域网技术可以缓冲这些问题,但是要真正解决这些问题还需要相关的网络新技术。

## 4 多媒体网络技术

由于传输多媒体信息需要的传输能力及资源远非普通网络所能提供的,因此需要一些比较新的网络通信技术。目前最有前途的网络技术就是:分布式光纤数据接口(FDDI)、异步传输模式(ATM)以及同步光纤网络(SONET)。

### 4.1 FDDI(光纤分布式数据接口)

FDDI 提供了比当前网络高得多的数据传输率:100Mb/s, FDDI 协议是美国国家标准协会 ANSI X3T9.5 委员会的标准局域网协议,也是针对100Mbps 光纤定时令牌传送局域网的 ISO 标准。它采用光纤传输介质、令牌访问方式、反向旋转的双环拓扑结构,并实现100Mbps 的数据传输率。FDDI 的标准共分四个子层,即 MAC、PHY、PMD 和 SMT。它们与 ISO 的 OSI 网络协议的对应关系如下所示。

OSI	FDDI	
数据链路层	LLC	
(DLL)	介质访问控制层(MAC)	SMT 站 管 理
物理层	物理协议层(PHY)	
(PL)	物理介质相关子层(PMD)	

FDDI 的主要特点如下:

- 高环路利用率;
- 高传输速率;
- 大的网络覆盖范围;
- 限定令牌可满足特殊传输要求;
- 完善的站级分布式管理体制 SMT;
- 高可靠性和良好的互操作性。

FDDI 采用令牌传递模式(token-passing scheme)来决定工作站的存取访问。进入环的第一个工作站可建立并测试与环的连接,然后产生一个在工作站间传递的令牌。当一个工作站接收到令牌时,它才能传送一组固定的数据帧。

FDDI 有几个通过带宽分配来实现的优先机制。第一个叫同步带宽分配(Synchronous Bandwidth Allocation, SBA)机制,它可以让管理员将一定量的带宽分配给一确定的工作站或一些工作站,让它们有更大的捕捉令牌机会。

第二个机制叫异步服务,它占用未通过 SBA 分配的带宽并将这部分带宽等分给环上的工作站。异步服务中,每个工作站有一个令牌旋转时间标记(Token Rotation Timer, TRT),可告诉工作站何时得到令牌,当令牌出现时,工作站将 TRT 或期望的到达时间比作目标令牌旋转时间(Target Token Rotation Time, TTRT),TTRT 是令牌旋转一周的预先设定时间,通常为 8ms。如果 TRT 低于 TTRT,工作站就能捕获令牌并传出异步数据帧。如果 TRT 大于 TTRT,则令牌来得太迟,只能进行异步服务工作站必须跟随 SBA 功能的工作站。

FDDI 构成的局域网跨度比总线型网络大,FDDI 在一个单独的局域网上可支持多达 500 个工作站,它的站间距离可达到 2 公里,网络最大长度有 200 公里,超过了局域网范围。FDDI 的双环拓扑,传输介质及相应的控制设施保证了它的高可靠性。FDDI 的双环包括主环与副环。主环用来进行正常工作,当出现故障时,副环可用来构成通路。如果是电缆故障,两边的站将信息流从主环绕到副环,以避免主环的故障。

尽管 FDDI 传输多媒体数据已足有余,100Mb/s 的速率要在 FDDI 构成的局域网上运行多媒体应用程序还是有问题。在一个典型的网络环境下,如果 FDDI 仅用来传输视频数据,一次也仅能同时运行 40 个会话。在一个大一点情况下,要同时需要 40 个以上多媒体会话及常规的网络传输也是很正常的。此外,FDDI 没有考虑所传送的媒体的数据特性,不能为不同的媒体数据选择不同的协议,它的传送机制不适合传送语音及视频信号,而且,FDDI 不可以用作骨干网连接不同的网络更限制了它的应用,所以说,FDDI 并不很适合于作多媒体网络。

#### 4.2 ATM

ATM 网络技术可以提供更高的带宽和更远的网络传输。ATM 网是由国际电报电话咨询委员会 CCITT 和服务厂商如 AT&T 等提出的,并在 1990 年提出的一组建议中确认为宽带综合服务数字网(B-ISDN)的传输方法。所以 ATM 通常又作为 B-ISDN 的同义语。

ATM 的数据传输速率比 FDDI 高。目前有支持 155Mbps 的产品,将来可达 622Mbps 甚至高达 2Gbps。ATM 并不在某一单项用途(如数据传输)上来精雕细琢,它强调的是融合:即声音、数据和图象全部拿来,从几十 Kbps 到几个 Gbps 的带宽范围任你挑选。ATM 排除了电路交换和分组交换的缺点而综合

了它的优点,它是一种以信元为单位的交换技术。简单地说,它把不同长度的信息切成一个一个 53 字节的小碎片,再进行传送。这个“切”和“送”的过程是用硬件来完成的,非常迅速。ATM 协议层次结构如下:

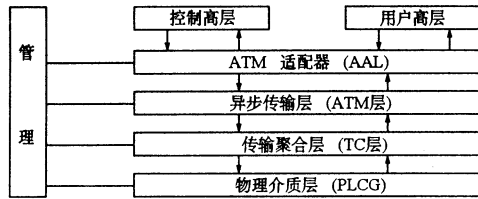


图 2 ATM 协议层结构

ATM 的特色主要有下面几点:

- 支持专用和公用网络;
- 对广域网和局域网采用相同的技术;
- 在普通线路上传输语音、数据和图象;
- 按需分配带宽;
- 提供一个简化的网络结构。

其主要特色要算按需分配带宽,这也是其区别于现有网络的一大优点。在一 ATM 交换机中,这是通过呼叫允许控制 CAC(Call Admission Control)来实现的。其含义是在连接建立之前,申请者经过一谈判过程获得所需的频宽。当然,这一频宽是申请者所能容忍的,即获得一定的业务质量保证。有了这一保证,申请者的多媒体信息便不会受网上繁忙程度的影响,因此,ATM 技术很适合于多媒体。由于包长固定,交换的效率很高,速度也很快。

象其它包交换网络一样,ATM 网络中传输的数据段也大小不一样,但在 ATM 中这些包大小均分成 48 字节一段,然后加上 5 字节的头信息,头信息包含数据的类型及该包来源和目的地。使用头信息可以给对时间敏感的数据(如:音频和视频数据)赋上适当特性,以便进行多媒体数据的无缝传输。

ATM 允许的数据传输率可达 2Gb/s,很容易满足传输多媒体信息的要求。在一个典型的局域中,ATM 可以为每一个用户提供 100Mb/s 的速率,而 FDDI 的 100Mb/s 速率则在所有网络上节点分享。

FDDI 规模有限,而 ATM 既可以在局域网也可以在广域网操作。ATM 还可以用作骨干网,连接不同的拓扑结构的网络。这一特性可使用户在扩展使用 ATM 时保护原来的投资。

由于 ATM 可在同一网络上实现电话、语音、数据通信,而且可通过多路复用技术,实现按需分配带宽,从而可最大限度地提高性能价格比。

当前的 ATM 标准并不支持虚电路交换,即网络节点之间的连接必须一直打开着。这一限制导致占用大量的带宽从而也就限制了工作站对 ATM 的使用,因为使用 ATM 需要很高的网络吞吐量,同样也限制了使用 ATM 作为局域网之间的骨干网,因为这需要不间断的连接。

### 4.3 SONET

SONET 通过从物理上提供极大的传输能力进一步加强 ATM,并把多媒体从局域网扩展到广域网。象 ATM 一样,SONET 也是包交换技术。SONET 以包形式或长度不定的虚容器形式传送数据,效率更高。SONET 的数据传输率可以从 51.84Mb/s 到 2.488Gb/s,足以满足多媒体网络的需要。

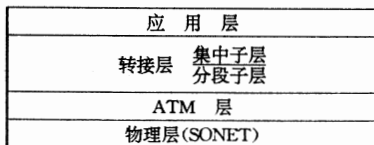


图3 ATM 4层模型

ATM 和 SONET 一起使用一个 4 层的模型(图 3)。用户层建立设备到转接层的连接;转接层功能类似于 OSI 模型中的网络层,它执行诸如:发送信息、载体的选择、选择服务器的质量等功能,转接层可把不同类

型的数据合并成一个流(stream)并保证每种类型数据能接收到它所需的服务,然后在分段子层把数据分成一个个 48 字节的单元,由 ATM 加上 5 字节头信息作纠正错误及确定交换路径使用;ATM 层的功能类似于 OSI 模型中的数据链路层的功能;最低层是物理层,由 SONET 操作,在这里 ATM 单元收集成 SONET 包。

### 5 结束语

本文从多媒体技术对网络的要求出发,讨论了与其相关的网络技术,并分析了其特点。从技术上讲多媒体网络技术已逐步成熟,在国外已开始进入实际应用,但在我国目前网络技术基础比较落后的情况下,到真正实施还有一段距离,相信随着广大科技工作者的努力,在我国普及多媒体网络技术也不是很遥远的事。

#### 参考文献

- [1] [美]坦尼伯姆. 计算机网络. 成都科技大学出版社
- [2] 钟玉琢等. 多媒体计算机技术. 清华大学出版社

## 编译程序开发技术研究

钟乐海 刘宴兵

(四川师范学院计算机科学系 南充 637002)

**摘 要** 本文讨论了在编译程序设计中涉及的一些问题,总结和对比了此领域的研究成果,提出了使用系统编程语言编写编译程序时可采用的几项有效的开发技术。

**关键词** 遍 自编译 自展 交叉编译 后端 抽象机

### 1 引言

编译程序是一个非常复杂的系统软件,虽然编译理论与编译技术的不断发展,已使编译程序的生产周期不断缩短,但目前研制一个较好的编译程序仍需要很长的时间,而且工作相当艰巨。因此,使编译程序的开发过程自动化、短期化、高效化是很有意义的事情。

早期构造的编译程序常用凑合的方法,进展速度慢,且缺乏结构性。本文提出的几种编译程序开发技术能克服这些缺陷。

### 2 考虑因素

在编译程序的设计过程中需要考虑诸多因素,而且在某种程度上,这些因素支配着编译程序的整个设计过程。

#### 2.1 遍数问题

许多编译程序的编写者感到单遍编译程序的概念是有吸引力的,这是因为这种编译程序无需顾虑遍之间的通讯、中间语言等问题,而且不难把程序的错误同源文本联系起来。也不存在一遍的程序应当如何修补错误以便向下一遍传送某些有意义的信息。单遍编译程序通常要比多遍编译程序快些,除非它们过多地回

溯或超前搜索以便解决问题,如果加上另一遍,这个问题就会被处理得更好些。单遍编译程序还可能产生出较慢的目标代码,因为它不可能有能力作大量的优化。

#### 2.2 中间语言

编写一个含有一遍以上扫描的编译程序所受的一个惩罚是:需要设计一个中间语言来表示遍与遍之间的源文件(即作为一遍的输出以及后继遍的输入)。多遍编译程序的一个方法是,它由一些预先的遍组成、建立符号表等等,以此来丰富这个源文本,使得分析(也可能还有代码生成)可在一次扫描中完成。

#### 2.3 目标语言

有时可能把一个编译程序同机器有关的部分和同机器无关的部分分开。如果要使编译程序成为可移植的,这显然是合意的。但这样做的一个方法是定义一个称为中间目标语言或中间语言的语言,它起着两个部分之间的一个界面作用,这两个部分指的是把源代码翻译成目标代码的同机器无关部分,以及从目标语言翻译成具体的机器代码的同机器有关部分。如果可使用相同的中间目标语言,则在  $n$  个机器上实现  $m$  个语言的问题实际上只需写出  $n+m$  个软件,即  $m$  个同机器无关的编译程序的‘前端’,以及  $n$  个同机器有关的编译程序的‘后端’,而不用实现  $n \times m$  个完整的编译

程序。中间目标语言也可当作一个机器,称为抽象机,因为一个编译程序的前端将把它看作象是一个目标机器。

### 3 有效技术

鉴于上面考虑的因素,下面提出几种编写编译程序时可采用的有效的开发技术,以便在尽可能短的周期内开发出效率高、移植性好和容错性强的编译程序。

#### 3.1 自编译

一个编译程序要涉及三个语言:源语言(记为 SL)、目标语言(记为 OL)和用来编译程序的语言(记为 WL),它们之间的关系用图 1 表示(“T”型图)。

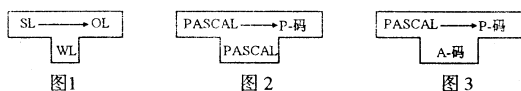


图 1

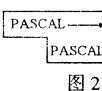


图 2

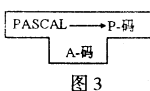


图 3

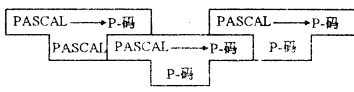


图 4

假设一台机器上已有一个 PASCAL 语言可以运行,那么我们可以用 PASCAL 语言编写一个 PASCAL 语言的编译程序(图 2),然后借助原有的 PASCAL 编译程序(图 3)编译后,可以得到一个可能比原有 PASCAL 编译程序更好的编译程序,这个编译程序与原来那个编译程序的差别是,它是用 PASCAL 语言本身写的,并用 PASCAL 原有编译程序产生的,这一过程可用图 4 表示。可把这种产生编译程序的方法称为自编译方法,它比起用汇编语言写编译程序要省得多,从而可以缩短开发周期。

#### 3.2 交叉编译

首先,用 A 机器的 PASCAL 语言写一个编译程序,它的源语言是 PASCAL,目标语言是 B-码(图 5),然后利用 A 机器的 PASCAL 编译程序进行编译,可以得到一个交叉编译程序(图 6),最后再利用所得到的交叉编译程序去编译图 5 中的用 PASCAL 写的 PASCAL 编译程序。这样,就可得到能在 B 机器上运行的编译程序了(图 7)。这种用 A 机器上的编译程序产生 B 机器的目标码的编译过程称为交叉编译。

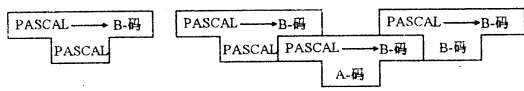


图 5

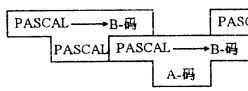


图 6

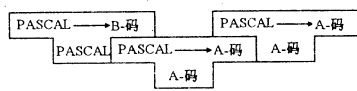


图 7

以上两种方法都是假设已经有一个系统编译语言可以使用,如果没有这样的系统编译语言,则可以采用下面的方法。

#### 3.3 自展

自展的基本思想是,首先确定一个非常简单的核

心语言  $L_0$ ,用机器语言或汇编语言写出它的编译程序  $C_0$ ,再把语言  $L_0$  扩充为  $L_1$  ( $L_0 \subset L_1$ ),并用  $L_0$  来编写  $L_1$  的编译程序  $C_1$ 。如此逐渐扩展下去,可以得到一个系统编程语言族:  $L_0 \subset L_1 \subset L_2 \dots \subset L_K$ ,则  $L_K$  便是我们所要的语言,其编译程序  $C_K$  可用  $L_{K-1}$  编写。这种滚雪球式自展方法可以大大减少开发工作量,因为最初的核心语言的编译程序比较简单,可以用较少的人力、物力迅速完成,而以后各级编译程序均可用前一级已实现的较高级语言编写,因此,完成既迅速又能保证质量。

#### 3.4 移植

如果一个程序能比较容易地从一台机器上搬到另一台机器上运行,则称该程序是可移植的。移植一个程序的工作量要远小于开发它的工作量才有意义。一个可移植的编译程序通常由两部分组成,一部分与源语言有关,称其为前端;另一部分是与目标机有关,称为后端。为了便于移植,通常设计一种抽象机来作为前、后端之间的接口。编译程序的前端将源程序转换为抽象机指令,后端再将抽象机指令转换为机器指令。具体过程如下:

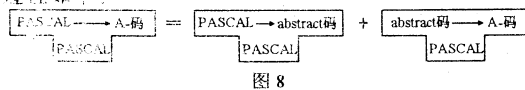


图 8

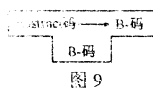


图 9

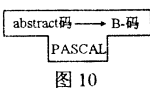


图 10

设 A 机器上有可移植的 PASCAL 语言编译程序,该编译程序是用 PASCAL 语言写的(图 8)。

首先用 B 机器语言写一个 B 机器的后端(图 9),这个后端也可以在 A 机器上通过交叉编译的方法得到。

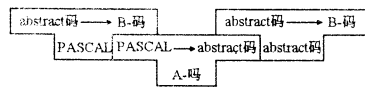


图 11

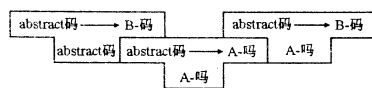


图 12

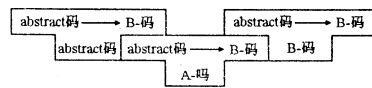


图 13

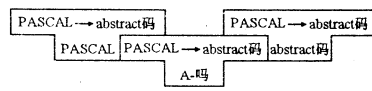


图 14

然后用 PASCAL 语言为 B 机器写一个后端(图 10),利用 A 机器上的前端可以得到抽象机的后端(图

11),再利用 A 机器上的后端便可得到交叉编译程序(图 12),最后用所得到的交叉编译程序去编译图 11 中的结果,即用抽象机写的后端,便可以得到 B 机器上运行的后端了(图 13),前端的移植过程如图 14 所示。

#### 4 后记

本文从不同的角度对编译程序的设计作了探究,提出了一些可靠的设计方案。当然,试图设计一个编译程序对任何输入它都会产生恰当的输出(目标代码或出错信息),这种想法已超出了当前的实际技术水平,要达到这水平还需作不懈的努力。

#### 参考文献

- 1 J. P. Trtmbly, The Theory and Practice of Compiler Writing, Mcgram-Hill, 1985.
- 2 Robin Hunter, The Design and Construction of Compilers, Jihn Wiley & Sons, 1981.
- 3 Crady Booch, Software Engineering with ADA, Benjamin/Cummings, 1983.
- 4 R. E. Berry, B. sc. Programning Language Transtation, Ellis Horwood Limited Publishers. Chichester, 1981.
- 5 陈火旺、钱家骅、孙永强. 程序设计语言编译原理. 国防工业出版社, 1980

## Windows 环境下图形的局部放大

张 海 潘光斌

(中物院电子工程研究所 成都 610003)

**摘 要** 本文介绍了在 WINDOWS 环境下如何实现对图形某个局部进行的放大操作,并给出了一个应用实例。

**关键词** 图形 放大 Windows 应用程序

### 1 前言

在现实世界里我们对于感兴趣的图象可以通过接近目标或采用放大镜等多种方式获得其细微的感受,而在计算机中,对于一幅图形,如果我们想观看它的某个局部,可以在相应的区域里定义一个观察窗口,然后将该窗口内的图形放大一定比例,这样我们也许就能发现放大前所不能看到的微弱变化。

对于上述方法我们一般将其称为橡皮筋技术,因为我们可以模仿橡皮筋的弹性特征随意定义需放大的区域,不过公开的文献通常只是从概念上简单地描述了该方法,而对于其实现技术细节则很少见到。考虑到该方法在实用中有较大的应用价值,本文较为详细地介绍了其实现的技术细节。

在具体实现时我们采用了剪辑板数据交换方式,为了更好地理解实现过程,有必要首先认识一下剪辑板的基本知识。

### 2 剪辑板的工作原理

剪辑板是 WINDOWS 环境下重要的数据交换工具,通常人们使用它来完成两个不同应用程序间的数据交换,不过这里它被用来在一个程序的两个消息之间传递信息。

简单地讲,剪辑板所传递的不是信息本身而是信息的句柄,需要使用该信息的程序或消息可以根据句柄找到所需内容。它的具体使用步骤是这样的:

首先我们需要打开剪辑板,这可以通过调用函数

OpenClipboard 实现,它的作用一方面是检查剪辑板是否可用,另一方面还可以获得对剪辑板的控制权,从而防止其他应用程序改变剪辑板内容,如果该函数的返回值不为零则表明调用成功。由于已经控制了剪辑板,显然可以利用它进行数据交换的工作了,不过在实质性工作开始前还应清除剪辑板上的垃圾,这时需要调用函数 EmptyClipboard,该函数执行后会释放剪辑板中数据的句柄,这些句柄是其他程序留在剪辑板上的,若不清除会造成当前程序对数据的错误引用。在做好上述准备工作后,我们就可以通过调用 SetClipboard 将待传送数据的句柄传送给剪辑板,然后用 CloseClipboard 将其关闭,当程序响应某个消息(如放大或 WM\_PAINT)时,它将打开剪辑板并使用 ClipboardFormatAvailable 检查数据是否可用。在本程序中数据指的是位图,不过这一步不是必须的,同样的功能使用 GetClipboardData 也能实现,如果剪辑板中没有位图,则该函数返回 NULL,否则返回位图的句柄。有了位图的句柄程序就能够生成内存设备环境,然后从剪辑板中把位图选进设备环境并在某个文档窗口中进行放大后图形的显示,至此我们就完成了整个数据交换的过程。

### 3 实例

下面的实例描述了图形的区域选取、放大的全过程。

当鼠标左键按下时,取得按下点的横坐标值与纵坐标值。将其赋予 zs,fs 与 zx,fx。其中 zs,zx 是初始点的坐标值,在一次从鼠标左键按下到放开的过程中保

持不变;而 fs,fx 表示前一鼠标左键按下点的坐标值,在后面处理消息 WM\_MOUSEMOVE 时,我们将会看到它们的值随鼠标左键的移动而改变。

Case WM\_LBUTTONDOWN:

```
zs=LOWORD(IParam);
zx=HIWORD(IParam);
fs=LOWORD(IParam);
fx=HIWORD(IParam);
bBlocking=TRUE;
org=MAKEPOINT(IParam);
org.x=zs;
org.y=zx;
InvalidateRect(hwnd,&rect,TRUE);
return 0;
```

当用户拖动鼠标左键时,需根据新的坐标值不断地画出新矩形以标识所选中的图形,通过调用函数 Line To 在处理每条新的 WM\_MOUSEMOVE 消息时,在所选图形周围画一个边框来实现。在每次鼠标移动时,需先覆盖掉前一次所画边框,这可以通过调用函数 SetROP2(二进制光栅模式设为 R2\_NOT)来实现。

case WM\_MOUSEMOVE:

```
if(bBlocking)
{
    ys1=LOWORD(IParam);
    yx1=HIWORD(IParam);
}
if(zs==0&&zx==0)
break;
if((ys1!=fs)|| (yx1!=fx))//判断鼠标位置是否改变。
{
    hdcZoom2=GetDC(hwnd);
    //清除前次所画边框。
    SetROP2(hdcZoom2,R2_NOT);
    MoveTo(hdcZoom2,zs,zx);
    LineTo(hdcZoom2,fs,zx);
    MoveTo(hdcZoom2,zs,zx);
    LineTo(hdcZoom2,zs,fx);
    MoveTo(hdcZoom2,fs,fx);
    LineTo(hdcZoom2,fs,zx);
    MoveTo(hdcZoom2,fs,fx);
    LineTo(hdcZoom2,zs,fx);
    //得到当前鼠标位置并画出新框。
    fs=ys1;
    fx=yx1;
    MoveTo(hdcZoom2,zs,zx);
    LineTo(hdcZoom2,fs,zx);
    MoveTo(hdcZoom2,zs,zx);
    LineTo(hdcZoom2,zs,fx);
    MoveTo(hdcZoom2,fs,fx);
    LineTo(hdcZoom2,fs,zx);
    MoveTo(hdcZoom2,fs,fx);
    LineTo(hdcZoom2,zs,fx);
    ReleaseDC(hwnd,hdcZoom2);
}
return 0;
```

当鼠标左键放开时,先调用自定义函数 InvertBlock()转换选中的图象,该函数的源代码在后面给出。然后创建一个内存设备环境,并根据 chd.x,chd.y 生成一个位图。如果成功生成这个位图,则将其选入内存设备环境,并用拷贝选中的图象。然后打开和清除剪辑板,把位图传送给剪辑板后将其关闭。

case WM\_LBUTTONUP:

```
ys=LOWORD(IParam);
yx=HIWORD(IParam);
if(bBlocking)
{
    chd=MAKEPOINT(IParam);
    chd.x=ys-zs;
    chd.y=yx-zx;
    InvertBlock(hwnd,org,chd);
    InvertBlock(hwnd,org,chd);
}
if(!bBlocking)
break;
if(chd.x==0||chd.y==0)
break;
hdcZoom1=GetDC(hwnd);
hdcMem=CreateCompatibleDC(hdcZoom1);
hBitmap=CreateCompatibleBitmap(hdcZoom1,abs(chd.x),abs(chd.y));
if(hBitmap)
{
    SelectObject(hdcMem,hBitmap);
    StretchBlt(hdcMem,0,0,abs(chd.x),abs(chd.y),
        hdcZoom1,org.x,org.y,chd.x,chd.y,SRCCOPY);
    OpenClipboard(hwnd);
    EmptyClipboard();
    SetClipboardData(CF_BITMAP,hBitmap);
    CloseClipboard();
    InvalidateRect(hwnd,NULL,TRUE);
}
else
    MessageBeep(0);
DeleteDC(hdcMem);
ReleaseDC(hwnd,hdcZoom1);
return 0;
}

void InvertBlock(HWND hwnd,POINT org,POINT chd)
{
    HDC hdc;
    hdc=CreateDC("DISPLAY",NULL,NULL,NULL);
    ClientToScreen(hwnd,&org);
    PatBlt(hdc,org.x,org.y,chd.x,chd.y,DSTINVERT);
    DeleteDC(hdc);
}
```



当处理消息 WM\_PAINT 时,打开剪辑板并检查位图是否可用。如果剪辑板中含有位图,则生成内存设备环境,从剪辑板中把位图选进设备环境,然后选择合适的参数进行放大显示。

```
case WM_PAINT:
    hdcZoom1=BeginPaint(hwnd,&ps);
    OpenClipboard(hwnd);
    hBitmap=GetClipboardData(CF_BITMAP);
    if(hBitmap!=NULL)
    {
        hdcMem=CreateCompatibleDC(hdcZoom1);
        SelectObject(hdcMem,hBitmap);
        GetObject(hBitmap,sizeof(BITMAP),(LPSTR)&bm);
        SetStretchBltMode(hdcZoom1,
        COLORONCOLOR);
        StretchBlt(hdcZoom1,0,0,cxClient11,cyClient11,
        hdcMem,0,0,bm.bmWidth,bm.bmHeight,SRC-
```

COPY);

//cxClient11,cyClient11 为客户区的长度和宽度。

```
SetCursor ( LoadCursor ( NULL, IDC_
ARROW));
```

```
DeleteDC(hdcMem);
```

```
}
```

```
CloseClipboard();
```

```
EndPaint(hwnd,&ps);
```

```
return 0;
```

#### 4 结束语

对图形进行放大有很多方法,剪辑板只是其中的一种途径,比如待放大图形的数据传递还可以通过动态链接库或动态数据交换 DDE 等方法实现,本文的目的在于抛砖引玉,相信大家经过认真思考后会找到更好的实现方法。

#### 参考文献

- 1 Microsoft Corporation 编,宋明华等译.《WINDOWS V3.0 程序设计指南和工具》.电子工业出版社,1991

(上接 54 页)

1. 单声道、8 位样本值(每个样本 1 字节) 样本 1    样本 2    样本 3    样本 4...	3. 单声道、16 位样本值(每个样本 2 字节) 样本 1                  样本 2... 低字节    高字节    低字节    高字节
2. 双声道、8 位样本值(每个样本 2 字节) 样本 1    样本 2    样本 3    样本 4 声道 0(左)    声道 1(右)    声道 0(左)    声道 1(右)...	4. 双声道、16 位样本值(每个样本 4 字节) 样本 1                  样本 2... 声道 0(左)    声道 1(右)    声道 0(左)    声道 1(右)...

波形数据信息是以一个样本值为单位连续(单声道情况下)或交替(多声道情况下)存放的每个样本值用一个整数来表示,其长度取包含该样本值最小字节数。例如,若样本值位数为 12,则用来表示样本的值整数长度为两个字节(16 位)。

#### 2 文件格式的获取和语音波形的输出

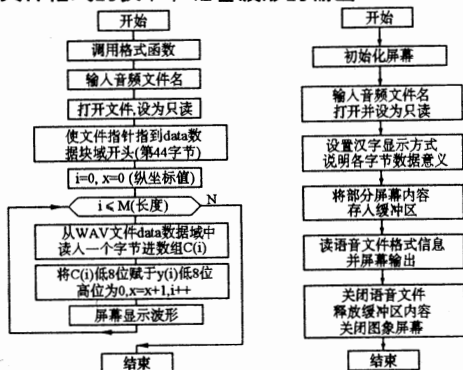


图1 音频文件波形的恢复

图2 文件格式信息的获取

图1是对多媒体语音信号文件进行波形恢复的程序设计框图,它可在不同环境下随时被调用显示信号的波形。图2是获取具体音频文件格式信息的程序框图。程序用 Turbo C 编写。

图1程序读出 PCM 方式下音频文件第 44 字节开始的数据信息并屏幕显示波形。在 8 位 PCM 方式下

读取 data 数据域时要将 8 数据读入一整型(16 位)变量,这里要进行位运算即先将 0 取反变成十六位 1 然后移 8 位再取反(成左八位 0 右八位 1)再和 C[i]与。图2读出 PCM 方式下某一音频文件 0~43 字节的所有格式信息,并用漂亮的汉字界面显示说明。因图像方式与 printf 屏幕输出会发生冲突,故将部分屏幕内容存入缓冲区,待格式信息输出完毕再释放。由于版面有限在此仅给出框图。

#### 3 语音数据与波形的应用

在语音信号处理中,需要对音频数据信息进行加工和处理。通常处理过程及处理前后的信息变化情况常以具体的语音信号波形中来表征。利用这里读出的音频信号数据,我们可以直接对其进行语音压缩、语音编、解码新方法的探索,并通过波形观察比较不同编码效果。

另外,利用多媒体语音系统我们还可以用语音数据和波形方便地进行噪声模拟分析,语音特征提取研究,以及语音识别和训练等应用方面的实验。如:通过设计软件和硬件电路,用声音去控制计算机工作。将人机界面从键盘数字领域引入模糊控制领域,训练计算机建立相应的语音字库。在硬盘上建立起字、词语音模板库。进行特定人的语音识别等等。

#### 参考文献

- 1 李智渊等编著.多媒体原理、技术与应用.电子工业出版社,1995.9
- 2 [美]L.R 拉宾纳等著,朱雪龙译.语音信号数字处理.科学出版社,1983.2
- 3 方建淳编著.语音合成技术与单片微机综合系统.北京航空航天大学出版社,1993.2

# WINDOWS 顶级窗口技术的实现

刘晓梅

(山东经济学院计算机管理系 济南 250013)

**摘 要** 该文对 WINDOWS 环境下的顶级窗口技术的概念,应用领域进行了详细地阐述,并介绍了这一技术的两种实现方法,还提示了编程的要点。最后给出 WINDOWS 下 Borland C 的编程实例。

**关键词** WINDOWS 顶级窗口 前台程序 后台程序

许多 WINDOWS 的工具软件,如多媒体演播控制软件,在线翻译软件、汉字输入软件等,大都提供了一种特殊功能,即可以将其设置为“顶级窗口”;在系统控制菜单或应用程序菜单中,增加了一条名为“总在最前面”(英文版为“Always On Top”)的命令,执行该命令后,即使在其它任何应用程序处于最大化运行状态时,该工具软件的窗口也是可见的。本文中将这些 WINDOWS 窗口称为顶级窗口。下面详细介绍顶级窗口的概念、应用领域和实现方法。

## 1 顶级窗口的概念

要了解顶级窗口的确切含义,必须先透彻理解 WINDOWS 的一个最基本的概念——前台应用程序和后台应用程序。WINDOWS 的多任务机制允许同时打开多个应用程序,但是其中只有一个应用程序,即最后被选定的应用程序会被 WINDOWS 指定为当前正在使用的程序,该应用程序被称为“前台应用程序”,它所在的窗口称“前台窗口”、“当前窗口”或“活动窗口”。“活动”即指当前正在使用的窗口或其最小化的形式——图标(Icon)。WINDOWS 总是将下一次击键或所选择的命令作用于活动窗口。其它未被选定的应用程序则一律被称作“后台应用程序”,它们所在的窗口统称“后台窗口”或“非活动窗口”。前台程序通常会享有更多的系统资源,如计算机处理时间、键盘输入等等。

区分前台应用程序和后台应用程序的方法有两种:

### 1) 根据标题栏的颜色区分:

活动窗口的标题栏总是以一种区别于其它窗口的标题栏的颜色显示,当然具体的显示颜色还取决于系统目前正在使用的配色方案,但不论在何种配色方案下,它都与其它标题栏的颜色有所不同。如果一个图标是活动的,它的标识名也将改变颜色。这是区分活动窗口和非活动窗口的最可靠的方法。

### 2) 根据窗口在 WINDOWS 台面上所处的位置区分:

WINDOWS 总是假定处于其最上层的窗口为活动窗口。这种方法有一个例外情况,就是如果最上层的

窗口是顶级窗口,那么它很可能不是活动窗口。

由此可知,顶级窗口的概念可以这样定义:如果有这样的一个应用程序,即便是处在后台运行的状态,其窗口或图标的工作区域也不会被其它窗口所覆盖,这种(应用程序)窗口称为顶级窗口。这里我们可以看出,虽然顶级窗口的位置总是处于 WINDOWS 工作台面的最前面,但它很可能并不是当前的活动窗口,而是一个后台应用程序。事实上,这一点也正是所谓的“顶级窗口”名称的由来。

## 2 顶级窗口的应用领域

随着 WINDOWS 系统的日益普及,越来越多的应用软件采用了顶级窗口技术。目前,顶级窗口的典型应用场合可以大致分为三类:

### 1) 各种中文输入方法的提示行

WINDOWS 中文版本一般均提供多种中文输入方法,为了保证这些输入方法的提示行不至于被其它应用程序覆盖,必须采用顶级窗口技术。实际上,这也是最早采用顶级窗口技术的一类典型应用。

### 2) 多种多样的工具软件

多种工具软件如时钟、计算器、ASCII 码表、编辑器、拨号器(Phone Dialer)、E-mail 和传真工具、卡片文档、书写器以及绘画器等等均可根据其特定需求设计为顶级窗口的形式。事实上,需要在线翻译及捕捉功能的各种英——汉、汉——英词典软件是这类工具中最为固定的顶级窗口的应用方向。

### 3) 各类多媒体软件

最为常见的如多媒体家庭影院及其它应用环境中的 VCD 播放的控制面板、声卡的调音台等多采用顶级窗口技术以方便用户的使用。随着多媒体应用领域的不断拓展,顶级窗口的应用势必更为普遍。

## 3 顶级窗口技术的实现方法

### 1) 利用 SetWindowPos() 函数

SetWindowPos() 函数的功能为改变指定窗口的大小、位置和窗口顺序。该函数的原型为:

```
void SetWindowPos(hWnd, hWndInsertAfter, x, y, cx, cy, wFlags)
```

参数说明如下:

参数	类型	描述
hWnd	HWND	窗口句柄
hWndInsertAfter	HWND	hWnd 前面的窗口
x	int	窗口左上角 x 坐标
y	int	窗口左上角 y 坐标
cx	int	窗口新宽度
cy	int	窗口新高度
wFlags	WORD	可能影响窗口位置和 大小的 16 位值

其中,当 hWndInsertAfter 取值为 HWND\_TOPMOST 时,窗口 hWnd 被设置为顶级窗口。反之,当取值为 HWND\_NOTOPMOST 时,顶级窗口功能被取消。

## 2) 利用 CreateWindowEx() 函数

任何一个 WINDOWS 程序员对 CreateWindow() 函数都不会很陌生,但使用过 CreateWindowEx() 函数的可能不会太多。该函数的功能是创建一个具有扩展格式的窗口。函数原型为:

```
HWND CreateWindowEx(dwExStyle, lpszClassName,
                    lpszWindowName, dwStyle, x, y, nWidth,
                    nHeight, hwndParent, hmenu, hinst, lpvCreateParams)
```

可以看出,该函数只不过比 CreateWindow() 多了第一个参数,其它参数完全相同,而这第一个参数 dwExStyle 为窗口的扩展格式,当该值取 WS\_EX\_TOPMOST 时,窗口设置为顶级窗口。

## 4 程序实例

下面是笔者采用第一种方法实现的一个程序实例,该程序在一个最简单的 WINDOWS 应用程序的基础上,通过对系统菜单的动态修改,增加了“总在最前面”命令。

有兴趣的读者可以参照本文介绍的方法,使自己的 WINDOWS 程序具备“顶级窗口”功能,编程注意事项如下:

- 1) 初始化部分(包括动态修改系统菜单,设置“顶级窗口”)的位置应在创建窗口之后,显示窗口之前;
- 2) 利用动态修改系统菜单功能时,必须在处理消息循环的最后调用 DefWindowProc() 函数。

下面给出所有的源代码清单,包括两个文件:topmost.def 和 topmost.c。该程序在 WINDOWS 3.1 下的 Borlandc3.1 环境调试通过。

topmost.def 文件:

NAME	TOPMOST
DESCRIPTION	“Basic Window”
EXETYPE	WINDOWS
CODE	PRELOAD MOVEABLE
DATA	PRELOAD MOVEABLE MULTIPLE
HEAPSIZE	2048
EXPORTS	WndProc

topmost.c 文件:

```
#include <windows.h>
#include <stdlib.h>
#include <string.h>

#define IDM_TOPMOST 500
int topmost=1;

long FAR PASCAL WndProc(HWND hWnd, WORD iMessage,
                        WORD wParam, LONG lParam);
int PASCAL WinMain(HANDLE hInstance, HANDLE hPrevInstance,
LPSTR lpszCmdParam, int nCmdShow)
{
    HWND hWnd;
    MSG Message;
    WNDCLASS WndClass;
    HMENU hMenu;

    if(! hPrevInstance){
        WndClass.cbClsExtra=0;
        WndClass.cbWndExtra=0;
        WndClass.hbrBackground = GetStockObject (WHITE_BRUSH);
        WndClass.hCursor=LoadCursor(NULL, IDC_ARROW);
        WndClass.hIcon=NULL;
        WndClass.hInstance=hInstance;
        WndClass.lpszClassName="TOPMOST";
        WndClass.lpszMenuName=NULL;
        WndClass.style=CS_HREDRAW|CS_VREDRAW;

        RegisterClass(&WndClass);
    }

    hWnd=CreateWindow("TOPMOST",
                      "Basic Window",
                      WS_OVERLAPPEDWINDOW,
                      CW_USEDEFAULT,
                      0,
                      CW_USEDEFAULT,
                      0,
                      NULL,
                      NULL,
                      hInstance,
                      NULL);

    hMenu=GetSystemMenu(hWnd, FALSE);
    AppendMenu(hMenu, MF_SEPARATOR, 0, NULL);
    AppendMenu(hMenu, MF_STRING, IDM_TOPMOST, "总在最前面");
    CheckMenuItem(hMenu, IDM_TOPMOST, MF_CHECKED);
    SetWindowPos(hWnd, HWND_TOPMOST, 0, 0, 0, 0,
                  SWP_NOMOVE | SWP_NOSIZE | SWP_SHOWWINDOW);
    ShowWindow(hWnd, nCmdShow);
    UpdateWindow(hWnd);
```

```
while(GetMessage(&Message,0,0,0)){
    TranslateMessage(&Message);
    DispatchMessage(&Message);
}
return Message.wParam;
}

long FAR PASCAL WndProc(HWND hWnd, WORD iMessage, WORD wParam, LONG lParam)
{
    HMENU hMenu;
    switch(iMessage){
    case WM_SYSCOMMAND:
    switch(wParam){
    case IDM_TOPMOST:
    hMenu=GetSystemMenu(hWnd,FALSE);
    if(topmost){
    CheckMenuItem ( hMenu, IDM _ TOPMOST, MF _ UNCHECKED
    topmost=0;
    SetWindowPos(hWnd,HWND_NOTOPMOST,0,0,0,0,
    SWP _ NOSIZE | SWP _ NOMOVE | SWP _ SHOWWINDOW);
    }
    else{
    CheckMenuItem ( hMenu, IDM _ TOPMOST, MF _ CHECKED);
    topmost=1;
    SetWindowPos(hWnd,HWND_TOPMOST,0,0,0,0,
    SWP _ NOSIZE | SWP _ NOMOVE | SWP _ SHOWWINDOW);
    }
    return(0);
    default;
    break;
    }
    break;

    case WM_DESTROY:
    PostQuitMessage(0);
    return(0);
    default;
    break;
    }
    return ( DefWindowProc ( hWnd, iMessage, wParam, lParam));
}
```

## Windows 环境下多媒体应用程序设计

姚 青 贾智平

(山东工业大学计算机系 济南250061)

**摘 要** 本文介绍了 Windows 环境下多媒体应用程序设计所依据的硬件、软件环境及工具。

**关键词** 多媒体 编程 WINDOWS

多媒体技术的应用已经渗透到社会生活的各个领域,教育培训系统,导购、导游系统,电子图书,演示系统,信息管理系统与办公自动化,讲演辅助系统,视频会议及通讯系统等。目前的所有应用软件开发几乎都与多媒体技术有关,因而掌握多媒体制作技术应该是广大计算机编程人员比较重要的一环。

多媒体应用程序设计可以基于 DOS 和 WINDOWS 两种环境,孰优孰劣,众说纷纭,笔者致力于 WINDOWS 环境下的多媒体应用程序的设计开发,在此方面积累了少量的经验,总结于此,以飨同仁。

### 1 Windows 的多媒体特性

#### 1) 媒体控制接口(MCI)

Windows 3.1 以上版本都定义了多种媒体接口标准,提供了功能强大的函数库。作为多媒体的标准平台,绝大多数多媒体厂商都在 Windows 上开发了大量

的产品,并支持 Windows 的标准接口。功能最强,应用最广泛的标准首推 MCI(Media Control Interface),MCI 覆盖了音频、视频、影片、激光视盘机和录象机等多媒体的主要产品,能够满足大多数应用软件开发的需要。MCI 的最大优点之一是应用软件与设备无关,更换设备时只要更换 MCI driver,应用程序无需修改即可操作新设备。

#### 2) 图形设备接口(GDI)

要在窗口的客户区域绘制图形,可以使用 Windows 的图形设备接口函数。Windows 中的图形基本上是由 GDI.EXE 中的函数处理的。GDI.EXE 模块调用各种驱动程序文件中的例程。GDI 的一个主要目标就是在输出设备上支持与设备无关的图形。

#### 3) 图形用户界面(GUI)

随着计算机硬件的发展,鼠标的诞生,显示器分辨率的提高,软件系统的用户界面被一种图形用户接口占据了主要的位置,这种图形用户接口用图标等方式

形象地给出系统的意图及各种操作命令的功能,即 GUI。它的优点是操作简单,易于掌握。GUI 采用了面向对象的设计概念,把人机界面归纳为包括窗口、菜单、按钮、选择等类。一个应用程序的 GUI 部分实际上就是这些类的对象的不同组合。

#### 4) 对象的连接和嵌入(OLE)

对象的连接和嵌入(Object Linking and Embedding)是创建包含连接、嵌入对象的复合文本的过程,这些对象可以是相同类型的,也可以是不同类型的,声音对象、图象对象、图形对象、动画等。

此外,Windows 的动态链接库在多媒体软件的制作中担当相当重要的角色。它是一个将不同应用系统环境下开发的程序有机结合的关键桥梁。

### 2 多媒体软件开发的软、硬件配置方案

#### 1) 硬件方案

(1)声音部分:采用声霸卡,可完成语音/音乐的录制和播放功能,实现系统的语音解说或背景音乐的播放。

(2)图象部分:采用视霸卡或动态图象卡,完成视频图象的数字化播放,单帧捕获、连续捕获功能,使系统能检索所捕获的静态或动态图象。另外,采用图象扫描仪亦可获得精美的静态图象。

(3)动画部分:配以 387 协处理器或 486CPU,可用 3DS 三维动画软件或 Animator 二维动画软件。

(4)显示部分:采用真彩色卡或 TVGA 卡,设置  $640 \times 480 \times 256$  色或更高的分辨率、显示模式,以求逼真的显示效果。

(5)人机交互部分:可采用鼠标、触摸屏。

#### 2) 软件方案

(1)采用多媒体制作平台,直接、快速、简便地构造应用系统。

(2)采用微软公司的 Visual Basic。它功能强大,具有较强的编程可视性。

(3)采用 Asymetrix 公司的 Windows 应用生成工具 ToolBook。

(4)对于动态视频的捕获,采用微软公司提供的功能强大的视频图象动态捕获软件 Video for Windows,它可同时处理声音和图象。

### 3 软、硬件功能介绍

#### 1) Visual Basic

Visual Basic 是 Microsoft 公司推出的一种最新的优秀的 Windows 开发工具,被认为是用于开发 Windows 应用软件的 C 或 C++ 的最佳取代软件之一。它具有以下特点:

(1)采用事件驱动编程机制,即面向对象技术,这里的对象是真正可以看得见的,可以规定大小的东西,例如:按钮、标签等(称为物体)。

(2)Visual Basic 的物体都有许多属性,无论在设

计还是在执行状态,都可随时更改它,因而大大扩展了对象的范围和活动方式。

(3)易使用性。不需单独的编译过程,编辑完程序码后即可投入运行,具有简单而功能强大的函数和方法,使得程序短小、精悍。

(4)支持 Windows OLE(对象的连接和嵌入)和 MDI(多文档界面),利用 Windows 应用程序接口(API),结合 Visual Basic 最具威力的三个特色:DDE, DLL 和无限扩增的控制图,采用一些技巧,即可完成许多神奇的功能。

#### 2) ToolBook

ToolBook 是美国著名的多媒体软件公司 Asymetrix 于 1990 年推出的一个多媒体软件包。它提供类似于传统的图书形式的用户界面,可供软件应用人员建立自己的多媒体平台。这一功能强大的软件包可以实现如下功能:

- ①图、文、彩色动画、音频混成;
- ②支持动态数据交换,不支持连接与嵌入;
- ③极强的 Hypertext 超级链接功能;
- ④可支持声音和音乐等时序媒介,在 MRK(Multimedia Resource Kit)的配合下,允许对 MCI 的调用;

⑤可显示外界存储的位图文件,并嵌入到应用程序中,也可播放其它程序的动画;

⑥支持 DLL,通过对 105 个功能函数的调用,可进行视频、数据库和 DOS 下的操作。

3) Video for Windows

Video for Windows 是由微软公司推出的 Windows 环境下视频处理软件。其主要特点是:

①可播放和编辑动态视频、动画、多媒体幻灯片(含声音信号),不需要特定硬件设备的支持;

- ②可以同时处理音频和视频;
- ③可以在 VGA 显示器上工作,以支持真彩色;
- ④可实现 30 帧/S 的实时动态连续播放和捕获;
- ⑤支持 CD-ROM;
- ⑥支持大多数视霸卡。

#### 4) 声霸卡

当前市面上流行的声音卡,共有近 10 种。声霸卡支持 DOS 和 Windows 两种环境下的开发与应用,它具有如下特点:

①输入方式:Line-in(在线输入)

MicroPhoto(麦克风输入)

MIDI(数字乐器输入)

CD-ROM;

②输出方式:双声道立体声;其输出功率为每个声道 4 瓦;有混响效果;

③录、放音取样频率:4—44.1kHz;

可播放声音数据文件的格式:. VOC; . ORG; . WAV; . CMF 等。

支持 DOS 和 Windows 两种环境的应用与开发,具有声音的编辑功能。

利用声霸卡配套的系统软件就可以实现系统的语音解说或前景音乐播放(包括混音)功能等,若想自己开发,也有完整的开发工具。

#### 5) 视霸卡

它将模拟的视频信号采样,转化为数字信号,处理后变成模拟 RGB 信号,采用混叠技术(Over lay)与 PC 机文本信息在显示器上混合显示。

该产品可捕捉、保持、控制及输出激光视盘、录像机和摄像机的图象,并把视频数字化,在视窗环境下作全动态的数字化图象播放,为多媒体用户带来高层次的享受。其主要功能如下:

①能在一个可移动,可变尺寸的窗口中显示全活动的数字化视频信号;

②可接收来自录像机、激光视盘、摄像机和电视图

象信号在微机上播放、定格、存储、处理;

③计算机 VGA 输出的图形、文字可以加到视频图象上;

④视频尺寸可变;

⑤视频的色调、饱和度、亮度和对比度可以调节;

⑥内含数字化的立体声调音台,每个通道的音量及总音量均可采用程序控制。

支持的软件有:

Microsoft Windows 3.1

Action 215V

Temptra 图形图象处理及多媒体制作系统。

以上介绍了开发一个 WINDOWS 环境下的多媒体软件依据的硬件和软件系统,当然,根据所需的应用系统的特点,未必需要采纳以上介绍的全部工具,读者可以在了解了这些软、硬件工具后自行选择。

## 计算机多媒体演示系统的开发

崔大勇 聂发如 薛 智 毛静涛

(上海电力学院 200090)

**摘 要** 本文结合一个实际例子,介绍了多媒体演示系统的开发过程,讨论了利用 Visual Basic 在 Windows 环境下开发多媒体演示系统的特点及方法。针对应用中所涉及的一些问题,如:开发多窗体多任务的应用程序,用 MCI 控件播放视频音频文件等,提出了解决办法,给出了应用程序,并进行了说明。

**关键词** 多媒体 演示系统 计算机

### 1 前言

计算机多媒体技术的发展,使得计算机不仅可以处理文字,图像等,还可以处理动画,声音,视频图像等,这就进一步丰富了计算机的信息表现形式,为计算机的更进一步应用创造了有利条件。一个完善的多媒体演示系统不仅应该可以图文并茂,还应该具有音响及动画效果,而计算机多媒体技术的发展正为实现完善的多媒体演示提供了良好的技术基础。

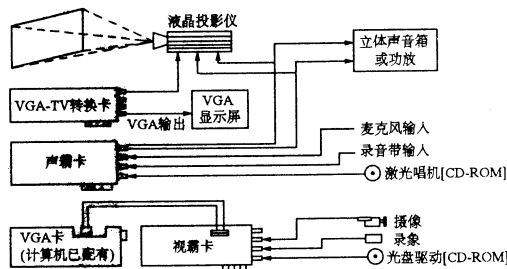


图1 学院介绍多媒体演示系统硬件配置

在本文中将以一个学院介绍的多媒体演示系统为

例,讨论计算机多媒体演示系统的开发。这里介绍的多媒体演示系统主要用于介绍学院的基本情况,或者作为介绍学院基本情况的辅助演示工具。于是,在该系统中应该能处理动画,声音,视频图像等信息,还应该能够图文并茂地查询有关的学院基本情况。

要能够实现以上的功能采用了图1所示的硬件配置,这是一种典型配置。

### 2 开发工具

Windows 的出现为多媒体技术提供了良好的环境,而怎样在 Windows 下开发多媒体演示系统是要首先解决的问题。Visual Basic 是目前开发 Windows 最方便和有效的工具,它具有以下特点:

- (1)可视化的设计工具;
- (2)面向对象设计思想;
- (3)事件驱动的编程机制;
- (4)结构化的高级程序设计语言;

(5)强大的多媒体支持功能,如 Visual Basic 3.0 专业版提供的 MCI 控件正是用于多媒体目的的用户控制部件,它实际上是一个高级的函数调用接口,可以把许多原本在 Windows API 中实现的功能和函数转



化为相应的属性和事件方法。而使用 MCI 控件可以读取各类媒体的素材文件,控制多种多媒体外设,如:音频卡,视频叠加卡,CD-ROM, VCR 等。

另外, Visual Basic 还具有使用动态数据交换 (DDE) 技术与其它 Windows 应用程序交换数据,通过动态连接数据库 (DLL) 技术调用 C 语言或汇编语言编写的程序以及 Windows API 函数,支持对象的链接与嵌入 (OLE) 技术,提供开发式数据库访问功能,可由用户自定义平台等特点,使得 Visual Basic 成为最方便和有效的 Windows 开发工具。于是,本系统采用 Visual Basic 作为开发工具。

### 3 开发中的几个问题

本计算机多媒体演示系统的主程序框架如图2所示,以下是对编程中的几个主要问题的说明。

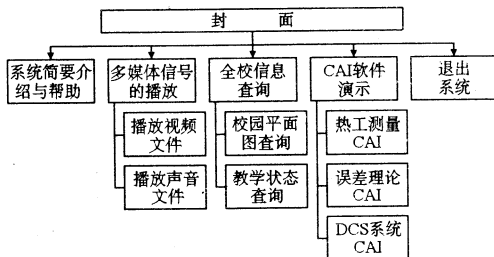


图2 学院介绍多媒体演示系统的主程序框架

#### 1) 主过程(main())过程

根据系统在 Option 菜单中 Project 选项中的设置,为了实现诸如动态绘图等的技巧,本程序设定为从 main() 过程开始运行。首先用 frmBegin. Show 语句把封面窗体 (Begin) 载入内存。

在 Begin 窗体中,有一个 Load 事件,当窗体被载入内存时就触发此事件。在 Load 事件中,把窗体中各个对象的 Visible 属性都设置为 False。这样,当 main() 过程用 frmBegin. Show 载入 Begin 窗体时,屏幕是一片空的区域,然后 main() 过程用 draw frmBegin 语句调用 draw 子程序,以实现动态绘图功能,即屏幕上会出现背景图案。

#### 2) 按钮及帮助模块

按钮可以有許多事件,系统中单击按钮产生的是 Button-Click 事件,而帮助模块则是通过单击 Begin 窗体上的相应按钮激活的,单击按钮产生 Button-Click 事件的程序如下:

```

Sub cmdintro-Click()
On Error Goto errexit
x=Shell("c:\siep\readme.exe")
Exit Sub
errexit:
Beep
MsgBox("文件c:\siep\readme.exe找不到"+Chr ¥(13)+"请重新安装本软件"),0
Exit Sub
  
```

End Sub

在以上程序中,先用 DIR ¥ 函数检测文件 c:\siep\readme.exe 是否存在,若不存在,用 MsgBox 显示一段出错信息并退出;若存在,则用 Shell 语句调用此可执行文件。

按钮的另一个重要事件为 GotFocus 事件。当按钮被选中时,触发 GotFocus 事件,在本系统中 GotFocus 程序为:

```

Sub cmdintro-GotFocus()
cmdintro.FontItalic=True
cmdintro.FontSize=13
cmdintro.ForeColor=&HC0&
Text1.Text="按钮1"
Text2.Text="按钮1导向的内容及按钮1使用方法"
  
```

#### 3) 播放视频影像

在系统中视频影像的播放是利用 Visual Basic 的 MCI.VBX 控件。VBX 文件与软件链接的动态库及普通动态链接库不同,VBX 控件在设计和运行时都可以使用。Visual Basic 3.0 中的 MCI.VBX 文件即一个高级的函数接口,它可以把许多原本在 Windows API 中实现的功能和函数转化为相应的属性和事件方法,这正体现了面向对象的程序设计思想。

利用 MCI.VBX 控件播放视频影像文件 (.AVI 文件) 的内容为:

- (1) MCI.VBX 控件的初始化;
- (2) 播放 AVI 文件 (单击 Play 键时的事件处理过程);
- (3) 暂停播放 AVI 文件;
- (4) 快进键的编程;
- (5) 快退键的编程;
- (6) 停止播放 AVI 文件;
- (7) StatusUpdate 事件;
- (8) Done 事件;
- (9) 静音 (Silent) 属性;
- (10) 滚动条的编程。

#### 4) 播放音频文件

音频文件的播放也是由 MCI.VBX 文件实现的。与播放 AVI 文件不同的是,必须把 DeviceType 属性定义为 "WaveAudio", FileName 属性定义为需要播放的 WAV 文件名。

#### 5) 基于超文本技术的查询系统

Windows 的帮助文件正是一种超文本文件,超文本技术可以实现非线性浏览,因此,可以用于开发查询系统。另外,Windows 的帮助文件还可以容纳图片以及视频影像,以实现丰富的查询系统。

从编程的角度看,Windows 帮助系统的开发应包括以下几个内容:

- (1) 编辑步骤专题源文件,它又包括:

- A. 指定热点;
- B. 指定专题名,关键字和浏览序号;
- C. 使用图形和图像;
- D. 创建不可滚动的标题区域。

(2)建立帮助项目工程文件;

(3)帮助源文件的编译。

#### 4 结论

在上面讨论了计算机多媒体演示系统的开发过程,关键问题和解决办法。在实践中,利用以上办法开发了一个图1所示的学院介绍多媒体演示系统,取得了

满意的效果。在开发中发现,尽管 Visual Basic 不是专用的多媒体开发工具,但是它有自己的特色,特别是对不太庞大的多媒体系统,利用 Visual Basic 开发多媒体演示系统具有方便和简单的特点。

#### 参考文献

- 1 Gary Cornell, The Visual Basic 3 for Windows Handbook, 学苑出版社, 1994. 8
- 2 杨学良,《多媒体计算机技术及其应用》,电子工业出版社, 1995. 2
- 3 胡晓峰,《多媒体系统原理与应用》,人民邮电出版社, 1995. 6

## 在 ORACLE \* Graphics 中使用图象和声音的技术

穆 斌

(合肥工业大学计算中心 230009)

**摘 要** 本文首先对 ORACLE \* Graphics 作一简略介绍。接着叙述了进口图象和出口图形的方法。最后对使用声音的技术作了较详细的阐述。

**关键词** 图象 声音 ORACLE

#### 1 引言

ORACLE \* Graphics2. 5是客户机/服务器结构下图形化数据库界面开发工具。它的突出特点是广泛采用了面向对象的技术。利用它,可以方便快捷地开发出 Windows 风格的数据库应用。

ORACLE \* Graphics 运行于客户机之上,其操作系统平台是 MS Windows。它向服务器端的 DBMS 发查询请求以获取数据。查询获得的数据将用图形化的方式表达出来。用于表达的对象可以是各种动态图表(如棒条图、圆饼图、折线图等),也可以是线型图形(如矩形、圆、弧、多边形、自由手绘图形)等。ORACLE \* Graphics 提供了丰富的工具和手段以制作这些图形化的对象。

ORACLE \* Graphics 设计系统有三个主要的工作场所。其一是布局编辑器(Layout Editor),它是最重要的可视对象设计场所,也是图表、图形等显示和居留的地方。其二是对象浏览器(Object Navigator),它是管理对象的场所。其三是 PL/SQL 编辑器(PL/SQL Editor),它是编辑、编译、运行 PL/SQL 程序的地方。这些程序作为交互式设计的必要补充,为 ORACLE \* Graphics 增添了灵活的过程化逻辑功能。

除了上述基本功能之外,ORACLE \* Graphics 还提供了与外部图象和声音的接口。下面我们就来介绍

这方面的使用技术。

#### 2 图象进口和图形出口技术

##### (1)进口图象

ORACLE \* Graphics2. 5作为一个图形化数据库开发工具,本身不支持图象的制作功能。但可以将其它多媒体软件(如3DS等)制作的图象文件进口到 ORACLE \* Graphics 中来。这里有一个图象文件 road. bmp,是合肥市道路网的格局地图。现在我们要把它进口到 ORACLE \* Graphics 中来。

在布局编辑器中用 Edit→Import→Image 打开进口图象对话框。选择 File radio-button,这表明图象来源于一个文件。在相应的栏目下打入该文件的路径和名字,例如:C:\city. road. bmp。在 Format 栏下选择图象格式类型,例如选 BMP 格式。并在 Quality 栏下选择图象品质,比如选 Good。最后按图符按钮 OK 即可将这幅地图进口到 ORACLE \* Graphics 中来。进口来的图象位于布局编辑器的左上角,且具有原先图象的大小尺寸。可以通过图象对象四周的控制点进行 X、Y 方向任意比例的缩放。最后可将其移位到布局编辑器上的目标位置。

同样可以利用 PL/SQL 编制一个过程化的进口程序来实施图象进口。下面的程序具有与上述交互操作等价的功能。

```
PROCEDURE import _ the _ image IS
the _ image OG _ OBJECT;
the _ layer OG _ LAYER;
BEGIN
```

```

the_layer:=OG_GET_LAYER('my_layer');
OG_ACTIVATE_LAYER(the_layer);
the_image:=OG_IMPORT_IMAGE('C:\city\road.
bmp',
    OG_FILESYSTEM,OG_BMP_IFORMAT);
END;

```

### (2) 出口图形

我们已经知道,ORACLE \* Graphics 可以根据数据库中的数据生成各种美观的图表,如圆饼图、棒条图等。反过来也可以将这些显示图表出口到图形文件中,以供其它软件调用。

在 ORACLE \* Graphics 布局编辑器中用 Edit→Export→Drawing 打开出口图形对话框。选 File radio button,表示出口的图形将存放在文件系统中。在相应的栏目下打入目标文件的路径和名字,例如,C:\city\chart.CGM。在 Format 栏下选择图形格式类型,比如选 CGM 格式。并在 CGM Encoding 栏下选取 CGM 编码方式,例如选 16bit 编码。最后撤图符按钮 OK,即可将 ORACLE \* Graphics 制作生成的图表存到文件 chart.CGM 中去。这以后,便可以在其它软件中使用这个图形文件了。

同样可以编写 PL/SQL 图形出口程序。下面的程序段与上述交互操作功能等价。

```

PROCEDURE export_the_drawing IS
the_layer OG_LAYER;
BEGIN
the_layer:=OG_GET_LAYER('my_layer');
OG_EXPORT_DRAWING('C:\city\chart.CGM',
    OG_FILESYSTEM,OG_CGM16_DFORMAT,the_
layer);
END;

```

## 3 声音使用技术

假若所配置的客户机支持声音功能(比如安装有声霸卡),就可以在 ORACLE \* Graphics 中利用声音媒介与用户交互。例如,当数据库中某些数据值过高或过低超过限制时,可以用声音作媒介播放警告信息以提醒用户注意。

### (1) 录音

在对象浏览器中选择 Sounds 结点,用 Navigator→Create 创建声音对象 Warning,并打开它的声音对话框。撤下 Record 按钮即可开始录音,用麦克风录下语音、音乐、或任何其它声音,再撤下 stop 按钮即完成录音过程。最后用 OK 按钮关闭对话框。这以后,便可以用 OG\_PLAY\_SOUND 内部过程来访问并播放这个声音对象了。

利用 PL/SQL 编程,甚至可以在应用系统运行时,由最终用户现场录音。下面的程序在应用系统执行时打开声音对话框,并让用户自己录制声音。

```

PROCEDURE record_warning IS
warn_sound OG_SOUND;
BEGIN

```

```

warn_sound:=OG_GET_SOUND('warning');
IF NOT OG_ISNULL(warn_sound)THEN
OG_RECORD_SOUND(warn_sound);
END IF;
END;

```

### (2) 进口声音

如果已经存在一个外部的声音文件,则可以将它进口到 ORACLE \* Graphics 中来。

选择 Edit→Import→Sound 以打开进口声音对话框。选 File radio button,这表明声音源是文件系统。在其后的栏目下打入源声音文件,例如 C:\windows\ding.wav。并在 Format 栏下选择适当的声音文件格式,比如 WAVE—PCM WAVE。最后撤 OK 按钮即可将声音进口到 ORACLE \* Graphics 中。这之后,可以用 OG\_PLAY\_SOUND 内部过程来调用播放这个声音。

### (3) 播放和编辑声音

在对象浏览器中双击一个声音结点即可打开它的声音对话框。现在,可以用 Forward 和 Rewind 按钮快进和快倒,也可以用滑动条进行定位。Record 按钮用于录制新的声音。反复进行定位——录制操作,即可完成声音的编辑修改。最后用 Play 按钮播放试听。

当然,最通常的方法还是利用 PL/SQL 编制程序以实现声音的适时播放。例如:

```

PROCEDURE Play_warning IS
the_sound OG_SOUND;
BEGIN
the_sound:=OG_GET_SOUND('warning');
OG_PLAY_SOUND(the_sound);
END;

```

### (4) 出口声音

在 ORACLE \* Graphics 中制作的声音可以被出口到声音文件中,以供其它软件使用。

在对象浏览器中选择要出口的声音对象,比如选择 warning。用 Edit→Export→Sound 打开声音出口对话框。选择 File radio button,并打入目标文件名,例如 C:\city\warning.wav。在 Format 栏下选择适当的文件格式,例如 WAVE—PCM WAVE。并选择适当的编码方式、数据类型和频率(通常用 Auto 选项)以及压缩方法。最后撤 OK 按钮即可完成声音的出口。

## 4 结束语

总之,通过在 ORACLE \* Graphics 中使用图象、图形和声音,可以进一步强化 ORACLE \* Graphics 自身的图表功能。由于开发出的界面具有多媒体特征,数据库应用也就显得更加生动和丰富。

### 参考文献

- 1 ORACLE Graphics 2.5 Developer's Guide Manual, ORACLE Corporation, 1995
- 2 ORACLE Graphics 2.5 Reference Manual ORACLE Corporation, 1995
- 3 PL/SQL User's Guide and Reference ORACLE Corporation, 1995

# RIFF 音频文件格式及其波形输出与应用

梁 雯

刘松林

(安徽大学工商管理系 合肥230039)

(安徽大学电子工程与信息科学系 合肥230039)

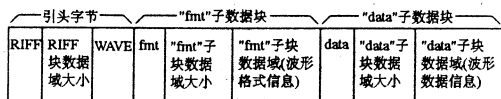
**摘 要** 本文详细介绍了多媒体声霸卡音频文件的存储格式及格式信息的提取与语音波形恢复方法和应用。

**关键词** 多媒体 声霸卡 RIFF 音频文件格式 波形还原

随着多媒体技术的发展,声霸卡也发展到现在的拥有 DSP 和 ASP 采样频率达44.1KHz 和16位分辨率,及32位总线结构的功能强大的设备,它具有:(1)录制、编辑和回放数字声音文件。包括对音频信号采样、量化、文件生成等操作。(2)控制声源,进行音频混合后数字化(通常用 Mixer 程序来控制卡上的混合器)。(3)在记录和回放声音文件时通过 DSP 对音频信号进行压缩和解压缩。(4)具有 MIDI 接口(乐器数字接口),在计算机上可控制多台带 MIDI 接口的电子乐器进行 MIDI 音乐 MID 文件存放、压缩、编辑和回放,(如用 Midiu Player 可在计算机上演奏和作曲)。(5)还具有语音合成和语音识别功能,让计算机朗读文本,让计算机接受用户语音指令等。其中音频信息是经过对音频信号的采样、量化再编码压缩,最后以文件形式存储在计算机硬盘或其它存储媒体上的。不同声卡和软件驱动程序录制的语音文件格式可能不同,但通常它们之间可相互转换。如 Creative Labs 用 VOC 作文件扩展名,而在 MicroSoft 的 Windows 下则以 WAV 为扩展名。借助于 Windows 中 Convert(数据文件格式转换工具)可将 Windows 不支持的格式转换成支持的格式。Windows 3.1 下的 Sound recorder 程序能把不同声源的语音数字化后合成为 WAV 文件,然后对其进行简单的剪裁、粘贴,还能加入回音。WAV 文件通过 OLE (Object Linking and Embedding)还可加入到其它的应用程序中去。

我们可以通过程序获取音频文件的格式信息,并输出波形进行语音分析和利用。

## 1 RIFF 音频文件格式



多媒体支持一种称作资源交换文件格式,简称 RIFF 格式。其格式类型为"WAVE",扩展名为 WAV。WAVE 文件至少由两个子数据块构成,一个是"fmt"

子数据块,用于存储波形格式信息,一个是"data"子数据块,用于存储波形数据信息。其文件格式如上:

1)引头字节包括:

0~3字节:CKID 数据块 ID 的四字符代码,这里是 RIFF;

4~7字节:Cksize 指定数据块中数据域大小的"双字";

8~11字节:四字符代码格式类型"WAVE";

12~15字节:四字符代码,波形格式信息"fmt";

16~19字节:过渡字节(一般为10H00H00H00H)。

2)"fmt"子数据块数据域记录波形格式信息。由两段组成,一段为[Wave-format];另一段为[format-Specific],两个数据段的数据结构定义如下:

①[Wave-format]的数据结构,它由5个字段组成。

[Wave-format]→struct {

WORD wFormat Tag; (20~21字节):指定 WAVE 格式种类

WORD nChannels;; (22~23字节):指定采用的声道数

DWORD nSample Per Sec; (24~27字节):指定采样频率

DWORD nAvg Bytes Per Sec; (28~31字节):指定每秒平均字节数

WORD nBlock Align; (32~33字节):data 块中数据的块调整

}

②[format-specific]数据结构,对于 PCM 格式,结构中仅包含一个字段。

[PCM-specific]→Struct {

WORD nBits Per Sample;指定 WAVE 文件样本值位数,占第34~35字节

}

3)"data"子数据块。

36~39字节:"data"子数据块标志四字符代码 data;

40~43字节:"data"子数据域大小;

44~...: 存放波形数据信息。

44字节以后存放的波形数据信息,其存放格式依赖于"fmt"子数据块中的"wFormat Tag"字段指出的格式种类。对于 PCM 方式的 WAVE 文件,其格式如下:

(下转45页)

# 从C++到Java

王 舒

(四川联合大学 成都610017)

**摘 要** 本文从分析Java与C++的主要区别着手,介绍了Java取消了C++中的某些特性和功能及其对C++功能的改进,并举一简单实例予以说明,以便熟悉C++的读者快速掌握Java程序设计。

**关键词** 声明 类 引入 界面 包 类库

Java来源于C++,又不同于C++,它比C++具有更加完善的面向对象性和安全性。作为网络语言,Java具有强大的Internet网络功能,同时又可像传统编程语言一样制作单独的应用程序,为使用者提供了一个良好的开发和运行环境。由于Java在语言风格和结构上与C++非常相似,对于熟悉C++的人来说,从了解Java与C++的区别入手,是一条快速掌握Java的捷径,可以达到事半功倍的目的。

## 1 Java取消的C++的特性和功能

Java取消了C++中难以理解、不易掌握和极少使用的语言成分,以及容易引起系统不稳定和不安全的因素。Java所取消的C++中的一些特性和功能具体如下:

- \* 结构和联合 结构和联合被Java中的类所取代,类包含了结构和联合的作用。

- \* 指针 Java取消了构造类型,并把数组封装到了对象中。

- \* #define 语句

- \* 多重继承 多重继承被Java中的接口所取代,接口是一个类的方法和变量,而并非这个类的全部结构

- \* 独立函数 Java中的函数都必须封装到某个类中。

- \* goto 语句 Java中代之以带有标号的Break和Continue语句以及异常(Exception)处理。

- \* 运算符重载

- \* 自动类型转换 Java中数据类型的转换必须显式实现。

## 2 Java对C++的改进及新增的功能

Java并非是C++的扩充,它是一种全新的完全面向对象的语言。Java对C++的改进及新增的功能反映在如下几个方面。

### 1) Java的数据结构

Java采用16位的Unicode字符集,而非ASCII码。数据类型的定义不依赖具体的系统,统一成一种标

准格式,如整数(int)在任何计算机上都是32位,浮点数则采用IEEE754标准的32位浮点数。Java的变量有两种类型,一种是简单类型(如整数类型、浮点数类型、布尔类型等),即基本类型;一种是复合类型,它们可以建立在任何简单类型的基础上,组成数组、类和接口等。变量的作用范围是从声明它的那一点开始,一直到声明它的块(一个块就是一对{}所包含的程序段)结束为止。当一个块中的某个变量又在其内部的某个块中再次声明,那么在外块中定义的变量就被掩盖起来暂时不用。

Java中字符串是通过字符串(String)类来实现的,而不象C++中通过字符数组来实现。字符串类可分为不变字符串类String和可变字符串类StringBuffer两种类型,如:

```
String OutputString=new String("Hello!");
```

```
StringBuffer varStr=new StringBuffer();
```

前者定义了一个字符串常量“Hello!”,后者定义了一个字符串变量。

此外,Java中的布尔型完全是一种单独的类型,它不能被表示成整数0或1。

在Java中建立数组时,数组的下标个数必须显式地使用new操作符来建立。例如:

```
int a[]=new int[10]
```

建立了一个含有10个整型数的数组,并把它赋给数组变量a。

### 2) Java程序代码的结构

Java程序至少由一个或多个编译单位组成,编译单位是一个存放源代码的文件,其扩展名为.java。每个编译单位包含四种不同成份的组合:类(class)声明、引入(import)语句、界面(interface)声明和包(package)语句。

Java中所有的类都是从超类Object(对象)继承而来的。换言之,Java的类是Object的直接子类或间接子类。如果没有显式的声明某个类是从哪个类中继承而来的,那么它将被认为是Object类的直接子类。

类声明的一般形式为:

```
[import<包名>.<超类名>]
```

```
class<类名>[extends<超类名>]{
    <实例变量声明>
    <方法声明>
}
```

其中,实例变量用来描述对象的属性或状态;方法则用来描述对实例变量的操作,为其它对象对本对象的实例变量的操作提供一个外部接口。〈实例变量声明〉的形式为:

```
[<修饰符>]<类型><实例变量名>;
<方法声明>的形式为:
[<修饰符>]<方法名>(<参数>){
    <方法体>
}
```

每个方法都必须返回一个值或者将返回值声明为空值 void。

实例变量和方法前的〈修饰符〉可以是:

public:表示它所修饰的实例变量和方法对所有的类来说都是可访问的。

protected:表示只有声明这些实例变量和方法的这个类才能使用。

private:表示它所修饰的实例变量和方法只能被这些变量和方法所属的类及其导出子类所访问。

friendly:表示它所修饰的实例变量和方法只能被同一个包中的类所访问。

除了上面的四种修饰符外,还可以把一个实例变量或方法声明成静态的 static,它表示一个实例变量或方法对它所属的类的所有对象来说都是相同的,可以共享。

下程序定义了一个简单的 Java 的类,其功能是在屏幕上输出一句问候语:

```
How are you!
class Greet {
    public static void main(String args[]){
        System.out.println("How are you!");
    }
}
```

Java 允许方法重载,即在多次声明同一方法时采用不同的参数。而要想覆盖一个方法,则必须在最初声明这个方法的类的子类中声明一个与它具有相同名字、相同返回类型以及相同参数表的方法。

用类声明为对象建立了一个模板后,如果想使用这个新对象就必须用构造函数来创建它。构造函数是类的一个特殊的方法,它的函数名与类名相同,可以有任意类型的参数但不具备返回类型。

Java 的界面定义了一组方法,但没有方法的具体实现。这样,几个类可以共享一个界面,而不必担心其他类怎样处理这些方法。在一个类中可以通过 implements 保留字来实现界面:

```
public class<类名>[extends<超类名>]implements
<界面名>[,...]{
    ...
}
```

Java 中的包由经常放在一起、具有相同的某种特性或组合起来用以完成某一功能的一组类和界面组成。同一个包中的类可以互相存取对方的实例变量和方法。在编译单位中的包由 package 语句定义:

```
package<包名>;
```

若编译单位中无 package 语句,则该编译单位将被置于一个缺省的无名包中。

### 3) Java 类库

C++中提供了大量的系统函数,而 Java 则在类库中提供了更强的功能。类库以包的形式存放,可以用引入这些包的方法来引入相应的类和界面。Java 提供的类库可以分为两个大类:Java 类库和 HotJava 类库。Java 类库为 Java 低层系统方法调用提供了三个包,它们是:

java.lang 包(基本类库):提供用于为基本程序功能封装方法的低层类。如:基本对象类、类型封装类、基本数学运算类、字符串类、系统类等。

java.io 包(输入/输出类库):如随机存取文件类、标准输入/输出类等。

java.util 包(实用程序类库):如堆栈类、日期类等。

可以用如下方式:

```
import java. *;
```

立即引入这些类库。

HotJava 类库不仅能够用于 HotJava 浏览器中,也可用于单独运行的环境或其他浏览器环境。它主要包括三个包:

java.awt 包(另一个窗口工具箱类库):提供用于创建图形用户接口运算的类。

java.browser 包(浏览器类库):提供用于浏览器环境以及浏览器环境中的应用程序的外貌特征的方法。

java.net 包(网络类库):提供以网络协议(如 Telnet、FTP、WWW 等)接口方式扩展 Java 的 I/O 类的功能的类。

### 4) Java 的多线程机制

Java 体系结构在多级上融入了多线程的特征,即在编程语言环境有实现线程化的内置修饰符,以及在即时运行环境(解释器中或浏览器中)包含线程化的自动垃圾回收器。

创建可实现多线程的类的方法有如下两种:

\* 创建一个 Tread(线程)类的继承类:

```
class<线程类名>extends Thread{
    ...
}
```



```
void run() {
    ... //方法体
}
```

其中,run()方法用于完成线程类的主体功能。如果要调用定义的线程,就必须创建该类的一个实例,其形式为:

```
<线程类名>(<实例名>)=new<线程类名>();
```

然后用 start()方法启动该线程的运行:

```
<实例名>.start();
```

这样该线程将独自运行,控制返回调用该线程的原线程(主线程)继续运行。

\* 创建一个实现 Runnable(可运行的)界面的类:

```
class<类名>implements Runnable {
    ...
    public void run() {
        ... //方法体
    }
}
```

其中,Runnable 界面的定义为:

```
package java.lang;
public interface Runnable {
    public abstract void run();
}
```

实现了 Runnable 界面的一个类,可以把该类的实例传递到一个新线程类对象的方法里,并在该线程中运行。其传递方法如下:

```
<类名>(<实例名>)=new<类名>;
new Thread(<实例名>).start();
```

### 5) Java 的安全保障机制

Java 的安全机制分为多级,包括 Java 语言本身的安全设计以及严格的编译检查和运行时检查。

Java 对内存访问进行了严格的控制;

\* 它不支持 C++ 中的指针,而代之以真正的数组和串,从而不再有意无意地访问任意的内存空间,造成用户数据区甚至操作系统内存区域的破坏。

\* 它提供了严格的数组下标检查机制,使用户无法用其他途径构造出指针。

\* 它不支持数据类型的隐式转换,杜绝了隐式指针的可能性。

\* 它还提供了自动内存管理机制,即一个自动的“垃圾”搜集程序。

Java 的内存布局不是由编译器分配,而是推迟到运行时再由 Java 解释器决定。对编程人员而言,Java 的内存分配和引用模型是透明的,它完全由底层的运行系统控制。

Java 运行环境提供的安全保障机制:

\* 字节码校验器 它用以确保代码段不存在伪造的指针、不违反访问权限以及严格遵循对象规范来访问对象。

问对象。

\* 字节码装载器 它把各个类装载到各自的内存区域。本地类只能显式地引用外来的类,而从其他结点传送过来的类不得通过任何途径对本地类进行操作。

\* 文件访问限制 客户端管理员可以限制从网络上装载的类的访问权限。

### 3 Java 小应用程序(applet)举例

为使读者进一步掌握 Java 与 C++ 之间的区别,并对 Java 编程有个全面深入的了解,试分析一个用于 HotJava 浏览器的 Java 小应用程序(applet)的简单例子。该 applet 的功能是以图像切换的方式循环显示“欢迎使用 Java”及“Welcome to using Java”。

该 applet 的源程序如下:

```
import browser.* //引入浏览器包中所有的类
import awt.*; //引入用于制作图形界面的 awt 包中的所有类和界面

class WelcomeTo extends Applet implements Runnable{
    //声明一个 Applet 类的子类 WelcomeTo,并放到一个线程中运行
    int i;
    int times=0;
    AudioData sound=null;
    Thread animationThread=null; //定义一个循环显示图像的线程
    Image frames[]=new Image[11]; //frames 用于存放循环显示的图像
    Protected void init(){
        resize(200,200); //调整窗口大小
        for (i=0;i<=10;i++) {
            frames[i]=getImage("doc:/demo/images/welcome"+
                "T"+i+".gif");
            //从外存的指定目录调入图像,图像名为 Ti.gif
            sound = getAudioData ("doc:/dome/audio/welcome.au");
            //调入欢迎音乐
        }
    }
    protected void start(){//启动循环显示线程
        animationThread=new Thread(this);
        //this 表示对一个实际对象自身的引用
        animationThread.start();
    }
    public void run(){//循环显示线程体
        int temp=0;
        boolean judge=true;
        play(sound);//播放欢迎音乐
        while(judge){
            for(i=0;i<=10;i++)repaint();//更新显示图像
            for (int j=0;j<=10000;j++)temp=temp+1;
```

```
//循环显示之间的短暂时延
}
}
public void paint(Graphics g){
g.drawImage(frames[1],0,0);//画出图像
}
}
```

用 java c 编译器对上述源程序进行编译,生成 WelcomeTo.class 字节码文件,再通过如下 HTML 程序来调用上面的 Applet:

```
<html>
<head>
<title>An Example</title>
</head>
<body bgcolor = # 0000110 text = # f111111 link = #
ff00ff>
<center>
<p>
<app class = "WelcomeTo">
<p>
Click<A HREF = "c:\windows95\javant\java\demo\
```

```
classes\WelcomeTo.java">
here</A>to see the source.
<p>
</body>
</html>
```

设上述 HTML 文档的名称是 welcome.html,所在目录为 C:\javant\java\demo,在 HotJava 浏览器中的 Document URL:处输入:

file:///c:/javant/java/demo/welcome.htm] 便可以看到所要显示的 Applet 了。

### 参考文献

- 1 Barry Boone, Java Essentials for C and C++ Programmers, Addison-Wesley Developers Press, 1996
- 2 Java Primer Plus, Waite Group Press, 1996
- 3 陈锡明、杨国纬, Java 程序设计及应用, 四川大学出版社, 1996. 11
- 4 麦中凡等, C/C++ 程序员 Java 编程, 清华大学出版社, 1996. 10
- 5 鼎点工作室, 实用 Java 编程, 西南交通大学出版社, 1996. 9

## PowerBuilder 与 ORACLE 的接口

肖江文 盛翊智

(华中理工大学自动控制工程系 武汉430074)

**摘 要** 本文阐述了 PowerBuilder 与 Oracle 数据库之间的 ODBC 接口及专用接口。

**关键词** PowerBuilder Oracle 数据库 ODBC 接口 专用接口

PowerBuilder 是功能强大、性能优异的数据库开发工具,它是按照客户/服务器(Client/Server,简称 C/S)体系结构研制的开发系统,具有以下特点:专业的 C/S 开发工具;面向对象的快速开发;同时支持多种关系数据库管理系统;内置的关系数据库;智能化的数据窗口对象;丰富的数据源和多种多样的数据显示格式;支持多种商业图形;有机结合的集成开发环境;完全支持 Windows 的窗口信息和控制;强有力的编程语言;支持多种平台;完整的开发工具包。

PowerBuilder5.0 是 Powersoft 公司于 1996 年 5 月发布的,较之以前的版本有质的飞跃,主要反映在:PowerBuilder5.0 从 4.0 的 P-Code 编译变成了完全的机器码编译,使应用的运行速度比 4.0 以前的版本提高了 2~50 倍;分布的 PowerBuilder;扩展了支持 OLE2.0 的能力;界面风格与 Windows95 完全一致;数据窗口增加了 OLE2.0 和 RTF 格式显示风格;可以快速浏览

对象的高级的对象浏览器等。该版本已无可争辩地成为了事实上的工业标准,每个致力于在大型数据库开发领域有所作为的开发人员都应该掌握这一标准。

PowerBuilder 是专为各种数据库设计的客户端的应用开发工具,它可以和其它数据库服务器(如 Sybase、Oracle、Informix 及 Microsoft SQL Server 等)构成客户/服务器体系结构。针对不同数据库,PowerBuilder 都有专用的接口,用以高速存储各数据库中的数据,并可充分发挥每种数据库的特长;PowerBuilder 还支持 ODBC(开放数据库互连)接口,可读取如 dBASE、FoxPro 等格式文件,充分利用各种不同的数据库中的数据。

Oracle 是高性能的大型关系数据库管理系统。早在 1979 年 Oracle 公司就为小型机(VAX)推出了基于 SQL 语言的商品化的关系数据库管理系统 Oracle,至今已发展到第七版,并且适用于多种大型、小型及微型机系统。Oracle 还在 DBMS 外层提供了一套完整的软件生产工具,大大提高了使用性能。

Oracle 与其他关系数据库相比具备的特点:

(1)兼容性:Oracle 关系数据库管理系统与 IBM 的 SQL/DS 和 DB2 完全兼容,其功能和用户接口与 SQL/DS 及 DB2 完全一致。为此,为 DB2 及 SQL/DS 编写的程序无需进行修改就可在 Oracle 上运行。

(2)广泛的应用环境:Oracle 是用 C 语言编写的,对机器的依赖性低。用于大、小、微型机上的各种版本都是相同的。

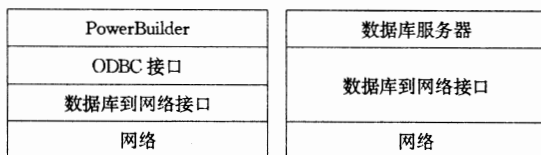
(3)Oracle 数据库共享:大型机(或小型机)与微型机上的 Oracle 数据库共享。

鉴于 PowerBuilder 作为客户端应用开发工具所具有的强大功能,及 Oracle 作为一种高性能的大型关系数据库管理系统所具有的上述优点,研究 PowerBuilder 与 Oracle 的 ODBC 接口或专用接口就显得极为重要。

所谓 ODBC,即开放数据库互连(Open DataBase Connectivity)标准接口,它是 Microsoft 公司 WOSA (Windows Open Service Architecture)的有机组成部分。ODBC 接口以 SQL 作为标准的查询语言来存取连接到的数据源。ODBC 允许单个应用存取多个不同的数据库管理系统,这使得应用开发者可以开发、编辑和发行应用而不管它操作的数据库管理系统究竟是什么。用户可以通过加载连接到不同数据库的驱动程序来建立与各种数据库的连接。

ODBC 调用分为四个层次:ODBC 应用软件(通过调用 ODBC 函数给数据库发送 SQL 语句并处理 SQL 返回结果)、驱动管理软件(管理驱动程序和装卸驱动程序)、驱动软件(处理 ODBC 函数调用,提交 SQL 请求到特定的数据源并返回结果及应用)、数据源(要存取的数据及其相关的操作系统、数据库管理系统和网络系统)。

PowerBuilder ODBC 接口的实现机理如下图所示:



PowerBuilder 通过 ODBC 接口连接到数据库的步骤如下:

(1)安装并调通网络。

(2)安装数据库,使数据库服务器自身能正常运行。

(3)安装该数据库的 ODBC 软件(从 PowerBuilder 的 Development Kit 盘安装,或从其它的软件包中寻找)。

(4)在 PowerBuilder 中配置 ODBC,设置数据源。

具体到 PowerBuilder 与 Oracle 的 ODBC 接口,连接步骤如下:

a. 安装网络软件(如 FTP TCP/IP),调通与数据库服务器的连接。

b. 把网络支持 Windows 的动态链接库拷贝到 PowerBuilder 所在的目录下,或者在 autoexec. bat 文件中指定网络软件的运行路径。

c. 安装 Oracle 的 SQL \* Net 软件(如 SQL \* Net for FTP TCP/IP)。安装完后,Oracle 会自动在 Windows 目录下产生其配置文件 Oracle. ini。下面为 Oracle. ini 的内容示例(设 Oracle 本身的安装路径为 c:\oracle7,FTP TCP/IP 的安装路径为 c:\pctcp):

```
[Oracle]
ORACLE_HOME=C:\ORACLE7
NLS_LANG=AMERICAN_AMERICA.WE8ISO8859P1
ORAINST=C:\ORACLE7\DBS
RDBMS70=C:\ORACLE7\RDBMS70
PRO15=C:\ORACLE7\PRO15
TCP_VENDOR=FTP
TCP_SERVICES_FILE=C:\PCTCP\SERVICES
```

d. 如使用 TCP/IP 网络,在 c:\pctcp 目录下产生 services 文件,内容为 orasrv 1525/tcp

e. 将 windows 目录下的 oracle. ini 文件拷贝至 oracle7 目录下并更名为 config. ora,然后将新文件的第一行[Oracle]去掉。另在 autoexec. bat 文件中加入如下两行内容:

```
set config=c:\oracle7\config. ora
:\oracle7\bin\mftp (如果使用了 SQL * Net 1. x for FTP TCP/IP)
```

f. 假设数据库服务器的 IP 地址为 202. 114. 108. 153,Oracle 的 SID 为 orac,则在 config. ora 和 oracle. int 文件中加入一行:

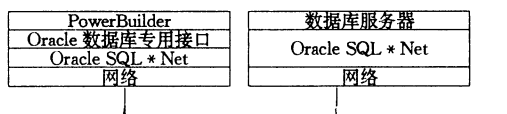
```
LOCAL=T:202. 114. 108. 153:orac
```

g. 运行 PowerBuilder,在 PowerBar 上选取 ODBC 配置画笔,然后在 ODBC 配置窗口(Configure ODBC)中选择 ODBC 驱动为 SDP Oracle7,再单击“Create”按钮,产生 ODBC 数据源定义 myoracle。

h. 在 PowerBar 上选取“Database Profile”画笔,窗口中将列出已定义好的 ODBC 数据源,此时选中 my-oracle,然后单击“OK”按钮返回,此后 PowerBuilder 中的所有操作都将针对 myoracle 数据源了。

事实上,由于 ODBC 是通用的数据库接口,调用层次多,因此通过这种接口操作数据库的效率很低。为此,PowerBuilder 还提供了到数据库的专用接口。这种接口的特点是调用直接,速度快,并充分发掘了特定数据库的特色,专家建议尽量使用这种专用接口。

PowerBuilder 到 Oracle 专用接口的实现机理如下图所示:



PowerBuilder 连接到 Oracle7 客户端的安装步骤中,前面六个步骤与使用 ODBC 接口时的步骤(从 a 到 f)相同,其特有步骤为:

g. 安装 PowerBuilder 的专用接口盘 Development Kit。

h. 将 pb5dk 目录下安装好的接口程序全部拷贝到 pb5 目录下,或者在 autoexec. bat 文件中指定 pb5dk 的运行路径。

i. 参数设置 (Vendors/or7, Logid/system, Log-Password/manager, ServerName/@T; scosysv; orac, DBMS/oracle)

j. 用 SQL \* Plus 安装运行 PowerBuilder 时要用到的数据字典:

sqlplus system/manager @ pbor7cat. sql

## 多文档界面在雷达仿真软件中的应用

张 海 潘光斌

(中物院电子工程研究所 成都610003)

**摘 要** 本文结合雷达仿真软件的开发实践介绍了多文档环境的应用和实现的关键技术。

**关键词** 多文档 Windows 应用程序 仿真软件

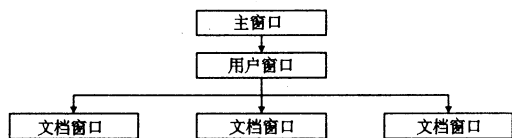
### 1 前言

在众多的商用软件中,我们几乎无一例外地会发现多文档界面的广泛应用。通过多文档界面及其他 WINDOWS 应用程序提供的特点,软件摆脱了过去在 DOS 环境下枯燥无味的缺点,展现给软件使用者一个接近自然的操作环境,从而使人们逐步放弃了过去一直使用的 DOS 环境而转入了 WINDOWS 环境。由此,它启发了软件开发者,为了使所开发的软件具有友好的用户界面,多文档技术是不可缺少的。

我们在研制雷达仿真软件的过程中,充分考虑到多文档界面的重要性,在设计上,根据数据的不同特点将其送到不同的窗口去显示,这样不但可以比较不同窗口中的数据,而且还可以开设新的窗口用于数据的处理,除此之外,所有窗口中的数据都可以与 WINDOWS 环境下的其他应用软件共享数据,整个过程极其方便、自然,下面就较为详细地介绍一下多文档界面的技术实现过程。

### 2 多文档窗口的组成和应用

顾名思义,多文档窗口是由多个窗口组成的,所以为了理解多文档应用程序,必须明确它的组成窗口。



一般来说,多文档窗口应用程序包含以下三种窗口:

①主窗口,又称框架窗口 ②用户窗口 ③文档窗口

它们之间的关系可用上图表示。

由上图我们可以知道,主窗口是用户窗口的父窗口,而用户窗口同时又是各文档窗口的父窗口,明白这一点对于创建各个窗口是非常重要的。需要指出的是,这里的主窗口与通常意义下的主窗口又有所区别。一般的主窗口,它的用户区可以全部被用户用来作各种输出,包括文本和图形,而多文档界面下的主窗口却是被用户窗口所填充,用户窗口本身并无任何用途,它只是作为文档窗口的底板窗口。对于文档窗口,尽管我们将其称为子窗口,但这种子窗口只能用于显示文档,它不能具有菜单,而且任何时刻只能有一个子窗口是活动的,且不能移到用户窗口之外。

多文档窗口的上述特点使其具有广泛的应用价值。在我们开发的雷达仿真软件中,由于采用了多文档界面,使整个仿真过程既增加了交互能力,又使过去我们认为枯燥无味的仿真变得生动、有趣。具体实施时,我们将仿真分为三种情形,它们分别是特定距离的分布型目标仿真,相对距离变化过程的仿真,以及任意信号的产生。对于第一种情形,我们所关心的是基于一选定的目标其雷达回波的时域和频域特性,这可以通过弹出两个文档窗口来完成,一个显示时域波形,另一个显示频域波形,如果我们想对比一下不同目标和不同距离的目标回波,可以打开更多的文档窗口。对于第二种情形的仿真,我们所研究的是在给定的距离变化区域里和特定目标的情况下,目标雷达回波的动态变化过程以及雷达信号处理终端的工作性能,为此,我们打

开了三个文档窗口,在第一个文档窗口中,显示距离变化轨迹,第二个窗口显示目标回波,这里,目标回波的时延和波形是随着距离变化而不断变化的,在第三个文档窗口中,我们显示出信号处理终端的框图,并随着距离的变化显示关键监测点的信号变化,由此我们可以对信号处理终端进行一定的性能评估。最后一种仿真是任意信号的产生,它可以通过给出函数表达式或进行数据表格的填写产生出任意信号。上述三种仿真既可以单独进行,又可以并行进行,也就是说,在进行某一种仿真时,如果它需要较长的时间,我们可以同时启动其它种仿真,而且所有打开的文档窗口可以并存于同一主窗口中,我们也可选取所感兴趣的文档窗口将它放大到整个主窗口,可以对任意窗口进行打印输出。

需要指出的是,整个处理过程是基于数据库进行的,关于仿真主程序与数据库的数据交换过程我们将另文论述。

下面我们分几个方面对多文档窗口的设计中的一些关键技术问题进行较为详细的论述。

### 3 多文档窗口的创建

对于任何一个 WINDOWS 应用程序,它都必须有一个入口点,这就是 WinMain() 函数,在该函数中,它主要要完成窗口类的登录工作,并显示和刷新主窗口。对于多文档窗口,它的窗口登录与通常的窗口并无多少不同,只是在登录文档窗口的窗口类时,我们使用了 wndclass.cbClsExtra,关于它的作用留在后面再谈。如果仔细思考一下,我们会发现一个问题,在多文档窗口组成一节中,曾谈到了三类窗口,可在这里,我们只登录了两类窗口,一个是主窗口,一个是文档窗口,为什么不登录用户窗口呢?原来在 WINDOWS 内部提供的预定义窗口类中,有一个叫“MDICLIENT”,它就是用户窗口的模板,所以不需要再另外登录了。

在已登录的窗口类的基础上,我们可以完成窗口的创建工作。首先是主窗口,它的创建可以通过下面的函数调用完成:

```
hWndFrame = CreateWindow ( szFrameClass, " RADAR
SYSTEM",
WS _ OVERLAPPEDWINDOW | WS _
CLIPCHILDREN,
CW _ USEDEFAULT, CW _ USEDE-
FAULT,
CW _ USEDEFAULT, CW _ USEDE-
FAULT,
NULL, hMenuInit, hInstance, NULL)
```

接下来就是用户窗口的创建。尽管它也是通过调用 CreateWindow 函数创建的,但是所不同的是,它是在窗口过程中创建的,也就是说,在窗口过程中,当收到 WM \_ CREATE 消息时,处理下面的函数调用而完成该窗口的创建:

```
hWndClient = CreateWindow ( " MDICLIENT", NULL, WS _
CHILD
| WS _ CLIPCHILDREN | WS _ VSCROLL | WS _
HSCROLL,
0, 0, 0, 0, hWnd, 1, hInst, (LPSTR)&clientcreate)
```

现在该进行文档窗口的创建过程了。文档窗口的创建过程与一般窗口有着本质的不同,这主要体现在两个方面,其一,它不是由 CreateWindow 函数调用实现的,其二,它是在客户窗口的窗口过程中建立的。具体地讲就是,由应用程序向客户窗口发一个 WM \_ MDICREATE 消息,

```
hWndChild = (HWND) SendMessage (hWndClient, WM _ MDI-
CREATE,
```

```
0, (long) (LPMDICREATESTRUCT)&mdicreate);
```

该消息的 iParam 参数含有一个指针,它指向名为 MDICREATESTRUCT 的数据结构,该结构包含了文档窗口的类名、标题名、主窗口的句柄、窗口的左上角坐标及窗口的大小、窗口的风格等信息。

对于前面所讨论的第二种仿真,一次需创建三个文档窗口。我们用 hWndChild1, hWndChild2, hWndChild3 标识三个具有不同类的文档窗口,在每次创建好一个文档窗口后调用函数 MoveWindow() 将这三个窗口按一定格式排列于用户区内。对每个窗口的操作在窗口过程中根据不同的标识值检索完成。

### 4 消息循环

Windows 应用程序采用消息驱动的机制来运行的,它的人口函数 WinMain 在完成窗口类的登录和创建后,将把控制权交给 Windows 操作系统,而自己只管理消息循环部分,它不断的从消息队列中提取消息,然后在完成一些键盘转换和处理工作后,又把消息发回 Windows 系统,由 Windows 将消息转发给相应的窗口过程,其代码如下:

```
while (GetMessage (&msg, NULL, NULL, NULL))
{ TranslateMessage (&msg);
DispatchMessage (&msg);
}
```

对于多文档应用程序,它的消息循环从总体来讲是类似的,但也有些不同。这主要体现在它是使用 TranslateMDISysAccel 函数来翻译子窗口的加速键,其实现代码如下:

```
while (GetMessage (&msg, NULL, NULL, NULL))
if (! TranslateMDISysAccel (hWnd MDIClient, &msg) &&
! TranslateAccelerator (hFrameWnd, hAccTable, &msg))
{ TranslateMessage (&msg);
DispatchMessage (&msg);
}
```

其中 TranslateMDISysAccel 函数将某些 WM \_ KEYDOWN 消息翻译成 WM \_ SYSCOMMAND 消息,如果不是,则调用 TranslateAccelerator 判断是否是应用程序定义的加速键消息。由于这两个函数均带有窗口的句柄,所以它们可以直接将消息发给相应的

窗口过程。如果既不是文档窗口系统菜单的加速键又不是应用程序定义的加速键,则该消息经 TranslateMessage 处理后,由 DispatchMessage 发给相应的窗口过程。

## 5 窗口过程

为了阐述这个问题,我们以仿真软件中一个文档窗口为例对其进行较为详细的解释。该文档窗口需实现以下功能:根据输入的文件名打开一数据文件,读入其中的点连成曲线,并能进行显示图形的滚动和所有点的数值读取等功能。

首先要根据输入的文件名打开数据文件,这可以在填写名为 ofn 的数据结构后调用 Windows 提供的 GetOpenFileName() 函数实现,因为打开的数据文件是以字符串形式保存的,还需要将其转化成相应的浮点数并连点成线,下面的程序块执行这一功能:

```
MoveTo(hdc,100,300);
j=1;
m=0;
n=0;
_llseek(hFile,0L,0);
for(i=0;i<=13000;i++)
{read(hFile,buffer,1);
m++;
if(*buffer==0x0d)
{ _llseek(hFile,n,0);
read(hFile,buf,m);
n=_llseek(hFile,1,1);
data=atof(buf);
LineTo(hdc,100+j,data);
j=j+1;
m=0;}}
```

考虑到在 Borland C 环境下仅用系统提供的滚动杆和 GDI 函数难以实现图象的滚动,鉴于图元文件是以二进制形式存储 GDI 函数的调用信息,因而占用的内存或磁盘空间很少,这对于仿真系统是非常有益的,因为在仿真计算过程中会产生大量的数据,如何节省内存的使用是很重要的。图元文件本身既可以存储在一个磁盘文件中,又可以存储在一个内存文件中,我们选择的是后者,这主要是考虑到所有的最终数据在计算完成之后,都将存入数据库或数据文件中,以便于对数据的查询,而图元文件由于存储的都是调用信息,因而难以实现对数据的直观查询。对于以内存为基础的图元文件,其句柄就是它的唯一标识,因而对它的读取是非常方便的。由于我们开发的软件是基于多文档窗口环境,有一个重要的问题需要解决,这就是当多个窗口公用一个窗口过程时,如何使任一窗口的私有数据与其他窗口的私有数据区分开来?。举例来说,在本软件中,我们会连续打开多个文档窗口,每个文档窗口含有一个回波波形,为了不使它们相互混同,以及便于后续的各种处理,必须能够找到一种有效的机制可以

把它们区分开来。经过仔细分析,终于在 Windows 程序的窗口类定义中找到了所需的答案,这就是设置类定义结构中的附加字段。在一般的 Windows 应用程序中,这个字段是设置为零,而在多文档程序中,需要将私有数据的有关信息存储到该附加字节中,其长度依需求而定。

为此,在程序中定义了一个结构型全局变量和指向它的一个近程指针:

```
typedef struct{
    HDC MeatHdc;
    HMETA hmf;
}Metadata
typedef MATADATA NEAR *NPMETADATA;
```

在该结构中,图元文件设备描述表和图元文件句柄被定义为私有变量。在处理 WM\_CREATE 消息时,为私有数据分配局部或全局的内存空间,然后建立图元文件,之后调用 LockUnlock() 和 SetWindowWord 释放内存空间,并把该数据内存句柄保存在窗口结构的附加字节中。

```
case WM_CREATE;
    hMetaData = LocalAlloc(LMEM_MOVEABLE|LMEM_ZEROINIT,
        sizeof(METADATA));
    npMetaData = (NPMETADATA)LocalLock(hMetaData);
    npMetaData->hdcMeta = CreateMetaFile(NULL);
    *****
    Create MetaFile.(程序略)。
    *****
    LocalUnlock(hMetaData);
    SetWindowWord(hwnd,0,hMetaData);
    //Initialize ViewPortOrg.
    XVORG=0;
    YVORG=0;
    //Initialize Scroller's Postion And Range.
    ScrollLine=100;
    SetScrollPos(hwnd,SB_HORZ, poshorzscroll, TRUE);
    SetScrollPos(hwnd,SB_VERT, posvertscroll, TRUE);
    SetScrollRange(hwnd,SB_HORZ,0,ScrollLine,FALSE);
    SetScrollRange(hwnd,SB_VERT,0,ScrollLine,FALSE);
return 0;
```

图形的滚动,可以在响应 WM\_VSCROLL 和 WM\_HSCROLL 消息时,通过一定的处理来完成。也就是说,根据滚动范围的大小,在响应不同的 SB\_消息时,不断地改变视口原点的位置,然后调用 InvalidateRect() 函数,连续地重画用户区,从而达到图形滚动的效果。此处以垂直滚动为例。

```
case WM_VSCROLL;
    switch(wParam)
    {case SB_LINEDOWN;
        posvertscroll+=1; //滑块位置改变。
        YVORG++; //视口原点位置改变。
        InvalidateRect(hwnd,NULL,TRUE);
        break;
```



```

case SB_LINEUP;
posvertscroll-=1;
YVORG--;
InvalidateRect(hwnd,NULL,TRUE);
break;
case SB_THUMBPOSITION;
posvertscroll=LOWORD(lParam);
break;
default;
break; }

```

图元文件的显示是在处理 WM\_PAINT 消息时通过调用 PlayMetaFile() 函数实现的。在调用该函数前需要先用 LocalLock() 或 GlobalLock() 函数分配内存,然后调用 GetWindowWord() 函数检索到该控制的句柄,从而获得有关图元文件的信息并绘出相应的曲线。

```

case WM_PAINT;
hdc=BeginPaint(hwnd,&ps);
SetMapMode(hdc,MM_ISOTROPIC);
SetWindowExt(hdc,xClientView,yClientView);
SetViewportExt(hdc,xClientView,yClientView);
SetViewportOrg(hdc,XVORG+50,YVORG-100);

```

```

hMetaData=GetWindowWord(hwnd,0);
npMetaData = ( NPMETADATA ) LocalLock
(hMetaData);
PlayMetaFile(hdc,npMetaData->hmf);
EndPaint(hwnd,&ps);
return 0;

```

## 6 结束语

通过以上论述,相信大家已经对 Windows 环境下多文档环境的应用和开发有了一个较为清晰的系统认识。尽管我们是将该环境应用到雷达仿真软件的开发之中,但由于其具有很强的通用性,如果将其引入到其他应用软件的开发之中,不但能够使软件的生命力大大提高,而且也促使程序员的设计水平上升到一个新的层次。由于篇幅所限,关于窗口其它功能的描述,在此就不再赘述了。

## 参考文献

- 1 《WINDOWS V3.0 程序设计指南和工具》 Microsoft Corporation 编 宋明华等译。
- 2 《PROGRAMMING WINDOWS 3.1》 海洋出版社 文都等译

# 多用户文档管理系统的设计

孙正严 张积友 范冰冰

(广州通信学院 510502)

**摘要** 本文主要讨论了在 SQL 客户/服务器结构下,多用户文档管理系统设计中的三个问题:文档正文和数据库的存放结构、文档正文与数据库的连接以及查询输入接口。

**关键词** 文档管理 远程用户 SQL 服务器 查询

## 1 引言

随着计算机网络技术的发展和普及,办公自动化管理在行政机关、工商企业、医院、学校等行业应用越来越广泛,而多用户文档管理在办公自动化管理中是很重要的部分。这里的多用户文档管理系统是指:多个用户可在局域网上、远程终端、或其他互联的局域网上,同时对文档系统进行输入、修改和查询。多用户文档管理系统特别适用于那些与外界联系广泛的、需要对公开资料进行查询的行业、也适用于各部门需要共享文档资源的行业。本文以某行政机关多用户文档管理系统的实现为例,阐述了在多用户文档管理系统设计中几个重要方面的设计思想与方法。

## 2 运行环境与数据结构

### 1) 系统开发与运行环境

该系统的开发平台选用的是面向对象设计语言 Visual Foxpro 3.0。在局域网上设有专用的 SQL 服务器,以提高系统的运行速度,特别是提高远程用户的查

询速度。网络上的用户被分为两大类,一类是局域网内各个部门用户,这类用户可以添加、修改、和查询记录;另一类为公共用户,主要是指远程用户,或其他互联网上的用户,这类用户只能查询,而不能修改记录。

被管理的文档分为三类,一类是本部门文档,这类文档只有本部门使用,其他部门和公共用户不能修改或查询记录;第二类是机关内部文档,这类文档由各部门建立,只能由建立的部门修改,但其他部门可以共享查询这些文档;第三类是公共文档,这类文档由各部门建立,只能由建立的部门修改,但其他部门和公共用户可以共享查询这些文档。

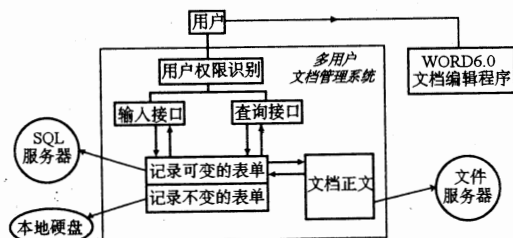


图1 多用户文档管理系统的结构

多用户文档管理系统的基本使用步骤是:用户首先在文档编辑程序下编辑好文档的正文(本系统选用的是 Microsoft 公司的 WORD6.0 文档编辑软件),然后再进入多用户文档管理系统,输入文档的有关信息(如文档标题、分类、部门、关键词、作者、日期等),供以后查询使用,并将文档正文连接到数据库上。用户可以在文档管理系统下,直接调用文档编辑程序(WORD6.0)来浏览或修改文档正文。

系统的结构示意图见图一。

## 2) 文档正文的存放结构

在多用户文档管理系统中,文档正文的安全存放或保密是极其重要的,本系统设计中,文档的保密是通过网络文件服务器,对文件存取的授权来实现的。通常网络服务器对文件的管理的安全机制是很完善的,这样就保证了文档正文的高安全性存取。文档的存放结构有图二所示的两种方式,其区别在于对用户进行读写授权或增加新的部门用户时,其操作的复杂程度的不同,通常应选用图二(a)的存放结构,因为其授权操作较为简单。

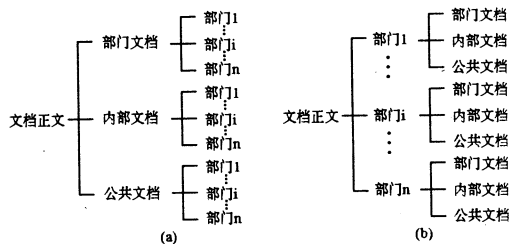


图2 文档正文的存放结构

## 3) 数据库的存放结构

数据库中的表单被分为两类,第一类表单里的记录是可变的,多个用户需要共享这些表单,通常它们被存放在 SQL 服务器上;另一类表单里的记录是固定不变的,通常它们被存放在本地硬盘上。这样做主要有以下两个优点:

- 可以提高数据库的安全性,由于记录信息的表单被存放在 SQL 服务器上,当用户需要使用这些表单时,必须要建立正确的连接字符串,才能打开这些表单,而与 SQL 服务器的连接字符串用户是不知道的,只有设计者才知道,因此用户很难脱离本系统直接进入 SQL 服务器打开这些表单。这样就利用了 SQL 服务器良好的安全性来实现了本系统数据库的安全性。
- 可以提高系统的运行速度,由于记录固定不变或无需共享的表单被存放在本地硬盘上,这样就可以提高读取数据的速度和减轻网络服务器的运行负担,特别是可以大大提高远程终端的运行速度。

## 3 文档正文与数据库的连接方法

文档正文与数据库的不同连接方法,直接关系到

系统的运行速度,下面将讨论两种不同的实现方法及其特点。假设数据库的表单(Table)为 WordFile.dbf。

1) 利用 Visual FoxPro3.0 的 OLE 技术来实现文档正文的连接

- 在表单中定义一个通用字段(General Field) wordFile。
  - 在窗体(FORM)上添加一个 OleBoundControl 的控制对象 OCB1。
- ```

令 OCB1. ControlSource=wordfile.wordfile
● 在窗体上添加一个命令控制按钮(CommandButton) LinkFile, 在 LinkFile.Click 事件中加入代码:
LOCAL lcFileName
lcFileName=GETFILE("DOC","连接文档正文","选择")
IF !EMPTY(lcFileName)
    REPLACE wordFile.wordfile WITH(lcFileName)
ENDIF

```

这种方法的优点是文档正文以图标或图形的方式显示,用户能直观看到文档正文;缺点是在窗体上连接或显示文档信息的速度较慢,这是因为在连接或显示文档信息的时候,OLE 控制将文档正文以图形点阵的方式读入,当文档正文很长或用户为远程用户时,程序运行速度会很慢,用户可能会难以接受。

2) 通过控制和管理文档正文的存放路径来实现文档正文的连接

- 在表单中定义一个字符型字段(Character Field) wordFile。
- 在窗体上添加一个命令控制按钮(CommandButton) LinkFile, 在 LinkFile.Click 事件中加入代码:

```

LOCAL lcFileName
lcFileName=GETFILE("DOC","连接文档正文","选择")
IF !EMPTY(lcFileName)
    REPLACE wordFile.wordfile WITH(lcFileName)
ENDIF

```
- 在窗体上添加一个命令控制按钮(CommandButton) ViewFile, 在 ViewFile.Click 事件中加入代码:

```

LOCAL lcFileName
lcFileName=wordfile.wordfile
IF !EMPTY(lcFileName)
    RUN/N C: \ WINWORD \ WINWORD.EXE
    &lcFileName
ENDIF

```

这种方法的优点是在窗体上连接或显示文档信息的速度快,因为程序只需处理文档正文的存放路径,而无需读入文档正文本身,缺点是不及前一种方法直观,但这一点在文档管理中并不重要。

## 4 文档查询的接口方式

选用什么样的查询接口方式,才能使用户快速而方便地查询到所需的内容,这主要体现在查询输入的方式上。由于系统采用的是客户/服务器结构,那么查询定位精度越高,在网络上传输的数据量就越少,这对远程用户来说是至关重要的。在面向对象设计中一般来说有以下三种方式接受用户的输入:(下面假设用户可以按照部门、文档分类和关键词三个字段来查询)

#### 1) 文本框(TextBox)

用户在文本框中输入需要查询的字段值,然后在数据库中查询到该值,其特点是初始化建立时间短,但由于用户事先不知道字段中有哪些值,使得用户很难准确定位要查询的记录,且用户需要拼写输入字段值,这给用户的操作带来了不便,用户使用不直观。在多个字段同时查询时,可能会出现空记录查询。

#### 2) 下拉列表框(ComboBox):

将要查询字段的所有的值都放入下拉列表框,用户在下拉列表中选择需要查询的字段值,其特点是直观,无需用户拼写输入,但由于要将查询字段的所有值都放入下拉列表中,所以建立时间长,不利于远程用户的使用。在多个字段同时查询时,可能会出现空记录查询。

#### 3) 相关的列表框(ListBox)组

主要用于多个字段的同时查询,当用户选择第一列表框中的某个值时,第二列表框中的内容是和第一列表框中的值逻辑与的关系,以此类推。例如用户在第一列表框中选择了财务部,在第二列表框中就只出现财务部有的分类,如果在第二列表框中选择了报表的分类,则在第三列表框中就只出现财务部报表分类中有的关键词。其特点是非常直观,建立时间较快,定位精度高,查询速度快,不会出现空记录查询。

在多用户文档管理系统的设计中,查询输入接口应选用第三种接口方式(相关的列表框组),特别适合远程用户的使用。

#### 5 结论

通过以上的分析,作者认为在多用户文档管理系统的设计中有三点值得注意的,一是文档正文与数据库应分开存放,有利于提高系统的运行速度;二是通过控制和管理文档正文的存放路径来实现文档正文的连接,而不适合采用 OLE 技术来实现文档正文的连接;三是采用相关的列表框组来实现查询输入接口,这是一种良好的用户接口方式。

## 读者意见调查表(复印有效)

读者姓名\_\_\_\_\_ 年龄\_\_\_\_\_

职务或职称\_\_\_\_\_

工作单位\_\_\_\_\_

电话\_\_\_\_\_

通信地址\_\_\_\_\_

邮码\_\_\_\_\_

●您对本刊总体影响:(优:√,良:△,差:×)

☐文稿可读性 ☐编辑质量 ☐印刷质量

☐栏目的设置 ☐版面设计 ☐核对质量

●您认为本期各栏目的质量如何(优:√,良:△,差:×)

☐综述评论 ☐研究探讨 ☐应用开发 ☐维护维修 ☐动态简讯

●您希望本刊新增哪些栏目或哪类的文稿?

●本刊存在的问题、不足及您的建议,请赐教。

注:①对认真填写并寄回此表的读者,我们均寄本刊最新出版的杂志一本,以示谢意。

②此表请寄至:成都市人民南路四段11号附1号 邮码 610041

《计算机应用研究》杂志社张钢编辑收,谢谢合作!

# 基于客户机/服务器环境下的 MIS 系统开发

倪凯民 唐弋清

(安徽省计算中心 合肥 230031)

**摘 要** 本文探讨在客户机/服务器环境下开发管理信息系统,使得客户机和服务器能有效地发挥各自的作用,缩短开发周期,对数据库的维护与管理更加方便容易。

**关键词** 客户机/服务器 MIS 数据库 数据共享

## 1 信息处理的传统方式面临挑战

随着计算机应用的普及和深入,以及网络技术的逐步成熟,图形化方式为主的用户界面技术的普及,特别是多媒体技术的发展,MIS 系统已从单一的应用发展到综合应用,从简单地处理深入到复杂处理。信息系统处理模式已从集中式、分散式发展到分布式处理模式,应用任务正从大型主机系统转移到以客户机/服务器为支撑的分布式网络上。

在原有的网络管理信息系统中,其主要特点是所有的处理都是在工作站上完成。文件服务器硬件虽是一台高档微机。但实现的不过是工作站文件的存储及打印扩展,而对数据的处理,诸如检索、更新、计算等仍由工作站完成。同时由于服务器只能简单地管理文件,所以当一工作站提出对文件的处理要求时,它只能将整个文件通过网络传输给用户,这样传来的文件中可能大部分对用户来说是当前不需要的,这无疑增加了网络的负担。这些问题严重地困扰着计算机用户和系统开发者。在客户机/服务器模式下开发 MIS 系统,将数据管理及处理工作从客户机上分离出来,使网络上计算机的资源各尽其能,功能较简单的微机用作客户机,而大量的数据处理工作则由功能较强的服务器来承担,服务器控制管理数据的能力由文件上升到数据库方式。这样大大降低了网络流量,使 MIS 的整体性能指标有了较大的提高。

## 2 客户/服务器的结构及特点

客户/服务器是以网络环境为基础,将计算机应用有机地分布在多台计算机中(如图 1),其中服务器负责数据的存储管理及文件服务,而客户机负责用户对数据及文件提出请求服务,由服务器把处理后的数据传送给客户,在网络中传输的仅仅是客户需要的那部分,而不是全部。这样大大降低了网络数据的流量,使系统的性能有了较大的提高。客户/服务器方式增加了数据的共享能力,数据库服务器上存放大量的数据,用户只需在客户机上用标准的 SQL(结构化查询语言)访问服务器中的数据,可以很方便地得到所需的各种数据及信息。

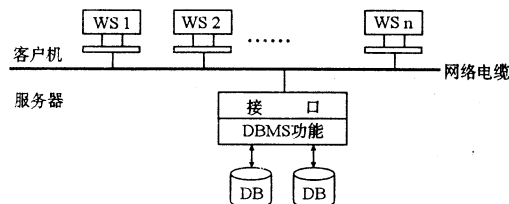


图1 客户/服务器结构

客户/服务器结构是一个开放的体系结构,使得数据库不仅要支持开放性,而且还要开放系统本身,这种开放性包括用户界面、软硬件平台和网络协议。利用开放性在客户机一侧提供应用程序接口(API)及网络接口,使用户仍可按照他们的熟悉的、流行的方式开发客户机应用。在服务器一侧,对核心 RDBMS 的功能调用,网络接口满足数据完整性、保密性及故障恢复等要求。有了开放性数据库服务器能支持多种网络协议,运行不同厂商的开发工具。而对某一个应用开发工具也可以在不同的数据库服务器上运行,存取不同数据源中的数据,给 MIS 系统的开发提供了极大的灵活性。客户/服务器内部结构如图 2。

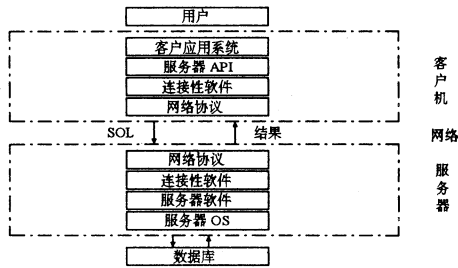


图2 客户/服务器内部结构

### 1) 客户/服务器的主要特点:

(1) 客户/服务器模式是一种分布式处理模式。应用任务分别由客户机和服务器分担,这样一方面充分发挥了客户机的处理能力,另一方面减轻了服务器的负担。它使得处理尽可能地向正在被处理的数据靠拢,网络的负载大大下降,而网络的响应速度则大大提高。

(2) 数据库是多用户的,这对数据的完整性、安全性、数据的访问控制非常有利。

(3)系统的扩充性好,可方便地在网络上增加客户机或服务器。

(4)一般来说,客户机是 PC 机,具有良好的图形界面和丰富的应用软件,对 MIS 系统的开发非常有利。

## 2) 客户/服务器的组成

因为客户/服务器计算的目的是解决应用问题,因此,从用户的角度看,其基本组成有三个部分:客户、服务器、客户与服务器的连接。

(1)客户 客户是一个面向最终用户的接口或应用程序。它通过向一个设备或应用程序(服务器)请求信息。然后将信息显示给用户。客户把大部分数据处理工作留给服务器,让服务器上的高档硬件和软件充分施展它的特长。通过将数据分析和图形表示从服务器上剥离下来,客户硬件便能大大地减少网络上的传输事务,使网络能为用户提供更加有效的信息流。

(2)服务器 服务器的主要功能有建立进程和网络服务地址,监听用户的调用,处理客户的请求,将结果交给客户,释放与客户的连接,所以服务器可以是大型机、中小型机或高档微机。要求高档的处理器,大容量内存,稳定快速的总线和网络传输,以及完整的安全措施。服务器为客户机请求提供所要求的服务,包括数据析取,数据加工,数据处理等。

(3)连接件 客户与服务器之间的连接是通过网络连接实现的。而对应用系统设计来说,此种连接更多的是一种软件通信过程(如网络协议等),对应用开发人员来说,客户服务器之间的连接主要是它所能使用的软件工具和编程函数(API)。近年来,各种连接客户和服务器的标准接口或软件大受欢迎,如开放的数据库连接(ODBC)就是一种被广泛采用的数据库访问连接件。ODBC 是一种基于 SQL 访问组织规范的数据库连接应用程序接口(API),该接口可在应用程序与一个或多个数据库服务器之间进行通信。客户应用只需与标准的 ODBC 函数打交道,采用标准的 SQL 语言来编程,而不必关心服务器软件的要求及完成方式。

## 3 客户/服务器在 MIS 中的作用

### 1) 数据管理

数据库服务器能透明地处理数据库中并发控制问题,如数据加锁、数据一致性控制及解决用户之间的冲突。数据库服务器提供了强有力的数据安全保证。能保护数据库中的数据不被客户端的数据请求所破坏、所有的数据请求必须由数据库服务器软件来执行,因此,即使是对数据库系统十分熟悉的用户也不能用非法的用户名及保密字,绕过数据库服务器的安全检查机制。

客户/服务器结构的数据库管理功能和数据库应用由数据库服务器来完成。它为客户机的应用和数据请求提供数据管理和数据处理功能,其中包括查询、更新、事务管理、安全等。客户机完成应用程序的执行

以及客户机和服务器之间信息、数据请求、结果的传递,这一模式大大提高了应用的执行效率,为数据库应用开辟了新的途径。

同时在客户/服务器结构系统可提供一套友好的开发环境。解决异构数据库之间的数据交换和互操作,保护用户前期投资,保证今后系统的可扩充性和各种硬件平台上的可移植性。

### 2) 共享数据库及本地数据库

一个 MIS 系统成功的关键是设计一个好的数据库以便提高数据的共享能力,充分利用各种资源的能力,使得数据能充分得到利用和共享。这就要求对数据进行分析,使共享的信息放在服务器中,不经常使用的数据放在本地,即建立共享数据库和本地数据库。共享数据库是管理信息系统的一个重要环节,从某种意义上讲,它决定了今后系统开发、运行管理和维护的难与易,对能否充分发挥网络硬件系统与网络软件支撑环境和功能同样起着重要作用。这个共享数据库可以放在系统服务器中进行统一的管理和维护,以提高系统的运行效率。设计一些本地数据库在一个 MIS 系统中也是不可缺少的,因为,在一个网络中,所有的数据都是共享方式是不切合实际的,设计好一个本地数据库,可以提高网络的整体效率,减少网络负荷,而且数据库的维护就方便容易。

### 3) 数据与程序独立

现在开发一个 MIS 系统大多数都采用先进的开发工具,在客户机开发应用程序无需考虑从何处取数据,是本地的,还是哪个服务器上的,这一点对 MIS 系统的开发非常重要。

### 4) MIS 系统的开发简单

客户/服务器的结构将应用分解为在不同客户平台和服务器平台上的多个子任务。因此,开发一个管理信息系统可根据不同的子任务同时开发,而且现在有很多开发工具,特别是客户一方提供的工具可开发窗口界面,提供许多支持开发人员不必考虑过多的细节(如 Power Builder),也不管与什么样的数据库连接。

## 4 客户/服务器结构应用实例

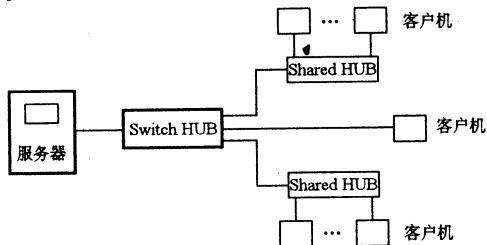
在信息管理领域。客户/服务器模式有很大潜力,一些大的信息系统也都采用 C/S 结构。以前采用其它结构设计的 MIS 系统,随着规模的扩大,也正逐步向 C/S 结构转变,这对于缩短开发周期,降低开发投入、扩展系统规模、吸纳新技术有着显著的意义。

我们在开发某部门管理信息系统时,选用了基于 PC-LAN 环境的 C/S 模式,主要理由有:PC 机的开发和应用已相当成熟和普及,特别是成熟的汉字处理软件和设备,PC 技术是最活跃的技术,无论是网络、数据库或多媒体,都已应用在 PC 机上,且成本相对低廉,符合我国国情,PC 机软硬件技术已能提供包括多 CPU、大存储容量、高安全性和较好容错能力的 PC 服

务器硬件平台,具备对称多处理、多线程、多任务的网络操作系统。这些给在 PC-LAN 上构造 C/S 结构系统奠定了技术基础。

要规划设计客户/服务器结构信息系统,主要应规划好三种体系结构,应用体系结构、数据库体系结构、技术体系结构。要掌握的原则就是量力而行,选择既能满足需求又易于开发、管理和维护且符合经济预算的方案。

下面给出 PC-LAN 环境的客户/服务器 MIS 结构图。



网络操作系统的选择。微软公司发布的网络操作系统 Windows-NT 以及基于 Windows-NT 的客户/服务器的应用软件——后台数据库服务器,以其技术先进和使用方便为广大用户所接受。它的前端操作平台——Windows NT workstation 具有支持范围广泛的重要事务应用程序和一系列丰富的开发工具。

选择 Windows-NT Server 作为网络服务器,就意味着优先选择 Microsoft SQL Server 作为后台数据库服务器。采用 Microsoft SQL Server 作为后台数据库服务□S 器,是与 Windows-NT Server 的完美结合,二者结合,集成度高,一致性好。

确定开发语言的原则是要考虑其先进性、灵活性、成熟性和可维护性。基于 Windows 环境的开发有很多种语言可以选择,如 Visual Basic、C、C++、Power-Builder、Visual Foxpro、Access 等等,VB 有简单易学、灵活方便、易于扩充的特点,很多厂家为其提供定制控件,从而大大增强了它的应用范围。因此,基于 Windows 环境的开发,可优先选择 Microsoft Visual Basic 及 Visual Foxpro,一则开发人员容易掌握,再则,程序人员还可根据需要,用 C 或 C++ 编写自己的定制控制,扩展其功能,使程序设计更加得心应手。使用 Visual Basic 和 Visual Foxpro 作为前台开发语言,

与 SQL Server 接口可使用 ODBC (Open Database Connectivity),其通用性好,允许最强的互操作性,编程简单。

系统分析与设计,应按照客户/服务器结构的特点来分析数据和业务要求。整个系统设计过程强调以数据为中心,先定义系统需求,后定义信息需求,再定义满足信息需求的数据要求,设计存储这些数据的数据库、而处理程序作为维护数据库的一次处理过程和从数据库取出数据进行加工、获得信息的二次处理过程。以数据为中心进行 MIS 系统设计的主要优点是保证了数据的完整性和一致性,并且在设计数据库的同时,可以并行设计各个业务子系统,提高开发效率。

采用客户/服务器结构设计 MIS 时,存在着众多的组合,它一方面可以给用户提供更灵活、选择性更大的系统建设方案,另一方面又给开发人员提出了更高的要求,特别是难以确定客户机和服务器各负担哪一部分应用处理任务。由于众多的原因也难以确定数据如何分布较为合理,只能视应用而定。这种具体的设计尚无划分规则可循,只有因地制宜,找出适合自己的方案,才能从客户/服务器这种技术体系中获得良好的效益。因而在设计和开发系统时最关键的是应用系统开发工具和环境具有足够的灵活性,允许在最后决策之前对不同的可能结构进行调整。一般的设计原则是界面和数据业务处理由客户机承担,数据管理(如查询、统计)由服务器完成。

## 5 结束语

客户/服务器结构的开放性、先进性、可扩充性和实用性确实给 MIS 建设带来了很多好处,首先客户/服务器结构改变了以往用户对供应商的产品及技术的绝对依赖性,用户变被动为主动,无论对硬件设备、系统软件还是应用软件,客户选择的余地都很大。再则系统的开放性保证了其技术的先进性,确保了系统的安全可靠、可升级、可维护、可扩充,能有效地保护用户的投资并满足用户未来发展的需要,使 MIS 建设进入良性循环。客户/服务器模式,本质上是一种计算的优化,是应用实现的简单化,基于我国国情和世界技术发展的趋势,广泛采用该计算模式。特别是基于 PC-LAN 的客户/服务器模式,将使计算机应用走上一个新的台阶。

## Novell 推出最新内部网平台

Novell 公司于今年十月推出名为 Intranet Ware 的产品使全球领先的网络操作系统 Netware 发展为适于任意规模商用的全套服务内部网的 Injnet 访问平台解决方案。

以目前的制作水平看,Injnetware 将允许 Netware 客户使其商业网络纳入 Internet 技术,同时保留现有网络投资。

Intranet Ware 结合了 Net Ware4.1 中的全部行业领先网络服务以及新的内部和 Internet 功能,因而可以为企业内部建立生成全面的解决方案。作为一个综合性解决方案 Intranet Ware 不仅可以为各机构提供将现有网络发展成企业所需的技术,还可以继续提供它们一直依赖的网络服务。

新的解决方案表明,现正在向具有 Intranet Ware 功能的基于服务的商业进行转移。Intranet Ware 将已经商业检测的所有网络服务向内部网络环境开放。通过提供此类产品,正在推出使用户业务走向二十一世纪所需的用途广泛的解决方案。

Intranet Ware 不仅使 5500 多万用户能够利用 Internet/内部网解决文字增强其现有的 Netware 网络的功能,而且还能为新用户提供内部所需



# 电子企业 CIMS 工程中 CAD/CAM 的集成

金涌 黄沛然 文远保

(华中理工大学计算机系 武汉 430074)

**摘要** 本文分析了电子 CAD/CAM 的一些特点和共性,结合一个电子企业 CIMS 工程的实践,提出了一个适合设计人员、工艺人员和操作人员协同工作的集成系统结构。本文还讨论了集成系统中责任数据的产生和管理问题。

**关键词** CIMS CAD/CAM 信息集成

## 1 引言

电子企业由于产品的特点与机械行业不同,其生产流程和管理组织方式也具有一定的特点。因此在实施 CIMS 应用工程时应设计适合于电子企业的集成方案。CAD/CAM 是电子企业 CIMS 工程中的关键,本文在国家 863 计划 CIMS 重点应用工厂项目—武汉邮电科学研究院 CIMS 工程实践的基础上,讨论电子 CAD/CAM 集成的一些共性问题。

## 2 电子 CAD/CAM 的特点

### 2.1 电子 CAD

电子工厂的产品一般以电路板为主,产品的设计软件工具包括原理图绘制软件、电路模拟软件和 PCB 板设计软件等。最终以 PCB 板的 CAD 设计文件提交。因此 PCB 板的 CAD 设计文件是电子 CAD/CAM 集成的主要数据来源。目前国内较常见的 CAD 设计软件有微机上的 ORCAD、TANGO、PROTEL 等软件和工作站上的 MENTOR GRAPH 等。电子 CAD 设计软件一般都可产生下列设计信息:

- 元件位号:如 R1,R2,C1,U1,J1;
  - 元件名称:如 J1/4W-500,100u,74LS123;
  - 元件封装:如 RC0805,DIP14,SIP6;
  - 元件引脚:如 1,2,3;
  - 元件参考点坐标(X,Y);
  - 元件摆放方向:如 0,90,180,270;
  - 节点号:如 net196;
  - 元件种类:SMT(表面贴元件)或通孔元件等;
  - 元件引脚号及相对元件参考点的坐标;
- 电子 CAD 设计软件一般可产生的设计文件有 PCB 版图文件、元件位置文件、元件清单列表文件、网表文件、钻孔文件、以及光绘机、绘图机或打印机的输出文件。这些文件是数据集成的主要数据来源。表一是

一个 ORCAD 的元件位置文件例子,表二是一个 MENTOR GRAPH 的 PCB CAD 设计软件的元件位置文件例子。

表 1:ORCAD 的元件位置文件例子(部分)

| Module |        |          |         |     |       |       |        |
|--------|--------|----------|---------|-----|-------|-------|--------|
| Module | Module | File     |         |     |       |       | Net    |
| Refer. | Value  | Name     | Orient. | Pin | X     | Y     | Name   |
| -----  |        |          |         |     |       |       |        |
| RL1    | TQ2-5V | 10DIP300 | (180)   | 1   | 80.64 | 56.51 | vcc    |
|        |        |          |         | 2   | 78.10 | 56.51 | n00026 |
|        |        |          |         | 3   | 75.56 | 56.51 | r1     |
|        |        |          |         | 4   | 73.02 | 56.51 | n00032 |
|        |        |          |         | 5   | 70.48 | 56.51 |        |
|        |        |          |         | 6   | 70.48 | 64.13 |        |
|        |        |          |         | 7   | 73.02 | 64.13 | n00002 |
|        |        |          |         | 8   | 75.56 | 64.13 | n00019 |
|        |        |          |         | 9   | 78.10 | 64.13 |        |
|        |        |          |         | 10  | 80.64 | 64.13 | n00043 |
| R1     | 4.3K   | RR04     | (0)     | 1   | 55.88 | 78.10 | a12    |
|        |        |          |         | 2   | 66.04 | 78.10 | gnd    |
| R36    | 8.2K   | RR04     | (270)   | 1   | 82.55 | 75.56 | n00011 |
|        |        |          |         | 2   | 82.55 | 65.40 | n00022 |

表 2:MENTOR GRAPH PCB CAD 设计软件的元件位置文件例子(部分)

```

# BOARD STATION COMPONENTS FILE FOR-
MAT 2.0
# Application: PACKAGE V8.2-5.1 Tue Aug 3 20:
26:09 PDT 1993
# date: Sunday August 27, 1995; 11:07:10
#
UNITS TN
#
# Component property is presented in the format (prop
_name, prop_value)
#
# Reference Part _ number Symbol Geometry Board _
location Properties

```

\* 国家 863 计划 CIMS 重点项目

收稿日期:1996 年 7 月 6 日

```

#
G1 pn-g G GG 1250000 12000000 1 0-
(REFLOC,"MM,-4.574,-0.495,0,BL,1.5,1.5,
0.254,std,1")
L3 pn-blm32a06 BLM32A06 LL 1750000 12900000 1
0-
(REFLOC,"MM,-1.0,-1.7,0,BL,1.5,1.5,0.254,
std,1")(INSTPAR,"600")
D4 pn-xz xz inductr 12600000 15000000 1 0-
(REFLOC,"MM,0.5,4.0,0,TL,2.0,2.0,0.254,std,
1")
IC21 pn-xc1736 xc1736 dip8 12050000 6750000 1 0-
(REFLOC,"MM,-0.4,-10.4,0,TL,2.0,2.0,0.254,
std,0")
D1 pn-wy wy wyg 18600000 16750000 1 180 -
(REFLOC,"MM,-0.0,-3.0,0,TL,2.0,2.0,0.254,
std,1")
D2 pn-wy wy wyg 13100000 16650000 1 180 -
(REFLOC,"MM,-0.5,-3.0,0,TL,2.0,2.0,0.254,
std,1")
R1 pn-res res rc0805 1000000 18300000 1 0 -
(REFLOC,"MM,4.0,3.0,0,CR,2.0,2.0,0.254,std,
1")(INSTPAR,"43")

```

## 2.2 电子 CAM

用于电路板生产的数控设备有光绘机,数控打孔机,数控插件机、数控点胶/点膏机、自动贴片机、自动测试仪或测试机床等。这些设备往来自不同的年代,不同的生产厂家,差别也较大,需要人工编程。CIMS工程的要求,就是要从CAD设计到数控程序的自动化,要求尽量少地人工干预,数据的传递,也应符合各设备的要求,一般以网络和串行通讯方式。

不同的数控设备要求不同的数据信息,有些数据不能直接从设计文件中提取,但总的来说数据信息包括元件信息、元件位置和方向信息以及工艺信息,不同的设备必须对这些信息进行筛选和集成。例如插件机要求剔除表面贴的元件,相反,贴片机、点胶/点膏机却只要保留贴片元件。同时,根据工艺要求,需要对特殊元件进行人工处理。例如,某些元件使用了替代元件;某些通孔元件需用人工插件等。这些问题,由工艺人员制定的工艺文件进行说明。工艺文件是生产过程中不可缺少的文件,同时也是数控设备编程时不可缺少的依据之一,工艺文件可由人工输入或辅助程序加人工输入产生。工艺文件被打印成硬拷贝即为工艺卡,工艺卡将作为生产文档保存。

## 3 电子 CAD/CAM 集成系统结构

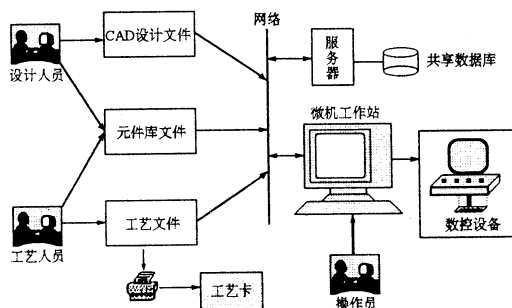
电子 CAD/CAM 集成系统结构如下图所示。

数据信息的来源可分为以下三个途径。

- (1) CAD 设计文件;
- (2) 工艺文件;

### (3) 人工输入。

CAD 设计文件由 CAD 设计人员提供,从 CAD 设计文件可提取元件位号、名称、封装、摆放方向等信息。



工艺文件反映了生产上的具体要求,不同的设备需要不同的工艺文件,有些设备不需要工艺文件或工艺数据可以间接地从 CAD 设计文件中提取(如根据焊盘的大小按工艺要求可计算出点胶/点膏的标准剂量),而有些设备需要人工确定工艺,如某些元件使用了代用品等信息。

值得一提的是,设计人员和工艺人员须共同制定一个标准元件库,元件库提供元件信息,如元件的型号规格、元件封装、元件的引脚及其相对元件参考点的坐标、元件的原始参考点、原始摆放方向以及其它特定设备所需的工艺数据,如元件的高度、元件引脚的剪切长度或打弯角度等。由于元件信息是相对固定的,因此最终可形成标准元件库。这样,对元件信息的任何更改将被自动带入新的设计或工艺中,而不必修改设计文件或工艺文件,以保证数据的一致性。

有些信息需由操作人员及时输入,如日期、根据厂标规定的板号、确省值的修改等可由操作员通过键盘输入或更改。

上述数据中,从 CAD 设计文件和工艺文件提取的数据作为全局数据通过网络存储到一个共享数据库中,各设备的 CAD/CAM 集成子系统通过访问该共享数据库获得设计、工艺数据和元件库数据。由操作人员输入的数据作为私有数据存放在本地微机机中,全局数据和私有数据集成,产生数控设备的编程数据,这些编程数据以文件形式通过网络或串行通讯传递到各设备的控制计算机中。

对于数据量较少的集成系统,也可以采用文件共享的方式实现数据共享,以节省数据库服务器的开支,但效率比数据库共享方式差。同时,在 CIMS 工程中往往并非仅仅考虑 CAD/CAM 集成的需要,而是综合考虑整个工厂的信息管理系统对网强数据库的需要。从开发的角度看,使用共享数据库可以一次提取全部的数据,而不必每种设备分别提取数据,从而使效率更高,更有利于保证数据的安全性、一致性和可维护性。

## 4 CAD/CAM 集成系统中的责任数据

电子工厂在传统的生产过程中,从设计到生产的信息传递和复制是由图纸的传递和复制来实现的,图纸除了包含设计信息和工艺信息外,还包含了设计者、审查者、复核者等责任信息。因此,图纸既是一个设计文件或工艺文件,同时也是一个责任文件。通过 CIMS 工程实现数据集成后,为明确责任和保证数据的权威性和有效性,责任数据的产生、保存和传递是一个不可忽视的问题。

在 CIMS 工程的电子 CAD/CAM 集成系统中,需建立一个责任数据库,该数据库由各责任人填写或修改各自责任和权限范围内的数据。当一个部件或 PCB 板的全部设计和工艺完成并填写了责任数据之后,由总工程师或责任工程师作最后的审查和确认。此后,该责任数据及其所对应的设计和工艺数据即作为一个版本统一并保存起来,不允许再进行任何的修改。从而保证数据的可靠性、安全性和一致性。

## 5 结束语

实现 CIMS 工程的关键是数据的集成和解决设计

人员、工艺人员和操作人员协同工作的环境问题,电子 CAD/CAM 的集成系统除了上述数控生产设备外还包括元件和 PCB 板的自动测试以及热分析、电磁分析等各种测试和分析仪器,这些仪器均需与 CAD 的数据进行集成。此外,集成系统需充分考虑各岗位人员协同工作的要求。在国家 863 计划 CIMS 重点应用工厂项目——武汉邮电科学研究院 CAD 工程中我们采用上述设计思想较好的解决了这些问题。

## 参考文献

- 1 Waimwright C E R, Harrison D k, Leonard R, CAD/CAM and DNC Links as a Strategy Manufacturing; Implications for CAPM, Computing & Control Engineering Journal, 1992, 2(3): 19-22
- 2 单吉弟,王国仁,于 戈,郑怀远, CIMS 环境下的数据集成技术, 东北大学学报, 1995. vol. 16(6): 623-627
- 3 应国旺, CAM 对 PCB 生产的作用, 电子技术, 1995, (7): 20-21

# 优化 Foxpro 菜单管理的一种策略

陆 遥

(广西师范大学数学与计算机科学系 桂林 541004)

**摘 要** 本文设计了一种菜单库结构,并以此为基础,提出了优化 Foxpro 菜单管理的一种策略,它可以方便具有模块化层次结构的事务处理软件的开发与维护工作。

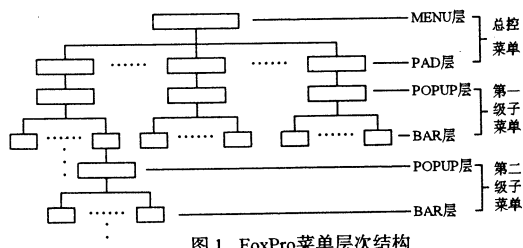
**关键词** 菜单 菜单库 数据库管理系统 事务处理 软件维护 Foxpro

用数据库管理系统开发过事务处理软件的人都知道,这类软件的结构是典型的模块化层次结构,模块的调用与被调用之间的控制关系非常清晰,非常符合软件工程学中对较大型软件在结构上的设计要求。这类软件涉及到的处理功能通常很多,而且功能的选用又完全由用户自由决定,故这类软件中普遍采用“菜单——点菜”的方式方便用户进行功能选择。由于各功能模块的处理功能通过菜单呈现在用户面前,所以系统的菜单结构直接反映了系统的模块结构。由此可见,菜单在事务处理软件的设计中占有重要的地位。

软件工程学还为我们揭示了一个事实,即软件维护工作是软件整个生命周期中时间最长、耗费人力和财力最多的一项工作。对事务处理类软件,一方面由于需处理的事务纷繁复杂,软件在设计阶段可能考虑不完善,另一方面,随着软件应用的深入,用户也可能提出新的处理要求或取消某些已无价值的处理功能,这就必须对现有的软件进行更新性维护。由于采用的是

模块化的设计方法,功能的增删实质上就是功能模块的增、删,但这会影响整个系统的模块结构,自然也就影响到系统的菜单结构;若是删除一级控制,或在两级控制之间插入一级控制,则系统结构和菜单结构受到的影响将更大。如何在系统结构发生较大变化时,维护人员对软件的修改量尽量小,这是软件开发人员要注意的问题,也是降低维护开支的一个方面。

Foxpro 是一种较先进的关系型数据库管理系统,它提供了一套层次分明的有效的菜单制作方法。Foxpro 的菜单层次结构如图 1 所示,这种层次结构是本文所讨论的菜单管理策略的基础。



一般的菜单管理方法是,将菜单的有关信息和用户“点菜”后需调用的模块名等直接写在菜单管理模块中,因此,在对系统功能作调整时,就要修改各级菜单管理模块,增加了维护工作量。要解决这个问题,首先要使菜单及功能调用方面的信息独立于菜单管理模块,最有效的办法就是利用数据库文件来存放这些信息,该文件可称为“菜单库”,而菜单管理模块则从菜单库中提取信息生成菜单并实现功能调用。以下给出笔者设计的菜单库结构及菜单管理程序。

## 1 菜单库结构及说明

### 1) 菜单库结构

| Field | Field Name | Type      | Width | Dec |
|-------|------------|-----------|-------|-----|
| 1     | CDFLM      | Character | 1     |     |
| 2     | CDMC       | Character | 4     |     |
| 3     | CDXX       | Character | 24    |     |
| 4     | SJCDM      | Character | 4     |     |
| 5     | XJCDM      | Character | 4     |     |
| 6     | DYMKM      | Character | 6     |     |
| 7     | CDZB       | Character | 5     |     |

图2 菜单库结构

### 2) 字段及使用说明

(1)CDFLM——菜单分类码。笔者根据 Foxpro 定义菜单的特点,将菜单结构中的三个结构层次——PAD 层,POPUP 层和 BAR 层——分别定义分类码为 1,2 和 3,不分子菜单级别。

(2)CDMC——菜单名称。对各 PAD 或 POPUP,此为对应的 PAD 名称(pad name)或 POPUP 名称(popup name),对 BAR,此为对应的 BAR 序号。

(3)CDXX——菜单条目信息。为对应菜单条目的提示(prompt)信息。

(4)SJCDM——上层菜单名。菜单结构中各层次的关系见图 1。对各级子菜单,其 BAR 与 POPUP 之间的关系在本库中的处理是一致的。

(5)XJCDM——下层菜单名。

注意:若一个 POPUP 为某个 BAR 的下级,则该 POPUP 没有上级菜单名,而任一层上的 POPUP 均无下级菜单名。这是根据 Foxpro 定义菜单时的特点确定的。

(6)DYMKM——调角模块名。给出用户选定某项功能后需调用的模块名。只有无下级子菜单的 BAR 才需填入此项。

(7)CDZB——菜单坐标。格式为:(行号,列号)。只有第二级(包括第二级)子菜单以下的各级子菜单才需此项,用以确定下一级 POPUP 的左上角坐标。

### 3) 菜单库示例

| Record# | CDFLM | CDMC | CDXX    | SJCDM | XJCDM | DYMKM | CDZB |
|---------|-------|------|---------|-------|-------|-------|------|
| 1       | 1     | zxt1 | 系统数据管理  | xted  | sigl  |       |      |
| 2       | 1     | zxt2 | 各类信息查询  | xted  | xxcx  |       |      |
| 3       | 1     | zxt4 | 系统服务    | xted  | xtfw  |       |      |
| 4       | 2     | sigl |         | zxt1  |       |       |      |
| 5       | 2     | xxcx |         | zxt2  |       |       |      |
| 6       | 2     | xtfw |         | zxt4  |       |       |      |
| 7       | 3     | 1    | 1 发排单管理 | sigl  |       | fpdgl |      |

|    |   |      |            |      |      |        |     |
|----|---|------|------------|------|------|--------|-----|
| 8  | 3 | 2    | 2 付印单管理    | sigl |      | fydgl  |     |
| 9  | 3 | 3    | 3 进度表管理    | sigl |      | jdbgl  |     |
| 10 | 3 | 4    | 4 各类代码文件管理 | sigl | dmwj |        |     |
| 11 | 3 | 1    | 1 查询发排单    | xxcx |      | cxfpd  |     |
| 12 | 3 | 2    | 2 查询付印单    | xxcx |      | cxfyd  |     |
| 13 | 3 | 3    | 3 查询进度计划   | xxcx |      | cxjdjh |     |
| 14 | 3 | 1    | 1 操作员管理    | xtfw |      | czygl  |     |
| 15 | 3 | 2    | 2 修改口令     | xtfw |      | xgkl   |     |
| 16 | 3 | 4    | 4 重建系统索引   | xtfw |      | cjsy   |     |
| 17 | 3 | 5    | 5 退出运行     | xtfw |      | tcyx   |     |
| 18 | 3 | 1    | 1 工厂代码文件   | dmwj |      | gcdm   |     |
| 19 | 3 | 2    | 2 图书门类代码文件 | dmwj |      | tsmldm |     |
| 20 | 2 | dmwj |            |      |      |        | 5,6 |
| 21 | 3 | 3    | 3 系统日期设置   | xtfw |      | setdat |     |

图3 菜单库示例

本例中,最高层——MENU 层的名称为“xtcd”;最低一级子菜单层次为二级,其 POPUP 名为“dmwj”。

### 4) 菜单库的使用

使用菜单库时,需要对其索引,索引关键字是:CDFLM+SJCDM。

## 2 菜单管理程序

```

set sysmenu off
private menun,padn,popupn,barn
private propt,popr,popc,tcdmc,tcdxx
use cdk && 打开菜单库文件
set order to tag cdsy && 打开菜单索引
go top
menun=sjcdm && MENU 名->menun
define menu &menun bar in screen
do while. not. eof()
tcdmc=alltrim(tcdmc)
tcdxx=trim(tcdxx)
do case
case cdfm='1'&& 以下定义各 PAD
padn=tcdmc
propt=tcdxx
define pad &padn of &menun;
prompt propt+"color scheme 1
popupn=alltrim(xjcdm)
if len(popupn)#0
on pad &padn of &menun;
activate popup &popupn
endif
case cdfm='2'&& 以下定义各 POPUP
popupn=tcdmc
if len(alltrim(cdzb))#0
popr=val(alltrim(cdzb))
popc=at(',',cdzb)
popc=val(substr(cdzb,popc+1))
define popup &popupn from;
popr,popc marg shad colo sche 8
else
define popup &popupn margin;
shadow color scheme 8
endif
case cdfm='3'&& 以下定义各 BAR

```

```

barn=val(tcdmc) && 取 BAR 序号
propt=tcdxx
popupn=alltrim(sjcdm)
define bar barn of &popupn;
prompt'\(' +propt color scheme 5
propt=alltrim(xjcdm)
if len(propt) # 0 && 若有下级菜单
on bar barn of &popupn;
activate popup &propt
endif
propt=alltrim(dymkm)
if len(propt) # 0 && 若有调用模块
on selection bar barn;
of &popupn do &propt
endif
endcase
skip
enddo
activate menu &menun
do while. t.

```

```

if' 退出运行' $ prompt()
exit
else
activate menu &menun
endif
enddo
close data
release menu &menun
set sysmenu on
set sysmenu to default
return

```

以此菜单管理程序与菜单库配合,当需要在系统中增、删功能时,只需简单地在菜单库中增、删菜单记录即可,对菜单结构的修改或对调用功能的变更,也只需对菜单库操作,不涉及程序的修改。

菜单库除可用于优化菜单管理之外,还为实现操作员的操作授权管理打下了基础,有兴趣的读者不妨一试,本文在此不作讨论。

## 怎样正确使用 Browse 命令防止窗口显示混乱

王晓武

(解放军总后勤部司令部 北京 100842)

Foxpro for Windows 数据库系统中,最常用、最灵活、最能发挥程序员想象力的命令无疑是 Browse。Browse 窗口命令具有非常丰富的可供程序员选择使用的参数,正确巧妙地应用这些参数,数据库程序员能够编写出简洁、美观、灵活多变的高效通用窗口程序。然而,正是由于 Browse 命令的功能和灵活性如此之强,编程时使用若不严谨就可能出现一些不可预测的错误,最常见的问题就是引起屏幕显示窗口的混乱。

许多程序员可能碰到过这种情况,当第一次调用 Browse 窗口时,屏幕窗口及数据信息一般均能按程序员的设计正确的显示,而当程序多次调用和退出 Browse 窗口以后,屏幕窗口信息常常会发生变化,时大、时小、位置也可能发生拉伸、压缩等漂移现象;还有一种常见的情况是,一个完整的 Foxpro 程序,在调用 Browse 窗口之前,其父窗口信息显示正常,而当调用返回或反复几次后,其父窗口的位置及内容很可能产生混乱;另有一种典型的情况,当一个完整的 Foxpro 程序调试编译运行正常后,一旦装入另一台电脑运行时,屏幕显示信息就会发生变化,与原来的设想并不一致,产生屏幕显示混乱错误,致使应用程序无法正常运行。

程序员在开发应用程序时如何避免产生上述错误

呢? 首先必须弄清产生这种错误的原因。我们知道,完整的 Foxpro for Windows 系统中,含有一个 FoxUser.dbf 数据库文件(不同于普通的 .dbf 文件结构)及其相关的备注文件 FoxUser.fpt,这两个相关的文件称作 Foxpro 的资源文件,此资源文件具有特殊的结构,并存储着 FoxPro 各项系统资源的相关信息,其主要内容包括:

- 系统窗口(命令窗口、计算器、日历等)的大小,位置等信息
- Browse 窗口的大小、颜色及窗口控制菜单的参数值
- 颜色集(CoLoR Sets)的定义
- 标签格式的定义
- 项目管理程序、文字编辑程序及表达式产生器的相关定义

正是由于上述资源文件中 Browse 命令的相关参数缺省值,在不同的系统环境下很难保证一致性,并且每一次 Browse 窗口关闭时,资源文件中其相关参数均会自动更新,而程序员开发的 Foxpro 程序并不依附于一个固定的系统资源文件。因此,程序开发时如果忽略了 Browse 命令在不同系统环境及父、子窗口环境下参数的一致性,就必然会发生前述屏幕混乱现象。比如,一般的 Foxpro 程序的主菜单窗口均采用修改系统窗

口的方法定义,若应用程序从主菜单的某一菜单项调用一个 Browse 窗口,此时,若 FoxUser.dbf 系统资源文件记载的系统窗口字体和 Browse 窗口字体不同,而程序进入和退出窗口又不进行相关的处理,当 Browse 窗口返回父窗口系统菜单状态时,必然会产生整个系统菜单大小、位置及内容的混乱现象。

弄清了产生错误的原因,编程时有针对性地正确使用 Browse 的命令参数并进行相关保护处理,即可避免错误现象的发生,实际工作中笔者摸索了一套从根本上避免错误,正确使用 Browse 命令的方法,本文介绍如下,仅供读者参考。

Browse 窗口命令参数中,与屏幕显示有关的参数主要是,窗口的起始坐标、字体的大小、窗口的标题及字体,显示信息的颜色等(参阅例程清单 example.prg)。当应用程序欲调用一个 Browse 窗口时必须用 MODIFY WINDOWS SCREEN 等命令修改屏幕参数的设置,同时严格定义 Browse 窗口的起始坐标、字体及颜色。当 Browse 窗口关闭返回上层父窗口(一般是系统窗口)时,要特别注意用 MODIFY WINDOWS SCREEN 及 MODIFY WINDOWS SCREEN TITLE 命令还原父窗口屏幕参数设置及标题字串,只有这样严谨编程,Browse 窗口无论调用多少次,也无论在任一台电脑或系统环境下运行,应用系统均不会发生屏幕混乱现象,另外要强调的一点是,Browse 窗口作为父窗口激活一个或多个子窗口或子 Browse 窗口时,其子窗口关闭返回之前也必须用 MODIFY WINDOWS SCREEN、MODIFY WINDOWS SCREEN TITLE 以及 SET SYSMENU AUTOMATIC 命令还原 Browse 父窗口的有关参数设置。尤其是 Browse 窗口嵌套使用时,更需要仔细严谨地定义和恢复各窗口的有关参数。程序员只要严格按上述要求编程,Browse 命令就能准确无误地按程序员的无穷想象力发挥其强大功能。

#### EXAMPLE.PRG 例程清单:

(程序运行环境:中文 WINDOWS 3.2 及 Foxpro for Windows 2.5)

```
. && 主程序略
PARA DNAM,SUBMENU
DNAM=UPPER(DNAM)
@ 21,24 SAY SUBMENU;
FONT "大黑",2t;
STYLE"BT";
COLOR RGB(255,255,0...)
@ 24,10 GET m_hcx;
PICTURE "@* HN 查询;打印;维护;退出";
SIZE 1.6,7.5;
DEFAULT 1;
FONT"黑体",16;
VALID PROSS()
```

```
READ CYCLE
@21,5 CLEA TO 28,76
RETU
```

```
FUNCTION PROSS
do case
case m_hcx=1
DO SCRE WITH DNAM
RELE ARRAY ALL LIKE AH *
case m_hcx=3
NUM=1
HHHB=""
FIEXP=""
USE @DD
COUN TO DNUM FOR DB_NAME=DNAM
COPY TO ARRAY DD FOR DB_NAME=DNAM
USE
DO WHILE NUM<DNUM
FIEXP=FIEXP+TRIM(DD(NUM,2))+':H="' +
TRIM(DD(NUM,3))+','
NUM=NUM+1
ENDD
FIEXP=FIEXP+TRIM(DD(NUM,2))+':H="' +
TRIM(DD(NUM,3))+','
use &DNAM
MODI WINDOW SCREEN && 转入 Browse 窗口定义
MODI WINDOW SCREEN TITLE;
"数据维护—Browse 窗口"&& 修改屏幕标题
DEFINE WINDOWS BWIN;
FROM 1.000,1.000 TO 16.400,79.000&& 定义 Browse
窗口坐标
BROWSE FONT"宋体",10;
FIELDS & FIEXP;
TITLE SUBMENU;
LOCK 1;
NOMENU;
NORMAL;
WINDOWS BWIN
use
rele windows bwin
rele array dd
MODI WINDOW SCREEN && 修改屏幕定义
SET SYSMENU AUTOMATIC && 恢复父(系统菜单)
窗口定义
MODI WINDOW SCREEN TITLE;
"应用程序主菜单"&& 恢复父窗口标题定义
case m_hcx=4
CLEA READ
flag=0
retu
endcase
```

# 如何实现 Foxpro 和 Sybase 之间的数据传输

成卫青

(南京邮电学院计算机工程系 南京 210003)

**摘要** 本文讲述了 Sybase 和 Foxpro 之间的数据传输问题,重点介绍了使用 Bcp.Copy To 命令及 Edit 软件将 Foxpro 数据库内容复制到 Sybase 数据表的实现方法。

**关键词** Bcp Copy To 数据表

## 1 引言

Sybase 是一个开放的、高性能的、具有联机事务处理能力的关系数据库系统。它采纳了客户/服务器体系结构,将数据应用与数据库分离,相应分为前端进程和后端进程。这两进程可以运行在同一主机。SQL Server 上,也可以分别运行在通过网络联接在一起的两台主机上,从而把工作负荷分摊在两台机器上。

Sybase 实现的客户/服务器技术给客户带来了诸多好处,尤其是对于企业级用户。其 SQL Server 主要有如下特色:

- 允许集成第三方产品的开放性;
- 允许一个网络上有多个 SQL Server 的可伸缩性;
- 允许平台间移植的对多平台支持的能力;
- 采用多线索化技术从而使得多用户加入时,数据库服务器的性能基本保持不变;
- 数据库可编程,客户可以编写自己的存储过程,存储过程驻留在支持各种各样的客户程序的 SQL Server 中,因而多个客户程序可以调用同一存储过程,从而降低了维护要求提高了数据完整性;
- 支持远过程调用,允许存储过程调用别的 SQL Server 上的存储过程;
- 事务处理功能等。

近年来,随着计算机网络技术的不断进展,网络化的观念已深入人心,人们不再满足于信息管理的孤立自动化,而是希望同一部门的上下级单位之间乃至不同的部门之间能够方便地交换信息又能泾渭分明(对不同级别单位用户设置不同的访问权限),做到既合也分。正因为此,Sybase 以其优越的性能越来越受到重视,应用也越来越普遍。

Foxpro 是一个编译型的集成环境式的数据库软件,它对数据安全性和数据完整的保证及对网络数据库的支持远不及 Sybase,但其具有成本低、编程简单、易调试和开发周期短等优点,对于安全性要求不太高的中小型应用场合,它还是颇受欢迎的。目前,用 Fox-

pro 开发的数据管理系统在数量上仍占优势,但是随着应用要求的提高,其局限性日益明显,已有相当数量的数据管理系统正面临彻底翻新的时期。那么,是不是什么都得从头开始呢,当然不是,任何在用的系统肯定都存在些以后仍要用的信息,比如某些数据库(如 Foxpro)的数据。若采用新的数据库系统(比如 Sybase)实现数据管理系统时,重新一条条录入原数据,费时费事不说,这项工作的枯燥一定会令人生厌至极。

下面就谈谈 Foxpro 与 Sybase 之间的数据传输。

## 2 Foxpro 到 Sybase 的数据转换

先说明 Sybase 数据库的概念,数据库是与一特定主题或目的有关的数据的集合,一个数据库可包括表、视图、索引等数据库目标(objects)

Foxpro V2.6 及以下版本的数据库基本对应于 Sybase 的数据表的概念。其库结构对应表的表定义。其字段对应表的列,而记录对应表的行。Visual Foxpro 3.0 的数据库及表的概念和 Sybase 的相似,本文叙说针对 3.0 以前的版本,但对它也同样适用。

要将 Foxpro 数据库中的数据复制到 Sybase 的表中。这里介绍两种方法:一个通过编程实现。另一个利用各自的命令或实用程序把文本文件作为中介来达到目的,下面叙述。另外,下文例子在装有 PC/DOS 版的 Open Client DB—Library(开放式客户 DB 函数库)和 PC/Windows 版的 Open Client DB—Library 的机上试行通过,对其它版本环境可作参考不可照搬。

### 2.1 编程法

使用具有 ODBC(开放式数据库连接)API 的开发工具,如 PowerBuilder 或 Visual Basic. VB 中,可通过 OpenDatabase 函数分别打开 Foxpro 和 Sybase 数据库,再用数据库的 Execute SQL 方法在 Sybase 库中创建与 Foxpro 库结构相当的表,之后从 Foxpro 库逐条读出记录并插入(insert)Sybase 表即可。

### 2.2 快速成批拷贝法

这种方法速度快也简单易行,除对含有不太常用的 Memo 型或 Picture 型字段的库无能为力以外都很行之有效。例如把 Foxpro 库 codes.dbf 的数据复制到



Sybase 数据库 kjgl 的表 codes 中,步骤如下:

(1)用 Foxpro 的 COPY TO 命令把待转换数据库的数据复制到一操作系统文本文件上,COPY TO 的用途是把当前被选用的数据库的数据复制到另一数据库或由 TYPE 等关键字规定的其它类型的文件,如 Lotus 表格、Excel 表格和 ASCII 文本文件等。

COPY TO C:\tmp\codes.txt TYPE DELIMITED

得到名为 codes.txt 的文本文件,其中一行存一条记录,字段间以逗号相隔,并且对字符型字段用双引号限界。

(2)Foxpro 逻辑型字段拷贝到文本文件是 'F' 或 'T',而 Sybase 的是 0 或 1;字符型一个带双引号而另一个没有。总之执行 COPY TO 命令得到的结果不可直接用来拷贝到 Sybase 的数据表,需做适当修改。使用全屏幕编辑软件如 DOSV5.0 以上提供的 EDIT 软件,选用 Search 菜单下的 Change... 命令依次作下列替换:

若有逻辑型字段且不是最后一个字段,则

,,=》,0,(仅用于 Foxbase)

,T,=》,1,

,F,=》,0,

若有逻辑型字段且是最后一个字段,则

手工在逗号是一行最后一可见字符的行尾添加字符 F(仅用于 Foxbase)

,T=》,1

,F=》,0

删除双引号,即“=”空串

(3)进入安装有 PC/DOS 版的 Open Client DB—Library 的目录的 BINR 子目录,那儿有几个 Sybase 实用程序:ISQL、BCP 和 DEFNCOPY。由数据库属主(dbo)运行 ISQL,创建新表 codes,其表结构必须与 codes.dbf 的库结构相一致。

1>CREATE TABLE kjgl.dbo.codes(...)

2>go

(4)Sybase 的实用程序 BCP(bulk copy)的功能是以用户指定的形式将数据库表拷贝到操作系统文件

中,或从操作系统文件中拷贝一数据库表,要实现快速成批拷贝,必须由数据库属主或系统管理员将数据库选项——select into/bulkcopy 置为真,新建数据库该选项缺省为假。改变过程如下:

1>use master

2>go

1>sp\_dboption kjgl"select into/bulkcopy",true

2>go

1>use kjgl

2>go

1>checkpoint

2>go

然后执行 BCP 实用程序:

bcp kjgl.codes in c:\tmp\codes.txt -c -t, -r\n -Usa

其中,参数 in 表示从文件到数据表的拷贝,拷出参数是 out;

-c 表示以字符类型执行拷贝操作;

-t 指定逗号作为缺省的域(列)终止符;

-r\n 指定回车换行符作为缺省的行终止符;

-U username 指定一个注册名,对于本例可以是系统管理员或数据库属主。

### 3 Sybase 到 Foxpro 的数据转换

把 Sybase 的数据表中数据复制到 Foxpro 的库文件,同 2 小节类似也有上述两种办法,编程法请查阅有关参考书。另一种先用 BCP 将数据表的数据复制到一文本文件中,再进行适当修补,使其成为 Foxpro 可接受格式,主要修改原日期型、bit 型和字符型列,最后在 Foxpro 环境中创建合适的库结构,使用与 COPY TO 相对的命令 APPEND FROM 将文本内容拷入库中,具体细节这里不再赘述。

### 4 结束语

本文详细地讲述了 Foxpro 到 Sybase 的数据传输,希望由此能给读者一点启发,那就是要灵活地运用系统软件提供的功能,并充分发挥想象力,注意事物间的相互联系。

## INFORMIX 在京展示数据库新技术

〈本刊讯〉英孚美软件(中国)有限公司近期已在北京新世纪饭店举办了 INFORMIX—Universal Server 产品发布会。会上,进行了 ORDBMS 的回顾与展望,INFORMIX—Universal Server 产品介绍,DataBlade 产品介绍,INFORMIX—Universal Server 案例分析和演示。

新一代数据库 INFORMIX—Universal Server 特有的 DataBlade 技术,能使所有类型的数据(正文、图像、多媒体、几何空间数据、Web 页面、时间序列等)放到单一、集成化的关系数据库中从而便于管理的功能,吸引了来宾。

世界著名的数据库厂商 INFORMIX 公司除在北京设立中国公司外,还设立了上海、广州和成都分公司,力拓中国市场。INFORMIX 公司资深副总裁 jeff Hudson、大中国区业务发展总监丘国梁及数据库专家、北京大学计算机系唐世渭教授、杨冬青教授出席了发布会。(英文)

## 用 FOXBASE 播放 WINDOWS 图像

何明昌

(梧州师范专科学校 广西贺县 542800)

**摘 要** 本文论述了由 Windows 制作的图形在 Foxbase 中播放的原理,并提供了两种具体实施的方法。

**关键词** BMP PCX 图形 图像

本文介绍能在 FOXBASE 下播放 WINDOWS 图像的两个具体方法,使得用 WINDOWS 画笔描绘的丰富图案得以在 FOXBASE 中实现,从而使广大用户自己的 FOXBASE 应用程序更加绚丽多姿。

实现原理:UCDOS3.1 具备强大的屏幕图像操作功能,它包括图像的保存,显示以及切割等功能。其使用的图形图像文件是标准的 PCX 格式文件。而 WINDOWS3.1 附件中的画笔除了支持 BMP 文件外,亦支持标准的 PCX 图形文件,因此,通过 UCDOS 屏幕抓图保存的图像文件,便可在 WINDOWS 的画笔中加以修改及编辑。另外,用 WINDOWS 画笔描绘的图形,如果能改为 PCX 标准格式存盘,则也可通过 FOXBASE 来调用 UCDOS 的屏幕特显功能。这样,至少有两种方法可在 FOXBASE 中,通过 UCDOS3.1 的特显功能来显示 WINDOWS 画笔所绘制的图形。

实现方法如下:

方法一:

先用 UCDOS 的屏幕抓图功能,形成一个标准的 PCX 文件,然后送 WINDOWS 中的画笔进行编辑。

操作步骤:

1、用含下列内容的批处理文件启动 UCDOS3.1

```
@ECHO OFF
CD\UCDOS
RD16
KNL
WB
RDSL
TX
CD\
```

2、启动 UCDOS3.1 后,然后按<CTRL>+<Print/Screen>,再按回车键,此时该目录下会有一个名为 ucimg000.pcx 的 PCX 格式图形文件。这个图形文件的具体图形是什么并不重要,因为图形还要到 WINDOWS 的画笔中描绘,这里只要它提供一个 PCX 格式。

3、按<CTRL>+<F5>退出 UCDOS,启动中文 WINDOWS3.1,在附件分组中激活画笔图标后,调入步骤<2>中的 UCIMG000.PCX 文件,进行图形图像编辑。编辑完后,可用原文件名保存,也可用另外的文件名保存。在本文中,假定文件名为 PHOTO.PCX,然后退出 WINDOWS。

4、此后,只要按步骤<1>的批处理文件方式启动 ucdos3.1,即可把步骤<3>中 PHOTO.PCX 保存的图像在 FOXBASE 程序中播放。这里主要是利用了

UCDOS3.1 的显示 PCX 图像功能[格式为:RPx,y,w,h,f 其中(x,y)=左上角坐标,w=宽度,h=高度,f=文件名(以\$结尾)]下面是一段调用的 FOXBASE 程序实例:

```
clear all
set talk off
set stat off
set scor off
set menu off
set prin to lpt3
set devi to prin
@0,0 say chr(14)+'[m10 c10x0f cul,0]'
set colo to 7/0
clear
@0,0 say chr(14)+'[rp0,0,300,300,photo.pcx ¥]'
wait''
@0,0 Say chr(14)+'[m10 c10xof cul,1]'
set print to lpt1
set devi to scre
clear
```

5、注意,由于 WINDOWS3.1 画笔中默认的图形文件是 .BMP 位映像格式文件,所以在 WINDOWS 的画笔下绘制的图形直接存盘是不能被 UCDOS 所用,即使存盘时把扩展名改为 .PCX 亦行不通。

方法二:

直接启动中文 WINDOWS3.1 附件中的画笔进行编辑,但存盘时应由原来默认的 .BMP 格式文件转换为用 PCX 格式文件即可。

操作步骤:

用 WINDOWS 的画笔绘制好图形后,按<ALT>+<F>激活文件项目,选中保存(S)一栏后,再按<ALT>+<T>(选择按类型文件保存文件),选中 PCX 文件一栏,此时,WINDOWS 画笔便把原来默认的 BMP 位映像文件自动转换为标准的 PCX 图形文件,然后填上文件名 PHOTO(扩展名 .PCX 可要可不要,系统会自动加上),此时存盘的图形文件是标准的 PCX 文件。

接着,按方法一中的第<4>步即可实现 WINDOWS 下的图形在 FOXBASE 中播放。

上述方法在兼容机 386/DX,VGA 彩显,2 兆内存上运行通过。

软件环境: DOS6.2 中文 WINDOWS3.1 UCDOS3.1 FOXBASE2.1

### 参考文献

- 1 何明昌. 用 Foxbase 制作立体投影汉字,《中国电脑教育报》,第 132 期。
- 2 王鉴泉 赵清泉.《中文 Windows 使用速成》,清华大学出版社。

# 向程序管理器增加程序组和程序项的简易方法

时丽珉 谢壮宁

(汕头大学工学院 汕头 515063)

**摘 要** 本文介绍两种在应用程序中向程序管理器增加程序组和程序项的方法,方法简单明了,可供有关人员参考。

**关键词** 程序管理器 增加程序组 添加程序项

## 1 引言

对于普通的 Windows 用户,在 Windows 环境下很容易采用用手键入的方式向程序管理器增加程序组,并且可在任一程序组添加程序项。另一方面,我们注意到:在安装 Windows 的过程中,SETUP 会自动在程序管理器内增加若干的程序组(如主群组、附件、应用程序、游戏等等),一般的 Windows 应用程序(如 Word、Excel 等)在安装时也会自动在程序管理器下增加一个或多个程序组,且在每个程序组内建立多个程序项。因此如何自动向程序管理器增加程序组和程序项,可能是用户尤其是软件开发者更为感兴趣的话题。

本文介绍两种在应用程序中向程序管理器增加程序组和程序项的方法,这两种方法均采用 Visual Basic 实现,与其他作者介绍的采用 C 语言实现的方法相比,本文介绍的方法更为简单,更加适合于一般的软件开发者。

## 2 实现方法

### 方法一:

Windows 的一项强大功能就是应用程序之间具有“动态数据交换”(Dynamic Data Exchange 简称 DDE)功能,利用此项特性一个应用程序可以接收别的应用程序传递过来的数据和命令;反之它也可以传递数据和命令给任何运行中的应用程序(当然该应用程序应该具备有接收和处理数据的能力,并且事先要在两应用程序之间建立链接关系)。因此在建立起应用程序和程序管理器的链接后,就可以直接向它发送命令建立新的程序组和添加程序项,以下程序执行后在程序管理器内新建一标题为附件一的程序组,并在该程序组上增添:书写器、画笔、终端仿真程序、记事本、记录器五个程序项。(需在窗体内先建立一名为 Text1 的文本框控制。有关 DDE 的使用方法,限于篇幅不可能一一介绍,请参考有关资料)

程序清单如下:

VERSION 4.00

窗体特性表略

```
Private Sub Form_Load()
```

```
Text1.LinkTopic="ProgMan1 Progman" '建立 Text1  
控制和程序管理器的 DDE 对话关系'
```

```
Text1.LinkMode=2 '设定链接模式(2 为冷链接)
```

```
Text1.LinkExecute"[CreateGroup(附件一)]" '建立标题  
为附件一的程序组
```

```
Text1.LinkExecute"[AddItem(WRITE.EXE,书写器)]"  
'增添一新的程序项
```

```
Text1.LinkExecute"[AddItem(PBRUSH.EXE,画笔)]"
```

```
Text1.LinkExecute"[AddItem(TERMINAL.EXE,终端  
仿真程序)]"
```

```
Text1.LinkExecute"[AddItem(NOTEPAD.EXE,记事  
本)]"
```

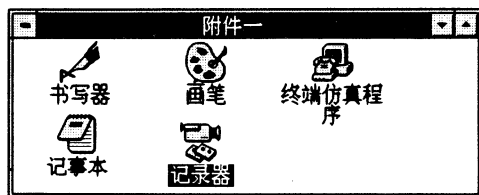
```
Text1.LinkExecute"[AddItem(RECORDER.EXE,记录  
器)]"
```

```
Text1.LinkMode=0 '终止对话关系
```

```
End '结束
```

```
End Sub
```

其执行结果所产生的程序项窗口见下图:



### 方法二:

方法一非常简单,在编程时应以首选考虑采用,在该方法中程序组和程序项的建立和增添是通过程序管理器的 CreateGroup 和 AddItem 两个函数实现的,其前提是必须了解同程序管理器进行 DDE 所能采用的这两个函数,要进行 DDE 通常需要对应用程序有更多的了解,手边要有有关该应用程序的详细文档。不采用 DDE 方法而利用 VB 提供 SendKeys 函数同样可以完成上述过程。通过 SendKeys 函数,在应用程序中可给其它运行中的应用程序(包括其自身)发送消息,控制程序的执行犹如手工控制一样。

仿照人工建立程序组和增添程序项的过程步骤,可构造出与方法一相应的两个函数 CreateGroup 和 AddItem,但注意首先要先用 AppActivate 将程序管理器变为活动窗口。

利用 SendKeys 建立程序组和增添程序项的执行

结果和方法一样,所不同的是该程序需编译成运行文件后才能正确执行(只要加一些 API 函数对窗口进行控制也可以解释方式运行)。在程序中的不少处加有 DoEvents 语句但被注释掉,如果加此语句,则在执行过程可仔细观察到每用 SendKeys 发送一个消息,就可以在屏幕上看到程序管理器的响应,犹如手工操作一样,否则只能看到一最终的结果(这和 Windows 本身不是真正的多任务环境有关,一个 Windows 应用程序在执行一个过程时,他实际上是独占了所有的 CPU 时间,虽然它在不断地给其它应用程序发消息,但由于其它应用程序分配不到 CPU 时间而一时无法响应,在发完消息后加入 DoEvents 正是起到可把控制转给响应程序的作用)。

程序清单如下:

程序清单,本程序需编译成运行文件后才能正确

执行

```
VERSION 4.00
'窗体特性表略
Dim CMD(10) As String
Const TabKey="{Tab}"
Const EnterKey="{Enter}"
Const AltKey="%"
```

```
Private Sub Form_Load()
  CreateGroup "附件一"
  AddItem "书写器", "WRITE", ""
  AddItem "画笔", "PBRUSH. EXE", ""
  AddItem "终端仿真程序", "TERMINAL. EXE", ""
  AddItem "记事本", "NOTEPAD. EXE", ""
  AddItem "记录器", "RECORDER. EXE", ""
End
End Sub
```

```
Private Sub CreateGroup(Group Name As String)
  AppActivate "程序管理器" '如果是西文 Windows 则应为
  AppActivate "Program Manager"
  CMD(0) = AltKey + "FN"
```

```
CMD(1) = "G" + EnterKey
CMD(2) = GroupName
CMD(3) = TabKey + TabKey + EnterKey
For i = 0 To 3
  DoEvents
  SendKeys CMD(i)
Next i
DoEvents
End Sub
Private Sub AddItem(Description As String, CMDLine
As String, Working DIR As String)
  CMD(0) = AltKey + "FN"
  CMD(1) = "I" + EnterKey
  CMD(2) = Description
  CMD(3) = TabKey
  CMD(4) = CMDLine
  CMD(5) = TabKey
  CMD(6) = Working _DIR
  CMD(7) = TabKey + TabKey + TabKey + EnterKey
  For i = 0 To 7
    'DoEvents
    SendKeys CMD(i)
  Next i
  DoEvents
End Sub
```

### 3 讨论

比较上述两种方法,从编程的可靠性和简洁性都应首选前者,但后者相对较为直观。应当指出的是并非所有的应用程序都具有 DDE 进行对话的能力,且在知道应用程序具备这种能力后,在使用时还需详细了解其数据交换的格式和应用程序所能接受的命令(这是一般 Windows 和 VB 使用文档所没有的),采用 DDE 实现和程序管理器除了本文用到的两个命令外还有一些命令如 ShowGroup、DeleteGroup 等等,但如果不了解这些命令及其用法就难以采用该方法;而采用 SendKeys 在理论上却可控制所有应用程序,它是模拟键盘对应用程序进行控制,只要知道使用方法就能控制一个应用程序,这是它的最为可取之处。

## 一个实用的下拉式菜单设计方法

雷霆

(焦作工学院图书馆 河南焦作 454000)

foxpro2.5 提供了丰富的命令与函数和菜单产生器 Menu Builder 用以制作菜单,给用户提供了较好的编写菜单条件。但笔者在实践中有一种更为简便的设计下拉式菜单的方法。

每当遇到以 foxpro 编写系统菜单程序时,笔者总是以自己已有的一个下拉式菜单程序为原型作些修改即成。这样,给编程带来了极大的方便,省时省力,而且还形成了自己的菜单风格。本方法简洁明了、容易掌握

及生成的菜单美观实用。本方法最为精炼的是利用宏替换选择执行项(过程),极大地缩减了程序代码行(本例若用 DO CASE 语句要增加近 150 行代码)。

现以一个企业 MIS 中的菜单程序(见附 1)为例介绍本方法。附 1 程序可在 foxpro2.5 和 UC DOS3.0 下运行。

读者可利用提供的程序稍加修改作为自己以后开发系统菜单程序的原型(最简单的只需修改研制者及研制者的声明。注:例中的研制者的声明由 ALT+L 激活)。在以后的使用过程中只要做以下几点修改即

可:

- (1)修改定义主窗口(WIN0)的标题内容;
- (2)修改定义各菜单项数组的个数及内容;
- (3)修改 menu bar 菜单条中菜单点的数目及 menu 中弹出式菜单中选择项的数目;
- (4)子模块名要代码化。如本例为 MK \*, 其中 \* 由 2 或 3 位数字代码组成 \*, 其值由菜单项所在的行和列决定,即 ltrim(str(ROW))+ltrim(str(COL))。

#### 附 1 程序清单

```
*****
* 名称:MAIN. PRG *
* 功能:主控模块 *
* 编程:雷霆 *
* 日期:94. 10. 26 *
*****
set escape off
set talk off
set color to
clear
set clock to 24,69
ROW=1
COL=1
K=0
do while. T.
set border to double
define window win0 from 0,0 to 23,79 title"单车租赁承
包管理系统";
COLOR SCHEME 1
activate window win0
@1,0 to 1,79 doub
@19,24 say "焦作工学院 雷霆"
if ROW=0. and. COL=0
K=0
endif
do while k=0
k=inkey()
set color to 7+/2+
@0,0 say"D-数据录入 C-帐务处理 S-数据查询 P-
帐表打印 W-系统维护"
enddo
if k=38
defi wind win00 from 5,20 to 15,59 title"严正声明"
acti wind win00
set color to GR+/R
@1,2 clear to 7,35
@2,4 say"本软件由焦作工学院高科技"
@3,4 say"开发公司研制,版权所有,严禁"
@4,4 say"复制!"
@5,4 say"研制人:雷霆"
@6,4 say"1994.10"
read
set color to W+/B
deac wind win00
```

endif

set color to 7+/+2,6+/+4

set mess to 24 center

dime TOP(5,2)

TOP(1,1)="D-数据录入"

TOP(2,1)="C-帐务处理"

TOP(3,1)="S-数据查询"

TOP(4,1)="P-帐表打印"

TOP(5,1)="W-系统维护"

TOP0="提示:请用光条进行功能选择,ESC 键拉起菜单。"

store TOP0 to TOP(1,2),TOP(2,2),TOP(3,2),TOP(4,2),TOP(5,2)

dime M-1(22)

M-1(1)="\" \* 日常单据 \* "

M-1(2)="运费单据"

M-1(3)="材料单据"

M-1(4)="燃料单据"

M-1(5)="转帐单据"

M-1(6)="取款单据"

M-1(7)="货款单据"

M-1(8)="互助金"

M-1(9)="其 它"

M-1(10)="\"

M-1(11)="退出系统"

M-1(12)="\"

M-1(13)="\" \* 月执行 \* "

M-1(14)="报停天数"

M-1(15)="周转量罚款"

M-1(16)="修理费"

M-1(17)="安全会罚款"

M-1(18)="贷款利息"

M-1(19)="其它扣款"

M-1(20)="\"

M-1(21)="超产税计算"

M-1(22)="规费处理"

dime M-2(12)

M-2(1)="试 算"

M-2(2)="修改单据"

M-2(3)="\"

M-2(4)="编制凭证"

M-2(5)="查阅凭证"

M-2(6)="打印凭证"

M-2(7)="\"

M-2(8)="记 帐"

M-2(9)="\"

M-2(10)="\" \* 本月结算 \* "

M-2(11)="帐目处理"

M-2(12)="报表计算"

dime M-3(14)

M-3(1)="存款车款"

M-3(2)="报停规费"

M-3(3)="\"

M-3(4)="\" \* 明细帐 \* "

M-3(5)="存款"

M-3(6)="租赁费"

M-3(7)="规费"

M-3(8)="贷款"

M-3(9)="互助金"

M-3(10)="车船税"

M-3(11)="保险费"

M-3(12)="押金"

M-3(13)="\"

M-3(14)="总帐"

dime M-4(13)

M-4(1)="车主余额表"

M-4(2)="\"

M-4(3)="规费收据"

M-4(4)="\"

M-4(5)="\" \* 明细帐 \* "

M-4(6)="部分打印"

M-4(7)="全部打印"

M-4(8)="\"

M-4(9)="总帐"

M-4(10)="\"

M-4(11)="资金平衡表"

M-4(12)="\"

M-4(13)="车主基况表"

dime M-5(13)

M-5(1)="数据备份"

M-5(2)="\"

M-5(3)="数据恢复"

M-5(4)="\"

M-5(5)="\" \* 数据录入 \* "

M-5(6)="车主基本情况"

M-5(7)="单车规费标准"

M-5(8)="\"

M-5(9)="\" \* 数据修改 \* "

M-5(10)="车主基本情况"

M-5(11)="单车规费标准"

M-5(12)="\"

M-5(13)="系统初始化"

Menu bar Top,5

menu 1,M-1,22,11

menu 2,M-2,12,12

menu 3,M-3,14,14

menu 4,m-4,13,13

menu 5,m-5,13,13

read menu bar to ROW,COL save

\* CHOICE 中存放. PRG 或过程名称

store "MK"+ltrim(str(ROW))+ltrim(str(COL)) to

CHOICE

do &. CHOICE

endd

clea all

return

## 双端口RAM实现任意波形发生器

袁波

(安徽大学人工智能所 合肥 230039)

程建溪

(东关水泥厂 安徽含山县 238151)

**摘要** 本文讨论了双端口RAM的结构、特点,利用双端口RAM同时读写同一RAM芯片的特点,实现了任意波形发生器中的应用。

**关键词** 双端口RAM 存储单元 数据传送 任意波形

### 1 概述

在科学实验中,为了调试设备,模拟特殊实验条件和工业现场环境,采用模拟实际信号方法进行研究,提出了任意波形发生器的设想。

多CPU系统信息交换,高速数据的传输、存储,对存储器提出的要求越来越高,采用双端口RAM进行数据交换,实现数据隔离,达到了实时模拟信号,它可以用较慢的器件实现快器件的功能,消除了单端口存储器信号的停顿,从而产生连续波形。

双端口存储器分为两类:一种是共用一组地址线,而有两组数据线及数据I/O系统的存储器;另一种是两组地址线,两组数据线及数据I/O系统的真正的双端口存储器。本文采用的是属于后者的IDT7130。

### 2 双端口存储器工作原理

双端口存储器IDT7130是Integrated Device公司的产品,采用CMOS工艺,输出类型为3-state,额定电压5.0V,容量1k×8bits,取数时间和写周期都是120ns。

IDT7130有两组地址线,两组数据线及控制线,允许两个端口对同一单元同时进行处理。

从图 1, 我们知道除存储阵列外, 该器件具有完全独立的两组逻辑电路, 通过各自的地址线, 可以分别选中不同的存储单元, 两组逻辑控制信号保证了两边操作的独立性。

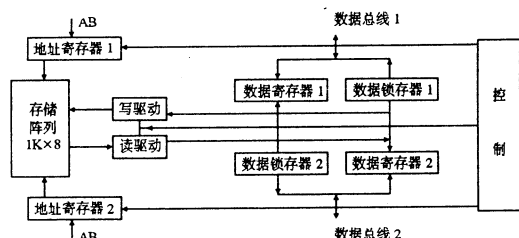


图 1 双口RAM的结构

同时为了防止双端口对同一单元有至少一个“写”操作, 造成“读”和“写”的错误, 设置了“忙”信号。当一端口是“写”操作时, 另一端口“忙”, 禁止另一端口“读/写”操作; 该端口“写”完毕, 另一端口才释放“忙”信号; 如果是两端口“读”同一存储单元, 没有冲突; 当两端口对不同存储单元操作时, 不产生“忙”信号。

### 3 双端口 RAM 实现数据传送

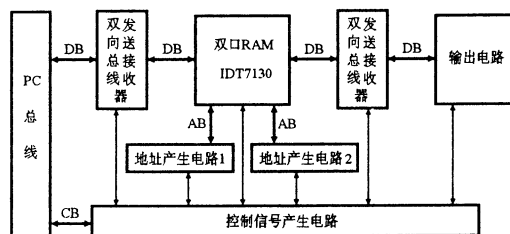


图 2 任意波形发生器工作原理图

微机处理一个模拟波形后, 产生了数据文件, 数据文件产生可以是模拟波形量化处理得到, 也可以是手

工产生。数据文件中数据经过 PC 总线传送至双端口 RAM 中存储, 当接收到“产生波形”指令时, 控制信号产生电路发出“控制”指令, 硬件电路实现双口 RAM 右输出数据, 经过输出电路 D/A 及平滑、滤波等, 恢复 PC 机的模拟信号, 同时 PC 机接口源源不断地向双口 RAM 写入数据, PC 机还一直在监测“忙”信号, 由于采用 TTL 电路, 速度与双口 RAM 匹配, 不会出现“左写右读”同一单元的情况, 但微机仍一直监测“忙”信号 (中断方式)。该卡在 486 微机上调试, 如果对一个正弦波周期采样 10 点 (8 位分辨率), 那么正弦波最大频率可达 500KHz, 实际产生 300KHz 频率以下正弦波信号是可靠的。

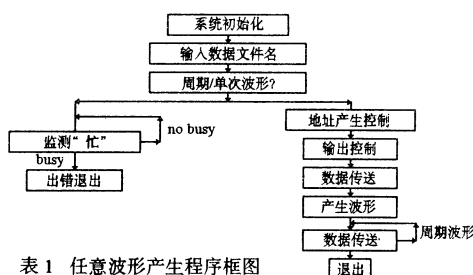


表 1 任意波形产生程序框图

### 参考文献

- 1 IDT Memory Components Data Book. Integrated Device Technology Inc. 1989
- 2 Memory Components Handbook. Intel Inc. 1989
- 3 《存储器集成电路》. 国防工业出版社, 1995. 9
- 4 周明德, 白晓笛. 《高档微型计算机》(上、下) 清华大学出版社, 1989

## 实用单片机集散控制系统通信机制的研究

刘志勤 陈 波

(西南工学院 四川绵阳 621002)

**摘 要** 本文介绍了一种用于集散控制系统的通信机制的设计过程, 从计算机网络的概念出发, 阐述了一种实用多机通信系统的结构、电路和软件设计的过程。

**关键词** 通信 上位机 下位机 网络

### 1 引言

通信在单片机集散控制系统中具有非常重要的作用, 其前级控制系统必须定时地向上位机发送计量数据及工作状态, 同时也必须接收上位机发布的多种操作命令, 并对运行参数进行修实系统的协调与控制。

本通信机制是针对一种前级控制机多达 8 台的集散控制系统而设计的多机通信系统。该系统已应用于我院的多个科研项目中。

### 2 通信网络结构

集散控制系统在结构上采用了分级设计的思想, 用多台处理机构成各站, 并可进行分散设置。因此, 各



站之间信息的传送显得十分重要。

为实现各处理机(即各站)之间的数据的合理传送,必须采取一定的网络形式,使各处理机形成一个计算机网络。根据系统要求,我们采用了星型网络结构,如图1所示。

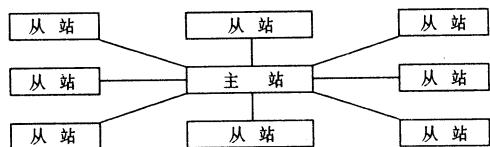


图1 星型网络结构

其中作为从站的集散控制系统下位机(前级控制机)共计有8台,用于生产过程中的具体控制对象,各自为一相对独立的闭环小系统;作为主站的上位机与各下位机之间连接成星型网络结构,在网络中各站有主、从之分,主站的处理机要求功能强,可靠性高。主从站之间的链路是专用的,因此线路的传输率高,而且便于程序集中研制和资源共享。

### 3 网络通信协议

本系统的多机通信采用的是主从式网络协议,它利用了8098串行口工作方式2的下述特点,即如果被接收的9位数据第9位不是“1”,则不会引起中断。多机通信的原理简述如下:

多机通信开始前主站和从站皆工作于方式2,当主站要向某一从站发送一数据块时,先发送一帧用于识别该从站的地址,然后再发送数据块。地址帧与数据帧的区别就在于前者的第9位数据为“1”,而后的第9位数据为零。对于工作在方式2的从站,数据帧是不可能使其中断的,但地址帧却可以使所有从站中断,以读取主站发送的地址。当某一从站得知接受到的地址与自己的地址相符时,则该从站便使其串行口转入工作方式3,以便接收主站随之发送的数据;此时未被寻址的从站仍工作于方式2。对主机发送的数据帧不予理睬,继续处理自己的事务。另外,互相通信的两个系统所用的波特率必须相等。假设使用10位帧传送,由于波特率误差引起偏移,在最后一位传送时保证位传送时间 $6/8$ 有效,则在一个方向上的偏差允许为1.25%,两个系统的偏差不应大于2.5%。

### 4 多机通信的实现

根据以上要求和8098的特点,我们设计了如下方式的多机结构:

#### (1)通信约定

8098单片机的串行口,虽具有全双工的异步通信功能,但没有应答信号,它只在发送完一帧数据后,产生一次中断申请发第二帧数据,并不管接收方是否取走数据,第二帧数据即可发送。为了使数据不产生丢失现象,我们设计了主从式串行通信方式。这种通信方式的特点是硬件结构及其简单,由于在实际工程中,每次

传送的数据量大,因此极为实用。

①下位机系统始终处于从态,上位机为主态。任何一次通信总是由上位机发起,它的第一个字节为从站号并引起所有前级微机中断,在中断服务程序中,当判定此数据为本机地址时,则将串行口改变为工作方式3,进行正常数据传送。

②本台从站在确认本次通信是针对自身时,将分机号发往主站,以通知本从站已经准备接收数据了。如果主站长时间接不到指定的从站的应答,再发一次地址帧,如仍收不到应答,说明这台前级机未启动或出了故障,它将转去进行与其余前级机的通信,以免造成死机现象。

③当上位机发出的命令是要求从站送回一批数据时,则本台下位机将组织好要发的一批数据,并启动发送程序,为了不造成发、接双方的时限偏差,这批数据必须依次发完才退出本次中断,这就要求主站一直等到收完这批数据后,才能进行其它工作。

④下位机接收数据则完全处于中断工作方式,每接收到一帧数据产生一次中断,直到接收完所有数据时,才又将串行口改为工作方式2,准备好进行另一次数据传送。

#### (2)传送的数据结构约定

##### ①主站发送下位机的数据结构形式

第一字节: 分机号 第9位必定为“1”,引起所有下位机中断。

第二字节: 传送字节数 给出本次通信传送字节数。

第三字节及以后是上位机发来的数据(每一帧为一字节),直到发完为止。

##### ②下位机发回上位机的结构形式

第一字节: 分机号 通知上位机,本机已接收到上位机发来的要求与本机通信的启动命令。

第二字节: 传送字节数 通知上位机本次传送的字节数。

第三字节及以后为传送的数据块,包括从站运行的各种参数。

#### (3)多机通信电路

通信硬件电路如图2所示。我们采用了MC1488和MC1489作电平转换,4N25光电隔离器隔离上位机与下位机之间的干扰,74LS05是OC门,其作用是在实现多机通信时,共用一条通信线实现线“与”方式连接。

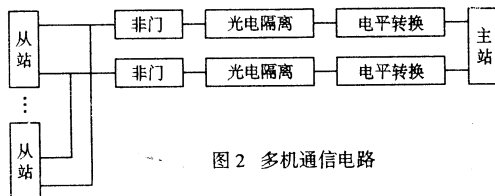


图2 多机通信电路

### 5 下位机串行口中断服务程序

下位机串行口中断服务程序框图如图 3 所示。

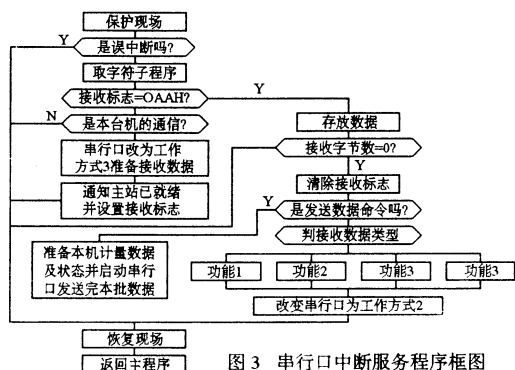


图 3 串行口中断服务程序框图

在通信程序中,当执行到串行口的中断服务程序,首先判断是否由接收中断引起,否则是一次误中断,这时将自动退出中断服务程序,当通信机号(地址)不相同也会自动退出,只有在接收标志等于 0AAH 时,

才能进行数据传送。通过试验证实这种技术对通信可靠性是很有效的。

在数据传送结束时,程序将对本次接收到的命令进行分析,根据不同的命令转入不同的程序进行功能处理。

### 参考文献

- 1 方建淳编著. MCS-96 系列 8098 单片机原理与应用技术 天津科学出版社,1990 年 12 月
- 2 孙涵芳等编. MCS-96 系列 16 位单片微型计算机 北京航空航天大学出版社,1989 年 2 月
- 3 [美]A. S. 坦尼伯姆原著. 曾华焱等译. 《计算机网络》. 成都科技大学出版社,1989 年 12 月
- 4 《Distributed control》. Automation. June 1992
- 5 《The Distribution System of the year 2000》. IEEE power Engineering Review

## NDP C-386 在 PC 环境下 DSP TMS320C30 系统中的应用

丁国盛

(中国船舶总公司第 723 研究所 扬州 225001)

**摘要** PC 环境下 DSP TMS320C30 系统是一个被广泛采用的高速实时数字信号处理系统。本文介绍一种高性能 PC 软件开发环境 NDP C-386 语言编译器,对于数据量处理较大的 DSP 系统,该语言在其中有较好应用。同时,本文指出在调试 NDP C-386 语言程序时应注意的问题。

**关键词** 数字信号处理(DSP) NDP C-386 语言 视频模拟器

### 1 概述

TMS320C30 是 Texas 仪器公司生产的第三代数字信号处理器(DSP),它具有每秒能执行 3300 万次浮点运算的能力(33MFLOPS)<sup>[1]</sup>,其主要特征有:

- 大部分指令都是 60 毫微秒单周期执行时间
- 两组 1K×32 位单周期在片 RAM(片内)
- 64×32 位指令缓冲存储器
- 32 位指令和数据字,24 位地址能寻址 16M×32 位

- 32/40 位浮点、整数和逻辑运算(ALU)

- 在片直接存储器存取控制器(DMA)

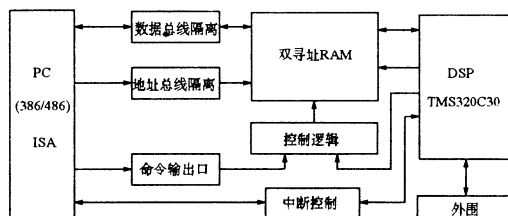
- 支持高级语言编程(C 语言)

- 两个 32 位定时器,两个串行端口

PC 环境下 DSP TMS320C30 系统将数字信号处理器的高速运算能力和微型机的完善操作系统结合起来,形成一个既能满足高速处理需要,又可以普及的数字信号处理系统,在图象处理、语音分析以及智能仪器

等领域得到了广泛的应用。

其原理框图一般如下所示:



TMS320C30 以插件板(DSPB)形式插在 PC386/486 机 ISA 插槽中,PC 机作为主系统,负责对 DSPB 进行数据加载,控制启停,监视执行,以及进行数据通讯等,DSPB 作为从系统,具有自己的处理器(TMS320C30)、存储器等资源,负责完成主系统加载的快速数字信号处理任务。

为了有利于主/从系统之间的控制和管理,在主机和 TMS320C30 间还常设计相互中断电路。

DSPB 上的存储器是由 TMS320C30 和 PC 机共

同管理,通过滑动窗(页面管理),将存储空间分为多个区,每区为  $16K \times 32$  位,占用 PC 机一段地址空间,由 PC 机通过向某一控制端口发出不同命令字,可分别访问 TMS320C30 管内的所有空间。

对于数据量处理较大的 DSP 系统,PC 机与 TMS320C30 制双寻址 RAM 将设计超过 DOS 可管理的 640K 常规内存,系统要求 PC 能产生相应的数据量,并可以对双寻址 RAM 中的大量数据进行修改,因而 PC 如何最有效的对扩展内存操作在系统中就显得尤为重要。

## 2 模式介绍

我们知道,MS-DOS 工作在 80386/80486(以下简称 80386)CPU 实模式下,实模式是微处理器初始化时所采用的模式,在实模式下 80386 只是相当于一个“快速”的 8086,只能寻址 1M 空间,80386 段寄存器内有指定段的实际地址,通过地址尺寸前缀和操作数前缀,可使用 32 位地址寻址,物理地址最多有 21 个有效位,即可在 00000H~10FFEFH 范围寻址。

保护模式是充分发挥 80386 性能的模式。在保护模式下,段寄存器内保存着 16 位的地址,即选择器(Selector),由它指示描述符表中的某一项,实际的段基地址被插在描述符表中,描述符寄存器保存着 32 位的段基地址。段选择器由三个子域组成,它们是段描述符索引号、TI 位和 RPL,每一个段描述符索引号都对应一个段描述符,段选择器的 TI 位用于标识所指向的段描述符是在广义描述符表(GDT),还是在局部描述符表(LDT)中,段选择器的 PRL 域与段的优先级有关。段描述符包含有关段的全部信息,段描述符分为内存段描述符和系统段描述符两类,内存段描述符由 8 个字节组成:段的基地址,段的限长和段的属性,系统段描述符用于描述 80386 操作系统表,任务等信息。在保护模式下,微处理器可支持 4GB 的存储空间<sup>[2]</sup>。

尽管 MS-DOS 提供了扩展内存管理程序,但其只是通过页面映射技术,通过建立 DOS 缓冲块,将扩展内存分页管理,并不能真正直接对扩展内存操作。可以通过实模式进入保护模式,然而,使用汇编语言,在 MS-DOS 下由实模式进入保护模式后再编写所要求的软件(产生数据、计算修改数据、装载 TMS320C30 等)将是非常繁杂的,并需要很好的编程技巧。

## 3 编程语言

C 语言由于其既具备高级语言的特点,同时又可直接对硬件直接操作,被广泛应用。一般的 C 语言编译器如 Turbo C, Borland C++ 等由于本身扩展内存。扩充内存的限制,即使在巨编译模式(Huge mode)下也只能产生 1MB 数据段 NDP C-386 是美国 Microway 公司专门为 386、486 微机开发的 C 语言<sup>[3]</sup>,它充分发挥了 386、486 微机的卓越性能,使编程寻址为全 32 位,NDP 系指数值协处理器,该语言有以下特

点:

(1)全 32 位寻址,突破 640K 限制,可分配 1M 以上数组。

(2)编译优化处理,全局数据流分析,最大限度利用了 386、486 和 NDP 的 32 位寄存器。

(3)丰富的内部函数和例程。

(4)编译生成的保护模式应用程序可以以调用实模式下的过程 and 应用程序,如 ROM-BIOS, MS-DOS 本身,以及 16 位的应用程序。

## 4 编程要点

我们在研制高密度电磁环境视频模拟器过程中,充分利用了 NDP C-386 编译器的以上功能使程序编制简洁明了,目标程序快捷高效<sup>[4]</sup>。

高密度电磁环境视频模拟器,用于模拟侦察接收机实时输出的数字化视频信号,用于对主处理机的调试和检修。硬件电路由 PC 环境下 TMS320C30 DSP 构成主从紧密耦合系统。PC 解算出每部雷达的脉冲绝对到达时间,并按其到达的先后进行排序,通过 PC 插槽 AT 总线将雷达脉冲参数 PDW 及相对到达时间传送给双寻址 RAM, TMS320C30 取相对时间间隔设定定时器,定时时间到达后,将脉冲数据经缓冲区传送到后续处理机,在 C30 发送数据的同时,PC 机重新计算变化的脉冲数据,利用 C30 定时器等待下一次发送的时间间隔,对双寻址 RAM 中的部分数据进行修改,实现 PC 机及 TMS320C30 的并行处理。

我们设计了  $2M \times 16\text{bits}$  双寻址 RAM, PC 产生接近 4M 的雷达脉冲参数,并能实时进行参数修改。如不采用 NDP C-386 其工作量将是巨大的。大量(大于 1M)脉冲绝对到达时间的排序,采用 NDP C-386 后,仅仅通过一个子程序调用即可实现。(见附:脉冲绝对到达时间快速排序子程序)。

调试 NDP C-386 语言程序时,除必须注意一些一般错误,如:顺序性错误、函数重名、恶性语法错误、边界错误,调用参数错误等等,还应特别注意:

(1)有关字长的问题

NDP C-386 为 32 位编译器,其基本数据类型字节数和数值范围与 16 位编译器有所不同,不加以注意,会引起不易检查的错误。

(2)全局变量

尽管全局变量的使用存在不少缺点,如:全局变量在程序的执行中都占用存储单元,模块间“内聚性”减弱,与其它模块“耦合性”加大,但合理正确的使用,可增加函数间数据关系的渠道,减少内存空间以及传递数据时的时间消耗。

(3)指针问题

指针是 C 语言的一个重要特色。指针问题归纳起来有两类:指针运算符理解错误和偶然使用了不合法的指针。

总之,必须保证指针有确定的地址指向,尤其在动态开辟内存时,所开辟内存空间应足够存放所有数据,否则出现野指针将造成系统中断死机现象。

附:脉冲绝对到达时间快速排序子程序

```
/* QUICK SORT SUB_PROGRAM */
/* iTem0:脉冲绝对到达时间指针 * TOA;
   item1:TOA 所对应 PDW 地址指针 * ADDR. */
sort(item0,item1,left,right)
float * item0;
int * item1;
int left,right;
{ int i,j;
float x,y;
i=left;
j=right;
x=*(item0+(left+right)/2);
do
{ while(* (item0+i)<x&&i<right)i++;
while(x<*(item0+j)&&j>left)j--;
if(i<=j)
{ y=*(item0+i);
```

```
* (item0+i)=*(item0+j);
* (item0+j)=y;
y=*(item1+i);
* (item1+i)=*(item1+j);
* (item1+j)=y;
i++;
j--;
}
}while(i<=j);if(left<j) sort(item0,item1,left,j);
if(i<right) sort(item0,item1,i,right);
}
```

### 参考文献

- 1 Third-Generation TMS320 User's Guide, Texas Instrumerts, 1988
- 2 吕晓庆. 80386/486 系统编程实践, 浙江大学出版社, 1993, P28—P34
- 3 谢俊. NDPC—386 用户手册. 北京希望电脑公司, 1991
- 4 丁国盛. 高密集电磁环境视频模拟器研究, 南京理工大学硕士学位论文, 1996

## 提高 8098 片内 A/D 转换器的方法

徐创文

(兰州工业高等专科学校 兰州 730050)

**摘 要** 本文主要阐述了 8098 单片机 HSO 输出的 PWM 脉冲调制信号滤波后作基准电压和输入信号差分放大后送 8098 A/D 进行转换, 从而获得高分辨率的方法。

**关键词** A/D 转换 分辨率 基准电压

A/D 转换器的分辨率的高低直接影响到测量系统精度的高低, 目前高分辨率的 A/D 转换器尽管能满足测量要求, 但价格昂贵, 8098 单片机片内 10 位 A/D 转换器基本上能满足大多数应用场合, 但研究以开发 8098 自身资源为基础, 提高其内部 A/D 转换器的分辨率, 从而拓宽它的应用范围还是具有一定的实际意义。

8098 内部含有一个 10 位的 A/D 转换器, 若取参考电压为 5V, 则其分辨率为  $\frac{5}{1024} \approx 5\text{mV}$ , 若进一步采取一些措施, 将其模拟信号划分为若干个段, 以段的首值作比较参考电压, 然后将输入模拟信号差分放大相应的倍数, 再送 8098 A/D 转换器进行转换, 最后将每段的结果相加, 便可得到相应分辨率高的转换值。本文以 10 位 A/D 提高到 12 位为例进行阐述。

### 1 段的划分

10 位扩展到 12 位需将模拟输入电压  $V_i$  划分成 4

段。

|   |                                        |                                 |
|---|----------------------------------------|---------------------------------|
| 若 | $0\text{V} \leq V_i < 1.25\text{V}$    | 取 $V_{\text{REF}} = 0\text{V}$  |
|   | $1.25\text{V} \leq V_i < 2.5\text{V}$  | $V_{\text{REF}} = 1.25\text{V}$ |
|   | $2.5\text{V} \leq V_i < 3.75\text{V}$  | $V_{\text{REF}} = 2.5\text{V}$  |
|   | $3.75\text{V} \leq V_i \leq 5\text{V}$ | $V_{\text{REF}} = 3.75\text{V}$ |

上述划分的依据是: 10 位 A/D 转换结果为 0~0124 对应模拟输入电压为 0~1.25V, 其分辨率为

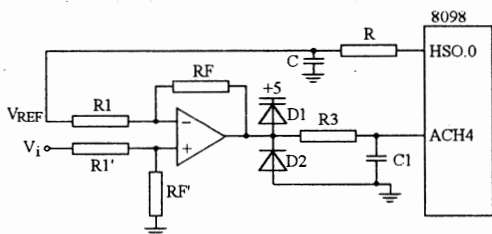
$$\frac{1.25}{1024} = 1.25\text{mV}, 12 \text{ 为 A/D 转换器的分辨率为 } \frac{5}{2^{12}} =$$

1.25mV。由此看出, 将输入的模拟信号 (0~5V) 划分 4 段, 对每段进行差分放大 4 倍, 再转换, 最后结果相加与 12 位 A/D 转换结果是一致的。

### 2 电路的组成

运算放大器采用自稳零斩波运算放大器 ICL7650, 该放大器具有极低输入失调电压和输入偏置电流, 具有较高的共模抑制比, 外接元件少, 使用方便。RI、RI'、RF、RF' 一律选 0.2% 的精密电阻。

为了保证运放的输入端处于平衡工作状态, 应使两个输入端对地的电阻相等, 同时为了降低共模增益,



应使  $R1=R1'$ ,  $RF=RF'$ , 可推导出  $V0$  与  $V_i$  之间的关系式为:  $V0 = \frac{RF}{R1} (V_i - V_{REF})$

若  $V_i - V_{REF}$  在  $0 \sim 1.25V$  之间变化,  $V0$  在  $0 \sim 5V$  之间变化, 应取  $\frac{RF}{R1} = 4$ 。

本电路中 ICL7650、 $R1$ 、 $R1'$ 、 $RF$ 、 $RF'$  组成差分放大电路, 对不同的模拟信号  $V_i$  与  $V_{REF}$  做差分放大后送 8098 ACH4 进行转换。

### 3 基准电压 $V_{REF}$ 的产生

$V_{REF}$  由 8098 的高速输出通道 HSO 产生, 8098 的 HSO 提供周期和占空比可改变, 分辨率为  $\frac{1}{65536}$  的输出信号, 经滤波后可以变为模拟信号, 由于输出的是 TTL 电平的调制脉冲, 因此为了获得较高精度的  $V_{REF}$  基准电压, 在滤波之前应通过缓冲器使 PWM 信号的摆幅变为  $0 \sim 5V$ , 图中缓冲器选 CD4049 反相型缓冲器,  $R$  和  $C$  的取值决定于输出信号的频率, 通过调节, 可获得最佳滤波。

### 4 软件设计

(1) 对于不同范围的模拟输入电压需选不同的基准电压, 因此应首先对模拟输入电压的范围进行判断。为了使硬件尽可能简单, 采用软件的办法来判断模拟输入电压的大小。为此首取  $V_{REF}=0$  对应的 CAM 数置 CAM 中, 使其  $V_{REF}=0$ , 若转换结果为 03FFH, 说

明  $V_i > 1.25V$ , 应换第二档, 置  $V_{REF}=1.25V$ , 若转换结果为 03FFH, 说明  $V_i > 2.5V$ , 依此类推, 往上递增, 若转换结果不超过 03FFH, 模拟输入  $V_i$  的转换结果计算关系式为:

$$AD-RESULT = 03FFH \cdot i + (0000H \sim 03FFH) \\ (i=0, 1, 2, 3 \text{ } i \text{ 代表 } V_i \text{ 所在段的段号})$$

(2) 在 HSO 引脚上产生一个脉冲信号需要设置三个事件, 第一个和第二个事件用以形成脉冲的上升沿(正跳变)和下降沿(负跳变), 第三个事件作为脉冲的周期, 若要产生连续的脉冲, 就必须不断写入这三个事件的命令字节和时间值, 程序中前两个事件触发后仅在输出端输出相应的高电平或低电平, 而第三个事件除了在输出端产生高电平外, 而且还应重写上述三个事件的命令字节和时间值, 以保证输出连续的脉冲, 因此前两个事件的触发并不开放中断, 而应在第三个事件触发后才开放 HSO 执行中断。输出 12 位 D/A 取相邻两次正跳变触发的时间为 0FFFH 即可得到 12 位 D/A 输出, 其值大小通过改变负跳变触发时刻以改变占空比来实现。为确保检测的准确性应保证 D/A 输出值稳定时启动 A/D 转换。在第四次中断服务程序中启动 A/D 转换, 判定  $V_i$  的区间, 若改变档次, 可改变负跳变触发时间值即可。软件实现流程图如图 1、图 2 所示。

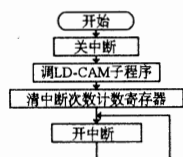


图1 主程序流程图

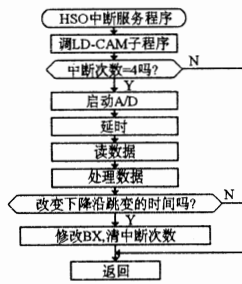


图2 中断服务程序流程图

## 网络电缆测试仪添利器

**本刊讯** 美国福禄克 (FLUKE) 公司北京办事处, 日前在北京发布了用于快速以太网安装和维护的电缆分析仪 DSP-2000, 为火爆的中国网络市场提供了便捷的电缆测量工具。

福禄克推出的电缆分析仪除具有高精度、高速度的特点外, 还有自动故障隔离功能, 可对电缆链路连接进行故障诊断, 迅速解决因电缆链路造成以太网的任何故障。近端串扰定位、显示网络利用率和碰撞信息, 是其所特有的领先功能。20 秒完成电缆双向全部测试工作, 测试速度比同类提高 2—8 倍。可记录 2000 个测试数据, 并可通过随机 Windows 环境软件 DSP-LINK 调入微机。提供的电缆管理软件, 可对测试结果分析、统计和管理, 生成测试报告。(云昌英)

## 世界上最高速的超级并行计算机——“CP-pacs”

据报导, 日本筑波大学近日推出了世界上运行速度最快的“CP-pacs”计算机, 其运行的实效速度达到每秒 3682 亿次(理论运算速度为每秒 3700 亿次)。

“CP-pacs”高速超级并行计算机, 由 2000 台高性能处理器并联组成, 主要用于模拟和分析宇宙诞生等物理现象, 以便揭开宇宙之谜, 验证素粒子理论等。由于“CP-PACS”性能高、速度快, 所以能够比其他超级计算机更高的精确度模拟银河系的形成。这种速度、性能好、精度高的“CP-PACS”受到美国、德国等世界计算机专家的好评, 也受到有关计算机用户的青睐。

摘自《计算机信息》

# 多媒体作品中声音卡的检测技术

夏 天 王化文 麻中华

(武汉大学计算机科学系 430072)

**摘 要** 该文阐述了在多媒体作品中自动检测计算机的声音卡的技术并给出了在 Authorware 中具体的实现方法。

**关键词** 多媒体 声音卡 Authorware

## 1 问题的提出

在一个多媒体作品中,声音占有很重要的地位。为了实现友好的用户界面和烘托各种气氛来提示用户,在各种多媒体系统中均采用了许多声音来获得这样的效果。Authorware 是一个常用的多媒体著作工具,有媒体的合成、编辑、控制和表演的功能。在利用 Authorware 制作多媒体系统的过程中,我们发现,如果一个多媒体的应用程序有许多用户时,有些用户的机器中可能没有声音卡。如果一台机器中没有声音卡,当多媒体的应用程序要播放音乐时,它就会给出一个错误的信息。如果有一种方法可以检测声音卡的存在,就可以避免这种错误的发生。本文给出了一种通过检测声音卡驱动程序来检测声音卡的技术并给出了在 Authorware 中具体的实现方法,此方法也适用于利用其他多媒体著作工具制作的多媒体系统。

## 2 检测技术与 Authorware 中的实现方法

如果一台机器中装有声音卡,那么在 system. ini 中有一项为“wave = \* \* \* \* \*”,不然,在 system. ini 中则没有一项为“wave = \* \* \* \* \*”。所以要检测声音卡是否存在,只需检查在 system. ini 中是否有一项为“wave = \* \* \* \* \*”。

下面给出 Authorware 中具体的实现方法:

为了检测声音卡驱动程序的存在,我们要把 system. ini 装入一个字符串变量,然后在字符串变量中搜索“wave =”这个串。如果这个串不存在,那么就没有声音卡驱动程序,Windows 就不能播放 .wav 文件。

打开一个新的文件,拖放一个计算(Calculation)图标到流程控制线中,把它的标题改为“Read System. ini”,打开计算图标,键入

```
systeminil:=ReadExtFile("C:\\windows\\system. ini")
```

```
systemini:=LowerCase(systeminil)
```

第一条语句是从 Windows 中读出 System. ini,并把它存储到自定义的变量 systeminil 中,第二条语句是将 systeminil 中的大写字母转变为小写字母。这里假设 windows 被装在 c:\windows 目录中。

然后拖放一个判断(Decision)图标到流程控制线中,在判断图标的右边拖放另一个计算图标,并把它标题改为“search for wave =”。打开计算图标,并且键入:

```
linenum:=linenum+1  
line:=GetLine(systemini,linenum)  
lineloc:=Find("wave=",line)
```

计算图标中的第一条语句是一个基本的计数器,每执行一次,linenum 就增加 1,第二条语句取出 systemini 中的第 linenum 行并把它存到 line 中,第三条语句是在 line 中查找“wave =”字段,如果在 line 中没有“wave =”,那么 lineloc 的值为 0,否则为“wave =”在 line 中第一次出现的位置。

关闭计算图标,打开判断图标。在判断图标的 Branching 域中选择 Sequential 项,在 Repeat 域中选择 Until True 项,并且键入

```
lineloc>0||linenum>LineCount(systemini)
```

在这一行中,判断图标一直循环下去,直到两个事件之一发生。

在流程控制线中,再拖放另一个计算图标,打开它,并键入

```
sound:=if(lineloc>LineCount(systemini),0,1)
```

这条语句表明,如果没有找到“wave =”字段,那么 linenum 就比 systemini 中的总行数大,就设定变量 sound 为 0,否则就设定 sound 为 1。

最后拖放一个声音(Sound)图标到流程控制线中,装入一个声音,在 Play Sound 域的 Time 项中键入 sound。

当执行这个文件时,如果在 system. ini 中有“wave =”字段,那么它就播放一次,不然就播放 0 次,而不会出现错误信息。

如果想播放 3 次的话,只需在 Play Sound 域的 Time 项中键入 Sound \* 3 即可。

### 要想学电脑 请订《电脑报》

《电脑报》以其可读性强,读者面广,栏目新颖,信息量大,订阅价低,读者至上,服务周到为特色,令广大读者爱不释手。欢迎订阅,欢迎赐稿。

本报邮发代号 77-19 统一刊号:CN51-0107

全年订价 42 元 地址:重庆人民路 236 号

电话:(0811)3876725、3876706 邮码:630015

# 微机硬盘不能启动故障原因与维修方法

王立伟

(中国第一汽车集团公司 长春 130011)

**摘要** 本文介绍了计算机硬盘常见故障的原因以及维护中采取的措施和排除方法。

**关键词** 硬盘 主引导扇区

在微机使用过程中常常遇到硬盘不能启动的故障,这些故障除了是由硬盘本身物理损坏和多功能卡出现的故障引起的原因外,大部分是由于硬盘启动过程中的某一环节出现故障产生的。所以我们应了解一些硬盘启动的原理,用安全、直接的方法排除硬盘故障。

硬盘自举过程简介:

微机加电,主板 ROM 中固化的 BIOS 监控程序工作,对系统硬件进行自检和初始化,读入硬盘分区的主引导记录,再由主引导记录调入引导分区的 BOO 程序,装入和执行 IO.SYS 和 MSDOS.SYS 并按 CONFIG.SYS 文件配置系统,再装入 COMMAND.COM 程序,并执行 AUTOEXEC.BAT 文件,执行完后后出现 DOS 提示符。当微机不能启动时就可按照这个顺序依次查找,确定故障的位置进行修复或用正确的程序重新覆盖。硬盘自举故障具体如下:

## 1 CMOS 配置表中硬盘参数设置错误

在微机主板上有一个由电池供电的小容量 CMOS 存储器,它的里面存储了一些系统配置参数。当系统启动时,固化在 ROM 中的程序自动对 CMOS 进行检查以确定系统硬件的配置情况。配置表中包含有硬盘类型一项,指定所用硬盘的容量、磁头数、扇区数、等参数。若这些参数被改动,硬盘出现故障。

**故障现象:**硬盘自举,能启动而部分硬盘不能读写,或当用软盘启动选硬盘时出现“invalid drive specification”而选硬盘无效。

**排除方法一:**开机或热启动后按 DEL 键(有些机型是按 CTRL+ALT+ESC 键或其它键,一般屏幕会有提示)进入 CMOS 配置表并找到硬盘设置项,可根据硬盘外壳标有磁盘参数,重新设置硬盘后,故障就可排除。

**方法二:**若硬盘外壳没有标注硬盘参数,则可把硬盘从机器上卸下来,找一台 CMOS 设置程序中有 Auto Detect Hard Dis 这一项的机器,通过该程序的自动检测得到硬盘参数。在有故障机器的 CMOS 设置中填上各项即可。

**方法三:**若硬盘外壳上没有标注,可执行下面的程

序,得到硬盘参数。

```
A)DEBUG
-A
XXXX:100 MOV AH,8
MOV DL,80
INT 13
INT 3
```

—G(此时屏幕上所显示的内容中,CH 的内容就是硬盘的最大柱面的低八位,CL 的高八位的内容是硬盘的最大柱面的高 2 位,CL 的低 6 位的内容是每个磁道的区数,DH 的内容是最大磁头数。)

—Q ;退出 DEBUG

有了正确的硬盘参数,再进入 CMOS 设置程序,进行正确的硬盘设置,则硬盘可恢复自举。

## 2 系统引导文件被破坏

IBMBIO.COM、IBMDOS.COM 和 COMMAND.COM,它们包含了 DOS 的基本输入输出命令、DOS 内核、内部命令等,若它们被破坏系统也不能启动。

**故障一:**当 IBMBIO.COM 和 IBMDOS.COM 被破坏时,故障现象是:在屏幕上显示:Non-system disk or disk error

**排除方法:**用带有 SYS.COM 程序的与硬盘 DOS 相同的系统软磁盘启动,在提示符下键入:A)SYS C:

命令结束后,硬盘中被破坏的两个隐含文件 IBMBIO.COM、IBMDOS.COM 就被系统盘中相同文件所代替。

**故障二:**如 COMMAND.COM 有问题,则屏幕上出现“Bad or missing Command”,此时可将正确的 COMMAND.COM 重新拷入,硬盘即可自举。

**故障三:**CONFIG.SYS、AUTOEXEC.BAT 均是文本文件,若这两个文件中所指定执行的程序有问题也可导致系统不启动的故障。

**排除方法:**可将这两个文本文件改名,只靠 DOS 系统启动,若 DOS 系统能启动,则进一部检查 CONFIG.SYS、AUTOEXEC.BAT 两文件中涉及到的程序。

## 3 硬盘主引导扇区被破坏

·主引导区内容简介:

主引导记录位于 0 面 0 柱 1 扇区,占 512 个字节,由三部分组成:



(1)主引导记录,占 446 个字节,地址偏移量为 00H~1BDH。它决定当前的活动分区。

(2)分区表,占 64 个字节,地址偏移量为 1BEH~1FEH,它分为 4 个 16 字节的登记项,每个登记项描述一个分区信息。

(3)硬盘有效标志,占 2 个字节,地址偏移量为 1FFH~1FFH,其值为 55AA。

由上可知,主引导区存放硬盘的重要信息,如果由于病毒或其它原因使硬盘主引导扇区被破坏,则硬盘不能启动。在没有主引导扇区内容的软盘备份的情况下,恢复硬盘自举是比较困难的,若用 FDISK 程序重新分区,并用 FORMAT 格式化新区,问题虽可解决,但硬盘上的信息将全部丢失。因此本文给出一简单汇编程序,它可在机器正常的时候将硬盘的主引导区保存在软盘上,当硬盘主引导区万一被破坏,可从软盘上对硬盘进行恢复,实现硬盘的自举。

方法如下:首先在软盘上保存硬盘主引导区内容

```
A:)>DEBUG
-A300
XXXX: 300 MOV DX,0080
      MOV CX,0001
      MOV BX,0100
```

INT 13

INT 3

```
-G=300
-NA:ROOT
-R CX
:200
-W 0100
-Q
```

将硬盘主引导区的内容保存在软盘 BOOT 文件中。

若硬盘主引导区被破坏,可将软盘上的 ROOT 程序的内容写到硬盘主引导区。

```
A:)>DEBUG BOOT
XXXX:300 MOV DX,0080
      MOV CX,0001
      MOV BX,0100
      MOV AX,0201
      INT 13
      INT 3
```

-G=300

-Q

通过以上介绍的几种排除计算机硬盘故障的方法,基本可达到硬盘日常维护的目的。

## SUN 工作站 IDPROM 内容失效后的启动方法

杨绍洲

(第一军医大学生物医学工程系 广州 510515)

作者在工作中碰到两例 SUN SPARC Station 2 工作站因内置电池失效导致自检不成功,无法正常登录的情况,现将碰到这种情况时的解决方法介绍给读者。

**故障现象:**在电池完全失效前一个月,工作站出现了时钟系统故障,时钟系统只有在开机状态时才能计时,关机后时钟系统停止计时,下次启动机器显示的时间是上次关机的时间。这阶段除了时间系统不能正常工作,其它功能均正常。这种现象持续约一个月,关机两天后再开机,出现故障:

开机约一分钟后屏幕才出现光栅,并显示:

Ethernet address ff:ff:ff:ff,host ID:ffffffffff.

The IDPROM contents are invalid

然后测试内存空间速度非常之慢,测试完内存之后显示:

Did not receive expected loopback packet.

Can't open boot device

Type b(boot),c(continue),or n(new command mode)

提示信息与正常时不一样,机器正常时自检后出

现“Login:”提示符,让用户输入登录名。而这时即使键入 boot 命令,仍出现上述的信息,不能引导系统。

**解决方法:**SUN SPARC Station 2 的 UNIX 操作系统提供了 setenv 设置环境命令。这时可以通过 setenv 命令配合适当的参数设置工作站的运行环境,引导系统。具体方法是:首先在“)”提示符下键入“n”,这时将出现“OK”提示符,然后在“OK”提示后键入:

```
setenv diag-switch? false
```

屏幕将显示 diag-switch? false 和 OK 提示符。这时再键入 boot 命令,就可以引导系统,自检完成后出现“login:”提示符,机器可继续使用。

**经验总结:**第一,我们平时应注意对机器保养,机器不能长期放置不用。若机器较长时间不用应该定期开机一段时间。这两台工作站出现故障前都放置了较长时间不用,故障的出现与这有关。第二,当机器出现故障预兆时,应及时检查维修,以免使故障范围扩大。第三,记录机器的一些重要参数。例如,工作站的机器号(hostid)是很重要的参数,但是由于各种原因,我们都没有把两台工作站的机器号记录下来,现在 IDPROM 失效了,不能得到正确的机器号,影响机器的应用。