

计算机应用研究

COMPUTATION RESEARCH OF COMPUTERS

1997 · 1



广西计算中心 通过考核验收

ISSN 1001-3695



001369007



广西计算中心海蓝集

地址：广西南宁市星湖路32号 邮编：5300

电话：(0771) 5851945 5860210 58516

传真：5852684



中国计算机学会会刊

ISSN 1001-3695

计算机应用研究

APPLICATION RESEARCH OF COMPUTERS

1997 · 1

广西计算中心
广西软件新技术实验室

GBH多媒体软件系列:

GBH赠送版

GBH正式版

地址: 广西南宁市星湖路32号 邮编: 530022 电话: (0771) 5862543

中国计算机学会会刊
中国科技论文统计源期刊



计算机应用研究 第14卷 第1期 (总第75期)

目次

综述评论

高度集成的多媒体系统开创多媒体时代的新纪元.....	查明华(1)
ISDN 技术及其应用.....	周虎 方正(3)
PC 计算机在物理化学实验中的应用以及有关问题的探讨.....	刘生昆 史振民 张俊生(6)
FDDI 网络技术概述.....	牟光灿(8)

研究探讨

提高应用软件可用性探析.....	张胜存(11)
二维线段的矩形窗口逐边裁剪算法研究.....	刘斌 王勇 黄树槐(15)
一种基于 POP3 的 INTERNET 电子邮件软件的设计与实现.....	李建成 喻成(17)
客户/服务器环境下数据库转换的实现.....	刘友华 闵建洪(19)
新型燃油系统的计算机仿真.....	马力中 陆展华 周华(21)
从网络管理到服务质量(QoS)管理.....	刘樟华 施萍(23)
计算机辅助地学断面管理及绘图系统的设计与实现.....	郑国政 袁仕章(26)
一种多维信息的数据压缩技术.....	周纪林(28)
Sybase/powerbuilder 环境中的用户管理.....	吴相林 张红涛 陈铁英(31)
超细颗粒布朗运动模糊图象的增强处理技术.....	陈卫 何振江 杨冠玲 罗昱 郭永彩 谢莉莉 吴开华(34)
OLTP 中间件——Tong EASY/TongLINK.....	余坤 辛涛(36)
用 GAL 器件设计地址译码器的新方法.....	郭显久(39)
通用型智能化实时数据显示与处理系统.....	程芳真 蒋滋康(41)
MASM6.x 中的 .EXE 文件压缩技术分析.....	贾志先(44)
时域波形智能识别的混合推理机制.....	刘艳斌 罗才双(46)
PRO * C 的研究与应用实现.....	谢鹏(49)
计算机数值控制中软件分频方法的研究.....	苗占毅 邓文新(51)
基于神经网络的储层敏感性评价专家系统.....	徐英卓(52)
Oracle CDE2 中对 Windows DLL 的调用.....	郭伟 石冰心(54)
Oracle 协同开发环境——CDE2.....	杨勇 周建国 曹奔星(57)
线性方程的两个算法.....	宁正元 王秀丽(59)

应用开发

Motif 程序中长任务的处理.....	易鉴良 谢立(61)
Windows 环境下图象处理的内存操作和显示技术.....	唐贵川 王建成 董金祥 何志均(65)
Internet 与异种网络互连中 DNS 名字的转换.....	宋志辉 朱华(66)
串行 EEPROM 的读写器的设计.....	孙洪军 孙秀云 王连军(69)
应用于机车车辆滚动振动试验台的监控系统.....	吴学杰(71)
利用 DCS 打印口实现数据通讯.....	卢苇 那思佳 卢军 李志蜀(74)
Turbo C 文本方式中文下拉式菜单的设计.....	胡诗国 宋威(76)
分布式控制系统并行通信的可靠性设计与差错检测纠正.....	李伟(79)
基于交流采样的六千伏电网微机自动监测系统.....	王红岩 祁璧(81)
容积调速系统的模糊控制研究.....	左健民 王积伟(84)
在华光排版系统上实现双页并排输出.....	马龙军(87)
子目录加密的策略与实现方法.....	朱家义 王玉芬(88)
用 C 语言实现对逻辑硬盘的隐藏.....	秦相林(91)
ATM 网络多媒体通信的具体实现.....	浦刚 周曼丽 魏蛟龙(94)
存储 IC 卡的读写方法.....	裴荣武 唐宁九 孙界平(96)
真彩色图象的模拟显示.....	张二虎(99)
计算机信息管理系统(MIS)在企业目标管理中的作用.....	傅晓强 蒋雅辉 童长祚(102)
在 Internet 中航行——Netscape Navigator.....	吴保平(104)

维护维修

一种程序运行失常自恢复控制的方法及其实现.....	罗杰(106)
硬盘驱动器引导型软故障解析.....	王雄舟(108)
入网多媒体 PC 机软硬件故障及其维护实例.....	余先庆(110)
对 AR3200 打印机使用手册的补充说明.....	王新华(111)

动态简讯

启事、广告.....	(7,10,25,38,68,73)
------------	--------------------

Soul Journal of Chinese Science & Technology Articles

APPLICATION RESEARCH OF COMPUTERS

VOL. 14 NO. 1 (Total 75)

CONTENTS

SURVEY

Height Integrated Multimedia System Ushered in a New Epoch in Multimedia Era	Cha Minghua(1)
Technology of ISDN and Its Application	Zhou Hu(3)
The Application of PC Computers in Experiments of Physical Chemistry and Approaching Questions	Liu Shengkun et al. (6)
Briefly Relating Technology of FDDI Network	Mu Guangcan(8)

RESEARCH

Probing and Analysing to Improve Availability of Application Software	Zhang Shengcun(11)
Algorithm Research of Edge by Edge Cut out of Rectangle Windows of Two—dimensional Line Segment	Liu Bin et al. (15)
Design and Realization Based on Software of INTERNET Electronic Mail of POP3	Li Jiancheng et al. (17)
Realization of Changing Database under Environment of Client/Server	Liu Youhua et al. (19)
Research of Numerical Simulation of New Type Fuel System	Ma Lizhong et al. (21)
From Network Management to Quality of Service (QOS) Management	Liu Dihua et al. (23)
The Design and Realization of Computer Aided Geoscience Transect Management and Drawing System	Zheng Guozheng Yuan Shizhang(26)
A Technology of Compressing Data for Multi—dimensional Message	Zhou Jilin(28)
User Management in Sybase/powerbuilder Environment	Wu Xianglin et al. (31)
The Enhancing Method for Indistinctive Image of Submicron Particles with Brownian Movement ..	Chen Wei et al. (34)
Middlewares for OLTP—Tong EASY/Tong LINK	She Kun Xin Tao(36)
A New Method of Designing Address Decoder by GAL Device	GuoXianjiu(39)
General and Intelligent Real—time Data Display and Handling System	Cheng Fangzhen Jiang Zikang(41)
Analysing Technology of Compressing .EXE File on MASM 6. x	Jia Zhixian(44)
A Mixing Reasoning Mechanism for Intelligence Recognizing of Waveform in Time Domain	Liu Yanbin Luo Caishuan(46)
Research and Implementation of the PRO * C Program	Xie Peng(49)
A Research on the Software Frequency Division of Computer Numerical Control	Miao Congyi et al. (51)
Expert System of Neural—Network—based for Formation Sensitivity Evaluation	XuYingzhuo(52)
Calling Windows DLL in Oracle CDE2	Guo Wei Shi Bing xing(54)
Oracle Cooperative Development Environment—CDE2	Yang Yong et al. (57)
Two Algorithms for Solving Linear Equations	Ning Zhengyuan Wang Xiu li(59)

APPLICATION & DEVELOPMENT

Processing Long Tasks in Motif Programs	Yi Jianliang Xie Li(61)
Operating Internal Storage for Handling Image under Windows and Technology of Display ..	Tang Guichuan et al. (65)
Conversion of DNS Name in Interconnection for Internet and Asynchronous Net	Song Zhihui et al. (66)
The Design of Reading and Writing Device for Serial EEPROM	Sun Hongjun et al. (69)
The Monitored Control System Used in the Roller Test Rig	Wu Xuejie(71)
Realizing Data Communication by DCS Printing Port	Lu Wei et al. (74)
Design of Text Way Chinese Language Downstream Pull Menu with Turbo C	Hu Shiguo et al. (76)
The Dependability Design and Errors Check and Correction of the Parallel Communication System Among PC Microcomput- er and Multi—Single—Chip—Microcomputers	Li Wei(79)
The Computer—based Monitoring System of 6KV Power Distribution Station with Alternative Sampling	Wang Hongyan Qi Kun(81)
Studying Fuzzy Control of Capacity Governing System	Zuo Jianmin et al. (84)
Realizing Putting Out Two Pages Abreast on Composing System of Huaguang	Ma Longjun(87)
Tactics and Methods of Subdirectory Protection	Zhu Jiayi Wang Yufen(88)
Realizing in Hidding Logical Disk with C Language	Qin Xiang lin(91)
The Realization of Multimedia Communication in ATM Net Work	Pu Gang et al. (94)
Method of Read Write on IC Storage Card	Gong Rongwu et al. (96)
Simulated Display of Real Color Image	Zhang Erhu(99)
Function of MIS for Computer in Business Target Management	Fu Xiaoqiang et al. (102)
Navigation in Internet—Netscape Navigator	Wu Baoping(104)

MAINTENANCE

A Control Method and Its Realization about Program Automatic Recovering from Insane Run	Luo Jie(106)
Analysing Bootstrap Soft Breakdowns of Driver on Harddisk	Wang Weizhou(108)
Breakdowns of Soft and Hardware on Multimedia PC Computers Entered Netware and Examples of Information	Yu Xianqing(110)
Additional Remarks for Operation Hardbook of AR3200 Printer	Wang Xinhua(111)

INFORMATION

Notice & Advertising	(7,10,25,38,68,73)
----------------------------	--------------------

高度集成的多媒体系统开创多媒体时代的新纪元

查明华

(清华大学力学系 北京 100084)

摘 要 本文以 Chromatic Research 公司的 Mpact Media Engine 芯片处理器为例,全面分析了实现高度集成的多媒体产品的必要性,介绍了高度集成化多媒体产品的最新进展,并指出了传统多媒体产品的缺陷。

关键词 Mpact Media Engine 高度集成 多媒体系统 芯片

0 引言

多媒体技术出现于 80 年代,而其形成商品化却是 90 年代初。它适应了人们运用计算机获取,传递和处理信息的需要,为整个社会进入信息化提供了必要的技术基础,成为计算机行业中发展最快的一个领域。

目前多媒体产品种类繁多,即使对于计算机行业的专业人员,在安装各种多媒体产品时特别是在同一系统安装多种多媒体产品时十分不方便,而多媒体的最终发展方向是应该面向广大的家用电脑市场,实现多媒体电脑的普及就必须使产品实用方便,可靠性高。很难想象需要计算机专业人员才能使用的产品能够大量走入家庭。因此实现多媒体电脑的普及必须解决产品的可靠性和易用性这两个关键问题。当前个人计算机主机的生产为多媒体产品提供了十分有益的借鉴。最新的个人电脑是将原来的各种计算机板卡集成于一块主板之上,由少数实力强大的硬件制造厂商来生产主板,而大多数个人电脑厂商利用集成了硬盘,CDROM 驱动器,软盘驱动,打印机,串行设备和键盘等各种设备接口的主板来制造整机就十分方便了。这种先进的设计方案大大提高了个人计算机的可靠性和可维护性。国外许多半导体厂商和硬件制造厂商为实现多媒体的高度集成进行了不懈的努力,做了大量的准备工作。随着计算机技术,信号处理技术,半导体技术特别是超大规模集成电路技术取得突破性的进展,实现高度集成化的多媒体产品的时机已经成熟。本文通过 Chromatic Research 公司的 Mpact Media Engine 芯片处理器,分析了实现高度集成的多媒体产品的必要性,介绍了高度集成化多媒体产品的最新进展。

1 传统多媒体产品的局限性

目前多媒体产品种类繁多,为实现某一类多媒体功能必须设计专用的多媒体板卡,如多媒体声卡, MPEG 回放卡,调制解调器,视频采集卡等五花八门的产品。目前大多数计算机软、硬件厂商把各种多媒体功能大致分为下列七大部分:

· MPEG 解码,包括对 MPEG I 和 MPEG II 两种标准的支持。

· MPEG 编码,包括采集,编辑和压缩。

· 具有显示加速功能的高分辨率真彩色图像显示

· 具有 2,3 维图形快速重建功能

· 具有波表合成功能的音频处理

· 支持语音和数据同时传输的高速调制解调器

· 支持 ISDN 及 POTS 方式视频电话及视频会议

下面是目前电脑市场已有的具有上述七种功能的典型产品:

多媒体产品一览表:

产品名称	用途	总线结构
Greative. AWE32	32 位带波表合成声卡	AT 总线
Matrox MGA	3D 加速真彩色显示卡	PCI 总线
OptiBase Promotion	MPEG I, II 回放卡	AT 总线
Real Magic Producer	MPEG I 采集压缩卡	PCI 总线
Hayes Accura288	28.8K 调制解调器	AT 总线
Intel Proshare200	ISDN 视频会议系统	AT 总线

通过上表,我们可以想象在同一台主机上,要装上所有这些多媒体卡可能会十分勉强,每一种多媒体卡都有自己特定的硬件参数设置,如 IORQ, DMA, IO 地址等,而每一种多媒体卡都有可能与其它的多媒体卡在硬件参数设置上发生冲突。同时现在个人计算机的操作系统也不统一,以 Intel 芯片为基础的操作系统就有 DOS、Window3.1、Window3.2、Window95、Window NT 及 OS2 Warp、ScoUnix 等。除此之外需要遵守的 PC 标准协议,还有许多,如 MCI, GDI, DCI20, Direct-Draw3D DDI, Direct 3D, Wave, Direct Sound, TSPI, ADPCM, Truespeech, VESA Vidco BIOS 等等。要求不同厂商的各种多媒体产品在各种操作系统下能够协调工作几乎是不可能的。

上述多媒体产品的局限性是由历史原因造成的。在多媒体发展的初期,由于受技术条件的限制,每一种

多媒体产品的功能比较单一,产品也没有一个统一的标准。随着计算机技术的发展,越来越多的厂商推出了各种各样的多媒体产品,一些在市场上占主导地位的产品逐步成为事实上的标准,如 Creative 的 Sound Blaster 成为音频产品的标准。同时一些软硬件厂商和国际组织针对用户对多媒体的要求和多媒体的发展趋势逐步制定了许多多媒体设备的产品标准。针对这些标准,原有多媒体产品的局限性便逐步暴露出来,研制高度集成的多媒体产品已成为当务之急。同时多媒体设备标准的建立为生产高度集成的多媒体产品提供了依据。一些多媒体制造厂商经过多年的努力,终于在最近推出了一批成熟的高度集成化的多媒体产品。

2 高度集成化的多媒体产品实现方式

目前实现高度集成化的多媒体产品主要有两种方式,它们分别是采用主机通用芯片的方案和采用多媒体数字信号处理器的方案。这两种方式在实现多媒体的高度集成化时各有千秋,但目前普遍认为利用多媒体数字信号处理芯片实现高度集成化的多媒体产品是一种比较好的方案。下面通过分别介绍这两种方案,我们可以清楚地看到这两种方案的优缺点,认识当今多媒体发展的主流。

2.1 利用通用芯片实现高度集成化的多媒体产品

利用通用芯片实现高度集成化的多媒体产品就是利用计算机主机上功能强大的芯片来实现多媒体的功能。这种方案的特点是可以利用针对现有芯片如 Intel 公司 X86 系列芯片基础上多年积累的巨大软件资源和成熟产品开发环境,广大厂商可以针对不同的多媒体应用集中精力开发相应的软件,减轻厂商硬件设计工作的负担。随着主机芯片处理能力的不断增强而使多媒体系统的性能得到提高。利用通用芯片开发高度集成化的多媒体产品目前主要集中在个人电脑市场上占主流的 Intel 公司的 Pentium, Pentium Pro 芯片和 IBM, Motorola 和 Apple 公司的 Power601, Power604 芯片基础上。Apple 公司早在八十年代末就在 Macintosh 个人电脑的操作系统上加入了多媒体音频系统,当前随着 Power604 芯片的出现,Apple 公司在它最新的 System7.5 操作系统上强化了对多媒体的支持,并在其最新一代的 PowerMac8500 系列个人电脑上设计了完整的音视频多媒体系统。而 Intel 公司在其推出 Pentium 芯片以后,于 1994 年推出了 Native Signal Processing(简称为 NSP)的方案,这个方案出现的背景就是 Intel 公司认为基于 Pentium 及 Pentium Pro 处理器芯片的能力已经大大增强,利用个人电脑中 Pentium 芯片的功能完全能够满足各种多媒体应用的需要,不再需要专用的多媒体芯片来实现特定的多媒体功能。因此在 NSP 方案中 Intel 公司针对多媒体的应用提出了许多新的标准。如用于三维真实造型的 3DR,用于多媒体视频的 Indeo,用于单频的 Native

Audio, 显视控制接口 DCI 及电话通讯界面 TAPI 等等,目前已有许多计算机软硬件厂商支持 Intel 的 NSP 方案。然而通过大约一年半的实际应用,人们发现采用 NSP 方案在完成多媒体音频和语音的应用时, Pentium 芯片的处理能力是绰绰有余的,但是对于 MPEG—2 标准数据的实时解码,复杂三维图形的重建和 MPEG—1 数据的实时编码这些计算量很大的多媒体应用来说,主机芯片就显得有些力不从心了。从图 1,我们可以很清楚地看到,对于 MPEG—1 数据的实时编码来说仅仅利用主机芯片的运算能力是远远不够的。即使采用多 CPU 的 Pentium Pro 系统或 Pentium Pro 下一代芯片也难以完成多媒体应用中巨大的计算量,而且采用多 CPU 的 Pentium Pro 方案必然造成产品的价格很高,给产品在销售中带来很大的困难,造成产品在对价格非常敏感的家用电脑(Home PC)市场上很难得到普及。因此在相当长的一段时间,实现高度集成化的多媒体内必须依赖多媒体数字信号处理器,才可能使产品具有良好的性能价格比。

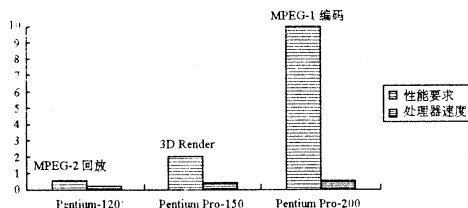


图1 多媒体应用对处理器的要求

2.2 利用多媒体数字信号处理芯片实现高度集成化的多媒体产品

利用多媒体数字信号处理芯片实现高度集成化的多媒体产品的优点很多,首先可以针对多媒体应用的特点来优化数字信号处理芯片的硬件结构,使其具有良好的性能。其次由于多媒体数字信号处理芯片可以通过软件编程来实现不同的多媒体应用,与传统的专用多媒体芯片相比,在实际应用中的灵活性大大增加。因此在 96 年中,多媒体专用的数字信号处理芯片就成为了一个热门,如 Micro Unity 的 MediaProcessor, Philips 公司的 Trimedia, Chromatic Research 公司的 Mpack Media engine 芯片和 Nvidia 公司的 NV1 芯片。这些产品不同于早期的数字信号处理芯片如 Texas Instrument 的 TMS320 系列芯片,其芯片的硬件结构主要是针对多媒体的应用来设计,并采用了许多超级计算机上所采用的最新技术如 VLIW(超长指令)SMID(单指令/多数据)和向量处理等等,多媒体数字信号处理器性能在某些方面已经可以与已往的超级计算机不相上下。同时由于这些数字信号处理芯片采用大规模工业化生产,并有巨大的市场需求量,这些数字信号处理芯片价格可以随着市场需求量的不断增长而很快降低,所以利用这些芯片构成高度集成的多媒体系统具有良好的性能价格比,有很大的发展潜力。下面通过介

绍 Chromatic Research 公司的 Mpack Media engine 芯片多媒体数字信号处理芯片,我们可以了解数字信号处理芯片实现高度集成化的多媒体产品的方式。

Chromatic Research 公司的产品由 Mpack Media engine 芯片和 Mediaware modules 两部分组成。Mpack Media engine 芯片是针对多媒体的数字信号处理芯片,Mediaware Modules 是与其相对应的微程序软件开发模块,它巧妙地采用了与 Pentium 及 Pentium Pro 相互补的结构,将大量的浮点运算由 Pentium 来完成,充分发挥 Pentium Pro 芯片浮点能力强的优点,简化 Mpack Media engine 芯片的硬件结构,同时在 Mpack Media engine 芯片中采用了 VLIW, DIMD 和向量处理等一系列超级计算机所用的技术,使其在整数处理能力上大大高于 Pentium Pro。芯片上有 5 个 ALU,包括一个用于运动估计,通过 792 位纵模式总线,与内部 4K 高速 SRAM 相连,SRAM 有 8 个端口,可以以 500Mbps 的速度与芯片外部的高速存储器完成数据交换。芯片在每个时钟周期可以同时完成 32 个整数运算,对于大多数指令,Mpack Media engine 芯片的速度可以达到每秒 20 亿次操作,而在完成运动估计运算(它是 MPEG 编码中的最常用计算)时,处理器的速度可以达到每秒 200 亿次操作。由于芯片采用了 PCI 标准的总线接口,因此,Mpack Media engine 芯片可以作为针对 Pentium Pro 芯片而设计的功能强大的多媒体协处理器,很容易地被集成到以 Pentium Pro 芯片为核心的计算机主板上,当然它也可以作为高度集成化的多媒体板卡的核心用于主机芯片为 Pentium 或 Pentium Pro 的计算机系统上。下面是 Mpack Media engine 芯片在多媒体应用中的结构示意图。

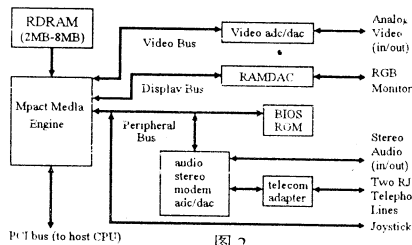


图 2

从上图我们可以看到,利用 Mpack Media engine 芯片。我们可以很容易实现本文开头所提到的多媒体应用所要求的全部功能。通过 Media Ware Module 软件开发包,我们可以利用 Mpack Media engine 芯片实现下面的多媒体应用:

- 具有色彩空间转换功能的视频回放处理
- MPEG—1 和 MPEG—2 标准的视频,音频或视音频解码
- MPEG—1 标准的视频,音频或视音频编码,可将视音频信号实时采集成符合 CCIR601 标准,以 YUV4:2:2 方式编码的 AVI—editable MPEG 文件

· JPEG 编码和解码

· Windows GUI 加速

· 最大可支持 8M RDRAM,达到 1280×1024 分辨率下,75Mhz 刷新速度的真彩色显示和真彩色 NTSC/PAL 视频输出。

· 支持阴影,光照,材质和抗褶皱功能处理的三维图形重建

· 符合音频工业标准(FM 合成方式和 MIDI 输入/输出处理)

· 支持波表音频合成,Dolby AC—2 音频解码和 3 维立体声效果

· 支持每秒 28.8K 传输率的调制解调器和全双工的语音电话,具有同时传输语音和数据的能力

· 支持 H. 320 和 H. 324 标准的视频电话和视频会议

在软件上,Media Module 支持目前所有个人电脑的工业标准和 Windows APIs。包括 MCI, GDI, DCI20, DirectDraw, 3D DDI, Direct 3D Wave, Direct Sound, TSPI, ADPCM, Truespeech, VESA Video BIOS。所以利用 Media Module 开发的多媒体产品可以与现有大量的个人电脑软件相兼容,因此利用 Mpack Media engine 芯片,我们可以很方便地实现高度集成的多媒体系统。

我们从图 2 中,还可以很清楚地看到采用 Mpack Media engine 芯片的集成多媒体系统具有非常良好的性能,其处理速度完全可以满足现有的多媒体应用需求,甚至对于计算量极大的 MPEG—2 的编码,利用 Mpack Media engine 芯片也可以做到接近实时,而目前 MPEG—2 的编码只能由很昂贵的专用系统来完成。在将来我们完全可以利用个人计算机上完成 DVD 的制作,使计算机的多媒体提高到一个新的水平。

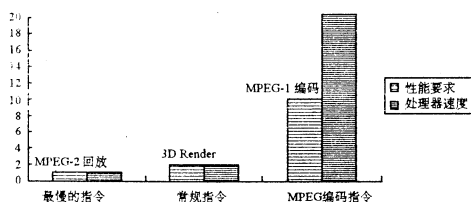


图 3 多媒体应用的要求与 Media Engine 处理器的性能比较

如图 3 所示,以 Mpack Media engine 芯片为核心的高度集成化的多媒体产品与传统多媒体产品相比,具有很大的价格优势。据预测以 Mpack Media engine 芯片为核心的多媒体产品大约为两万人民币左右,而与其功能相当的传统多媒体产品却需要十万元之巨。而且随着 Mpack Media engine 芯片为核心的多媒体产品需求量的提高,其价格还可以大大下降。我们可以利用它享受高清晰度的 DVD 影碟,优美的高保真音乐,通过 Internet 与远隔万里的朋友面对面地交谈,通过计算机虚拟现实技术在我们从未到过的宇宙空间中遨游。现在的电视,影碟,音响,传真,电话将完全被高性

ISDN 技术及其应用

周 虎 方正

(上海交通大学计算机系 200030)

摘 要 本文简要介绍综合业务数字网 ISDN 及其技术特点,并探讨其应用领域和发展前景。

关键词 ISDN 网络 视频会议

1 ISDN 简介

曾经一度受到冷遇的 ISDN 技术,再度受到人们的高度重视。如今 ISDN 在北美、欧洲和亚洲等广泛地区的发展已初具规模,发展 ISDN 已经是各国电信业的必然趋势。

综合业务数字网(ISDN)是一种交换式数字技术。它将常规的电话网升级为一个全数字化的端对端网络。ISDN 利用单独的交换式网络连接进行混合信息传输,包括语音、数据、视频图像、传真和图像等。

ISDN 分两类:基本速率界面(BRI)和主速率界面(PRI)。两者都包含多个 B(Bearer)信道和一个 D(Data)信道。BRI 使用电话公司现有的双线铜缆线路,并将线路上的两个 B 信道和一个 D 信道进行多路复用。

B 信道可用于传输声音、数据、视频图像和传真、数据传送形式有线路交换语音(CSV)、线路交换数据(CSD)和分组交换数据(PSD)。B 信道工作在 56kbps 或 64kbps 速率,视如何划定及电话网络是否可传送专

用信道(clear-channel)的数据呼叫而定。两个 B 信道的传送数据形式可分别设定,如设 B1 为 CSV, B2 为 CSD。D 信道用于发送信号、呼叫设置和呼叫者标志。ISDN 将 D 信道用于带外呼叫设置,使呼叫设置时间仅为 1 秒钟左右。D 信道工作在 16kbps 速率。在 BRI 线路上, D 信道还可用来以 9.6kbps 的速率传送分组交换式数据。利用两种信道, ISDN 将声音、数据、视频图像和传真等综合在一条独立的线路上发送到目的地。

在标准化方面,国际电信联盟(原名 CCITT)电信标准化组织(ITU-TSS)将 ISDN 定为一种国际远程图像的标准。它制定了两类主速率界面标准:23B+D 和 30B+D。前者采用 23 个 B 信道和一个 D 信道,北美的 PRI 服务采用这种标准(NI-1);后者采用 30 个 B 信道和 1 个 D 信道,这种线路多被欧洲和亚洲国家采用,北美的 NI-1 标准与之兼容,因此可作为 ISDN 的国际性应用标准。

此外,ITU-TSS 将 ISDN 业务划分为三大类:

(1)承载业务(Bearer Service)

这是单纯的信息传送业务,由网络提供,它包含了

收稿日期:1996 年 6 月 30 日

能的多媒体电脑所取代。国外的专家预计以 Mpack Media engine 芯片为核心的多媒体产品在 96 年中大量上市,在 97 年开始大量进入家庭。随着中国国际化的日益深入,我们完全有理由相信在不远的将来,高性能多媒体的电脑大量进入中国百姓的家庭将不是梦想。

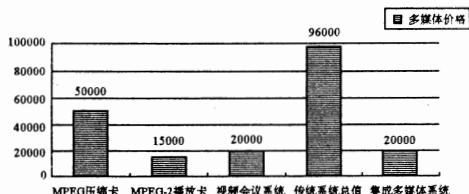


图 4 传统多媒体产品与集成多媒体产品价格比较

3 结束语

随着多媒体系统集成化程度的提高,多媒体系统的价格会不断下降。由于目前众多的多媒体产品生产厂家在硬件设计中做了许多重复性的工作,投入了大

量的研制资金。而单一功能的多媒体产品造成电脑市场上多媒体产品品种繁多,但每种产品的数量又受一定的限制,厂商为回收产品初期投入的巨额研制费用必然造成多媒体产品的定价很高,很难被广大的家用电脑用户所接受。在多媒体产品高度集成化后,大量的厂商只需在少数几种通用的硬件基础上就可以开发各种各样的多媒体系统,特别是当 Mpack Media Engine 芯片这类通用多媒体硬件与计算机主板集成在一起时,其制造成本可以大大下降,多媒体系统整机的价格就可以被广大家庭所接受,而巨大的家用电脑需求量又为多媒体的发展提供更加广阔的舞台,我们完全可以预测在未来一两年中高性能多媒体电脑会像彩电音响那样走入到寻常百姓的家庭。当前我们要抓住机遇,避免重复开发过时的多媒体硬件产品,而应当集中力量研制适合高集成度高性能多媒体要求的各种软件,开发具有中国国情的各种计算机媒体,为即将到来的多媒体普及时代作好充分的物质准备。

OSI 参考模型的 1—3 层。

(2) 用户终端业务 (Teleservice)

面向用户的通信或信息处理业务。由网络和终端设备共同提供,包含 OSI 参考模型的 1—7 层。

(3) 附加业务

在承载业务和用户终端业务基础上附加的业务

ISDN 的设备从技术上来说叫做 CPE (customer premises equipment),用于和 ISDN 设备相连,这种设备通常为终端适配器(TA)和网络终端 1 类(NT1),它具有电话、调制解调器、网桥或路由器的功能。

2 ISDN 的技术特点

ISDN 有许多重要的优点,其中最重要的就是通过一个单独的交换式网络连接进行混合通信,包括声音、数据、视频图像等等。

ISDN 的交换式特点使得 ISDN 比固定式点到点租用线路灵活得多。它使得网络上的任何人都可以从一个站点到另外任意站点实现按需连接,一个单独的接口可连接任意数量的站点。与调制解调器传输相比,ISDN 的交换式模拟和专线数字技术误码率低,还可以充分利用虚拟连接和按需使用带宽,用户使用多少带宽,就支付多少费用。

ISDN 的呼叫设置很快,不超过 1 至 2 秒钟。与之相比较,通过常规的拨接线路,调制解调器需要 40 秒钟实现一个连接。

3 ISDN 技术的应用

3.1 Internet 访问和远程局域网访问

这两种应用是 ISDN 应用最热门的方向。对于个人用户或小型办公机构,调制解调器一直是访问 Internet 的最常用方式。它无法适应大量图形化信息的传输要求。如果采用 ISDN,访问 Internet 的速度将提高到 2—5 倍,而且费用相差也不大。

ISDN 可以取代局域网间的租用线路将多个局域网互连,而且还能够使主机或网络端口分享多个远端设备的接入,这种特性比专线租用更灵活和经济。本地的局域网还可以与大型主机相连,远端局域网中的终端或工作站都可以通过 ISDN 成为本地局域网的延伸或扩展,共享主机的资源。在局域网的互连中 ISDN 可以通过电路交换连接,也可以提供分组交换连接,随着帧中继业务的发展。ISDN 也可以提供帧方式的连接。ISDN 可以使多个局域网构成一个虚拟网络,用户可以象访问本地局域网一样访问远程局域网。

3.2 视频会议和屏幕共享

视频会议是 ISDN 的理想应用领域。现有的视频会议系统分 room—based 和 computer—based 两种。

前者采用若干条 ISDN—BRI 线路并且逆向多路复用 B 信道以达到每秒 15 到 30 帧的视像速率。后者采用一条 ISDN,视频速率为 12 到 15 帧。如果采用 PRI 线路,则可以得到高达 1536kbps 的数据传输速率,而一般说来,384kbps 的速率就足以传送高质量的图像。

屏幕共享是使两个以上的用户可以共同观看和操作同一个文件。ISDN 建立一组用户的话音桥路和数据桥路,为多个已经同时提供会议电话和数据会议通信业务。这样一组用户可以同时观看、编辑同一个文件。

3.3 语音服务

利用 ISDN 可以建立会议电话系统。提供 ISDN 网络将多个话音终端连接起来,允许一个用户同时与多个用户进行通信。使用 ISDN 线路的电话可以象普通的电话一样,和世界任何一部 ISDN 或普通电话联络。

3.4 传真

利用 ISDN 的 G4 传真可以达到 $400 \times 400\text{dpi}$ 的分辨率。无论从质量还是速度上,这都比传统的 G3 传真优越得多。

此外,ISDN 还可应用于连锁店的经营管理、电力公司、煤气公司、自来水公司的自动读表系统等等广泛领域。

4 ISDN 的建设

由于数字型非话业务的发展,以 ISDN 的发展要求日益迫切。ISDN 的国际标准和实施技术已经基本成熟。但是 ISDN 的建设工程复杂,因此是一个相对漫长的过程。根据其它国家和地区的建设经验,ISDN 的建设可以分为以下四个步骤:

(1)把目前兼容各种业务的模拟电话网改造成数字交换和数字传输相接合的综合数字网,即 IDN

(2)把电话用户环路改造为数字用户环路并可通过一个接口通入各种通信业务的数字终端,实现具有统一编号的 64kbps 的窄带综合业务数字网。

(3)建设大于 64kbps 速率的宽带综合业务数字网

(4)建成能传送和处理各种通信业务的宽带 ISDN

参考文献

- 1 AIC. ISDN BASE TECHNOLOGY GUIDEBOOK, Seoul, Asian ISDN Council, Oct, 14, 1992
- 2 W. Stallings, ISDN, An Introduction, Macmillan, 1989
- 3 赵慧玲编著,综合业务数字网技术及其应用,人民邮电出版社
- 4 程时瑞编著,综合业务数字网,人民邮电出版社

PC 计算机在物理化学实验中的应用以及有关问题的探讨

刘生昆 史振民 张俊兰

(延安大学化学系 716000)

摘 要 本文论述了 PC 计算机在物理化学实验中的应用和 PC 计算机在化学学科中所产生的巨大影响,并对有关问题进行了分析与探讨。

关键词 PC 计算机 物理化学实验 应用

1 引言

在信息时代的今天,计算机作为人类征服自然,改造世界,不断开拓新领域,创造新奇迹的工具,越来越广泛应用于人类社会的各个领域。例如用于地质、邮电、气象、水利、电力、金融、卫生、国防、教育、科技、工农业生产等各个部门,肩负着数据处理、电子通讯、信息管理、经济预测、自动控制、机械设计、智能机器人等重任,并已产生巨大影响。计算机渗入到各行各业,各门学科,化学当然也不例外。计算机与化学结合,产生了一个新的领域——计算机化学。或更具体地说,是计算机在化学中的应用。这是一门边缘学科。边缘学科是促进科学发展的新的动力。回顾化学的发展史可以发现,每次化学的重大进展都是与引入新技术、新概念、新思想密切相关。人们最早研究的化学是无机化学,当推广到与生命物质有关时,产生了有机化学。引入量的概念之后,产生了分析化学。把物理方法引入化学后,又产生了一系列的化学分支。如引入热学、力学、光学、电学、磁学、量子学等概念便产生了化学热力学、化学动力学、光化学、电化学、磁化学、量子化学等等。

随着我国教育事业的繁荣发展,为适应物理化学实验教学的需要,计算机正在以相当的速度进入物理化学实验室。目前,我国各高校都先后装备了各种类型的计算机,特别是 PC 计算机的覆盖率正在逐年增长。PC 计算机以它内存容量大、功能全、速度快等优点已在物理化学实验教学中起到了重要作用,为实验的数据处理、实验模拟、实验控制、辅助教学、试题库管理、教学资料管理、学籍管理、药品试剂管理、教学设备管理等提供服务,并已取得明显的效果。

本文从计算机在物理化学实验中应用的现状入手,在分析计算机在物理化学实验中应用目前所取得成果的同时,结合笔者在近年来使用 PC 计算机过程中的一些体会,阐述了 PC 计算机在化学领域中所产生的巨大影响,其目的在于探讨化学中那些学科更适合 PC 计算机去占领,力求为 PC 计算机的进一步应用起到抛砖引玉的作用,不妥之处,敬请各位同行指正。

2 计算机在物理化学实验中的应用

2.1 实验数据处理。获得实验数据之后往往要进行实验数据处理,有些实验要进行复杂的数学运算,如曲线拟合,谱峰分辨,最小二乘法,最优化方法,回归方程,经验方程的选择等。

2.2 实验模拟。实验模拟是通过计算机模拟一个生产流程或一个实验体系,以达到掌握或了解某一方法的目的。

2.3 实验控制。实验室使用微机进行仪器控制,实验测试以及采集实验数据日趋普及,用微机控制仪器能提高仪器的精度、灵敏度、稳定性。微机采集实验数据准确性高,且可避免人为的误差。在物理化学实验仪器中,大到核磁、红外、紫外,小到移液管、滴定管都可以用微机来控制,计算机在物理化学实验仪器的自动化、智能化方面起着重要作用。

2.4 计算机辅助教学。用计算机辅导学生学习主要是通过人机对话的形式进行,教师把多年积累的教学经验和对学生答案中可能出现的错误做充分分析后,以程序的形式事先储存在计算机中,学生调出所需要的程序后,计算机通过荧光屏显示提问,学生通过键盘进行回答,计算机根据学生的回答作出反响。

2.5 试题库管理。考试是构成教学环节的一个重要组成部分,教师用考试这个手段来检查教学效果,学生通过考试来衡量自己掌握课堂内容的程度。无论是课外作业还是考试都需要习题,用计算机储存习题并加以分类处理,为教师选题提供了极大的方便,习题储存可按章节分类,也可按难易程度分类。教师根据需要可从试题库中选择合适的习题组成课外作业和试题。

2.6 学籍管理。记录学生入学时间、毕业年代、专业、选修课程、课程成绩、名次等事宜。

2.7 教学资料管理。记录保存与教学内容有关的文献、数据、图谱、试卷、习题、论文等历史记录,教师可根据需要,从计算机中调用有关信息。

2.8 其它管理。包括仪器、药品、设备等各项管理。

3 计算机在化学学科中所产生的重大影响

计算机与化学结合是化学教学的必然趋势,并且正在产生重大影响,主要表现在以下几个方面:

3.1 产生新的边缘学科。如计算机化学、化学计量学、计算分析化学、量子化学近似计算方法、有机化合物的拓扑表示方法等。

3.2 促进了理论化学的发展。虽然理论物理的发展较早,但是理论化学的发展较为缓慢。这是因为理论化学的计算工作量庞大,非人力所能胜任,只有在计算机发展的基础上才有可能。如量子化学、复杂反应体系的化学力学、多组分体系的化学平衡等都需要计算机才能解决。

3.3 测试仪器的改进。许多测试仪器配有计算机,用计算机自动控制测试仪器,采集测试数据是实验室使用计算机的重要方面,这不仅减轻实验人员劳动强度,避免主观读数误差,更重要的是大大加速了测试速度并提高了精度和灵敏度。

3.4 化学情报的检索。情报信息是现代社会的的重要组成部分,科学研究离开情报将会闭塞、落后、重复前人工作。用计算机检索情报文献,加速信息交流的速度之快是人工操作无法比拟的,这就为科学研究活动赢得了大量宝贵的时间,减少不必要的浪费。

3.5 化学数据库和专家系统的建立。目前已发展了各类化学数据库,通过数据库的检索可以方便地获得所需要的资料。数据库和智能模拟相结合,能部分地代替专业性很强的脑力劳动,形成一个专家系统或智能数据库。近代的先进测试仪器,往往都配有小型专家系统,测试结束后,能将测试数据和库存数据进行比较,并作出结论。

3.6 化学教学上的应用。辅导学生自学,在计算机上模拟化学实验,进行教学管理等。

3.7 实验数据处理的改进。使化学工作者摆脱凭经验作图或表格式的数据处理的落后方式,得以采用以概率统计方法为基础的严格的数据处理,以便从中抽取更多更重要的信息,引出新的结论。

3.8 加快实验室成果的工业化。过去实验室成果

的工业化,往往需要较长时间,但应用计算机之后,实验室成果可以通过计算机模拟放大直接对生产规模的设备进行设计,大大加速了实验室成果的工业化。

4 有关问题的探讨

从以上讨论我们可以看出,计算机应用到化学领域,为化学学科的发展,实验测试技术以及数据处理等开辟了一些新的领域。但毕竟这方面的研究探索工作起步较晚,有些方面还需要进一步的研究与探索,如:合成路线途径的选择;按需要性能设计分子结构;从动力学数据出发自动推断反应机理;从各种图谱数据综合判断结构及组成;各种物质性能的共性规律及数据的产生;工程化学模化系统的发展;计算机辅助科研、辅导教学、辅助设计;物质微观结构研究与宏观性能规律的关联等等。

5 结束语

总之,迎接计算机时代的到来,向传统工作方式挑战,不仅是社会发展的必然趋势,也是人类发展的必然结果。目前,把PC计算机引入物理化学实验教学中,为化学学科的发展,实验测试技术以及数据处理等开辟了一个崭新的领域,用计算机替代教师、学生的传统手工作业,不仅可能,而且可行,实践证明,用计算机方法研究化学问题,产生了一系列新的边缘学科,而这些边缘学科的开拓,将会不断地影响化学的发展,树立化学史上一个又一个的重要里程碑。

参考文献

- 1 程光钺、陈庚华、忻新泉等编著,计算机应用基础,高等教育出版社,1987年4月。
- 2 许志宏、王乐珊编著,无机热化学数据库,科学出版社,1987年9月。
- 3 刘生昆、张俊兰编著,物理化学实验数据处理计算机程序集,陕西人民教育出版社,1989年9月。
- 4 李克安、童沈阳编,分析化学中的数值计算方法,北京大学出版社,1990年2月。

方便实用的全屏幕编辑数据输入程序

(本刊讯)河南省计算中心的科技人员李金铭于最近采用Foxbase设计出一则全屏幕编辑数据输入程序,并举一反三地总结出在其他语言下编程全屏幕编辑程序的一般思路,操作简便,极为实用。需此程序清单者,请同李先生联系:

地址:河南省郑州市花园路27号

邮编:450003

单位:河南省计算中心

姓名:李金铭

FDDI 网络技术概述

牟光灿

(中航总公司成都飞机设计研究所 610041)

摘 要 FDDI 网络技术是目前较为热门的三种高速网络(FDDI、ATM、交换 Ethernet)技术中,最成熟、应用最广泛的一种。本文就 FDDI 网络技术简述如下。

关键词 FDDI 网络 FDDI 信息帧 TVX 计时器

1 FDDI 网络结构

1.1 网络协议

FDDI(光纤分布式数据接口)技术是完全按标准实现的技术。因此,从一开始就具备高性能、高可靠性和互操作能力。这为它的普及应用奠定了基础。

FDDI 协议是美国国家标准协会 ANSI X₃ T9.5 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。

在所有网络协议和接口标准中,FDDI 提供了站级网络管理能力及多级 SMT 服务能力。FDDI 的双环故障自动闭合能力提高了网络的可靠性。

FDDI 标准共分四个子层,即 MAC、PHY、PMD 和 STM。它们与 ISO 的 OSI 网络协议的对应关系如表所示:

OSI	FDDI	
数据链路层(DLL)	LLC	
	介质访问控制子层(MAC)	SMT 站管理
物理层(PL)	物理协议子层(PHY)	
	物理介质相关子层(PMD)	

MAC 子层规定了定时令牌协议和帧格式。

PHY 子层的主要功能是构成光纤环,定义了时钟频率和数据编码。

PMD 子层的内容主要是对结点与光纤介质接口中的交变部分的规定。

SMD 主要负责协调其它三层的活动。

1.2 FDDI 主要性能

FDDI 网络的主要性能是:

- a:数据传输率:100Mbps
- b:通信介质:光纤
- c:拓扑结构:反向旋转的双环
- d:链路控制方式:令牌访问方式
- e:站间最远距离:2KM(单模光纤可达 25KM 以

上)

f:数据包长最大值:可达 4500 字节

g:网络最大长度:200KM(超过局域网范围,进入校园网与城域网领域)

1.3 FDDI 信息帧

1.3.1: 帧结构

PA SD FC DA SA INFO FCS ED FS 其中:

PA:发送帧的前导码,通常由 8 个字节组成,是可变的。各节点利用弹性缓冲器,通过对前导码的增减,调节相互之间的时钟同步。PA 前导码一般不能少于 4 个字节。

SD:帧开始定界符,由 JK(1100010001)符号对组成,表示帧的开始。

EC:帧控制域,指示地址长度、类别、令牌类型、帧类型。1 个字节长。

DA:目的地址,6 个字节长。

SA:源地址域,6 个字节长。

INFO:信息域,范围 0~4500 字节。

FCS:帧校验序列域,含 CRC 值,4 个字节长。

ED:帧结束定界符(TT 表示令牌,T 表示帧)。

FS:帧状态域,含 E、A、C 状态指示器。

E——错误指示器,指示是否有 CRC 错误;

A——地址指示器,指示地址是否被目的站识别;

C——帧拷贝指示器,指示目的站是否复制信息帧。

1.3.2 帧类型

FDDI 的帧类型有以下几种:

- 令牌
- 数据帧
- MAC 管理帧
- SMT 站管理帧

a)FDDI 令牌

- 结构

PA SD FC ED

- 形式

* 有限制令牌(FC=11000000 同步方式)

* 无限制令牌(FC=10000000 同步方式)

b) 数据帧

- 同步 LLC 帧 (FC=1L010000)
 - 异步 LLC 帧 (FC=0L000PPP,
- 其中 PPP=优先级, 111 最高)

c) MAC 管理帧

- 信标帧 (FC=1L000010 同步方式)
- 请求帧 (FC=1L000011 同步方式)

信标帧与请求帧属于填充帧, 前者在有故障情况下的自动复位时填充双环, 确定故障位置; 后者则填充初始化时的环路, 确定首发令牌的站点。FDDI 环路的初始化发生在如下情况:

- * 站点入、出环路
- * 令牌丢失

d) SMT 站管理帧

SMT 站管理帧常用的类型是: (全是异步方式):

- 邻站信息帧 (FC=0L001111)
- 状态信息帧
- 回响帧
- 资源分配帧
- 请求否定帧
- 扩展服务帧

在上述各类帧中的 FC 值里, L 表示地址长度: L=1 48 位地址 L=0 16 位地址。

SMT 站管理帧完成 FDDI 站点管理协议的基本管理功能, 包括差错统计、配置信息及站点控制命令等。它的优先级可视情况而变。通常按低优先级处理, 对具有快速故障恢复要求的系统, 优先级会有所不同。

2 FDDI 的技术特点

2.1 关键技术

2.1.1 ETR 提高了环路利用率

FDDI 采用了早期令牌释放技术 (ETR)。只要某站完成了数据发送, 就将令牌释放。FDDI 的每一个站都分配了最大量的令牌持有时间。若在该时间内已完成数据发送, 站立刻释放令牌。所以, ETR 提高了环路的利用率。这样, 站就有了多余的带宽。若又收到一个早期令牌, 则又可使用多余的时间进行数据发送。所以, FDDI 动态分配额外带宽。若某站需要额外发送时间, 就可从动态池中取出。

为了控制环上各站较好地利用环路, FDDI 设置了三种计数器: TRT, TVX, THT。

a) 令牌轮转计时器 TRT

TRT 指示令牌按计划时间到达指定站。对 FDDI 网络而言, 令牌的实际到达时与计划到达时间是不同的, 且时间值随环路中某站的加入和退出而变化。若 TRT 计数超过 TRT 值 (令牌轮转时间) 的 2 倍时, MAC 将记录一个迟到的令牌。在严重错误的隔离过程中, SMT 将使用计数器上的累计值。

b) 令牌持有计数器 THT

环初始化期间, 处于一个请求发送令牌的竞争过程。环路上所有站点协商解决传输时间的分配, 均有一个 THT 控制本站发送数据的时间长度。不论何时, 一旦令牌提早到达 (这意味着 TRT 值变小), THT 值相应增大, 使该站的数据发送时间延长。

c) 有效传输计时器 TVX

TVX 计时器可用来探知环路是否有效。只要站收到有效帧或令牌, TVX 就清零, 然后重新计数。若 TVX 计时达到某预定值 (缺省值 4ms), 则表示这段时间内没有收到有效的帧或令牌。这时, 站将进行环恢复过程, 发出信标帧, 通知各站, 环有故障。整个过程均由 SMT 控制。

2.1.2 高效率的数据编/解码系统

FDDI 属于串行基带传输系统, 必须用代码流来组合数据与时钟信号。要恢复串行代码流的数据, 必须内含同步时钟信息。

FDDI 的 PHY 使用 4B/5B 代码结构完成数据的编/解码。这种代码方案具有 80% 的效率。而 Ethernet 规则采用的曼彻斯特编码技术, 仅有 50% 的带宽利用率。

4B/5B 的代码结构由二部分组成:

- a) 将收到的 MAC 层 4 位 MRZ (不归零) 符号变为 5 位 NRZ 符号;
- b) 将 5 位 NRZ 符号变为传输用的 NRZI (不归零反相) 符号。

数据的编/解码由“符号”传输来完成。根据 FDDI 网络协议, PHY 接收来自 OSI 第二层 MAC 子层的 4 位 MEX (十六进制) 符号, 将其转成 5 位符号, 然后又变为 NRZI 符号, 再通过 PMD 子层发送上网传输。由此可见, 从 MAC 到 PHY 之间的符号信息描述了一个特殊的 5 位代码序列, 称为“代码组”。这种编码方案的好处是降低了时钟要求, 实现了 100Mbps 高速传输速率, 而所需的时钟频率仅为 125MHz。若采用曼彻斯特码, 则需 200MHz 的时钟频率。

5 位码方案还可确保任何有效位的传输。传输中所遇到的连续零不超过 3 个, 接收站就不会丢失时钟。5 位符号的发送时间称为符号时间, 等于 $5 \times 8\text{ns}$ (代码元位时间)。

2.1.3 可满足站点的特殊传输请求

当某站请求特殊传输, 可使用限定令牌方式。即某站可发送一个延长对话的初始部分到目标站, 然后再发限定令牌。目标站接收到初始对话, 立刻进入限定令牌方式。当限定令牌到达目标站时, 目标站回答并发送一个限定令牌。这样, 两站即可交换数据帧和用于延长对话的限定令牌。若某站发送完最终对话帧, 立刻发送出一个非限定令牌, 则限定令牌方式结束。站管理实体有一个最大限定令牌方式时间, 并加以监视。若超过这个时间, 站管理将取消这次扩展对话。

限定令牌主要用来完成一种特殊的异步传输。FDDI 支持同步和异步两种传输服务:

a) 同步服务给各站一个被保证的带宽和响应时间,各站要在 2 倍的 TRT 内保证能看到令牌。这类服务适用于带宽和响应时间可预料,并通过站管理预先分配给各站。但是,这种允许传输的时间预算是有限的。为了确保各站间的顺利传输,站管理负责的带宽分配过程可设置最大传输时间,它是分配给同步传输总时间的百分值。

b): 异步服务用于带宽要求不太好预计或响应时间要求不到临界状态的应用。它是从剩余带宽池中瞬间分配的,故而传输不能保证,与环上所有的站有关。对于一般的异步传输。一旦传输开始,站允许传输的时间受 THT 限制。

2.1.4 完善的 SMT 站级管理体制

站管理提供的是站级的控制能力,管理 FDDI 各层之间的进程,维持相邻上下游站与本地站的协调工作。根据 SMT 的功能,它们由三部分组成,其任务及主要功能如下:

a) 帧服务

主要是建立管理帧,用以传递信息到其它站的站管理实体。SMT 将产生多种帧治环路传输,用来控制与诊断。

b) 连接管理(CMT)

在站管理功能中,管理物理层与环路的连接与断开的操作以及物理层实体与 MAC 实体之间的连接管理。它主要由三部分组成:实体协作管理(ECM)、物理连接管理(PCM)、配置管理(CFM)。

c) 环管理(RMT)

环管理接收来自 MAC 和 CFM 的状态信息,并报告 MAC 的状态。这些状态包括错误条件,比如被套牢的信标等及探测到的重复地址站。通过这些状态报告,SMT 将采取相应动作,以维持环的正常操作。

2.2 容错及可靠性

FDDI 的高可靠性主要来自于它的双环拓扑、传输介质及相应的控制设施。

与 Ethernet 网和一般的令牌环网不同,FDDI 具有故障容错机制,用于一旦在环路中断或站故障时恢复到一种可操作状态。

FDDI 的双环拓扑由主环和副环组成。主环是正常工作环,当出现故障时,副环用以构成通路。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。这种环绕(Warp)功能也用于站故障的恢复。若双环上的某站故障,两边的站将从故障站绕开。上述两种情况说明 FDDI 双环故障容错功能。但是,若某站恰好断电(这在网络中是常有的事),则环将该断电站视为“故障”,而采用环绕措施。当然,这不是所期望的操作。因为,双环上的某站断电属正常关机中断。

对于正常的关机中断,有二种解决办法:

一是采用光旁路开关,将关机站旁路出环路,以维持环的整体性。二是使用环的树形拓扑,用单连接站。每个站经双连接集中器与双环连接。由于站不与双环直接相连,关机成故障时,双连接集中器将探测出光信号的损失,并对相应端口加以旁路,所以不影响双环。但是,若出现干线电缆或 HUB(集线器)故障,则整个环将瘫痪。

除此之外,光纤传输介质具有良好的抗干扰性、高保密性和信号衰减小等优点,为网络带来了一定的安全性。

3 FDDI 的基本应用方式

从网络的应用来说,总体上可分为三层:广域网、主干网和各部门的局域网。

作为主干网领域,将来的发展趋势将以 FDDI 为主要代表。而 FDDI 网的主要应用方式为:后端网、前端网、主干网。而最常用的主干网方式。

后端网可将各种高速大容量存储设备或打印机等外设由 FDDI 网连到主机,以便解决设备间存在的瓶颈问题。FDDI 充裕的带宽能保证大容量的数据传输。

前端网是将高性能工作站通过 FDDI 网与主机连接,以便使主机上的公用数据库或传送大型作业给主机。像运行图象和 CAD/CAE/CAM 等带宽集中式的应用,最适应 FDDI 网的高信息量传输能力。

主干网是各种独立的低速 LAN。它是由网桥,路由器或网关通过 FDDI 光纤介质连在一起形成的。FDDI 有充分能力连接大量的 LAN。主干网为低速网络或设备提供了高速通路。由于光纤的距离优势,一个主干网足以覆盖一个大型企业或大学校园。

本刊启事

为更快地推动、促进我国计算机产业的迅猛发展,及时地为计算机科研、开发、生产、管理等部门提供计算机行业的最新发展动向,加速高新尖端技术、最新软硬件的开发、移植、引进,及时地为广大读者及计算机爱好者奉献更多更新的计算机专业技术资料,并使众多作者、译者脱颖而出,《计算机应用研究》杂志社现正开展优惠出版各类计算机技术专著业务,欢迎广大作者、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

提高应用软件可用性探析

张胜存

(双鸭山矿务局信息中心 佳木斯 155103)

摘 要 本文讨论了在开发设计应用软件时必须予以重视和考虑的用户界面设计、容错设计、安全保护设计以及提高软件可维护性设计等问题。这些问题的解决有助于从整体上提高应用软件的可靠性、可维护性、安全性和用户界面的友好性与容错性能等,从而开发出用户满意的、可用的应用软件。

关键词 应用软件 可用性 设计 用户界面 容错

0 概述

一个大中型计算机应用软件项目的开发,不同于编制一个小规模的自用程序或是简单的报表程序。前者比后者需要更多人员(一般3至10人或更多人员)的参与和更多时间(一般1至3年或更长时间)的付出,并且按照工程化方法来完成整个过程。

对于企业部门来说,应用软件系统一般是作为企业计算机管理信息系统的一个子系统来开发研制的,其成败则直接关系到整个企业计算机管理信息系统实施进程的快慢,甚至影响到整个企业现代化管理水平的提高。

某些应用软件项目开发失败,固然有一个或多个因素,如项目组织人员管理能力的强弱和人员配备得是否合理、开发单位参与人员业务能力即素质的高低与经验的多少,以及用户协作单位领导的支持与重视程度、用户协作单位与人员的配合好坏,还有开发所需软硬件环境是否具备,等等,不一而论。

但是,一个“不好”的应用软件,由于其不易为用户(或操作使用人员)所接受,从而用户不愿意或拒绝使用,也是导致软件生存期缩短甚至软件夭折的一个不容忽视的原因。可见,除去用户协作单位的因素外,开发人员自身所起的作用是重大而无可替代的。

说某个应用软件“不好”,是用户当使用它时将会很“耗神”与“费力”,总有一种不自在的感觉。对于这种软件,“不好”点可能体现在它:不好看——屏幕信息单调呆板不直观、或组织无序杂乱无章,打印的报表不够整洁清晰、不够美观漂亮;不好懂——屏幕或操作手册上尽是些不土不洋或土洋结合的“计算机行话”,与业务语言相差甚远,难理解;不好学——需要记住的东西太多(诸如未进行提示化处理的代码、各种操作规则与要求等等),记忆负担过重,某些内容记不住时或脱离操作手册的话就会束手无策、不知所措;不好用——操作复杂、不易掌握,或者需要学点儿计算机方面的专业知识后才能进行某些软件功能的简单操作;也

可能是软件测试不完备,当其正式投入运行时错误百出,以致无法进行正常的操作与运行,或软件脆弱、无容错能力,稍有不慎的操作就会使整个系统运行失败而不可恢复甚至瘫痪、死机;或软件结构处理不当,运行时只能走直线,不能拐弯,更不能退回重新开始,不能更改错误的操作步骤或修正错误的输入数据;等等。这样的软件,用户不满意、不认可、不买帐,其可用性则大大地打了折扣。

通过上述对“不好”的应用软件的分析,下面结合软件人员在开发应用软件过程中常常忽视的某些具体问题进行实质性地讨论。在此,并就笔者多年来在应用软件开发方面的实践体会,谈一些自己不成熟的想法与通常的做法,以起到抛砖引玉之作用,不当之处望同行指正。此处所谈及的几个特殊设计问题,并非一般软件工程中的总体设计与详细设计方面的内容,而是贯穿于整个软件开发全过程的一些特殊的加工与处理手段,期望运用这些手段来提高应用软件的“好”的程度,或有助于提高应用软件的开发质量和开发的成功率,最终作为一个可用的系统由用户所承认与接受。

1 提高用户界面的友好性设计

一个应用软件系统能否被用户接受,人机界面是很重要的因素。人机界面所包含的内容很多,首要的一条就是,它是软件开发人员通过计算机屏幕与用户进行交流的桥梁,是将用户与电算化后的用户业务相联系的纽带,也是用户操作计算机的媒介。故在这里使用“用户界面”这一称谓。在进行应用软件设计时,特别是在软件的编码阶段,应该在提高用户界面的友好性方面多做一些工作,以使操作人员操作起计算机来既简单又方便,有某种亲切、自然而贴近手工业务处理的感觉。这一点往往易被初次开发设计应用软件者所忽略。

一般来说,一个在功能方面开发得比较完备的应用软件,其在用户界面方面设计得“好”的标准是,只要用户(操作人员)谙熟自己的业务,经过软件开发设计人员对其进行简单的上机培训或指导,或花很少的时间参阅一下说明书,即可在短时间内甩掉纸、笔与计算器或算盘、甩掉手工帐本与文档卷宗,而过渡到直接使

用该软件系统在计算机上进行日常的业务处理工作。

友好的用户界面在应用软件系统中必须至少要体现出以下几点:

- 屏幕格式统一,功能说明与帮助,操作提示与引导

- 键盘使用规范化,选单操作

- 数据输入预赋初始值,容错

具体来讲,屏幕格式要有统一布局,界面应尽量简单规范、清晰美观;软件的各功能选项要有简明扼要的说明与适当的帮助信息;各操作步骤要进行提示与引导,提示信息要直观、一目了然,且易理解、无二义性。键盘的使用要规范化,每个键特别是各个功能键的定义要固定,起码在同一个软件中应如是;选单操作,各个功能选项应既能用光标选择又能用单键选取,并能进行鼠标操作,尽量减免文、数字的输入量。数据输入预赋初始值,单一选择项目与批量数据输入格式都预赋最合适的默认初始值,并且都力求达到使操作最方便;应用软件系统对用户的所有输入操作都应具有完备的容错能力;等等。

我想,只要我们的应用软件都做到了这些或仅仅做到其中一大部分的话,应该说其在可用性方面至少前进或提高了一大步。

下面仅举两个提高用户界面友好性的小例子:

(1) 功能菜单选择操作

一般应用软件系统在完成编码后,程序的各功能模块或主选单都要至少提供两种选择功能选项的基本方法,即

① 功能码选择法:亦即“热键”法。将各功能选项(一般不宜超过 10 个)依序定义一个一位的数字编号作为功能码,或使用功能标题的英文词汇首字母或其第一个汉字的汉语拼音字头作为功能码。当运行软件时,用户根据各功能选项的说明与帮助提示信息的描述,敲一下某功能选项所对应的功能码即可选定该功能。某些软件中使用一部分组合键作为功能码。

② 光标条选择法:现在绝大部分编程语言都支持该方法,某些较早期的高级语言虽不能直接支持该方法,但通过程序设计也是能实现的。当操作软件时,按动←、→、↑或↓键,将特殊颜色(一般为高亮度)显示的大光标条移到欲要选择的功能选项的位置上,再敲一下回车键↵即可选定该功能。

功能码方式和光标条方式选择功能选项的方法,其本质是一致的。软件中应注意预置一个最合适的功能码初始值,亦即注意将大光标条最初放置于最合适的一个功能选项位置上。同时软件中还要支持用鼠标进行选单点取的操作。

(2) 退出某一操作的方法

当用户在软件系统提示下选择菜单、输入数据或进行其它任何操作时,若不想继续向下进行、或想停止

当前这一操作过程甚至想直接返回前边的操作步骤去,则可敲击 Esc 键,软件应即刻停止或中断运行并退出当前的处理过程。这里,“Esc 键”就是一个规范的、唯一确定的完成中断或退出某一操作的特定的功能键。

要想让软件如期地达到这一目的,程序设计时就要在各个环节上对用户的操作按键进行恰当的处理。

2 容错设计

对于一个刚接触计算机不久的操作使用人员来说,如果他(她)操作计算机就是为了运行你设计的应用软件,假如软件在容错能力方面欠考虑的话,其结果可想而知。但对于一个有相当熟练程度的操作使用人员来说,如果运行一个容错能力很差的应用软件,则会使他(她)极度反感,甚至不愿使用这种软件,以至于宁肯手工处理相应的业务。因此,容错设计在应用软件设计过程中也是不可忽视的。

具体的容错设计方法是,在人机交互式的响应过程中,对所有输入(主要指键盘输入)数据的类型与范围全部进行有效性检查,各个处理过程采用错误陷阱技术,对各类能够预测但不能修正又不可避免的误差、或者无法预测是否有可能发生的错误进行黑箱式封装,出错时给用户提示一个易于理解的错误信息。

这里只讨论由于用户误操作所产生错误情况下的容错设计。用户的每一个操作步骤,包括单键选取、数值与字符串输入等,都应在应用软件系统的严密监视与控制之下进行。如若操作不当,或一个非法键被敲击,或输入的数据类型错、数值超限,等等情况,系统都应给出相应的提示信息与蜂鸣警告,并依具体情况可以要求用户重新进行操作。任何错误的操作,都不能也不应该造成整个应用软件系统的运行中断、运行失败甚至瘫痪;对最终用户来说,只要是在适当的允许范围内,随意的“瞎敲乱打”都不会使软件“卡壳”。因此,正确的数据被接受。错误的数据被排除,关键性的处理环节有确认步骤,前行与后退操作自如便捷,这是应用软件设计过程中应予考虑和遵守的起码准则,这也是软件可用性的必要条件。

输入数据的有效性或合理合法性检查与错误陷阱处理,是根据应用软件对用户业务流程和数据流程的处理过程与处理方式要求而灵活设计的,并且以不增加操作的复杂性为前提。

错误提示信息的显示形式与蜂鸣器的蜂响方式,是与界面的友好性联系在一起的。友好的用户界面也是减免出错特别是减免产生误操作错误机会的有效手段。一般来说,除标题、选单等信息外,错误提示信息与正常的帮助信息、操作步骤引导提示信息以及辅助显示信息等相比,在显示的色彩、亮度与窗口格式等方面是不相同的。一般情况下,错误提示信息窗口应放于屏幕正中间最为醒目的地方,且大小适度;色彩方面都应

较多地使用红或黄颜色作背景色,辅以黑、白或红色作前景色,可相反,并且以高亮度和/或闪烁的形式来显示,这样的色彩搭配与显示形式比较分明与刺激,易于引起警觉。而蜂鸣器鸣响的频率与音长也要因不同类型的操作错误而有所不同。

3 安全保护设计

一个应用软件系统的安全性及其保护机制,在整个软件的生命周期中一直处于极为重要的地位。应用软件的安全控制措施,包括数据合法性检查、操作员注册使用权控制、操作员操作权限控制、数据加密处理、数据备份与灾害恢复,也包括容错处理等多项。

作为一个应用软件,如果在安全保护方面设计得当、考虑周到,整个系统就能避免被人为破坏或非法使用,既不易被非授权用户所运行,又不易被授权用户越权使用某些特殊功能,偶遇灾害侵袭时亦能恢复正常工作,这样系统和系统数据的安全性就能得到一定的保障。否则,其后果将不堪设想,尤其是在网络环境下运行应用软件的单位或部门,当发生意外事故后有时会造成不可挽回的损失。特别是对于一些大中型应用软件系统,如果没有健全的安全保护措施,软件本身是很脆弱而不可用的,用了将会后患无穷。

下面分别对几种常见的应用软件安全保护方式进行初步的探讨。

(1)系统数据合法性检查:当用户启动应用软件系统后,在其注册前,软件本身应该对当前的运行环境或上次正常结束该软件运行时的现场进行一系列的合法性检查,特别是当发现有人为因素直接手工对软件系统本身的数据库进行了添加、修改、删除等操作后,系统将警告用户甚至拒绝该用户注册。

(2)操作员注册使用权控制,即用户身份识别。用户的系统维护人员可以利用软件提供的系统维护功能或辅助功能,对使用这一应用软件系统的各类人员(如用户单位领导、各业务处理人员和其他专/兼职操作人员等)定义操作员编号、口令字及各功能选项的操作权限。每一位操作人员运行软件时,都要向软件系统提供操作员编号和口令字,如果口令字输入错误,则不能有效地运行该软件或根本无法进入系统内,非授权操作者(即不合法用户)即被拒之门外了。操作员注册使用权控制,对于防范普通非授权用户来说,是一项很好的安全控制措施。

另外需注意,应允许每位已授权用户随时可以改变自己的口令,且口令在输入时不显示或显其它字符;口令字符串长度一般不少于三个字符,不允许重复一样的一个字符做口令字符串,等等。

(3)操作员操作功能权限控制:取得了注册权的用户,并不一定取得了对软件的所有功能的使用权。应用软件设计时对此应予控制。进入软件系统内的操作人员使用各功能时,各功能要根据操作员号取到相应的

操作权限,只有当操作权限为许可时,才允许进入其功能选项下进行相应的业务处理活动。限制用户的越权操作,对某些应用软件来说至关重要。

由于各应用软件系统的规模大小千差万别,且其各功能模块的层次分布与控制结构亦不尽相同,操作权限设计可有特权级法、口令法和权限选择法等多种。(详见《计算机应用研究》1996年第1号增刊,笔者所撰《应用软件中的操作权限设计》一文。)

(4)数据备份与恢复:数据的备份与恢复设计,在各应用软件中作为系统数据安全性控制的有效手段是必不可少的。用户(特别是用户的系统维护管理人员或网络管理员)可以并且应该经常(每日、每周、每月)使用系统提供的相应功能做数据备份工作,以便在系统或系统数据出现不可预料的错误、故障时进行最近期的有效性数据恢复,以使损失降低到最小程度。软件设计中,还应该对非正常掉电、关机或其它软硬件故障造成的软件中断运行时的现场进行某种程度的恢复问题予以考虑。

(5)数据加密处理:数据加密处理是数据乃至整个应用软件系统的保密控制措施之一。每一个应用软件系统一般都要有十几、几十甚至成百上千个主数据库文件及更多个附属或辅助文件,并且随着时间的推移,其数目亦会越来越多。软件应提供一种或几种有效的加密方式(算法),对所有重点库甚至全部数据库的存储进行加密处理,以使得在其它任何软件环境下都无法对库内容进行窥视。

加解密处理是以牺牲更多的存取时间和存储空间来换取数据本身的安全性作为代价的。虽如此,这对于像金融单位的、企业财会部门等的应用软件系统来说是十分值得的和必须的。

4 提高可维护性设计

任何事物都不会是一成不变的,每一个应用软件也是如此。随着改革的深化,经济在发展、市场在变化;业务部门也存在着机构调整、管理模式改变、业务流程变动等等情况。内外环境的不断改变,要求软件也要进行相应的适应性改动、调整,或功能完善与增强、性能改进与提高,以及应用范围适度扩充,等等;还有,需要及时排除验收测试过程中未发现的隐藏的软件错误。这就是对软件进行维护。

如果软件的可维护性较好,这种改动或调整就容易做;否则,牵一发而动全身,其困难程度是可想而知的。这也对应用软件的开发人员提出了较高的要求。提高整个应用软件的可维护性是提高其可用性的一项重要措施。随着计算机产业的迅猛发展,各行业各领域的各种新的应用软件不断涌现,版本不更新或是维护跟不上去的应用软件很容易被用户所抛弃。

不管怎样,对于大中型应用软件的维护工作必须慎重从事,不允许有丝毫的马虎大意。这种维护须是在

应用软件系统自身的控制之下进行的,或者是系统所允许的,否则将会引起系统的混乱。

这里,我们将整个软件系统的维护分为软件维护与数据维护,前者即源程序维护及相关文档资料的改动,后者为系统运行所用的各类参照数据,基础数据和用户数据的维护。对于如何提高应用软件的源程序的可维护性,像采用结构化设计方法、利用自动开发工具、使用最先进最新式的支撑环境、提高软件文档的质量等,市面有许多软件工程方面的书中有详尽的描述,这里不再赘言。

对于数据维护,把将来在使用过程中有可能需要改变的各类参照数据、基础数据等设定为一种特殊的用户数据——应用软件的系统数据。在使用面向过程的系统开发环境下开发应用软件时,这是提高应用软件系统在可修改性和可扩充性方面灵活性的一大诀窍。对于系统数据,软件在设计时应提供能由授予了相应权限的用户进行增、删、改操作的功能入口,并尽可能地预测将来用户使用过程中会产生的某些变化,将这些变化的预测情况尽量地转化为应用软件系统能够控制的系统数据。在系统数据所标定的各业务流程、处理过程、软硬件运行环境等改变时,用户可根据需要自己进行适应性的系统数据维护,这将省去进行源程序修改所存在的诸多弊端。需说明的是,这也同时增加应用软件在这方面设计的复杂性与难度。

5 设计约定

应用软件进入详细设计阶段后,为了能在计算机上有效地实现软件所具有的各种功能,对某些方面的具体内容进行一定的约定(或称为设计约束)是很有好处的。这种约定是开发的准则和规范,甚至是质量的保证;或者说是整个项目组的所有开发人员必须遵循的质量标准,也是他们进行最底层相互配合与联系的纽带。软件开发人员应该严格按照这些约定进行设计与编码,如此有望生产出一个较实用的应用软件来。

对于面向过程的应用软件开发环境来说,设计约束大多体现在系统环境参量、全局变量、过程与公用子程序、功能键设定,以及数据库文件工作区、子目录、工作目录、程序模块与文件命名等等。下面试举两例:

(1)功能键定义

将每一个功能键赋予各自特定的功能,便于用户上机操作;意义相似或相近的功能赋予同一个键位亦便于记忆。这也是提高用户界面友好性的一项小举措。例如在“煤炭销售软件”中使用了如下的功能键定义,它们用作在某个主屏幕画面下对操作类型的选择:

- F1 帮助、说明、显示附加提示信息
- F2 显示、查询显示、翻页显示、显示附加信息
- F3 作图、绘图、显示图形
- F4 输入、录入、增加(增加日请车计划)

F5 修改、调整(调整配车方案)、变更(变更月运输计划)

F6 删除、作废(作废合同)、取消与恢复

F7 查找、检索、跳转

F8 计算、重新计算、统计、汇总(求和)

F9 标志、标记(标记车号)、选择、匹配(匹配车皮)

F10 打印、拷屏

F11 磁盘操作、磁带操作、对点通讯(发送或接收数据)

F12 结束、确认、继续

PgUp 前屏、上屏

PgDn 后屏、下屏

Home 第一屏、第一笔数据

End 最后屏、最末笔数据

Esc 退出(中断正常操作、非正常退出、返回前一个操作步骤)

每个键位根据不同的软件模块、功能选项或不同的屏幕内容,其功能标题的文字定义不尽相同,但意义必须相近或相似。由于键盘上功能键的特定布局,其F1、F4、F5、F8、F9、F12等键位操作起来较为方便些,因此在该软件中对它们分配了较为重要且使用频度最高的一些内容。

(2)子目录设定

将不同性质的数据文件和/或程序文件分别存放或者安装在主机(网络服务器或单机)硬盘的专有目录或特定目录的不同子目录下,建立层次布局,既方便操作,又容易管理,实不失为一种好办法。

例如,“煤炭销售软件”的安装程序在用户的网络服务器硬盘中创建一个Y:\MTXS目录作为该应用软件的专有目录,在此基础上再分别建立几个子目录分别存放不同类型的文件,如下:

Y:\MTXS\PRG 存放各类程序文件

Y:\MTXS\DBF 存放数据库文件,及其索引、备注等文件

Y:\MTXS\DOC 存放各种有关该软件的文档

Y:\MTXS\FMT 存放格式文件

Y:\MTXS\FMT\SCREEN 存放屏幕文件

Y:\MTXS\FMT\REPORT 存放报表文件

Y:\MTXS\FMT\LABEL 存放标签文件

Y:\MTXS\FMT\MENU 存放菜单文件

Y:\MTXS\FMT\PROJECT 存放项目文件

Y:\MTXS\TXT 存放文本文件

Y:\MTXS\MEM 存放内存变量文件和宏定义文件等

Y:\MTXS\GEN 存放各种生成的屏幕、查询、菜单程序文件

Y:\MTXS\TMP 存放其它各种杂项或临时文

二维线段的矩形窗口逐边裁剪算法研究

刘斌 王勇 黄树槐

(华中理工大学机械学院 武汉 430074)

摘 要 本文详细介绍了一种二维线段的矩形窗口逐边裁剪算法。该算法充分利用了矩形窗口的凸性和窗口边界平行于坐标轴的特点,通过对二维线段与窗口边界的各种相对位置关系进行详细分类,以减少裁剪求交的次,从而减少了算法的算术运算量,达到快速裁剪线段的目的。该算法原理简单,便于理解和实现,且运算速度较快。

关键词 裁剪 矩形窗口 二维线段 计算机图形学 CAD

1 引言

线段的窗口裁剪是计算机图形学中的一项基本操作。在二维线段的窗口裁剪中,最简单也是最常用的是矩形窗口的裁剪。国内外提出了多种多样的关于二维线段的矩形窗口裁剪算法,其中最著名的有:Cohen-Sutherland的分区编码算法[1]、基于硬件实现的中点分割算法[2]、梁友栋-Barshy算法[3]等。近年来,对这个问题的研究仍有许多,如Nicholl-Lee-Nicholl算法[4]、Andreev-Sofianska算法[5]以及FLC算法[6]等。另外,文献[7]曾提出一种以被裁线段为核心,并可以直接拓广到二维和三维图形进行裁剪的算法。本文提出一种以窗口边界为核心,对被裁线段进行逐边裁剪的算法。该算法在充分利用矩形窗口的凸性和窗口边界平行于坐标轴的特点,通过对二维被裁线段与窗口边界的各种相对位置关系进行详细分类,以简单判断为主,从而减少了裁剪求交的次,减少了算法的算术运算量,达到快速裁剪线段的目的。

2 被裁剪线段与窗口边界的位置关系分类

大家知道,二维线段的矩形窗口裁剪,就是要判断并求得该直线段哪些部分在窗口内,哪些部分在窗口边界上和哪些部分在窗口外,从而对于那些在窗口边界上和在窗口内的部分进行显示处理。

2.1 线段与矩形窗口的相对位置关系

通过分析可知,任一给定的线段与矩形窗口的相对位置关系只有四种,即:

(1)线段与矩形窗口的边界无交点,此时线段或全在窗口外,或全在窗口内,如图1中的线段AB、UV和JK;

(2)线段与矩形窗口的边界只有一个交点,如图1

中的线段EF、GH、CD和ST;

(3)线段与矩形窗口的边界有两个交点,如图1中的线段LM;

(4)线段全部或部分落在窗口的边界上,如图1中的线段NP和QR。

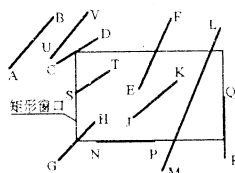


图1 线段与矩形窗口的相对位置关系

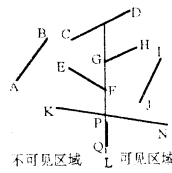


图2 线段与矩形窗口边界的相对位置关系分类

因此,对于上述四种相对位置关系来说,矩形窗口对线段的裁剪过程表现为:

(1)对与矩形窗口边界无交点的线段进行全部裁掉或全部保留;

(2)对与矩形窗口边界只有一个交点的线段,则以交点为分界点,一部分裁掉,而另一部分则保留;另外,还有两种特殊情况;

①当此交点为被裁线段的端点时,该线段应全部裁掉或全部保留;

②当此交点为窗口角点且线段的两个端点均在窗口之外时,则线段应全部裁掉;

(3)对与矩形窗口边界有两个交点的线段,则保留两交点之间的线段,而将其他部分裁掉;

(4)对于全部或部分落在窗口边界上的线段,则保留线段与窗口边界公共的部分。

2.2 线段与矩形窗口边界的相对位置关系分类

由于矩形窗口边界与坐标轴平行的特点,因此,线段与矩形窗口的四条边界的相对位置关系分类具有相似性。不失一般性,设以矩形窗口某边界线段L为

件

这种结构的定义,既没有过多目录、子目录造成的管理负担,又避免了同一目录下含有多种类型的文件

和太多数量的文件造成的查找和存取困难;简单清晰、直观明了。不但给程序编制人员带来了方便,也深受用户的软件管理维护人员的欢迎。

收稿日期:1996年6月1日

考虑对象,则边界线段所在的直线将平面区域分为两个部分,其中窗口侧部分称为可见区域,而另一部分称为不可见区域,如图 2 所示。则任意被裁线段与窗口边界线段的位置关系共有如下七种类型:

类型一:被裁线段全在边界的不可见区域,如图 2 中的 AB 线段;

类型二:被裁线段的两个端点分别在边界 L 所在直线的两侧,但被裁线段与边界线段 L 无交点,如图 2 中的 CD 线段;

类型三:被裁线段的两个端点分别在边界 L 所在直线的两侧,被裁线段与边界线段 L 有一个交点,如图 2 中的 KN 线段;

类型四:被裁线段的一个端点在边界 L 所在直线上,而另一个端点在边界 L 的不可见区域,如图 2 中的 EF 线段;

类型五:被裁线段的一个端点在边界 L 所在直线上,而另一个端点在边界 L 的可见区域,如图 2 中的 GH 线段;

类型六:被裁线段的两个端点全在边界 L 的可见区域,如图 2 中的 IJ 线段;

类型七:被裁线段的两个端点均落在边界 L 所在直线上,如图 2 中的 PQ 线段;

3 逐边裁剪算法的原理与实现

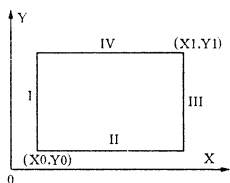


图 3 矩形窗口定义及逐边裁剪顺序

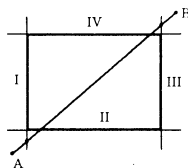


图 4 算法的算术运算量估计

设矩形窗口左下角顶点的坐标为 (X_0, Y_0) , 右上角顶点的坐标为 (X_1, Y_1) 。其边界逐边裁剪顺序为边界 I、II、III 和 IV, 如图 3 所示。若已知某线段, 其两端点的坐标分别为 (X_a, Y_a) 和 (X_b, Y_b) , 则矩形窗口对其进行逐边裁剪的算法原理和实现过程如下:

(1) 对被裁线段进行 BOX 测试, 即:

a. 当符合下列不等式之一者, 则被裁线段应全裁掉, 并转(5); 如图 1 中的 AB 线段被全部裁掉。为了叙述方便, 称此测试为 BOX-OUT 测试:

$$X_0 > \max(X_a, X_b) \quad X_1 < \min(X_a, X_b)$$

$$Y_0 > \max(Y_a, Y_b) \quad Y_1 < \min(Y_a, Y_b)$$

b. 当同时符合下列条件时, 则被裁线段应全保留, 并转(4); 如图 1 中的 JK 线段被全部保留。为了叙述方便, 同样称此测试为 BOX-IN 测试:

$$X_0 \leq \min(X_a, X_b) \quad X_1 \geq \max(X_a, X_b)$$

$$Y_0 \leq \min(Y_a, Y_b) \quad Y_1 \geq \max(Y_a, Y_b)$$

若不能通过上述 BOX-OUT 和 BOX-IN 测试进行裁剪的线段, 转(2);

(2) 用窗口边界 I 对线段进行裁剪, 即将被裁线段与边界 I 进行类型测试, 并根据不同类型作下列相应的裁剪处理。由于类型一在(1)中已作处理, 故此时不会出现类型一的情况, 并且类型七中两个端点均在或均不在边界上的线段也在(1)中作了处理。

a. 若为类型二, 此时交点的坐标 $Y \in [Y_0, Y_1]$, 将交点替换线段在边界不可见区域的端点, 如图 1 中的 UV 线段被边界 I 裁剪后, 将交点替换端点 U; 并对修正后的线段作 BOX-OUT 测试, 若是, 转(5); 若否, 转(3);

b. 若为类型三, 此时交点的坐标 $Y \in [Y_0, Y_1]$, 将交点替换线段在边界不可见区域的端点, 并对修正后的线段作 BOX-IN 测试, 若是, 转(4); 否则, 转(3); 在对修正后的线段作 BOX-IN 测试时, 只对线段的另一个端点进行测试即可。

c. 若为类型四, 转(5);

d. 若为类型五、类型六和类型七, 不作处理, 转(3);

(3) 依次取窗口边界 II、III 和 IV 对被裁线段作类似于以上述窗口边界 I 为核心的裁剪操作, 并当边界 IV 对线段进行裁剪时, 在修正线段的一个端点后, 无须对修正后的线段作 BOX 测试, 因为此时的线段肯定在矩形窗口内。

(4) 得裁剪后保留的线段, 并将此线段加以显示;

(5) 裁剪结束。

该算法只有在被裁线段与窗口边界所在直线有交点时, 才对被裁线段作端点修正处理, 并对修正后的线段作 BOX 测试, 以加快裁剪速度。

4 交点计算及算法的算术运算量估计

对于平行于某坐标轴的线段的裁剪, 可以利用矩形窗口边界平行于坐标轴的特点, 直接导出交点坐标, 而无须进行求交运算。而对于一般位置的线段 AB, 如图 4 中所示, 设其两端点的坐标分别为 (X_A, Y_A) 和 (X_B, Y_B) , 则其直线方程为:

$$Y = K(X - X_A) + Y_A \quad \text{或} \quad X = KK(Y - Y_A) + X_A$$

$$\text{其中 } L = (Y_B - Y_A) / (X_B - X_A), KK = 1/K.$$

由于该算法采用以矩形窗口边界为核心对被裁线段进行逐边裁剪的方法, 因此, 在最特殊的情况如图 4 中的 AB 线段位置, 此时算法的算术运算量分别为: 加法四次, 减法六次, 乘法四次和除法两次。其他一般位置线段的裁剪, 该算法的算术运算量均不大于这个数目。另外, 有的位置线段被裁剪时无须进行任何算术运算, 但只要不能用 BOX-OUT 和 BOX-IN 测试出的一般位置线段的裁剪, 其算术运算量至少为加法一次, 减法三次, 乘法一次和除法一次。

5 结论

本文介绍了一种以矩形窗口边界为核心, 对被裁线段进行逐边裁剪的算法。在算法实现过程中, 动态地

一种基于 POP3 的 INTERNET 电子邮件软件的设计与实现

李建华 喻 成

(长沙铁道学院信息技术研究中心 410075)

摘 要 本文介绍了邮局协议 POP3(Post Office Protocol Version 3)的工作原理,讨论了基于 POP3 协议的 INTERNET 电子邮件用户代理软件的设计思想,并提出了一种电子邮件用户代理软件的设计方案。

关键词 INTERNET 电子邮件 用户代理 POP3 SMTP 客户机/服务器

1 前言

INTERNET 是目前最大的全球性计算机网络。在 INTERNET 上最主要的应用就是电子邮件(E-Mail)。INTERNET 上的电子邮件系统都使用简单邮件传输协议 SMTP(Simple Mail Transport Protocol)来传递邮件。发送电子邮件的节点必须支持 SMTP,而接收或转发电子邮件的节点还必须提供足够的资源(CPU 时间、磁盘空间等等)来维护应该报文传输系统 MTS(包括报文存储器和报文传输代理 MTS)来接收与转发邮件并负责本地邮件的分发。在 UNIX 系统中,提供了称为 Sendmail 的精灵进程来完成邮件的传递、接收与转发。而对于想要通过 INTERNET 来发送与接收电子邮件的广大 PC 用户来说,维护一个 MTS 无疑是不切实际的。为了进入 INTERNET 发送与接收自己的 E-Mail,PC 用户可以以终端仿真方式到提供 E-Mail 服务的 UNIX 主机上,但这种方式既落后又因为联机处理而通讯费用昂贵。为解决这一问题,人们提出了一种基于客户机/服务器模式的方案:即在能支持 MTS 的邮件主机上为 PC 节点提供一种邮件下载(mail-drop)服务。PC 节点机通过某种协议与邮件

主机进行互连,从邮件主机上浏览或下载自己的 E-Mail。目前 INTERNET 上的这种协议有:POP3(Post Office Protocol Version 3)以及 IMAP(Interactive Mail Access Protocol)等。

本文重点讨论 POP3 协议的工作原理和基于 POP3 协议的用户代理(UA)的设计。

2 POP3 的工作原理

POP3 称为邮局协议版本 3,其核心是采用客户机/服务器模式为 PC 节点机提供一种读取邮件主机(该主机支持 MTS)上邮件的机制。提供 POP3 服务的主机称为 POP3 服务器,请求 POP3 服务的主机(通常为 PC 机)称为 POP3 客户机。

在基于 UNIX 操作系统的 POP3 服务器主机上,通过编辑 inetd.conf 与 services 文件增加 POP3 服务。POP3 服务器进程也是一精灵进程,它在 TCP 端口 110 上监听 POP3 客户机的连接请求。当需要 POP3 服务的客户机与服务器主机建立一个 TCP 连接(通过 TCP 端口 110)时,POP3 服务器发送一个如:“+OK POP SERVER EADY”字样的握手信号。于是,客户机与 POP3 服务器便开始一个 POP3 会话,它们分别交换命令与应答信息,直到连接被关闭或被中止为止。

POP3 的命令包含一个关键字和相应的变量;而

收稿日期:1996 年 6 月 8 日

将交点对被裁线段的端点进行修正,同时对修正后的线段作 BOX 测试,以加快对线段的裁剪速度。该算法充分利用了矩形窗口的凸性和窗口边界平行于坐标轴的特点,通过对二维被裁线段与窗口边界的各种相对位置关系进行详细分类,以简单判断为主,从而减少了裁剪求交的次数,使算法的算术运算量减少到最小。该裁剪算法原理简单,便于理解和实现,且运算速度较快。

参考文献

- 1 W. M. Newman and R. F. Sproull, Principles of Interactive Computer Graphics, McGrawHill, New York, 1979
- 2 R. F. Sproull and I. E. Southerland, A clipping Divider, FJCC, Thompson Books, Washington D. C. 1968
- 3 Y. D. Liang and B. A. Barsky, A New Concept and Method for Line Clipping, ACM Transactions on Graphics, 3 : 1 (1984)
- 4 T. M. Nicholl, D. T. Lee and R. A. Nicholl, An Efficient New Algorithm for 2D Line Clipping, SIGGRAPH'87, Computer Graphics, 21 : 4(1987)
- 5 R. Andreev, and E. Sofianska, New algorithm for two dimensional line clipping, Computers & Graphics, 15 : 4 (1991)P519-526
- 6 王骏,梁友栋,彭群生,具有最少算术量的二维线裁剪算法,计算机学报,1991.14(7): 495-504
- 7 刘卡林,孔庆复,介绍一种新的裁剪算法:计算机应用研究,1989(4): 27-30

POP3 服务器的应答信息包含一个应答标识符(“+OK”或“-ERR”)及一个可能带有附加信息的关键字,所有命令与应答均以一对回车换行符结尾。

一个 POP3 会话包含三种状态:一旦建立 TCP 连接且 POP3 服务器发出握手信号,会话便进入授权(AUTHORIZATION)状态;在授权状态,客户机必须向服务器证明自己的身份,在成功地证明身份以后,会话进入事务(TRANSACTION)状态;在这一状态,POP3 服务器处理客户机发出的读取邮件的请求事务,这些请求事务包括:获得邮箱中报文的数量与长度、检索邮箱中的报文、删除邮箱中的报文等等。当执行完这些事务后,客户机通过向服务器发 QUIT 命令而进入修改(UPDATE)状态;在该状态,POP3 服务器提交事务、释放资源。会话结束,连接被关闭。

POP3 协议中包含以下主要命令:

USER name 向服务器验证用户注册名(用户 ID)

PASS string 向服务器验证用户口令

QUIT 如果验证身份失败,关闭会话与连接(以上三条命令在授权状态时有效。)

STAT 获得用户邮箱中报文的数量与总长度

LIST[msg]列出用户邮箱中某条或所有报文的信息

RETR msg 检索用户邮箱中某条报文的内容

DELE msg 删除用户邮箱中某条报文(在事务状态下只作删除标记)

LAST 获得用户邮箱中以被访问过的报文的最大 ID 号

REST 复位删除标志及最大被访问的报文 ID 号

QUIT 退出事务状态进入 UPDATE 状态(以上命令在事务状态时有效)

图 1 描述了一个完整的 POP3 会话过程。

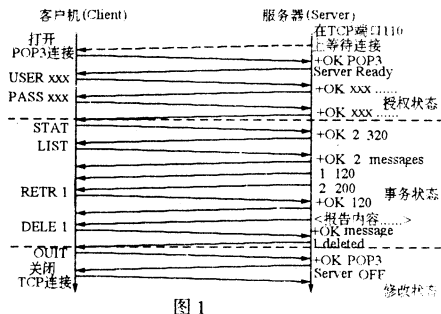


图 1

3 基于 POP3 的 INTERNET 电子邮件用户代理(UA)软件的设计

在邮件系统中,用户代理(UA)是一个客户端应用程序,它的任务是编制报文、检查拼写错误、格式化报文、发送和接收报文、以及管理存储报文用的电子文件

夹等等。PC 节点机上的 INTERNET 电子邮件用户代理通过 SMTP 来发送邮件给与 PC 相连的邮件主机,SMTP 是一种基于 TCP/IP 的协议,描述了邮件报文的格式以及传递邮件的方法。主要用于主机与主机之间的邮件交换。SMTP 事务处理分以下三个阶段:(1) MAIL 命令,并标识发送者;(2) 一组 RCPT 命令,并给出接收者信息;(3) DATA 命令,用于给出邮件数据。最后是邮件结尾标识,以确认该事务。因此,PC 机上的邮件代理程序也是通过直述三个步骤来与邮件主机上的 SMTP 服务器进程进行通信,以完成电子邮件的发送任务。关于 SMTP 协议的具体细节,有兴趣的读者阅读参考文献[3]。同时,用户代理程序通过 POP3 协议从提供 POP3 服务的邮件主机上读取邮件。当然,SMTP 服务器与 POP3 服务器可以是同一主机也可以不是同一主机。

PC 机上的用户代理程序与邮件主机之间的关系如图 2:

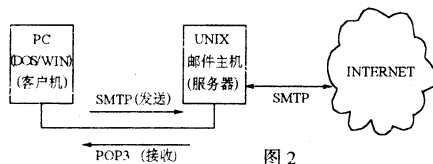


图 2

用户代理程序利用 SMTP 发送电子邮件给邮件主机的处理流程如图 3:

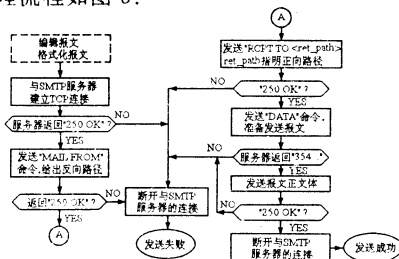


图 3

用户代理程序利用 POP3 从邮件主机读取电子邮件的处理流程如图 4:

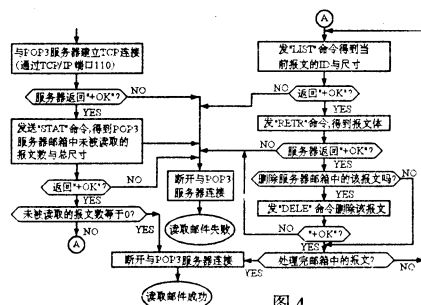


图 4

4 结束语

上面讨论了基于 POP3 的 INTERNET 电子邮件用户代理程序的设计思想。我们在 MSWINDOWS 平台上利用 WINSOCK 开发了一个基于 POP3 的 INTE

客户/服务器环境下数据库转换的实现

刘友华

闵建洪

(南京大学信息管理系 210093)(南京大学计算机科学与技术系 210093)

摘要 本文首先探讨了传统 MIS 模式向 C/S 计算环境转换的必要性,然后对其中的数据库转换与互连方法进行了描述与比较,在此基础上分析了用程序接口法实现数据库转换的过程,并给出了相应的实现算法。

关键词 客户/服务器 数据库转换 管理信息系统

1 传统 MIS 模式向 C/S 环境转换的必要性

近年来,计算机技术的发展有三大趋势:硬件规模小型化,导致个人计算机迅速普及,并渗透到各个领域;网络技术逐步成熟,使得资源得以共享,并能充分发挥各种计算机的优势;GUI 技术的普及,特别是多媒体技术的发展,使计算机对人们更友善、更方便。能充分组合这些技术的计算机应用模式,便是客户/服务器(client/Server 简称 C/S)计算环境;而最能让各种中小部门和单位,特别是采用管理信息系统(Management Information System 简称 MIS)的单位,便是基于 PC-LAN 的 C/S 计算。

我国将计算机真正用于企事业信息管理的时间并不长,且基本上选用的是如下的三条技术路线:第一种方式是以大中型机作为主机,通过终端与主机相连的集中式管理系统,这种方式软硬件系统一般均由厂家提供,造价较高,系统难以维护和升级,且开放性差;第二种方式是将微机联成网络,实现共享的文件服务系统,这种方式下的所有应用处理均在工作站上完成,服务器虽是一台高档计算机,但实现的不过是工作站的文件存储及打印扩展,其网络效率不是很高;第三种方式是在微机上实现的单机应用,该方式因为资源不能得到共享,各自独立的模块难以进行信息交换和集成,不可能充分发挥 MIS 的作用和效能。而客户/服务器计算方式,除了具备资源共享的网络方式的优点外,能更充分地发挥服务器的作用,处理任务由客户和服务双方分担,较高级较复杂的任务(如数据库服务器所完成的高级数据管理工作)由服务器来完成。

因此,将传统 MIS 的应用模式转变成客户/服

器计算模式,对于发挥计算机的作用,接纳应用新技术,扩展应用规模,减少系统开发周期,有着巨大的经济效益。

2 数据库转换及互连的方法及其比较

由传统 MIS 模式向 C/S 计算模式发展转换时,除实现硬件实体的转换、计算模式的转换外,还必须考虑异种数据库的转换及互连。当某一应用由旧的数据库系统(如 XBASE 系列)上升为新的数据库系统(如 Oracle, Informiz 或 Sybase)之后,或者与外界计算机系统发生信息交换时,就出现了两个系统中的数据如何共享与转换的问题。解决这个问题的方法有如下三种:

1) 数据更新法

用这种方法实现旧新数据库系统转换,是指将旧的数据格式更新为新的数据格式,同时,应用程序和用户界面也将被更新,从而达到旧系统完全被新系统所取代。数据更新法有利于保持数据结构与应用接口的一致性,便于统一管理,但数据更新和应用程序的重构工作量比较大。

2) 数据互连法/ODBC 法

数据互连法或称着 ODBC(Open DataBase Connectivity)法使得一个数据库应用程序能访问不同的数据库系统,从而允许应用程序开发者在进行数据库应用开发时,不用针对某个特点的 DBMS。数据库互连法通过动态连接库(驱动管理程序)的使用,某数据库应用能调用相应的动态库(驱动程序),从而访问指定的数据库,达到异种数据库之间的互连。数据互连法的应用,可在新的系统环境(C/S)下能方便地对老系统数据进行访问与交换,而不用经过数据转换与系统更新,也能进行方便的操作,从而节省大量的转换工作。

3) 程序接口法

详细介绍。

参考文献

- 1 William 等. INTERNET 使用大全. 清华大学出版社, 1995. 5
- 2 Harley Hahn. The Internet Complete Reference. 学苑出版社, 1994. 11

收稿日期:1996年6月30日

RNET 电子邮件用户代理软件。我们用 Borland C++ 开发一个 Visual Basic 的 WINSOCK 控件,并在 Visual Basic 中利用该控件实现客户端的 SMTP 与 POP3,同时利用 VB 的可视化编程特性设计报文编辑、邮件管理等功能以及用户界面。由于篇幅限制,在这里不作

应用程序接口可以直接在操作系统下运行,也可以嵌入数据库应用程序之中来实现不同数据库的连接与交换。该方法不仅能解决异构数据库之间的连接和信息交换,还可用来解决许多普通应用程序难以解决的复杂问题。

3 用程序接口法实现数据库转换的过程

传统的数据库转换(即不采用我们上述的方法),大都是用手工操作实现。如要实现 Foxpro 下数据库到 Oracle 数据库的转换,则首先进入 Foxpro 系统的命令行状态,将 DBF 文件转换成一个临时的文本文件,再退回到操作系统状态,然后进入 Oracle 系统的 SQL * PLUS 命令行状态,建立一临时性基表,将前面产生的文本文件内容转入该临时基表中,最后由此临时基表的数据对其它基表数据进行更新。由于这一过程难以用批处理的方式自动实现,而且在 Oracle 的应用程序中(如 SQL * FORMS)运行 Foxpro 系统较为困难,大多受内存等资源的限制,因而难以进行集成环境的调用。

利用 Pro * C 编程,则可以实现这一直接调用过程。将 Pro * C 程序经过预编译、编译和连接而产生可执行文件之后,可在操作系统下运行,或在 SQL * FORMS 的触发器中直接调用而实现这一转换过程。

4 数据库转换的实现

下面以 Foxpro 数据库向 Oracle 数据库的数据传输为例,来说明以程序接口(Pro * C)的方式实现这一数据转换的方法。

Foxpro 数据库文件格式可分为三个部分:头文件信息部分;字段结构说明部分;数据记录部分。

头文件信息部分共 32 字节,其含义如下:

字节	含义
1	标识符
2—4	最近修改日期(yy/mm/dd)
5—8	文件现有记录数
9—10	数据库结构所占字节数
11—12	每条记录的长度
13—32	保留

字段结构说明部分,其长度由字数决定,每字段说明占 32 字节,含义如下:

字节	含义
1—10	字段名
11	保留
12	字段类型(C、N、D、L、M、G)
13—16	保留
17	字段长度
18	小数位数

字段描述后,结束符为 0DH,每条记录的第一位为删除标志(24H 表示删除,20H 未删),整个文件结束符为 1AH。

现有 Client 编 Foxpro 库文件 ZGDHF.DBF,其库结构为:

ZGGH C(4) /* 职工工号 */

JE N(6,2) /* 电话费金额 */

在 Server 上有工资管理系统下的 Oracle 基表 PAY01,令有与 ZGDHF.DBF 库对应的两个域:

ZGGH C(4) /* 职工工号 */

.

.

.

DHF N(6,2) /* 电话费 */

现实现 ZGDHF.DBF 中的 JE 对基表 PAY01 的更新,更新条件为“职工工号”相同。其算法的主要步骤如下:

1)定义 DECLARE 节:

VARCHAR uid[20]; /* 用户名 */

VARCHAR pwd[20]; /* 口令 */

VARCHAR yggh[4]; /* 职工工号 */

float dhf; /* 电话费 */

2)定义 SQL 通讯区 SQLCA;

3)定义各类变量:

FILE *fp; /* 文件名柄 */

unsigned char del-flag; /* 记录删除标志 */

int record-num, struct-len; /* 记录条数及库结构长度 */

4)实现与 Oracle 的连接;

5)打开 ZGDHF.DBF 文件:

fp=fopen("ZGDHF.DBF","rb");

6)读取记录数及库结构长度:

fseek(fp,4,0);

read(fp,&record-num,4);

read(fp,&struct-len,2);

7)记录是否为空:

if(record-num<1) goto(16) else goto(8);

8)文件指针定位在第一记录:

i=1;

fseek(fp,struct-len,0);

9)记录是否读完:

if(i>record-num) goto(15) else goto(10);

10)读删除标志:

read(fp,del-flag,1);

11)该记录是否删除:

if(del-flag!=0X24) goto(12) else goto(14);

12)读取当前记录数据(职工工号、电话费用);

read(fp,yggh.arr,4);

yggh.len=strlen(yggh.arr);read(fp,&dhf,b);

13)执行 Oracle 的基表更新语句:

新型燃油系统的计算机仿真

马力中 陆展华

周 苇

(上海工程技术大学六系 200335) (上海铁路局 200071)

摘 要 本文数值模拟了一种适合劣质燃油(重油)的新型燃油系统,提出了处理喷油器和燃油分配器边界条件的准维模型以及用于喷油泵边界条件的新模型。该燃油系统已应用于 MAN B&W L60MC/MCE 等大功率柴油机。与常规的数值模拟相比,求解管内流动的二阶双曲型偏微分方程的边界条件发生了很大变化,这是燃油系统研究的难点和关键。模拟结果与试验结果较为一致。本研究还进行了不同功率点变型方案的模拟研究和结构参数的优化设计分析,以满足不同船舶的需要。

关键词 数值模拟 优化分析 燃油系统 柴油机

1 引言

目前国内船用大功率柴油机制造在技术和质量方面均已进入国际先进行列,部分产品已进入国际市场。但传统的研究、生产、管理方式所造成的制造周期长、效率低严重地影响了产品在国际市场的竞争能力。因此必须加速推进计算机应用开发研究,使该领域发生一场深刻的技术变革。为此,中国船舶工业总公司(原六机部)组织上海交大、上海工程大等七所高校和研究所以共同开发《计算机辅助船用大功率柴油机设计、制造、管理集成系统》第一期工程^[1],简称 CADIS-I,并在全国造船行业推广使用。该集成系统包括:设计信息管理子系统(DDS)、生产管理子系统(DPS)、CAD 子系统和 CAPP/CAM 子系统等,该集成系统功能框图

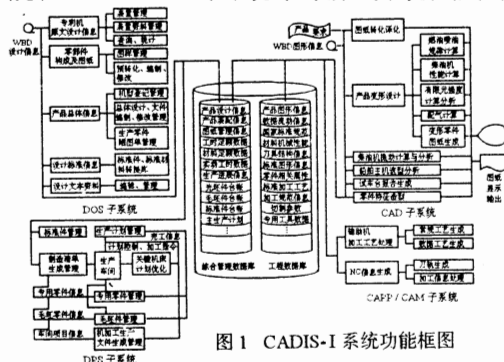


图1 CADIS-I 系统功能框图

本项研究承国家重点企业技术开发项目资助,并为上海市高校科技发展基金资助项目。

收稿日期:1996年7月24日

见图1^[2]。

经论证,集成系统主要应用产品对象为引进的国际上最先进的 B&W 系列大功率柴油机。该机运用了先进的燃油喷射技术。为了加强应变能力,开发变型机设计功能,满足船车对发动机功率 P 和转速 N 的不同需求,必须进行燃油系统的数值模拟,并通过交互设计,产生、修改有关图纸,但由于“无法获得有关理论计算资料和全套各种配置的图纸,只能走自行设计的道路”^[1]。

本文简介部分研究结果。

2 新型燃油系统简介

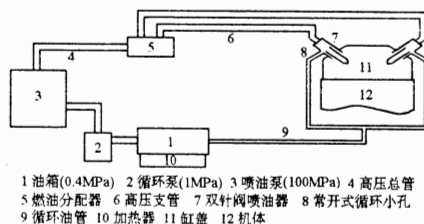


图2 燃油系统原理示意图

“一泵二嘴四针阀”循环式燃油系统原理示意图如图2。该喷油泵没有传统的出油阀,而设有构思巧妙的“常闭贯通型套筒式吸油滑阀”,该阀利用弹簧力的平衡和燃油压力的变化控制燃油从循环泵进入喷油泵腔内的通道。除了喷射前自动开启约 50°CaA (凸轮转角)进行吸油外,其他时刻为常闭,喷油泵3经滑阀的内腔与高压总管4之间始终贯通。每缸有两个喷油器,且每个喷油器有内外双针阀(分别称为 slide, spindle)。燃

```
EXEC SQL UPDATE PAY01 SET DHF = :dhf
WHERE ZGGH = :yggb;
```

14)itt;goto(9)

15)提交 Oracle 更新事务;

16)关闭 DBF 文件;Close(fp);exit(0)。

参考文献

1 专辑 《客户—服务器技术与应用》北京学苑出版社

1995.3

2 孙宏昌,刘金亭等,Oracle 应用系统开发工具 北京清华大学出版社 1995.6

3 刘友华,传统 MIS 模式向 C/S 环境的转换技术 情报学报 96 增刊

4 孟小峰等,开放数据库互连 ODBC, 北京清华大学出版社 1995.11

油压力波支配着内外针阀的运动,而内针阀的运动又同时控制着外针阀进油通道和喷油器体上的一个“常开式循环小孔”8(起切换油路作用),该孔 8 通过循环油管 9 与油箱 1 相通。

该系统内的燃油流动状态与传统的系统不同,而且针阀启闭响应快。

3 数学模型的建立

3.1 高压燃油总管、支管内流动的数学模型

小管径中的一无流动问题,至今尚未进行完全充分的研究,目前高压管道中的流动仍沿用大管径中的流动理论,大量计算实践表明^[3],计算结果与试验结果尚能很好吻合。在设及摩擦的情况下,根据流体力学基本方程,可推导出:

$$\text{连续方程} \quad \frac{\partial u}{\partial x} + \frac{u}{a^2 \rho} \frac{\partial \rho}{\partial x} + \frac{1}{a} \frac{\partial a}{\partial x} = 0$$

$$\text{运动方程} \quad \frac{\partial u}{\partial t} + u \frac{\partial u}{\partial x} + \frac{1}{\rho} \frac{\partial p}{\partial x} + 2ku = 0$$

式中: u 、 p 、 ρ 、 x 、 t 、 a 和 K 分别为燃油速度、压力、密度、沿管长坐标,时间、音速和流体阻力系数。

用特征线法求解上述方程,耗时较长,一般略去输运项 $u \frac{\partial u}{\partial x}$ 和 $\frac{\partial p}{\partial x}$,经推导可得:

$$\frac{\partial^2 u}{\partial x^2} - \frac{1}{a^2} \frac{\partial^2 u}{\partial t^2} - \frac{2k}{a^2} \frac{\partial u}{\partial t} = 0$$

可用有限差分法求解上述偏微分方程。

3.2 边界条件的建立

燃油系统研究的难点在于边界条件,该新型系统更为复杂。由图 2 可见,高压总管 4 的边界由喷油泵 3 和分配器 5 构成,二根不同长度的支管 6 的边界分别由共同的分配器和两只喷油器 7 构成。

本研究运用流体力学和固体力学基本理论,建立准维模型,加上约束条件,可得到描述喷油泵、分配器和喷油器的十一个微分方程等,其冗长的数学表达式和一百多个的变量详见资料^[4]。

由于方程较多,计算实践表明,所选取的迭代格式和方程的迭代次序将是至关重要的。其次差分间隔对解的误差和收敛性也有重要影响。由实验得知^[5],喷射系统内具有高达 3~4KHZ 的振动分量,为了充分反映这些分量,时间间隔 ΔT 一般应在 15~30 μ s。在本研究中 ΔT 取 15.55~18.07 μ s 模拟计算取得成功。 ΔT 过大或过小,均造成误差过大,或无法收敛,甚至造成计算机“死机”,亦即在高频振动分量的每个周期内,差分节点数应取 10~20 个。一般,低压区的计算精度较差,因此,本研究在低压区取了不同的较高的迭代精度。

4 模拟实例

运用上述模型对 B8.W L60 大功率柴油机基本机型和六种变型机进行了数值模拟,即在功率 P —转速 N 平面上共取 7 个不同的功率点,如图 3。三条曲线为等扭矩曲线,其中 L_1 为基本型功率点, L_2 — L_7 为变型

机功率点,改变近十个结构,进行模拟,优化研究。

基本机型功率点 L_1 的数值模拟和试验结果分析如下

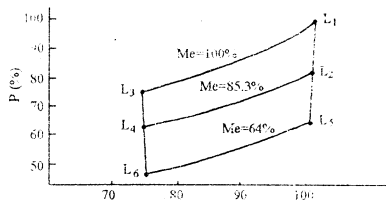


图 3 不同功率点的分布

本研究模拟了泵端、分配器和咀端三个子区域等五点的压力,套筒式吸油滑阀和内、外针阀三个升程 h_1 , h_2 和 h_3 和喷油规律。图 4、图 5 和表 1 为计算机输出结果,可见与常规燃油系统不同,压力波的上升和下降均较陡。

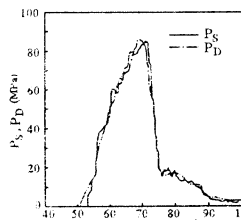


图 4 分配器压力 P_D 和内针阀压力 P_N 数值

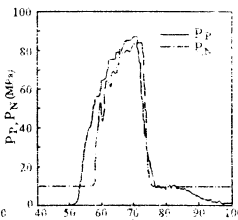


图 5 喷油泵压力 P_P 和喷油器压力 P_N 数值

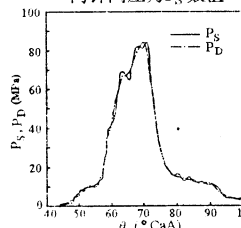


图 6 P_D 、 P_N 实测值

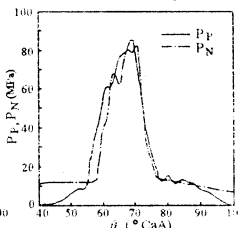


图 7 P_P 、 P_N 实测值

图 6、图 7 为实测曲线,与模拟结果有较为满意的吻合。其有限的误差在于:对于新型结构的喷射系统尚未能进行深入细致的基础性试验研究,许多输入参数,比如分配器中燃油分支所引起的紊流耗散能量,吸油滑阀,双针阀等变截面通道的动态流量系数,以及吸油滑阀上密封环的动态阻尼等未能取自于精确的试验结果,而是分析大量有关常规燃油系统的基础试验和文献^[6]的基础上确定的。

由图 5、图 7 还可知,咀端和泵端压力曲线交叉(常规燃油系统两者大致等距),即咀端压力下降的始点晚于泵端,而终点又早于泵端。根据表 1 的计算机输出结果,可以作出喷射过程品质的判断。

表 1 计算机输出结论

1	喷射始点:55.78CaA 喷射终点:74.34CaA 喷射持续角:18.36CaA
2	无二次喷射
3	内针阀有穴蚀,其他点无穴蚀
4	各点压力峰值不高于最大限制值

从网络管理到服务质量(QoS)管理

刘棣华 施 萍

(吉林工学院计算机系 长春 130012)

摘 要 随着用户对多媒体应用需求的增加,网络负载将急剧增加。如何对网络资源进行综合管理与合理使用是一个急待解决的问题。本文在介绍 OSI 网络管理的基础上重点描述了 QoS(Quality of Service)的定义和管理,分析了网络管理和 QoS 管理之间的关系。最后给出了一个结合网络管理进行 QoS 管理的初步方案。

关键词 网络管理 QoS QoS 管理

1 引言

近几年来,计算机网络的发展异常迅猛,用户对网络性能的要求也越来越高,要保证为广大用户提供令人满意的服务,就需要有一个高效的网络管理系统对网络进行管理。同时,为适应多媒体应用的需要,网络必须对不同的媒体、不同的应用提供不同质量的服务,并进行服务质量(QoS)管理。本文着重讨论网络管理和服务质量管理的关系与相互作用,并给出结合网络管理进行服务质量管理的初步方案。

2 网络管理

网络管理就是要对复杂的网络进行控制,使其达到尽可能高的运行效率。网络管理系统通常指用于改善网络性能、实现网络管理的一组软件。

2.1 网络管理系统的组成

一个网络管理系统应由四部分组成:多个被管代理(Agent)、至少一个网络管理器(Manager)、一种通用的网络管理协议以及一个或多个管理信息库(MIB—Management Information Base)。Agent 驻留在各网络设备上,监测所在网络设备的有关状况。Manager 驻留在网络管理工作站上并通过网络管理协

议和各 Agent 进行通信。MIB 中包含了各 Agent 的若干特征值,Manager 通过查询这些特征值来实现监测功能,通过改变这些值来实现控制功能。

2.2 网络管理系统的功能

OSI 为网络管理制定了一个由两部分构成的框架:其一是层管理(Layer Management),它控制和协调各特定通信层中的被管理资源;其二是系统管理(System Management),它控制和协调属于整个开放通信系统的被管理资源。系统管理从管理范围上讲比层管理要广泛得多,其功能包括五方面:配置管理、差错管理、性能管理、安全管理和记帐管理,这五方面已被普遍地接受为网络管理的五大功能。其中,性能管理和服务质量管理的关系最为密切。

一般地讲,性能管理的含义是确保网络可被访问,避免发生拥挤,使用户可以有效地利用网络访问,避免发生拥挤、使用户可以有效地利用网络资源。也就是说,在性能管理过程中要监视网络设备及其相关的连接,给出利用率和出错率的报告,确保用户对网络设备和连接的需求不超过允许的容量范围,为用户提供可靠的、一致的服务。

性能管理的内容包括:

1)收集当前关于网络设备和连接使用情况的数据

收稿日期:1996年7月10日

5 结束语

(1)本研究建立的模型能较好地模拟与众不同的循环式燃油系统。

(2)为进一步改进计算系统,提高计算精度,数值模拟还有许多方面值得研究。

(3)计算的差分间隔一般取 $15 \sim 30 \mu\text{s}$ 最为恰当,本研究取为 $17 \mu\text{s}$ 左右。

(4)根据实际生产工艺,成本等约束条件的限制,最可行的优化参数为喷油器弹簧预紧长度 L ,喷油孔直径 d 和柱塞直径 d_p 。研究表明, d 和 d_p 的影响较为明显, L 的影响不大。(上海交通大学刘焱棠教授、周校平副教授为本文提出了宝贵建议,深表感谢。)

参考文献

- 1 CADIS—I 研制通讯
- 2 施渭滨,缩短柴油机制造总周期的有效途径,内燃机电脑应用 1995(3)P157—164
- 3 Milan Marcic and Zlalko Kovacic, Computer simulation of the Diesel injection, SAE 851583
- 4 徐兆坤,发动机燃油喷射系统计算机模拟研究和辅助设计,上海高校科技发展基金资助项目(鉴定文件),上海工程技术大学,1995
- 5 松岗信,模田克果,关于柴油机燃油喷射系统的研究,燃料喷射及燃烧译文集。哈尔滨船舶工程学院出版社,1980
- 6 N. B 阿斯达赫夫,米顾颂译,柴油机的供油与燃油雾化,国防工业出版社,1977

2) 对所收集的数据进行分析, 给出关于较高使用情况的趋势

3) 设置使用阈值

4) 用模拟方法确定如何改变网络使其性能最佳

性能管理过程中要收集的数据有: 吞吐率、负载量、传输延迟、等待时间、响应时间和服务质量。其中服务质量主要包括服务的使用情况、可获得性、漏检故障率和失败概率。

3 QoS 管理

3.1 QoS 定义

QoS (Quality of Service) 的最初定义由 CCITT (现称 ITU-T) 给出: “QoS 是一个综合指标, 用于衡量使用一个服务的满意程度”。QoS 的进一步定义可在 RACE (Research into Advanced Communication for Europe) 中找到: “QoS 描述了关于一个服务的某些性能特点。这些性能特点是用户可见的, 它以用户可理解的语言表示为一组参数。这些参数具有客观值或者主观值”。客观值刻画了系统的行为性能 (如失败概率、吞吐量), 主观值刻画了系统的其它服务性能 (如安全性、优先级)。

3.2 OSI 参考模型中的 QoS

在 OSI 参考模型中, 互联系统被构造成分层结构, 各同级层可建立逻辑连接。各层协议的质量可定义为对于该层来说可见的性能参数, 即一组供该层使用的 QoS 参数。因此, 每一层都有一组提供给该层用户的 QoS 参数。

实际上, OSI 参考模型只是为在低速网络的数据传输设计的, OSI 参考模型给出的 QoS 支持仅局限于这类应用的 QoS 需要。它只是静态地定义了支持会话层和传输层的有关参数。为使应用程序能够利用 QoS 机制, OSI 高层 (应用层和表达层) 只是简单地将 QoS 参数直接映射到低层。

3.3 ATM 中的 QoS

ATM 网络技术是 B-ISDN 的核心技术。它可同时支持数据、声音和图像的传输。CCITT 的 “I-系列建议” 为 ATM 层定义了一组相当复杂的 QoS 参数, 它包括三级:

- 1) 呼叫控制级
- 2) 连接级
- 3) 信元控制级

在呼叫控制级的 QoS 包括连接延迟、连接失败率、释放失败率和释放延迟等。连接级和信元控制级的参数包括信元峰值到达率、峰值持续时间、信元平均到达率、峰值与平均到达率之比、信元丢失率、比特出错率、信元插入率等。

3.4 QoS 框架

目前, 对于 QoS 的需求和对于通信协议功能的需求具有同等的重要意义。一些研究机构已在努力通过

定义一组将已存在的 OSI 和 CCITT QoS 服务通用化、形式化的接口, 使现有系统得到扩展, 同时使用系统工程方法制订出一个关于 QoS 说明和管理的全局框架, 希望这个为集成新的 QoS 管理机制而提供的框架能够避免同层之间的功能重复, 进而有效地进行 QoS 管理。

由 ISO 启发的 “服务质量框架新工程项目” 首先为 OSI 范围内的 QoS 规范做出了非常有意义的贡献。该工程的主要目的是定义参考模型和标准术语, 使 OSI 标准能够在 QoS 方面进行扩展。

英国曼彻斯特大学由 UKSERC 基金支持的 “服务质量结构 (QoS-A) 工程” [2] 是另一个试图为分布系统中的 QoS 建议框架进行研究的项目。QoS-A 工程的目的是设计和实现一个分布系统环境中的 QoS 框架, 该环境包括运行于 ATM 网络之上的多媒体应用。

3.5 QoS 管理

QoS 管理是指计算机网络系统按照一定的管理策略对用户应用所需要的 QoS 进行分类统计, 并调用相应的资源分配和调度程序去保证用户得到所需 QoS 的过程。

QoS 管理包括以下几方面:

(1) QoS 说明和映射

QoS 说明按系统处理方式分层定义所需要的 QoS。QoS 映射提供在不同系统层之间 QoS 表达的自动转换功能。

(2) QoS 协商、资源预订和允许控制

QoS 协商指用户和系统以及用户和用户之间就所传输信息的服务质量进行交互, 最后根据应用和系统资源确定系统和用户的 QoS。资源预订负责预定支持给定 QoS 的某些资源。这些资源包括 CPU、缓冲区、传输带宽等。允许控制将 QoS 产生的资源需求和系统中的可用资源进行比较。确定系统是否可满足所希望的 QoS。

(3) 维护和监视

QoS 维护动态地保证各系统部件所要求的性能维持在确定的范围之内。QoS 监视允许系统各层跟踪从低层获得的 QoS 并将之与最初需求相比较。QoS 监视经常采取反馈环的形式, 即在监视过程中从被监视部件上获取 QoS, 将之与需求值进行比较, 然后在必要时对资源进行调整。

(4) QoS 重协商

当系统无法满足某一行为的 QoS 要求时, QoS 重协商对 QoS 说明和协商功能进行调整, 最终实现对该行为的所有相关 QoS 进行全局再配置。

4 网络管理和 QoS 管理的关系

1) 如前所述, ISO 定义的网络管理包括故障管理、计费管理、配置管理、性能管理和安全管理。其中性能

管理通过收集和分析性能参数估价系统资源的运行状况及通信效率等系统性能。这些性能参数也包括 QoS 参数。因此,网络管理自然也包括对某些 QoS 的管理。

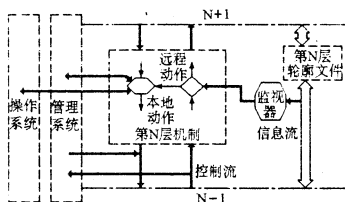
2)从另一方面看,QoS 可分为主观 QoS 和客观 QoS。客观 QoS 与网络性能参数有着非常密切的关系,而主观 QoS 与网络性能并没有直接的关系,它只与用户和提供商的观点有关。因此,QoS 管理是相对于网络应用系统的诸方面而言,而网络管理往往是指维护电连接体中可访问的那部份 QoS 层。

3)不同的组织机构可以对网络管理有不同的看法,因而对同一术语有不同的定义。如将性能管理定义为“性能管理功能包括监视和分析被管网络及所提供服务的性能。为能维持预先制定的性能级别,可以用性能分析的结果激发诊断测试过程和改变原有配置”[9],则 QoS 管理可被视为性能管理的子功能。

4)QoS 管理系统可利用通用的网络管理协议进行通信。网络管理系统中使用的 MIB 和 Manager-Agent 结构也可用于 QoS 管理系统。

5 一个可行的 QoS 管理框架

下图是一个对应于 OSI 参考模型第 N 层的 QoS 管理框架。



此管理框架描绘了 QoS 管理和操作系统及网络管理的关系,给出了对应于 OSI 参考模型第 N 层用于 QoS 管理的逻辑部件及其相互作用。管理框架的功能包括:

1)一个 QoS 说明,它引起与操作系统的协商并产生带有相应协议机制的轮廓文件;

2)一个 QoS 监视器,一旦连接被建立,由 QoS 监视器监视 QoS 参数;

3)一个来自监视器的关于违犯 QoS 的指示,这个指示根据网络管理信息库 MIB 中的参数产生,并引起本地动作去进行再配置和 QoS 重协商;

4)如果重协商和再配置失败,向上一层发出指示;

5)在控制 QoS 的过程中与网络管理系统的相互作用。

6 结束语

在多媒体应和高速网络技术的推动下,QoS 已成为目前国际上网络研究的热点。在我国缺乏巨额资金大量建设高速宽带网的条件下,如何结合网络管理,以 QoS 为目的,在使用资源最少的情况下获得最好的用户满意度将成为一个极为有意义的问题。

参考文献

- 1 U. Black, Network Management Standards: SNMP, CMIP, TMN, MIB, and object libraries, 2nd ed, 1995
- 2 A. Campbell, G. Coulson, and D. Hutchison, A Quality of Service Architecture, ACM Computer Communications Review, April 1994
- 3 M. Fty, A. Richards and A. Seneviratne, Framework for Implementing the Next Generation of Communication Protocols, Proc. 4th Internat Workshop, 1993
- 4 D. Kenney, C. Sluman, P. Pranschke, Management Requirements Arising from a Nato Study of Quality of Service, IFIP TC6/WG66 Symp. on Integrated Network management, 89, pp133-140
- 5 A. Leinwand and K. Fang, Network Management: a practical perspective, Addison Wesley, 1993
- 6 N. Simoni and S. Zraty, Qos: From definition to Management, 4th IFIP Conf. on High Performance Networking, Dec, 1992
- 7 M. Sloman (ed), Network and Distributed Systems Management, Addison-Wesley, 1994
- 8 Wallace, J. Lightsey, Network Management and the Corporate Mission, In: First International Symposium on Integrated Network Management, P. 30-100, 1989, North-Holland.
- 9 Bibliography on Network Management, Computer Communication Reviews, January 1994

康柏 ANSIDA 金牌 西南总代理

成都 ★★ 泛讯

金牌原装商用机

COMPAQ

三年质保

全新奔腾系列

TEL: (028) 5590818

奔腾 5/75 8M. 630M 13500 元
奔腾 5/100 8M. 1.2G 19800 元

FX 家用/教育电脑

1. 286/16	1M内存, 1.44M软驱, 30M硬盘,	单显 1800 元
2. 386SX/33	1M内存, 1.44M软驱, 30M硬盘,	单显 2300 元
3. 386DX/40	1M内存, 1.44M软驱, 30M硬盘,	VGA单显 2500 元
4. 386DX/40	4M内存, 1.44M软驱, 340M硬盘,	彩显 5000 元
5. 486DX/66	4M内存, 1.44M软驱, 630M硬盘,	彩显 5100 元
6. 486DX/80	4M内存, 1.44M软驱, 630M硬盘,	彩显 5200 元
7. 586/100	8M内存, 1.44M软驱, 850M硬盘,	彩显 7700 元
8. 586/120	8M内存, 1.44M软驱, 1.2G硬盘,	彩显 8600 元
• 真16位声卡	300元	• 准16位声卡 200元
• 电影卡(2.0版)	820元	• 四通光驱 550元
• 游戏操作杆	130元	• Canon 喷墨打印机 2100元

我公司所用配件为高品质全新配件、购机五台扣3%、十台扣5%。

购机者一年保换, 三年保修(免费升级, 只收成本费)。

成都市泛讯电子公司

地址: 成都市人民南路三段十五号 邮编: 610041

计算机辅助地学断面管理及绘图系统的设计与实现

郑国政

袁仕章

(沈阳轻型汽车研究所 110035) (沈阳重矿机器股份有限公司 110043)

摘要 本文介绍了在微机上开发的计算机辅助地学断面管理及绘图系统的总体结构和功能,并对系统中关键技术的具体实现作了较为详细的阐述。

关键词 边界处理 图案 线型 参数化绘图

1 引言

全球地学断面计划是国际岩石圈计划中的重要前沿课题,在国际地学组织的倡导下,全球共编制了 170 余条地学断面,我国岩石圈编委会编制了 11 条中国地学大断面。但这些断面图的绘制几乎全部依赖于手工,因而造成绘图工作量大,绘图手段落后,不能满足对断面图随时更新的需要,也不便于查询和修改。

计算机辅助地学断面管理及绘图系统将 CG(计算机辅助绘图, Computer Aided Graphic)和 CAD(Computer Aided Design)技术应用于地学断面图的绘制,采用计算机来对地学断面图及文字资料进行管理,不但从根本上解决了传统的手工成图方式所带来的种种弊端,而且将现有的研究成果深入到一个更新、更高的层次上来;不但为地学断面在计算机内成图提供了最直接有力的工具,而且直接对地学断面的机内资源进行科学规范的管理;从而显著地提高了地学断面的绘图质量和效率,更便于查询和修改。实践证明,该系统功能全面、灵活实用、简单易行、通用性强、新颖可靠,该系统不但能够方便地进行地学断面图的绘制,图形和文字资料的输入,而且能够对地学断面进行全面系统的管理(主要是查询和修改)。它的开发研制在地学断面管理和绘制方面具有开创性和普及推广价值。

2 系统的总体结构和功能

系统的硬件环境:486 型微机,内存 4MB;TVGA 彩色图形显示器,HUSTON TG1100 型数字化仪;

系统的软件环境:MS DOS6.22, AUTO-CAD10.0, AUTOLISP 二次开发语言,UCDOS 5.0

系统由查询、修改、绘制和辅助工具等四个子系统组成。

2.1 查询子系统

在查询子系统中,用户可以进行多层次的查询操作,也可以利用本系统所提供的多种查询方法对有关信息进行查询操作。如果将全球的 170 多条地学断面全部或部分输入到计算机中,在本系统中可以按照地学断面

列表、按地区、按国家、按索引图等方法来查找指定的地学断面,这一过程在本系统中称为外层查询;在查找到指定的地学断面后,用户可以进行内层查询,也即对有关这条地学断面的信息进行查询操作,如查询地质条带、图例、说明文件、地质构造单元等;在查找到指定的图后,可以对图上的几何信息、地理信息、实体信息进行查询操作,如查询任一点在图上的位置、该点所处的经济和纬度,任意二点在图上的距离、实际距离及经纬度差,任意区域的面积和周长;对有关的地学信息可以进行鹰眼加速目标搜索,它能在一张很大的图中迅速搜索到用户所需要的目标,并将散布在图中的目标的分布情况以数次亮显的方式展现在用户眼前,从而极大地方便了查询,提高了看图速度,使查询更具有目的性和效率。被搜索的目标可以是等值线、地层代号、河流、湖泊、海洋、地名、地质构造单元、岩性符号、断裂、断层、大地电磁测深点、震级、断面线等。

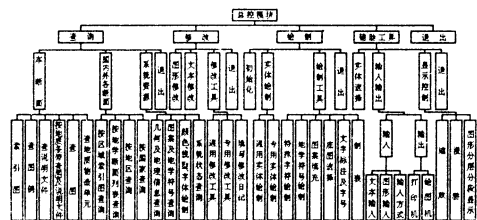


图 1 系统结构

为便于用户修改和绘制方便,用户还可以对系统状态、修改档案、图纸信息及系统资源等信息进行查询,其中修改档案查询中,用户可以要求显示历次修改档案、前一次的修改日记,最近的修改日记;系统资源可以根据用户需要来帮助用户全面了解系统所有的线型库,阴影图案库、图形符号库、特殊字符库、字体库、字型库、颜色、层的信息等。

2.2 修改子系统

修改子系统为用户提供了多种图形修改工具,如改换填充区域的颜色、修改填充图案、修改图形符号、修改文字等。在修改工作完成之后,用户可以填写修改日记,系统将根据用户所填写的内容,自动对其进行分类和归档。

2.3 绘制子系统

在绘制子系统中,用户可以绘制色标列表、图例边框、国界线、高导层、上地幔高导层、主要断面线、次要断面线、热流测点、钻孔、主要深断裂、坐标尺、表格、震级标注等地学图件。系统还提供了地层代号及地质年代、物理单位、罗马数字等 127 个特殊字符,地学符号 47 个及各种文字标注字号以供用户在进行地质断面图绘制过程中选用。

为便于使用本系统进行制图,系统提供了含有丰富地理要素,如海岸线、国界线、省界、河流、城市、湖泊等的地理底图,用户可以根据需要选取所需的地理底图和地理要素。

这样,根据勘探、航测及分析处理所得的数据,在本系统所提供的交互环境下,就可以方便地绘制出地质断面图。

2.4 辅助工具子系统

辅助工具子系统提供了多种辅助性工具。在显示控制方面,提供了清屏、低速图纸漫游和高速图纸漫游,从而实现了地图上八个方向不同步长的漫游,极大地方便了用户看图;实现了定位显示地图,从断面上选取某一地名或地质构造单元,随即全屏显示以该地名或地质构造单元为中心的地带的地质指标;也可以由不同的图案与颜色来表示各地理单元的构造和地质指标,可以根据用户的要求显示地理底图、清绘图、图案填充图及着色图。在输入输出方面,用户可以进行文本和图形的输入输出,为便于与系统外的资源进行交流,系统还提供了与 DOS、AUTOCAD、C、PASCAL、FORTRAN 等高级语言的接口,用户可以随时进入到所熟悉的语言环境中进行编辑和运行。在实体选择方面,为用户提供了 26 种实体选择方式,大大地方便了用户操作。

3 系统中关键技术的实现

3.1 三种不理想边界的处理及对由其所构成的区域进行填充的实现

着色与图案填充是地质断面图的特色。在着色和图案填充时,如果直接选取理想边界作为填充区域的边界,就可以获得满意的填充效果。所谓理想边界即构成填充区域边界的实体刚好在它们的端点处相交,并构成闭合区域;与理想边界相对应的是不理想边界,即有的边界线超过了相交点或构成的区域为非闭合区域。如果直接选取不理想边界作为填充区域的边界,就将产生错误的填充,也就不能获得满意的填充效果。为了解决不理想边界所构成区域的填充问题,本系统将不理想边界大致分为二类,并分别给予处理。第一类是非理想闭合边界,尽管这类边界构成了闭合区域,但有的边界线超过了相交点,本系统采用切边恢复法对这类边界进行处理,并对由其所构成区域进行填充。所谓切边恢复法即首先将端点超过相交点的边界在相交点

处将闭合区域外的多余边界切掉,将不理想边界转化为理想边界,然后对该区域进行填充,随后恢复为原始边界。第二类是非闭合边界,这类边界又分为二种,第一种是简单非闭合边界,即构成填充区域的边界是非闭合的开口边界,对于这种边界本系统采用加边恢复法对其进行处理,并对由其所构成的区域进行填充,所谓加边恢复法即首先将边界封闭,随后对其所构成的区域进行填充,最后再将边界恢复为原来的样子。第二种是复杂非闭合边界,即构成填充区域的边界不但具有非闭合的开口边界,而且有的边界线超过了相交点,对于这类边界,本系统综合采用前述二种方法进行处理

3.2 阴影图案和线型自动生成

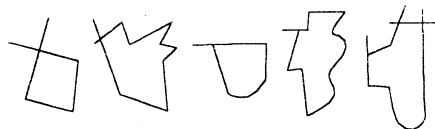


图2 非理想闭合边界



图3 采用切边恢复法对非理想闭合区域进行处理及对其所构成区域进行填充的程序框图

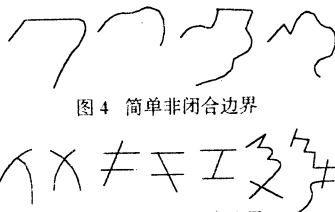


图4 简单非闭合边界

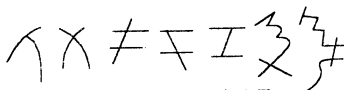


图5 复杂非闭合边界

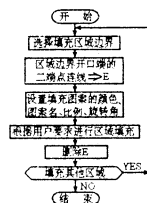


图6 采用加边闭合恢复法对简单非闭合边界进行处理及对其所构成区域进行填充的程序框图

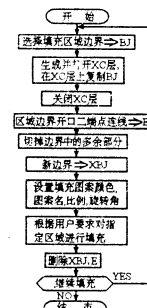


图7 采用切边恢复法和加边闭合恢复法对复杂非闭合区域进行处理及对其所构成区域进行填充的程序框图

一种多维信息的数据压缩技术

周纪林

(华中理工大学汉口分校 武汉 430012)

摘要 一个信息的全面特性描述有赖于它的一组属性,一个物体或过程包含着系列全部或部分值相等的这样一组属性。可以采用固定码本对这些数据进行压缩,但更为有效的是采用动态码本。动态码本压缩技术特点适用于由外部事件引起的采样数据的压缩,笔者成功地将这一技术用于汽车运行状态记录仪中。

关键词 固定码本 动态码本 压缩技术

1 信息的多维性研究

自然界或科学技术中的某一点的状态可以用它的

收稿日期:1996年6月28日

若干属性值来定义,所用属性种类越多,该点就描述得越全面,每个属性值种类范围越大,该点就被描述得越精确,我们用以下形式定义A类信息:

$$I_A(a_n[L_n, U_n], a_{n-1}[L_{n-1}, U_{n-1}], \dots, a_j[L_j, U_j]),$$

为便于在使用本系统过程中自行定义和添加阴影图案和线型,系统采用交互参数化方法编制了阴影图案和线型自动生成程序,只要给出由点和线所组成的样本,程序自动对样本的实体数据信息进行处理之后转化为阴影图案数据信息或线型数据存入图案库和线型库以供需要时调用。另外,本系统还开发了80种图案和线型,以满足阴影图案填充和绘制不同线型的曲线时使用。

3.3 特殊线型曲线的绘制

标准线型库中所定义的线型都是由共线的点和线所组成的,对于地学断面图中的一些具有特殊线型的曲线,标准线型库中所定义的线型就无能为力了,为了解决地学断面图绘制过程中,各种具有特殊线型曲线的绘制问题,系统将特殊线型的曲线分解为基线和与基线成一定角度的小曲线段,并采用分步绘制的方法,编写了特殊线型曲线的绘制程序。所谓分步绘制即首先利用系统标准线型库中所定义的基线线型来绘制基线,然后绘制与基线成一定角度的小曲线段。

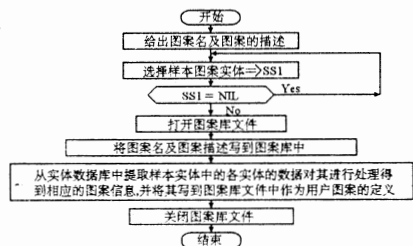


图8 阴影图案自动生成程序框图

3.4 图形输入的实现

地学断面图非常复杂,图形输入工作量大,要求高,因此在现有地学断面图的输入过程中,采用了交互式输入、参数编程法、交互参数化、扫描输入、数字化仪输入等五种现有的图形输入方法。

3.5 参数化绘图

由于参数化绘图可通过给定参数不同的数值而得到不同的集合模型,非常适合于地学断面图这类系列化图件的绘制,因此本系统的程序设计过程中始终贯穿着参数化设计的思想。对于复杂的地学断面图,本系统采用参数图形嵌套的方法,即通过调用其他参数图形来实现图形不同层次的参数化,以获得满足需要的参数化过程。

4 结论

系统的特点:

(1)功能灵活、模式通用:尽管本系统是以东乌珠穆沁旗至东沟地学断面为主开发的,只要对本系统稍加扩充,即可对国内及全球的其他地学断面进行绘制和管理。

(2)人机界面友好:本系统的人机界面集软件封面制作、计算机动画技术及音响效果于一体,并采用对用户操作具有良好的引导作用的屏幕菜单、图标菜单和下拉菜单。3、良好的图形存储方式:由于本系统采用动态图形存储方式,从而节省了大量的磁盘空间。

本系统也适用于区域地质填图、震中分布图、矿产资源分布图等图样的绘制和管理,系统中所采用的解决办法和实现过程也适用于其他行业的CAD及计算机可视化管理,如美术设计、广告创意、服装设计、建筑装饰、交通监测。

参考文献

- 1 Autocad 10.0 用户参考手册. 海洋出版社, 1989.
- 2 张海藩. 软件工程导论. 北京:清华大学出版社, 1994.
- 3 卢建勋, 夏怀宽. 内蒙古东乌珠穆沁旗至辽宁东沟地学断面(1:1000000)及说明书. 地震出版社, 1992.
- 4 J. W. H. Monger. Guidelines For The Global Geoscience Transect Projects.

..., $a_1[L_1, U_1]$)

其中 a_j 是 I_A 的一类属性, 其取值种类范围为 $[L_j, U_j]$, 即至少 L_j 种, 最多可有 U_j 个; 它决定了被描述信息的精度。

由此, 某一“点 i ”的信息可表示为:

$$I_{Ai}(a_n[i_n], a_{n-1}[i_{n-1}], \dots, a_j[i_j], \dots, a_1[i_1])$$

其中 i_j 是属性 a_j 的一个值。

在连续或离散采样时, 某一维属性值可能和已采集的同维属性值相同, 也可连续相同。设有不同时刻的同类信息 I_r, I_k 则它们的相似度可定义为:

$$\alpha = \frac{I_r}{I_k} = \sum_{j=1}^n \frac{I_r(a_j)}{I_k(a_j)}$$

其中 $I_r(a_j), I_k(a_j)$ 分别表示信息 I_r, I_k 的第 j 维的属性值。

再定义:

$$\frac{I_r(a_j)}{I_k(a_j)} = 1 \quad \text{当 } I_r, I_k \text{ 第 } j \text{ 维属性值相等}$$

$$\frac{I_r(a_j)}{I_k(a_j)} = 0 \quad \text{当 } I_r, I_k \text{ 第 } j \text{ 维属性值不等}$$

可见 $0 \leq \alpha \leq n$

设 I_c 为标准信息, 则称:

$$\alpha_{cj} = \frac{I_r(a_j)}{I_c(a_j)}$$

为测量分量相似度, 而平均相似度则为:

$$\alpha_s = \frac{I_r}{I_c} / n = \alpha / n$$

平均相似度可以作为信息的冗余度表征, α_s 越大, 则数据的可压缩潜力也越大。

信息 I_r 中的 $I_r(a_j), I_r(a_{j+1})$ 可能存在相关性, 比如 $I_r(a_j)$ 表示时间的分, $I_r(a_{j+1})$ 表示秒, 则取值在 1—59 时, $I_r(a_j)$ 的值不变, 当 $I_r(a_{j+1})$ 取值为 0 时, $I_r(a_j)$ 值将加 1。同理, $I_r(a_j)$ 和 $I_{r+1}(a_j)$ 也存在相关性, 如 $I_r(a_j)$ 为 30 分, 若采样周期为 1 分钟, 则 $I_{r+1}(a_j)$ 必为 31 分。利用这种相关性, 可以通过预测编码的方法进行数据压缩, 但这种方法要求采样周期是固定的, 输入数据是平稳随机过程所得, 或由自适应预测所得。如果取样由外部随机条件触发, 则这种相关性的统计规律就难确定, 所以也不能采用哈夫曼(Huffman)编码方法。另外, 对于实时时间(包括年)这一类信源, 其样本上限为 ∞ , 采样周期可能从 1 秒到几十天, 因此, 一般的编码方法很难胜任。

2 固定码本及数据压缩算法

如上面定义的信息 I_A , 由于 $I_r(a_j)$ 的概率极难确定, 因此设计一种有冗余的码本:

$$y = \{y_i | i = 1, 2, \dots, N\}$$

$$\text{其中 } y_i = \{a_n[i_n], a_{n-1}[i_{n-1}], \dots, a_j[i_j], \dots, a_1[i_1]\}$$

令 I_A 的每一个属性值 $V_{aj}(j=1, 2, \dots, n)$ 可用一个字节表示, 并且已知, 设码本从 $P+1$ 单元起存放, 则码本结构如图 1

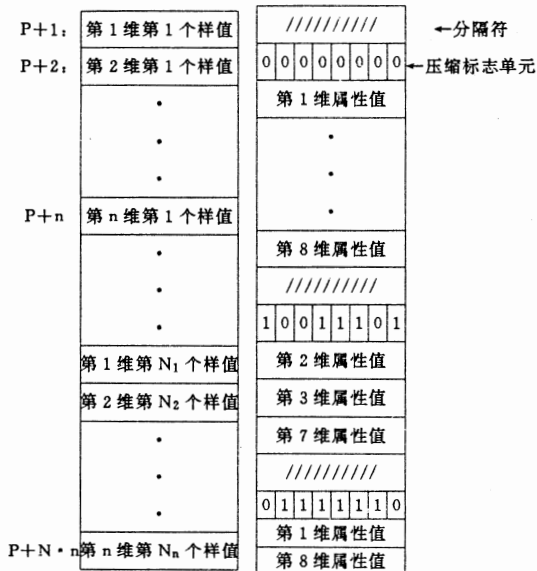


图 1 固定码本结构 图 2 动态码本和压缩编码区

图中 $N_i (L_i \leq N_i \leq U_i, i=1, 2, \dots, n)$ 是 I_A 各维的取值个数。则码本个数为:

$$N = N_1 \cdot N_2 \cdot \dots \cdot N_n = \prod_{i=1}^n N_i$$

因为一个码本由 n 个字节构成, 所以码本共占有的字节数为:

$$N_s = N \cdot n$$

若 I_A 的一个值 V_{aj} , 必须用 1 个字节表示, 则码本有字节:

$$M = N_s \cdot 1 = N \cdot n \cdot 1$$

固定码本的各样本值必须已知, 若输入序列中有码本没有的值, 则必须修改码本, 或者采用模糊量化方法, 这就必然产生误差, 这种误差在一定范围内对有些信息是允许的, 但对诸如时间一类的信息是不允许的。

设采样数据为 $X: X_1, X_2, \dots, X_i, \dots$, 对 X 划分:

$$\emptyset X = \{R_i | i = 1, 2, \dots, N\} \text{ 且 } \bigcup_{i=1}^N R_i = X.$$

有映射 $f: X \rightarrow Y$, 使 $y_i = f\{x | x \in R_i\}$

则利用固定码本进行数据压缩的算法如下:

L_2 : 采样一个信息 I_r ;

For $i=1$ TO N DO

Begin $b:=0$; For $j=1$ TO n DO

Begin If $a_{cj}=0$ Then go to $L1$ Else

Begin $b:=b+1$; If $b=n$ Then

Begin $R[g] := \text{add}(y_i)$; $g:=g+1$; go to $L2$

End

End

End $\{j\}$

$L1$: End

其中 j 为信息中属性序号, i 为码本区序号, $R[g]$ 为压缩后的编码记录区, g 为其指针。

利用固定码本和以上压缩算法形成的编码形式简

单,解码时只要按所记录的码本信息地址取出对应的信息属性值即可。它的缺点是码本区长,编码速度慢,所采样的信息属性值必须完全已知,但有些码本样值可能“终生”无用。

关于压缩比的讨论如下:设信息中每个属性值长度为 1 个字节,则码本有 $M = N \cdot n \cdot 1$ 个字节,那么存放一个码本地址所需的字节数为 $\text{INT}(\log_2 M/8) + 1$ (INT 表示取整,当 $\log_2 M/8$ 余为 0 时不加 1)。压缩时仅记录每个码本的地址 $\text{add}(y_i)$,若不压缩,则存放一个相应数据内容必须有 $n \cdot 1$ 个字节,则压缩前/后之比为 $n \cdot 1 / (\text{INT}(\log_2 M/8) + 1)$ 。例如某类信息有 4 种属性,它们的取值种类分别为 2、4、2、4,每个属性值可用一个字节表示,则码本有 64 种值的组合方式,共占 256 个字节,经压缩,只需用一个字节表示码本地址,否则要用 4 个字节存放一个信息。

3 动态码本及压缩技术

鉴于以上讨论的信息的特殊性以及用固定码本进行压缩的缺点,采用码本和记录区合二而一的动态压缩技术是解决以上问题的较好办法。

信息采集、编码开始时,在存储区首先置一个分隔符,其下再设一个压缩标志单元,设均用一个字节表示。压缩标志单元的第 j bit 表示当前采样信息的第 j 维属性值是否被压缩。假设 $\text{bit}_j = 1$ 表示被压缩,也即当前信息的第 j 维属性值与紧上一个信息的第 j 维属性值相同 $\text{bit}_j = 0$ 表示当前第 j 维属性值未被压缩,则在标志单元下依次记录该值(设可用一个字节表示),图 2 是动态码本及压缩后编码区的例(信息有 8 个属性)。

记当前是对第 i 个信息进行压缩,标志单元为 L_i ,从左向右逐位被处理, m_k 表示在第 k 个信息中没有被压缩的属性值的个数, $\text{add}(L_k)$ 表示 L_k 单元的地址,则压缩算法如下:

```

For k=1 TO i Do  $m_k = 0$ ;
For j=1 TO n DO
Begin For k=i-1 TO 1 Step -1 DO
Begin If  $L_k(\text{bit}_j) = 1$  Then GO TO  $l_2$  Else
Begin If  $m_k = 0$  Then  $m_k = 1$  Else  $m_k = m_k + 1$ ;  $g = \text{add}(L_k) + m_k$ ;
If  $l_i(a_j) = R[g]$  Then Begin  $L_i(\text{bit}_j) = 1$ ; GOTO 12 End;
Else
Begin  $L_i(\text{bit}_j) = 0$ ;
 $g = \text{add}(L_i) + m_k$ ;
 $R[g] = l_i(a_j)$ 。
End
End
End
12; End

```

解码过程是:设当前读第 i 个记录,找到其分隔单元,设为 d_i 标志单元 L_i 的地址为 $\text{add}(L_i) = \text{add}(d_i) +$

1,从左至右读 L_i 各 bit,若 $\text{bit}_j = 1$,则 $i-1 \rightarrow i$,向后找倒数第 2 个压缩记录,直到 $\text{bit}_j = 0$,读取该记录所记第 j 维属性值,作为当前记录第 j 维属性值。若当前记录第 j 维标志 $\text{bit}_j = 0$,则直接读取本记录所记第 j 维值。

以上方法的优点是码本值不必预先知道,每个码本值均有效,压缩比大。缺点是编、解码算法较复杂,速度较慢。

关于压缩比的讨论如下:设压缩前某一记录各维值共占 n 个字节(设每维值只需一个字节),当采样的信息各维值等于紧上一记录值时,该记录占有 $\text{INT}(n/8) + 1 + 1$ 个字节(当 $n/8$ 的余数为 0 时仅加 1 个 '1'),其中有 1 个 '1' 是分隔符。此时的压缩比为 $n / (\text{INT}(n/8) + 2)$ 。若当前采样各维值均不等于紧上记录值时,则必须全部记录,此时占有 $n + \text{INT}(n/8) + 2$ 个字节,其余情况介于这两种情况之间。

动态码本压缩过程中,某维属性在连续相等后出现一次不相等后又出现连续相等,按以上算法是将最近一次的值作码本,而事实上在向后不远处早已存在一个同样的码本,这就出现了冗余。若想消去这种冗余,势必增加时间和空间的开销。

4 一个实例

在一个汽车驾驶状况记录、报警系统中,被采录的数据有时间,加速度,即时车速,刹车使用情况,左、右、前、后运动状态,各种灯光的使用状况等。如果用固定周期进行采样,则有限的存储空间很快就被用完,同时可能漏掉重要的信息。因此采用了外部条件触发随机采样、多维向量形式进行数据压缩存储。特别是时间信息,不可能安全确定其定义域,但也存在着大量的冗余,例如 1 月 23 日 14 时 50 分 20 秒,当在 14 时 50 分后需每 5 秒采样记录一次数据时,则 1 月 23 日 14 时等都有冗余。该系统的采样周期也很难预测,因此采用了前面叙述的动态码本数据压缩方法记录实时数据,结果可供实验,事故分析处理等使用。

参考文献

- 1 Multimedia Interface Design. Edited by Meera M. Blatner, Roger B. Dannenberg, Acm press Addison-wesley Publishing company.
- 2 JUSTIN ZOBEL, ALISTAIR MOFFAT. Adding compression to a full-text Retrieval System. Software-practice and Experience. vol. 25(8), 891—903 (AUGUST 1995).
- 3 姚庆栋等编. 图像编码基础. 浙江大学出版社
- 4 林福宗、陆达编. 多媒体与 CD-ROM. 清华大学出版社
- 5 高文著. 多媒体数据压缩技术. 电子工业出版社
- 6 LIU XIAOFAN, GUO CHUNBIAO, The Method of Equal Length code for Compressing Two Level Image. Computer Applications and software NO. 3. 1991
- 7 TAN QIANGMIN. Data Compression with HASH codes, Computer Application and Software NO. 3. 1990

Sybase/Powerbuilder 环境中的用户管理

吴相林 张红涛 陈铁英

(华中理工大学自控系 武汉 430074)

摘要 本文介绍了 Sybase/Powerbuilder 环境下 Unix 用户、Sybase 用户和 database 用户等三种不同的用户概念,同时阐述了该环境下的 client 端利用 Powerbuilder 编程实现各种用户管理功能的途径与方法。

关键词 client/server 模式 Sybase/Powerbuilder 环境 用户管理

1 引言

进入 90 年代以来,一种新型的计算机网络构造方法——客户/服务器结构(C/S 结构)得到迅速发展。C/S 结构是指一个复杂的计算机应用任务合理地分解为多个子任务,由多个服务器和客户机分别承担。客户机和服务器之间体现为“服务请求/服务响应”关系,即用数据库服务器完成数据处理功能,而客户机完成应用事务的组织和人机界面的实现。这样既合理有效地利用了客户机和服务器的资源,又大大减少了网络的通信负担,发挥了分布式系统的各种优点,改善了系统运行的总体性能。

C/S 模式所构造的计算机应用系统是有多个用户参与的复杂的网络应用系统,其用户管理的内容包括用户名称管理、用户口令管理、用户分组管理、用户权限管理和用户进程管理等诸多方面,这些是各种 C/S 模式的应用系统都必须认真分析、全面设计的共性问题。本文针对上述问题,探讨了在 client 端借助 Powerbuilder 的友好人机界面实现各种用户管理功能的途径与方法,将应用系统的其他功能和用户管理功能融合成为一体,改善了多用户管理与实际应用系统分离而给系统管理人员带来的诸多不便。

1 Sybase/Powerbuilder 环境的特点

Sybase/Powerbuilder 环境(S/P 环境)是近年来十分流行的典型 C/S 体系结构软件环境。它把 Sybase 作为服务器(server)端的数据库软件产品,Powerbuilder 作为客户机(client)端的开发工具,体现了数据处理速度快、人机界面友好、开发移植效率高等优点,

受到许多应用系统研究和开发者的青睐。其基本结构如图 1 所示。

Sybase 是基于 C/S 体系结构的数据库软件产品,它由三个主要部分组成:一是进行数据库管理和维护的一个联机(on-line)关系数据库管理系统 Sybase SQL Server;二是支持数据库应用系统的建立与开发的一组前端工具软件 Sybase SQL Tools;三是可把异构环境下其它厂商的应用软件和任何类型的数据连接在一起的接口 Sybase Open Client/Open Server。SQL Server 是个可编程的数据库管理系统,它是整个 Sybase 产品的核心软件,起着数据管理、高速缓冲区管理、事务管理的作用。为此 SQL Server 提供一些重要的管理和控制功能:采用单进程、多线程技术和基于成本优化的高性能存取控制,由 SQL Server 实现的安全性和完整性控制,多用户多任务环境下的并发控制和事物完整性控制,系统故障情况下的恢复和对系统运行的监控及系统性能的调优,支持分布式的查询和更新等。

Powerbuilder 是一种完全按照 C/S 体系结构充分体现面向对象设计思想的数据库前端开发工具,通过对不同数据库采用不同接口的形式,可同时支持多种关系数据库,它具有以下良好的功能和特点:

●Powerbuilder 是 Windows 环境下客户端的开发工具。它开发出的应用程序具有 Windows 风格,可以和服务器端的数据库协同工作,构成 C/S 工作方式,顺应了计算机技术的发展潮流。

●支持多种关系数据库管理系统。如:ORACLE、SYBASE、INFORMIX、DB2、SQLBASE 等。还可读取如 dBASE 格式的文件充分利用来自不同数据源的数据。

●最具特色的 DataWindow 对象使用可以方便地操作关系数据库的数据,无需直接使用 SQL 语句,用户就可以查询、修改、插入和删除数据,可以以多种表现形式(文字,图形,多媒体等)来构造人——机界面。

●Powerbuilder 使用了面向对象的方式,这能使系统开发人员在无需精通专用语言的情况下迅速转向面向

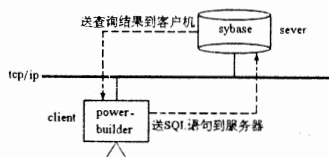


图 1 S/P 环境基本结构

对象的开发。Powrebuilder 应用是由一系列对象组成的,包括有窗口、菜单、函数、数据窗口和各种控制对象,需要创建的对象定义完毕,系统的开发也就完成。

Powerbuilder 正以其优异的性能逐步成为 C/S 应用开发工具的标准,它使应用的开发速度更快,开发成本更低,开发出的质量更高,完成的功能更强,可移植性更好。

2 三种不同的用户概念(Unix、Sybase、Database)

S/P 环境中三种必须严格区别的用户概念,即 Unix 用户、Sybase 用户和 Database 用户。三种用户间既有本质的区别,又有密切的联系,在讨论用户管理设计之前先将这些用户概念一一加以介绍。

●Unix 用户是登录到 Unix 操作系统的用户。数据库管理系统(DBMS)与所处的操作系统密切相关,DBMS 在运行时要依赖与操作所提供的功能,操作系统会影响 BDMS 产品所提供的功能以及这些功能的实现方式。Unix 操作系统是 Sybase 经常使用的一种多用户环境,它以分时方式让多个用户分享计算机资源,实际上 Sybase 就是 Unix 的一个用户。如果希望以终端方式(或仿真终端方式)登录到 Unix 上去,然后用 isql 与 Sybase 交互,则必须是 Unix 的某一合法用户;如果客户机只想登录到数据库服务器 Sybase 上去,则只须是 Sybase 的合法用户,并不要求一定是 Unix 的用户。Unix 具有严密的用户管理系统,对 Unix 用户的管理应当符合 Unix 的规范,并由 Unix 的超级用户 root 来完成。

●Sybase 用户是登录到 Sybase 数据库服务器上的用户。基于 Unix 的 Sybase 数据库管理系统,为自己建立了特殊的网络接口,它从网络上接收客户端的数据库访问要求,根据要求访问数据库和进行必要的数据处理,然后把结果送回到客户端。Sybase 在 Unix 中表现为一个进程,但它利用多线程技术提供了多用户的功能。虽然 Sybase 它利用了 Unix 的设施,但自身提供了一些重要的管理和控制,它的基本数据存储方法不受 Unix 文件系统或原始磁盘的影响,它执行自己的存储管理和缓冲方案。Sybase 有一套独立于 Unix 的用户管理系统,通常情况下由 Sybase 的系统管理员 sa 来执行用户管理的功能。每一个 Sybase 用户有自己的注册名和口令,用户可以不向 Unix 注册而直接登录到 Sybase 数据库服务器上去。

●Database 用户是可以存取某一 Database 的用户。Sybase SQL Server 在安装时就有三个系统数据库:master 数据库、model 数据库和 tempdb 临时数据库,随后还可建立一些用 Create Database 命令创建的用户数据库。每个 Database 都有自己的用户集合,记录在 sysusers 系统表中,Database 的创建者通常是该 Database 的所有者和第一个用户,其他用户由 Database 的所有者使用 sp __ adduser 系统过程将容

许使用该数据库的 Sybase 用户添加进去。显然,所有 Database 的用户一定是 Sybase 的合法用户,但并不是所有 Sybase 用户都是某个 Database 用户。特别应当注意的是,虽然每个 Database 用户一定是 Sybase 的用户,但这两个用户的名称可以是不一样的。换句话说,一个 Sybase 用户可以是多个 Database 的用户,而且这些用户名称可以不一样。

3 用户管理的内容

S/P 环境中的用户管理包含很多内容,由于本文重点讨论 C/S 结构中的用户管理问题,故下面仅就 Sybase 用户和 Database 用户的管理内容分类加以叙述,而不涉及 Unix 用户的管理内容。

●Sybase 用户及其口令管理

为了确保系统的安全性,防止不合法使用,以免数据泄露、更改和破坏,可利用 Sybase 设定各类用户及其口令。用户无论用哪种方式与 Sybase 对话都必须首先回答正确的用户名和口令才能注册成功。系统过程 sp __ addlogin 增加新的 Sybase 用户到 SQL Server,但它不授予用户存取任何用户数据库的权限;系统过程 sp __ droplogin 从 SQL Server 中删除 Sybase 用户;系统过程 sp __ password 用来改变 Sybase 用户的口令。显然,只有 Sybase 的系统管理员 sa 才能调用上述三个系统过程。

●Database 用户的添加与删除

系统过程 sp __ adduser 使 Sybase 用户成为 Database 用户;系统过程 sp __ dropuser 从 Database 用户集合中删除某个用户。Database 的所有者和系统管理员 sa 都可以调用这两个系统过程。

●Database 用户分组管理

当应用系统中的数据库用户比较多时,应当对这些用户进行分组管理。执行不同功能的用户分属不同的组,每个用户只能属于一个组。“组”提供了一种简便的一次可对多个用户授权和收回授权的方法。数据库的所有者可以根据存取该数据库的用户的不同属性定义一些组名,组可以用系统过程 sp __ addgroup 在什么时候创建,但通常应当对数据库的组名统一规划。用户可被系统过程 sp __ change group 设置到(或重新设置到)某个组,用户的组可以在工作性质改变时及时加以改变。

●用户权限的管理

越来越多的管理信息系统以网络为开发背景,它们面向多个用户,以菜单形式向用户提供各种功能,但并不是每一个 Database 用户都能随意存取该 Database 中的所有对象,需要采用用户权限管理来规定其存取权限。SQL Server 的管理系统通过 SQL 的 grant 和 revoke 这两条命令给用户和组授予和收回各种各样的权限,它们可指定哪个用户或组可执行哪些命令,以及对哪个表、视图或列进行哪些操作。对组设

定存取权限,可以省去逐个对用户设定权限的繁琐工作和避免同组用户权限不一样的错误,同时,如果用户因工作岗位变化,只要改变该用户的属组,其存取权限就自动跟着改变。

●用户进程和登录管理

登录到 Sybase 的 SQL server 上的用户必然拥有一个用户进程(DBMS 管理的进程),系统管理员可以在需要的时候用 sp_who 系统过程监视和观察系统当前有哪些用户进程,还可以必要时用 kill 命令“杀死”某些用户进程。用户每次进入和退出应用系统的时间也应当记录保存,以便必要时备查。

4 客户机上运行的用户管理系统

下面介绍的用户管理系统是在客户机上用 Powerbuilder 实现的,它为系统管理员提供了良好的 windows 风格的用户界面,避免了管理者面对服务器环境中单调、乏味的命令提示符和难于记忆的键盘命令。下面就该用户管理系统设计方案的主要内容介绍如下,分成表结构和数据窗口、用户管理 window 和数据窗口控制、菜单和 script 等三个部分介绍。

●表结构和数据窗口

应用系统中,用户数据库名为“CWGL”,该数据库中用于“用户管理”的表有三个:用户列表(YHLB)、对象权限表(DXQXLB)和用户登录表(YHDL)。YHLB 的数据结构见表一,为 YHLB 建立的数据窗口 dw_yhlb 样式见表二,其中为了口令的保密性,在 dw_yhlb 数据窗口中,用户口令这一列的 Edit style 为 password 形式。表三是 YHDL 的数据结构,表四是数据窗口 dw_yhdl 的样式,其中“登录时间”和“退出时间”都是由 powerbuilder 中 today()和 now()两个函数转化成字符串后相加而成。

字段名	类型	宽度	字段含义
yhbh	char	4	用户编号
yhxm	char	8	用户姓名
yhzh	char	6	用户组号
yhkl	char	10	用户口令

表一 YHLB 的数据结构

用户编号	用户姓名	用户组号	用户口令
zjg	张建国	G1	*****
lsl	李胜利	G2	*****

表二 数据窗口 dw_yhlb 的样式

字段名	类型	宽度	字段含义
yhbh	char	8	用户编号
yhxm	char	8	用户姓名
dlrqsj	char	22	用户登录时间
trqsj	char	22	用户退出时间

表三 YHDL 的数据结构

用户编号	用户姓名	登录时间	退出时间
------	------	------	------

zjg	张建国	96.05.20.14:21:50	96.05.20.17:56:32
lsl	李胜利	96.05.20.14:25:06	96.05.20.18:01:20
lsl	李胜利	96.05.21.08:05:50	96.05.21.11:58:04

表四 数据窗口 dw_yhdl 的样式

字段名	类型	宽度	字段含义
dxmc	char	8	对象名称
qxmc	char	8	权限名称
G0	char	1	G0 组是否有权
G1		1	G1 组是否有权
...
G9	char	1	G9 组是否有权

表五 DXQXLB 的数据结构

对象名称	权限名称	0 组	1 组	...	9 组
dmdw	select	+	+		+
jfxm	insert	+	+		
kmye	update	+	-		-

表六 数据窗口 dw_dxqqlb 的样式

表五和表六分别是 DXQXLB 的数据结构和相应的数据窗口 dw_dxqqlb 的样式。这里用户组共分 10 个,每一个用户分别处于其中一组(见表二)。如表六所示,若表中一行记录中的某个组为“+”,则表示给该组授予相应对象的相应权限,如果为“-”,则表示收回该组对相应对象的相应权限。

●用户管理 window 和数据窗口控制

用户管理窗口 w_yhgl 应当是整个应用系统的一个子窗口,见图二。其中的数据窗口控制 dw_yhgl 为用函数 libraryexport()动态生成各种数据窗口所公用,窗口 w_yhgl 的下部是一些操作数据窗口的按钮。dw_yhgl 的 rbuttondown 事件(鼠标右键)将弹出一个菜单 menu_yhgl,由此去分别执行用户管理的各个功能。

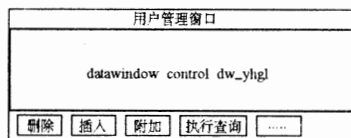


图2 用户管理窗口 w_yhgl

●菜单和 script

用户管理的各项功能都由菜单 menu_yhgl 列,它们包括新增注册用户、删除注册用户、更改注册口令,新增数据库用户、删除数据库用户、更改数据库用户的组号、授予用户权限、回收用户权限、终止用户进程等菜单项。鼠标在数据窗口 dw_yhgl 中按右键即可在光标处弹出菜单 menu_yhgl,选择相应的菜单项就能执行该项用户管理功能。下面仅就“更改数据库用户的组号”菜单项的 script 介绍如下:

当系统管理员激活“更改数据库用户的组号”菜单项的“clicked”事件后,客户机首先读取有关数据,然后组织成相应的动态 SQL 语句并执行,最后完成对用户

超细颗粒布朗运动模糊图象的增强处理技术

陈 卫 何振江 杨冠玲 罗 昱

郭永彩 谢莉莉 吴开华

(重庆大学光电精密机械研究所 630044)

摘要 在研究 $1\mu\text{m}$ 左右超细颗粒布朗运动的显微 TV 计算机系统中,需对图象信噪比(SNR)近于 1:1 的显微模糊图象进行增强处理,以提取超细颗粒的布朗运动信息。本文通过对 CCD 面阵结构和快速采样的显微 TV 模糊图象的分析,决定采用:1. 减去经多次平均后的背景图象;2. 平滑滤波处理;3. 选定特定的阈值进行非线性增强处理。最后得到清晰的在时刻 T_i 的超细颗粒布朗运动位置图象。

关键词 布朗运动 超细颗粒 CCD 图象增强 计算机

1 引言

利用微粒的布朗运动轨迹来测量 $1\mu\text{m}$ 左右超细粉体的粒径及其分布,这是一个全新的技术路线。^[1]在已建立的显微 TV 计算机系统中,首先要对图象信噪比(SNR)近于 1:1(如图 2、图 4. 对比)的快速连续采样的显微图象(受普通光学显微镜分辨极限的限制, $1\mu\text{m}$ 左右颗粒的显微成像只能是一个模糊的亮斑或暗斑,不可能得到清晰的轮廓外观成像)进行增强处

理,实现超细颗粒布朗运动的直观可视化,为提取超细颗粒在时刻 T_i 的位置信息和轨迹跟踪,做好图象的前期预处理工作。

2 CCD 面阵结构引入的噪声特点

图 1. 是 CCD 面阵图象传感器结构示意图,^[2]光敏阵列单元接收光能量产生光电荷在转移控制栅控制下移入垂直存储移位寄存器,在行同步脉冲控制下按行送入水平存储移位寄存器,水平存储移位寄存器在水平移位脉冲控制下逐点输出每个象元的光电信号,就得到了视频输出信号,经图象卡高速 A/D 采样就可获得原始目标的数字图象。当获得的原始图象信噪比

※本文工作得到国家自然科学基金资助※

收稿日期:1996 年 5 月 20 日

组号的设置。下面是菜单项“更改数据库用户的组号”的“clicked”事件处理程序。

```
introw __c
string str __zh, str __name, str __comm
if user __name() < 'sa' then /// 系统管理员无权执行该指令
    messagebox( '提示信息', '您无权执行该命令 !!!' )
    return
end if
if upper(dw __name) < 'DW __YHLB' then // 若 dw __yhl
    中所选数据窗口不
        return // 是 dw __yhlb, 则返回
    end if
w __yhl dw __yhl. accepttext()
row __c = w __yhl. dw __yhl. getrow()
str __zh = lower(w __yhl. dw __yhl. getitemstring(row __
c, 'yhzh'))
// 取出所选行的用户组号
str __name = w-yhg. dw-
yhl. getitemstring(row-c, 'yhbh')
// 取出所选行的用户编号
str __comm = 'sp __changegroup' + str __zh + ', ' + str __
name // 组织动态 SQL 语句
sqlca. autocommit = true
execute immediate 'use __cwgl', // 打开用户数据库 CWGL
```

```
execute immediate : str __comm; // 执行系统过程 sp __
changegroup
sqlca. autocommit = false
return
```

5 小结

在当前开发应用系统中, Sybase/Powerbuilder 是一种典型的 C/S 模式工作环境, 当然, 服务器端还可安装 Oracle, informix 等数据库, 客户端也可使用其他的开发工具, 但是任何应用系统都需要与之配套的用户管理系统。本文介绍的用户管理系统充分利用了 C/S 体系结构的特点, 用户只需要在客户端组织管理需求, 系统自动形成相应的 SQL 命令, 然后通过网络将 SQL 命令传输给服务器, 由服务器完成相应的用户管理工作。本文所述用户管理系统运用在本校行政 MIS 系统中, 受到系统管理员的一致好评。

参考文献

- 1 李强、陶浦洲: Sybase 数据库技术大全, 科学出版社龙门书局, 1995.
- 2 侯志平、蒋红: Powerbuilder 使用方法与实例, 北京晓通网络数据库研究所, 1995.
- 3 Gordon B. Davis, Margrethe H. Olson, Susan Conger and Janice I. Degross, "Management Information System", McGraw-Hill, Inc., 1986. 7

近于 1:1 时,就必须考虑噪声的引入途径、特点及其图象增强的算法。根据对显微 TV 计算机系统采样的模糊图象进行分析发现主要有如下两种噪声源:

首先是电源波动引入的噪声,电源波动引起照明系统的光源波动和 CCD 工作不稳定,解决途径是选用高精度的稳压电源对输入电压进行稳压,以降低或消除电源电压波动的影响。

其次就是 CCD 面阵结构特点引入的噪声,这也是最主要的噪声源。由采样的原始图象(如图 2、图 4)可知:光照较弱的情况下,视频输出具有较大的开关噪声影响,即由水平移位脉冲和垂直移位脉冲形成的噪声。水平移位脉冲形成的噪声在象素之间具有一定的相互影响性,因为光电荷是以“电荷包”的形式在水平存储移位寄存器的势阱中移动,由于移位寄存器之间的差异,致使电荷包并非 100% 的从一个势阱移入下一个寄存器的势阱,再加上移位脉冲的影响,就形成了水平方向(X 方向)的既具有随机性又具有相互关联的含有噪声的图象信号^[3]。Y 方向垂直存储移位寄存器产生噪声相互关联性远小于 X 方向的噪声信号,其任意随机性则更大,因为 Y 方向的移位脉冲(行同步脉冲)频率远小于 X 方向的移位脉冲频率(X 方向移位脉冲频率大约为行同步脉冲频率的 480~768 倍,依 CCD 面阵的象元而定),移位速度较低,因此电荷包中的电荷的迁移率就大大高于 X 方向移位寄存器,引入的开关噪声就大大低于 X 方向。而行与行之间的相邻象素噪声则表现出较大的无关性。这就为后续的增强处理提供了有利的条件。

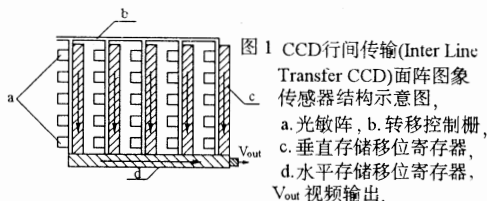


图1 CCD行间传输(Inter Line Transfer CCD)面阵图象传感器结构示意图,
a. 光敏阵, b. 转移控制栅,
c. 垂直存储移位寄存器,
d. 水平存储移位寄存器,
Vout 视频输出。

3 布朗运动模糊图象的增强^[4]

图 2. 为没有布朗颗粒时采样的背景数字图象,图 4. 为有布朗颗粒时采样的数字图象,由图 2、图 4. 可知:模糊的布朗颗粒图象是包含在背景和噪声信号之中的,而信噪比又近于 1:1,因此第一步要去掉较强的背景信号。

对图 2. 的背景图象可表示为:

$$G_{Bi}(x, y) = G_{Bo}(x, y) + n_i(x, y) \quad (x, y = 0, 1, 2, \dots, m-1) \quad (1)$$

式中:

$G_{Bi}(x, y)$ 表示第 i 次采样的背景图象;

$G_{Bo}(x, y)$ 表示不含噪声的背景图象;

$n_i(x, y)$ 表示第 i 次采样图象中所含的噪声信号。

对 k 次采样的背景图象进行平均,使 $\overline{n(x, y)} \rightarrow 0$, 这样就可得到不含噪声的背景图象:

$$G_{Bo}(x, y) = \frac{1}{k} \sum_{i=1}^k G_{Bi}(x, y) \quad (x, y = 0, 1, 2, \dots, m-1) \quad (2)$$

对 k 值的设定,我们经验性地定为 $k=8$, 因为 8 次平均的效果远高于 4 次平均的效果,与 16 次平均的效果相差不多,取 2³ 次平均是为了充分运用计算机的移位除法功能,提高运算速度。对连续 8 次采样的背景图象平均结果如图 3. 所示。

对图 4. 含有目标信息的数字图象可表示为:

$$G_i(x, y) = G_{oi}(x, y) + G_{Bo}(x, y) + n_i(x, y) \quad (x, y = 0, 1, 2, \dots, m-1) \quad (3)$$

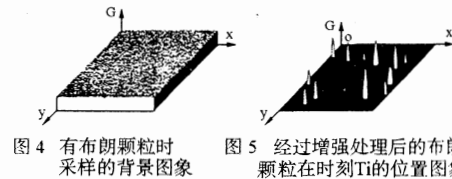
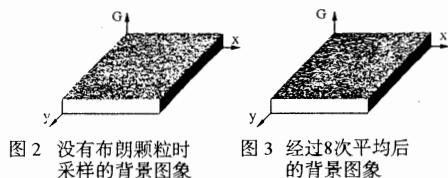
式中:

$G_i(x, y)$ 表示第 i 次采样的含有目标信息的数字图象;

$G_{oi}(x, y)$ 表示第 i 次采样的颗粒模糊数字图象。

从(3)中减去背景图象(2)得 $G'_i(x, y)$:

$$G'_i(x, y) = G_o(x, y) + n_i(x, y) \quad (x, y = 0, 1, 2, \dots, m-1) \quad (4)$$



对 $G'_i(x, y)$ 进行滤波处理,由前述 CCD 面阵结构引入的噪声特点确定选用 3×3 的窗口进行平滑滤波处理,以降低噪声 $n_i(x, y)$ 使其成为 $n'_i(x, y)$, 提高 $G'_i(x, y)$ 的信噪比,即:

$$\begin{aligned} G''_i(x, y) &= \sum_{(p, q) \in S} \omega(p, q) G'_i(p, q) \\ &= G_{oi}(x, y) + n'_i(x, y) \quad (x, y = 0, 1, 2, \dots, m-1) \quad (5) \end{aligned}$$

式中:

$G''_i(x, y)$ 表示经平滑滤波处理后的图象;

$n'_i(x, y)$ 表示经平滑滤波处理后的剩余噪声信号。

(x, y) 表示邻域 S 内的中心象素;

$\omega(p, q)$ 表示平滑滤波加权值,其矩阵表示为:

OLTP 中间件——TongEASY/TongLINK

余 坤 辛 涛

(电子科技大学微机所 成都 610054)

摘要 以“金卡”工程和各类城市综合网络为主的信息增值服务极大地刺激了 OLTP 发展,同时也对 OLTP 的平台国产化,开方性和性能提出了更高要求。基于 UNIX 平台在事务处理方面的缺陷,本文在 X/Open OLTP 规范模型的基础上,对其核心 TM 进行扩展,设计和开发了二个中间件产品——TongEASY 和 TongLINK,填补了我国在此领域的空白。

关键词 OLTP TongEASY/TongLINK

1 开放 OLTP 特性

事务处理市场的巨大需求曾经极大地促进了计算机和数据库系统的发展和普及。今天,计算机网络和分布式数据库系统的成熟及开放的热潮又促使事务处理系统在短短的几年中产生了质的飞跃。事务处理从集中走向分布,从昂贵的专有系统走向便宜、软硬件模块可选的开放分布式处理系统。

同时,OLTP 系统也相应经历了三代的风风雨雨。其中,第一代以 IBM 的 CICS 专用系统为代表,为事务处理系统树立了很好的典范,第二代是在象 UNIX 这样的开放平台上实现的 OLTP 系统,集中了事务处理,分布式处理和开放性三大软件技术,如 Encina^[1]

等,然而它们的开放性主要指系统平台的开放,各厂家仍然各自为政,缺乏互通性,这与开放性的最终目的是相抵触的;第三代 OLTP 系统强调 OLTP 本身的开放性,以 X/Open 制定的规范为代表,本文称之为开放 OLTP,如著名的 Tuxedo^[2],Yop End^[3]等。本文开发的 TongEASY 和 TongLINK 也是对此所作的尝试。

毫无疑问,分布式处理、事务处理和开放性当今三大软件技术将完美地融合在开放 OLTP 中,这使开放 OLTP 在保证事务原子性、完整性的基础上,具有负载均衡、可伸缩性和位置无关性等分布式处理能力,以及开放性带来的性能价格比方面的好处。

2 X/Open 模型

2.1 基础模型

X/Open OLTP 模型的提出是 OLTP 发展史上的一次巨大飞跃。如图 1 所示,X/Open OLTP 模型由以

※本文为“城市金卡工程支撑软件”项目※

收稿日期:1996 年 5 月 5 日

$$W = \frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \quad (6)$$

对 $G''_i(x, y)$ 选定特定的阈值 T 进行非线性 $F[G'']$ 放大拉伸处理:

$$G'_{oi}(x, y) = \begin{cases} F[G''_i(x, y)] & G''_i(x, y) \geq T \\ 0 & G''_i(x, y) < T \end{cases} \quad (7)$$

式中:

T 为根据每幅采样数字图象的象素灰度值和灰度直方图 $P[G''_i(x, y)]$ 而动态计算得出的阈值(其中 P_G 为大于灰度值 G 的象素百分率,其值大小由实验数据而经验性地选定):

$$T = \max \left\{ \frac{1}{3} \max [G''_i(x, y)], [G \leftrightarrow \int_G^{\max [G''_i(x, y)]} P(g) dg \geq P_G] \right\} \quad (8)$$

$F[G''_i(x, y)]$ 表示对图象灰度值进行非线性放大的函数,表达式为:

$$F[G''_i(x, y)] = G_m \sqrt{G''_i(x, y) / G_m} \quad (9)$$

式中:

G_m 为数字图象所能表示的最大灰度值。

经过以上处理过程后,最终得到如图 5. 所示的超细颗粒做布朗运动在时刻 T_i 的颗粒位置图象。

4 结束语

以上处理技术已在我室建立的显微 TV 计算机系统上采用 TURBO C 编程调试运行通过,对大量采样图象处理证明此方法切实可行。由此测定出的最小颗粒粒径为 $0.3 \mu m$ 左右,粒径分布与电子显微镜测定的结果能很好地吻合,因而证明了整个技术路线的正确性。

参考文献

- 1 Zhengjiang He, et al., Research on sensitivity threshold and parameters optimization of weak light detecting system in submicron particles measurement, 1st international symposium on test and measurement, 1995. 7. 579-582;
- 2 袁祥辉, 固体图象传感器及其应用, 重庆大学出版社, 1992. 1. 21-36;
- 3 宗孔德, 胡广书, 数字信号处理, 清华大学出版社, 1990. 8. 166-171;
- 4 周新伦等, 数字图象处理, 国防工业出版社, 1988. 6. 106-129.

下成员组成:AP、RM 和 TM^[4]。X/Open 组织定义了这些成员的功能和相互操作的界面,但并没有定义各成员实现的细节,为软硬件平台的选择和开发提供最大的灵活性。

2.1.1 AP(应用程序)

这是 OLTP 与用户的接口,负责将用户请求转换为内部服务请求。对分布式交易,X/Open 定义了 AP/TM 接口——TX——实施交易的边界控制。而 AP/RM 接口一般由 RM 厂商所定。

2.1.2 RM(资源管理程序)

RM 包括三部分(如图 2):

. Server(服务程序)。

Server 提供 AP(客户方)所要求的服务,一般与 AP 一样,由应用程序开发者所写。

. RMS(资源管理系统)

这常常指数据库管理系统(DBMS)或文件系统(FS)之类共享资源的管理系统。它一般由一独立厂商提供。当然,Server/RMS API 也由该厂商供应。

. R(资料)

由 RMS 管理的各种共享资源,如数据库(DB)、文件、打印设备和磁带备份等。

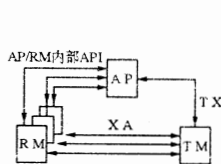


图1 X/Open OLTP模型

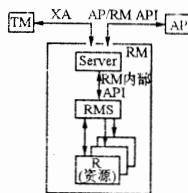


图2 RM模型

2.1.3 TM(交易管理程序)

TM 是 X/Open OLTP 的核心。它能支持 2 阶段提交,保证分布式交易的完整性和原子性,也能完成交易请求的分解、转发、优先级调度和监控,保证系统的负载平衡,为 OLTP 系统的容错性、安全性和分布式处理提供实质性基础。

TM 必须服从 TX、XA 接口规范,并提供这两种接口的 API。

因此,在 X/Open OLTP 模型中,只有 TM 是 OLTP 开发商必须提供的。利用其 TX、XA 接口,可开发出适用于各种平台各种应用的服务。所以 TM 是一中间件产品。本文为此开发的产品称为 TongEASY。

2.2 扩充模型

九十年代各种网络的急速登场,TM 作为 OLTP 的核心,作为一个中间件产品,要求 TM 必须适应电话网、X25 网、光纤网、卫星网及 ATM 网等响应延迟相差几十~几千个数量级的各类网络。这时 TM 的性能将产生极大影响。

因此,本文认为,将 TM 中的通信网络部分分离出来,组成独立的 CM(通信管理)模块,并向 TM 或其

它应用提供统一的 NET 网络接口(如图 3)。至于使用

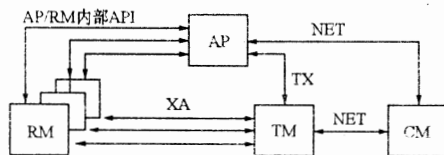


图3 扩充X/Open OLTP模型

什么网络、协议和介质,可在 CM 的配置中设置。CM 的设计目标应是:

. 适应于各种通用网络和介质;

. 与 AP、RM 和 TM 一样,是独立模块,可由第三方厂家供应,但服从 NET 接口。NET 接口目前只能由开发者所定。

. 能完成异种网间路由、报文的转发和相互备份。

为此,本文开发了 CM 产品 TongLINK。显然,基于此种目的的 TongLINK 也是一类中间件。

3 UNIX 平台上的 OLTP

3.1 难点

UNIX 作为开放系统的旗手,客户/服务器模型的代表及其极高的性能价格比,无疑成为开放 OLTP 的首选平台。然而,UNIX 本身不是专为事务处理设计的,在关键任务处理所要求的完整性、实时性上缺乏必要的支持^[5],使之在很长时间内难于进入 OLTP。这主要表现在:

. 无事件机制;

. 口令保护机制的脆弱性及超级用户权限太大;

. 难于查错,不能自行解决问题,恢复工作;

. 磁盘空间用完时,工作停止;

. 备份太耗资源,远程备份尤其如此,并阻塞网络;

. 分时系统,无实时机制。

八十年代末九十年代初 UNIX 的统一之战还在延续,其主要成果是安全性、实时性,可移植性等已大大加强;然而,事务处理的关键技术——数据完整性、可恢复性——没有实质性改进,这正是 TongEASY 首先应解决的问题。

3.2 TongEASY

3.2.1 TM 设计模型

根据 UNIX 的非事务特性,本文将 TM 分二层处理(如图 4)。第一层在 UNIX 上增加事务处理和容错处理能力,为 OLTP 提供基础;第二层在第一层基础上完成分布式处理、服务调度和监控等服务

服务调度/负载均衡	分布式事务管理	通信	配置、监控	第2层
基本事务处理	容错处理	编码/解码	安全服务	第1层

图4 TM 功能模型

3.2.1.1 第一层——基础服务

. 基本事务处理——完成二阶段提交。保证事务的完整性和原子性;

. 容错处理——同一服务可在多个 TM 上处理

当本地服务出故障时,本地 TM 可将此服务转发给最近能处理该服务的 TM。另外,它还可在服务过程中建立校验点,以便在出错时从最近的校验点恢复服务;

· 编码/解码——为报文结构提供统一的不依赖于硬件平台的编码规则和传输语法,为 TM 多平台应用打下基础。

· 安全服务——实现了 64 比特 DES 算法和分层密钥分配管理,可对报交加、解密及设置 MAC、DAC 等。

3.2.1.2 第二层——TM 功能模块

· 服务调度、负载平衡——目前,每一种服务都是由一进程来实现的,事务处理的大规模性决定了 TongEASY 性能的高低受系统进程数的影响。本系统通过建立进程“Cache”,使服务进程在服务完成后,在一定时间内,仍能保留在“Cache”中,等待下一步请求,当“Cache”命中率很高时,系统开销很小,可以极大提高系统性能;同时,TongEASY 通过提供服务请求优先级队列,限制每个服务类的最大进程数和系统总的进程数,动态调度服务,平衡系统负载。并可在基于 UNIX SVR 4.0 以上的 UNIX 版本上提供实时服务。

· 分布式事务管理——完成交易注册、注销和本地交易的执行;并能分解全局交易为多个子交易,使用二阶段提交技术,交与不同的 RM 处理,且提供相应的出错处理和恢复功能,保证分布式交易的完整性和原子性。

· 通信处理——包括两部分:本地 IPC 通信和网络路由。TongEASY 充分利用了 UNIX 提供的 IPC 机制与 AP 和 CM 交换信息的。

· 配置监控——参数配置,菜单驱动。通过动态监控交易过程,可由 TongEASY 动态改变交易结果或手工改变,避免“死”交易对系统性能和交易完整性造成不良影响。

3.3 TongLINK

正如前面描述的,TongLINK 设计的基点也很高,它应象 TM 一样作为一种中间件来开发。它应适用于多种软硬件平台,向上层应用屏蔽网络的复杂性,提供统一的 NET 接口。因此,菜单配置、参数驱动和各种 IPC 的使用贯穿于整个 TongLINK 的设计与开发中。

TongLINK 由配置和网络监控以及数据完整性处理模块组成。

· 配置和网络监控——参数配置,菜单驱动,主要需要配置主线路和备份线路类型以及线路状态监控周期。通过简单的网络状态动态监控,当网络状态不通时,可动态或手工启动备份线路;主线路恢复时,又可自动切换到主线路。

· 数据完整性处理——UNIX 事务处理的缺乏主要是没有数据完整性机制。TongEASY 的交易完整性

是建立在 TongLINK 数据完整性之上的。TongLINK 的数据传送支持小数据量(小于 1KB)交易包和大数据量(1KB 以上)文件传送方式(主要用于批量交易)。以 1KB 数据为单位设立校验点,一旦线路中断或网络阻塞而不通,在一定的时间内,TongLINK 试图建立备份线路,并重传上一校验点的数据。

4 结束语

本文根据 X/Open OLTP 模型设计和开发的 TongEASY、TongLINK 有 10 万行 C 语言源程序。目前,它已在 HP-UX Sun Solaris、Sequent PTX 和 SCO UNIX 的多种版本 UNIX 上移植成功。在短短一年中,已在银行、证券等多种 OLTP 系统进行试应用,取得了上百万的经济效益,节省了大量外汇。

开放 OLTP 才开始起步,潜力很大。本系统虽然充分利用了 UNIX 本身提供的高性能机制,实现了进程“Cache”、动态调度、负载平衡和其它内存和进程管理技术,增强了 UNIX 系统分布式事务处理方面的能力,获得了较高的性能价格比。通过一年的试运行,它已完全有能力承担“金卡”工程的挑战。

参考文献

- 1 Jeffrey L. Eppinger & Scott. Dietzen, Encina: Modular Transaction Processing”, IEEE Spring Symposium on Transaction Processing in 1992, p378.
- 2 Juan M. Andrade, Mark T. Carges, M. Randall MacBlane, “Open On-line Transaction Processing with the TUXEDO System”, IEEE Spring Symposium on Transaction Processing in 1992, p366.
- 3 Randy Smerik, “An Overview of TOP END”, IEEE Spring Symposium on Transaction Processing in 1992, p372.
- 4 X/Open, “The X/Open Reference Model for Distributed Transaction Processing”, X/Open Company Ltd., Guide November 1991.
- 5 Gary R. McInain, 《OLTP Handbook》Addison Press 1993.

征稿启事

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。

热忱欢迎广大作者踊跃赐稿!

《计算机应用研究》杂志社

用 GAL 器件设计地址译码器的新方法

郭显久

(大连水产学院电子工程系 116023)

摘要 本文采用数码比较器的原理给出用 GAL 器件设计任意地址范围译码器的新方法。在该方法中,提出了三个规则,利用这三个规则可以迅速简便地写出设计所需的地址译码器的逻辑表达式。这种方法对于使用 FM 汇编软件的用户非常实用,也可以用于软件编程。

关键词 GAL 器件 地址译码器 数码比较器

1 引言

在微机应用系统中,地址译码器是经常使用的。用于地址译码的器件一般为两类:一类是集成的固定译码器,比如,二—四译码器、三—八译码器、四—十六译码器等等,这种译码器使用方便,但由于译码器各输出端的地址间隔固定(一般为 16KB、8KB、4KB 等)的不足,这类译码器使用起来不够灵活;另一类是用 GAL 器件设计的地址译码器,且也是 GAL 器件最常见的应用方式,它可以设计出任意地址范围的译码器,并且还具有保密性。目前普遍应用支持 GAL 器件的设计软件有 FM 和 ABEL^[1],其中 ABEL 具有逻辑化简、模拟及自动生成设计文件等功能,并且还允许用户选择最自然的方式描述设计;FM 软件的特点是直观、简单、对于初级设计者容易掌握,但它只支持逻辑表达式的设计描述。

对于使用 FM 软件的用户在用 GAL 器件设计地址译码器时,虽然有些特殊的地址范围可以直接写出其地址译码器的逻辑表达式,但对任意地址范围的译码器,写出其逻辑表达式就不那么容易了。因此,如何直接地以最简式描述出任意范围地址译码器的逻辑表达式,这仍然是一个比较实际的问题。为此,本文提出了用数码比较器的原理设计任意地址范围译码器的方法,给出了用该方法的三个规则。利用这三个规则,可以直接写出地址译码器的最简逻辑表达式。

2 设计原理及规则

数码比较器是用来比较两组数码大小的,对于数码 A 和 D,其逻辑输出的真值有三种可能:大于、小于和等于,为了表示这三种逻辑表达式,引用符号 $F_{D>A}$ 、 $F_{D<A}$ 和 $F_{D=A}$ 。根据数码比较器输出的逻辑表达式,现用到地址译码器的设计中,对于数码范围 $[D_1, D_2]$,若有数码 A 满足 $F_{D_1-1<A}$ 、 $F_{D_2+1>A}$,则 $F = F_{D_1-1<A} \cdot F_{D_2+1>A}$ 的含义为:当 $D_1 \leq A \leq D_2$ 时, F 有效,由此可以看出逻辑表达式 F 正是地址范围为 $[D_1, D_2]$ 译码器的译码功能。根据这个原理得出设计任意地址译码器

的三个规则,现介绍如下:

规则 1:对于地址范围 $[D_1, D_2]$,若地址 A 满足逻辑表达 $F_{D_1-1<A}$ 和 $F_{D_2+1>A}$,则该地址范围的地址译码器的逻辑表达式为 $F = F_{D_1-1<A} \cdot F_{D_2+1>A}$ 。

规则 1 指出了写地址译码器逻辑表达式的方法即首先对低端地址 D_1 减 1,对高端地址 D_2 加 1,然后再分别写出 $F_{D_1-1<A}$ 和 $F_{D_2+1>A}$ 的逻辑表达式,最后将二者相与,便可得出地址译码器的逻辑表达式 F,这样通过 FM 软件对逻辑表达式 F 的编辑和对 GAL 器件的写入,就完成了地址译码器的设计。

由规则 1 看出,要想完成地址译码器的设计,还必须首先写出象 $F_{D_1<A}$ 和 $F_{D_2>A}$ 这样的最简逻辑表达式,如果采用常规法来写,比如全值表法或判断法^[2],对地址译码器输入变量较多时(比如大于四个),写全值表或简化的全值表就相当麻烦,并且化简也较困难。作者根据地址译码器的特点即地址范围的两端点地址为已知总结出写 $F_{D_1<A}$ 和 $F_{D_2>A}$ 最简逻辑表达式的两个规则,下面分别给予介绍,并通过例子说明它们的用法。

设数码 D 及 A 分别以二进制形式表示为

$$D = D_n D_{n-1} \cdots D_2 D_1 D_0$$

$$A = A_n A_{n-1} \cdots A_2 A_1 A_0$$

其中, D_i 和 A_i ($i=0 \sim n$) 分别表示它们的第 i 位, D_0 已知。

求 $D < A$ 的逻辑表达式 $F_{D<A}$ 的规则为:

规则 2:此规则只涉及到 D 的右端第一个 0 位起至最高位所对应的 A 中的那些位,所有操作均从右至左进行。其操作规则如下:

a:将 D 中遇到第一个 0 位起向左数连续的 0 位所对应 A 中的那些位相或。

b:接下来遇到的第一个 1 位起向左数连续的 1 位所对应 A 中的那些位相与,再和前面结果相与。

c:接下来再将遇到的第一个 0 位起向左数连续的 0 位所对应 A 中那些位相或,再和前面结果相或。

d:上述操作中,若遇到最高位则结束,否则重复 b、c 操作,直至最高位为止,所得结果即为 $F_{D<A}$ 的最简逻辑表达式。

求 $D \triangleright A$ 的逻辑表达式 $F_{D \triangleright A}$ 的规则为:

规则 3: 此规则只涉及到 D 的右端第一个 1 位起至最高位所对应 A 中的那些位, 所有操作均从右至左进行, 其操作规则如下:

a: 将 D 中遇到的第一个 1 位起向左数连续的 1 位所对应 A 中那些位的非相或。

b: 接下来遇到的第一个 0 位起向左数连续的 0 位所对应 A 中那些位的非相与, 再和前面结果相与。

c: 接下来再将遇到的第一个 1 位起向左数连续的 1 位所对应 A 中那些位的非相或, 再和前面的结果相或。

d: 上述操作中若遇到最高位则结束, 否则重复 b、c 操作, 直至最高位为止, 所得结果即为 $F_{D \triangleright A}$ 的表达式。

例 1: 已知 $D = 1FFFH$, 求 $F_{D \triangleright A}$ 的逻辑表达式。

解: 将 D 、 A 写成二进制数形式

$D = 0 \ 0 \ 0 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1$
 $1 \ 1 \ 1$

$A = A_{15}A_{14}A_{13}A_{12}A_{11}A_{10}A_9A_8A_7A_6A_5A_4A_3A_2A_1A_0$

由规则 2 得:

$$F_{D \triangleright A} = A_{15} + A_{14} + A_{13}$$

该逻辑表达式的含义是: 如果 A 中的 A_{15} 、 A_{14} 和 A_{13} 位有一位为 1, 不论其它位值如何, 则 $F_{D \triangleright A}$ 为真, 即 $A \triangleright D$ 。

例 2: 已知 $D = 9C6FH$, 求 $F_{D \triangleright A}$ 的逻辑表达式。

解: 将 D 、 A 写成二进制数形式

$D = 1 \ 0 \ 0 \ 1 \ 1 \ 1 \ 0 \ 0 \ 0 \ 1 \ 1 \ 0 \ 1$
 $1 \ 1 \ 1$

$A = A_{15}A_{14}A_{13}A_{12}A_{11}A_{10}A_9A_8A_7A_6A_5A_4A_3A_2A_1A_0$

由规则 2 得:

$$F_{D \triangleright A} = A_{15}(A_{14} + A_{13} + A_{12} + A_{11}A_{10}(A_9 + A_8 + A_7 + A_6A_5A_4))$$

按照例 1 同样可解释该逻辑表达式的含义。

例 3: 已知 $D = C000H$, 求 $F_{D \triangleright A}$ 的逻辑表达式。

解: $\because D$ 的右边第一个 1 位为 D_{12}

\therefore 根据规则 3, $F_{D \triangleright A}$ 只涉及到 A 中的 A_{15} 、 A_{14} 、 A_{13} 和 A_{12} 位。

将 D 、 A 写成二进制数形式

$D = 1 \ 1 \ 1 \ 1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0$
 $0 \ 0 \ 0$

$A = A_{15}A_{14}A_{13}A_{12} \dots$

由规则 3 得:

$$F_{D \triangleright A} = \overline{A_{15}} + \overline{A_{14}} + \overline{A_{13}} + \overline{A_{12}}$$

该逻辑表达式的含义是: 若 A 中的 A_{15} 、 A_{14} 、 A_{13} 、 A_{12} 位有一位为 0, 不论其它如何, 则 $F_{D \triangleright A}$ 为真, 即 $D \triangleright A$ 。

例 4: 已知 $D = 0DE5H$, 求 $F_{D \triangleright A}$ 的逻辑表达式。

解: $\because D = 0 \ 0 \ 0 \ 0 \ 1 \ 1 \ 0 \ 1 \ 1 \ 1 \ 1 \ 1$

$0 \ 0 \ 1 \ 0 \ 1$

$A = A_{15}A_{14}A_{13}A_{12}A_{11}A_{10}A_9A_8A_7A_6A_5A_4A_3A_2A_1A_0$

由规则 3 得:

$$F_{D \triangleright A} = \overline{A_{15}}\overline{A_{14}}\overline{A_{13}}\overline{A_{12}}(\overline{A_{11}} + \overline{A_{10}} + \overline{A_9}(\overline{A_8} + \overline{A_7} + \overline{A_6} + \overline{A_5} + \overline{A_4}\overline{A_3}(\overline{A_2} + \overline{A_1}\overline{A_0})))$$

3 地址译码器设计举例

某 8098 单片机应用系统中, 程序存储空间为 16KB, 数据存储空间为 32KB, 并且还需扩展的 I/O 口为: 8155、8255 和 8279 各一片, 其各部分地址空间安排如下:

EPROM27128: 2000H—5FFFH

RAM62256: 6000H—0DFFFH

I/O: 8155: 0E00H—0EFFH(RAM)

0F00H—0F05H(IO/ \overline{M})

8255: 0100H—0103H

8279: 01FEH—01FFH

对于上述系统扩展, 如果采用一片集成的固定译码器是不能实现系统扩展的, 那么用 GAL 器件就能很容易地实现系统的扩展要求。下面按照本文提供的方法, 写出用 GAL 器件实现符合本系统要求的地址译码器的逻辑表达式。由于各扩展芯片的片选端均为低电平有效, 所以, 逻辑表达式的输出均进行非操作。

(1) EPROM: 2000H—5FFFH

$\therefore D_1 = 2000H - 1 = 1FFFH, D_2 = 5FFFH + 1 = 6000H$

\therefore 根据规则 2 和 3 得:

$$F_{D_1 \triangleright A} = A_{15} + A_{14} + A_{13}, F_{D_2 \triangleright A} = \overline{A_{15}}\overline{A_{14}} + \overline{A_{15}}\overline{A_{13}}$$

由规则 1 可得出 EPROM 的片选信号 $\overline{CS_1}$ 的逻辑表达式为:

$$\overline{CS_1} = F_{D_1 \triangleright A} \cdot F_{D_2 \triangleright A} = \overline{A_{15}}A_{14}\overline{A_{13}} + \overline{A_{15}}\overline{A_{14}}A_{13}$$

(2) RAM: 6000H—0DFFFH

$\therefore D_1 = 6000H - 1 = 5FFFH, D_2 = 0DFFFH + 1 = 0E000H$

\therefore 根据规则 2 和 3 得:

$$F_{D_1 \triangleright A} = A_{15} + A_{14}A_{13}, F_{D_2 \triangleright A} = \overline{A_{15}} + \overline{A_{14}} + \overline{A_{13}}$$

由规则 1 得 RAM 的片选信号 $\overline{CS_2}$ 的逻辑表达式为:

$$\overline{CS_2} = F_{D_1 \triangleright A} \cdot F_{D_2 \triangleright A} = \overline{A_{15}}A_{14}A_{13} + A_{15}\overline{A_{14}} + A_{15}\overline{A_{13}}$$

(3) 8155: 0E00H—0F05H

$\therefore D_1 = 0E00H - 1 = 0DFFH, D_2 = 0F05H + 1 = 0F06H$

$$\therefore F_{D_1 \triangleright A} = A_{15} + A_{14} + A_{13} + A_{12} + A_{11}A_{10}A_9$$

$$F_{D_2 \triangleright A} = \overline{A_{15}}\overline{A_{14}}\overline{A_{13}}\overline{A_{12}}(\overline{A_{11}} + \overline{A_{10}} + \overline{A_9} + \overline{A_8} + \overline{A_7}\overline{A_6}\overline{A_5}\overline{A_4}\overline{A_3}(\overline{A_2} + \overline{A_1}))$$

\therefore 8155 的片选信号 $\overline{CS_3}$ 的逻辑表达式为:

$$\overline{CS_3} = \overline{A_{15}}\overline{A_{14}}\overline{A_{13}}\overline{A_{12}}A_{11}A_{10}A_9\overline{A_8} + \overline{A_{15}}\overline{A_{14}}\overline{A_{13}}\overline{A_{12}}A_{11}A_{10}A_9\overline{A_7}\overline{A_6}\overline{A_5}\overline{A_4}\overline{A_3}(\overline{A_2} + \overline{A_1})$$

$$IO/\overline{M} = A_8$$

通用型智能化实时数据显示与处理系统

程芳真 蒋滋康

(清华大学热能工程系 北京 100084)

摘要 本文介绍了一种可用于计算机监控与仿真的通用型智能化的实时数据显示与处理系统,并给出了其功能、特点与实现方法。该系统使得用户摆脱了繁琐的编程过程而方便地使用本系统提供的各种数据显示方式和处理方法来完成自己的工作。

关键词 计算机应用 智能 仿真 监控

1 引言

在计算机仿真和监控系统中,存在着大量的数据需要显示与处理,以满足人机交互的需要。而且在这些系统中,数据量大、显示需求形式多种多样,因此需要显示与处理系统不仅要有良好的用户界面,而且能够提供各种显示与处理方式。

在目前的仿真和监控系统中,一般的处理方法都是根据自己的要求进行特定情况下的设计开发,这种方法开发周期长,无法重复利用,而且给用户的维护带来了极大的不便。本人在多年的实际工作过程中,对此深有体会,因此结合自己的实际工作经验,并在对各种需求研究的基础上,设计了一种简单易用的通用型智能化的实时数据显示与处理系统。

2 功能

本智能显示与处理系统是基于微机 DOS 操作系统

统进行开发,主要功能简介如下:

●数据表格显示

实时数据以表格的方式显示出来,使用户可以系统地了解某一组相关数据,并便于进行比较。

●动态曲线显示

实时数据以动态曲线的方式显示出来,使用户不但可以了解当前运行数据,而且可以了解相应参数的历史情况和变化趋势。

●系统图形显示

实时数据在相关的系统图形上显示出来,使用户可以以直观、形象的方式了解当前系统的运行情况。

●直方图显示

实时数据以直方图的形式进行显示,使用户可以方便地比较若干个相关参数的值的大小。

●数据超限报警

用户给定报警信息及相应数值的上、下限,系统可跟踪相应的参数,并及时显示超限状态,并给出报警提

收稿日期:1996年7月8日

(4)8255:0100H—0103H

由于该地址范围小,因此直接写出 8255 片选信号 CS4 的逻辑表达式为:

$$\overline{CS4} = \overline{A_{15}} \overline{A_{14}} \overline{A_{13}} \overline{A_{12}} \overline{A_{11}} \overline{A_{10}} \overline{A_9} \overline{A_8} \overline{A_7} \overline{A_6} \overline{A_5} \overline{A_4} \overline{A_3} \overline{A_2}$$

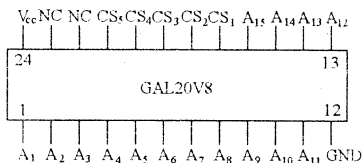
$8A_0 = A_0$ $8A_1 = A_1$ ($8A_0$ 和 $8A_1$ 为 8255 的地址选择线)

(5)8279:01FEH—01FFH

直接写出其片选信号 CS5 的逻辑表达式:

$$\overline{CS5} = \overline{A_{15}} \overline{A_{14}} \overline{A_{13}} \overline{A_{12}} \overline{A_{11}} \overline{A_{10}} \overline{A_9} \overline{A_8} \overline{A_7} \overline{A_6} \overline{A_5} \overline{A_4} \overline{A_3} \overline{A_2} \overline{A_1}$$

$9A_0 = A_0$ ($9A_0$ 为 8279 的地址选择输入线)



在上述地址译码器的逻辑表达式中, $A_i (i=0 \sim$

15)均为地址线, $CS_j (j=1 \sim 5)$ 为各片的片选信号,低电平有效,其中 $CS4$ 和 $CS5$ 是直接写出的,也可以采用本方法,结果是一样的。为了系统扩展的要求,选用 GAL20V8,其构成的译码器的引脚图如下:

4 结束语

本文详细介绍了用 GAL 器件设计任意地址范围的地址译码器的方法,利用该方法中的三个规则,可以很容易地写出地址译码器输入和输出的逻辑表达式,并且基本不需要化简。该方法对于用 GAL 器件设计任意地址范围的地址译码器,且采用 FM 汇编软件的用户是非常实用的。也可以利用这三个规则进行软件编程,通过对地址译码器的描述,自动生成地址译码器的逻辑表达式和设计文件,使设计更加简便。

参考文献

- 1 居伟编著. 可编程逻辑器件的开发与应用. 人民邮电出版社. 1995. 2
- 2 袁永升编. 数字逻辑电路设计. 高等教育出版社. 1985 年. 9.

示。

●报表打印的功能

根据用户给定的报表格式和需要打印的相关数据,在不影响系统运行的情况下,进行完全的实时打印。

●菜单的自动生成

在本系统中,菜单分为主菜单和显示选择菜单,前者提供系统主要的操作命令和选择显示组,后者用来选择选定组内的显示画面。

用户可根据自己的要求,按照一定的规则给出自己的主菜单文件。系统以菜单标识符来识别菜单命令,用户只要正确给定了标识符,主菜单形式可以自由组合。

对某一类显示方式,用户可以根据实际系统分成若干组,并将每一组存放在一个文件中。用户利用主菜单选择了某一组后,系统可以自动提取出该组对应文件的标题项信息而形成显示选择菜单,供用户选择一个显示画面。

用户在使用本系统时,无需自己编程,只要按照一定的格式给出相应的数据文件,系统即可自己完成其它必要的所有工作。

本系统在设计和实现上都采用面向对象的方法,把数据表格、动态曲线、系统图形、直方图、数据超限报警等共同的地方使用一个基类实现,而这些类本身只需要实现其自身所特有的功能。在以后需要新的显示方式时,也比较容易进行扩充。

由于在设计思想和实现方法上,各种显示方式具有一定的相似性,下面我们就以比较典型的数据表格显示为例进行具体的描述。

3 文件格式

在实际的显示中,显示内容包括静态画面、文字及动态变化的数据,如图 1 所示,如何把它们结合起来,给用户一个方便的接口呢?

名称	单位	数值	运行值
汽包压力	MPa	9.8	9.82
汽包温度	℃	540	540.3
汽包水位	mm	11.3	11.34
炉膛温度	℃	1210	1210.4
炉膛压力	MPa	12.0	12.01
炉膛流量	kg	213	213.5
炉膛速度	mm	20	20.1
炉膛温度	℃	200	200.0
炉膛压力	MPa	12.0	12.01
炉膛流量	kg	213	213.5
炉膛速度	mm	20	20.1
炉膛温度	℃	1000	1000.0
炉膛压力	MPa	14.0	14.01
炉膛流量	kg	40.354	40.354
炉膛速度	mm	30.089	30.089

图 1 数据表格显示

为此,系统把需要显示的内容分为两部分:静态数据和动态数据。动态数据由通信模块提供,静态数据包括表格的表头、格式、文字等,由用户在文件中给定。

每个文件中可以包含多页表格,每个表格的内容在[page]... (page end)关键词之间定义。

```
Table For ZNXS System 3.0, Copyright by Cheng Fangzhen
[page]
(title)
锅炉主要参数
(column number)
4
(column name)
名称
单位
额定值
运行值
(column color)
BLACK BROWN MAGENTA RED
(is data)
0001
过热蒸汽流量 T/H 220 1
过热蒸汽压力 MPa 9.8 2
过热蒸汽温度 ℃ 540 3
汽包压力 MPa 11.3 4
(page end)
[page]
(page end)
[end]
```

数据表格文件格式

以上是一个表格文件的内容,第一个[page]和 (page end)之间的内容定义了图 1 所示的表格。

在对表格的各种形式进行分析的基础上,我们认为一个表格的内容应包括:标题、表格列数、各列的名字、各列的颜色、以及各列实际的显示内容。为了辨别各列的内容是静态数据还是动态数据,我们采用了属性标记的方法,各列对应一个属性标记,1 表示该列为动态数据,0 表示该列为静态数据。

对于这些内容,我们采用了一些关键字来标记各段内容:[]内为主关键字, ()内为次关键字,各关键字的含义为:

●[version]表示下一行为版本信息;

●[page]表示一个表格的开始;

●[end]表示文件结束。

●(title)表示下一行为表格标题;

●(column number)表示下一行为表格列数 N;

●(column name)表示下面 N 行为各列的名字;

●(column color)表示下一行为 N 列的颜色;

●(is data)表示下一行为 N 列的属性标记;

●(text and order)表示以下若干行为表格中各行所要显示的具体内容。对于一行中各列的内容,如果对应的列属性标记为 0,则为静态数据,否则应为一整数值表示所要显示数据的地址(从仿真或监控系统中传来的数据为一实数数组,该地址为数据的数组下标值);

●(page, end)表示页的结束。

通过这种动静结合的数据表示方式来表示各种形式的表格,这样用户无需进行编程,就可以很方便地实现自己的显示要求。

4 实现方法

表格的显示由类 TableClass 完成,它由基类 ZNXSClass 派生出来。基类主要提供画面数据的存储与管理、数据链表的管理、画面显示与动态数据刷新的框架程序以及文件读入和检查所需的基本函数等,并提供了大量的虚函数供衍生类重载。表格类则提供了表格数据结构,并重载了基类的虚函数以实现表格文件的读入、表格数据结构的存储与管理、静态画面和动态数据的显示等功能。

[version]

(text and order)

4.1 数据结构

类 TableClass 的设计是根据表格的数据内容,并结合表格显示的特点设计的。它在基类 ZNXSClass 的基础上添加了针对表格的一些数据变量和函数,包括从文件中读入的数据和一些显示所需的辅助变量。其中最主要的是保存一页表格的结构 TableStruct,它保存从文件中读入的表格的静态数据并为动态数据分配空间,其内容如下:

```
struct TableStruct{           //表格数据结构
    char * title;              //表头
    char ** name;              //各列名字
    int nLine;                 //表格行数
    nColumn;                  //表格列数
    int * color;               //各列显示颜色
    int * isData;              //各列属性标记
    int nText;                 //静态文字列数
    int nData;                 //动态数据列数
    struct LineStruct{         //行结构
        char ** text;          //行的静态文字
        int * order;           //行的动态数据地址
        float * data;          //行的动态数据
        LineStruct * next;     //指向下一行的指针
    } * lineHead;              //指向首行的指针
    TableStruct * last, * next; //指向上一页、下一页的指针
};
```

表格数据结构

由于采用了链表式数据结构,使得数据和内存的管理变得非常方便,不但对数据个数可以不加限制,而且优化了内存的使用。

4.2 数据读入

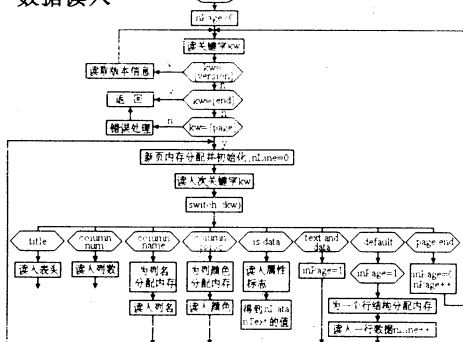


图2 文件读入程序流程图

数据文件是系统与用户的接口,其读入与分析是本系统一个重要的组成部分。数据读入过程主要包括读入、语法检查、分析、内存分配等。其具体过程如图2所示。

在程序中,采用了变量 inPage 来表示是否正在读取一页表格,如果是,且读入的不是次关键词,则读入一行表格数据,直到遇到<page end>关键字。

4.3 显示

在某一页表格初次被选中时,需要显示表格静态画面和当前动态数据,以后则只需显示变化的数据。

初次显示过程和动态数据显示刷新流程图如图3(a)、(b)所示。

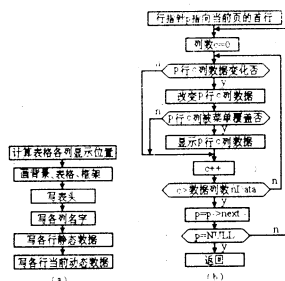


图3 静态画面和动态数据显示流程图

其中,计算显示位置是根据表格的内容,计算表格的各部分各所需占空间的大小,使它显示在屏幕的中央。

5 特点

系统在设计 and 实现过程中考虑了用户的各种需求,所有工作都是我们在自己工作经验的基础上进行分析总结的基础上完成的,为用户提供了最大限度方便。其主要特点有:

5.1 灵活的数据格式

为用户提供了方便的数据表示方式,用户可以在此基础上设计自己的数据表示形式。

- 每个文件可以包含很多页表格;
- 画面内容与显示数据可由用户任意定制;
- 在数据读入时可以对数据进行错误检查并指出非法内容。

5.2 友好的用户界面

●西文状态下的全部汉化。系统完全汉化,无需任何汉字环境的支持,便于普通用户的使用。由于采用了对显卡内存的直接存取,使得汉字具有极高的显示速度,同西文显示几乎没有任何区别。

●键盘与鼠标的同时操作。系统可以同时使用键盘和鼠标进行操作各种,以满足不同用户的要求,为用户带来了方便。

●方便的菜单操作。菜单按照用户的要求自动生成后,将操作内容一目了然地呈现在用户面前,用户可方便地执行相应的操作。

5.3 最少的内存需求

在软件编制过程中,对计算机内存的使用作了合理的安排,全部采用动态内存的方法进行内存管理。除了必须保留历史数据的曲线,在系统开始运行时,将全部数据自动读入内存外,其它数据均是在需要显示时才读入。这样,尽管用户存在大量的显示数据,但在任何时候所需要的内存量都是很小的。因此,用户无需担心因数据过多而无法处理。

5.4 版权与帮助信息的自动生成

研制人员可以根据自己的要求而改写版权显示文

MASM6.x 中的 .EXE 文件压缩技术分析

贾志先

(新疆财经学院基础部 乌鲁木齐 830011)

摘要 MASM6.0、6.11 是 Microsoft 公司于 1992、1993 年先后推出的汇编程序软件,是目前 MASM 的最新版本。比起以前的版本,该软件提供了不少新的功能,其中一个功能是引入了文件压缩技术。用 MASM6.x 建立的 .EXE 压缩文件,给用户调试和分析程序带来了一定的困难。下面,我们对 MASM6.x 中的 .EXE 文件压缩技术做一分析。

1 MASM6.x 中 .EXE 文件的压缩方法

1. 压缩范围:数据段、代码段、附加段、堆栈段。
2. 压缩方法:第一个重复字符在 37H 个以上,以后重复字符在 8 个以上进行压缩;否则不压缩。
3. 压缩存储格式:
压缩字符 长度 结束码
其中:压缩字符占一个字节,长度占两个字节,结束码占一个字节,结束码有以下几种:

B0 压缩结束

B1 第一个压缩结束

B2 非压缩结束

例如:00 00 04 B1 表示第一个压缩 400H 个 0 字符;
20 08 00 B0 表示压缩 8 个 20H(空格) 字符;
... 56 01 B2 表示非压缩 156H 个字符 结束。

2 MASM6.x 中 .EXE 压缩文件的还原方法

为了 .EXE 压缩程序能正常执行,MAASM6.x 在压缩程序中加入程序外壳,用来还原压缩程序和转到程序入口。

MAASM6.x 中 .EXE 压缩文件的还原方法为:首

先将程序外壳移到高地址上,然后从高地址向低地址进行还原,接着进行重定位,最后转向还原后的程序入口进行执行。

在 DEBUG 下,对压缩的 .EXE 文件进行调试,可以反汇编出以下程序外壳代码:

C>DEBUG.EXE

文件名

-U10 11B

XXXX:0010 8BE8 MOV BP,AX

XXXX:0012 8CC0 MOV AX,ES

XXXX:0014 051000 ADD AX,0010

XXXX:0017 0E PUSH CS

XXXX:0018 1F POP DS

XXXX:0019 A30400 MOV [0004],AX

XXXX:001C 03060C00 ADD AX,[000C]

XXXX:0020 8EC0 MOV ES,AX;设置移动外壳的附加段址

XXXX:0022 8B0E0600 MOV CX,[0006];取程序外壳长度

XXXX:0026 8BF9 MOV DI,CX

XXXX:0028 4F DEC DI

XXXX:0029 8BF7 MOV SI,DI

XXXX:002B FD STD

XXXX:002C F3 REPZ

XXXX:002D A4 MOVSB;移动程序外壳

XXXX:002E 50 PUSH AX

XXXX:002F B83400 MOV AX,0034

XXXX:0032 50 PUSH AX

收稿日期:1996 年 5 月 7 日

件和帮助信息文件,系统可自动生成显示画面。

6 结束语

目前本系统主要应用于 DOS 系统,由于软件全部采用 C++ 语言编制而成,具有良好的移植性,可以方便地移植到其它机型和操作系统。

作者还准备开发一套结合本系统的辅助工具,主要用来辅助用户进行各种数据文件的生成,以更加方便用户的使用。

在使用过程中,根据用户的进一步要求,还可能要增加一些新的功能,以不断适应实际的需要。

参考文献

- 1 Ted Faison. Borland C++3.1 编程指南. 北京:清华大学出版社,1993
- 2 严蔚敏,吴伟民. 数据结构(第二版). 北京:清华大学出版社,1992
- 3 求伯君等. 新编深入 DOS 编程. 北京:学苑出版社,1993
- 4 尹彦芝. C 语言高级实用教程. 北京:清华大学出版社,1992
- 5 董士海等. 图象格式编程指南. 北京:清华大学出版社,1994
- 6 邱玉春等. AutoCAD<10-11>使用手册. 北京:电子工业出版社,1993

```

XXXX,0033 CB RETF ;转程序外壳下一指令
XXXX,0034 8CC3 MOV BX,ES
...
XXXX,0040 B91000 MOV CX,0010
XXXX,0043 BOFF MOV AL,FF
XXXX,0045 F3 REPZ
XXXX,0046 AE SCASB ;查找压缩程序结尾
;注:在压缩程序和外壳之间为 1—16 个字节 FF
...
XXXX,0066 03F0 ADD SI,AX
XXXX,0068 8EDA MOV DS,DX ;调整源串逻辑地址 DS,SI
...
XXXX,007C 03F8 ADD DI,AX
XXXX,007E 8EC2 MOV ES,DX ;调整目的串逻辑地址 ES,DI
XXXX,0080 AC LODSB ;取结束码
XXXX,0081 8AD0 MOV DL,AL
XXXX,0083 4E DEC SI
XXXX,0084 AD LODSW ;取长度
XXXX,0085 8BC8 MOV CX,AX
XXXX,0087 46 INC SI
XXXX,0088 8AC2 MOV AL,DL
XXXX,008A 24FE AND AL,FE
XXXX,008C 3CB0 CMP AL,BO ;是压缩结束码?
XXXX,008E 7505 JNZ 0095 ;不是转 0095
XXXX,0090 AC LODSB ;取压缩字符
XXXX,0091 F3 REPZ
XXXX,0092 AA STOSB ;还原压缩字符
XXXX,0093 EB06 JMP 009B
XXXX,0095 3CB2 CMP AL,B2 ;是非压缩结束码?
XXXX,0097 756D JNZ 0106 ;不是转 0106
XXXX,0099 F3 REPZ
XXXX,009A A4 MOVSB ;非压缩字符复制
XXXX,009B 8AC2 MOV AL,DL
XXXX,009D A801 TEST AL,01 ;是第一个压缩结束码?
XXXX,009F 74B1 JZ 0052 ;不是转 0052
XXXX,00A1 BE3201 MOV SI,0132 ;SI 指向重定位表项数
XXXX,00A4 0E PUSH CS
XXXX,00A5 1F POP DS
XXXX,00A6 8B1E0400 MOV BX,[0004]
XXXX,00AA FC CLD
XXXX,00AB 33D2 XOR DX,DX
XXXX,00AD AD LODSW ;取重定位表项数
XXXX,00AE 8BC8 MOV CX,AX
XXXX,00B0 E313 JCXZ 00C5
XXXX,00B2 8BC2 MOV AX,DX
XXXX,00B4 03C3 ADD AX,BX
XXXX,00B6 8EC0 MOV ES,AX
XXXX,00B8 AD LODSW ;取重定位表
...
XXXX,00C0 26 ES,
XXXX,00C1 011D ADD [DI],BX ;重定位

```

```

XXXX,00C3 E2F3 LOOP 00B8
...
XXXX,00E1 8BC3 MOV AX,BX
XXXX,00E3 8B3E0800 MOV DI,[0008]
XXXX,00E7 8B360A00 MOV SI,[000A]
XXXX,00EB 03F0 ADD SI,AX
XXXX,00ED 01060200 ADD [0002],AX;设置还原后的入口段址
XXXX,00F1 2D1000 SUB AX,0010
XXXX,00F4 8ED8 MOV DS,AX ;设置数据段段址
XXXX,00F6 8EC0 MOV ES,AX ;设置附加段段址
XXXX,00F8 BB0000 MOV BX,0000
XXXX,00FB FA CLI
XXXX,00FC 8ED6 MOV SS,SI ;设置还原后堆栈
XXXX,00FE 8BE7 MOV SP,DI ;段址和栈顶指针
XXXX,0100 FB STI
XXXX,0101 8BC5 MOV AX,BP
XXXX,0103 2E CS;
XXXX,0104 FF2F JMP FAR [BX];转还原后的入口
XXXX,0106 B440 MOV AH,40
XXXX,0108 BB0200 MOV BX,0002
XXXX,010B B91600 MOV CX,0016
XXXX,010E 8CCA MOV DX,CS
XXXX,0110 8EDA MOV DS,DX
XXXX,0112 BA1C01 MOV DX,011C;显示字符串"Packed"
XXXX,0115 CD21 INT 21 ;file is corrupt"
XXXX,0117 B8FF4C MOV AX,4CFF
XXXX,011A CD21 INT 21 ;错误退出
CS:0000—000F 及 CS:011C—文件结束为程序外壳的数据部分,其中:
CS:0000—0003 字节用来保存压缩程序还原后的入口地址;
CS:0004—0005 字节用来保存中间结果;
CS:0006—0007 字节用来保存程序外壳的长度;
CS:0008—000B 字节用来保存压缩程序还原后的堆栈段相对段址和栈顶指针;
CS:000C—000D 字节用来保存压缩程序还原后的节长度;
CS:000E—000F 字节内容为 52 42
DS:011C—0131 字节为字符;"Packed file is corrupt";
DS:0132-0133 为压缩程序还原后的重定位表项数;
DS:0134 开始为压缩程序还原后的重定位表。

```

3 使用 MASM6.x 对 .EXE 文件进行压缩的方法

使用 MASM6.x 对 .EXE 文件进行压缩的方法为:

1. 在 PWB(程序员工作平台)下设置方法:

在 MASM6.0 的 PWB 中,选择 Options 子菜单的 Link Options...项,在 Link Options 对话框中选择(Set Release Options...)按钮,在 Set Release Options 对话框中选择复选项[]Pack EXE File;

在 MASM6.11 的 PWB 中,选择 Options 子菜单

IF 基准存在 THEN 子波名下限 “=”

基准位置 “+” 值

IF 形态框架 \wedge “DO”形态匹配器

THEN 子波位置

IF 样本库 \wedge “DO”样本匹配器

THEN 子波位置

以上带下划线的名词或数值均由规则器填入,这样就产生了规则库。

除了上述规则外,还有一类重要的函数求解规则库。其一般形式如下:

IF 输入参数表 THEN “DO”函数

IF 输入参数表 \wedge “DO”函数 THEN

输出参数表

这里需要说明的是“DO”函数。“DO”函数实际上是一组可执行文件。可用于确定时域波形的峰、谷值,时域波形的面积,子波的起、止点,聚类分析,分段直线拟合,曲线拟合等。这些函数各自独立,全部存放在一个子目录为 TOOLLIB 的工具库内。每个函数都有其输入接口和输出函数。当推理机检测某条规则,遇到类型为“DO”的前提条件及其函数名,且在动态库中存在该函数调用的所有输入参数值,推理机就能触发该函数进行计算,同时将生成值填入动态库中。

基于以上产生式规则库,推理机采用目标驱动的混合推理策略。这种控制过程可用结构化语言描述如下:

BEGIN

IF X is stored in global database

THEN X is TRUE

ELSE

IF there are any rules whole consequence

include X

THEN for each such rule;

IF all premises are TRUE

THEN add all consequences to the

global database

AND X is TRUE

ELSE

IF the user says that X is TRUE

THEN X is TRUE

ELSE X is FALSE

END

2 推理策略及其它优化措施

推理机采用深度优先搜索策略,按优先度方法来选择规则,即推理机一般先用前提的未知量少的规则。使用如下公式:

$$\text{优先度} = \frac{\text{前提未知量}}{\text{前提总量}} \times 100\%$$

推理机总是先搜索整个规则库,将结论匹配的规则暂时登记起来,然后计算各个前提的优先度,最后选用一个优先度值最小的规则进行匹配。同时规定最大

搜索深度,如果推理中当前搜索深度大于或等于最大深度,推理机将回溯并进行剪枝操作,先将推理树的这部分登记在拒绝表中,然后从推理树中剪去这部分枝叶。

一条规则仅触发一次,即某条规则触发后,就将其从规则库中动态地删除去。需要备一张删除表,该表的作用是记录被删除规则的规则号。待下一次目标求解时,Load-kb()函数将依据删除表有选择地载,使得程序执行会更快些。

3 推理链的生成及推理机的实现

推理链的根结点就是所要求的目标值,推理链生成的过程见图二。

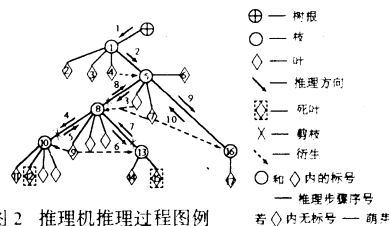


图 2 推理机推理过程图例

此处,

树根:即目标结点,一般附带有前一缀词。

枝:指匹配某一规则而生成的结点,若该规则中有一前提值为 FALSE,这条树枝要被剪去。

叶:指匹配某一规则中的前提而生成的结点,满足条件的叶是活叶,否则为死叶。

剪枝:指推理机删除枝结点的操作。

萌芽:指匹配的规则中还未匹配证实(不管结果可能是 TRUE 还是 FALSE)的前提条件,萌芽是潜在的叶结点。

衍生:就是枝的叶结点,因在事实库内找不到证据而去规则库寻找相匹配的规则结论,从而匹配的规则生成一条新枝的过程。

用户输入一目标后,推理机首先在事实库中寻找与目标匹配的事实,这是现成值。如果没有,则推理机建立一个树根结点,进入反向推理求值状态。图二中 1~10 表示生成枝和剪枝步骤。(1)步:生成枝 1,然后检查枝 1 的叶结点,叶 2、叶 3、叶 4,结果叶 2、叶 3 为活叶,叶 4 衍生出枝 5。(2)步:生成枝 5,叶 6 为活叶,叶 7 衍生出枝 8。(3)步:生成枝 8,叶 9 衍生出枝 10。(4)步:由于叶 12 是死叶,所以枝 10 被剪去。(5)步:返回枝 8,检查叶 9,发现叶 9 可以衍生出枝 13。(6)步:进入枝 13,发现叶 15 是死叶,剪枝返回枝 8。(7)步:返回枝 8 后,发现叶 9 再也衍生不出其它树枝。(8)步:将枝 8 剪去并返回枝 5。(9)步:叶 7 衍生成枝 16,枝 16 只有一个叶 17 且是活叶。至此,只有一条枝,即枝 16 是不必剪的,但这时还要返回,因为枝 5 上还有一个萌芽,不知该萌芽生成的枝是死的还是活的,需要返回检查。由此可知推理机的工作过程是边长树枝边修剪枯死的枝

叶。推理机停止工作的条件是:(1)成功:遍历的每条枝上叶均为活叶;(2)失败:返回至根时,已没有其它枝可长。

必须提到一个细节,即被剪去的枝连同使叶子至死的事实均被收集到一垃圾桶内(garbage-bin)。称之为垃圾只是比较形象而已并非没有用。garbage-bin的作用是记录剪去该枝叶的原因,即拒绝其规则成立的事实和该规则叶。

下面给出 C 语言的 back-reason()源程序。

```
int back __ reason (RULE __ BASE * rule, REASON __
TREE * rsn __ tree)
{
    while (rule == NULL) { /* no rules can be choosed, it
possibly failures */
/* if it comes to the root, then failures */
        if (curdepth < 0) return (FAILURE);
/* if it probably has another choice, then try */
/* abandon the invalid branch, then choose another choosible
rule */
        rsn __ tree = clip __ branch (rsn __ tree);
        rule = choose __ rule (rule __ head, rsn __ tree->leaf);
        curdepth--; /* decrease the searching depth by one */
    }
/* try the choosed rule */
    rsn __ tree = gen __ branch (rsn __ tree, rsn __ tree);
    leaf = first __ leaf (rsn __ tree, rule->prms);
    do {
        if (premise __ match (rule->prms) == FALSE)
            while ((rule = choose __ rule (rule __ head, leaf)) !=
NULL)
                if (back __ reason (rule, rsn __ tree) == SUCCEED)
                    break;
/* no other rules can be choosed, then abandon the branch,
return failure */
        if (rule == NULL) break;
        rule->prms = rule->prms->next;
        if (rule->prms != NULL) /* if next premise exists, generates
another leaf */
            leaf = next __ leaf (rsn __ tree->leaf, rule->prms);
        else break;
    } while (1);
    if (rule == NULL) return (back __ reason (rule, rsn __
tree));
/* add the newfact to the fact base */
    produce __ new __ fact (fact __ head, rsn __ tree, rule);
    return (SUCCEED); /* the current rule is true */
}
```

back-reason()设置了两个出口条件:

(1)返回值为 SUCCEED 的出口,条件是凡所有叶为活叶的枝,都可返回。

(2)返回值为 FAILURE 的出口,条件是当再也长不出新枝(即无规则可供匹配)时。

递归调用条件:

(a)当叶结点需衍生时;

(b)需要剪枝并生成新枝叶时。

以上两个条件,只要满足其中之一就执行递归调用。

back-reason()由三大模块组成。(A)块执行剪枝和选择有希望生成枝的规则,其中有一失败出口;(B)块进行搜索枝中每一叶结点的存活状况,叶的衍生就在此期间,需要剪枝和长枝的递归出口亦在其中;(C)块完成添加新事实和返回成功标志的功能。

back-reason()共调用了七个函数:

clib-branch(), choose-rule(), gen-branch(), first-leaf(), premise-match(), gen-leaf() 和 produce-new-fact()。

这里 clib-branch()和 gen-branch()为链表操作,前者是删除一个链结点,后者用于增加一个链结点。gen-leaf()和 first-leaf()除了有增加一个链结点操作外,还有结构体赋值的操作。choose-rule()完成选择一条新假设的操作,当找到一条新假设时,当前搜索深度计数器加 1,而后判定该值是否超界,若越界就放弃这条假设返回空指针,否则返回该条规则指针。premise-match()用于判定该假设中的一条前提是否直接成立,直接成立是指可以直接从事实库中找出前提条件成立的证据。produce-new-fact()在一条假设成立后,生成一条新事实,并将该事实添加到事实库中。

其运行过程是这样的,在求解过程生成求解树,且记录了整个求解调用的规则和待触发的函数。当达到某个节点,该节点所登记的规则前提条件均已满足(TRUE),则反向推理结束。然后,推理机开始“正向”推理。“正向”推理从反向推理结束的那个节点起沿求解树反向运行。只要遇到有函数的节点(类型为“DO”),系统便从工具库中调用该函数计算求值,并将求出的值存入动态库中,一直到目标值求出为止。这样系统才算求出一个目标。如果系统只有这样一个目标,系统就结束运行,否则系统继续进行推理求解。

总之,混合推理求解目标值过程是分两步完成的。第一步是探索过程,但没有进行实质性的计算求值(如曲线拟合,求解子波起止点),而是生成求解所需的反向链。第二步在第一步试探成功的基础上,按第一步既定的求解步骤沿求解链的反向进行实质性的数值计算。这样的混合推进机制既满足了时域波形智能识别的要求,又可避免冗余计算,提高求解效率。

参考文献

- 1 田盛丰等,《人工智能原理与应用——专家系统、机器学习,面向对象的方法》,北京理工大学出版社,1993.8.
- 2 查建中,《智能工程》,北京,机械工业出版社,1992.3.
- 3 谔淮扣,《语言处理程序》,北京,清华大学出版社,1993.5.

PRO * C 的研究与应用实现

谢 鹏

(西安石油学院计算机系 710065)

摘要 本文论述了 PRO * C 程序的三个组成部分,并以实际应用为例介绍了 PRO * C 程序的设计方法。

关键词 主变量 定义节 通讯区 科目

1 引言

ORACLE 系统由于其诸多的优点,现已被大量用于信息管理系统。信息管理系统通常要输出大量的报表(如各种日报表、月报表和年报表等)。尽管 ORACLE 提供了 SQL * REPORTWRITER 这种专门用于生成报表的软件,但由于国内外管理体制的差异使 SQL * REPORT WRITER 对生成国内常用报表有很多不便。为此,必须设法通过其它途径产生报表。

ORACLE 系统提供了 PRO * C 工具,使用户可以使用 C 语言对 ORACLE 数据库中的数据进行操作。其中一种操作方法就是在 C 语言程序中嵌入 SQL 语句,这种由 C 语句和 SQL 语句共同构成的程序我们称为 PRO * C 程序。通过 PRO * C 程序可以很方便地实现特定表格的输出。本文对 PRO * C 程序的结构给予论述,并给出具体的应用实例。

2 PRO * C 程序的组成

由于 PRO * C 程序是由 C 语句和 SQL 语句共同构成的程序,因此,PRO * C 程序有其独特的格式要求。

通常,一个 PRO * C 程序由三部分组成,它们分别是:定义节;SQL 通讯区;程序体。而在程序体的开始必须先进行 PRO * C 与 ORACLE 系统的连接。

2.1 定义节

PRO * C 程序是将 SQL 语句嵌入到 C 程序中的,因此称 C 语言为宿主语言。SQL 语句中有时需要使用宿主语言中的变量,例如在 SQL 的 SELECT 语句中,需将查找到的数据通过 INTO 子句送到这些变量中,或者在 WHERE 子句需要使用这些变量。这种变量称为主变量。在 SQL 语句中所使用的主变量都必须在 DECLARE 节中加以说明。

DECLARE 节是以语句

```
EXEC SQL BEGIN DECLARE SECTION;
```

开始,以语句

```
EXEC SQL END DECLARE SECTION;
```

结束的。

SQL 语句中所用的全部主变量均在这两个语句中间加以定义。在 SQL 语句中使用主变量时,在每个

被引用的主变量前必须要加一个冒号“:”,以便识别。主变量在 C 语句中引用时,不能加冒号“:”。

为了便于对变长字符串的处理,ORACLE 系统提供了 VARCHAR 伪类型。

VARCHAR 伪类型变量在 DECLARE 节中定义,然后在程序中可以引用。例如:

```
EXEC SQL BEGIN DECLARE SECTION;
```

```
VARCHAR uid[20],pwd[20];
```

```
EXEC SQL END DECLARE SECTION;
```

这里定义了数组 uid、pwd 是 VARCHAR 伪类型,它的作用相当于定义了两个结构变量:

```
struct{  
    unsigned short int len  
    unsigned char arr[20]  
}uid;
```

和 struct{
 unsigned short int len
 unsigned char arr[20]
}pwd;

VARCHAR 伪类型变量 uid 和 pwd 在使用时是将字符串存放在字符数组 uid.arr(或 pwd.arr)中。需要注意的是,uid.arr(或 pwd.arr)数组中的字符串没有 '\0' 作为字符串结束标志,字符串的实际长度是在 uid.len(pwd.len)中存放着。而在后面调用 printf("%s"),(输出对象)输出字符串时,要求(输出对象)以 '\0' 结尾,这需要程序设计者自己加上去。

2.2 SQL 通讯区

在程序中,常常需要知道 SQL 语句执行的情况,需要知道 SQL 是否执行成功,如果执行不成功是何种原因造成的,程序可以根据不同的执行情况采取不同的措施。为了将 SQL 语句执行情况的信息返回给程序,需要在 PRO * C 程序中定义程序与 ORACLE 系统的通讯区,ORACLE 系统将 SQL 语句的执行的情况送入该通讯区,程序可以从该通讯区得到有关的信息。

SQL 通讯区是通过语句:

```
EXEC SQL INCLUDE SQLCA;
```

来实现的。

2.3 程序体

在程序体的开始 PRO * C 必须与 ORACLE 系统连接起来后,才能够访问 ORACLE 数据库。连接时,用户要提供用户名和口令,系统对用户名和口令进行检查校验。如果用户提供的用户名与口令正确,才能连接成功。

连接是使用 SQL 语言的 CONNECT 语句来完成的。在 PRO * C 中引用 CONNECT 的方法之一是:

```
EXEC SQL CONNECT: <用户名> IDENTIFIED BY: <口令>;
```

在执行这条语句之前必须对主变量置正确的用户名和口令的值,否则无法连接成功。

在前面定义变量的基础上,再假设用户名为“XP”,口令为“XPXP”,则可按如下方式将 PRO * C 与 ORACLE 进行连接:

```
main()
{
    strcpy(uid. arr, "XP");
    uid. len = strlen(uid. arr);
    strcpy(pwd. arr, "XPXP");
    pwd. len = strlen(pwd. arr);
    EXEC SQL CONNECT: uid IDENTIFIED BY: pwd;
    .
    .
    .
}
```

连接成功后,在程序体中可以使用 SQL 语句对 ORACLE 数据库进行查询等操作,再通过 C 语句对查询的结果进行各种加工(比如输出)操作。

3 PRO * C 应用实例

在企业“内部银行”中,每天有一项必须完成的业务就是根据当天发生的原始凭证(或称传票)汇总出“科目平衡表”。当“科目平衡”时,将日报表“科目平衡表”打印出来。

按照财政部规定,科目平衡表有其特殊的格式要求,样表如下:

各科目平衡表

××商厦内部银行				1996 年 03 月 30 日				编号 70			
编号	科目	借方	贷方	编号	科目	借方	贷方				
101	现金	2061494.1	2158329.19	211	其它应付款	1550450	580000				
102	银行存款	1661896.11		201	短期借款	500000	580000				
129	其它应收款	137832	10130	231	待摊费用						
100	商场货款			555	财力费用	72906.4					
				321	本年利润		1741519.42				
				200	商场存款						
	合计	1692763.02			合计		1692763.02				

(样表:数据无实际价值)

本表的内容是根据每日的“原始凭证库”汇总而来的,为打印方便起见,特建立“科目平衡中间表”这样一个临时数据库,其库名为:NHT _ KMPHZJB,意为“科目平衡中间表”,该表的结构为:

```
KMBH1 CHAR(3)      JF2    NUMBER(12,2),
KMMC1  CHAR(20)     DF2    NUMBER(12,2)
JF1    NUMBER(12,2), NIAN   CHAR(2)
DF1    NUMBER(12,2), YUE    CHAR(2)
KMBH2  CHAR(3)      RI     CHAR(2)
```

```
KMMC2  CHAR(20),    BH     NUMBER(3,0)
```

下面给出打印“科目平衡表”的 PRO * C 程序框架,后附有源程序清单。

需要说明的是,NHT _ KMPHZJB 中的字段 JF1,DF1,JF2,DF2 都是数字型的,而在下面的 PRO * C 程序中将与之对应的主变量 jf1,df1,jf2,df2 都定义成 VARCHAR 伪类型,这样做的目的是为输出方便。另外,在 DECLARE 节定义的主变量都比库中的相应字段要长,这是为了加串结束标志“\0”

```
#include <stdio. h>
#include <string. h>
EXEC SQL BEGIN DECLARE SECTION
VARCHAR uid[20],pwd[20];
VARCHAR kmbh1[4],kmmc1[21],kmbh2[4],kmmc2[21];
VARCHAR nian[3],yue[3],ri[3];
VARCHAR jf1[15],df1[15],jf2[15],df2[15];
int bh;
EXEC SQL END DECLARE SECTION;
/* 以上为 DECLARE 节 */
EXEC SQL INCLUDE SQLCA;
/* 本句是定义通讯区 */
/* 下面是程序体 */
main()
{
    FILE * fp;
    strcpy(uid. arr, "XP");
    uid. len = strlen(uid. arr);
    strcpy(pwd. arr, "XPXP");
    pwd. len = strlen(pwd. arr);
    EXEC SQL WHENEVER SQLERROR GOTO sqlerror
    EXEC SQL CONNECT: uid IDENTIFIED BY: pwd;
    .
    .
    .
} /* end main */
```

4 结束语

ORACLE 系统中的 SQL 语言是非过程化的语言,它是专用的数据库操作语言,不足之处是过程控制不易实现。而 C 是功能极强的程序设计语言。因此,在用 ORACLE 开发应用时,正确使用 PRO * C 将会极大的方便开发,并大大提高开发效率。

参考文献

- 1 Tom Portfolio,《Programmer's Guide to the ORACLE Precompilers》,Copyright Oracle Corporation,Printed in the U. S. A
- 2 沈金发等,《关系数据库系统 ORACLE》,清华大学出版社
- 3 卢有杰、吴炜煌编,《C 语言高级程序设计》,清华大学出版社

计算机数值控制中软件分频方法的研究

苗丛毅 邓文新

(齐齐哈尔轻工学院计算机科学与工程系 161006)

摘 要 本文对计算机数值控制中的软件分频进行了研究,给出了扩大分频比及消除输出脉冲不均匀性的方法。

关键词 分频 数值控制 计算机控制

1 引言

分频是计算机数值控制中实现坐标联动的常用方法。硬件分频的优点是速度快,不占用 CPU 的全部时间,并行性好,但分频比必须圆整为整数从而引起累积误差,虽然可以采取一些措施来解决但都较复杂。本文介绍的软件分频方法其分频比可以是分数,消除了圆整化误差,对于规模比较小的数控系统不失为一种较适用的控制形式。

2 软件分频原理

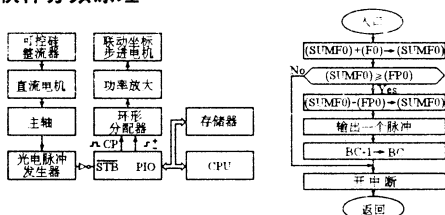


图 1

图 2

图 1 为用软件分频实现一个坐标控制时的框图。设主轴转过 $\Delta\theta$ 角时光电脉冲发生器发出的脉冲数为 M , 对应联动坐标移过 $Y\text{mm}$ 时步进电机需要的脉冲数为 N , 定义分频比 A 为:

$$A = \frac{M}{N} = \frac{(FP_0)}{(F_0)} \quad (1)$$

先将 M 和 N 分别存入内存单元(FP0)和(F0),N 又是控制联动坐标 Y 长度的计数脉冲个数,N 送 BC 寄存器。脉冲发生器每发一个脉冲通过反相后送 PIO 的 $\overline{\text{STB}}$ 端,PIO 向 CPU 申请一次中断,在累加器里执行一次加(F0),其和为(SUMF0),当(SUMF0)小于(FP0)则无脉冲输出中断返回,当(SUMF0)大于或等于(FP0)则输出一个脉冲,同时 BC 减 1,直至 BC=0 为止。中断流程如图 2。

3 扩大分频比的方法

字长为 8 位的 CPU 只能计 0~255 以下的无符号整数,对于大分频比的控制可将分频比 A 分解为:

$$A = \frac{(FP_0)}{(F_0)} + \frac{(FP_1)}{(F_1)} \times \frac{(FP_2)}{(F_2)} \quad (2)$$

三个分数的分子分母分别存入(FP0)、(F0)、(FP1)、(F1)、(FP2)、(F2)六个单元,当(FP0)=(FP1)=(FP2)=255,(F0)=(F1)=(F2)=1时,最大分频比为 $255+255 \times 255=65280$ 。扩大分频比后的中断流程

如图 3 所示。现举例说明如下。

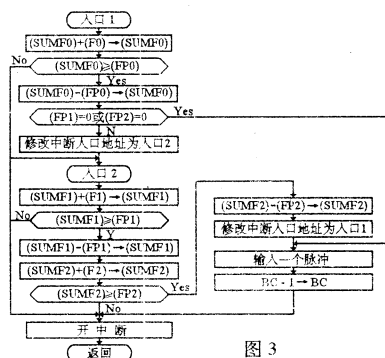


图 3

设 $A=32/9$, 分解为 $A=(4/3)+(4/3) \times (5/3)$ 即 $(FP0)=4, (F0)=3, (FP1)=4, (F1)=3, (FP2)=5, (F2)=3$, 按框图 3 走一遍的结果如表 1 所示。由表 1 可见脉冲发生器发出 32 个主脉冲 I, PIO 输出 9 个脉冲 L, 同时累加器的结果 $(SUMF0)$ 、 $(SUMF1)$ 、 $(SUMF2)$ 都变为 0。从表 1 也可看出输出的脉冲是不均匀的。

表 1

主脉冲序号 M	输出脉冲序号 N	ΔM	累加寄存器(SUMF0)、(SUMF1)、(SUMF2)中的和或余数
1			$(SUMF0) = 3 + 0 = 3;$
2			$(SUMF0) = 3 + 3 - 4 = \underline{2};$
3			$(SUMF1) = 3 + 0 = 3;$
4			$(SUMF1) = 3 + 3 - 4 = \underline{2}; (SUMF2) = 3 + 0 = 3;$
5	1	5	$(SUMF1) = 2 + 3 - 4 = \underline{1}; (SUMF2) = 3 + 3 - 5 = 1;$
6			$(SUMF0) = 2 + 3 - 4 = \underline{1};$
7			$(SUMF1) = 1 + 3 - 4 = \underline{0}; (SUMF2) = 1 + 3 = 4;$
8			$(SUMF1) = 0 + 3 = 3;$
9	2	4	$(SUMF1) = 3 + 3 - 4 = \underline{2}; (SUMF2) = 4 + 3 - 5 = 2;$
10			$(SUMF0) = 1 + 3 - 4 = \underline{0};$
11	3	2	$(SUMF1) = 2 + 3 - 4 = \underline{1}; (SUMF2) = 2 + 3 - 5 = 0;$
12			$(SUMF0) = 0 + 3 = 3;$
13			$(SUMF0) = 3 + 3 - 4 = \underline{2};$
14			$(SUMF1) = 1 + 3 - 4 = \underline{0}; (SUMF2) = 0 + 3 = 3;$
15			$(SUMF1) = 0 + 3 = 3;$

基于神经网络的储层敏感性评价专家系统

徐英卓

(西安石油学院钻井信息中心 西安 710065)

摘 要 本文介绍了利用神经网络技术对储层敏感性评价专家系统的设计及实现。该专家系统在知识获取和推理控制策略方面优越于传统的专家系统。

关键词 神经网络 专家系统 敏感性评价 BP 算法

1 引言

储层敏感性是指外来流体与储层内敏感矿物之间发生物理化学反应,造成储层渗透率降低的现象。储层敏感性评价是石油工业中储层保护工程的一个重要环节,它通过岩石学分析、测井、试井等数据资料确定出储层敏感特性参数,从而预测未取得岩样的地层中潜在敏感性类型及程度。评价的结果可指导施工方案的

优化设计,配合各施工过程地层伤害后的伤害类型及原因的诊断,为选择伤害的处理措施提供依据。由于影响储层敏感性的因素众多,机理复杂,并且从岩石学分析、岩石物性分析等方面得到的数据往往具有模糊性、随机性等特点,因此,对敏感性的评价采用传统的数学方法来建立精确的数学模型是非常困难的。鉴于神经网络具有如下独特的优点:(1)能完成复杂的输入与输出之间的非线性映射;(2)处理信息的高度并行性;(3)信息处理与存储合二为一;(4)具有很强的自组织、自学习能力。(5)具有很强的鲁棒性和容错性。因此,我们同塔里木油田合作,利用神经网络技术开发了储层敏

收稿日期:1996 年 7 月 15 日

* 本文工作得到国家 863 计划项目(863-306-04-10-1)的资助。

16	4	5	(SUMF1)=3+3-4=2;(SUMF2)=3+3-5=1;
17			(SUMF0)=2+3-4=1;
18			(SUMF1)=2+3-4=1;(SUMF2)=1+3=4;
19	5	3	(SUMF1)=1+3-4=0;(SUMF2)=4+3-5=2;
20			(SUMF0)=1+3-4=0;
21			(SUMF1)=0+3=3;
22	6	3	(SUMF1)=3+3-4=2;(SUMF2)=2+3-5=0;
23			(SUMF0)=0+3=3;
24			(SUMF0)=3+3-4=2;
25			(SUMF1)=2+3-4=1;(SUMF2)=0+3=3;
26	7	4	(SUMF1)=1+3-4=0;(SUMF2)=3+3-5=1;
27			(SUMF0)=2+3-4=1;
28			(SUMF1)=0+3=3;
29			(SUMF1)=3+3-4=2;(SUMF2)=1+3=4;
30	8	4	(SUMF1)=2+3-4=1;(SUMF2)=4+3-5=2;
31			(SUMF0)=1+3-4=0;
32	9	2	(SUMF1)=1+3-4=0;(SUMF2)=2+3-5=0;

注:△M 为连续两个输出脉冲 L 所对应的主脉冲 I 序号 M 的间隔数

4 输出脉冲不均匀性的分析及处理方法

从表 1 可以看出,最大△M 发生在累加器(SUMF0)、(SUMF1)、(SUMF2)的余数最小时,最小△M 发生在三个累加器的余数最大时。由此可见,造

成△M 不均匀的原因就是每次输出脉冲 L 时,三个累加器中的残留余数不一致。存在残留余数的原因是分解产生的三个分频比均为分数,而我们正是利用这个残留余数实现了累积误差的消除。进一步分析,由于(FP1)/(F1)和(FP2)/(F2)是相乘的关系,(SUMF1)和(SUMF2)中的残留余数对△M 不均匀性影响最大,若二者为零△M 的不均匀性将得到极大的改善。只有(FP1)/(F1)和(FP2)/(F2)为整数即(F1)和(F2)为 1,(SUMF1)和(SUMF2)中的残留余数才能为零。而任何一个假分数总可以分解为一个分数和一个整数相加的形式,所以(FP0)/(F0)为分数而(FP1)/(F1)和(FP2)/(F2)为整数这种分解是可能的,此时△M 不均匀性最大时为 1。

5 实验的结论

为了验证我们分析脉冲不均匀性的原因和措施,我们在 TP801 上作了实验。将一个脉冲源(如图 4 所示)的输出经反相器接 PIO 的 STB 端,用三个发光二极管 L1、L2、L3 来显示主脉冲 I、输出驱动脉冲 L 和方向电平信号。实验结果表明理论分析完全正确。

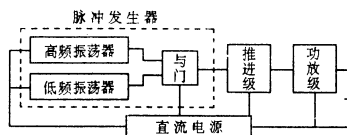


图 4

参考文献

- 1 李国伟等·微机控制纤维缠绕机分频误差分析及处理·哈尔滨工业大学学报,1988 年 4 月。

感性评价专家系统。该系统为钻井技术人员提供一个专家水平上的咨询信息,以使专家的知识成为人类共同的财富。

2 系统的基本结构

储层敏感性评价专家系统采用神经网络技术进行设计。它主要由权值学习模块、敏感性评价模块、结果解释模块、权值知识库和数据库组成。其具体结构如图1所示。

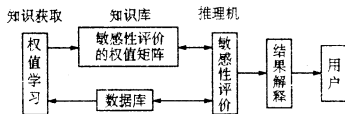


图1 系统基本结构框图

(1)权值学习模块 它完成系统的知识获取功能,包括提出神经网络的结构(包括网络层数、输入、输出及隐单元个数),利用BP算法,通过领域专家提供的大量实验数据和范例对网络进行训练,完成敏感性评价权值矩阵的生成,从而达到知识获取的目的。

(2)敏感性评价模块 该模块是系统的推理机,推理机的推理过程实际上是数值计算过程,通过数值计算完成敏感性评价。

(3)结果解释模块 该模块通过知识库和数据库对推理过程进行记录,然后根据这些记录对求解过程作出说明。

(4)权值知识库 它存储在样本训练过程中生成的权值矩阵,它是储层敏感性评价知识的隐式表示。

(5)综合数据库 存储与领域知识相关的数据。

3 储层敏感性评价知识的组织

3.1 评价知识的分类

通过与领域专家反复接触,我们了解到储层敏感性类型是由敏感矿物的类型决定的,共分五种,即为速敏、水敏、盐敏、碱敏、酸敏;而每一种敏感性类型的敏感程度又根据其矿物含量、存在形态、储层物性、外来流体矿化度的大小等各种因素划分为五个等级:强、中一强、中、中一弱、弱。因此,我们可将敏感性评价这种分类等级关系抽象出它的层次分类模型如图2所示,其前后层次间具有继承关系,即上一层的分类结果是下一层等级划分的必要条件。



图2 敏感性评价的层次分类模型

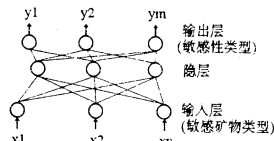


图3 敏感性评价第一级神经网络结构

3.2 分类模型的神经网络映射

神经网络不仅有输入层和输出层,而且有隐层(可以是一层或多层)。输入信号从输入层向前传播到隐层,经作用函数作用后,再把隐层结点的输出信号传播

到输出层,最后给出结果。结点的作用函数通常选取S型函数,如取为: $f(x) = 1/(1 + e^{-x})$ 。根据上面的分类模型我们可实现它与神经网络的映射,映射时总体上采用多级网络组织形式,将知识进行分类组织。我们先将与敏感性类型判别相关的知识映射为一个神经网络,而将储层敏感矿物类型作为这级网络的输入,敏感性类型作为这级网络的输出。然后将与敏感性程度等级划分相关的知识映射为另一个神经网络,在此网络中,将与敏感性类型相对应的敏感矿物含量、存在形态、储层物性及外来液体矿化度的大小等因素作为网络的输入,将敏感性程度作为网络的输出。上下级神经网络间具有主从式逻辑关系,即根据上一级网络的推理结果来确定下一级推理网络的结构,进一步向下推理。按照这种多级网络组织结构,推理机则分级进行推理,最终导出结果。敏感性评价第一级(第二级类似,从略)神经网络结构如图3所示。

4 神经网络的学习

神经网络的学习过程对应于传统专家系统的知识库的建立,即知识获取问题。神经网络的知识获取不需要由知识工程师来整理、总结、消化领域专家的知识,只需用领域专家解决问题的实例或范例来训练网络,使其在同样输入条件下能够获得与专家给出的解答尽可能相同的输出。

4.1 网络的学习算法

本系统的神经网络学习采用改进的BP学习算法。图4为系统学习过程原理图。BP算法为误差驱动的梯度算法,其误差函数定义为:

$$E = 1/2 \sum_k (v_k - T_k)^2$$

式中 V_k 为输出层神经元 k 的实际输出; T_k 为期望输出。其基本的权值修正公式为:

$$\Delta w_{ji}(n+1) = -\eta \partial E / \partial w_{ji}(n)$$

式中 n 为迭代次数, η 为学习速率。从数学上看, BP 算法可归结为一非线性的梯度优化问题,因此不可避免地存在局部最小问题,且其收敛速度很慢,通常需要上千次或更多。为此,本系统在现行算法的基础上,通过增加动量项,对权值修正公式进行了改进,改进后的公式为: $\Delta w_{ji}(n+1) = -\eta \partial E / \partial w_{ji}(n) + \alpha \Delta w_{ji}(n)$

式中 η 意义同上, α 为动量因子。上式表示在当前权的改变上加入前次权值的变化量,以平滑梯度方向的剧烈变化。

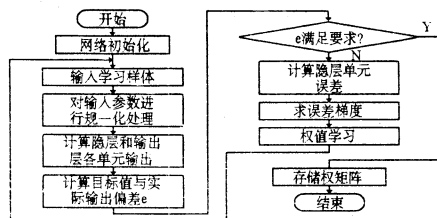


图4 系统学习过程原理图

Oracle CDE2 中对 Windows DLL 的调用

郭 伟 石冰心

(华中理工大学计算中心 武汉 430074)

摘 要 本文讨论了在 Oracle Forms 4.5 的例程和触发器设计中如何用 PL/SQL 语言直接调用 Windows 下动态连接库函数的方法,根据这个方法,可以在不退出 Oracle Forms 的情况下,运行动态连接库函数。

关键词 Oracle CDE2 外部函数 动态连接库 PL/SQL 界面

1 前言

Oracle CDE2 是 Oracle 公司推出的第二个基于 Windows 界面的高端开发工具,它主要包括三个工具包,分别是:Oracle Forms 4.5, Oracle Reports 2.5, Oracle Graphics 2.5。这三个开发工具的程序设计语言是 PL/SQL 语言,用 PL/SQL 语言编程时,可以使用开发环境下提供的许多内部例程(Built-in Procedure)和函数(Function)实现对数据库的操作及界面的许多

收稿日期:1996 年 5 月 5 日

4.2 网络参数的选择

(1)学习速率 η 与动量因子 α 和 α 的选取原则是要有助于加快网络收敛而又不引起振荡。一般来说, η 越大越有利于收敛,但到学习后期会引起振荡效应。而 α 的适当选取有益于抑制振荡,使网络能较快地跳出局部极小,加快收敛过程。一般在学习初期 α 宜选较小的值,在经过一定次数的学习误差显著减少后,增大 α 可有效地抑制振荡,但 α 不宜过大,否则影响收敛速度。鉴于以上原因,我们在每次迭代后根据系统误差的大小对 η 和 α 进行自适应调整。当总误差较大时, η 和 α 取较小值;当总误差较小时, η 和 α 取较大值。调整算法为:取 $\eta = \eta_0 e^{-E_n}$, $\alpha = \alpha_0 e^{-E_n}$, 权值修正公式为: $\Delta w_{ji}(n+1) = -\eta_0 e^{-E_n} / \partial E / \partial w_{ji}(n) + \alpha_0 e^{-E_n} \Delta w_{ji}(n)$ 这里 η_0, α_0 为大于 0 小于 1 的数。

(2)隐层神经元数 隐层神经元个数的选择是一件非常重要而困难的事,其困难在于目前尚无很好的解析式作为依据。隐单元数(隐层神经元数)太少,网络可能训练不出来,或不强壮。适当增加隐单元数,可增加非线性函数映射的复杂性,促使网络形成所需要的分类边界。但隐单元数超过一定限度时,收敛速度大幅度下降,使训练学习时间太长。为了取得合适的隐单元数,本系统采用变隐单元数方法。即将隐单元数设为一个动态值,先根据问题的规模与复杂性预定一个较大的隐单元数,在学习过程中,再根据学习情况对过小权值的隐单元删去,然后重新学习,以确定一个新的权值,最后得到一个大小合适的自适应网络。

5 并行推理与多级控制策略的实现

操作处理。

动态连接库(Dynamic Link Library-DLL)是 Windows 的一种函数库形式,由于它具有共享及动态性质,所以对于利用动态连接库编写程序可以节省运行时内存的开销,提高程序的运行效率。

一般情况下,在 Oracle CDE2 环境下进行应用开发,其内部例程和函数已经足够适应许多的功能需求,但有时对于一些客户的特殊需求,就需要在 Windows 下用 3GL 语言进行编程设计动态连接库或直接调用 Windows 动态连接库,而 Windows 的动态连接库不能

神经网络的正向传播应用构成了其推理过程,它实际上是一种并行运算机制。依据敏感性评价层次分类模型,多次调用网络的权值矩阵,根据求解目标,逐级进行推理控制,实现了本专家系统的推理控制策略。其实现过程如图 5 所示。

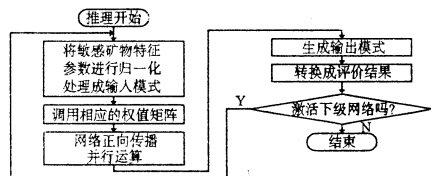


图 5 推理控制策略实现框图

6 结束语

本文提出的基于神经网络的储层敏感性评价专家系统已用 Borlandc++ 4.0 语言在 486 微机实现,并已投入油田现场应用,应用效果良好。本系统具有如下特点:

(1)实现了敏感性评价分类模型的多级神经网络映射。

(2)采用了改进的 BP 学习算法,能有效地缩短网络的学习时间。

(3)实现了常规符号推理所难以实现的并行推理机制,提高了推理速度。

参考文献

- 1 施洪宝. 神经网络及其应用. 西安:西安交通大学出版社, 1994.
- 2 钟义信等. 智能理论技术—人工智能与神经网络. 北京:人民邮电出版社, 1992.
- 3 胡守仁. 神经网络应用技术. 长沙:国防科技大学出版社, 1992.

直接在 Oracle CDE2 中运行,那么如何在 Oracle CDE2 的例程和触发器中调用 Windows 下的动态连接库 DLL,本文就这个问题进行一些探讨。

为了叙述方便,本文针对在 Oracle Forms 4.5 的例程和触发器中调用 Windows 动态连接库进行讨论,动态连接库的编程语言也假定为 C 语言或 C++ 语言。

2 Oracle Forms 4.5 中的外部函数

在 Oracle 开发工具中,外部函数(Foreign Function)是指用 3GL 语言(如 C, C++)编写的能够增强 Oracle Forms 应用的子程序。所以,在 Windows 下用 C、C++ 等语言编写的动态连接库 DLL 也被 Oracle Forms 看作是外部函数。由于 Oracle CDE2 工具环境有独立的 PL/SQL 语言,不支持用 C、C++ 语言编写的程序直接运行,所以动态连接库中的函数不能在 Oracle Forms 中运行。要在 Oracle Forms 中运行动态连接库的函数(或称外部函数),有两种方法可行:一种是通过用户出口(User Exit)界面;另一种是通过 PL/SQL 界面。前者在运行外部函数时要重新启动 Forms Runtime 环境,系统资源消耗比较大,运行速度相对较慢;而后者是在当前 Forms Runtime 环境下直接调用运行外部函数,故消耗资源少,速度快。后者的调用方法是 Oracle CDE2 工具才支持的,第一代的 CDE 工具不具备这样的功能。本文就通过 PL/SQL 界面调用动态连接库函数的方法进行讨论。

3 Oracle Forms 4.5 中外部函数调用机制

在 Oracle Forms 中调用外部函数机制就是将处理控制暂时交由外部函数,待外部函数执行完成以后,再返回处理控制给 Oracle Forms。由于外部函数的数据格式、数据类型等对 Oracle Forms 的 PL/SQL 来讲是“不认识”,为了实现对外部函数的调用,实现 Oracle Forms 与外部函数的通信,就必须创建一个 PL/SQL 界面,通过该界面 PL/SQL 才能与动态连接库中的函数进行通信。因此,为了实现在 Oracle Forms 的例程和触发器中对动态连接库的调用,必须做两件事:(1)用 C 或 C++ 编写一个动态连接库;(2)在 Oracle Forms 中创建一个 PL/SQL 界面。本文假定动态连接库已经创建完成,专门讨论如何在 Oracle Forms 中创建 PL/SQL 界面,并通过该界面实现对 DLL 函数的调用。

4 Oracle Forms 中 PL/SQL 界面的设计

PL/SQL 界面是在 Oracle Forms 中用 PL/SQL 语言编写的。一般是在 Oracle Forms 中设计一个 PL/SQL 程序包(PL/SQL Package)。在这个 PL/SQL 包中,设计以下内容:

- 初始化动态连接库的每一个函数;
- 设计 PL/SQL 界面;
- 创建 PL/SQL 函数。

Oracle 为申请调用 C 或 C++ 语言编写的外部函数,提供了一个 ORA_FFI 包,在该包中有初始化外部函数和创建 PL/SQL 界面所需的支持函数和数据类型定义,关于 ORA_FFI 包的细节请参考有关 Oracle 资料。

4.1 初始化动态连接库的函数

Windows 操作系统是一个多任务操作系统,Windows 给每一个运行的函数或对象分配一个实例句柄。在 Windows 下程序调用函数或对象运行时,首先要获得该函数或对象的句柄,通过该句柄才能发送或接收参数。因此,PL/SQL 界面设计时初始化的目的是为了得到动态连接库和函数的句柄,在 PL/SQL 环境下重新定义参数的数据类型等,初始化动态连接库的函数至少应有以下步骤:

● 装载动态连接库

用函数 ORA_FFI.LOAD_LIBRARY(dirname VARCHAR2, libname VARCHAR2)

dirname 是动态连接库放置的路径,libname 是动态连接库的文件名。该函数返回一个库句柄,类型为 libHandleType。

● 注册动态连接库中需使用的每个函数

用函数 ORA_FFI.REGISTER_FUNCTION(libHandle libHandleType, funcname VARCHAR2, callstd PLS_INTEGER, =C_STD)

libHandle 是前面装载动态连接库时得到的库句柄;funcname 是需使用的函数名;callstd 表示函数调用标准,缺省 C_STD 表示是 C 调用标准。该函数返回一个函数句柄,类型为 funcHandleType。

● 注册动态连接库的参数

用函数 ORA_FFI.REGISTER_PARAMETER(funcHandle funcHandleType, cargtype PLS_INTEGER);

funcHandle 是前面注册函数时得到的函数句柄;cargtype 是 C 程序函数中参数在 PL/SQL 界面中的表示。如 C_CHAR 表示 char, C_FLOAT 表示 float, C_CHAR_PTR 表示 char * 等等。该函数没有返回值。

需要注意的是,用该函数注册参数时,每次只能注册一个参数。

● 注册动态连接库中的返回值

除非 C 程序函数使用 void 类型,否则都要注册返回值。用函数

ORA_FFI.REGISTER_RETURN(funcHandle funcHandleType, creturntype PLS_INTEGER);

funcHandle 是注册函数得到的函数句柄;creturntype 表示 C 程序函数返回类型在 PL/SQL 界面中的表示。

对一个外部函数的初始化至少应有以上四个步

骤。除此之外有时可能还要做更多的初始化工作,如用 `ORA_FFI.FIND_FUNCTION(...)` 得到函数的句柄等等,这里不再详述。

4.2 PL/SQL 界面设计

PL/SQL 界面设计的目的是针对 C 程序的函数原型定义,在 PL/SQL 环境中创建一个类似的函数定义的界面,该界面就是为了在 Oracle Forms 中也能调用该函数。设 C 程序中函数原型如下:

```
float tax(float amt,float rate)
```

则 PL/SQL 界面的定义如下:

```
FUNCTION ffi_tax (funcHandle ORA_FFI.FUN-  
FUNCHANDLETYPE,x,NUMBER,y NUMBER)
```

```
RETURN NUMBER;
```

```
PRAGMA interface(C,ffi_tax,11265);
```

在 PL/SQL 界面里对数据类型的定义应依照 PL/SQL 语言的规定来确定,如针对 float 数据类型,PL/SQL 里对应为 NUMBER;char 数据类型,PL/SQL 里对应为 VARCHAR2 等等。另外,上述定义语句中 PRAGMA... 行是固定的命令行,该命令行的作用是发送一个控制给内存地址段,通过该地址段中的 Oracle 程序实现与动态连接库的通信。该行中 C 表示外部函数用 C 语言编写,ffi_tax 处为 PL/SQL 界面定义中的函数名。

如果动态连接库中函数为 void 类型,则 PL/SQL 界面可以定义成例程 (PROCEDURE),那么定义中就没有返回值。

4.3 创建 PL/SQL 函数

在初始化了外部函数和创建了 PL/SQL 界面后,接下来就是设计 PL/SQL 函数。这是因为 PL/SQL 程序中只能调用 PL/SQL 函数。通过 PL/SQL 函数,再通过 PL/SQL 界面,实现对动态连接库函数的调用。Oracle Forms 中对动态连接库函数调用过程可描述如图 1 所示:

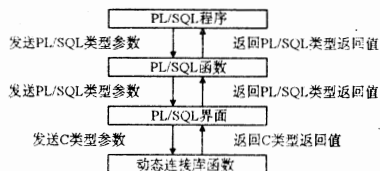


图 1 Oracle Forms 对 DLL 函数调用示意图

如上面提到的 tax 函数,可设计 PL/SQL 函数如下:

```
FUNCTION tax(x NUMBER,y NUMBER)
```

```
RETURN NUMBER IS
```

```
BEGIN
```

```
RETURN(ffi_tax(ffi_handle,x,y));
```

```
END;
```

5 应用举例

为了使读者了解外部函数调用设计的全过程,下面举一个简单例子予以说明。

设已经提供了一个 C 程序动态连接库,动态连接库的名及放置目录如下:C:\libs\mathlib.dll;该动态连接库是一个数学函数库,有一个计算幂指数的函数原型如下:

```
int to_power(int x,int y)
```

为了在 Oracle Forms 中调用该函数,PL/SQL 包设计如下:

(1)PL/SQL 包说明

```
PACKAGE mathlib IS
```

```
FUNCTION to_power(x PLS_INTEGER,y
```

```
PLS_INTEGER)
```

```
RETURN PLS_INTEGER;
```

```
END;
```

(2)PL/SQL 包体程序

```
PACKAGE BODY mathlib IS
```

```
/* 声明动态连接库和函数句柄 */
```

```
mathlib_lhandle ORA_FFI.LIBHANDLETYPE;
```

```
to_power_fhandle ORA_FFI.FUNHANDLE-  
TYPE;
```

```
/* 创建一个 PL/SQL 函数界面 */
```

```
FUNCTION ff_to_power (fhandle ORA_FFI.FUN-  
CHANDLETYPE,x
```

```
PLS_INTEAGE,y
```

```
PLS_INTEAGE)
```

```
RETURN PLS_INTEAGE;
```

```
PRAGMA interface(C,ff_to_power,11265);
```

```
/* 创建 PL/SQL 函数,对 PL/SQL 界面发送参数及接  
收界面的返回值 */
```

```
FUNCTION to_power(xPLS_INTEAGE,y PLS_IN-  
TEAGE)
```

```
RETURN PLS_INTEAGE IS
```

```
BEGIN
```

```
RETURN(ff_to_power(to_power_fhandle,x,y));
```

```
END;
```

```
BEGIN/* 包体,初始化外部函数 */
```

```
mathlib_lhandle:=ORA_FFI.LOAD_LIBRARY('C:  
\libs\','mathlib.dll');
```

```
to_power_fhandle:=ORA_FFI.REGISTER_FUNC-  
'TION(mathlib_lhandle,'to_power');
```

```
ORA_FFI.REGISTER_PARAMETER(to_  
power_fhandle,ORA_FFI.C_INT);
```

```
ORA_FFI.REGISTER_PARAMETER(to_  
power_fhandle,ORA_FFI.C_INT);
```

```
ORA_FFI.REGISTER_RETURN(to_  
power_fhandle,ORA_FFI.C_INT);
```

```
END;
```

6 结束语

综上所述,为了在 Oracle Forms 中调用 Windows 下的动态连接库,就必须设计一个 PL/SQL 界面,PL/SQL 界面设计中包含了初始化外部函数、PL/SQL 界面设计、PL/SQL 函数设计三个步骤。PL/SQL

Oracle 协同开发环境——CDE2

杨 勇 周建国 曹丹星

(华中理工大学计算中心 武汉 430074)

摘 要 当计算机协同操作越来越受到广泛关注的时刻,Oracle CDE2 以其图形开发界面给信息系统的开发人员和使用者以全新的面貌和感受。CDE2 是 Windows 环境下客户端的开发工具,它开发出的应用程序可以和服务端端的数据库协同工作,构成 Client/Server 工作方式。在协同开发环境中,应用开发人员能高效率地创建、生成信息系统;使用人员能对开发出的系统进行方便、灵活的操作。

关键词 协同开发环境 信息系统 数据库 应用开发工具

1 引言

CDE2 是 Oracle 公司近期推出的一套最新前端开发工具,该产品是在 Windows 环境下,基于 Oracle 最新数据库 Oracle V7 版的协同服务器技术而研制开发的,以全路径透明和结点自治的方式访问存储在一台或多台服务器上的数据。

Oracle 协同开发环境即 Oracle CDE (Cooperative Development Environment) 是新一代集成的软件生命周期开发环境,用以实现高生产率、大型事务处理及客户机/服务器 (Client/Server) 结构的应用系统。CDE 以强有力的可移植性,支持多种数据源、多种图形用户界面及多媒体、多民族语言、计算机辅助工程 (CASE) 等协同应用系统。在此环境中的工具可以集成使用,以期达到充分利用、共享信息系统资源。从系统的策划、分析、设计、实现到系统运行维护,CDE 都能使系统不同阶段的用户协同工作,整合系统。

在 CDE 中,信息系统开发人员能与最终用户协同工作,采用先进的计算机辅助工程 (CASE) 中的 Oracle Designer 来分析业务需求,并设计完整的解决方案。在系统分析及模型化的任一阶段,开发人员可以直接抽取业务流程说明和信息模型,生成复杂的应用系统。CDE 能把系统开发人员的开发经验和技巧与最终用户对实际业务的丰富知识结合起来,快速生成有效

的应用系统。

由于 Oracle 系统能在 80 多种类型的大、中、小型计算机和微型计算机上运行,并能支持 20 余种网络通讯协议,甚至在异种数据源之间具有分布性、开放性和可移植性的特点。因此,Oracle CDE 生成的所有信息应用系统,适合于能运行在所有最终用户平台上的各种最终环境,能与不同的用户界面相适应,能支持不同的网络环境。CDE 甚至能与所有重要的计算机辅助工程 (CASE) 工具及数据库系统一起工作。因而,应用系统能运行在所有平台上,存取网络环境中任一节点的数据。

2 协同开发环境 (CDE) 工具

Oracle CDE 是一族产品,可以细分为三个类: CDE2 CASE, CDE2 工具和 CDE2 数据存取。该产品提供了一套基于用户业务需求的全面解决方案。本文侧重说明 CDE2 工具部分的内容。

CDE 2 工具部分包括三个集成化了的应用开发工具: Oracle Forms 4.5; Oracle Reports 2.5; Oracle Graphics 2.5。这些产品通过使用集成化的和可重用的成份,提供了一种面向对象的开发环境,这一环境将强有力的缺省性质,说明性的非过程化的方法和易于重用的应用成份组合为一个整体。缺省性质能确保对开发过程的快速启动。说明性方法将缺省特性扩展到那些可能需要广泛地予以重用的功能上。当需要某些特性处理要求时,应对处理逻辑以一种一般性的且可重用的方式予以定义,以便这种处理逻辑能被所有开发

收稿日期:1996 年 4 月 26 日

界面的建立是根据所需要调用的函数原型来设计的,一旦 PL/SQL 界面创建正确,在 Oracle Forms 中使用动态连接库函数也就象在 C 程序中调用函数一样方便。如上例要想得到 2^4 的值,则在 Forms 的例程或触发器中用以下表达式 `x:=mathlib.to_power(2,4)` 即可。

值得指出的是,Windows 下已有的许多动态函数库,都可以用上述提供的方法调用。作者在 Oracle Forms 开发设计中,为了将图形文件读入 Image 项中,用创建 PL/SQL 界面的方法调用 Windows 的 `GetOpenFileName(&ofn)` 函数,打开“Open File”窗口,从窗口中可以选择任意磁盘目录下的图形文件,返回

给 Oracle Forms。对于其它应用,如需要修改 Windows 下的文本文件内容,也需要调用动态连接库函数来实现。总之,通过 PL/SQL 界面调用动态连接库函数的方法应用领域十分广泛,设计人员掌握这种方法,可以设计出各种功能的应用。

参考文献

- 1 《Oracle Forms Developer's Guide》,Release 4.5 Part NO. A23403-1
- 2 《Oracle Forms Advanced Techniques Manual》,Release 4.5 Part NO. A23404-1
- 3 《Oracle Forms Reference Manual》,Release 4.5 Part NO. A23405-1

者和应用所共享。在 CDE 2 的应用开发工具中,表格 (Form) 与表格 (Form) 之间,可用内部例程 Call_Form; Open_Form; New_Form 的任一例程来相互调用或置换。而在表格 (Form) 环境中,需调用运行其它 CDE2 工具时,如:报告 (Report), 图形 (Graphics) 联机文档 (Book), 可使用 Run_Product 内部例程。也可用 OG.Open 程序包调用 Graphics 工具中生成的图表。

2.1 集成化的工具集

CDE2 的所有工具均共享同一个基本技术:使用同一个适配的用户接口 (以便具有平台可移植性);使用同一个程序设计语言 (PL/SQL);使用同一个布局编辑器和同一个信息存储体 (Oracle 7)。无论是建立一个表格 (Form), 一个菜单 (Menu), 一个程序链接库 (Library), 一份报告 (Report) 或一个图形模块 (Graphics), 这些全部使用同一个设计接口, 该接口包括以下几个关键部分:

2.1.1 对象导航器 (Object Navigator)

对象导航器为应用提供了一种结构化的表示,借此工具,开发者对某项应用的各个成份一目了然。对象导航器是促进对软件重用的主要工具。通过它,很容易在不同的应用间,不同的库表间和不同的 Oracle 服务器间,或是在某项应用的内部,拖动和增加应用对象。在此环境中,还可利用窗口菜单控制条的搜索和替换功能,快速查找对象和程序代码,进行有效维护。对象导航器的结构与所含内容如图 1 所指部分。

2.1.2 属性调配器 (Properties)



图 1 对象导航器与属性调配器

属性指的是表格 (Form) 中的每个项和对象所具有的特性,属性决定了在给定环境下对象的外观和行。对于设计者而言,必须确定应用于每一对象的是哪些属性,它们归于哪一类以及哪些属性能满足自己应用程序的特殊需要。属性调配器让开发人员能够对当前所选择的一组对象的一类属性作检查和修改。多个被选对象的这些属性,可被视作某些属性的并或交。这样的机制下,可使一个操作改变多个对象的多项属性。当创建一个新对象时,它的属性自动设置成缺省值。通过设置属性,能很快定义应用的外观和基本功能。属性调配器是一个独立的窗口,它显示一个对象当前的特

性,许多对象的属性还可以在运行时用 Set_... 内部例程设置,用 Get_... 内部例程来获取对象的属性值。属性调配器所包含的内容请参照图 1 所指部分。

2.1.3 Oracle 过程编辑器

PL/SQL 是一种强有力的和可靠的编程语言,既用在 CDE 2 工具集里去定义应用逻辑,又用于 Oracle 7 数据库中作服务器一端的程序设计。过程编辑器正是为 PL/SQL 实施编辑、解释和排错的环境。因此, CDE2 工具集提供了一种统一的 Client/Server 编辑程序,以实现应用逻辑分解在客户机和服务器之间。使用 PL/SQL 编写的存储过程,一旦编译并存入数据字典,它即成为一个数据库对象,可被与该数据库相联的任意多个应用所引用。在编辑器中,可进行编译,检查 PL/SQL 语句的语法错误。Oracle 过程编辑器如图 2 所示。

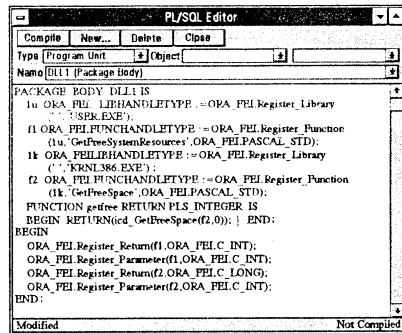


图 2 PL/SQL 编辑器

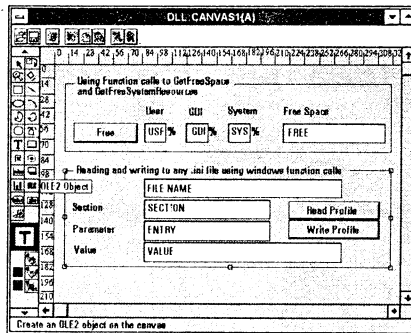


图 3 布局编辑器

2.1.4 布局编辑器 (Canvas)

在对象导航器中可创建生成一个和 / 或多个工作区,每个工作区都有一个被称为“视口”的属性,工作区中处在视口内的部分区域称为“视图”。布局编辑器可使开发人员在视图上布署一些交互项和正文图形版对象,这些项可在运行时的一个窗口上被显示。在 CDE 2 中,有四种类型的视图可被选用,它们分别为:内容视图 (Content), 堆栈视图 (Stacked), 垂直工具条视图 (Vertical Toolbar), 水平工具条视图 (Horizontal Toolbar)。在运行时,一个窗口和分配在其上的工作区视图的显示是可导航和程序控制相结合进行处理的,一个窗口中同一时刻只能有一个内容视图被显示。然而,一个堆栈视图、垂直工具条视图、水平工具条视图

线性方程的两个算法

宁正元 王秀丽

(福建林学院计算机科学系 南平 353001)

摘要 本文介绍线性方程的列初等变换求解算法和一种新的行初等变换求解算法。这两个算法的共同特点是,都使线性不定方程和线性确定方程的求解过程同一化,都可用于对任意线性方程或方程组的求解。尤其是列初等变换求解算法的研究成功,用事实纠正了在数学界长期认为的“只用列初等变换或用行、列初等变换不能解线性方程组”^[1]的错误结论。

关键词 线性方程 系数矩阵 初等变换 求解算法

1 几个概念

在介绍求解算法之前,我们先简要介绍与之相关的几个重要的概念——最简准下三角阵,最简准上三角阵,准标准形和与之相关的几个数学定理,更详细的理论与定理证明可参看文[2]和文[3]的有关内容。

定义 1 对任意 (m,n) —矩阵 A 从首行首列起反

复做如下三种列初等变换所得到的矩阵 Q 称之为 A 的最简准下三角阵:

(1)找出当前行中从当前列起某个非零元素所在的列与当前列交换,若无这样的列,改变当前行为下一行另找;

(2)用适当的数分别乘当前列后加到其它各列,使当前行中除当前列上的元素外其它元素都为零;

(3)用一个数乘当前列,使当前行中唯一的非零元

收稿日期:1996年10月15日

可以同一个内容视图和任意多个其它的堆栈视图同时显示。CDE 2 的所有工具均使用同一个所见即所得式(WYSIWIS 即 What You See Is What I See)的图形编辑程序,以浏览和编辑应用的布局,包括屏幕、报告和图形对象。该编辑器与众多 Windows 环境下的应用编辑软件类似(如:Foxpro 屏幕生成器,Powerbuilder),提供了一系列按钮,用于对可视对象作描绘、缩放、格式化和定位处理。布局编辑器如图 3 所示。以上描述过的各种成份协同工作,构成了一个相容的和强有力的开发环境。对象导航器,布局编辑器和属性调配器全都是自动异步工作的。从而可使开发人员以对自己最为有效的方式观察和编辑对象、程序代码和定义各相关属性。

2.2 应用开发工具

CDE 2 是客户端的开发工具,它必须与数据库服务器配合起来构成 Client/Server 体系结构才能工作。

2.2.1 Oracle Forms 4.5

这是基于表格(Form)的应用快速开发工具。Form 是指表中的数据按一定的形式布局以便逐屏显示处理。开发者可通过菜单、屏幕确省定义和综合使用按钮(Push Button)、广播组(Radio Groups)、值表(lists)、框线(boxes)、形象标志、图象(Image)以及报警标志(Alert)等完成应用开发,通过已建立的格式对数据库进行简捷直观的查询、插入、修改和删除等操作,是支持原型法应用开发的强有力工具,其触发器(Trigger)机制为维护数据完整性、数据的检验等提供了重要保障。

2.2.2 Oracle Reports 2.5

这是数据和正文格式化报告处理工具,用它可建立适应面广的各种格式的报表,如表格、主/细结构、多列邮件标签、矩阵表等,而且可将不同格式的报表合并成一个报表。Oracle Reports 充分利用 SQL 的所有功能,支持无限次数的数据库查询并能存取文本和图形图象数据。同时,可不必编程而采用描述方式完成数据统计功能,实现较复杂的分组计算。

2.2.3 Oracle Graphics 2.5

这是电子图形软件,具有交互式图形编辑功能,可创建一个复杂、直观的应用程序,它把数据转换成可视信息,把正文、图表、绘画以及多媒体对象结合在一起,以有意义的图形画面格式,显示数据之间的相互关系。利用其多媒体处理能力,可动态地操纵图形图象,录放声音。在应用程序里可以定义一个具有可视属性的对象,它会随数据的变化而变幻图形。因而,Oracle Graphics 对监测、跟踪数据的应用来说是个理想的工具。利用其图形按钮、多重活动窗口和明细信息结构,可以开发出强大的图形界面的执行信息系统和交互式决策支持系统。

3 结束语

CDE 2 是一个用于生成和维护可缩放的 Client/Server GUI 应用的、高效率的开发环境。使用说明性的、非过程化的方法,可在不必编程的情形下生成绝大部分应用所需的功能。这个高级的 CDE 2 环境的关键部分是重用和继承性,它们能使开发人员有效地提交满足实际需要的应用系统。CDE 2 为在当前和可预见的将来高效率地开发 Client/Server 应用系统迈出了关键的一步。

素为 1。

定义 2 对任意 (m, n) -矩阵 A 的最简准下三角阵 Q 做适当的行交换, 使 Q 的前 r 行前 r 列形成 r 阶单位矩阵后得到的矩阵 S 称之为 A 的准标准形。

定义 3 对任意 (m, n) -矩阵 A 从首行首列起做如下三种行初等变换得到的矩阵 T 称之为 A 的最简准上三角阵:

(1) 找出当前列中从当前行起某个非零元素所在的行与当前行交换, 若不存在这样的行则改变当前列为下一列另找;

(2) 用适当的数分别乘当前行加到其它各行, 使当前列中除当前行上的元素外其它元素都为零;

(3) 用一个数乘当前行, 使当前列中唯一的非零元素为 1。

定理 1 当在齐次线性方程组的系数矩阵 A 下接 n 阶单位矩阵 E , 利用列初等变换把 A 转化为最简准下三角阵 Q 时, 这些初等变换同时把 E 转化为 G , G 的 G_{r+1}, \dots, G_n 这 $n-r$ 个列向量构成该齐次线性方程组的一个基础解系。

定理 2 当在齐次线性方程组的系数矩阵 A 下接 n 阶单位矩阵 E , 利用列初等变换和行交换把 A 转化为准标准形 S 时, 这些初等变换同时把 E 转化为 H , H 的 H_{r+1}, \dots, H_n 这 $n-r$ 个列向量构成该齐次线性方程组的一个基础解系。

定理 3 在任意线性方程组的系数矩阵 A 下接 n 阶单位矩阵 E 而右接常数矩阵 B , 当列初等变换和行交换把 A 转化为它的准标准形 S 的同时, 相同的列初等变换把 E 转化为 H , 而相同的行交换把 B 转化为 B^* , r 是 S 中的非零列数, 此时

(1) 当 $r < m$ 且 $S_{m-r, r} \times B_r^* \neq (b_{r+1}^*, \dots, b_m^*)^T$ 时线性方程组无解, 否则有解且至少有一组解为 $H_{n, r} \times B_r^*$;

(2) 当 $r = n$ 时, 线性方程组对应的齐次方程组无基础解系, 否则 $r < n$ 时 $H_{n, n-r}$ 中的 $n-r$ 个列向量 H_{r+1}, \dots, H_n 是该齐次方程的一个基础解系。

定理 4 给任意线性方程组的系数矩阵 A 右接常数矩阵 B , 当行初等变换把 A 化为最简准上三角阵 T 的同时也把 B 化为 B^* , r 是 T 中非零行数, 此时

(1) 当 $r < m$ 且 $b_i^* (i = r+1, \dots, n)$ 不全为零时线性方程组无解, 否则至少有这样一组解, T 中非标准列所对应的未知元为零而标准列所对应的未知元为该列元素 1 所在行对应的 $b_i^* (i \leq r)$;

(2) 当 $r < n$ 时线性方程组有无穷多组解, 线性方程组所对应的齐次方程组的基础解系可由 T 的 $n-r$ 个非标准列的主对角线上的元素 0 改为 -1 而得到。

2 列初等变换求解算法

为了保证良好的数值运算精度, 下面给出由定理 3 得到的算法中采用了行主元素消去法求最简准下三

角阵。在算法中, 符号 A 为 (m, n) -系数矩阵, B 为 $(m, 1)$ -常数矩阵, E 为 n 阶单位矩阵, C 为 $(n, 1)$ -特解矩阵; m, n 为常量, i, j, k, r 为整型变量, t 为实型变量, K_1, \dots, K_{n-r} 为任意常数, E_{r+1}, \dots, E_n 表示 E 的后 $n-r$ 个列向量, X 为未知元列向量。该算法已在微机上用 GW-BASIC 编程调试通过, 一个程序可以解决任意线性方程的求解问题, 包括不定方程、确定方程和特殊情形——一元一次方程。源程序清单略, 下面仅给出算法的形式化描述。

(一)[初始化]

$0 = > i, r$

(二)[寻找行主元 j]

(1) $i+1 = > i, r+1 = > r, j$

(2) 对于 $k = r+1, \dots, n$ 做
若 $|a_{ik}| > |a_{ij}|$ 则 $k = > j$

(3) 若 $a_{ij} = 0$ 则

① $r-1 = > r$

② 若 $i < m$ 则转(二)否则转(六)

(三)[交换第 r 列与第 j 列]

若 $j \neq r$

(1) 对于 $k = i, \dots, m$ 做

$a_{kr} = > t; a_{kj} = > a_{kr}; t = > a_{kj}$

(2) 对于 $k = 1, \dots, n$ 做

$e_{kr} = > t; e_{kj} = > e_{kr}; t = > e_{kj}$

(四)[用 $-a_{ij}/a_{ir}$ 乘第 r 列加到第 j 列]

对于 $j = 1, \dots, n$ 做

若 $j \neq r$ 则

(1) 对于 $k = i, \dots, m$ 做 $-(a_{ij}/a_{ir}) * a_{kr} +$

$a_{kj} = > a_{kj}$

(2) 对于 $k = 1, \dots, n$ 做 $-(a_{ij}/a_{ir}) * e_{kr} +$

$e_{kj} = > e_{kj}$

(五)[用 $1/a_{ir}$ 乘第 r 列并检查 A 是否已为最简准下三角阵]

(1) 对于 $k = i, \dots, m$ 做 $(1/a_{ir}) * a_{kr} = > a_{kr}$

(2) 对于 $k = 1, \dots, n$ 做 $(1/a_{ir}) * e_{kr} = > e_{kr}$

(3) 若 $i < m$ 且 $r < m$ 则转(二)

(六)[行交换化 A 为准标准形]

对于 $j = 1, \dots, r$ 做

若 $a_{jj} = 0$ 则

(1) $j+1 = > j$

(2) 当 $a_{jj} = 0$ 反复做 $i+1 = > i$

(3) 对于 $k = 1, \dots, r$ 做

$a_{ik} = > t; a_{jk} = > a_{ik}; t = > a_{jk}$

(4) $b_i = > t; b_j = > b_i; t = > b_j$

(七)[判定方程组是否有解]

若 $r < m$ 则对于 $i = r+1, \dots, m$ 做

(1) $0 = > t$

(2) 对于 $j = 1, \dots, r$ 做 $t + a_{ij} * b_j = > t$

(3) 若 $t \neq b_i$, 则无解结束

(八)[有解, 求特解]

对于 $i = 1, \dots, n$ 做

(1) $0 = > c_i$

(2) 对于 $j = 1, \dots, r$ 做 $c_i + e_{ij} * b_j = > c_i$

(九)[输出全部解]

若 $r = n$ 则输出唯一解 $X = C$

否则输出全部解 $X = C + k_1 E_{r+1} + \dots + k_{n-r} E_n$

(十)[算法结束]

Motif 程序中长任务的处理

易鉴良 谢 立

(南京大学计算机系 南京 210093)

摘 要 本文基于 MVC 体系结构提出了一个在 Motif 程序中处理长任务的交互模型,讨论了实现该模型常用的几种方法,比较了它们在交互性能、复杂程度、额外开销和适用范围等方面的优缺点。

关键词 Motif 交互式程序 长任务 事件驱动

1 引言

当前,大部分应用程序都基于 X Windows、Microsoft Windows 等一些具有良好的交互功能的窗口系统。一个应用程序要具有良好的交互性能至少应该满足以下两个要求:1)用户能始终保持对程序运行的控制;2)程序始终保持向用户作适当的反馈。这样才能使用户感觉到程序是在自己的控制下运转,通过监视程序的反馈了解程序的工作情况,以便决定下一步的动作。

一般情况下,要做到上述两点并不很困难;但在有些情况下,例如在执行编译程序、大数据量计算或打印

等冗长任务时,要继续保持程序的交互性就有一定的难度,这是因为:1)应用软件的设计者容易倾向于让一个逻辑上完整的任务在整个运行过程中自始至终掌握控制权,不间断地运行,但对于运行时间较长的任务而言,这样做会引起使用者的焦急和反感;2)如果在一个冗长任务的处理过程中允许用户干预,将不得不在程序中引入复杂的控制结构和数据管理,有些设计者由于怕麻烦而予以回避。由于上述原因,有的程序设计者就简单地让一个交互式程序调用一个冗长任务而加任何处理,使一个长任务长时间地占有控制权,大大降低了程序的交互性能。

X Windows 是一个工业标准,它提供了一个移植性很强的用户界面系统,Motif 是 X 的一个高层工具

收稿日期:1996年5月19日

3 行初等变换求解新算法

下面给出由定理 4 构造的求解算法,该算法中采用了列主元素消去法求系数矩阵的最简准上三角阵。算法中使用的符号 A 为 $(m, n+1)$ -增广矩阵, m, n 为常量, i, j, k, r 为整型变量, t 为实型变量。该算法也可解决任意线性方程的求解问题,已用 GW-BASIC 编程在微机上调试通过,源程序清单略,算法的形式化描述如下:

1)[初始化]

$0 = > r, j$

2)[寻找列主元]

(1) $j+1 = > j, r+1 = > r, i$

(2) 对于 $k = r+1, \dots, m$ 做

若 $|a_{kj}| > |a_{ij}|$ 则 $k = i$

(3) 若 $a_{ij} = 0$ 则

① $-1 = > a_{ij}; r-1 = > r$

② 若 $j < n$ 转 2) 否则转 6)

3)[交换第 i 行与第 r 行]

若 $i \neq r$ 则

对于 $k = j, \dots, n+1$ 做

$a_{ik} = > t; a_{rk} = > a_{ik}; t = > a_{rk}$

4)[用 $-a_{ij}/a_{rj}$ 乘第 r 行加到第 i 行]

对于 $i = 1, \dots, m$ 做

若 $i \neq r$ 则

对于 $k = j, \dots, n+1$ 做

$a_{ik} - (a_{ij}/a_{rj}) * a_{rk} = > a_{ik}$

5)[用 $1/a_{rj}$ 乘第 r 行并查 A 是否已为最简准上三角阵]

(1) 对于 $k = j, \dots, n+1$ 做 $a_{rk}/a_{rj} = > a_{rk}$

(2) 若 $j < n$ 则转 2)

6)[判定方程组是否有解]

对于 $i = r+1, \dots, m$ 做

若 $a_{in+1} \neq 0$ 则无解结束

7)[有解,确定特解]

对于 $j = 1, \dots, n$ 做

若 $a_{jj} = -1$ 则 $0 = > x_j;$

否则 $a_{jn+1} = > x_j;$

8)[确定基础解系]

若 $r < n$ 则

对于 $j = 1, \dots, n$ 做

若 $a_{jj} = -1$ 第 j 列为基础解系中一个解向量

9)[输出全部解,算法结束]

参考文献

- 1 武汉大学数学系编,线性代数,人民教育出版社,1980.
- 2 王秀丽、宁正元,线性方程的列初等变换求解算法,现代电子技术,1995.No. 2.
- 3 王秀丽、宁正元,线性方程组求解的两种新方法,数学教学研究,1995 优秀论文专辑.
- 4 许卓群、张乃孝等编,数据结构,高等教育出版社,1989.
- 5 上海计算技术研究所编,电子计算机算法手册,上海教育出版社,1982.
- 6 宁正元、王秀丽编著,结构化程序设计教程,陕西电子杂志社,1994.

箱。Motif 程序基于事件驱动,它通过连续不断地处理事件队列,响应窗口暴露(Expose)、鼠标器输入(ButtonPress)等事件,来完成程序的功能。

在一个 Motif 程序中,如果程序员简单地调用一个运行时间较长的任务,程序将在一段时间内不能返回事件循环,这会对程序造成严重的影响。首先,如果应用程序窗口的大小被重新设定或者窗口暴露出来,由于程序无法处理.Resize、Expose 等事件,窗口中会留下一大片空白,严重破坏程序的外观;更严重的是,如果用户不了解程序正在干什么而在应用程序界面上乱按鼠标器或键盘,Motif 程序将按时间先后对这些事件排队,并在长任务返回时依次处理队列中的这些事件,其结果是不可预见的和灾难性的,可能会造成程序的崩溃^[1]。因此必须很好地解决 Motif 程序在运行长任务时与用户的交互问题。

本文基于 MVC 体系结构提出了一个在 Motif 程序中处理长任务的交互模型,它能有效地解决上述的问题。

2 MVC 体系结构与交互模型

2.1 MVC 模型

模型—视图—控制器(MVC)体系结构提供了一种强有力的方法来实现支持相同信息的多重表示的系统,但它在交互系统中还有许多潜在的用途^[2]。我们将利用 MVC 框架来处理 Motif 程序中的冗长任务。

MVC 框架建立在三种类型的对象基础之上:Model、Viewer 和 Controller。模型(Model)可用于表示原始数据或者事物实体等,当 Model 中的数据改变时,它将通知所有的视图进行更新;视图(Viewer)一般有多个,分别以不同的形式向用户显示 Model 全部或部分的数据,它们必须能响应 Model 发出的更新消息,以便根据自己的能力及时反映出 Model 当前状态的变化;控制器(Controller)允许用户操作 Model 中的数据,它常常与 Viewer 结合使用,因为用户必须通过 Viewer 了解 Model 情况的前提下才能对数据进行正确的操作。

2.2 适用于 Motif 程序处理长任务的交互模型

Motif 风格的程序中界面设计与实际动作是分开的,它们之间必须要通过回调函数建立起联系,因此整个程序是按用户界面组织的,事件处理是中心环节。结合 MVC 框架和 Motif 程序的这个特点,我们将采用图 1 所示的交互模型来处理 Motif 程序中的长任务。

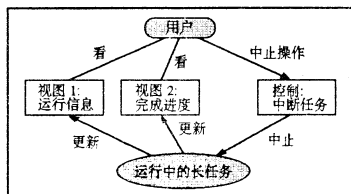


图 1

从图中可以看到,我们将运行中的长任务作为数据来源,它相当于 MVC 框架中一个不断变化的 Model,它将根据自己的情况不断向视图发出更新消息。一共有两个视图:视图 1 显示有关任务运行的消息,例如在编译过程中显示扫描的遍数、编译的行数等,使用户能较详细地了解任务目前的执行情况;视图 2 显示目前完成的任务占任务总量的百分比,使用户对该任务何时能完成有一个大概的估计。当然这两个视图不一定要全实现,也可加入其它的视图。在任务运行期间,程序向用户提供的唯一的控制是将任务中止,以便用户在不愿意任务继续进行时发出中止命令;在此不提供其它控制是因为那样做会大大增加程序编制的复杂性,同时在多数情况下也没有这个必要。

3 交互模型的实现方法

3.1 视图和控制的一种实现

我们在 Motif 中的 WorkingDialog Widget 的基础上来实现上述的两个视图和一个控制。在调用一个长任务时,首先创建一个 WorkingDialog Widget 并把它映射到屏幕上,表示任务开始运行,视图和控制将在 WorkingDialog 的不同部位上显示出来:

1) 运行信息的显示:根据任务的执行情况,把任务的运行信息描述成文本方式的复合串(XmString),并将 WorkingDialog 的 XmNmessageString 资源设置为该复合串,并随时根据新的情况不断地更新这个资源,就能在 WorkingDialog 的正文区显示任务的运行情况。

2) 完成进度的显示:完成进度用圆饼图来显示,整个圆圈表示整个任务,圆圈中黑色扇形区域表示已完成的任务,同时在圆圈中再写上进度的百分比表示。一个任务的圆饼图将从一个空心的圆圈(0%)开始,到一个黑色实心的圆(100%)为止。根据任务的实际进度,把上述的图象画在一个 Pixmap 中,将 WorkingDialog 的 XmNsymbolPixmap 资源设置为该 Pixmap,就能在 WorkingDialog 窗口左上角显示出一个表示任务进度的动画。

3) 控制按钮的实现:在 WorkingDialog 中有三个缺省的按钮(OK、Cancel 和 Help),我们将 OK 和 Help 按钮关闭,同时把 XmNcancelLabelString 资源设置为“Stop”,显示一个用于中止任务执行的 Stop 按钮。然后把 InterruptCallback()回调函数安装在 Stop 按钮上,该函数完成中止任务执行的命令。

视图和控制也可以用其它方法来实现,但必须注意的是采用的方法不能过于复杂,否则将在这些辅助设施上消耗过多的 CPU 时间,大大延长任务的处理时间,结果得不偿失。

3.2 冗长任务中处理事件的方法

程序对用户中止操作的响应以及运行信息和进度的显示都要求 Motif 程序能经常地返回事件循环,同

时, Motif 程序中窗口内容和三维阴影等设施的维护也要求 Expose 等事件能得到较为迅速的处理。因此, 怎样在处理长任务时使程序不断地返回事件循环, 成为 Motif 程序编写中的一个关键问题。下面我们将详细讨论解决这一问题的几种方法。

(1) 忽略法(忙光标法)

表示应用程序正在忙的最简单的方法是将光标显示为忙, 通常是一个手表形光标。同时还必须关闭用户输入, 因为如果允许用户在处理长任务时还可以进行其它操作会大大增加程序的复杂程度, 实现起来会十分困难。这两点是以下几种方法中也都必须要做到的。

可以用 Xlib 函数 XDefineCursor() 将应用程序的顶层 Shell 的光标设为 XC_watch(手表形光标)来显示一个“忙光标”。关闭用户输入的方法有多种, 例如: 1) 显示一个对话框, 将对话框的 XmNdialogStyle 资源设置为 XmDI ALOG_FULL_APPLICATION_MODEL, 这种对话框关闭除对话框本身外向应用程序的所有输入; 2) 创建一个 Input-Only 窗口作为顶层 Shell 的儿子, 当程序处理长任务时, 映射该窗口并覆盖所有应用程序的窗口(Input-Only 窗口不可见, 因此对外观无影响), 这样所有输入均被该窗口接收并丢弃, 同样起到禁止输入的效果。

忽略法的缺点是: 无法实现用户对长任务的中止要求, 无法处理 Expose 等事件来维护 Motif 程序的外观, 也无法显示任务的运行信息, 交互性极差, 优点是: 实现简单, 额外开销小, 不破坏长任务的完整性。忽略法适用于运行时间为几秒钟的任务, 因为这样的任务虽然要花一定的时间, 但用户还不会等得不耐烦, 因此不值得在交互设计方面花大量的精力。

(2) 工作过程法

可以将长任务分解成很多小任务块, 然后用工作过程(Work Procedure)的方法模拟后台任务来完成。工作过程用函数 XtAppAddWorkProc() 来登记, 安装好以后, Motif 程序的执行过程是这样的: Xt 先检查是否有事件等待处理。如果有, 就先处理这些事件; 如果没有, 就调用工作过程, 等到工作过程结束后, Xt 再次检查事件队列, 如此反复进行直到工作过程返回 True 值或者用 XtRemoveWorkProc() 函数将工作过程移去为止。为了保持程序良好的交互性能, 工作过程的每次调用只能占用很小的一个时间片。

采用工作过程法处理冗长任务过程是: 程序接到长任务启动命令后首先进行初始化, 初始化工作包括完成忙光标法中的两个设置, 在 WorkingDialog 的 Stop 按钮上安装中止回调函数, 登记工作过程 WorkProc() 以及映射 WorkingDialog 窗口, 然后返回事件循环。这样, 程序就能通过不断地调用 WorkProc() 来处理任务和更新任务运行信息与完成进度的显示。由于程序能经常返回事件循环, 因此能及时处理

Expose 等事件来维护 Motif 程序的外观, 并能在对话框中 Stop(即 Cancel)按钮被按时及时中止任务的运行。

工作过程法的优点是: 能显示任务的运行情况, 能向用户提供中止任务的功能, 具有良好的交互性能, 缺点是: 一个任务必须在一次工作过程调用后记住自己的状态, 以便在下一次调用时接着上次的工作继续进行下去, 因此程序数据结构和控制结构较复杂, 适用范围有一定的限制, 有些任务由于无法很合理地分割而无法采用这种方法, 例如编译程序就不适合作这样的分割。工作过程法所适用的任务一般有这样两个要求: 1) 任务能在逻辑上较为合理地分割成许多小块的任务; 2) 任务具有非独占性, 适合于在后台处理。

(3) 定时器法

定时器法与工作过程法有些类似, 也要将任务分成小块, 用 XtAppAddTimeOut() 安装超时回调函数 TimeOutCallback()。当到达一定的时间间隔时, Xt 将调用 TimeOutCallback()。由于超时回调在一次调用后自动清除, 因此如果任务还没有完成, 需要在超时回调函数中再次用 XtAppAddTimeOut() 安装超时回调函数, 以便在下一个时间间隔后再次调用超时回调函数继续执行任务。使用超时回调方法处理长任务的过程可以参照工作过程法。

定时器法的优缺点也与工作过程法相似。这种方法适合于处理稳定的、周期性的任务, 例如进行周期性的测量等。

(4) 自处理法

自处理法不把控制权移交给 Xt Intrinsics, 而是直接在长任务执行过程中使用 Xlib 和 Xt 的一些低级函数对事件队列进行处理。下面是一个典型的处理过程:

```
Initialize();  
While(! TaskHasDone){  
    if(CheckInterruptEvent())  
        InterruptTask();  
    .... /* Continue doing task */  
    XmUpdateDisplay();  
}
```

首先进行初始化, 完成置忙光标、关闭输入、打开对话框等工作。在任务的每个循环中, 首先检查是否有中止按钮事件, 如果有就调用 InterruptTask() 中止任务的执行, 如果没有就继续处理任务中的这次循环, 在循环结束时还要调用 XmUpdateDisplay(), 它的作用是为 Expose 事件检查事件队列, 任何迫近的 Expose 事件都被发送, 其它事件仍留在队列中。这种技术在自处理法中是很重要的, 因为程序在长任务运行期间不回到 Xt 的事件循环, 在对话中报告任务运行信息和完成进度等消息就需要用这种技术来显示。这里所介绍的只是一个非常简单的自处理过程, 程序员利用自处理法

的灵活性还可以编写出功能强大得多的程序。

自处理法的优点是:事件队列由用户自己处理,用户在长任务执行过程中能对整个程序保持较强的控制能力,程序的交互性能良好;缺点是:由于所有的事件都由用户来考虑,因此程序设计时头绪较多,程序结构较复杂,没有充分利用 Motif 编程的方便性。自处理法的适用范围也有一定的限制,它要求任务中有循环、递归等结构,以便周期性地检查事件队列并处理其中的事件,并且每次循环或递归都不能太长,否则将影响程序的交互性能。

(5)子进程法

有些任务是无法分割也无法在任务中插入代码的,例如调用一个系统命令或进行一次数据库查询等,这样的任务就无法使用工作过程法或自处理法来完成,于是很自然地会想到用子进程的方法。

子进程法的工作过程如下:1)进行初始化,包括置忙光标、关闭事件输入和映射对话框等;2)父进程调用 pipe() 建立管道,用于子进程与父进程之间进行通信;3)父进程调用 fork() 创建一个子进程(有可能还要在子进程中调用 exec() 启动一个任务);4)父进程关闭管道的 A 端,从管道的 B 端接受来自子进程的信息;子进程关闭管道的 B 端,并将控制台输出 stdout 和 stderr 重定向到管道的 A 端,这样父进程和子进程间就建立起了一个数据从 A 向 B 流动的单向通信管道;5)父进程用 XtAppAddInput() 函数让 Xt Intrinsics 在处理事件的同时监视一个附加的文件描述符,在这里就是管道的输入端(B 端);6)子进程继续执行任务,并向管道写入新的运行情况,父进程中如果发现管道中的数据已经可以读取,就调用 XtAppAddInput() 对应的回调,读入这些数据并在 WorkingDialog 中显示出来;7)如果用户中止任务,父进程就向子进程发出 kill 命令结束任务。

使用子进程法的优点是:程序的交互性能较好,长任务能作为一个整体进行设计,因此完整性较好,适用范围广,能用作处理长任务的一个通用模板;缺点是:设计父进程和子进程之间的通信相对复杂,父进程对子进程的控制能力不很强,另外创建一个子进程的开销较大,子进程法几乎能用于所有的长任务处理,特别是那些无法在任务中加入事件处理的任务。

(6)线程法

在有些 UNIX 系统中,提供了线程机制,例如 SunOS 中有轻量进程(Light Weight Process)设施。在子进程法中如果用 exec() 启动一个任务需要进行一次文件操作,在父进程和子进程之间要通过管道、FIFO 等进行通信,而在线程法中创建一个线程只需要一次内存操作,两个线程间的通信可以通过共享来实现,因此线程法的开销比子进程法的开销要小得多。采用线程法处理长任务的基本思路与子进程法相同,只

是在创建任务和线程间通信等方面有一些技术上的不同,例如在 lwp 库中使用 lwp_create() 来创建一个线程,并有一套 mon_ 打头的信号量管理函数来管理共享变量,实现线程间的通信。

使用线程法具有子进程法所具有的所有优点,而且能有效地降低任务的额外开销,同时使用共享变量能方便地进行双向通信,降低了程序编写的复杂度,因此线程法比子进程法具有更大的优点。遗憾的是有的系统中没有提供线程设施。

3.3 几种方法的比较

表 1 是上述六种方法的一览表,比较了这些方法在几个主要方面的性能,我们可以根据具体的任务情况来选择。

表 1

	信息反馈	用户控制	开销	复杂度	完整性	适用范围
忽略法	无	无	极小	极简单	好	小任务
工作过程法	有	有	小	较简单	差	可分块的任务
定时器法	有	有	小	较简单	差	周期性的任务
自处理法	有	强	小	复杂	好	有循环递归的任务
线程法	有	有	较小	较简单	好	绝大多数任务
子进程法	有	有	较大	较简单	好	独立性强的大任务

4 效果测试

我们对忙光标法(BCursor)和线程法(Thread)的开销与用户的感受进行了测试,结果如图 2 所示。(a)中横坐标表示任务所需的执行时间,纵坐标表示加入交互处理后实际花费的时间。可以看出忙光标法为斜率接近 1 的直线,额外开销几乎为零;线程法的斜率略大于 1,有一定的额外开销。但总的来说增加的开销是较小的,对任务的执行不会产生很大的延迟。(b)中横坐标表示任务所需的执行时间,纵坐标表示用户感觉中的任务执行时间的长度。可以看到线程法能使用户很好地忍受一个长任务,而忙光标法在任务较长时将使用户很不耐烦。

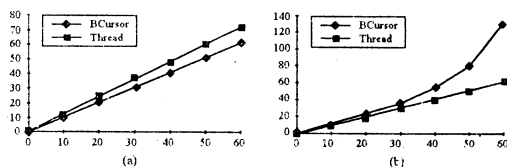


图 2

虽然在采用了交互模型处理长任务后,任务执行时间略有加长,但从用户的角度来看,等待任务完成的时间仿佛缩短了,起到了良好的心理效果。因此使用该交互模型来处理长任务是很有效的。

参考文献

- 王劲松等,“C++和 OSF/Motif 面向对象的混合编程技巧”,北京希望电脑公司。
- 任杰,一侠等,“C++下 Motif 编程”,北京大学出版社,1994。
- 龚雨,曹田等,“Motif 实用编程大全”,学苑出版社,1994。

Windows 环境下图象处理的内存操作和显示技术

唐贵川 王建成 董金祥 何志均

(浙江大学 CAD&CG 国家重点实验室, 浙江大学计算机 AI 所 杭州 310027)

摘 要 Windows 已成为 PC 机事实上的标准环境, 在该环境下开发的图象处理软件具有广泛的应用背景, 本文阐述了该环境下图象处理的基础技术内存操作和显示, 并运用于一个参数化的类属的图象处理软件系统的原型开发之中。

关键词 图象处理 内存操作 显示

1 引言

Intel 处理器和 Microsoft 的窗口软件交相辉映, 使微机系统突然生彩, 气势直逼工作站和小型机, 越来越高的显示器分辨率和越来越大内存, 更使微机在计算机视觉和多媒体^[1]图象处理方面享有一席之地。传统微机+图象处理卡+专用图象显示器+图象库组合的图象处理系统, 因为不可移植性, 专用性和较高的价位, 急需更新。鉴于此, 我们在微机上 Windows95/3.1^[2]环境下开发了一个参数化的类属的图象处理软件原型系统。实践表明, 该系统实用, 可靠, 可重用。本文主要介绍该原型的基础技术, 内存图象的处理与显示。

2 内存对象操作

2.1 图象处理内存对象的选取

Windows 有全局堆和局部堆现两种内存资源。全局堆由所有程序共享全局可分配, 通过调用 GlobalAlloc, 请求 Windows 从全局堆分配一个指定大小的内存对象, 返回分配的内存对象句柄, 如果全局堆中没有足够的内存, GlobalAlloc 返回 NULL。检查 GlobalAlloc 的返回值很重要, 管理全局内存的许多函数都要根据句柄来确定内存对象。

图象处理前, 调用 GlobalLock 锁住全局内存对象 (对应一幅图象) 的句柄, GlobalLock 返回的远指针在调用 GlobalUnlock 之前有效。如果对象已经锁住, 这时即可以进行图象处理。局部堆除建立在任何数据段外, 用法类同全局堆。

较局部堆的 64KB 的限制, 全局堆适合图象处理大内存的需求, 可保证多窗口图象处理的需要。原型用移动的全局堆作为图象缓存, 程序员无须再为 64KB 的段限烦恼而直接在大内存中操作图象, 免除了基于块映射的存储空间切换, 简化了程序复杂度。

2.2 WIN32 的高速 DIB 绘制支持

另外, WIN32 提供了一个函数 CreateDIBsection, 如果对图象没有比例变化, 它可以快速直接写视频, 提高显示速度。Microsoft 公司为了更快地对视频操作, 还提供 WinG 供开发者使用。

2.3 内存图象处理

Windows 对内存严格管理, 操作内存时要有一定顺序。图象处理前先载入图象, 申请一块图象长度的全局堆, 获取内存对象句柄, 上锁, 然后读入图象, 读完后开锁。这里省略了判断语句, 给出示意性程序:

```
hDIB=GlobalAlloc(可移动的, 0, 图象长度); /* 申请全局堆 */  
lpDIB=GlobalLock(hDIB); /* 锁取全局堆指针 */  
lReadImage(hFile, lpDIB, 图象长度); /* 读入图象 hFile 到 lpDIB 处 */  
GlobalUnlock(lpDIB); /* 开锁 */  
然后进行图象处理,  
hWnd=GetCurrentMDIWnd(); /* 取当前 MDI 子窗句柄 */  
hDIBInfo=GetWindowWord(hWnd,...); /* 取设备无关位图信息头 */  
lpDIBInfo=GlobalLock(hDIBInfo); /* 取设备无关位图信息头指针 */  
lpDIB=GlobalLock(lpDIBInfo->hDIB); /* 取位图内存首地址 */  
ImageProcess(ProcessType, lpDIB); /* 对 lpDIB 指向的内存图象进行处理 */  
GlobalUnlock(lpDIBInfo); /* 开锁 */  
GlobalUnlock(lpDIBInfo->hDIB); /* 开锁 */
```

说明: GetCurrentMDIWnd 是多窗口多图象处理中, 取当前图象窗的句柄; 在 ImageProcess 函数中, 用户可以对 lpDIB 指向的内存象常规的内存访问进行 ProcessType 指定的处理。处理后, 用户并不立即看到处理结果, 而要发 WM_PAINT 消息给 Windows 系统, 下节将进一步说明。

3 图象显示

3.1 显示接口函数

Windows 显示图象不是常规的往视频写, 而是提供了两类显示接口函数: 位图函数和设备无关位图函数。

位图函数 BitBlt() 将位图从内存设备描述表 (有关设备描述表的介绍参阅 WINDOWS 资料) 拷贝到屏幕以显示一个内存位图。StretchBlt() 区别于 BitBlt 将位图拉伸或压缩后拷贝到屏幕上。

设备无关位图函数 SetDIBitsToDevice() 直接将设备无关位图全部或部分地设置到输出设备上。

Internet 与异种网络互连中 DNS 名字的转换

宋志辉

(贵州教育学院实验电教中心 贵阳 550003)

朱 华

(贵州省科委计算中心 贵阳 550002)

摘 要 由于 Internet 不是一个单一、规范、标准的实体,而是由数千个不同的较小网络互连而成的一个庞大的混合体,因而在发送 E-mail 时,它们并不一定遵守 DNS(Domain Name System)的国际协定的命名规则,从而使用户倍感困难。本文在查询部分资料及实践的基础上,列出部分常用网络地址转换的一般方法,供 Internet 用户参考。

关键词 Internet DNS 转换

1 前言

收稿日期:1996 年 10 月 10 日

StretchDIBits()区别于 SetDIBitsToDevice 将设备无关位图拉伸或压缩后拷贝到屏幕上。设备无关位图函数允许位图显示在任何给定分辨率的设备上,而系统使用什么颜色管理过程则无关紧要。设备无关位图的优点之一是程序可以直接显示它们而无需创建一个中介内存位图。

3.2 图象显示

鉴于通用性,原型采用 SetDIBitsToDevice 和 SetDIBitsToDevice 显示位图。本可以用 Windows 提供的 SetDIBits()和 GetDIBits()访问位图中象素,但图象处理往往有图象区域块的处理,也为了速度上的需要,原型采用锁图象内存来操作图象

原型法显示图象的过程大致为:

(1)在 WM_PAINT 消息下,组织 SetDIBitsToDevice 和 SetDIBitsToDevice 以显示图象;

(2)用户决定图象处理行为,锁图象,处理图象,开锁;

(3)调用 InvalidateRect 函数,以在屏幕上显示图象。

图象处理后,须产生 WM_PAINT 消息,因此要调用 InvalidateRect 显示图象。InvalidateRect(hWnd, lpRect, bErase)函数为窗口产生 WM_PAINT 消息,标识一部分用户域(即图象窗或其中一块区域)无效(即需要重绘)。例如下面语句使变量 hWnd 指定的窗口占全部用户域无效,

InvalidateRect(hWnd, NULL, TRUE);

hWnd:待更新的窗口,lpRect:指明裁剪区,NULL 表明整个客户区重绘,bErase:为 TRUE,则擦背景,为 FALSE,则保留背景

如果不想等待从应用程序队检取 WM_PAINT 消息,可以用 UpdateWindow 函数产生一个立即 WM

Internet 不是一个单一、规范、标准的实体,而是由数千个不同类型的较小网络互连而成的一个庞大的混合体。这些早期的小网络由于开发目的不同,应用领

WM_PAINT 消息,如果无效用户域存在,UpdateWindows 函数从指定窗口的消息队中取出 WM_PAINT 消息直接发送到窗口函数。

4 结束语

我们在 Windows 95/3.1 下,用 Visual C++ 开发了这个参数化和类属^[3]的图象处理分析原型,运用上述内存操作和显示过程,增强了图象处理分析的适应性和实用性。

该原型还是一个通用的功能易扩充的系统,用户在这个原型上,可以在短时间内迅速开发出专用的或功能更强大的图象处理分析软件系统。另外原型还提供多图象窗口处理技术和动态灰度观察以及图象任意区域的打印等功能。原型的图象处理部分用 C 完成类及类属的特征,没有用 C++ 面向对象的功能,具有广泛的移植性。

附图(略)参数化的类属的通用图象处理软件原型示例图同时打开十几幅图象(512 * 512 * 8 位)进行处理分析(增强,恢复,分割,分析等)

某些图象可缩为图标,可灰度动态显示,可子窗口处理,可……

参考文献

- 1 唐贵川,孙杉,多媒体技术及其发展,计算机应用研究,1993.4
- 2 Microsoft Corp, Microsoft Windows 3.1 Guide to Programming, Microsoft Press. 1992
- 3 蔡希尧,陈平,面向对象技术,西安电子科技大学出版社,1993.11

附:有需要该图象处理分析软件系统源程序者,可来信说明,来信请寄:

310027 浙江杭州浙江大学 621 信箱

唐贵川(收)

域不同,在网络基础结构的某些方面存在着许多限制,从而阻碍了 Internet 用户在更广大领域中的应用。近几年来,各种高性能 PC 机不断涌现,许多用户已习惯于将高性能 PC 机置于 Internet 网络环境中,从而使 Internet 用户随之而不断膨胀。在大家越来越深刻地认识到 Internet 的许多优点的同时,也希望任何一种通讯服务都能提供这种全球性连接。遗憾的是对于 Internet 来说,存在的问题并非用户们想象的那样简单,在许多应用细节上不同的小网络各有特点。其中一个问题就是在这些不同的小网络中发送 E-mail 时,它们并不一定都遵守 DNS(Domain Name System)的国际协定的命名规则,而使用户倍感不便。本文在实践和查询部分资料的基础上,不一定规范地提供一个将小网络地址与 Internet DNS 命名规则的名字互换的方法。由于手上资料及篇幅的限制,只能列出部分常用网络或服务机构。

2 转换

2.1 AT&T Mail

AT&T Mail 是为满足 AR&T 商务通信的需要而开发的,它提供 E-mail 文件传送、BBS、传真等服务。

若希望从 AT&T Mail 上将一个消息通过 E-Mail 发送到 Internet 上名为 dnsuser@sub.domain 的用户,则在 AT&T Mail 上的消息的目的地址应为:

internet! sub.domain! dnsuser

若要从 Internet 上通过 E-mail 将消息发往 AT&T Mail 中帐户名为 attmail user 的用户,首先应将帐户名转换为一个按 DNS 规则命名的用户名,其方法是用@将用户帐户名与 AT&T Mail 的网关名相连接,AT&T Mail 的网关名是:attmail.com。本例消息的目的地址应为:

attmailuser@attmail.com

2.2 CompuServ

CompuServe 是美国最大和最著名的联机服务机构,在大多数美国城市,都可用当地电话呼叫 CompuServe 的拨号式访问,从中你可以找到各式讨论话题的联机论坛和许多供应商提供的技术支持信息,例如:Microsoft, BorLand, Lotus 等。

一个 CompuServe 帐户名是由两个数值字段组成的,它们之间用一个逗号“,”分开。例如:12345,1234。当前,CompuServe 提供了一个到 Internet E-mail 的网关,我们可以通过 telnet 连接到 CompuServe,但这种访问方式所负担的通讯费用较高。

若希望从 CompuServe 帐户中给 Internet 上的名为 dnsuser@sub.domain 的用户发一个 E-mail 消息,则该 E-mail 消息的目的地址应为:INTERNET: dnsuser@sub.domain,即在 Internet 用户名前加上 INTERNET;注意所有字母必须大写。

在 CompuServe 中有两种帐户,个人帐户和团体

帐户。当希望从 Internet 中给 CompuServe 帐户发送 E-mail 时,应知道该帐户是个人帐户还是团体帐户。若是个人帐户,则只需以句号“.”代替原目的帐户中的逗号“,”,然后再加上网关名。例如:目的帐户是个人帐户 12345,1234,则发送 E-mail 的目的地址应为:12345.1234@CompuServe.com。

在 CompuServe 中,团体帐户的格式为:organization: department: csuser,当需要给其发送一个 E-mail 消息时,目的地址应为:csuser@department.organization.compuserve.com。注意:有的团体用户没有 department 字段,此时,目的地址中省掉“department.”即可。

2.3 Connect

Connect 是一个专业信息网,它提供 E-mail 和文件信息库服务,Connect 有一个专门针对 Internet 的 E-mail 网关 connectinc.com。

若要从 Connect 中给 Internet 上的用户 dnsuser@sub.domain 发送一个 E-mail 消息,则消息的目的地址应为:DASN,同时请在消息的第一行写下:

“dnsuser@sub.domain”@DASN

当希望从 Internet 上给 Connect 的用户 CONUSER 发送一个 E-mail 消息时,该消息的目的地址应为:

CONUSER @connectinc.com

2.4 Prodigy

prodigy 是 Sears 公司与 IBM 公司合作开发的一个较新的联机服务网络,它使用专用软件,除提供 E-mail 和文件传输服务外,还提供图形用户接口,该专用软件叫 Mail Manager,适合于 IBM PC 机及其兼容机和 Macintosh 平台。

当希望从 Prodigy 上向 Internet 的其它用户发送 E-mail 消息时,请别忘了应有一个名为 Mail Manager 的软件包,有了这个软件包之后,只需按 DNS 命名规则,给出 Internet 用户名即可。

从 Internet 上发送 E-mail 消息给 Prodigy 的用户时,消息的目的地址格式应为:userid@prodigy.com.userid 是 Prodigy 分配给其用户的 ID,使用时,只需将上述格式中的 userid 换为用户的 ID 即可。

2.5 Applelink

Applelink 是专门提供给 Apple 计算机用户的信息服务网络,其目的是为 Apple 及其相关的第三方产品提供 E-mail 服务和联机的技术支持。Applelink 支持一个至 Internet 的网关 applelink.apple.com。

若希望从 Applelink 上给 Internet 上的用户发一个 E-mail 消息,则应将“@internet#”加到 Internet 用户的 DNS 名后。例如:给 Internet 用户 dnsuser@sub.domain 发送一则 E-mail 消息,则该消息的目的地址应为:

dnsuser@sub.domain@internet #

注意: Applelink 有一个限制, 其消息的地址不能超过 35 个字符。

若需从 Internet 上给 Applelink 的用户发送一条 E-mail 消息, 则直接把 Applelink 支持的网关地址加到 Internet 用户的 DNS 名字后。例如给 Applelink 的用户 alinkuser 发送一条 E-mail 消息时, 其目的地址应为:

alinkuser@applelink.apple.com

2.6 America Online

America Online 是美国商业性联机服务机构网络, 它向用户提供 E-mail、联机讨论、邮件清单、网络新闻和文档等服务。目前, America Online 只提供至 Internet 的邮件网关服务。

从 America Online 发送一个 E-mail 消息到 Internet 用户十分简单, 只需在制作消息时, 将消息的相应目的地址设置为 Internet 接收者的 DNS 名字的即可。

当需从 Internet 上给 America Online 用户发送一个 E-mail 时, 则必须删除 America Online 用户名中的所有空格, 全部字符均用小写, 从而将 America Online 用户名转换为 DNS 允许的名字, 再将 America Online 的域名(aol.com)加到后面。例如: 一个 America Online 用户名为 John Q, 则转换后的名字应为:

johng@aol.com

注意: America Online 消息的最大长度是 32k 字节, 其中若出现不可打印字符时, 一律转换为空格。

2.7 BITNET

BITNET 是由许多商业机构、研究院(所)和大学组成的全球性网络, 在 BITNET 和 Internet 间可以使用 E-mail 和文件传输服务, BITNET 用户名的一般格式是: USER@NODE。在 BITNET 和 Internet 之间有多个网关, 通常使用最多、最著名的是 cunyvm.cuny.edu 和 mitvma.mit.edu。

当希望从 BITNET 上给 Internet 上的用户 dnsuser@sub.domain 发送一则消息时, 请先注意你的 BITNET 主机上运行的邮件程序。你可以先尝试直接把消息发给 DNS 名(如: dnsuser@sub.domain)。若失败, 则再试将消息发给 dnsuser%sub.domain@gateway, gateway 用上述两个网关名之一代替。

若要从 Internet 上发一条消息给 BITNET 上的用户 BITUSER@BITSITE, 其目的地址应为:

bituser%bitsite@gateway

其中: gateway 用合适的 BITNET 网关名代替。

2.8 EARN

EARN 是欧洲学术与研究网络(European Academic and Research Network)的简称, 它连接了欧洲、中东和非洲的许多研究机构 and 大学。EARN 采用与 BITNET 相同式样的地址和方法, 这里就不再重复叙述了。

2.9 DFN

DFN 是德语 Deutsches Forschungs Netz 的简称, 通常称为德国科学网络, 它是一个基于 X.400 的大型网络, 它连接了德国的几乎全部研究机构和大学, 它的网关同时也是 EARN 或 BITNET 的网关。

当从 DFN 上给 Internet 上的用户发送 E-mail 消息时, 其方法与 BITNET 相同, 它也使用 BITNET 的网关。

当给 DFN 用户发送 E-mail 消息时, 目的地址的写法取决于对方主机连接的是基于 X.400, 还是 EARN。若是给基于 X.400 的用户 dfnxuser 发送 E-mail, 则目的地址应为: dfnxuser@sub.domain.dbp.de。若是给基于 EARN 的用户 earnuser.earnsite, 则目的地址应为: earnuser@earnsite.bitnet。

2.10 Calvacom

Calvacom 是一个在法国范围内的商业性消息传送服务网络。

当从 Calvacom 上给 Internet 上的 DNS 名为 dnsuser@sub.domain 的用户发送 E-mail 消息时, 其消息的目的地址应为:

EM/dnsuser@sub.domain。

若是从 Internet 上给 Calvacom 上的用户 caluser 发送 E-mail 消息时, 消息的目的地址应为:

caluser@calvacom.fr。

3 结束语

全世界大多数较大型的网路与 Internet 间都存在某种类型的连接, 它们将会过渡到采用 Internet 式样的 E-mail 地址, 特别是 Internet 主机数量的持续增长和许多新网络的加入, 都会转而使用 TCP/IP 协议簇, 以保证与其它网络的兼容性, 并适应 Internet 未来的发展, 这将给大多数非计算机专业的用户带来极大的方便。

Scott Yanoff 定期编辑和公布一些对照清单, 它的电子邮件地址为 Yanoff@csd4.csd.uwm.edu, 你可以从 csd4.csd.uwm.edu 主机上通过匿名 ftp 检索到该清单, 也可使用 Mr. Yanoff 帐户名中的 finger 工具软件, 了解获得该清单的其它途径。

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)

串行 EEPROM 的读写器的设计

孙洪军

(东南大学电子工程系 南京 210096)

孙秀云 王连军

(南京理工大学化工学院 210000)

摘 要 通过对串行 EEPROM 的指令时序和微机并行打印机适配器的介绍分析,本文给出了一种串行 EEPROM 读写器的设计方法,该读写器具有硬件线路简单,软件设计容易,操作者使用方便的优点。

关键词 串行 EEPROM 读写器 并行打印机适配器 指令时序

1 引言

EEPROM 即电可擦除可编程只读存储器,目前在各种智能化仪器仪表中得到广泛的应用。和 RAM 相比,EEPROM 具有掉电后存储信息不会丢失的优点;和 ROM 相比,EEPROM 又具有可重写入存储信息的优点。串行 EEPROM 其存储容量一般远小于并行 EEPROM,但是由于采用了单线数据位流的串行输入输出,大大减少了串行 EEPROM 在应用系统中和其它部分的互连线。由于串行 EEPROM 的上述特点,它特别适用在智能化仪器仪表中存储一些为数不多的预设的数据参数,如电子锁中用户密码的存储,BP 机内地址码的存储,IC 卡中用户数据的存储等等。因此,有必要设计出串行 EEPROM 的读写器以方便地对串行 EEPROM 进行读写操作,下面将重点结合 AT93C46,讨论如何设计串行 EEPROM 的读写器。

2 串行 EEPROM AT93C46 的引脚功能和指令时序

AT93C46 是美国 ATMEL 公司推出的串行 EEPROM,采用 CMOS 工艺制作,可靠性高,可重复使用 100,000 次,三线串行接口,具有内部写时钟,容量为 1K 位。对内部存储单元的访问以寄存器为单位,寄存器有 16 位数据位,每个寄存器都有唯一的地址,地址位数为 6 位。AT93C46 的引脚如图 1。

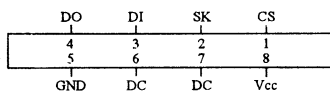


图 1 AT93C46 的引脚图

各个引脚的功能定义如下:

1. CS:片选信号,在指令操作期间,CS 保持高电平,在连续两条指令之间,CS 至少要保持 250ns 的低电平。

2. SK:时钟信号,为所有操作提供时钟基准,所有操作码、地址位、数据位均在 SK 的脉冲上升沿输入或输出。

3. DI:串行数据输入端,操作指令由此输入。

4. DO:串行数据输出端,在执行读指令时,寄存器内数据由此输出;在执行擦/写或片擦/片写指令后,DO 为 1 指示芯片忙,正在进行数据擦/写,DO 为 0 指示擦/写结束,可以执行下一条指令。

5. GND:所有信号公共地。

6. NC:无用管脚。

7. NC:无用管脚。

8. Vcc:接+5 伏电源。

AT93C46 提供 7 种操作指令,指令的格式为起始位+操作码位+地址位[+数据位],所有的位流信息均在 SK 时钟信号的上升沿输入或输出,每一位持续一个时钟周期,具体的操作指令格式见表一。对于 READ 指令,串行数据位流在地址位输完后下一个时钟周期开始在 DO 端输出;另外,在向指定寄存器写入数据前,必须先擦除该寄存器。

指令	起始位	操作码	地址位	数据位	说明
READ	1	10	$A_5 \sim A_0$		从指定寄存器中读数据
WRITE	1	01	$A_5 \sim A_0$	$D_{15} \sim D_0$	向指定寄存器中写数据
EWEN	1	00	11XXXX		允许写入或擦除数据
ERASE	1	11	$A_5 \sim A_0$		擦除指定寄存器中数据
ERAL	1	00	10XXXX		擦除所有寄存器中数据
WRAL	1	00	01XXXX	$D_{15} \sim D_0$	向所有寄存器中写入数据
EWDS	1	00	00XXXX		禁止写入或擦除数据

表一 串行 EEPROM AT93C46 的操作指令格式

对于不同容量,不同系列,不同公司生产的串行 EEPROM,虽然在芯片引脚,指令集和指令时序上会稍有差别,但其指令都是由一系列串行位流的输入输出所构成,这是所有串行 EEPROM 的共同特点。

3 读写器的设计

3.1 硬件设计

目前,微型计算机在各个单位得到广泛的应用,采用微机作为读写器的硬件设备,一方面无需额外的硬件设备的投资;另一方面,可充分利用微机丰富的软硬件资源,设计出简单易操作的读写器。用微机对串行 EEPROM 进行读写操作,就是要利用微机上空闲的 I/O 口的输入或输出来模拟出串行 EEPROM 的各条指令的时序。通过对微机硬件结构的研究可以发现,无需额外制作插板,利用微机的并行打印机适配器就可以很方便的实现微机和串行 EEPROM 的连接,通过对适配器 I/O 口的编程来实现对串行 EEPROM 的读写操作。

微机上的并行打印机适配器是用来驱动外设打印机的,它有三个 I/O 端口:数据端口,控制端口,状态端口。其中数据端口为输出端口,用来向打印机输出数据,口地址为 378H;控制端口为输出端口,用来向打印机发出控制命令,口地址为 37AH;状态口为输入端口,用来读出打印机的当前状态,口地址为 379H。这三个端口的部分位通过一个 25 针 D 型插座引出。对于输出端口,向端口各位写入 1 或 0,则在相应插座引脚出现 +5V 或 0V 的高低电平;对于输入端口,引脚的高低电平状态则反映在端口的相应位上。适配器的三个端口的各位和 D 型插座的引脚对应关系如表二所示,其中 D 代表数据端口,C 代表控制端口,S 代表状态端口,GND 为公共地。

引脚序号	1	2—9	10	11	12	13	14	15	16	17	18—25
说明	C ₀	D ₀ —D ₇	S ₆	S ₇	S ₅	S ₄	C ₁	S ₃	C ₂	C ₃	GND

表二 并行打印机适配器的 25 针 D 型插座各引脚的定义

有了以上对打印机适配器的知识做准备,读写器无需复杂的硬件电路设计,只需制作简单的插座及连线插头,利用打印机适配器的 25 针 D 型插座把串行 EEPROM 和微机相连,对串行 EEPROM 读写时换上插头及连线,然后运行读写器的软件,使用完毕后换回打印机插头即可。图 2 是串行 EEPROM AT93C46 和 25 针 D 型插座的一种连线方式。

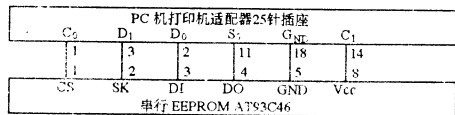


图2 串行EEPROM AT93C46与并行打印机适配器插口连线方式

3.2 软件设计

1. 指令时序的软件模拟

以串行 EEPROM AT93C46 为例,对其 7 条指令进行分析后可以看出,每条指令都由以下基本操作重复构成:(1)芯片使能;(2)芯片非使能;(3)一位串行数据的输入;(4)一位串行数据的输出。那么,利用 C 语言对 I/O 口进行访问,对于图 2 所示的连线方式,可以

方便的设计出实现这些基本操作的函数,分别介绍如下:

① 芯片使能和非使能

```
void CS(int i)
{
    if(i==1)/* i 为 1,Vcc 和 CS 均为高电平 */
        outportb(0x37a,0x03);/* 芯片使能 */
    if(i==0)/* i 为 0,Vcc 为高电平,CS 为低电平 */
        outportb(0x37a,0x02);/* 芯片非使能 */
    delay(.);/* 延时 1ms */
}
```

② 1 位串行数据的输入

```
void DI(int i)
{
    if(i==1)/* DI 端输入 1 */
        outportb(0x378,0x01);/* SK 为 0,DI 为 1,时钟负半周期 */
        delay(1);
        outportb(0x378,0x03);/* SK 为 1,DI 为 1,时钟正半周期 */
        delay(1);
    }
    if(i==0){
        outportb(0x378,0x00);/* SK 为 0,DI 为 0,时钟负半周期 */
        delay(1);
        outportb(0x378,0x02);/* SK 为 1,DI 为 0,时钟负半周期 */
        delay(1);
    }
}
```

③ 1 位串行数据的输出

```
int DO(void)
{
    unsigned char temp;
    outportb(0x378,0x00);/* SK 为 0,时钟负半周期 */
    delay(1);
    outportb(0x378,0x02);/* SK 为 1,时钟正半周期 */
    temp=inportb(0x379);/* 读出 DO 端状态 */
    delay(1);
    if(temp&0x80)return 1;/* DO 为高电平,返回 1 */
    else return 0;/* DO 为低电平,返回 0 */
}
```

有了上面这三个函数,串行 EEPROM AT93C46 的各条指令都可以通过适当调用这三个函数来实现,具体实现各条指令的 C 语言程序这里就不再介绍。

2. 界面设计和参数传递

对于实际应用中的串行 EEPROM,其指定寄存器中的特定数值都有其实际的物理意义。对于读写器的操作者来讲,他所知道的只是他所读出或写入的参数,

应用于机车车辆滚动振动试验台的监控系统

吴学杰

(西南交通大学牵引动力国家重点实验室 成都 610031)

摘要 本文系统地介绍了机车车辆滚动振动试验台监控系统的硬、软件结构,并对其中一些检测方法和软件技术作了详细的分析。

关键词 监控 中断 检测

0 引言

牵引动力国家重点实验室所建造的机车车辆滚动、振动试验台可在室内实现机车车辆、转向架及轮对在轨道上进行的 1:1 动态模拟试验,具有规模大结构复杂等特点,主要包括 4 轴动力驱动控制系统、16 通道可控液压伺服控制系统、动态试验数据测试、润滑、工业电视监视等,各部分相对独立,完成各自的控制、测试任务。为使各部分有机地结合在一起,完成复杂的组合试验任务,我们设计了一套计算机监控系统来对各子系统进行协调、管理、控制和实时监测,以确保试验的顺利进行。

1 系统构成

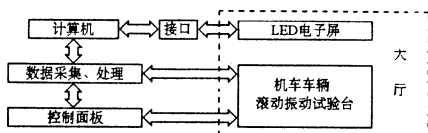


图 1

收稿日期:1996年7月13日

而无需了解这些参数在串行 EEPROM 中的存储形式及地址,也无需了解各条指令的时序,所有的这些细节都要被屏蔽起来,这就涉及到界面的设计和参数的传递问题。读写器的用户界面用 FOXPRO 来设计,用菜单指引操作者选择所要进行的操作,用对话框指引操作者进行参数设置;对串行 EEPROM 的读写由于涉及 I/O 口操作,用 C 语言来实现比较方便。二者之间通过数据库来传递参数。读写器软件的主流程如下:

(1) 菜单选择:是读码转(2),写码转(4),退出转(7)。

(2) 调用 C 语言读码程序:读出串行 EEPROM 中存储数据;进行数据格式转换;将数据按照所代表的参数意义写入数据库;返回。

(3) 打开数据库,在屏幕显示各项参数的设置情况;转(1)。

(4) 弹出对话框,进行各项参数设置;将参数设置情况写入数据库中。

(5) 调用 C 语言写码程序:读出数据库中的参数

该系统由 ComPAQ386 微型计算机、数据采集、处理、LED 电子屏、控制面板和相应的软件系统构成,如图 1 所示。

(1) 数据采集、处理

数据采集通道配置如下:32 通道的模拟量输入,14 个计数/定时通道(其中 12 个用于外部脉冲量输入,2 个用于定时中断),24 个开关量输入通道和 16 个开关量输出通道。

模拟量输入通道主要用于监测试验台各高速轴承的温度,关键部位的位移、加速度等(其中温度和位移具有超限报警功能),12 通道的脉冲量输入通道用于测量滚轮的转速及电机与滚轮之间传递的扭矩,开关量输入通道用于监测试验台各子系统和主要设备的工作状态、控制面板的操作指令和各类故障报警信号。控制指令通过开关量输出通道去控制各被控对象。

1.1 模拟量检测

本系统中模拟量的测量包括:轴温、位移、加速度等,各类传感器或二次仪表的输出信号均为电压信号,这些信号用约 15m 的双芯屏蔽电缆送到模拟量输入接口板,供测量用,其接口板输入信号范围为 $\pm 5\text{V}$ 。所

设置信息;数据格式转换;向串行 EEPROM 中写入数据;返回。

(6) 转(1)。

(7) 返回 DOS。

4 结束语

对于不同系列不同容量的串行 EEPROM,均可设计出和微机合适的连线方式,通过对 I/O 口的操作,用软件来模拟出各条指令的时序,对串行 EEPROM 进行读写操作;同时可利用微机丰富的软件资源,根据其在应用中的需要,设计出易于操作的用户界面。基于上述设计思想设计出的读写器,具有硬件线路简单,软件设计容易,操作者使用方便的优点,有一定的推广价值。

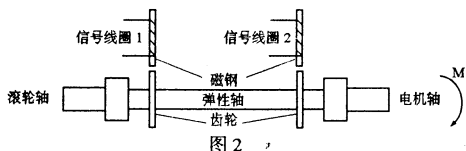
参考文献

- 1 杨一民,王剑秋. 93C 系列串行 EEPROM 的使用. 上海:电子技术 1995(5)P22
- 2 张载鸿. 微型机(PC 系列)接口控制教程北京:清华大学出版社 1992 P188

有信号均采用差分方式输入,以避免长线传输入带来的共模干扰。

1.2 脉冲量的检测

脉冲量输入接口板用于测量滚轮的转速和所传递的扭矩,所采用的传感器为转速、扭矩合成式传感器,其工作原理如图 2 所示。弹性轴(安装在电机轴和滚轮之间)的两端装有两只齿轮,齿轮上方分别安装有两条磁钢,磁钢上各有一信号线圈。弹性轴转动时,信号线圈中分别感应出同频率的脉冲电势信号,当外加力矩为零时,两电势信号相位差为零,外加力矩时,弹性轴产生扭转变形,在弹性范围内(传感器测量范围内),其外加力矩大小与两电势信号的相位差成正比,而两电势的频率与转速和齿轮齿数(为 120 齿)的乘积成正比。基于以上测量原理采用如下测量方法:



1.2.1 转速

转速测量中,采用 M 测速法,即将传感器送来的两路脉冲信号分别接入 3 路计数器中,如图 3(a)所示,通过定时器 1 每隔 1 秒钟向 CPU 发出一次中断申请,CPU 响应中断后,从计数器 1 中读出计数值 M,并将计数器 1 清零,根据下式计算转速

$$N = 60 \cdot M / 120 (\text{转/分})$$

$$= \frac{M}{2} (\text{转/分})$$

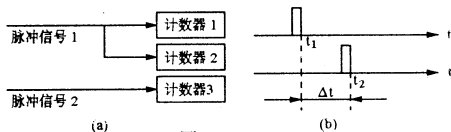


图 3

1.2.2 扭矩

图 3(a)中,计数器 2 和计数器 3 设定为计数结束中断模式,其计数初值为 1,无自动重装能力。在转速测量的定时中断服务程序中开放计数器 2 的中断请求,经一个脉冲上升沿和下降沿后,计数器 2 向 CPU 发出中断请求,CPU 响应中断后,立即读取系统时钟 t_1 并开放计数器 3 的中断请求,在脉冲信号 2 的一个上升沿和下降沿后,计数器 3 向 CPU 申请中断,在其中断服务程序中读取系统时钟 t_2 后返回,这样就获得了信号 1 和信号 2 相邻两脉冲下降沿之间的相位差。 $\Delta t = t_2 - t_1$,如图 3(b)所示,结合传感器出厂时的系数 K 和此时的转速 N,可求得扭矩:

$$M = K \cdot \Delta t \cdot N$$

1.3 开关量的检测和输出

24 个开关量检测包括:各子系统、主要设备的状

态检测,操作面板各开关状态及各子系统对控制指令的执行情况,试验大厅各操作按钮(供大厅工作人员巡检用)的状态检测等,一旦某一状态出现故障,立即报警,其信号传输线路如图 4(a)所示,监控系统对各子系统的开关控制指令是通过图 4(b)所示的传输线路来控制被控对象的。

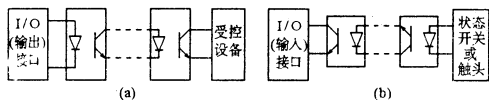


图 4

(2) LED 电子屏

悬挂于大厅一侧上方的电子显示屏(3.5m × 1.4m)可使大厅工作人员及时了解控制中心发出的各种控制指令、信息、试验进程和重要试验参数,以确保工作人员和试验过程的安全。

LED 电子显示屏及其接口如图 5 所示,通过插在主机槽内并与 VGA 显示卡相连的接口卡获取 VGA 显示器指定区域(与电子屏对应)的数据、时钟和行场信号后送入控制器中,经控制器处理后,分场将数据送到驱动器,点亮相应的 LED 显示单元,从而实现数据或信息在 LED 电子屏和 VGA 显示器上指定区域内的同步显示。

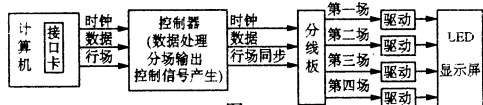


图 5

2 软件系统

本软件以金山 SPDOS NT1.0 汉字系统为工作平台,采用 Visual C++ 语言编写而成。由主程序、数据采集和处理、键盘输入、显示 4 个功能模块组成,其中第二和第三个模块以中断服务子程序的方式实现,以保证系统的实时性。

主程序用于整个系统进行初始化,数据采集、处理用于实时监测试验系统重要的反馈信号,同步跟踪、显示和执行(输出控制)操作人员对试验进程的控制,完成试验进程中,各种异常情况和故障的检测,并根据具体情况显示相关信息,产生不同级别的报警信号,键盘输入模块完成人一机对话和文字指令键入任务,显示模块可将相应的数据、信息和指令以字符、文字和曲线等形式反映在 VGA 显示器上。下面,对主要功能模块作一些介绍。

2.1 主程序

主程序比较简单,主要完成各 I/O 端口初始化,设置各中断服务子程序的中断向量等,其框图如图 6 所示。

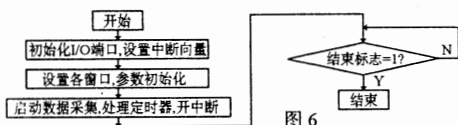


图6

2.2 显示模块

监控系统中采用VGA显示器对整个试验系统中的重要数据、指令、文字命令、试验进程进行监视。显示器上设置有LED显示、模拟示波器、预显、操作提示4个窗口,如图7所示。

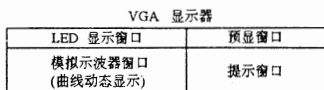


图7

模拟示波器以波形的形式动态显示反馈信号的变化(反馈通道号由用户通过键盘输入模块选择),预显窗口用于显示用户键入的汉字命令,并可在此窗口进行修改,确认无误后,可将此窗口内的信息在LED窗口显示(同时在大厅电子屏上显示),操作提示窗口作为用户的人—机对话界面,以实现不同的键盘输入功能。LED窗口与大厅电子屏相对应,同步显示图8所示的内容。

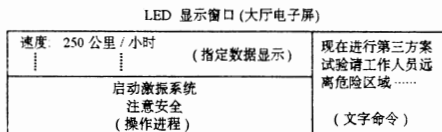


图8

其中数据显示部分实时显示试验台一些主要反馈参数,如转速等,操作进程部分以汉字形式显示试验的操作进程和步骤,文字命令部分显示控制中心操作人员向大厅发出的文字命令(数据来自预显窗口)。

2.3 键盘输入模块

键盘输入模块是以中断服务子程序的方式实现的,在主程序的初始化中,已将键盘输入中断处理权截留给用户,因此,一旦键盘有输入,立即执行本键盘中断服务子程序,其程序框图如图9所示。

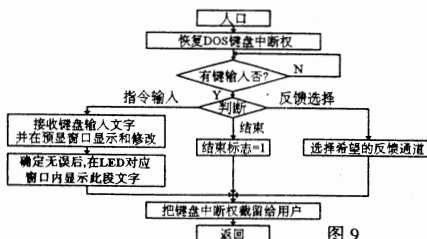


图9

2.4 数据采集、处理模块

本模块完成对模拟输入信号、开关量输入信号及脉冲量输入信号的采集和处理,并将控制指令通过开关量输出通道控制各子系统。其程序框图如图10所示。

本模块是通过中断嵌套来完成的(定时器1定时中断、计数结束中断)。

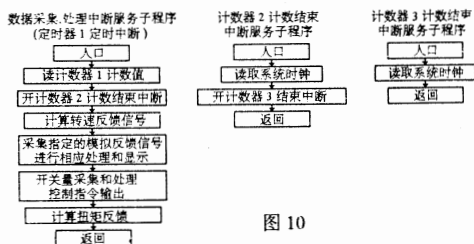


图10

3 系统抗干扰措施

由于试验台规模大、布局复杂,在监控系统周围15m内存在着种类较多的干扰源,如可控硅整流、开关电源、大功率电——液伺服系统等,这些设备与计算机系统共用一个电源,不可避免地带来干扰,我们在系统设计时采取以下措施来克服这些干扰。

- (1)采用开关稳压电源作为监控系统的电源,以提高系统电源的质量。
- (2)所有信号传输线均采用双绞屏蔽线,以防止空间电磁干扰。
- (3)所有信号均进行隔离后再进入计算机。
- (4)采用专用地线。
- (5)采用数字滤波技术,可有效地提高数据采集的可靠性。

(6)软件设计中采用“陷阱”技术有效地防止了干扰引起的“飞程序”和“死机”故障(利用定时器2)。

4 结束语

本监控系统将试验台各关键部位的反馈信号及各主要设备的状态以汉字、数据、动态曲线等形式生动地反映在控制中心的VGA显示器和大厅电子屏上,信号的变化完全由计算机监测,并具有超限自动报警功能,大大减轻了操作人员的工作强度并使操作人员对异常情况和故障能及时进行处理,为机车车辆滚动振动试验台的正常运行提供了有力的保证。

参考文献

- 1 SPDOS NT 编程指南. 求伯君主编. 科学出版社
- 2 微计算机检测技术及应用. 王俊省主编. 电子工业出版社

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 邮码:630015
电话:(0811)3876725、3876706

利用 DCS 打印口实现数据通讯

卢 苇 李志蜀

(四川联合大学计算机系 成都 610000)

那思佳

(抚顺腈纶化工厂计算中心 抚顺 113000)

卢 军

(抚顺市计算机技术研究所 抚顺 113000)

摘 要 DCS 集散控制系统以其准确可靠和便于管理, 在我国的各行各业尤其是石化行业得到广泛应用。该厂的丙烯腈和腈纶生产装置都引进了 DCS 系统。由于资金等方面的原因没有引进用于管理的上位机。因此, 使用 PC 机接收来自打印机接口的定时打印数据, 进而实现实时调度系统, 对生产的调度和管理具有重要的现实意义。

随着厂办公楼内的计算机网络安装完成, 即可以在厂领导和各处室的计算机上看到腈纶装置的生产流程实时画面, 为生产、经营和科研服务。

本文是对这项工作的一個总结, 希望得到专家、同行的帮助和建议, 以便不断改善系统。同时也希望本文所使用的方法能为其它 DCS 系统的开发工作提供一个线索和思路, 使 DCS 系统发挥出更大的作用。

关键词 DCS 打印口 数据通讯

1 概述

该厂的腈纶装置使用美国 FOXBORO 公司的 I/A 智能控制系统, 这套系统的通讯处理机 (CP) 提供了四个 RS-232 标准异步通讯口, 连接打印机等外围设备。同时, DCS 提供了定时打印功能, 通过系统设置可以定时 (实验时我们采用 5 分钟间隔) 打印实时数据。IBM-PC 系列微机也使用 RS-232 接口用于微机系统之间的互连, 或与串行打印机、绘图仪等外围设备通讯。通过实验我们发现可以使用微机模拟串行打印机接收定时“打印”数据, 由于腈纶 DCS 与厂办公楼距离较远, 所以我们在中间使用一对光电隔离长线驱动器和电话线连接。连接方法如图 1.1 所示。

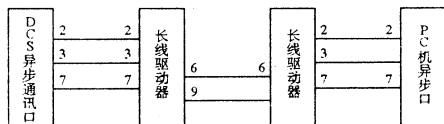


图 1.1 DCS 异步通讯口与 PC 机连接示意图

在用于接收的微机上使用汇编语言编制一个接收程序, 程序将接收到的数据处理并存入网络服务器的



图 1.2 实时调度系统网络结构图

文件服务器中。在网络工作站上使用流程图程序绘制

腈纶装置的流程图, 并且将保存了网络服务器上的实时数据, 显示在流程图的相应位置上, 实现实时调度系统的动态画面。网络结构如图 1.2 所示。

在整个工作中, 最为关键也是最为复杂的工作是 DCS 实时数据接收, 因为缺少先例和资料, 我们无法对 DCS 编程, 只能使接收程序适应 DCS 的规程; 出现错误也无法反馈到 DCS, 只有接收程序质量和速度要求高; 接收程序需要使用文教昭 INS8250 芯片编程, 编写中断服务程序等高级编程技术。

主程序的框图如图 1.3 所示, 中断服务程序将在本文第二部分给出。并在以后部分中对接收程序各部分做出详细叙述。

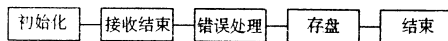


图 1.3 接收主程序框图

2 实时数据的接收

DCS 系统提供了定时打印功能, 根据不同的实时性需要可以在 DCS 上设置不同时间间隔的定时打印, 这样 DCS 可以定时向打印机接口传递实时数据。FOXBORO 公司的 I/A 系统使用的打印机接口为 RS-232 25 脚口型连接器, 通过编程可以使一台 PC 机代替打印机接收实时数据。

IBM-PC 系列各类微机的串行口都采用 RS-232 异步通讯标准, RS-232 的口型连接器有 25 个引脚, 其中 20 个引脚已被 RS-232 标准定义, 而一般只

使用其中 9 个,各引脚的定义见表 2.1。

2	TD	发送数据
3	RD	接收数据
4	RTS	请求发送
5	CTS	清除发送
6	DSR	数据装置就绪
7		信号地
8	DCD	载波检测
20	DTR	数据终端就绪
22	R1	振动铃指示

表 2.1 RS-232 主要引脚定义

IBM-PC 系列机的 RS-232 接口,是以可编程的 INS8250 为核心的,它使 PC 机有能力同标准 RS-232 接口通讯,INS8250 内部有 10 个寄存器,IBM-PC 为该端口提供的地址为 3F8H~3FFH 和 2F8H~2FFH 这两个端口,分别称为异步通讯 1 和异步通讯 2。本文使用异步通讯 2。

8250 芯片内部寄存器 I/O 地址如表 2.2。各寄存器意义这里不再赘述。

I/O 口地址	寄存器名称	功 能
3F8H *	发送保持寄存器	输出
3F8H *	数据接收寄存器	输入
3F8H **	波特率因子 (LSB) ***	输出
3F9H **	波特率因子 (MSB) ***	输出
3F9H *	中断允许寄存器	输出
3FAH	中断标志寄存器	输入
3FBH	线路控制寄存器	输出
3FCH	MODEM 控制寄存器	输出
3FDH	线路状态寄存器	输入
3FEH	MODEM 状态寄存器	输入

* 线路控制寄存器第 7 位 (DALB)=0

** DALB=1

*** 波特率因子即除数锁存器

表 2.2 8250 端口地址

本文采用 INS8250 中断请求方式编写程序,在初始化中需要做如下工作:

1、允许 8259 中断优先级为 3。

2、将 8250 中断程序入口地址赋给中断向量单元 OBH * 4。

3、8250 初始化 (波特率,通讯格式等)

4、根据需要,设置中断允许寄存器相应位的状态。

初始化后,一旦 8250 发生中断请求 ZRQ3,在允许中断的情况下,便可执行接收实时数据的中断处理程序,程序的初始化部分如下:

```
cli
mov ax 0
mov es ax
mov di
mov ax offset receive
cld
stosw
```

```
mov ax,0
mov es,ax          mov dx,2fbh
mov di,0bh * 4     mov al,0ch
mov ax,offset receive out dx,al
cld                mov dx,2fch
stosw              mov al,0bh
mov sx,cs          out dx,al
stosw              mov dx,2f9h
in al,21h          mov al,01h
and al,0f8h        out dx,al
out 21h,al
```

中断处理程序的任务是当异步通讯中接收到一字节数据,引起 8250 的 IRQ3 中断后,把接收到的数据依次存入缓冲区。由于本程序采用 9600 波特传输,对中断处理程序的执行速度要求较高,所以尽可能减轻中断处理程序的工作。流程见图 2.1。

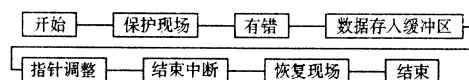


图 2.1 中断处理程序流程图

这样,可以将 DCS 打印机接口发生的数据按次序保存在数据缓冲区 buffer 中,以备存盘和错误处理使用。

3 错误处理

DCS 打印机接口,只能接收诸如打印机忙,缺纸等打印机发生的信号。因此无法将接收数据情况反馈给 DCS 系统。为了提高传输速度以提高实时性,我们采用 9600 波特的速率,传输的错误是难以避免的。我们初步统计出错率大概在 3% 左右。

如果将接收的数据直接存盘使用,即使在接收过程中丢失一个数据位,那么后面的所有数据都附“错位”,使数据接收出错。

为了解决这个问题,我们采用了每两个数据之间增加三个“*”的方法来作为数据的间隔。通过检查“*”之间的数据位数是否为数据长度来确定传输过程中是否出现丢位。我们接收的数据长度是 10 位。采用三个“*”作为数据间隔的原因是即使在传输中丢失一个或二个“*”用以间隔数据,而三个“*”同时丢失的概率是 3% 的立方,大约是亿分之一。因此可以满足要求。

在数据传输的前面,有一些诸如时间,打印机控制等信息,我们采用动态的方法从确定数据信息的开始位置。具体的做法是查找由中断接收程序存于数据缓冲区 buffer 中数据出现的第一个“*”,在此之后出现的数字信息即为数据。如得到的“*”之间的数据信息长度 (rec-len) 等于 10,说明接收正确,将数据分组出来写入文件缓冲区 buff 中,如果 rec-len 小于 10,说明接收数据出现“丢失”,只调整两个缓冲区的指针,而

Turbo c 文本方式中文下拉式菜单的设计

胡诗国

(佛山科技学院 佛山 528231)

宋 威

(南华工商学院 广州 510000)

摘 要 本文通过调用 Turbo c 提供的 BIOS 中断函数 int86() 的子功能, 介绍用 C 语言设计的文本方式中文下拉式菜单。此菜单具有速度快、模块小、人机界面友好等特性。

关键词 文本方式 中文 菜单 人机界面 C 语言 程序设计

1 中文菜单设计思路

目前, 使用 C 语言编制的下拉弹出式菜单已风靡整个软件市场。下拉弹出式菜单的完善性和友好的人机界面越来越受到世人的青睐。用 Turbo 2.0 设计的西文下拉式菜单一般工作在文本方式下, 而 C 语言所编的中文菜单大多使用计算机图形方式, 即先用 C 语言的图形驱动程序进行初始化, 然后再读汉字点阵字模库。通过在屏幕上描像素点来绘制图形汉字, 这样速度慢、效率低、模块大。现给出一种汉字系统文本方式下拉式菜单的程序, 供用户使用, 也可供 C 语言爱好者参考。

程序用 Turbo C 2.0 编译, 生成约 15k 大小的 EXE 可执行文件。在 UC DOS 汉字系统下运行。

收稿日期: 1996 年 7 月 19 日

不保存数据, 如果 rec-len 大于 10, 说明“*”丢失, 做两次文件缓冲区指针调整。

至此可以在文件缓冲区中得到依次排列的数据信息, 而信息的总长度保存在 buff 中, 为数据存储做准备。

4 数据的存储

为了实现实时数据的共享, 我们将接收, 排错得到的数据, 存储于网络服务器上, 我们使用美国 DEC—NET 计算机局域网, 服务器使用 VAX—4200 小型计算机。使用 DEC—NET 的文件服务器功能, 数据不仅可以其它 PC 工作站使用, 也可以被 VAX—4200 小型计算机使用。PC 机上我们使用 PCSA3.1 作为网络软件, 它可以保证程序正确地使用 DOS 的句柄或文件/设备读写系统功能。

句柄方式文件管理在要建立或打开一个文件时准备一个 ASCII—Z 字符串, 以 DS:DX 批明 ASCII—Z 字符串的位置, 用 3CH 或 3DH 号系统功能建立或打开此文件, 若成功, DOS 在其内部为文件建立一个控制区, 并由 AX 返回一个 16 位的句柄, 只要将句柄放入 BX 中, 就可以使用 2FH、40H、42H 号系统功能来完成文件的读、写、关闭和定位等操作。

存盘的具体过程是: 利用字符串“M:D.DAT”,

菜单程序具有以下特点:

(1) 利用 Turbo C 2.0 提供的 DOS、BIOS 中断功能函数 int86() 来修改 CCDOS 汉字显示的寄存器参数串, 从而编制成屏幕显示汉字子函数 dispzh();

(2) 利用 BIOS 中断 10H 的子功能号 2 移动光标, 再用子功能号 8 取得当前光标处的字符, 最后用子功能号 9 将取得的字符按指定属性重新显示在当前光标处, 从而编制成按属性显示指定区域字符的子函数 attri();

(3) 通过显示背景色空格符与显示彩色前景色汉字的交替变换, 来达到移动菜单的效果;

(4) 利用 Turbo c 2.0 的 bioskey() 函数来测试键盘上的特殊键并返回 ASCII 和扩展 ASCII 的值, 从而使用方向箭头键、回车键、ESC 键及组合功能键, 以便达到控制菜单移动、确认和退出等目的。

OOH 和 3CH 号系统功能在网络文件服务器上建立文件 D.DAT, 使用得到的句柄将文件缓冲区 buff 中的 2970 个字节内容写入 D.DAT 中, 之后关闭文件。

考虑到今后系统的完善, 除了保存实时数据, 我们还保存了 48 个小时的历史数据, 历史数据除包括数据信息还包括时间信息。基本过程与实时数据的存盘过程一致, 所不同的是在打开文件后, 需要将指针定位在正确的位置上, 这是通过对时间信息检索实现的。用当前的实时数据代替当前的“最老”的数据信息。

5 系统的应用与展望

目前, 我们使用南京自动化所开发的一组程序, 使用 D.DAT 中的实时数据可以在网络工作站上看到腓纶装置的动态流程图画面, 为领导和各处室指挥生产和了解生产情况提供快速、准确的信息。同时成本大大低于上位机和增加 CRT。

由于流程图是在 PC 机实现的, 汉字系统功能强大, 流程图的修改也十分容易, 可以按照使用者的要求更改画面, 使这更为直观, 对于不太熟悉英文的人来说, 尤为重要。

由于保存了历史数据, 我们计划逐步开发历史趋势图等项目, 以便于指挥生产。同时可以对这些数据加以研究, 从而发现不同的工艺指标对产品产量和质量以及消耗的影响, 提高生产装置的经济效率。

程序取名为 CMENU.C, 在 Turbo c 2.0 集成环境下编译成 CMENU.EXE 文件, 然后进入 UC DOS 汉字系统, 打入文件名并回车。则在屏幕中央显示汉字菜单, 且第一条用鲜明色彩显示。此时, 可用上下箭头移动菜单来寻找用户选项, 再用回车键呼出下拉菜单, 并且用上下箭头移动子菜单中的目录, 再用回车键确认, 即可选中所需条目, 在弹出菜单时, 可用 ESC 键清除弹出菜单, 而显示主菜单。这种友好的界面风格, 给中文用户带来便利。

程序中首先用 DOS 命令 CLS 清屏, 接下去的 rect() 函数是本程序的主要函数, 用它在屏幕上显示菜单(中文、西文均可)。这个函数有 4 个参数, 即字符串指针数组 *P、实参指针数组中元素的个数 num、字符属性 atr 和存储菜单在屏幕上显示之行、列位置的数组 inf[4] (其中 inf[0] 存储起始行、inf[1] 存储终止行、inf[2] 存储起始列、inf[3] 存储终止列)。此函数靠反复调用 disphz() 和 atri() 两个函数来实现。画矩形框函数 kuang() 靠调用 atri() 函数来实现。擦除字符函数 erase() 靠调用 disphz() 函数来实现。而光标定位函数 gotoxy() 则是 disphz() 函数的主要调用对象。up() 和 down() 两个函数分别是向上、向下移动菜单指针的函数, 均有 4 个参数。即指针初值 i、当前菜单在屏幕上的行列位置数组 inf、当前菜单的字符属性及指针的字符属性。此两函数稍作修改, 可成为向左、向右移动菜单指针的函数。

在菜单程序中, 当移到某选项通过回车确认后, 连到调用 self(), 这确认项转去执行的模块, 有主菜单号 x 和子菜单项 i 两个参数传递, 据此就可去计算所需内容, 此函数需用户根据需要自己编写。当然用户在改编成自己的实用菜单时, 还需将前面的指针数组初值改成适当的汉字。

2 中文菜单设计程序

程序如下:

```
#include "string.h"
#include "dos.h"
#include "conio.h"
union inkey{
    unsigned char ch[2];
    int ii;
};
cc;
void atri(), kuang(), erase();
main()
{int i, jj, i0, k, inf[4], inf1[4], inf2[4];
char *f[]={"系统", "家庭财务", "名片管理", "备忘录", "图书资料", "退出系统"};
char *f0[]={"系统程序 1", "系统程序 2", "系统程序 3", "系统程序 4", "系统程序 5"};
char *f1[]={"家庭财务程序 1", "家庭财务程序 2", "家庭财务程序 3", "家庭财务程序 4", "家庭财务程序 5", "家庭财务程序 6"};
```

```
char *f2[]={"名片管理程序 1", "名片管理程序 2", "名片管理程序 3"};
char *f3[]={"备忘录程序 1", "备忘录程序 2", "备忘录程序 3", "备忘录程序 4"};
char *f4[]={"图书资料程序 1", "图书资料程序 2", "图书资料程序 3", "图书资料程序 4", "图书资料程序 5", "图书资料程序 6", "图书资料程序 7"};
inf1[0]=2; inf1[1]=22; inf1[2]=2;
inf1[3]=76;
system("cls"); /* 清屏 */
kuang(119, 23);
/* 用指定颜色画屏幕矩形框 */
rect(6, f, 126, inf);
/* 在屏幕中央写主菜单 */
atri(inf[0], inf[0], inf[2], inf[3], 74);
/* 用鲜明色彩显示第一条主菜单 */
i=0; jj=0; i0=0;
cc.ch[1]=-1;
while(cc.ch[1]!=45) /* 主循环 */
{cc.ii=bioskey(0);
/* 将按键值读入一联合中 */
if(cc.ch[0]){
/* 如果低位字节不为 0 */
switch(cc.ch[0]){
/* 判断低位字节 */
case 13: if(i==inf[1]-inf[0])
(cc.ch[1]=45; break;
if(jj==0){
erase(inf, 23);
if(i==0)rect(5, f0, 126, inf2);
else if(i==1)rect(6, f1, 126, inf2);
else if(i==2)rect(3, f2, 126, inf2);
else if(i==3)rect(4, f3, 126, inf2);
else if(i==4)rect(7, f4, 126, inf2);
atri(inf2[0], inf2[0], inf2[2],
inf2[3], 74);
gotoxy(23, 56);
printf("按<Esc>键返回上级菜单");
jj=1;
/* else self(x, i); 调用实际相接模块 */
break;
case 27: if(jj==1)
{jj=0; i0=0; erase(inf1, 23);
rect(6, f, 126, inf);
atri(inf[0]+i, inf[0]+i, inf[2],
inf[3], 74);
}
break;
}
else{
switch(cc.ch[1]){/* 如果低位字节为 0 */
case 72: if(jj==0)i=up(i, inf, 126, 74);
else i0=up(i0, inf2, 126, 74);
break;
case 80: if(jj==0)i=down(i, inf, 126, 74);
```

```

else i0=down(i0,inf2,126,74);
break;
default: break;}
})
system("cls");
void gotoxy(x,y)/*光标定位*/
int x,y;
{union REGS r;
r.h.ah=2;r.h.dl=y;r.h.dh=x;
r.h.bh=0;int86(16,&r,&r);
}
void dispzh(x,y,p,atri)
/*用DOS中断在屏幕写汉字*/
int x,y,atri;
char *p;
{union REGS r;
register int i,j;
for(i=y;*p;i++)
{gotoxy(x,i);r.h.ah=9;r.h.bh=0;
r.x.cx=1;r.h.al=*p++;
r.h.bl=atri;int86(16,&r,&r);
}
}
void atri(row1,row2,coll,col2,atri)
/*按属性显示指定区域的字符*/
int row1,row2,coll,col2,atri;
{int i,k,x,y;
char j;
union REGS r;
r.h.ah=3;r.h.bh=0;int86(16,&r,&r);
x=r.h.ah=2;y=r.h.dl;
for(k=row1;k<=row2;k++)
for(i=coll;i<=col2;i++)
{r.h.ah=2;r.h.dh=k;r.h.dl=i;
r.h.bh=0;int86(16,&r,&r);
r.h.ah=8;r.h.bh=0;
int86(16,&r,&r);j=r.h.al;
r.h.ah=2;r.h.dh=k;r.h.dl=i;
r.h.bh=0;int86(16,&r,&r);
r.h.ah=9;r.h.bh=0;r.h.bl=atri;
r.h.al=j;r.x.cx=1;
int86(16,&r,&r);
}
r.h.ah=2;r.h.dh=x;r.h.dl=y;
r.h.bh=0;int86(16,&r,&r);
return;
}
void kuang(atril,atri2)
/*按指定颜色进行屏幕显示*/
int atril,atri2;
{atri(0,0,0,79,atril);
atri(24,24,0,79,atri2);
atri(1,23,0,1,atril);
atri(1,23,78,79,atri2);
atri(1,23,2,77,atri2);

```

```

}
int rect(num,p,atr,inf)
/*按指定属性显示菜单*/
int num,atr,inf[4];
char *p[];
{int row,col,i,j=1;
for(i=0;i<num;i++)
if(strlen(p[i])>j)j=strlen(p[i]);
row=(24-num)/2;col=(80-j)/2;
for(i=0;i<num;i++)
dispzh(row+i,col,p[i],atr);
atri(row,row+num-1,col,col+j-1,atr);
atri(row+num,row+num,col+1,col+j,0);
atri(row+1,row+num-1,col+j,col+j,0);
inf[0]=row;inf[1]=row+num-1;
inf[2]=col;inf[3]=col+j-1;
gotoxy(0,0);
}
void erase(inf,atri)/*擦除汉字*/
int inf[4],atri;
{char *cz=" ";
int i,j;
for(i=inf[0];i<=inf[1]+1;i++)
for(j=inf[2];j<=inf[3]+2;j++)
dispzh(i,j,cz,atri);
gotoxy(0,0);
}
int up(i,inf,t1,t2)/*菜单指针上移*/
int i,inf[4],t1,t2;
{atri(i+inf[0],i+inf[0],inf[2],
inf[3],t1);
i--;
if(i<0)i=inf[1]-inf[0];
atri(i+inf[0],i+inf[0],inf[2],
inf[3],t2);
return(i);
}
int down(i,inf,t1,t2)/*菜单指针下移*/
int i,inf[4],t1,t2;
{atri(i+inf[0],i+inf[0],inf[2],
inf[3],t1);
i++;
if(i>inf[1]-inf[0])i=0;
atri(i+inf[0],i+inf[0],inf[2],
inf[3],t2);
return(i);
}

```

参考文献

- 1 谭浩强编著,C程序设计,北京:清华大学出版社,1994年11月第1版
- 2 叶欣编译,TURBO C 2.0 参考手册,北京:中国科学院高级电脑技术公司1990年5月第1版
- 3 李政人主编,电脑实用编程技巧与操作诀窍102例,成都:电子科技大学出版社1995年4月第1版

分布式控制系统并行通信的可靠性设计与差错检测纠正

李伟

(浙江丝绸工学院电子系 杭州 310033)

摘要 本文研究了分布式控制系统并行通信的可靠性设计与差错检测、纠正技术,分析了干扰信号的来源与抑制方法,叙述了传输协议的差错检测与纠正原理,给出了抗干扰的接口信号传输原理图及基本通信流程图。

关键词

1 引言

在以 PC 系列微机为主机, MCS—51 系列单片机为从机组成的分布式控制系统中,主从机之间的信息传递有两种方式,一是串行,二是并行。串行通信的特点是硬件具有经济性,在进行长距离数据传输时,可利用现成的电话线进行数据交换,而且 PC 机及单片机本身具有串行口,可用串行口直接联网,主要技术问题可由软件解决,因此,串行通信被广泛地应用于工业控制的微机系统中,串行通信中的可靠性、抗干扰性和检错、纠错技术较为成熟。但串行传输数据的速度较慢。并行通信的特点是按字节传送,数据传速高,传输距离近,通常不进行检错、纠错,而把接收到的数据直接使用,硬件没有串行通信那样经济。对于那些数据传速要求高,实时性强的系统,采用并行通信方式可解决传输的快速问题。但由于并行通信抗干扰能力弱,传输距离近,又不对传输的数据进行检错、纠错,再加控制现场的恶劣物理环境,使得并行通信中的差错成了致命的问题。一旦通信发生错误,就有可能使系统出现混乱状态,这极大地阻碍了并行通信方法的使用。本文结合硬件和软件,提出了解决并行通信差错的方法,在硬件上通过可靠性设计提高系统的抗干扰能力,在软件上通过传输协议来检错、纠错。

2 可靠性设计

微机系统一般性的可靠性设计问题,在许多论著中有较多的论述与研究,在本文中只作一般性的分析,重点是分析并行接口信号传输的可靠性设计。

2.1 干扰来源及其抑制

电源干扰。电源性能的优劣在很大程度上将直接影响系统的可靠运行,由于工业现场用电情况复杂,会在电网上产生高频振荡电压,经供电电源进入系统,给微机系统造成很大干扰。对电源干扰,可采用低通滤波器和隔离变压器来抑制,在电源上使用隔离变压器,一次和二次线圈间加静电屏蔽层,以断开干扰通路,在用电设备前侧,电源线上加带有屏蔽的罩的电源滤波器,以吸收大部分“毛刺”,抑制干扰进入微机系统。

空间的外来干扰。空间的外来干扰主要是电场、磁场和交变的电磁场干扰。对空间干扰主要是采取屏蔽

措施,计算机和外设用封闭的金属罩等位屏蔽接地,对供模干扰有较强的抑制作用。

地电流干扰。由于实际地电阻不为零,因此,在地线中有电流流过时会在各点造成电位差,干扰系统的正常工作。正确的接地是抑制干扰的重要方法。凡电气连接的各种地要采用“单点接地”。即模拟地与数字地,信号地与电源地应分开处理,最后接至一个总的系统。

信号通道的干扰。信号通道的干扰主要来自信号传输线,其抑制方法即是下面要叙述的。

2.2 并行通信接口的抗干扰设计

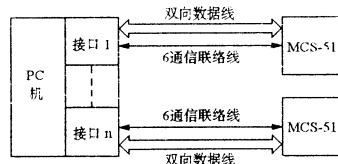


图1

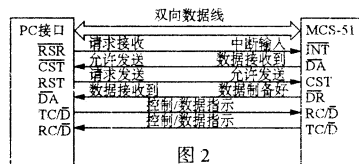


图2

图1是并行通信系统的结构原理图。主机与从机的并行接口可用8255来实现。图2是主机接口与从机的数据线、信号联络线连接图。在计算机并行通信过程中,干扰系统产生误码的主要原因是传输中的干扰,即信号通道的干扰。差错有两种可能,一是八位数据的某一位在传递中由于信道引入的干扰,导致误传产生的,二是由于联络信号受到干扰导致的数据丢失或多收产生的。由于在并行通信中是用TTL电平直接传递信息,在理论上规定传送距离最大可达15米,而在传输过程中,特别是在工业现场中,不可避免地存在着各种干扰,难免会产生误码,超过这个距离时,传输误码率会升高,使系统工作不可靠,因此,在分布式控制系统中直接应用这种TTL电平传送的标准是难以使系统正常工作的。为了有效地抑制信号传输过程中产生的干扰,提高信号传输距离,我们把接口信号的传输设计成差分平衡传输,其特点是:

(1)把输入的TTL电平信号转换成差分信号,然后去驱动输出端差分电路,差分电流以一定的电流

流驱动两根传输线,产生差分信号。

(2)用双绞线来双端传输信号,在信号传输过程中,外界干扰在传输线上引入的噪声对双绞线的两端来说是共模信号,差分放大器具有很强的抑制能力。因此,干扰信号在发送端不会影响驱动器输出,在接收端却能为接收器所抑制,保证了数据的正确传输。

(3)差分信号经过长距离传输后,在接收端幅度减小,但接收器的输入级是高灵敏的差分放大器,能接收上百伏毫伏的小信号加以放大,然后转换成 TTL 电平输出。因此,能进行长距离传送。

为了进行差分平衡传输,可采用 MC3478 四差分驱动器作为发送器,MC3486 四差分接收器作为接收器来实现。

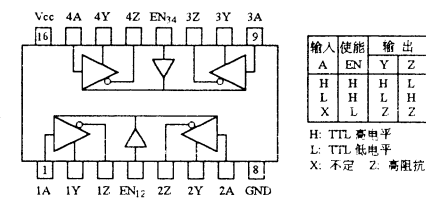


图 3

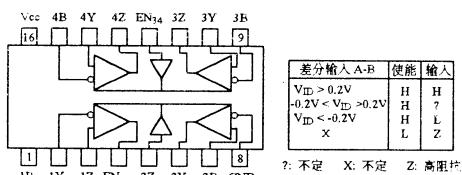


图 4

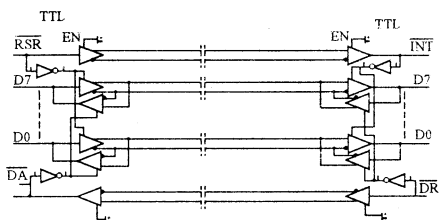


图 5

MC3478 是具有三态输出的四差分驱动器,其管脚和功能表如图 3。

MC3486 是具有三态输出的四差分接收器,其管脚和功能表如图 4。

图 5 是并行通信接口数据信号传输原理图。对于通信联络线,因传输是单方向的,故用一对收发器直接连接就可,控制使能端接高电平,电路构成较简单,如图 5 中的 RSR 信号转换所示,其他联络线的转换亦相同。对于数据线,由于要进行双向传递,须用两对收发器构成,信号线 RSR 和 DA 经反相后控制数据线的传送方向,当 RSR 线有效时,数据由主机向从机发送,DA 线有效时,数据由从机向主机发送。当系统不进行数据交换时,数据传输线处高阻态。

信号传输经上述转换后,极大地提高了系统并行通信的抗干扰性,当传速为 1MKB/S 时,传输距离可达 1KM,能适应一般工业过程控制的需要。

3 差错的检测与纠正

尽管通过上述可靠性设计后,系统的并行通信有了很强的抗干扰能力,出现误码的可能性很小,但差错还可能发生,要保证系统正确可靠,还必须对传送的数据信息进行检错与纠错。为了在并行通信系统中进行检错、纠错,必须通过传输协议来实现。

3.1 传输协议

传输协议规定如下:

a. 对主从机之间发送命令采用发送两次,接收方将收到的两数据进行比较,若相同,则发送成功,否则请求重发。

b. 对数据传送,采用数据包的形式,被发送的数据包含有错误检测信息,以使接收方能够确认是否在通信过程中产生了错误。检测信息是有传输协议计算出来的一个字节的检验和,并加在数据包的末尾。检验和是通过数据包中的所有字节的值加起来实现的,然后再取结果的模 256 的余数作为检验和,通信程序就把这个余数作为数据包的一附加字节送往接收计算机。接收计算机再进行相同的计算,并将结果与收到的检验和比较,若相同则正确,否则请求重发。

3.2 基本通信程序流程图

在由 PC 机及单片机组成的分布式系统中,通信只是系统功能的一部分,为了使数据发送方掌握主动权,通信程序发送采用查询方式,接收用中断驱动。根据以上传输协议,给出图 6 和图 7 所示的基本通信程序的发送、接收程序流程图如下:

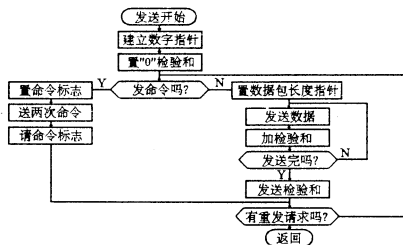


图 6

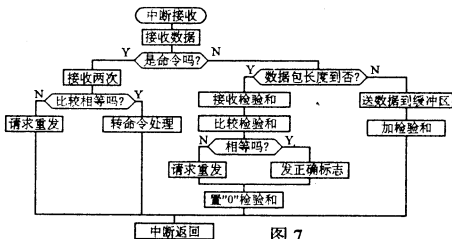


图 7

4 结束语

本文对分布式系统并行通信的可靠性设计与差错检测、纠正方法的研究,对解决分布式控制系统并行通信中的误码问题,提高系统的可靠性、稳定性具有实际意义,颇具推广价值。

参考文献

- 何立民, 单片机应用文集(2). 北京航空航天大学出版社, 1993 年
- 李慧军等. 计算机通讯与联网技术, 中国科学院希望电脑公司

基于交流采样的六千伏电网微机自动监测系统

王红岩

(河北煤炭建筑工程学院 邯郸 056038)

祁 堃

(邯郸矿务局 邯郸 056002)

摘 要 本文介绍了一种基于交流采样的六千伏电网微机自动监测系统,采用了傅里叶级数展开算法,实现对电压、电流、功率因数和电量的测量和计算,利用软件完成了需要较多硬件才能实现的多种功能,节省了大量的硬件投资,减少了日常维护量。该系统的采用对于提高供电系统运行效益,保证安全供电具有重要意义。

关键词 傅里叶级数 计算机监测 配电站

1 引言

随着微电子技术和微型计算机技术的发展,以微型计算机为核心的电网自动化系统在国内一些电网中相继投入使用。这对于电力系统的有效和安全运行起到了很好的保障作用。煤矿是我国的用电大户,实现煤矿变、配电站的自动监测,使供电系统合理、经济、安全地运行,对于缓解供电紧张局面,提高供电系统的自动化管理水平具有十分重要的意义。

邯郸矿务局王风煤矿有两个六千伏配电站:六千配电站(以下简称六千站)和大桥配电站(以下简称大桥站)。经上述两个配电站分别向全矿井上下、局钢铁厂、和水泥厂等地区供电。系统复杂,管理难度大,传统的电话查询、人工抄表的管理方法已越来越不适应。如何最大限度地提高系统运行效益,提高供电系统管理水平,保证安全供电就成为机电工作需要解决的重要课题之一。为此研制了六千伏电网微机自动监测系统。

2 系统的功能及技术指标

2.1 系统的功能

(1)实时采集六千站和大桥站的各路进线电压、电流、开关状态及各进、出线的电流。

(2)实时采集各开关柜的运行状态以及接地和跳闸事故信号。

(3)显示各开关柜的电压、电流、功率因数、有功和无功功率等参数。

(4)显示两配电站的模拟接线图和运行状态图。

(5)具有拷屏功能,可以把某一时刻配电站的运行状况图打印出来。

(6)具有打印日报、月报报表的功能,并可以进行数据的查询。

2.2 技术指标:

(1)模拟量:六千站两段母线电压,28个开关柜的电流,共计30路

大桥站两段母线电压,13个开关柜的电流,共计

15路

(2)数字量:六千站31路

大桥站16路

(3)采样精度:12bit

(4)画面切换时间: $<2s$

(5)巡检周期:5s

(6)传输距离: $>3km$

3 系统的硬件设计

3.1 系统结构

本系统由设置在配电站的下位机和调度室中的值班机与管理机组成了一个分布式的计算机监测系统,其组成框图如图1所示。

其中下位机和值班机为STD总线工业控制机。下位机负责采集配电站的各种数据及开关信号,经过计算和处理后把结果通过调制解调器传送到调度室中的值班机。值班机既是模拟屏的控制器又起接力和中转的作用,把从下位机接受来的数据送模拟屏显示,还把接受来的数据向PC机发送。PC机用来显示配电站的接线图、运行状态、运行参数以及查询数据,打印报表。由于设置了值班机并且具有数据存储功能,因而PC机可以以脱机方式工作,即只在需要时与值班机联络通讯,其他时间可处于关机状态或进行其他的工作,如文字处理、设备管理等。

3.2 硬件设计与选型

PC机的选型主要考虑价格、实用性、兼容性等因素,根据当时国内市场销售微机行情,结合现场采用脱机方式工作特点,选用了功能较强但价格较低的386兼容机。运行实践表明,这样的配置足以胜任现场的工作。

对值班机和两分站的下位机的要求是可靠性高,开发周期短,易扩展和易维护,经比较选用了航空航天部502所的STD总线MCS-51工业控制机。

4 信号采集方案

目前对交流信号的处理一般有三种方法:

(1)采用商品化的变送器。

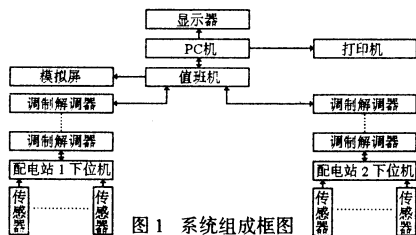


图 1 系统组成框图

(2) 采用均方根值法, 即在一个信号周期内采样 N 点, 由以下公式计算出电压、电流的幅值。

$$U = \sqrt{\frac{1}{T} \int_0^T u^2(t) dt} \approx \sqrt{\frac{1}{N} \sum_{k=1}^N u^2(k)}$$

$$I = \sqrt{\frac{1}{T} \int_0^T i^2(t) dt} \approx \sqrt{\frac{1}{N} \sum_{k=1}^N i^2(k)}$$

(3) 傅里叶级数展开法, 即把周期信号展开成傅里叶级数, 由采样点和傅里叶级数的系数可以计算出周期信号的幅值和相位。经过比较, 方法 1 的精度高, 安装和使用方便, 但成本高。方法 2 硬件投资少, 软件计算量较少, 但不容易求出 $\cos\varphi$ 。方法 3 硬件投资少, 软件计算量稍大, 但精度较高, 容易计算出 $\cos\varphi$ 值, 且具有较强的滤波功能。下面介绍方法 3 的原理。

由信号分析的理论可知, 周期信号 $f(t)$ 的傅里叶级数的展开式为:

$$\begin{aligned} f(t) &= \frac{A_0}{2} + \sum_{k=1}^{\infty} A_{km} \sin(k\omega t + \varphi_k) \\ &= \frac{A_0}{2} + \sum_{k=1}^{\infty} A_{km} \cos\varphi_k \sin k\omega t + \sum_{k=1}^{\infty} A_{km} \sin\varphi_k \cos k\omega t \\ &= \frac{A_0}{2} + \sum_{k=1}^{\infty} B_{km} \sin k\omega t + \sum_{k=1}^{\infty} C_{km} \cos k\omega t \end{aligned}$$

其中: A_{km} 为第 k 次谐波的振幅。

φ_k 为第 k 次谐波的初相位。

当 $k=1$ 时, $A_{1m}\cos\varphi_1$ 和 $A_{1m}\sin\varphi_1$ 分别是基波的实部幅值和虚部幅值, 无论 $f(t)$ 为正弦波、失真的或者是迭加有干扰的正弦波, 只要能计算出 $A_{1m}\cos\varphi_1$ 和 $A_{1m}\sin\varphi_1$, 即得到了理想正弦波的实部和虚部幅值。因而求正弦波的实部和虚部幅值就变为求傅氏级数的系数

$B_{1m} = A_{1m} \cos\varphi_1$, $C_{1m} = A_{1m} \sin\varphi_1$, 而 $B_{km} = \frac{2}{T} \int_0^T f(t) \sin(k\omega t) dt$ 经离散化后, 得:

$$B_{km} = \frac{2}{nT_0} \sum_{j=1}^{n-1} f(jT_0) \sin(k\omega_j T_0) T_0$$

$$= \frac{2}{n} \sum_{j=1}^{n-1} f(jT_0) \sin\left(\frac{2\pi k j}{n}\right)$$

$$C_{km} = \frac{2}{n} \sum_{j=1}^{n-1} f(jT_0) \cos\left(\frac{2\pi k j}{n}\right)$$

$$\text{令 } k=1 \text{ 则 } B_{1m} = \frac{2}{n} \sum_{j=1}^{n-1} f(jT_0) \sin\left(j \frac{2\pi}{n}\right)$$

$$C_{1m} = \frac{2}{n} \sum_{j=1}^{n-1} f(jT_0) \cos\left(j \frac{2\pi}{n}\right)$$

其中 n 为一个信号周期内的采样点数, T_0 为采样周期。由奈奎斯特采样定理可知, 只要采样频率大于最高信号频率的两倍, 即可无失真地恢复出原信号, 采样频率越高, 精度越高, 滤波性能越好, 但相应的计算量也越大。经计算机模拟后, 决定每个周期采样十二个点, 这样可在精度和计算量之间取得一个折衷。在上面的公式中令 $n=12$, 即可得出计算正弦电压和电流的公式如下:

$$U_{Rm} = \frac{1}{6} [(U_3 - U_9) + \frac{1}{2} (U_1 + U_5 - U_7 - U_{11}) + \frac{\sqrt{3}}{2} (U_2 + U_4 - U_8 - U_{10})]$$

$$U_{Xm} = \frac{1}{6} [(U_0 - U_6) + \frac{\sqrt{3}}{2} (U_1 - U_5 - U_7 + U_{11}) + \frac{1}{2} (U_2 - U_4 - U_8 + U_{10})]$$

$$I_{Rm} = \frac{1}{6} [(I_3 - I_9) + \frac{1}{2} (I_1 + I_5 - I_7 - I_{11}) + \frac{\sqrt{3}}{2} (I_2 + I_4 - I_8 - I_{10})]$$

$$I_{Xm} = \frac{1}{6} [(I_0 - I_6) + \frac{\sqrt{3}}{2} (I_1 - I_5 - I_7 + I_{11}) + \frac{1}{2} (I_2 - I_4 - I_8 + I_{10})]$$

$$U_m = \sqrt{U_{Rm}^2 + U_{Xm}^2}$$

$$I_m = \sqrt{I_{Rm}^2 + I_{Xm}^2}$$

$$\sin\alpha = \frac{U_{Xm} I_{Rm} - U_{Rm} I_{Xm}}{U_m I_m}$$

$$\cos\varphi = \cos(\alpha - 30^\circ) = \frac{\sqrt{3}}{2} \cos\alpha + \frac{1}{2} \sin\alpha$$

$$\sin\varphi = \sqrt{1 - \cos^2\varphi}$$

$$P = \sqrt{3} U_l I_l \cos\varphi$$

$$Q = \sqrt{3} U_l I_l \sin\varphi$$

其中: U_{Rm} : 线电压的实部幅值。

U_{Xm} : 线电压的虚部幅值。

I_{Rm} : 线电流的实部幅值。

I_{Xm} : 线电流的虚部幅值。

U_m : 线电压幅值。

I_m : 线电流幅值。

U_l : 线电压有效值。

I_l : 线电流有效值。

α : 线电压与线电流的夹角。

φ : 相电压与相电流的夹角。

P : 三相电路有功功率。

Q : 三相电路无功功率。

由以上公式即可计算出电压、电流、功率因数、有功功率、无功功率等参数。

5 系统的软件设计

整个系统的软件分为三部分: 下位机软件, 值班机软件和管理机软件。

5.1 下位机软件

下位机软件完成的功能如下:

- (1)完成模拟量和开关量的信号采集。
- (2)根据傅里叶级数的展开式计算电压电流的幅值及功率因数,计算有功功率、无功功率及电量等参数。
- (3)根据信号屏的开关信号搜索跳闸的信号。
- (4)向值班机发送数据。

由于下位机采用的是STD总线MCS-51工业控制机,本部分软件采用MCS-51汇编语言编写。其中,数据采集和串行通讯部分分别由定时中断和串行中断完成。

5.2 值班机软件

该部分软件完成的功能如下:

- (1)接受分站发送来的数据。
- (2)乘开关柜互感器变比和校正系数。
- (3)判断有无越限及跳闸和接地故障,如有则声光报警。
- (4)存储整点的电压、电流、功率因数,累计有功、无功电量等参数。

(5)向管理机发送数据。

本部分软件采用MCS-51汇编语言编写。

5.3 管理机软件

该部分软件完成的功能如下:

- (1)实现管理机与值班机的数据通讯。
- (2)显示各分站的模拟接线图及开关运行状态和运行参数。
- (3)拷屏功能,即把某一时刻某站的运行状态及参数以图形方式用打印机记录下来。
- (4)校正值班机时间。
- (5)向值班机发送电压和电流的校正系数。
- (6)定时接收前一天的数据并存盘。
- (7)查询各开关柜的电压、电流、有功功率和无功功率等参数。

(8)打印用电情况日报表,负荷日报表,月用电量统计表,代表日统计表。

上述的1-6项功能用C语言编程实现,7,8项功能用Foxbase+语言编程实现。两种语言通过磁盘文件进行数据交换。采用C语言和Foxbase+语言混合编程,充分发挥了C语言图形功能强和Foxbase+语言数据处理能力强的优势。

6 结论

本系统采用了傅里叶级数展开算法,实现对电压、电流、功率因数和电量的测量和计算,利用软件完成了需要较多硬件才能实现的多种功能,既大量地节省了系统投资费用,同时也减少了日常维护量。

系统投入运行后,提高了配电系统的科学管理水平,给王风矿的电力调度管理系统带来极大的方便和可观的经济效益。本系统可以准确地实时监测尖峰负荷,可以根据电力调度指挥系统的决策,抑制尖峰负荷,保障电网的安全运行。可以进行峰谷负荷调度,根据目前供电部门对企业核收峰、谷、平段电价各不相同的管理办法,使用本系统后,调度人员可以方便地利用井下各泵房水仓的储水能力,尽可能地将占矿山用电量近一半的排水用电调整到谷段或平段,降低综合电耗,为企业节约开支。直接监视井下泵房各泵运行情况,及时发现各泵运行中出现的问题从而编排出井下各泵房排水系统优化表,多开高效泵,及时检修效率降低的水泵,节约了大量电能。根据现行用电管理办法,使用微机自动监测系统后,王风矿年可节约电费百万元以上。

系统投入运行一年多以来,稳定可靠,方便实用,并于一九九五年十二月通过了河北省煤炭厅组织的技术鉴定。

参考文献

- 1 魏庆福等,STD总线工业控制机的设计与应用,科学出版社,1992。
- 2 陈建铎,Intel单片机应用技术,陕西电子编辑部。

(上接第109页)

0496(01F0) 53 4D 53 44 4F 53 20 20 20 53 59 53 00 00 55
AA

F1 Help F2 Index F3 Exit F6 Sector F7 Edit F8 Name

本窗口的中间部分是磁盘内容的16制的表示,右边是相应的ASCII码。Absolute Sector 0000000表示绝对扇区即引导扇区。

按F6功能键进入菜单

Change Sector

(.)BOOT sector 引导扇区

(0)First FAT Sector 第一个FAT扇区

(0)First ROOT Sector 第一个根目录扇区

(0)First DATA Sector 第一个数据扇区

(0)Change ciuster #

(0)Change Sector #

Continue Exit

分别选择欲显示的有效扇区,查找被病毒覆盖或修改的扇区及相应的地址并用F7功能键进行编辑修改。

3. 系统丢失

由于系统丢失而引起的引导型故障,只须按屏幕提示进入COMS SETUP菜单,重新设置正确的硬盘参数即可。

参考文献

- 1 386/486微型计算机系统原理与维修 朱乃传 主编 人民邮电出版社
- 2 微机常见故障的自我排除及其使用技巧 张仰森 编著 电子工业出版社

容积调速系统的模糊控制研究

左健民

(南京机械高等专科学校 210013)

王积伟

(东南大学 210096)

摘 要 本文以泵控液压马达容积调速系统的实时控制为例,介绍了以 MCS-51 单片机为主控单元的模糊控制系统的设计。实验研究表明,模糊控制算法较常规的 PID 控制算法控制精度高,超调量小,动态品质好。

关键词 容积调速 控制算法 模糊控制

1 前言

液压容积调速系统因其效率高、体积小,在大功率液压系统中被广泛采用,由于它是非线性、多变量、参数时变的系统,其数学模型比较复杂,在某些情况下,甚至无法找到其输入/输出之间的明确的函数关系,因此给设计控制器带来很大困难。用常规的 PID 控制器对其进行控制,但由于参数时变, PID 调节参数难以整定,所以难以达到预期效果。为此,我们研究了以模糊控制理论为基础的实时控制系统。

所谓模糊控制,就是把操作者的经验总结成若干条法则,经过必要的数学处理,存放于计算机中去,并仿照人脑的模糊推理过程确定推理法则,计算机根据现场实时测试的数据经处理输入的模糊信息,按照控制规则作出模糊决策,发出控制指令。这就实现了模糊控制,模糊控制具有鲁棒性好的特点,参数变化的影响也小。

2 控制系统的硬件设计

控制系统的硬件框图如图 1 所示,控制系统由 MCS-51 系列单片机 8031 扩展而成,片外扩展一片 EPROM2732,用以固化系统控制软件,一片 RAM6116 用于存储有关数据,经传感器采集来的数据由 ADC0809 实现 A/D 转换后输入单片机,键盘和显示器由 74LS164 通过串行口和 P_{1.5}、P_{1.6} 与单片机相联,以实现输入和监控;输出接口由 8031 的 P_{1.1}~P_{1.3} 输出,经过光电隔离后输入功放电路以驱动步进电动机,由步进电动机通过机械传动装置驱动伺服变量泵的变量连杆。从而实现以调节液压泵的输出而稳定液压马达的输出参数的闭环控制系统。

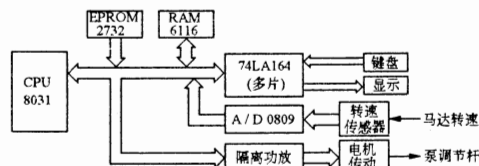


图 1 系统硬件框图

机,由步进电动机通过机械传动装置驱动伺服变量泵的变量连杆。从而实现以调节液压泵的输出而稳定液压马达的输出参数的闭环控制系统。

3 模糊控制器的设计

用模糊控制器代替人进行控制时,需要把不断地测量到的过程输出的精确量转化成模糊量,经过模糊决策后,再将决策的模糊量转化成精确量去实现控制动作,因此设计模糊控制器必须解决三个问题:(1)精确量的模糊化;(2)模糊控制规则的构成;(3)模糊判决。在液压系统的模糊控制中,与模糊化有关的精确量一般是偏差 E,偏差变化率 dE/dt (E) 和控制量 U,液压系统的模糊控制框图如图 2 所示:

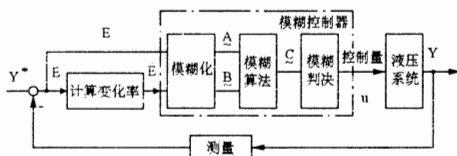


图 2 模糊控制系统框图

(1) 精确量的模糊化

在把精确量转化成模糊量时,按照人们的常规思维方式,把精确量分为几档,每一档对应一个模糊集,在工程应用中,可以把精确量 E、E' 和 U 设定为 [-6, +6] 之间变化的连续量,如精确量不在 [-6, +6] 范围内,而在已知范围 [α, β] 内则可通过下式变换:

$$Y = \frac{12}{\beta - \alpha} \left[X - \frac{\alpha + \beta}{2} \right] \quad (1)$$

把在 [α, β] 间变化的变量 X 转化为 [-6, +6] 之间的变量 Y,这样就可以只讨论论域在 [-6, +6] 区间内的模糊子集分布。再将这一论域内的连续精确量离散化分为若干档为:

$\tilde{E} = \{ \text{正大(PL)}, \text{正中(PM)}, \text{正小(PS)}, \text{正零(P0)}, \text{负零(N0)}, \text{负小(NS)}, \text{负中(NM)}, \text{负大(NL)} \};$

\sim

$E = \{ \text{PL}, \text{PM}, \text{PS}, 0, \text{NS}, \text{NM}, \text{NL} \}$

$$\tilde{U} = \{PL, PM, PS, 0, NS, NM, NL\}$$

并将 E 、 \dot{E} 和 U 的论域分为下列量化等级:

$$E = \{+6, +5, +4, +3, +2, +1, 0, -1, -2, -3, -4, -5, -6\}$$

$$\dot{E} = \{+6, +5, +4, +3, +2, +1, 0, -1, -2, -3, -4, -5, -6\}$$

$$U = \{+6, +5, +4, +3, +2, +1, 0, -1, -2, -3, -4, -5, -6\}$$

由此根据实际系统和操作者的经验,就可以写出各档的隶属函数,例如对于 NL(负大),在论域 E 中, -6 为最负的值,因而可以认为 -6 对 NL 的隶属度为 1,若认为 $-5, -4, -3, -2$, 对于 NL 的隶属度分别为 0.8, 0.7, 0.4 和 0.1,则可写出论域 E 中对于 NL 的模糊集。

$$\tilde{A} = (1.0, 0.8, 0.7, 0.4, 0.1, 0, 0, 0, 0, 0, 0, 0, 0)$$

同理可以写出论域 E 中对于 NM, NS, N0, P0, PS, PM, PL 的模糊集,这八个模糊子集的隶属函数如表 1 所示。

对于误差变化量 \dot{E} 和控制量 U ,一般把它们分成七档,即 NL, NM, NS, 0, PS, PL, PM, 其模糊子集所对应的隶属函数如表 2 和表 3 所示。

隶属函数的值在现场调试时还可进行调整,以期达到最佳控制效果。

表 1 偏差 E 的模糊集的隶属函数表

$\mu(E)/E$	-6	-5	-4	-3	-2	-1	0	1	2	3	4	5	6
A1 NL	1.0	0.8	0.7	0.4	0.1	0	0	0	0	0	0	0	0
A2 NM	0.2	0.7	1.0	0.7	0.3	0	0	0	0	0	0	0	0
A3 NS	0	0.1	0.3	0.7	1.0	0.7	0.2	0	0	0	0	0	0
A4 N0	0	0	0	0	0.1	0.6	1.0	0	0	0	0	0	0
A5 P0	0	0	0	0	0	0	1.0	0.6	0.1	0	0	0	0
A6 PS	0	0	0	0	0	0	0.2	0.7	1.0	0.3	0.1	0	0
A7 PM	0	0	0	0	0	0	0	0.2	0.7	1.0	0.7	0.3	0
A8 PL	0	0	0	0	0	0	0	0	0.1	0.4	0.7	0.8	1.0

(2) 模糊控制算法的设计

本系统采用的模糊控制器(如图 2 所示)输入是二维的,输出是一维的,根据模糊数学的似然推理原则,当输入为 \tilde{E}_i 和 \tilde{E}_j , 输出为 \tilde{U}_{ij} 时,有模糊关系;

$$\tilde{R} = \bigcup_{ij} (\tilde{E}_i \times \tilde{E}_j) \times \tilde{U}_{ij}, i=1, 2, \dots, n; j=1, 2, \dots, m \quad (2)$$

式中“ \times ”的含义定义为

$$\mu_{\tilde{R}}(e, \dot{e}, u) = \bigvee_{i=1}^n \bigvee_{j=1}^m \mu_{\tilde{E}_i}(e) \wedge \mu_{\tilde{E}_j}(\dot{e}) \wedge \mu_{\tilde{U}_{ij}}(u) \quad (3)$$

式中 $e \in E, \dot{e} \in \dot{E}, u \in U$

表 2 偏差变化率 dE/dt 的模糊集的隶属函数表

$\mu(\dot{E})/\dot{E}$	-6	-5	-4	-3	-2	-1	0	1	2	3	4	5	6
B1 NL	1.0	0.8	0.4	0.1	0	0	0	0	0	0	0	0	0
B2 NM	0.2	0.7	1.0	0.7	0.2	0	0	0	0	0	0	0	0
B3 NS	0	0.2	0.7	1.0	0.9	0	0	0	0	0	0	0	0
B4 0	0	0	0	0	0	0.5	1.0	0.5	0	0	0	0	0
B5 PS	0	0	0	0	0	0	0	0.9	1.0	0.7	0.2	0	0
B6 PM	0	0	0	0	0	0	0	0	0.2	0.7	1.0	0.7	0.2
B7 PL	0	0	0	0	0	0	0	0	0	0.1	0.4	0.8	1.0

表 3 控制量 U 的模糊集的隶属函数表

$\mu(u)/u$	-6	-5	-4	-3	-2	-1	0	1	2	3	4	5	6
c1 NL	1.0	0.8	0.4	0.1	0	0	0	0	0	0	0	0	0
c2 NM	0.2	0.7	1.0	0.7	0.2	0	0	0	0	0	0	0	0
c3 NS	0	0.1	0.4	0.8	1.0	0.4	0	0	0	0	0	0	0
c4 0	0	0	0	0	0	0.5	1.0	0.5	0	0	0	0	0
c5 PS	0	0	0	0	0	0	0.4	0.8	1.0	0.4	0	0	0
c6 PM	0	0	0	0	0	0	0	0	0.2	0.7	1.0	0.7	0.2
c7 PL	0	0	0	0	0	0	0	0	0	0	0.1	0.8	1.0

若偏差和偏差变化率分别为模糊集 \tilde{E} 和 \tilde{E} , 则由合成规则可得控制量 \tilde{U} 。

$$\tilde{U} = (\tilde{E} \times \tilde{E}) \cdot \tilde{R} \quad (4)$$

式中“ \cdot ”为合成算子,这里取 $\max - \min$

当 \tilde{E}_i 和 \tilde{E}_j 取不同值时,根据操作者的经验,相应模糊控制变量的输出值 U 的大小如表 4 所示,根据表中所列的关系,通过式(2)或式(4),即可求出实际的控制值。

(3) 模糊控制器的模糊判决

根据泵控系统要求实时性好的特点,选择从属度最大原则的判决方法,即取模糊子集中从属程度最大的那个元素 U^* 作为执行量。

表 4 典型控制规则(模糊状态)表

$\tilde{A} \backslash \tilde{B}$	NL	NM	NS	N0	P0	PS	PM	PL
NL	PL*	PL*	PM	PM	Ps	P0	0	0
NM	PL	PL	PM	PM	PM	PS	0	0
NS	PL	PL	PM	PS	PS	0	NM	NM
0	PL	PL	PM	0	0	NM	NB	NB
PS	PM	PM	0	NS	NS	NM	NB	NB
PM	PM	0	NS	NM	NM	NM	NB	NB
PL	0	0	NS	NM	NM	NM	NB*	NB*

* 表示一般不大容易出现的情况

4 模糊控制软件设计

在本控制系统中,模糊控制器的输入为

$$E = Y - Y^* \quad dE/dt = (E_i - E_{i-1})/T$$

其中, Y 为系统反馈值, Y^* 为系统设定值, E_i 为 KT 时刻的偏差, E_{i-1} 为 $(K-1)T$ 时刻的偏差, T 为采样周期。

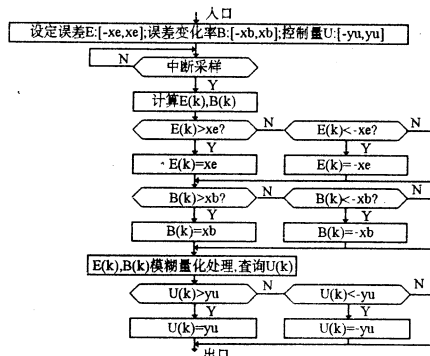


图3 模糊控制算法查表算法图

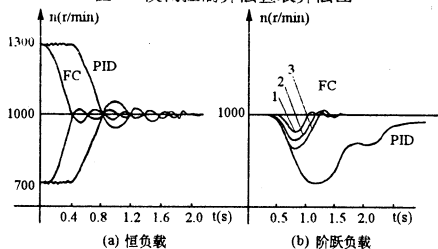


图4 动态响应曲线

在对泵控容积调速系统液压马达的转速的控制中,对于一定的工作速度下 E 的变化范围为 $[-500, 500]$ (r.p.m.), E 的变化范围为 $[-1000, 1000]$ (rpm/s), U 的控制量程范围为 $[0, 255]$, 通过式(1)的线性变换,将其转化为论域 $[-6, +6]$ 范围内不同量化等级的量,根据模糊变量 E 、 E 和 U 的隶属函数表1、表2和表3,结合控制规则表4,运用式(2)和式(4),取隶属度最大原则,可得该控制系统的模糊控制器控制量 U 取值表(表5)。由于此表所占内存小于256个字节,因而可方便地存入计算机中,在实时控制中,计算机根据系统采样所得比较后的偏差并计算出偏差变化率,经模糊化处理后可直接与表中行、列比较,搜索出所需要的控制变量 U 并输出。二维模糊控制器查表算法的程序

(上接107页)

```
MOVX A,@DPTR;
PUSH A;
INC DPTR
MOVX A,@DPTR;查表找出跑飞段的起始地址高8位
并进栈
```

```
PUSH A;
RETI;将跑飞段的起始地址置入PC使程序从该处恢复
```

4 结语

本文介绍的程序运行失常自恢复控制的方法同时

框图如图3所示。

表5 模糊控制表

u \ E	E	偏差变化率 dE/dT												
		-6	-5	-4	-3	-2	-1	0	1	2	3	4	5	6
-6	-6	6	6	6	6	6	6	6	3	3	1	0	0	0
-5	-5	6	6	6	6	6	6	6	3	3	1	0	0	0
-4	-4	6	6	6	6	5	5	5	3	3	1	0	0	0
-3	-3	6	5	5	5	5	5	5	2	1	0	-1	-1	-1
-2	-2	3	3	3	3	3	3	3	1	0	0	-1	-1	-1
-1	-1	3	3	3	3	3	3	1	0	0	0	-2	-1	-1
0	0	3	3	3	3	1	1	0	-1	-1	-1	-3	-3	-3
+1	+1	1	1	1	1	0	0	-1	-3	-3	-3	-3	-3	-3
+2	+2	1	1	1	1	0	-2	-3	-3	-3	-3	-3	-3	-3
+3	+3	0	0	0	0	-2	-2	-5	-5	-5	-5	-5	-5	-5
+4	+4	0	0	0	-1	-3	-3	-6	-6	-6	-6	-6	-6	-6
+5	+5	0	0	0	-1	-3	-3	-5	-5	-5	-5	-5	-5	-5
+6	+6	0	0	0	-1	-3	-3	-6	-6	-6	-6	-5	-5	-5

5 实验结果

根据以上设计思想设计了模糊控制器,经现场调试和测试,恒负载和阶跃负载条件下的动态响应曲线如图4所示,从响应曲线可以看出,采用模糊控制的控制效果要优于传统的PID调节器,其调节时间几乎不受负载阶跃幅值的影响,且控制精度也较高。

6 结论

(1)模糊控制不需要建立系统精确的数学模型,这对于参数严重时变的液压系统来说是一种非常有效的控制策略,且控制算法简单,在控制和调节中能取得较为满意的控制效果。

(2)本控制系统的硬件、软件设计基本满足了控制和调节的需要,达到了在扰动条件下,稳定系统的输出,实现实时控制的目的。

参考文献

- 1 李士勇等编著.模糊控制和智能控制理论与应用.哈尔滨工业大学出版社.1990.12
- 2 蒋静坪编著.计算机实时控制系统.浙江大学出版社.1992.4

从时间和空间上对微机系统的程序运行进行监控,大大提高了对程序运行失常识别的准确性、及时性和自恢复控制的实时性、针对性,实现电路简单可靠,是提高工业测控领域中软件抗干扰能力的有效措施。

参考文献

- 1 何立民.单片机应用系统设计.北京:北京航空航天大学出版社,1990
- 2 孙涵芳等.MCS-51、96系列单片机原理及应用.北京:北京航空航天大学出版社,1988

在华光排版系统上实现双页并排输出

马龙军

(云南财贸学院计统系 昆明 650221)

摘 要 本文介绍了如何在华光排版系统上实现双页并排输出的一种方法,有效地解决了华光排版系统不能并排输出双页的问题。

关键词 华光 排版 注解

1 问题的提出

华光排版系统(即 BD 排版语言)是北大方正公司推出的排版专业性软件,是我国印刷界推广使用的排版印刷语言,也是普及率最高的、性能最好的排版印刷语言。它具有丰富的字体、字号及各种修饰,有百余种底纹图案供选择使用,由于它是用语言进行排版描述,使得比其它排版系统具有更强的功能。用它可实现任何复杂版面的排版,诸如换字体、各种复杂的表格设计、底纹图案的设置、各种数学物理公式、化学方程式及符号等,正因为如此,全国各大报纸及主要杂志均采用华光排版语言进行排版。但华光排版系统的打印输出是按页输出的,没有提供双页并排输出的功能,这在需用八开版面并排印刷十六开两页是很不方便的。我们知道,一般的文字材料、稿件、文件及教材等都是十六开版面,而我们办公最常用的印刷机械如机械印刷机、电动印刷机、一体印刷机及八开小胶印机等,版面都是八开的,在八开版面上并排放置十六开的两页,无论从印刷的角度来讲,还是从装订的角度来讲,都是很常用而且很方便的。笔者在使用华光排版语言的过程中,通过对其排版语言的分析,找到了解决这一问题的有效方法。

2 问题的解决

华光排版语言的基本单位是注解,华光排版语言虽没有提供并排打印输出功能,但却提供了分栏注解,这恰是我们所要利用的。首先,我们将版心的宽度定为我们所要并排打印两页的宽度,即此时“一页”的宽度应等于所印材料两页的宽度加中缝的宽度。例如,我们要并排输出两页、每页宽 32(五号字),中缝宽为 8,那么,华光系统排版的一页应为 $32 \times 2 + 8 = 72$ 字宽。而其高度就是所要印的页的高度,但这个高度不包括页码行。其次,在宽度和高度定好后,利用分栏注解,将排版的“一页”分为两个等栏,并且两栏在结束时不拉平。这可在文件开始加【FL(2K<中缝宽度>)】注解,在文件结束符 Ω 前加【FL)X】注解。经过上述处理,即可实现双栏并排输出。这就等于所印资料的双页并排输出。

剩下的问题就是如何加入页码,一般来讲,页码都是放在每页的下边,为了实现这一目的,我们可采取如下方法来实现,首先,在排版格式文件(即与排版文件名同名,扩展名为 PRO 的文件)中除了说明如前面要求的版心宽、高之外,还要加上眉说注解。将书眉说明在每页的下面,即用注解【MSX<字号、字体>】(也可在排版时选定“书眉说明”一项完成书眉说明)。其次,在分栏开弧注解之后加入单、双眉注解,注解内容就是页码,这样,就把页码当做页底部放的书眉而排在两栏的下边,有效地解决了页码的问题。

由于页码是变化的,为了在应用中方便,我们可采用自定义注解来定义单、双眉注解的内容,其定义格式为:

【ZD<自定义注解名>】(【MM(】.<两个页码间的距离>.(【MM)】【ZD)】

其中 \bigcirc 两边的点是使页码两边加点,如若采用其它的标记,即可用其它符号。

上述工作做好后,即可先排版一次,然后显示所排文件。记下排版的每页两栏中的任一位置(一般找易查找的位置),然后再回到编辑状态,在上述位置上加入注解【=<自定义注解名>】(【页码 1 Ω 页码 2【=】),这里的自定义注解名应与前面的自定义注解名同名。页码 1 和页码 2 是作为自定义注解的两个实参(对应于 \bigcirc),它应是排版文件的一页两栏的“栏页码”,即分别是所印材料两页的页码。由于一页两栏的篇幅很大。加之可在其任意位置插入自定义注解,所以上述功能的实现是及其容易的。

3 实例

我们在实际应用中,对于 EPSON-1600K 或 3070 等打印机,建立一个格式文件。3070. PRO(或 1600. PRO)其内容如下:

【BX5SS,35,80,*2】【MSX3H】

每次要排文件时,即可将上述格式文件拷贝成所要排版文件的格式文件。这样,不必每排一个文件就要说明一次。在正文中,如前所述的加入

【FL(2K10)正文内容【FL)X】

注解及

子目录加密的策略与实现方法

朱家义 王玉芬

(河南职业技术学院计算机室 新乡 453003)

摘 要 本文通过分析 DOS 对子目录的管理方法,提出了子目录的加密策略。并用例子实现了子目录加密的具体步骤。

关键词 子目录 安全 加密

1 引言

目前,一般的微机都配置大容量的硬盘,对于多人共用一机来说,人们常常希望对自己的文件进行有效的保护,防止非法访问,这种保护有许多方法,本文是在笔者实践基础上,系统地阐述了子目录加密的策略和方法。

2 DOS 的目录管理方法及特点

DOS 分配 32 个字节来登记子目录的信息,格式是:

00 08 0B 0C 16 18 1A 1C 1F

--	--	--	--	--	--	--	--	--

其中 00—07 是子目录名

08—0A 是扩展名

0B 是属性

0C—15 是保留区

16—17 是时间

18—19 是日期

1A—1B 是起始簇,表示分配给该子目录的起始簇号

1C—1F 是字节长度

其主要特点:

(1)子目录的属性是 10H

(2)子目录的文件长度为 0

(3)子目录的第一个簇号由起始簇号指示,而分配的其它簇号只能跟踪文件分配表 FAT 的簇链号。

(4)由起始簇域指示的扇区位置上存放相应的子目录登记表。每一个子目录开始的两个目录项特定为 (2EH) 和 .. (2EH, 2EH)。当且仅当存在这两个登记项,才能用子目录删除命令删除该子目录。

3 子目录加密的策略和方法

根据 DOS 对目录的管理方法,子目录加密可采取以下策略(因篇幅所限,笔者仅以加密 C:\FOX 为例,只对少数方法的操作进行示范,其它的方法读者可自己实现)。

3.1 修改子目录名实现加密

子目录名是对子目录访问的重要标志,在子目录名中加入一些不易识别的字符,从而达到加密的目的。

3.1.1 建立子目录时加入扩充 ASCII 码或半个汉字。扩充 ASCII 码由 ALT 和键盘右端的数字键所组成,数字是 000~255 的三位数。半个汉字的输入方法是:先键入一个汉字,然后用 ←BACKSPACE 键删去

收稿日期:1996 年 6 月 10 日

【ZDY(【MM(【.○.【KG18】.○.【MM)】(ZD)】

之后,先排一次版,记住每页两栏中的任意位置,再回到编辑状态下,在各位置加入如下注解

【=Y(【页码 1Ω 页码 2【=】

如第一页的两栏中应加入

【=Y(【1Ω2【=】

第二页的两栏中应加入

【=Y(【3Ω4【=】

等等,依此类推全都加完后再排版,即可完成双页并排输出。

关于版心的宽度,要随打印机而定,如 2024,1724 等打印机的字号比标准的略大,所以,格式文件一般设定为:

【BX5SS,32,72,* 2】

【MSX3H】

其分栏注解中两栏中间距离一般设定为 8。

这里需指出的一点是,上述版心及栏间距的设定完全可根据自己的实际需要而定。

4 结束语

华光系统是国内目前最普及、最先进的电子排版系统,开发华光系统的功能,充分地发挥其作用,无疑是非常有实用价值的。上面所述方法,有效地解决了华光系统不能并列打印双页的问题(这里不包括激光印字机的双页输出功能),这对于广大拥有华光排版系统及使用蜡纸的各种印刷设备进行印刷的用户来说,是极其方便的。

参考文献

1 BD 排版语言。

汉字的右半部分。

3.1.2 把子目录名中的一些字符改为空格(ASCII 码为 20)

用 DEBUG 具体做法是:

(1)调用 DEBUG

(2)键入 L02A1 2

(3)键入 D100,找到 FOX 子目录(如图 1)

C:\>debug		
-10 2 al 2		
-d100		
0A47:0100	41 55 54 4F 45 58 45 43-42 41 54 20 00 00 00 00	AUTOEXEC-
0A47:0110	00 00 00 00 00 00 D1 21-A6 20 6C 04 37 00 00 00	BAT...
0A47:0120	54 50 20 20 20 20 20 20-20 20 20 10 00 00 00 00	...1.1.7..
0A47:0130	00 00 00 00 00 00 0B 23-A6 20 DA 0C 00 00 00 00	TP...
0A47:0140	53 45 54 55 50 20 20 20-42 41 54 20 00 00 00 00	...#...
0A47:0150	00 00 00 00 00 00 53 58-46 1A 71 01 76 03 00 00	SETUP BAT..
0A47:0160	46 4F 58 20 20 20 20 20-20 20 20 10 00 00 00 00	..SXF.q.v
0A47:0170	00 00 00 00 00 00 6C 24-A6 20 0A 13 00 00 00 00	FOX....
		...1\$...

图 1

(4)用修改功能 E,把子目录 FOX 的第 2、3 字节改为 20(如图 2)

-e161		
0A47:0161 4F.20 58.20		
-d100		
0A47:0100	41 55 54 4F 45 58 45 43-42 41 54 20 00 00 00 00	AUTOEXEC-
0A47:0110	00 00 00 00 00 00 D1 21-A6 20 6C 04 37 00 00 00	BAT..
0A47:0120	54 50 20 20 20 20 20 20-20 20 20 10 00 00 00 00	...1.1.7..
0A47:0130	00 00 00 00 00 00 0B 23-A6 20 DA 0C 00 00 00 00	TP...
0A47:0140	53 45 54 55 50 20 20 20-42 41 54 20 00 00 00 00	...#...
0A47:0150	00 00 00 00 00 00 53 58-46 1A 71 01 76 03 00 00	SETUP BAT
0A47:0160	46 20 20 20 20 20 20 20-20 20 20 10 00 00 00 00	...SXF.q.v
0A47:0170	00 00 00 00 00 00 6C 24-A6 20 0A 13 00 00 00 00	F...
		...1\$..

图 2

(5)键入 W 存盘

(6)键入 Q 退出

用 PCTOOLS 修改的方法:具体实现是用 PC-TOOLS 进入主菜单后,按 F3,选择“DISK SERVICES”中的“VIEW/EDIT”功能项,击 F2 键,再选择 R,搜索到 FOX 子目录(如图 3)。

Advanced PC Tools 5.0

----- Disk View/Edit Service -----

Path=C:

Absolute sector 0000161, System ROOT		
Displacement	Hex codes	ASCII value
0256(0100)	41 55 54 4F 45 58 45 43 42 41 54 20 00 00 00 00	AUTOEXEC-
0272(0110)	00 00 00 00 00 00 D1 21 A6 20 6C 04 37 00 00 00	BAT
0288(0120)	54 50 20 20 20 20 20 20 20 20 10 00 00 00 00	! 1 7
0304(0130)	00 00 00 00 00 00 0B 23 A6 20 DA 0C 00 00 00 00	TP >
		# +

0320(0140)	53 45 54 55 50 20 20 20 42 41 54 20 00 00 00 00	SETUP BAT
0336(0150)	00 00 00 00 00 00 53 58 46 1A 71 01 76 03 00 00	SXF q v
0352(0160)	46 4F 58 20 20 20 20 20 20 20 10 00 00 00 00	FOX >
0368(0170)	00 00 00 00 00 00 6C 24 A6 20 F7 04 00 00 00 00	1 \$
0384(0180)	46 5A 33 20 20 20 20 20 20 20 10 00 00 00 00	FZ3 >
0400(0190)	00 00 00 00 00 00 2C 22 A6 20 7F 04 00 00 00 00	,."
0416(01A0)	E5 20 20 20 20 20 20 20 20 20 10 00 00 00 00	>
0432(01B0)	00 00 00 00 00 00 6D B7 B4 20 F7 04 00 00 00 00	m+
0448(01C0)	E5 51 20 20 20 20 20 20 20 20 10 00 00 00 00	Q >
0464(01D0)	00 00 00 00 00 00 23 15 B5 20 F8 04 00 00 00 00	#
0480(01E0)	E5 41 4E 47 31 20 20 20 20 20 20 10 00 00 00 00	ANG1 >
0496(01F0)	00 00 00 00 00 00 8E 9E B0 20 F7 04 00 00 00 00	*

Home=beg of file/disk End=end of file/disk

ESC=Exit PgDn=forward PgUp=back F2=chg sector num F3=edit F4=get

name

图 3

修改第 2、3 字节为 20(如图 4)。按 F5 存盘,然后用 ESC 键退出。

Advanced PC Tools 5.0

----- Disk View/Edit Service -----

Path=C:

Absolute sector 0000161, System ROOT		
Displacement	Hex codes	ASCII value
0256(0100)	41 55 54 4F 45 58 45 43 42 41 54 20 00 00 00 00	AUTOEXEC-
0272(0110)	00 00 00 00 00 00 D1 21 A6 20 6C 04 37 00 00 00	BAT
0288(0120)	54 50 20 20 20 20 20 20 20 20 10 00 00 00 00	! 1 7
0304(0130)	00 00 00 00 00 00 0B 23 A6 20 DA 0C 00 00 00 00	TP >
0320(0140)	53 45 54 55 50 20 20 20 42 41 54 20 00 00 00 00	# +
0336(0150)	00 00 00 00 00 00 53 58 46 1A 71 01 76 03 00 00	SETUP BAT
0352(0160)	46 20 20 20 20 20 20 20 20 20 10 00 00 00 00	SXF q v
0368(0170)	00 00 00 00 00 00 6C 24 A6 20 F7 04 00 00 00 00	FOX >
0384(0180)	46 5A 33 20 20 20 20 20 20 20 10 00 00 00 00	1 \$
0400(0190)	00 00 00 00 00 00 2C 22 A6 20 7F 04 00 00 00 00	FZ3 >
0416(01A0)	E5 20 20 20 20 20 20 20 20 20 10 00 00 00 00	,."
0432(01B0)	00 00 00 00 00 00 6D B7 B4 20 F7 04 00 00 00 00	>
0448(01C0)	E5 51 20 20 20 20 20 20 20 20 10 00 00 00 00	m+
0464(01D0)	00 00 00 00 00 00 23 15 B5 20 F8 04 00 00 00 00	Q
0480(01E0)	E5 41 4E 47 31 20 20 20 20 20 20 10 00 00 00 00	# >
0496(01F0)	00 00 00 00 00 00 8E 9E B0 20 F7 04 00 00 00 00	ANG1 >
		*
Home=beg of file/disk End=end of file/disk		
ESC=Exit PgDn=forward PgUp=back F2=chg sector num F3=edit F4=get		
name		

图 4

3.1.3 把子目录名改为对应的小写字母。用 DIR 可显示所加密的子目录,但用相应的目录操作命令无法操作。

这种加密法简单易行,但可靠性不高。经过有限次试验可以查找到该子目录。

3.2 修改属性实现加密

子目录的属性域用一个字节来描述,各位意义是:

b7 b6 b5 b4 b3 b2 b1 b0

保留	保留	档案	子目录	卷标	系统	隐含	只读
----	----	----	-----	----	----	----	----

利用非子目录属性值或组合成的非子目录属性值来完成加密。

3.2.1 改变子目录项的属性为隐含属性 12, 使 DIR 无法列出该子目录名称, 但用 PCTOOLS 等工具软件可搜索出来。

3.2.2 改变子目录属性为卷标属性(08), 由于 DOS 只识别第一个卷标, PCTOOLS 和 DIR 均无法搜索该子目录, 从而避免了那些初学者对该子目录的访问。

还有许多种组合方法, 不再一一说明。

3.3 修改起始簇实现加密

子目录的起始簇指明了该子目录的第一个簇号, 子目录与起始簇是一一对应的, 若改变了起始簇的值, 就可打乱所管理的内容, 搜索不到其中的文件, 从而达到加密目的。由此可见, 对它的修改方法多种多样, 仅举例说明如下:

3.3.1 起始簇死锁法。具体做法是: 改变子目录项中的“文件起始簇”项(位移 1A 和 1B)的内容(注意: 牢记该内容), 使该内容指向其父目录所在的簇号, 使 PCTOOLS 等在搜索路径时死锁, 从而彻底阻止其运行。尽管用 DIR 可看到该子目录, 但对该子目录中的文件是不可访问的, 所访问的只是父目录中的文件。

3.3.2 借用子目录法

具体操作如下:

(1) 首先在根目录下建立一个空子目录(例如 WANG), 用 PCTOOLS 查出“WANG”子目录登记项 1A 和 1B 字节的内容即 F7 和 04。

(2) 找到要加密的子目录 FOX 子目录登记项 1A 和 1B 字节的内容即 0A 和 13(要牢记, 如图 5)。

Advanced PC Tools 5.0

----- Disk View/Edit Service -----

Path=C:

Absolute sector 0000161, System ROOT

Displacement	Hex codes	ASCII value AUTOEXEC
0256(0100)	41 55 54 4F 45 58 45 43 42 41 54 20 00 00 00 00	BAT
0272(0110)	00 00 00 00 00 00 D1 21 A6 20 6C 04 37 00 00 00	117
0288(0120)	54 50 20 20 20 20 20 20 20 20 20 10 00 00 00 00	TP)
0304(0130)	00 00 00 00 00 00 0B 23 A6 20 DA 0C 00 00 00 00 00	# +
0320(0140)	53 45 54 55 50 20 20 20 42 41 54 20 00 00 00 00	SETUP BAT
0336(0150)	00 00 00 00 00 00 53 58 46 1A 71 01 76 03 00 00	SXF q v
0352(0160)	46 4F 58 20 20 20 20 20 20 20 20 10 00 00 00 00	FOX)
0368(0170)	00 00 00 00 00 00 6C 24 A6 20 0A 13 00 00 00 00	1 \$
0384(0180)	46 5A 33 20 20 20 20 20 20 20 20 10 00 00 00 00	FZ3)
0400(0190)	00 00 00 00 00 00 2C 22 A6 20 7F 04 00 00 00 00	.,
0416(01A0)	57 41 4E 47 20 20 20 20 20 20 20 10 00 00 00 00	WANG)
0432(01B0)	00 00 00 00 00 00 6D B7 B4 20 F7 04 00 00 00 00	m +
0448(01C0)	E5 51 20 20 20 20 20 20 20 20 20 10 00 00 00 00	Q)
0464(01D0)	00 00 00 00 00 00 23 15 B5 20 F8 04 00 00 00 00	#
0480(01E0)	E5 41 4E 47 31 20 20 20 20 20 20 10 00 00 00 00	ANG1)
0496(01F0)	00 00 00 00 00 00 8E 9E B0 20 F7 04 00 00 00 00	*

Home=beg of file/disk End=end of file/disk

ESC=Exit PgDn=forward PgUp=back F2=chg sector num F3=edit F4=get name

图 5

(3) 将 0A 和 13 改为 F7 和 04(如图 6)。

Advanced PC Tools 5.0

----- Disk View/Edit Service -----

Path=C:

Absolute sector 0000161, System ROOT

Displacement	Hex codes	ASCII value AUTOEXEC
0256(0100)	41 55 54 4F 45 58 45 43 42 41 54 20 00 00 00 00	BAT
0272(0110)	00 00 00 00 00 00 D1 21 A6 20 6C 04 37 00 00 00	117
0288(0120)	54 50 20 20 20 20 20 20 20 20 20 10 00 00 00 00	TP)
0304(0130)	00 00 00 00 00 00 0B 23 A6 20 DA 0C 00 00 00 00 00	# +
0320(0140)	53 45 54 55 50 20 20 20 42 41 54 20 00 00 00 00	SETUP BAT
0336(0150)	00 00 00 00 00 00 53 58 46 1A 71 01 76 03 00 00	SXF q v
0352(0160)	46 4F 58 20 20 20 20 20 20 20 20 10 00 00 00 00	FOX)
0368(0170)	00 00 00 00 00 00 6C 24 A6 20 F7 04 00 00 00 00	1 \$
0384(0180)	46 5A 33 20 20 20 20 20 20 20 20 10 00 00 00 00	FZ3)
0400(0190)	00 00 00 00 00 00 2C 22 A6 20 7F 04 00 00 00 00	.,
0416(01A0)	57 41 4E 47 20 20 20 20 20 20 20 10 00 00 00 00	WANG)
0432(01B0)	00 00 00 00 00 00 6D B7 B4 20 F7 04 00 00 00 00	m +
0448(01C0)	E5 51 20 20 20 20 20 20 20 20 20 10 00 00 00 00	Q)
0464(01D0)	00 00 00 00 00 00 23 15 B5 20 F8 04 00 00 00 00	#
0480(01E0)	E5 41 4E 47 31 20 20 20 20 20 20 10 00 00 00 00	ANG1)
0496(01F0)	00 00 00 00 00 00 8E 9E B0 20 F7 04 00 00 00 00	*

Home=beg of file/disk End=end of file/disk

ESC=Exit PgDn=forward PgUp=back F2=chg sector num F3=edit F4=get name

图 6

解密时只需将 F7 和 04 改为 0A 和 13 即可。这种加密方法操作简单, 安全度高。无论用 DIR 还是用 PCTOOLS 都无法搜索 FOX 中的文件。

起始簇加密时, 最重要的是加密者本人要记下所加密子目录的起始簇号。否则加密者本人也无法使用该子目录。要恢复所加密的子目录须用 DEBUG 完成。

3.4 修改长度实现加密

由于子目录的长度簇域值为 0, 这也是 DOS 识别子目录的重要标志之一, 把长度值改为非 0 数据也可起到加密的作用。

3.4.1 改变子目录项中的“文件长度”项, 把最后四个字节内容改为 FF, 并改变属性字节内容为隐含属性, 使 DIR 和 PCTOOLS 等均无法显示该子目录。

这种方法仅仅使被加密的子目录不显示, 其它操作一切正常。缺点是如果别人知道加密的子目录, 就可毫无困难地进入该子目录。

3.5 综合加密

将上述方法结合起来使用, 可提高所加密子目录的安全度, 例如: 用 PCTOOLS 将文件名的第一个字节改为 05, 然后将属性值改为 02。

4 小结

上述几种加密方法是在太极-386 硬盘上实现的, 稍作修改, 即可对软盘的子目录进行加密, 望读者能够妥善使用。

参考文献

- 1 张载鸿编. 局部网操作系统 DOS 高级技术分析. 国防工业出版社. 1988 年 4 月.
- 2 张昆苍编著. 操作系统原理 DOS 篇. 清华大学出版社. 1994 年 5 月.

用C语言实现对逻辑硬盘的隐藏

秦相林

(黑龙江商学院计算机中心 哈尔滨 150076)

摘 要 本文重点剖析了含有逻辑硬盘的硬盘内部控制结构;提出了隐藏逻辑硬盘的算法。最后,给出了用C语言实现此算法的实用程序。

关键词 逻辑硬盘 分区表 扩展分区登记项 实际分区登记项 系统标志字节

1 问题的提出

现今的微计算机系统大多使用大容量硬盘,而且为了方便,往往利用DOS的FDISK命令将硬盘化分为几个逻辑硬盘。这样,可以将自己的程序和数据放入其中的一个逻辑硬盘下。但是,对于那些从事科研等工作的人来说,却又希望自己的程序、数据不被别人随意浏览和修改。为此,笔者寻找了一种能隐藏逻辑硬盘的办法。

2 逻辑硬盘的内部控制结构及隐藏方法

首先来剖析一下用FDISK建立的逻辑硬盘的内部控制结构。

硬盘分区表位于硬盘主引导记录所在的0柱面、0磁头、1扇区,自偏移1BEH处开始,共占六十四个字节,每十六个字节为一个分区登记项,这样共有四个分区的信息。由于DOS3.30以下和DOS3.31及以上的版本对硬盘分区的划分情况不同,而目前的绝大多数机器使用高版本DOS,所以本文重点讨论高版本的情况。对于在DOS3.31以上版本来说,其中的第三、第四个分区登记项没有使用,而以00H覆盖。

第一个分区登记项(自本扇区1BEH偏移处开始、1CDH偏移处结束)是实际分区登记项,用来说明C盘的起始位置、硬盘空间的占用情况以及操作系统的类型等。C盘的操作系统类型由自本扇区1C2H偏移处的字节描述。

第二个分区登记项(自本扇区1CEH偏移处开始、1DDH偏移处结束)是扩展分区登记项,用来说明有无扩展分区,及扩展分区表所在位置(如果有扩展分区的话)。该分区登记项中与本文有关的字节含义如下:

自该扇区偏移1D2H处:系统标志。该字节的数据表明有无扩展分区。如果为05H,则说明有扩展分区;如果为00H,则说明无扩展分区;如果为05H,则有下面的解释:

自该扇区偏移1CFH处:第一个扩展分区表所在的磁头号;

自该扇区偏移1D0H处:第一个扩展分区表所在

柱面号的高2位和扇区号(6位);即将该字节以二进制形式展开,分成高2位和低6位。

自该扇区偏移1D1H处:第一个扩展分区表所在柱面号的低8位。

这里,给出一个实例。

比如,将一个硬盘的0柱面0磁头1扇区的数据读出后,得到自该扇区偏移1D2H处的系统标志字节为05H,这说明有扩展分区。

偏移1CFH处的数据为00H;偏移1D0H处的数据为81H,相当于二进制的10000001;偏移1D1H处的数据为42H,相当于二进制的01000010。经合成得出柱面号为1001000010,相当于242H;扇区号为000001,相当于01H。于是,第一个扩展分区表在第242H柱面,0磁头(由偏移1CFH处的字节描述),1扇区,它在该扇区中的位置和硬盘分区表类似,自偏移1BEH处开始。

把第242H柱面0磁头,1扇区的数据读出,得到第一个扩展分区表的数据。

该扩展分区表中的结构与硬盘分区表的结构类似,第一个分区登记项仍是实际分区登记项,但它是用来说明D盘的起始位置、硬盘空间的占用情况以及D盘的系统类型等。其中该实际分区登记项中自该扇区偏移1C2H处的字节为D盘的系统标志字节,用来说明D盘的类型。如果其值为00H,则说明D盘没有使用;如果为01H、04H或06H,则说明是DOS系统类型。至于其值究竟是01H、04H还是06H,这与D盘所占的硬盘空间有关。由于此时读出的系统标志字节为06H,说明有D盘,且是DOS类型。

该扩展分区表中偏移扇区开始1D2H处的系统标志字节(它是第二个分区登记项中的内容)为05H,由于该说明项是扩展分区登记项,则数据05H说明还有扩展分区。读出偏移1D0H处及1D1H处的数据分别为81H和F9H,相当于二进制的10000001和11111001。合成后的柱面号为1011111001相当于2F9H;扇区号为000001,即01H;磁头号为00H(由偏移1CFH处的数据得到)。于是,第二个扩展分区表在硬盘上的位置是2F9H柱面、0磁头、1扇区,自该扇区

1BEH 偏移处。然后将该扇区的数据读出,从而得到 E 盘的情况及第三个扩展分区表的位置。

这样一直做下去(就可以类似地得出表示 E、F 等逻辑硬盘系统类型的系统标志字节的位置),直到有那样一个扩展分区表:自该分区表所在扇区偏移 1D2H 处的系统标志字节(即该扩展分区表中第二个分区登记项中的字节)为 00H,则说明以后不再有扩展分区表,也就相当于不再有其它的逻辑硬盘。

可以看出,每一分区表中的扩展分区登记项实际上指示了下一个扩展分区表的位置(柱面、磁头号、扇区号)。每一个扩展分区表都含有一个实际分区登记项,实际上是逻辑硬盘的说明。它指出该逻辑硬盘在物理盘上的实际起始、终止位置及其系统类型;第一个扩展分区表中的实际分区登记项用来说明 D 盘的情况,依此类推;如果哪一个扩展分区表中实际分区登记项的系统标志字节(它位于该扩展分区表所在扇区的偏移 1C2H 处)为 00H,则机器就会绕过其所对应的逻辑硬盘,而将下一个逻辑硬盘当成这一逻辑硬盘。因此,通过将某一逻辑硬盘对应的这一字节改为 00H(以表示没有使用)的方法来隐藏这一逻辑盘;若恢复隐藏,则只需重新将这一字节原来的值恢复过来即可。而且,任何一个逻辑硬盘被隐藏,机器都不会再占用属于它的硬盘空间。

3 实现的算法

考虑到读者运用此手段的方便性,以及一些读者对逻辑硬盘内部控制结构存在着不同程度的理解,本文给出了能实现对逻辑硬盘隐藏的实用程序。下面是用程序实现隐藏及恢复逻辑硬盘的算法。

这里,利用数组 `buffer[]` 存放分区表所在的扇区数据;利用数组 `part[][]` 存放每个逻辑硬盘对应的扩展分区表的位置(柱面号、磁头号和扇区号)。其中, `part[i][1]`、`part[i][2]`、`part[i][3]`、`part[i][4]` 分别表示第 `i` 个扩展分区表所在扇区的柱面号、扇区号、磁头号和该表所对应逻辑硬盘的系统标志字节。而且,使用三个变量: `cyl`、`head` 和 `sec`, 分别表示柱面号、磁头号和扇区号;使用变量 `num` 来统计硬盘上所含逻辑盘个数。

该程序被执行时,屏幕出现“logical disk code attrib system sign byte”的字样。其中:

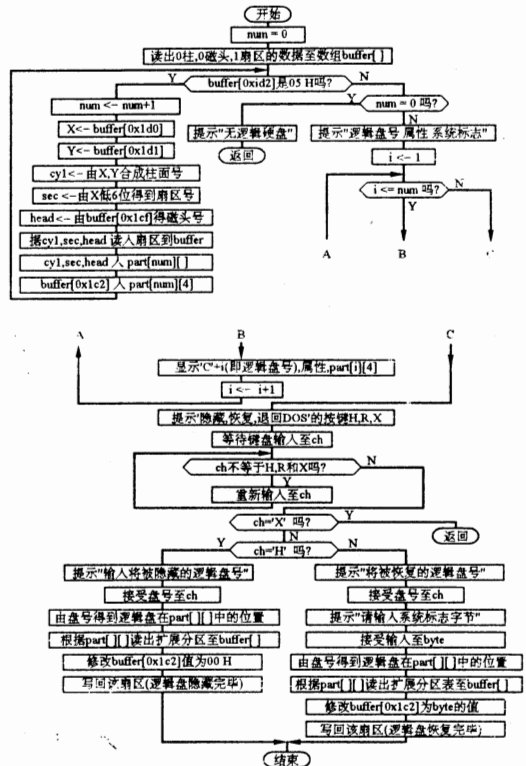
“logical disk code”下面显示出所有逻辑硬盘的盘符。

“attrib”下面显示出对应的逻辑盘的属性,若逻辑盘此时正常,则显示“normal”;若已被隐藏,则显示“hidden”。

“system sign byte”下面显示出对应逻辑硬盘的系统标志字节(1、4 或 6),如果想隐藏哪个盘,就要事先记住该盘的系统标志字节,以便在恢复它的时候使用。如果此时该盘已被隐藏,则该字节为‘0’。

此外,屏幕上还出现三个选择:H、R 和 X,分别表示“隐藏”、“恢复”和“退出”。如果选择“H”,则屏幕出现提示,要你输入你要隐藏的逻辑盘符;如果选择“R”,则系统提示你,输入你要恢复的逻辑盘符,之后还要输入该盘的系统标志字节。

只要在完成上述工作后,重新启动你的计算机即可。



4 源程序清单

```
#include<stdio. h>
#include<bios. h>
#include<fcntl. h>
#include<sys\types. h>
#include<sys\stat. h>
#include<stdlib. h>
void main(void)
{
    int part[22][4],result,num,i,x,y,xy,cyl,sec,head,t;
    char buffer[512],byte,ch;
    clrscr();
    num=0;
    result=biosdisk(2,0x80,0,0,1,1,buffer);
    if(result){
        printf("failed in reading partision table !!! \n");
        exit(1);
    }
    while(buffer[0x1be+0x04+0x10]==0x0005){
        num++;
        x=buffer[0x1be+0x10+0x02]&0x00ff;
```

```

y=buffer[0x1be+0x10+0x03]&0x00ff;
xy=_rotr(x,6)&0x03;
cyl=_rotr(xy,8)+y;
sec=x-_rotr(xy,6);
head=buffer[0x1be+0x10+0x01];
biosdisk(2,0x80,head,cyl,sec,1,buffer);
part[num][1]=cyl;
part[num][2]=sec;
part[num][3]=head;
part[num][4]=buffer[0x1be+0x04];
}
if(buffer[0x1be+0x04+0x10]!=0x0){
printf("\n");
printf("partition table error !!! \n");
exit(1);
}
if(num==0){
printf("there is no logical disk !!! \n");
exit(0);
}
printf(" logical disk code attrib system sign byte\n");
printf(".....\n");
for(i=1;i<=num;i++){
if(part[i][4]==0)
printf("    %c    hidden    %x\n",'C'+i,part[i]
[4]);
else
printf("    %C    normal    %x\n",'C'+i,part[i]
[4]);
}
ch='';
printf(".....\n");
printf("Hidden a logical Disk--H Restore a logical Disk--
R\n");
printf("exit to DOS --X\n");
printf("\n");
printf("Please input option(H,R OR X):\n");
while(ch!='h'&&ch!='H'&&ch!='R'&&ch!='r'&&ch!='x'&&ch!='X'){
ch=getche();
if(ch=='x' || ch=='X'){
exit(0);
}
if(ch=='h' || ch=='H'){
printf("\n");
printf("input logical disk code to be hidden(D to %c)
\n",'C'+num);
ch=getche();
ch=tolower(ch);
while((ch-'c')>num || (ch-'c')<=0){
printf("\n");

```

```

printf("invalid code,input again !!! \n");
ch=getche();
}
t=ch-'c';
biosdisk(2,0x80,part[t][3],part[t][1],part[t][2],
1,buffer);
buffer[0x1be+0x04]=0x0;
biosdisk(3,0x80,part[t][3],part[t][1],part[t][2],
1,buffer);
clrscr();
printf("Succeed in hidden the logical disk !!! \n");
printf("Please restart your computer... \n");
}
if(ch=='r' || ch=='R'){
printf("\n");
printf("Input the logical disk code to be restored(D to
%c):\n",'C'+num);
ch=getche();
ch=tolower(ch);
while((ch-'c')>num || (ch-'c')<=0){
printf("\n");
printf("invalid code,input again !!! \n");
ch=getche();
}
printf("\n");
printf("input the system sign byte(1,4,OR 6):\n");
byte=getche();
while(byte!='1'&&byte!='4'&&byte!='6'){
printf("\n");
printf("invalid value,input again !:\n");
byte=getche();
}
t=ch-'c';
biosdisk(2,0x80,part[t][3],part[t][1],part[t][2],
1,buffer);
buffer[0x1be+0x04]=byte-'0';
biosdisk(3,0x80,part[t][3],part[t][1],part[t][2],
1,buffer);
clrscr();
printf("Succeed in Restoring the logical disk !!! \n");
printf("Please restart your computer... \n");
}
}

```

5 结束语

本文给出的程序在 Borland 的 TURBO C++1.0 环境下编译通过;所讨论的内容在 DOS6.0 版本管理下的 200MB 硬盘上得到了验证。该程序不仅能在 DOS 环境下运行,而且不加任何转换就能在中文 WINDOWS 3.1 下直接运行。

ATM 网络多媒体通信的具体实现

浦 刚 周曼丽 魏蛟龙

(华中理工大学电子与信息工程系 武汉 430074)

摘 要 本文围绕如何真正实现 ATM 网络中特有的多媒体通信的问题,对 ATM 局域网仿真和 ATM 网络通信的内部机制进行详细的讨论。在实践中,我们采用 Fore System 公司的 ATM 交换机 ASX-200 构造了 ATM 网络,并通过对 ATM 网卡(ESA 200 PC Card)的 ATM API 进行编程,能真正实现在 ATM 主机之间同时进行视频、语音和数据的实时通信。

关键词 ATM 网络 多媒体通信 ATM 局域网仿真 ATM API ESA 200 PC CARD 视频语音和数据实时通信

0 引言

ATM 技术使用异步时分(Asynchronous Time Division,ATD)复用技术的快速分组交换方式,以其灵活、高速的交换结构使 B-ISDN 能够完成从窄带语音、数据业务到宽带视频通信的综合传输,从而成为了 B-ISDN 的核心技术。目前 ATM 在电话、视频传送、远地高速业务、局域网、段(Segment)交换及虚拟局域网等领域应用已经逐渐成熟,各方面的产品已走向商业化。

ATM 网络提供了动态带宽分配、数据的可变速率传输、多个视频业务共享信道和可由用户(而不是信道速率)控制的恒定的图像质量等新的网络特性,那么,实际应用中究竟如何利用这些特性来实现 ATM 网络特有的多媒体业务呢?下面,我们将围绕 ATM 局域网仿真、ATM 网络通信的内部机制和 ATM 网络多媒体通信的具体实现这三方面来进行讨论。

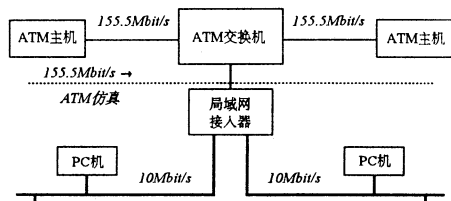


图1 ATM网络开发环境

1 ATM 局域网仿真

为了与传统的局域网透明地交换数据流,ATM Forum 定义了局域网(LAN)仿真。LAN 仿真的目的是为已存在的局域网协议(IP、IPX、Appletalk、DECnet、Banyan Vines 等等)提供一个在 ATM 上运行的方法,决定了 ATM 如何与已有的以太网、令牌网和 FDDI 网共同操作。LAN 仿真提供了三个功能:①数据的包装和发送、②地址解释、③多点广播组的管理。当

数据业务量不是很大时,利用 ATM 局域网仿真和 ATM 网络 155.5Mbit/s 的带宽能够较好地支持一些多媒体通信。但是,LAN 仿真这种方法只适合于数据传输,而不适合于多媒体的通信,因为 LAN 采用的是共享网络带宽且其通信实时性不够好,这时候 ATM 交换机只是相当于一个高速的分组交换机。当网络中视频、音频等多媒体数据增长或主机有突发数据要传输时,同一条链路上的多媒体通信之间以及多媒体通信与数据通信之间都会相互影响,以致造成信元的时延抖动和大量丢失,使得多媒体传输的质量大大降低。因此,采用 ATM 局域网仿真的方法来实现多媒体业务并不能充分利用 ATM 带来的网络新特性,不足以保证多媒体通信的质量。

2 ATM 网络通信的内部机制

为了开发 ATM 网络特有的多媒体通信,我们应该弄清楚 ATM 网络通信与一般局域网通信其内部机制的区别。ATM 网络通过用来引导信元从源节点到达目的地的一系列映射来建立所需的虚通路(VC)和虚通道(VP),因此其网络通信是 Connection Based,而且连接是单向的。

A、虚连接

ATM 的基本特点就是面向连接方式工作:在信息传送到网络之前,必须先有一个逻辑(虚)连接建立阶段,这个阶段使网络预留必要的资源(如果有资源可用)。如果没有足够的资源可用,就会向请求的终端拒绝这个连接。

当信息传送阶段结束之后,资源被释放。这种面向连接的工作方式使网络在任何情况下都能保证一个最小的分组丢失率,即一个最高的质量。每个虚连接将由一个号码(标志)来识别,而这个号码仅在虚连接的每段链路上具有局部意义。

B、虚通路(VC)

由于 ATM 网是面向连接的,每个连接由信头的 VCI 子段来标识,而 VCI 值在呼叫建立时分配。一个

VCI 仅在 ATM 节点之间的一段链路上有局部意义, VCI 值在 ATM 节点上进行翻译。当一个连接释放时, 相关链路上的 VCI 值将被释放, 这些值可以重新用于其它连接。

这种 VCI 方法一个有趣的优点是可用多个 VCI 值来表示多成分业务, 象可视电话、TV 等。例如, 可视电话由 3 个成分组成: 数据、语音、图象, 这 3 个成分可以在分开的 VCI 上传送, 这样在连接过程中网络可以根据实际情况动态地加入或去掉某些成分。

C. 虚通道(VP)

未来的宽带网支持端点之间的半固定连接, 这种半固定连接必须传输大量同时存在的连接。这个概念叫做虚通道或虚以网。按照这个概念, 网络资源以半固定方式分配。这样可以对可用的网络资源进行简单而有效的管理, 虚通道由信头的 VPI 子段来标识。在 ATM 节点处只对 VPI 进行交换。每个通道最多可由 64K 条虚通路组成。每条用自己的 VCI 来标识。ATM 网络中 VP, VI 示意图如图 2 所示。

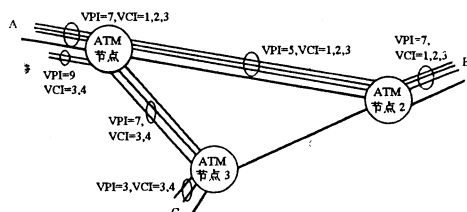


图2 ATM网络中VP, VI示意图

ATM 节点处 VPI 交换关系

ATM 节点 1		ATM 节点 2		ATM 节点 3	
VPin	VPout	VPin	VPout	VPin	VPout
7	5	5	7	7	3
9	7				

D. 优先级

ATM 中可将信头置位为不同的优先级来区分逻辑连接。优先级有两种: 时间优先级和语义优先级, 它们都可以以连接为基础分配(每个 VPI 或每个 VCI), 也可以信元为基础分配。在前一种情况下, 一个虚通路/虚通道中的所有信元具有相同的优先级; 在后一种情况下, 一个虚通路/虚通道中的信元可能会有不同的优先级。优先级的控制可以保证过负载情况下只有低优先级的连接会丢失信息。

由于 ATM 传输时采用统计时分复用的方法, ATM 网难免会出现拥塞和信元丢失和时延抖动现象, 这些都会降低多媒体通信的质量。因此, 根据不同源地多媒体数据传输的需要, 我们可以利用虚通道 (VP) 来建立不同的虚连接, 在同一虚通道中为视频、语音、数据等分别分配不同的虚通路 (VC), 同时可以为不同的 VP 或 VC 分配不同的优先级, 以保障对时

差敏感的多媒体数据的传输不受网络数据流量变化的影响; 在另一方面也可以同多媒体编码结合起来, 将带有重要编码信息的信元分配为高优先级信元, 带有次要编码信息的信元分配为低优先级信元, 以减少信元丢失对终端多媒体数据重建的影响, 从而真正实现“保证质量”的多媒体通信。

3 ATM 网络多媒体通信的具体实现

下面, 我们以 Fore System 公司的 ATM 交换机 ASX-200 为例, 在如图 1 的 ATM 主机上都装有 ATM 网卡 (ESA 200 PC Card) 和 Windows 95 多任务操作系统。因此可以在两台 ATM 主机之间同时进行视频、语音和数据实时通信。

根据以上的分析, 为了充分利用 ATM 网络带来的新特性, 我们对视频、语音和数据的传输采用不同的 ATM VC, 同时为每个 VC 分配自己的 QoS (Quality of Service) 类别, 也就是保证了不同媒体所需的不同优先级的传输质量。ATM VC 的建立只能通过 ATM 网卡 (ESA 200 PC Card) 的 ATM API 实现。ATM 网卡的软件结构如图 3 所示, 软件分为运行在主机上和 i960 控制处理器上 (200 系列网卡上) 的两部分。运行在 i960 上的软件负责 ATM 信元处理和 ATM 适配层的处理 (AAL3/4 和 AAL5), 在主机上的软件, 或称为设备驱动程序, 负责 ATM 信令和 ATM 地址处理 (包括 Fore Systems 的 SPANS 协议), 并向操作系统中的高层协议模块提供接口, 如 TCP/IP、Fore Systems 的 ATM API。另外, 设备驱动程序还包括一个 SNMP 代理, 跟踪 ATM 统计数据并把它们报告网络管理平台。

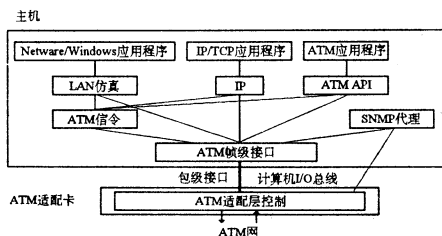


图3 ATM网卡软件结构

我们通过 ATM API 的编程来控制 ATM 网卡, 使它向 ATM 交换机发送我们所要求的 QoS 参数, 如果 ATM 网能够提供用户所需要的服务质量, 则会在目的站和源站之间建立一条 VC, 并保证服务质量。我们的开发平台 Windows 95 是 32 位的多任务操作系统。因此, 我们采用了 32 位的编程语言 Visual C++ 4.0 进行编程。Visual C++ 4.0 中提供了许多新的特性, 通过这些特性可以方便地利用多线程编程对视频、语音和数据的同时传输进行控制, 并且可以充分运用 Windows 95 提供的接口对多媒体数据进行 32 位的快速处理。

我们很容易注意到 ATM 网卡软件结构中, 还提

存储 IC 卡的读写方法

龚荣武 唐宁九 孙界平

(四川联合大学计算机系 成都 610054)

摘 要 介绍存储 IC 卡及其使用的芯片的性能、特点、引脚功能以及与 PIC16C54 微控制器的接口和读写方法。

关键词 IC 卡 串行 EEPROM 微控制器 指令 时序

1 概述

IC 卡(Integrated Circuit Card)又称为智能卡(Smart Card)是法国人 Rolornd Morono 于 1974 年发明的,距今已有近 20 年的历史。为促进 IC 卡在全世界范围内的推广和使用,国际标准化组织于 1987 年专门为 IC 卡制定了国际标准 ISO/IEC 78/6-1-2-3,对接触型 IC 卡的结构尺寸、通信协议作了规定。

IC 卡是继磁卡、条码卡之后的又一枝新秀。条码卡存储的信息单一且不能修改,又没有加密性,所以只能作为一种标识代码,供信息查询。磁卡存储的信息容量较条码卡大,易于读、写,所以应用较广泛。但磁卡也有它的缺点,比如说抗破坏性。例如在强磁场、静电、扭弯、刮伤等情况下存储在里面的信息容易被丢失。磁卡本身价廉,但磁卡的读卡装置售价都在几百元,而复杂一点的读卡装置要卖到几千元。在使用寿命上磁卡也不如 IC 卡。IC 卡是把一种具有电可擦除的串行 EEPROM 集成电路芯片嵌于塑料基卡之内。它的信息保存期都在 100 年以上。读/写次数高于 100000 次以上。所以一张 IC 卡至少也可以使用十年以上。IC 卡过去在欧洲使用较多,但据国际电子商情 96 年 5 月的报导,今年在美国亚特兰大开幕的奥运会期间,美国东南部的三大银行与全球最大信用卡集团 Visa International 合作推出 150 万张 IC 卡以供人们购物之用。由于 IC 卡系统具有很强的加密性,首先是芯片的容量大,最小容量也达 128 个字节,最大容量可达几千个字节。且对存储器可以任意的分区读写,在不同的区域

存放不同的信息,这就可以作到一卡多用。在保密方面,对普通的存储卡,采用特定的技术,也具有较强的保密性。对于加密存储卡,存储区的访问受逻辑电路的控制,只有密码核对正确后,才能进行读写操作。密码核对有次数限制,密码核对不正确超过规定的次数,IC 卡将自动拒绝读取卡中的信息。由于 IC 卡为可以随身携带的数字电路。因此,数字电路的各种硬件加密手段都可以用于 IC 卡系统中,以提高系统的加密性。

根据 IC 卡所使用的集成电路的不同,可以将 IC 卡分成接触型和非接触型两大类,其中接触型又分为:

(1)存储 IC 卡,卡中的集成电路为一般的串行 EEPROM。这种卡的主要功能是存储信息。其容量可以在 1024bits—16384bits,可以按 8bits 划分成一个字节,故容量为 128 字节—2K 字节。这样大的存储量在使用上也是相当方便了。

(2)加密存储卡,其容量为 1Kbits~2Kbits。芯片自身带有密码比较逻辑,芯片上电后,首先问芯片输入密码,密码位数与具体芯片有关,然后芯片进行密码比较。连续几次比较不成功(次数与具体芯片有关),芯片将死锁。

(3)CPU 卡,卡中的芯片是带有加密逻辑和串行 EEPROM 的增强性 INTEL 80C31 微处理器,内部有 8K 字节 EEPROM 程序存储器,8K 字节 EEPROM 应用数据存储器,768 字节内部 RAM,512 位公共钥匙算法协处理器。这种芯片的 IC 卡可以为安全性要求特别高的应用提供安全措施。

非接触型 IC 卡又叫射频卡,卡中的集成电路除了带有加密逻辑,串行 EEPROM、微处理器外,还带有射频收发的相关电路。

收稿日期:1996 年 7 月 20 日

供了利用 TCP/IP 和 LAN 仿真(如 IPX 或 SPX 等)开发方式。在实验中,我们通过人为地造成了网络数据量大幅度的增加,并同时采用不同传输机制开发的多媒体通信软件的性能进行考察。实验结果十分明显:随着网络的不断拥塞,基于 TCP/IP 和 LAN 仿真的多媒体数据传输由于缺乏内部机制的保障,其传输效果越来越差,不但实时性差,而且数据丢失严重;而基于 ATM API 和 ATM 网卡开发的多媒体数据传输,真正

达到了“保证质量”的视频、语音、数据的同时实时传输,充分实现了 ATM 网络带来的多媒体新特性。

参考文献

- 1 ForeRunner ASX—200 ATM Switch User's Manual
- 2 ForeRunner ESA—200PC ATM EISA Bus Adapter for Personal Computer User's Manual
- 3 《异步传递方式》,马丁·德·普瑞克 著 程时端 刘斌 译,电子工业出版社。

由此可见所有的 IC 卡的一种基本功能就是通过串行 EEPROM 来读/写信息。IC 卡作为一种信息的载体,使用者可随身携带,从而方便了很多应用领域。例如作成 IC 卡电话,IC 卡作成的银行信用卡等。

2 接触型存储 IC 卡使用的集成电路芯片的简介

(1) 存储器卡使用的集成电路

美国 ATMEL 公司的 AT24C01/02/04/08/16

美国国家半导体公司的 NM24C02/C04/C08/C16

这两家公司的这种集成电路为二线式(即有一根时钟信号线,一根数据 I/O 线)的串行 EEPROM。

上述 AT24 的存储容量分别为 128 字节/256 字节/512 字节/1024 字节/2048 字节, NM24 的存储容量分别为 256 字节/512 字节/1024 字节/2048 字节。

用这样的芯片作成的集成电路的引脚为:

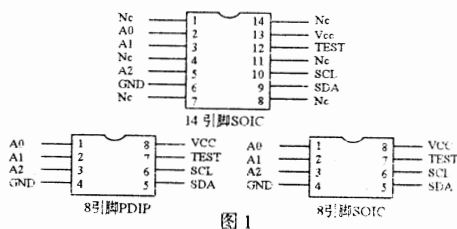


图 1

其中 A2—A0 可外接高电平或低电平,代表器件地址。SCL 为时钟信号输入端, VCC 为正电源输入端, 输入电压为 3V—6V 范围内有效。GND 为接地端。

用这样的芯片作成的 IC 卡的引脚为:

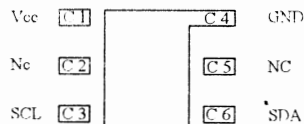


图 2

作成 IC 卡时,器件地址为 000,即 A2A1A0 均接地。

美国国家半导体公司还提供 NM24C03/C05/C09/C17,其存储容量为 256 字节/512 字节/1024 字节/2048 字节。它与 NM24C02/C04/C08/C16 系列不同之处在于多了一个写保护引脚。当将 WP 引脚接 VCC 时,串行 EEPROM 的存储区的上半部分将不允许写入,只能读出。当将 WP 引脚接地时,可以读写整

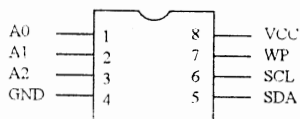


图 3 八引脚封装

个存储区。由这种芯片作成的集成电路引脚为:

(2) AT24 或 NM24 系列集成电路的工作方式

(1). 时钟和数据传送:当 SCL 时钟信号处于高电平时,此时 SDA 引脚作稳定的逻辑状态,代表输入或输出的 1bit 的数据信号。高电平代表“1”,低电平代表“0”。作为数据 I/O,只有当 SCL 处于低电平时,才可改变 SDA 的逻辑状态。比如说由“1”变“0”,或由“0”变“1”。

当 SCL 处于高电平时,如果 SDA 由高变低,则芯片认为是起始信号,如果 SDA 由低变高,则芯片认为停止信号。其总线时序为:

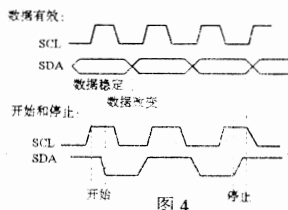


图 4

无论是发送数据还是接受数据,其数据均以字节为一个单位,且 D7 在最前面, D0 在最后面。

(2) 器件地址:对 AT24 系列或 NM24 系列,在向芯片发出开始命令后,均需要向芯片提供器件地址,使器件能进行读写操作,器件地址应按下表的规定使用。

器件型号	器件地址
24C01, C02	MSB LSR [1 0 1 0 A2 A1 A0 R/W]
24C04	MSB LSR [1 0 1 0 A2 A1 P0 R/W]
24C08	MSB LSR [1 0 1 0 A2 P1 P0 R/W]
24C16	MSB LSR [1 0 1 0 P2 P1 P0 R/W]

图 5

表中 A2A1A0 应与集成电路芯片引脚的连接对应。当 A0 端接入高电平时, A0 为 1, 否则为 0, 但已经作成 IC 卡时, A2A1A0=000。

P2P1P0 表示该位器件地址为 0。

R/W 在读器件数据操作时该位为 1, 向器件写数据时, 该位为 0。

(3) 写操作和读操作应按下面的时序图进行编程。
写字节操作:

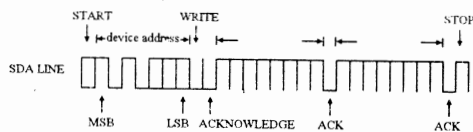


图 6

即首先向器件发起起始命令, 然后输入器件地址, 此地址的 LSB—0 表示按写方式工作, 从器件获取一位

低电平应答信号。只有得到了这一位应答信号,才表明器件已正确地收到了器件地址,然后再输入一个字节的字地址,字地址是指器件内部的存储单元地址由 0 到它的最大地址值。再从器件读取一位低电平的应答信号,表明器件已正确地收到了字地址。最后向器件写入一个字节的数据,再从器件读取一位应答,最后发停止操作命令。

写页面操作:

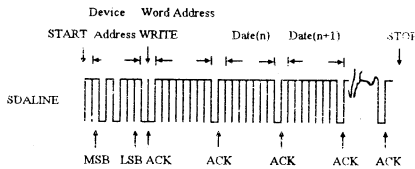


图 7

对 1KB/2KB 的 EEPROM 能进行 8 字节页面写入,4KB,8KB 和 16KB 的 EEPROM 能进行 16 字节页面写入。发出写页面与写字节一样,但只需发出第一个数据字节的地址,以后芯片可以自动进行地址+1 的工作,故可以连续写入 8 个字节或 16 个字节的数据,最后收到 STOP 命令而停止写页面操作。

随机地址读取操作:

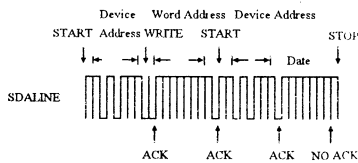


图 8

此操作,首先向芯片发出起始指令,以写方式向芯片写入器件地址,在接到芯片的应答后,向芯片写入字地址,在收到芯片的应答后,再向芯片发出起始命令,以读方式向芯片发出器件地址,在收到应答后,即可向芯片读取指定字地址的一个字节的数据。然后给出 STOP 命令,停止随机地址读操作。

除上面所述的方式外,还有立即读取,顺序读取,可查询芯片的手册,在此不再一一说明。

3 读写 24 系列的实现方法

24 系列的读写操作的最高工作频率为 100KHz,它们是 CMOS 电路,所以 I/O 电平要满足 CMOS 要求。为实现读写操作,当然要选用一种 CPU,且它的 I/O 通道要具有位操作功能。例如用 80C31 的 P1 口就可以。但考虑到 80C31 组成系统复杂且实际使用时 80C31 的很多功能用不上,我们选用 Microchip 公司的 PIC16C54 微控制器。它的 B 口、A 口均可位操作,它有 32 条 RISC 指令,完全可满足要求,系统所需外围器件相当少。系统硬件的构成如下图所示:

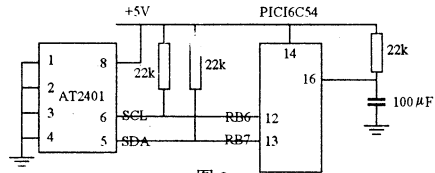


图 9

下面给出了读写 AT24 系列的三个主要子程序,即发起起始命令子程序,向 AT24 写 8bit 数据的子程序,数据在调用子程序前应放在 TXBUF 单元中,此数据可以是器件地址,也可以是字地址,也可以是数据;向 AT24 读取一个字节的子程序,读出的数据放在 RXBUF 单元中。通过开发系统的仿真运行证明读写正确。

其程序如下:

```
SLA EQU 0
RTCC EQU 1
PC EQU 2
STATUS EQU 3
CARRY EQU 0
DCARRY EQU 1
ZERO EQU 2
PD EQU 3
TO EQU 4
PA0 EQU 5
PA1 EQU 6
PA2 EQU 7
FSR EQU 4
RA EQU 5
RB EQU 6
RC EQU 7
TEMP1 EQU 8
TEMP2 EQU 9
TEMP3 EQU 10
TEMP4 EQU 11
ICTIME1 EQU 16
ICTIME2 EQU 17
TXBUF EQU 18
RXBUF EQU 19
B COUNT EQU 20
COUNT EQU 21
EEPROM EQU 22
ADDR1 EQU 23
ADDR2 EQU 24
RRXX1 EQU 25
RRXX2 EQU 26
ICIOCW EQU 27
ATIOW EQU 28
ICS DA EQU 5
ICSCL EQU 4
ATSDA EQU 7
ATSCL EQU 6
SAVEKEY EQU 28
TRISBUF EQU 12
ORG $00
MOVLW $3F
RB
ATSTAR BSF RB,ATSDA;发送起始命令的子程序
NOP
BSF RB,ATSCL
NOP
BCF RB,ATSDA
NOP
BCF RB,ATSCL
RETLW 0
ATTX MOVLW 8;发送 8bit 数据的子程序
```

真彩色图象的模拟显示

张二虎

(西安理工大学 805# 710048)

摘要 本文介绍了 VGA 显示器上模拟显示真彩色显图象的方法,颜色抽取策略,并阐明了色族的划分及数据结构分析。

关键词 模拟显示 分辨率 色差 色族

0 引言

真彩色图象文件中,红绿蓝分量各用一个字节表示,故可有 2^{24} 种颜色。而 VGA 显示器一般同屏可显示 16 色或 256 色,怎样在 VGA 显示器上模拟显示真彩色图象呢?

真彩色图象的模拟显示有两种方法:一是抖动法;二是选色法。抖动法即是用相邻几点各具一定色,来模拟一点较复杂的颜色。它是以牺牲空间分辨率以求多色的,由于 VGA 显示器在 256 色方式下的分辨率为 320×200 ,分辨率较低,所以常采用选色法。

1 选色法介绍

选色法的基本思想就是将原图象中的所有颜色归

收稿日期:1996 年 4 月 6 日

```

MOVWF COUNT
MOVLW $3F
TRIS RB
ATTXLP1 BTFSS TXBUF,7
        GOTO ATBIT0
        BSF RB,ATSDA
        NOP
        GOTO ATCLK1
ATBIT0 BCF RB,ATSDA
        NOP
ATCLK1 BSF RB,ATSCL
        NOP
        BCF RB,ATSCL
        BCF 3,0
        RLF TXBUF,1
        DECFSZ COUNT
        GOTO ATTXLP1
        MOVLW $BF
        TRIS RB
        BSF RB,ATSCL
        NOP
        BTFSC RB,ATSDA
        BSF EEPROM,1
        NOP
        BCF RB,ATSCL
        BTFSS EEPROM,7
        GOTO RET2
        MOVLW $3F
        TRIS RB
        BCF RB,ATSDA
        NOP
        BSF RB,ATSCL
        NOP
        BSF RB,ATSDA
        NOP
        BCF RB,ATSCL
        BCF RB,ATSDA
RET2 RETLW 0
RATDAT MOVLW 8;读数据的子程序
        MOVWF COUNT
        CLRFB RXBUF
        MOVLW $BF
        TRIS RB

```

```

BCF EEPROM,0
BSF RB,ATSCL
NOP
BTFSC RB,ATSDA
BSF EEPROM,0
NOP
BCF RB,ATSCL
BTFSC EEPROM,0
BSF RXBUF,0
BCF 3,0
RLF RXBUF
DECFSZ COUNT
GOTO ATRXLP1
RRF RXBUF
BSF EEPROM,0
MOVLW $3F
TRIS RB
BTFSS EEPROM,0
GOTO ATBITOU
BSF RB,ATSDA
NOP
GOTO ATCLK2
ATBITOU BCF RB,ATSCL
        NOP
ATCLK2 BSF RB,ATSCL
        NOP
        BCF RB,ATSCL
        RETLW 0

```

4 结束语

由于 IC 卡的很多优点,它将会对磁卡提出挑战。例如在金融业可以作为“信用卡”,“电子钱包”IC 卡,它将为提供一种新的服务手段。在通讯业,可以用它作为预先付款卡,像今天的磁卡电话中磁卡所起的作用一样。在交通运输业,消费娱乐业,医疗保险业等方面均有很大的应用前景。未来多功能 IC 卡的普及,将使一张 IC 卡作为进出门的钥匙,运输购物的凭证,电子邮件的入网许可证等多种用途。它是人类通向电子时代的钥匙。可否这样来估计,有朝一日,由于 IC 卡的普及,就可说“一卡在手,可走遍全球”。

并成为 256 色,进行显示,即抽出 256 种代表色。

颜色抽取的方法有三种:

(1)将颜色空间中的所有颜色分成 256 个区域,各区域用一种颜色表示。

(2)一幅图象所记录的颜色往往局限于某一范围,可在此范围内划分为 256 个区域。

(3)直接计算各像素颜色的色差,合并色差较小的颜色。

从图象整体效果来看,一幅具体的图象一般都有一个主色调。例如人像常以红、黄色为主,风景则多为蓝、绿色,颜色均匀分布的情况极少。这样如果预先设置 256 色,则这一组颜色很难适应各种图象。假若在红、黄区色彩较多,则显示风景时就显得呆板、无层次感,加上某些特殊效果的图象,如中国水墨画,黑色层次丰富,却又有细微的色调变化,这就需要非常特殊的调色板。因此设计一个“万能”调色板不可能。前面的两种方法虽简单,却不实用。

通常选用第(3)种方法,可以做到“按图取色”。总的原则是:

- (1)各像素颜色均有一近似色。
- (2)主色调颜色应保留较多颜色数。
- (3)充分利用调色板的每一项颜色。
- (4)根据图象处理的偏重需要而适当修改选色原则。

2 颜色抽取策略

2.1 色差理论

颜色抽取的关键是合并近似色。这就要求能够用数值大小来表示两个颜色的差异,即色差。一般是在色空间中用两个颜色的距离来表示色差的,但要求色空间是匀色空间,即色空间中颜色距离的大小与视觉色彩差异成正比。标准的匀色空间有 CIE 1976 L*a*b* 表色系统和 CIE 1976 L*u*v* 系统。由于 L*u*v* 系统的色度图保留了马蹄形的光谱轨迹,比较适合对光源色加以研究,故一般采用 L*u*v* 表色系统计算色差。

2.2 色差公式的建立

VGA 显示器采用的 RGB 色空间是显示器的设备色空间,与 CIE 1931-rgb 真实三原色系统所采用的三原色不同。为了应用 L*u*v* 的色差公式,必须测出设备三原色在 CIE-rgb 系统的坐标,然后通过线性变换,将显示器的 RGB 色转换到 L*u*v* 系统中去,进行色差计算。

当然上述工作较复杂,为简单起见,仍以设备空间中的两色的距离表示色差。由于红绿蓝色的人眼宽容度不同,应加权进行修改。

设 X、X' 两种颜色的红绿蓝分量分别为 RGB 和 R'G'B',则其色差为:

$$d(x, x') = W_1(R-R')^2 + W_2(G-G')^2 + W_3(B-B')^2$$

其中 W_1 、 W_2 、 W_3 为相应的加权系数。综合考虑经验数据通常取 $W_1=1$, $W_2=2$, $W_3=1/4$ 。

2.3 色差公式的改进

若一幅图象的颜色直方图比较均匀,则采用合并色差较小的颜色,效果不错。但当直方图不均匀或有噪声点时,会违背前面介绍的处理原则:主色调颜色类应保留较多的颜色数。因主色调颜色类之间的差异很小,根据色差公式计算结果,往往被合并,这样体现不出图象的层次色调。同时也使噪声点占用了大量的颜色,这也违背了“充分利用调色板”这一原则。

我们把仅使用“色差”这单一原则合并颜色,使相似色成为一群的方法,称作“群间相似度分群法”。

改进的方法是考虑各群中的像素数。

引用误差平方和准则函数。设有群 X_i ,其像素数为 n_i , m_i 为群的均值向量, C 为总群数,即 $m_i = \frac{1}{n_i}$

$$\sum_{x \in x_i} x.$$

$$\text{定义误差平方和准则函数 } J_e = \sum_{i=1}^C \sum_{x \in x_i} ||x - m_i||^2$$

实际 J_e 为方差的平方。显然,对于一种集群划分,使得 J_e 有最小值,可以认为颜色再现误差最小。如何得到最佳划分?可以从初始划分开始,每次合并两群,使得 J_e 增加值最小,这与最小生成树的思想类似。下面研究合并两群时, J_e 的增加量。

设将合并的群为 x_i 和 x_j ,其像素点数与均值向量分别为 n_i, n_j 和 m_i, m_j 。则 x_i, x_j 两群的准则函数为:

$$J_i = \sum_{x \in x_i} ||x - m_i||^2$$

$$J_j = \sum_{x \in x_j} ||x - m_j||^2$$

合并为一个群后,均值向量 m 和准则函数 J 及增量 ΔJ 计算如下:

$$m = \frac{n_i m_i + n_j m_j}{n_i + n_j} = m_j + \frac{n_i(m_i - m_j)}{n_i + n_j} = m_i + \frac{n_j(m_j - m_i)}{n_i + n_j}$$

$$J = \sum_{x \in x_i + x_j} ||x - m||^2 = \sum_{x \in x_i} ||x - m||^2 + \sum_{x \in x_j} ||x - m||^2$$

$$= J_i + J_j + \frac{n_i n_j}{n_i + n_j} ||m_i - m_j||^2$$

$$\text{则 } \Delta J = J - (J_i + J_j) = \frac{n_i n_j}{n_i + n_j} ||m_i - m_j||^2 = \frac{n_i n_j}{n_i + n_j} d(m_i, m_j)$$

因而将合并集群的指标改为 ΔJ ,此方法称之为“最小误差准则分群”。

从下面的图 1 的示意性直方图可以看出最小误差准则分群可以保留主色调。

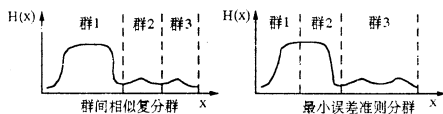


图 1 两种分群法比较

2.4 集群法流程图

设 C 是最终所需颜色数, C_1 是集群数, n 是最初颜色数。其实现流程如图 2 所示。

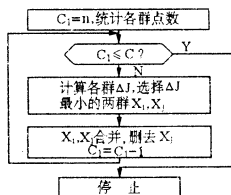


图 2 集群法流程图

3 算法与实现

3.1 实现难点

若开始有几种颜色, 则为找出 ΔJ 最小的两群, 需比较 $\frac{n(n-1)}{2}$ 个 ΔJ 值, 当 $n=2^{24}$ 时则 $\frac{n(n-1)}{2} \approx 2^{47}$, 计算量太大, 况且要纪录各色归入第几群, 又要 2^{24} 个存储空间, 很难行得通。为此采用了一些较新的想法。

3.2 色族划分及数据结构

原始算法要比较集群中两两群差 (即 ΔJ), 以求出最小值, 这是最耗时耗空间的一步。设想若能将各色排列成线性顺序, 使得任一色与之群差最小的两色相邻, 则可只计算比较相邻色群群差。即对于队列

$C_0 C_1 \dots C_{i-1} C_i C_{i+1} \dots C_n$ 中任一色群 C_i , 恒有关系 $\Delta J_{C_i C_{i+1}} \leq \Delta J_{C_i C_{i+R}} (R > 1)$

$$\Delta J_{C_{i-1} C_i} \leq \Delta J_{C_{i-R} C_i} (R > 1)$$

图象中的颜色是散乱在色空间中的点, 怎样排序呢?

如图 3 所示, 将平面划分为若干放射状区域, 则每一区域角度愈小, 区域内各点愈近于在同方向的轴射线上。只要划分足够细, 即可做到“按区域排列”, 色空间情况类似。

因此将色空间划分为若干色族, 各族中颜色具相似色度, 即成功地将全部颜色线性排序。

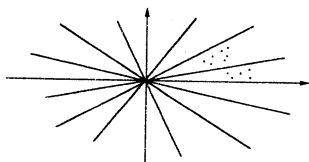


图 3 色族划分示意图

实际采用 12 个色族, 各族划分标准如表 1。

采用什么样数据结构来记录各色群呢? 考虑到由于颜色需经常插入、合并, 且数目较多, 采取分块链接方式。具体如下:

```

struct RGB{
    unsigned char r,g,b; /* 颜色向量 */
    unsigned long n; /* 群中像素数 */
}
  
```

```

struct COLORFAMILY
{
    struct RGB group[40];
    int length;
    struct COLORFAMILY * next;
};
  
```

为了减少移动元素个数, 在每块 40 个颜色中, 留最后 5 个作为空余量, 以避免紧密排列, 而使每插入一个色都移动大量数据。

表一 色族划分标准

区号	名称	条件 1	条件 2	条件 3
0	红	$R = \max$	$R \geq 2G$	$R \geq 2B$
1	红品	$R = \max$	$R < 2B$	$B > 2G$
2	红黄	$R = \max$	$R < 2G$	$G > 2B$
3	绿	$G = \max$	$G \geq 2R$	$G \geq 2B$
4	绿黄	$G = \max$	$G < 2R$	$R > 2B$
5	绿青	$G = \max$	$G < 2B$	$B > 2R$
6	蓝	$B = \max$	$B \geq 2R$	$B \geq 2G$
7	蓝青	$B = \max$	$B < 2G$	$G > 2R$
8	蓝品	$B = \max$	$B < 2R$	$R > 2G$
9	红灰	$R = \max$	不在 0, 1, 2	
10	绿灰	$G = \max$	不在 3, 4, 5	
11	蓝灰	$B = \max$	不在 6, 7, 8	

3.3 排序法及数据结构

每次删除一个群之后, 有两个群差将改变, 从而破坏已排好序的表。即对于线性排列的色集群 $C_0 C_1 \dots C_{i-2} C_{i-1} C_i C_{i+1} \dots C_n$

若合并群 $C_{i-1} C_i$ (删去 C_i), 则由于均值向量和集群中像素数发生改变使得 $\Delta J_{C_{i-2} C_{i-1}}$ 和 $\Delta J_{C_i C_{i+1}}$ 发生改变。若这两个改变破坏了整个群差排序表, 则需重新排序, 代价昂贵。

鉴于上述情况和色群数很多情况, 选用堆排序。堆实际上是个完全二叉树。所需时间是 $O(n \log_2 n)$; 排序原地进行, 只需一个用于交换的结点。

堆用一个数组存储。其中一个结点定义如下:

```

struct NODE
{
    double dj; /* 群差 */
    struct COLORFAMILY * p; /* 群差第一个群所在块指针 */
    int site; /* 第一个群在块中位置 */
    unsigned long pre, succ; /* 前后群差在堆中位置 */
};
  
```

$pre, succ$ 两值指出当这一群差的两群合并后, 群差值应发生变化的两结点。

3.4 抽样统计策略

一幅图象很大时, 一般都是采用分块扫描, 其覆盖面不广, 可以使每次抽样都从整幅图象范围内找, 比如每隔十行抽取一行进行统计, 这样的方法我们称为“分次不分块”。

若情况不是很特殊, 一次扫描即能反映图象色彩情况。但是为了不漏掉任一点, 就必须全部扫描。因此可以根据具体情况选定扫描次数。

参考文献

- 1 杨光红 “Windows 色彩抖动原理”《中国计算机用户》1994.1
- 2 胡成发 《印刷色彩与色度学》印刷工业出版社. 1993

计算机信息管理系统(MIS)在企业目标管理中的作用

傅晓强 蒋雅辉 童长祚

(江西省交通研究所 南昌 330003)

摘 要 本文介绍了计算机信息管理系统(MIS)在信息管理自动化、标准化、规范化、系统化、科学化中的巨大作用,例举了江南冶炼厂使用该系统的实例进一步生动地阐明了主题。

关键词 (MIS)目标管理 信息管理

企业现代化管理,包括现代化管理的方法和现代化管理手段。现代化管理的方法——目标管理,现代化的管理手段——应用电子计算机技术,对企业信息进行管理,以达到信息管理自动化、标准化、规范化、系统化、科学化,即通常所说的计算机信息管理系统(MIS)。

目标,是人们通过自己的各种活动所需达到预期效果。目标管理是系统、综合的现代化管理技术,是企业推行系统工程的重要手段。目标管理,是五十年代美国出现的一种新的管理制度,当时美国产业面临激烈竞争的局面,急需强化企业素质,提高工作效率,增强竞争能力,而美国当时一般企业机构臃肿,管理组织僵化,工作效率不高,严重阻碍了企业的发展。因此,急需一种新的更有活力的管理制度来取而代之,目标管理就是在这种情况下应运而生的。

目标管理,在美国广泛地应用于工业、金融、公用事业等行业的公司,取得了极大的成果。日本是世界上最早引进目标管理的国家,日本引进后,与本国的以及原先引进的管理制度结合起来,取长补短,充实提高,加以日本化,形成了具有日本特色的,效率更高的管理制度,并获得了相当的经济效益。

传统管理的最大弊端是信息观念差,因而造成管理失控,推行目标管理的目的是对企业的全部经营活动按总目标进行动态控制,这就需要及时了解企业经营活动的情况,如材料消耗情况、资金状况、产品情况、市场情况、运输情况以及每个部门和个人完成目标的情况都需要反映出来,加以整理、归纳、传递并进行分析判断,及时做出纠正偏差的决策,指导企业的经营活动协调地进行。

加强信息管理,落实目标控制,首先需要加强信息源的整顿。健全原始记录,按照标准化、规范化、系统化的要求进行数据和数据统计管理,搞好系统的报表传递程序,加强电子计算机的开发应用。为了达到信息反馈系统化、网络化,还需加强生产经营信息反馈中心的建设。可采用电子计算机,多媒体技术,闭路电视,信息指令卡、统计图表等现代化手段和方法,来实现目标控制,这不仅可以大大提高指挥效率,而且为正确决策提

供了信息。

我国江南某冶炼厂,是一个曾为国家经济建设作了较大贡献的企业,建厂40年来,一直未得以全面的技术改造和综合治理,因此形成生产工艺落后,设备老化,环境污染严重的局面。近几年来,该企业自筹资金,并吸收外资,与日本一公司和香港一公司合作,采用先进的工艺技术,对本企业进行全面的技术改造,以期达到更新生产工艺,提高生产效率,降低能源消耗和治理环境污染的目的,将企业建成为现代化、无公害、效率好的具有世界第一流管理水平和技术水平的冶炼厂。

随着我国现代化建设的发展和科学技术的进步,以及改革开放政策的实施,国内外影响企业的经济、技术、市场等各种因素,将越来越复杂。在这种情况下,这个具有国内一流技术和水平的企业,要在强手如林的竞争者面前立于不败之地,就必须建立计算机网络系统——信息管理系统(MIS),以实现信息收集、处理、存储和传输,建立综合数据库,建立现代化,高效率的管理体制,达到管理现代化,决策科学化的目的。

这个企业的信息系统(MIS),它的主要任务是采用先进的计算网络技术(NOVELL 网)、先进的数据库管理系统(FOXPRO)和先进的系统工程技术,对本企业的生产、经营及生产经营相关的信息群进行科学管理。

为完成企业的决策、策划、财务、经贸、基建、材料设备、生产技术、行政和人事等方面的信息处理和信息传递自动化,从而提高各职能部门的办公效率和业务管理能力。在这个基础上,利用系统科学和行为科学中的理论,为企业员工提供各种分析情况的方法和手段,并辅助企业领导合理地做出决策,为最终形成一个信息流动快、工作效率高、处理事务能力强、决策正确果断、步调一致的管理体系奠定基础。

企业总的目标,包括总的方针,具体目标和措施目标。企业总方针,即实现企业经营目的,经营策略和经营思想而采取的行动纲领。具体目标,即定量目标,就是一定时间内,一切生产经营活动追求的理想期望值、技术经济指标和员工受益某欲达到的目标,为实现企业目标,还要制订部门目标,所有目标都要在总目标的指导下,综合平衡,衔接配套,形成目标网络,全面落实。措施目标,即保证措施,就是须达到具体目标所采

取的主要措施,反映整个活动过程的相应标准,受具体目标制约。

企业的总目标是:引进现代科技,采用先进工艺和现代化管理模式,尽快建成一个现代化、高效率的企业。各部门按各自的情况,制定部门目标,各部门的目标如下:

行政事务部:

1、做好调研和信息反馈工作,为领导决策,充分发挥参谋、助手作用。

2、做好接待工作,塑造企业形象。

3、做好办公设施的规划、管理,达到现代化水平。

4、做好档案管理工作,提高文档、科档、财档利用率。

5、做好保安工作,确保一个良好的工作环境。

策划部:

1、根据企业总的目标,编制切实可行的综合计划。

2、完成企业管理模式的确定和定员定编。

3、完成生产准备中的有关任务。

4、协助开发方,完成开发和研制计算机信息管理系统(MIS),并使正常运行。

财务部:

1、完成出资计划的催缴和资金的筹措,以确保用款计划的完成。

2、加强核算,降低成本费用和各项费用支出。

3、完成各类财务报表的编制。

材料设备部:

1、确保工程和生产所需国内材料、设备的采购、供应、存储工作。

2、做好物质管理,为今后生产打好基础。

经贸部:

1、确保翻译和外事工作不影响企业的基本建设和生产经营活动。

2、确保引进设备和材料工作不影响企业的基本建设和生产经营活动。

工程部:

1、认真履行部门职责,搞好勘察、设计与施工的协调,使工程建设严格按规范进行。

2、实行项目管理与专业管理相结合,履行建设合同中规定的权力、义务,使工程实际进度达到要求。

3、搞好工程投资管理,以工程施工图概、预算为依据,结合工程建设实际及国家政策性调整,严格进行预、决算审核,保证建设资金的有效、节约运用。

4、搞好部门管理,信用劳动合同,遵守规章制度,维护企业整体利益。

生产技术部:

1、根据企业确定的目标,做好员工培训工作,确保基建、生产需要。

2、参与设计过程中的工艺流程确定和决策。

3、参与基建期间设备的安装、验收、调试、单体试车、联动试车工作。

4、参与国内、外大型设备订货的技术交流、讨论和资料整理。

5、制定本部门岗位操作规程及技术规范。

以上是这个企业的总目标和部门目标,整个企业从总经理到普通员工,都以高昂的斗志、满腔的热情为之而奋斗,为了更有效的完成目标任务,就必须采用电子计算机技术参与企业管理,这就是企业 MIS。

该企业 MIS 包括二十二个子系统,其中有:

• 总(副总)经理办公事务管理子系统

• 决策支持子系统

• 经济预测子系统

• 综合统计子系统

• 计划管理子系统

• 综合数据管理子系统

• 基建管理子系统

• 经贸管理子系统

• 办公室事务管理子系统

• 文档管理子系统

• 账务处理子系统

• 财务报表子系统

• 工资管理子系统

• 固定资产管理子系统

• 人事信息管理子系统

• 物料平衡管理子系统

• 生产指令管理子系统

• 生产报表管理子系统

• 技术经济指标管理子系统

• 产品销售管理子系统

• 客户档案管理子系统

• 仓库管理子系统

所有的子系统,都是为企业的总目标。

总之,MIS 是社会生产力发展和科技进步的产物,为了适应激烈的市场竞争,在公司管理中逐步形成一种生产计划与控制为主导的管理模式,管理思想和方法。在电子计算机的帮助下,对企业生产诸要素进行优化组合和合理配置,使生产和经营活动过程中的人流、资金流,信息流处于最佳状态,以达到投入产业最优化,这就是企业 MIS 的作用所在。

完善的企业 MIS 运转后,能检查和督促企业的目标管理实施状况,帮助企业完成目标管理中的各项任务。一般说来,可以保证按期交货,增加产品市场竞争能力,降低库存积压,加速资金周转,缩短生产周期,提高设备利用率,降低产品成本,提高企业管理水平,并可获得显著的经济效益和社会效益。

在 Internet 中航行——Netscape Navigator

吴保平

(南京大学国际商学院经济信息中心 210093)

摘 要 本文向我们推荐了 Internet 的有效浏览工具: Netscape Navigator, 只有借助这一有效工具才能迅速准确地获得在 Internet 上的无穷资源, 并分六个方面对 Navigator 的特点、功能及其使用方法作了简单介绍。

关键词 Internet Navigator4.0 WWW 服务器 网络浏览器

0 引言

美国付总统戈尔提出的美国信息高速公路法案及美国政府制定的国家信息基础设施引发了全美乃至全球的 Internet 热。越来越多的机器被连到 Internet 上; 越来越多的人访问着 Internet。Internet 把人们之间的距离拉近了: 人们能从 Internet 上迅速得到信息或及时将新的信息向 Internet 用户发布。Internet 的使用推动全球科技, 生产等的快速发展, 加速了全球的合作与进步。

Internet 上的资源很丰富。要迅速、准确获得我们所需的资源, 就需借助一个有效的工具。这个工具就是继 Mosaic 之后出现的 Internet 浏览工具——Netscape Navigator。该工具自推出之日起受到广大 Internet 用户的欢迎, 并获得 1996 年计算机市场最佳产品的冠军。目前, 该产品还在不断完善: 预计 1996 年 6 月正式启用 Navigator3.0, 并于 1996 年底推出 Navigator4.0。

本文将分六个方面对 Navigator 特点、功能及使用方法作一简单介绍。

1 Navigator 特点

Netscape Navigator 是一个强有力的商品化的网络浏览器。它支持用户使用鼠标选项, 并可在 14.4kb/s 的调制解调器以及较高频宽线上平滑运行。它具有良好的图形交互界面并可在 Microsoft Windows, X Windows 以及 Macintosh 操作环境中使用。它融合了安全插口层(Secure Sockets Layer)协议, 使用户可安全地进行在线出版, 金融服务以及在线购物。另外, 它还具有动态更新文档的特点。

Navigator 不仅可显示文本文件和超文本文件而且还支持多媒体。Navigator 把针对 Internet 服务的多种访问功能集成到一个统一、无缝的浏览系统中: 它支持电子邮箱(E-mail)、新闻组讨论(Netnews)、远程文件传输(FTP)、远程登录(Telnet)、信息浏览服务(Gopher)、自动内容搜索(WAIS)、自动标题搜索

(Archie)等服务。它还可保存用户访问踪迹, 为用户浏览文件设置书签以及允许用户更改字体、字号和背景等。

Navigator 是 World Wide Web 客户程序。使用 Navigator 浏览文件, 用户不必知道文件存在何处, 文件所在的机器使用何种操作系统。Navigator 界面友好, 能及时提供反馈信息并具有易学、易用等特点。

2 WWW 上的页

Netscape Navigator 所浏览的文件以页形式存放在 WWW 服务器上。每页都有唯一的一个统一资源地址标识(URL: Uniform Resource Location), 以便用户检索。

URL 地址格式为: <协议>://<主机名>/<路径名>/<文件名>。<协议>表示文件服务器所使用的通讯协议, 例如: http 是超文本传输协议。<主机名>是存放该文件的主机的地址。<路径名>是该文件在主机上的目录路径。<文件名>是该文件的名称。例如 http://www.whitehouse.gov/lib/journal 就是一个典型的 URL 地址。

Netscape 页是用超文本语言(HTTL)编写的, 集文字和菜单(又称为链)为一体。Netscape 页中菜单项(链)是高亮度显示的带下划线的文字, 带有有色框的图像或图标。URL 内嵌在链中, 所以链将一页和另一页相连。

Netscape 页自上向下由以下几个部分组成: 标题条、下拉菜单、工具条按钮、地址域、目录按钮、页正文、状态域以及传输进程条。标题条显示当前页的标题, 下拉菜单提供诸如文件保存、打印、属性设置等操作。工具条按钮一般可用来改变当前 Netscape 页的属性。地址域显示当前页的 URL 地址。目录按钮则提供用于浏览 Internet 的工具。用户可在 Option 菜单中分别将工具条、目录按钮、地址域设置为不显示, 从而扩大正文显示面积。状态域显示当前操作的反馈信息。传输进程条显示页传输的进度。

3 在 Internet 中漫游

当用户启动 Netscape Navigator 时, 屏幕显示某

个首页(home page)(首页可由用户设置,缺省为 Netscape 首页)。

用户可浏览当前页,或单击某个链将所连页显示在屏幕上浏览。用户在浏览过程中,通过单击链,可轻松地从一个机器访问到另一台机器,而无需知道机器所处位置,文件所处位置,从而实现在 Internet 海洋中透明,无缝地漫游。配备多媒体电脑的用户不仅可看到文字、图像或动画而且还可听到声音。

用户浏览的过程以页标题表的形式保存在历史表(History)中。单击其中一项,则可将对应页显示在屏幕上。Navigator 还提供了向后(Back)和向前(Forward)功能;单击 Back 按钮,则可将当前页的前页显示在屏幕上;单击 Forward 按钮,则可将当前页的后页显示在屏幕上。

Navigator 支持用户为经常访问的页或当前来不及访问的页设置书签,以免用户下次访问时需经过漫长的搜索过程才找到所感兴趣的页。页书签的使用提供了永久、快速检索页的手段。正因如此,Navigator 提供了多种设置书签文件的方式:用户在浏览过程中,可执行 Bookmarks/Add Bookmarks 为当前页设定书签,也可将别人的书签输入到自己的书签文件中或将自己的书签输出给别人。通过执行 Windows/Bookmarks,用户可维护书签文件。

Navigator 还提供了用户直接键入 URL 地址以浏览对应页的方法。用户可在 Location 域或 Open 对话框(执行 File/Open Location 所得)内的 Open Location 域内键入 URL,从而将对应页显示在屏幕上。

用户在传输下页的过程中,如因等待时间过长或其它原因想终止传输,可单击 STOP 按钮。

4 Navigator 的搜索机制

用户可通过输入文件的 URL 地址找到该文件;用户也可通过漫游 Internet 得到自己感兴趣的文件。前者要求用户必须知道文件的 URL 地址;后者未必能使用户顺利、迅速得到用户所需的文件。针对上述情况,Navigator 提供了许多搜索机制,允许用户根据主题词、页标题等搜索相应的页、新闻或文档。

单击 Netsearch 按钮,用户便可在屏幕上看到 Navigator 提供的 Netsearch 目录表: A2Z、ACCU-FIND、AIRS II、ALTA VISTA、AMAZING ENVIRONMENTAL ORGANIZATION WEBDIRECTOR!、ARGUS/UNIVERSITY OF MICHIGAN CLEARINGHOUSE、C | NET ' S SHAREWARE.COM、DEJANEWS、THE ELECTRIC LIBRARY、EXCITE、GAMELAN、GNN SELECT、IBM INFOMAKET、INFOSEEK GUIDE、LYCOS、

MAGELLAN、NERD WORLD MEDIA INTERNET SUBJECT INDEX、100HOT WEBSITES、OPEN TEXT INDEX、POINT、THE SOFTWARE SHARING RESOURCE LIBRARY、STARTING POINT、WEBCRAWLER、W3 SERVERS、WHAT ' S NEW、WHOWHERE、YAHOO!。

其中,ACCUFIND 用来搜索信息库、新闻、书以及 Internet。通过使用 AIRSII,用户可输入搜索词对文件内容进行模糊搜索。C|NET'S SHAREWARE.COM 可让用户轻而易举找到 Internet 上的软件,以便用户浏览及下装。INFOSEEK GUIDE 可被用来按关键字对文件内容进行搜索,以找到最相关的匹配、相关的话题以及来自于流行杂志的新闻和观点。它常被用来寻找 E-mail 地址、股市行情及公司文件等。WHOWHERE 可让用户快速搜索网上的人或机构从而得到相应的 E-mail 地址或其首页地址。

5 Navigator 的电子邮件/新闻组

用户可用 Navigator 发送邮件、新闻稿。单击当前页右下方的信封图标或 File/New Mail Message,用户可在随即出现的 Netscape Mail 或 Netscape Message Composition 对话框内,填入接收方的邮件地址,邮件的主题词以及正文并将其发送。用户还可让邮件附带某一文件或当前页的内容发送。

通过执行 Windows/Netscape News,用户在随即出现的 Netscape News 对话框内不仅可发送、收看邮件而且可参与新闻组讨论;阅读别人的观点、看法且发表自己的观点和看法。每个新闻组都有一个能反映其话题的名字,例如 rec.garden.orchids 是一个关于兰花栽培、杂交及一般管理的讨论。

6 Navigator 的属性设置

为了让用户以自己喜爱的方式访问 Internet,Navigator 允许用户设置或更改当前操作的属性:通过执行 Option/Preference,用户可选择字体、字的大小、页的背景色、链被激活前后的颜色;用户还可设置 Navigator 启动时的首页。

用户可执行 Option/Show Toolbar、Option/Show Location、Option/Show Directory Button 让 Toolbar、Location 或 Directory Button 显示或不显示在屏幕上。如不显示,则显然扩大了当前页的显示区域。

Netscape 页由文字、图像或图标组成。传输图像要花很长时间。用户可执行 Option/Auto Load Images 关闭自动图像装入,这样在页传输过程中图像均被图标取代,文件传输的速度大大提高。不过用户可手动恢复图标所在处的图像。

一种程序运行失常自恢复控制的方法及其实现

罗 杰

(江西师范大学计算机系 南昌市 330027)

摘 要 本文介绍一种程序运行失常自恢复控制的方法及其在 MCS-51 单片机中的实现。该方法对程序运行失常的识别及时准确,自恢复控制的实时性、针对性强,电路简单可靠,能有效地保证程序在正确的时、空范围内运行。

关键词 程序跑飞 自恢复 越界 超时 8031 单片机

1 引言

在工业测控中应用的微机,由于现场环境的恶劣,不可避免地要受到各种强干扰源的干扰,常常使微机系统运行失常,导致测控失灵甚至造成重大事故。因此,抗干扰的设计成为计算机应用系统研制中影响系统可靠性的重要内容。除了采取硬件措施尽量避免干扰的窜入外,一旦干扰已造成程序失常,如何使程序失常造成的影响减到最小,并使程序迅速自动恢复正常,这是人们关心的问题。本文阐述了一种程序运行失常自恢复控制的方法及其在 MCS-51 单片机中的实现,该方法对程序运行失常的识别力强,实时性好,能自动及时地使程序恢复正常,并使运行失常造成的影响减到最小,有效地保证了程序在正确的时、空范围内运行,电路简单可靠,不影响系统的正常工作。

2 程序运行失常自恢复控制的方法

微机系统程序运行失常跑飞,一般是由于遭受强干扰后程序计数器 PC 的值偏离正常值所造成。如,进入非程序区,引起程序混乱及进入无意义的死循环,使系统严重失控。为了使程序迅速及时自动恢复并使影响尽可能地减小,总的原则是尽可能在程序跑飞处恢复正常运行,当无法这样恢复时再让系统自动复位恢复运行。为此,应从两方面采取措施:一是在空间上加“围栏”,以程序区地址空间为限,对 PC 飞离正常程序区进行及时快速识别捕捉,并使程序在跑飞处恢复。二是在时间上加“围栏”,以程序运行的正常时间为基准设置程序运行时限,一旦超时即可判定异常,立即使系统自动复位。这样就保证了 PC 值在正常程序区内时系统出现失常死机也能及时自动恢复正常。这两方面的措施实施时要求自身抗干扰能力强,并且不影响系统的正常工作,因而应以不受程序影响的硬件电路为主。具体做法如下所述。

所谓空间上加“围栏”就是按程序区所占实际地址空间范围通过硬件电路限制 PC 的值进入非程序区,一旦 PC 的值越界,立即由电路引发异常中断,由中断

服务程序完成程序从跑飞处恢复。原理如图 1 所示。

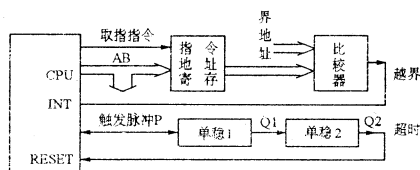


图 1 自恢复控制原理图

CPU 一般都有取指令指示控制信号输出(如 Z80 的 \overline{M} , MCS-51 单片机的 \overline{PSEN} 等),在 CPU 将当前指令地址 PC 值送上地址总线期间该控制线上输出有效脉冲,在此利用它将 PC 值打入指令地址寄存器,随即由比较器将此地址与预置的界地址(程序区末地址)比较,若越界(大于末地址)则立即使 CPU 中断,进行恢复处理。这样处理达到了较高的实时性,能及时制止程序跑飞。

若由于干扰造成程序在程序区内出现混乱,最后进入死循环而 PC 的值又没越界,对这种情况可由时间“围栏”使系统恢复正常,原理如图 1 所示。时间“围栏”由两个单稳电路级联而成,单稳 1 的输出 Q1 作为单稳 2 的触发脉冲,单稳 2 的输出 Q2 作为系统复位信号输入 CPU。由于测控系统中的软件一般都是循环式的,若在大循环的结尾安排使单稳 1 的触发脉冲 P 线上输出一个脉冲的指令,这样,在程序正常运行情况下,CPU 将周期性地产生触发脉冲 P,设其周期为 T。设单稳 1 和单稳 2 的暂态时间分别为 t_1 和 t_2 , t_1 、 t_2 的值可由单稳 1 和单稳 2 电路中 R、C 的选取而定。若将 t_1 设置为略大于 T,则单稳 1 将在暂态中被反复再触发,从而使 Q1 保持暂态(设为高电平),这样单稳 2 输出 Q2 将保持稳态(设为低电平),系统不发生复位(设 RESET 要求高电平复位)。若由于干扰使程序发生混乱进入死循环,则 P 将停止脉冲输出,从而使 Q1 输出变低, Q2 将被触发产生一个正脉冲,其宽度 t_2 若被设置为复位信号要求的宽度,则系统即被复位恢复。整个过程 P、Q1、Q2 的时序关系如图 2 所示。

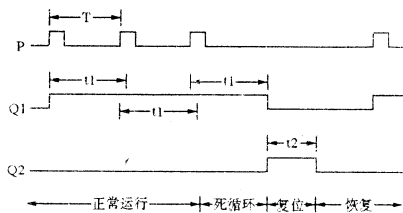


图 2 自动复位控制时序

由于采用的是单稳电路定时,因而“围栏”本身的抗干扰能力很强,可靠性高。

被空间“围栏”识别捕捉到的程序跑飞引起 CPU 中断后,如何使程序在跑飞处恢复呢?我们可以将程序按关联性强弱划分为 i 段,并设置一个程序运行轨迹记录器,在运行某段时先将该段的段号置入记录器再运行该段。这样,若程序跑飞使 CPU 中断后,可在中断服务程序中通过查询记录器找到跑飞的段,然后再由段号查地址表使程序恢复到该段起始处继续正常运行。

上述时空“围栏”有效地保证了系统的软件在正确的时空范围内运行。

3 在 MCS-51 单片机中的实现

以单片机系统为例,上述方法在 MCS-51 单片机构成的系统中的具体实现电路如图 3 所示。

图 3 中,指令地址寄存器采用了二片 8D 触发器 74LS374,比较器采用了二片 8 位数字比较器 74LS683,单稳电路采用了一片双可再触发单稳态多谐振荡器 74LS123。74LS683 和 74LS123 的逻辑功能分别如表一、表二所示。

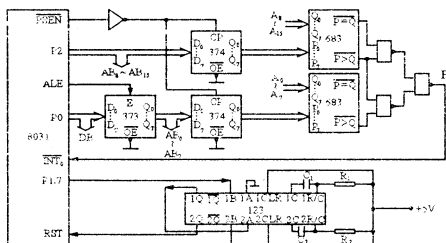


图 3 在 8031 系统中的实现电路

16 位的指令地址在 $\overline{\text{PSEN}}$ 信号下跳沿的作用下送入两片 8D 触发器 74LS374,然后与用开关预置的界地址 $A_0 \sim A_{15}$ 经两片 74LS683 进行比较,若大于界地址则 F 端变低,向 8031 $\overline{\text{INT}_0}$ 发中断请求,进入恢复处理。从 8031 时序看,由于在每个机器周期的 S5P2, $\overline{\text{INT}_0}$ 的电平会被锁到内部保持寄存器中,故在此无须设置中断请求触发器,响应中断进入中断服务程序,由于 PC 进入正常程序区,故 F 端立即回到高电平,从而自动撤除了 $\overline{\text{INT}_0}$ 请求。P1.7、1Q、2Q 信号分别如图 2 中的 P、Q1、Q2 波形,一旦程序运行超时异常,则单稳电路通过 2Q 输出复位信号使 8031 复位,从而使系统

恢复正常运行。

表一

输入	输出	
P、Q	$\overline{P=Q}$	$\overline{P>Q}$
$P=Q$	L	H
$P>Q$	H	L
$P<Q$	H	H

表二

输入	输出			
CLR	A	B	Q	\overline{Q}
L	\times	\times	L	H
H	L	\uparrow	\downarrow	\uparrow
H	\downarrow	H	\downarrow	\uparrow
\uparrow	L	H	\downarrow	\uparrow

由于 MOV_C 查表指令执行时,也由 $\overline{\text{PSEN}}$ 信号控制读取程序存储器 ROM 中的数据,此时图 3 中的 74LS374 寄存的将是数据地址,而数据区地址允许超出程序区界地址,这样将造成执行 MOV_C 指令正常读取数据时也发生 $\overline{\text{INT}_0}$ 中断的错误。避免这个错误的方法有二种,一是将界地址放宽设置到程序区后的数据区之尾。二是将程序存储器 ROM 中的数据安置到外部数据存储器的 ROM 中,使程序和数据处于不同访问控制方式的存储空间。第一种方法降低了识别跑飞的能力,使得由于跑飞,PC 值进入数据区时无法及时拦截恢复,从而对系统造成较大影响。第二种方法配合上述的“空间围栏”可以使界地址紧接在程序区末,彻底避免了 PC 值闯入非程序区及由此造成的混乱。

$\overline{\text{INT}_0}$ 为最高级可屏蔽中断请求,保证程序跑飞的及时恢复。设主程序运行轨迹记录器为 SN 单元,它记录了 CPU 当前运行的程序段号。主程序被分为 i 段,段号编为连续的偶数 $0, 2, 4, \dots, 2(i-1)$ 。在主程序中每进入一段的运行,先将该段的编号置入 SN 单元。这样,当程序跑飞,则 SN 就记录了跑飞所在的段,通过 SN 记录的段号查段地址表可得到跑飞段的起始地址,将它置入 PC 即可使程序从该段恢复。设段地址表首地址为 TAB,此表内顺序连续存放了主程序中 i 段程序的段起始地址,每个地址占 2 个字节,这样, TAB 加上 SN 记录的某段号值即指向该段起始地址所在单元。TAB 表放在外部数据存储器的 ROM 中。

恢复跑飞程序的 $\overline{\text{INT}_0}$ 中断服务程序如下:

```
ORG 0003H
```

```
POP A;弹出无用的断点地址
```

```
POP A
```

```
MOV DPTR, #SN
```

```
MOVX A, @DPTR;把跑飞段的段号值送入 A
```

```
MOV B, A
```

```
MOV DPTR, #TAB;把段地址表首地址送入 DPTR
```

```
MOV A, DPL;查表找出跑飞段的起始地址低 8 位并进
```

栈

```
ADD A, B;
```

```
MOV DPL, A;
```

```
MOV A, DPH;
```

```
ADDC A, #00H;
```

```
MOV DPH, A;
```

```
(下转 86 页)
```


硬盘驱动器引导型软故障解析

王维舟

(西安理工大学学报编辑部 西安 710048)

摘要 硬盘驱动器是计算机系统中故障率最高,对资源造成损失最大的部件为了确保微计算机系统速度快安全可靠地运行,本文主要以硬盘引导型软故障发生的原因作以分析,并提出相应的修复方法。

关键词 引导型软故障 引导型病毒 主引导扇区 DOS 引导扇区

硬盘驱动器在微机系统中作为主机的外部存储设备,具有存储容量大,运行速度快,安全可靠的特点。但由于众多的原因,硬盘驱动器也是计算机系统中故障率最高,对资源(数据、文件、成果)损失最大的部件。硬盘驱动器的故障不外乎两种形式,第一是硬盘的物理损坏,如光电控制系统损坏;磁盘划伤。第二是硬盘软损坏,如软件故障;误操作;病毒感染等。后者多发生在硬盘的引导扇区。

在使用中,硬盘软故障的发生率大大高于硬件故障的发生率。所以,本文主要以硬盘引导型软故障发生的原因作以下分析,并提出相应的修复办法。

1 硬盘引导型故障的表现形式,屏幕提示及故障分类

归纳引导型软故障,大致可分为三类。

第一,硬盘的主引导扇区被破坏,其屏幕提示分别是:Disk boot failure;Basik.... OK;Error liading table;Miss operation system;等。当由 A 驱动器启动,进入 DOS 操作系统在 A)提示符下键入 C:并回车,屏幕提示:Invalid drive specification,当键入 Format C:对硬盘进行格式化时,屏幕仍提示上述信息,用 Fdisk 文件建立或查看硬盘分区,选择菜单第 4 项 Display partition data 显示分区信息为:No partitions defined

第二,硬盘的 DOS 系统引导记录被破坏。屏幕提示信息分别为:Bad or missing command interpreter;Invalid drive specification;Non-system disk or disk error replace specification 等。当由 A 驱动器启动,进入 DOS 操作系统在 A)提示符下键入 C:回车,可以进入 C)并且文件完整,工作正常。

第三,统配置丢失或配置不当。屏幕提示:Drive not readyerror inset boot in A:Press any key when ready;或 C:drive error。

2 故障分析及排除

1. 主引导记录被破坏

主引导记录被破坏是硬盘故障中较为复杂的一种。引导记录位于硬盘的第一个物理扇区,即 0 面 0 道 1 扇区,其中存放有硬盘的主引导程序;分区信息以及引导合法标志三部分,即第一引导模块。它不依赖任何 DOS 系统,但是,DOS 系统必须依赖它才能实现自举。

主引导程序的长度为 240 的字节,位于偏移量为 0—0EFH 的区域,主要用于硬盘的引导。磁盘分区表的长度为 64 个字节,位于偏移量为 1BEH—1FDH 的区域,用于描述硬盘的分区状况。这 64 个字节分为四个登记项,登记与每一部分有关的信息。偏移量 1FEH—1FFH 的两个字节为 55H 和 AAH 用于分区的合法标志,是引导扇区的有效标志。

在硬盘启动时 ROM 引导程序从硬盘 0 道 0 磁头 1 扇区读进主引导记录,并放入内存的某一区域,检查最后两个字节是否为 55H、AAH,如果是,则从内存 XXXX:XXXX 处执行主引导程序,主引导程序获得控制权后,检查分区中每一个分区记录找到一个活跃的分区,查到该分区引导记录所在扇区的柱面号。磁头号 and 扇区将它读入内存 XXXX:XXXX 处,判断分区记录最后两个字节是否为 55H、AAH 并把控制权交给它,由它引导操作系统。如果发生错误将显示以上信息,然后进入一个死循环。

所以主引导程序、分区信息以及引导合法标志三者任何一项被破坏均导致硬盘引导失败。

主引导扇区示意图

主引导扇区	空	硬盘分配表	55	AA
0	0D9	1BE	1FD	1FE 1FF

修复主引导扇区软故障方法的讨论。

硬盘的主引导记录是由 DOS 系统中 FDISK 文件创建的,所以,最简便的方法是用 FDISK 重写主引导记录,但是将导致文件分配表的丢失以至硬盘上信息全部永久性丢失。所以,除非硬盘内容无关重要,一般应慎用 FDISK.COM、FORMAT.COM、HDFORMAT.EXE、DM 文件以及调用固化 BIOS 程序中的 HARD DISK UTILITY 子程序,而尽可能的用相应的软件方式来修复主引导扇区。

方式一:

对主引导扇区进行编辑修改,使其与同类型机器,同类型硬盘,DOS 版本相同之主引导扇区相同。步骤如下:

(1)借助一台启动正常,且与有故障微机的硬盘分区情况相同的微机,在 A 驱中插入一张格式化后的空盘,键入以下语句序列:

C:\DEBUG

-A100

3F8D:0100 MOV AX,0201;读 1 个扇区

3F8D:0103 MOV BX,1000;置缓冲区为 CS:1000

3F8D:0106 MOV CX,0001;读 0 柱面第一扇区

3F8D:0109 MOV CX,0080;硬盘 0 号磁头

3F8D:010C INT 13; 磁头 I/O 中断

3F8D:010E INT 20; 程序终结

3F8D:0110

-G=100

-W=1000 0 50 1

(2)以 DOS 软盘启动有故障的微机,进入 DEBUG 状态

键入语句序列

A>DEBUG

-L1000 0 50 1

-A100

1136 0100 MOV AX;0301

1136 0103 MOV BX;1000

1136 0106 MOV CX;0001

1136 010C INT 13

1136 010E INT 20

1136 0110

-G=100

至此,硬盘主引导记录修复完毕。

方法二:

用一张完好的带有 CPAV1.2 版的 BOOT-SAFE.EXE 文件的软盘插入一台正常运行硬盘型号。DOS 版相同的微机,将其硬盘分区信息数据备份下来,步骤如下:

A:\BOOTSAFE C:/M/

屏幕提示:

Save partition table to drive A:?

键入“Y”回车,就在软盘上建立了一个 CBOOT 文件。

将备份盘插入有故障的 A 盘中,进行如下操作:

A:\BOOTSAFE C:/R/

屏幕提示:

Are you sure you want to continue?

键入 Y 回车,就将硬盘分区表数据内容写入有故障的硬盘上。

方法三:

硬盘的主引导扇区由于其可写性,所以是计算机病毒攻击的主要目标。这种病毒称为引导型病毒,它主要侵袭主引导扇区,破坏分区信息表,以及 DOS 引导扇区。它主要是用计算机病毒的全部或部分逻辑取代正常的引导记录,而将正常的引导记录隐藏在磁盘的其它存储空间内。由于硬盘的引导区是计算机正常工作的先决条件,所以此类病毒可在运行伊始即获得系统控制权,其传染性较强。

引导型病毒用消毒软件较难彻底消除,可行的方

法是从一台无病毒的同类型微机(硬盘相同,DOS 版本相同,逻辑分区相同)上将其硬盘的主引导扇区拷入有病毒的微机的主引导扇区。对原扇区进行覆盖。对关键部门(如科学研究,金融系统,关键性管理部门等)可将其微机硬盘的主引导扇区用动态调试软件 DEBUG 备份,以备主引导扇区被破坏时恢复。

2. DOS 引导记录被破坏

DOS 引导记录被破坏较主引导被破坏的可能性更大。DOS 引导记录位于硬盘分区的第一个逻辑扇区,即 0 道 2 扇区,也是第二引导模块,即 BOOT 块。只有此模块才能完成 IO.SYS MSDOS.SYS COMMAND.COM (MS DOS 版本),或 IBMBIO.COM IBMDOS.COM COMMAND.COM (IBM DOS 版本)的安装。

修复 DOS 引导记录方法的讨论

方法一:

对已断定是 DOS 引导记录被破坏且排除病毒感染的硬盘,用于其版本相同的 DOS 操作系统从 A 盘启动,用 SYS.COM 将 IBMBIO.COM IBMDOS (IO.SYS MSDOS.SYS)传入 C 盘,并用 COPY 命令将 COMMAND.COM 文件拷入 C 盘。有时用 SYS 命令时,系统将出现 No room for system on destination disk 提示,表明目的磁盘的根目录第一、二个目录项已被别的文件占用,两个系统隐含文件不能被传送。此时可用 pcshell 工具软件进行调整。

方法二:

用 DEBUG 命令直接修复。

A>DEBUG

-L 100 2 0 1

-D 100 2FF

这样读入内存的就是系统的引导记录,其中最后三个字节应该为 00 55 AA。PCSHELL 工具软件调用其程序验证。

方法三:

由于 DOS 系统的开放性,所以其引导记录是病毒入侵。破坏的重点,由于某些引导型病毒不但破坏性大,而且潜伏、转移性强。所以,某些消毒软件只能治表,不能治本。用 PCSHELL 工具软件中 View/Edit (查看/编辑文件)项可以解决这一问题。

进入 PCSHELL 窗口,选择菜单 View/Edit disk 项,出现如下屏幕:

```

DISK EDIT
Absolute Sector 00000000 SYSTEM BOOT DISK ABS SEC
00000000
0000(0000) EB 3C 90 4D 53 44 4F 53 35 2E 30 00 02 04 01
00
0000(0010)
.....
.....
.....(下转 83 页)

```

入网多媒体 PC 机软硬件故障及其维护实例

余先庆

(中科院等离子体所计算机室 合肥 230031)

摘 要 本文主要介绍入网多媒体 PC 机软硬件故障并以实例说明其排除方法。

关键词 入网多媒体 PC 故障

入网多媒体 PC 机是指能够进入国际互连网络如 INTERNET 网的并支持多媒体的 PC 机。随着入网多媒体 PC 机逐步走进家庭,如何解决其常见软硬件故障也将成为人们普遍关心的问题之一。本文下面就将介绍一下入网多媒体 PC 机常见软硬件故障并将以实例说明其排除的方法。

1 入网多媒体 PC 机软硬件故障

入网多媒体 PC 机除了会出现普通 PC 机的软硬件故障外,还常会在多媒体方面和网络方面发生软硬件故障。本文主要就是讨论入网多媒体 PC 机所特有的在多媒体方面和网络方面的软硬件故障:

(1)多媒体方面的故障:其硬故障主要是指 CD-ROM 驱动器、声卡、解压卡等多媒体处理器件、插卡所引起的故障。其软故障主要是指多媒体操作系统、多媒体驱动软件和多媒体应用软件被破坏所引起的故障。

(2)网络方面的故障:其硬故障主要是指调制解调器、电话线路、路由器等网络通讯部件上所引起的故障。其软故障主要是指网络通讯软件、驱动软件或相应的应用软件被破坏所引起的故障。

2 入网多媒体 PC 机软硬件故障分析与排除

(1)多媒体方面故障分析与排除:

其硬件方面的故障对用户而言主要可以采用一级维修即更换部件、插卡的方式维修。其关键在于判断故障点在什么地方,常用的方法有:

a、拔插法:即通过插件的“拔出”或“插入”来寻找故障点的方法。

b、替换法:即用好的插件或部件替换可能有故障的地方来寻找故障点的方法。

其软件方面的故障主要从 CD-ROM、声卡、解压卡等多媒体设备驱动程序的安装设备是否正确,操作是否不当和是否有病毒入侵等原因进行针对性分析与解决。

举例说明如下:

问题 1:播放 VCD 影碟时,无图像。

首先应检查 CD-ROM 驱动器有无故障,这时你

可用一个 CD 程序光盘来检查一下,如果 CD-ROM 驱动器对 CD 程序盘也不能读,说明 CD-ROM 可能有问题,先检查一下 CD-ROM 的连接和其驱动程序安装有无差错,如果也没问题,那么就需要更换一台好的 CD-ROM 了。如果 CD-ROM 能读 CD 程序盘,那么说明 CD-ROM 本身没问题,这时需要再检查一下解压卡了,先检查解压卡的连线,接触性以及其驱动程序的安装有无差错,如果都没有问题,那么说明解压卡本身出现了故障,则更换一个好的解压卡后即可正常工作。

问题 2:在播放 VCD 影碟时,常出现图像抖动。

因为它仍能出现图像,说明 CD-ROM、解压卡硬件无故障,图像常会抖动是因为对解压卡的跳线设置不当造成的,通常只要将跳线设为 PAL 制式即可。

问题 3:用其长期播放 VCD 影碟过程中,有时会发现原来非常流畅的画面和声音变得断断续续,播放质量逐步下降。

遇到这种情况可以从以下三个方面来寻找原因:

①VCD 影碟、CD-ROM 驱动器是否被污染了,应确保其清洁;②CD-ROM 本身是否出现了故障;③CPU 速度的影响,在播放 VCD 影碟时,必须使机器设为高速状态。

问题 4:在播放 CD 唱碟时,无声音。

首先检查一下音箱声音开关是否开了,如果不是音箱关着的问题,则需检查一下是否 CD-ROM 出现了问题,你可以将一耳机插入到 CD-ROM 的耳机接口中,如果仍听不到声音,则表示 CD-ROM 出现了问题,这时先检查一下 CD-ROM 的连接及其驱动程序的安装有无差错,如果这些都没问题,那么说明要更换一个好的 CD-ROM 了,之后即可正常工作。如果能听到声音,则表明是声卡部分出现了问题,这时可先检查一下声卡的连线、接触性以及驱动程序的安装方面有无差错,如果均无问题,那么只要更换一个好的声卡即可正常工作。

问题 5:声卡使用一段时间后,在 Windows 3.1 环境下播放声音文件出现异常,任何声音文件都只播放一个音节便不出声了。如果此后再播放其它声音文件则系统提示:“该设备已被占用,待其空闲后重试。”这

对 AR3200 打印机使用手册的补充说明

王新华

(上海第三航务工程局一公司电脑中心 200434)

在市场上销售的 STAR-AR3200 打印机,有较好的打印功能。从说明书上给出的调整字间空距、字体

收稿日期:1996 年 6 月 25 日

时转到其它应用程序均能正常操作。出现这种故障后,在退出 windows 时会出现死机现象。但在未启动 windows 时,可在 DOS 下播放声音文件和 CD 唱碟,当用随卡带后软盘重新安装声卡的驱动程序,问题仍存在。

产生这个问题的原因是由于声卡的 windows 驱动程序出现了故障,只要替换掉这个有问题的驱动程序即可。用随卡带的软盘重新安装声卡驱动程序问题仍存在,是因为如果原来的驱动程序仍存在,则安装时不能被更新替换,只要将原来的出现故障的声卡 windows 驱动程序删除掉后,再装一个好的声卡 windows 驱动程序即可。

(2)网络方面故障分析与排除

其硬件方面的故障主要涉及到电话线路、路由器等整个网络系统上的故障,这时一般用户本人无法解决,必须向其网络服务提供者那里请求其解决,因此用户与其网络服务提供者应保持良好的联系。

其软件方面的故障排除主要从对相应网络通讯协议软件,调制解调器驱动程序等配置得是否合理,减少误操作,预防 worm 等网络病毒等方面着手。

举例说明如下:

问题 1:在使用 E-mail 时,假如收到如下 3 个 E-mail:

mail>DIR

From Date Subject

1. SMTP%"MAILER-DAEMON@ 25-JUN-1996

Returned mail:user unknown

2. SMTP%"MAILER-DAEMON@ 25-JUN-1996

Returned mail:host unknown

3. SMTP%"MAILER-DAEMON@ 25-JUN-1996

Returned mail:cannot send message for 3 hours

这 3 个 E-mail 表示你曾经发出的 3 个 E-mail 目前均没有成功发送出去。

收到的第一个和第二个 E-mail 是告诉你曾发出的 E-mail 的地址出错:第一个 E-mail 是告诉你用户名不能识别,可能是敲错了;第二个表示你发送的

选择、高密/高速打印等打印状态选择有三种方法:一种是开机后在脱机的状态下设定,再联机就能使用。这种方法在打印机重新开机时要重新设定。另一种是通过说明书给出的指令写在程序中来设定。硬件变化,软

E-mail 地址中的主机名出错,这时你需要检查一下你曾发送的 E-mail 的地址,修改正确后,需要重发。第 3 个 E-mail 表示你这个 E-mail 在你所属的网管中心主机上发送了 3 小时还未能发送出去,但这种情况一般你不需要再重发,它会自动地再给你重发。

问题 2:访问 ftp 服务器时,最常见的问题是被错误信息所截获如:

unknown host

造成这种错误的原因是敲错了 ftp 服务器的地址,检查一下 ftp 服务器地址的拼写,然后再试,如还出错,你如果能用 ping 命令,你可以借助键入 ping-s {hostname}ll 检查 ftp 服务器的存在和可访问性。

问题 3:当通过匿名 ftp 服务器传送文件时,在传送过程中突然中断。

因为匿名 ftp 服务器一般支持的客户访问是有固定数额的,当已超过这个数额,你就不能访问这台匿名 ftp 服务器了。即便你已匿名访问上这台 ftp 服务器,如果你在发送文件时,速度太慢、时间过长,这台 ftp 服务器也会自动取消你的访问。一般出现这种情形,你最好选择在夜间没有太多人上网的时候访问这台 ftp 服务器。

3 结束语

作者希望本文对读者在解决入网多媒体 PC 机故障时能够起到一点帮助与借鉴作用。本文的形成还得到了编辑部哲明先生、我所计算中心杨则群先生和我所出版组李晓波女士的大力协助,在此一并表示衷心地感谢。

参考文献

- 1 余先庆 入网多媒体 PC 机的配置及其应用 计算机应用研究 1996 年第 5 期
- 2 赵明生 多媒体系统常见故障与检修 微电脑世界 1996 年第 5 期
- 3 李涤非 声霸卡使用中常见问题及解决办法 电子与电脑 1996 年第 5 期
- 4 kevin kelly 等 Internet 奥秘 学苑出版社 1994 年

件就必须修改;还有一种是通过控制面板上的按钮,在开机的同时按下不同的按钮来设定。这种设定可使打印机在上电以后直接设定在某一状态;如同 EMS 设置方式一样。

在第三种开机功能设置过程中,该说明书上给出的图示和说明书写的与实际操作过程有误。

说明书中指出开机功能设置是:“在开机的同时按下不同的按钮。”并给出下图:

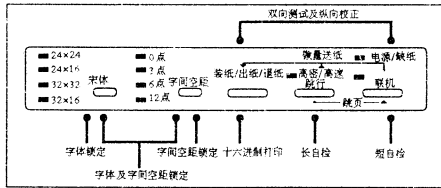


图 4-2 控制面板的开机功能

实际设置步骤是在开机后进行的。

如若要设置在开机后,打印机直接处于:字体在

“24×24 宋体”,字间空距在“0”点,高速状态下。

操作步骤为:

(1)开机后,按联机按钮使打印机处于脱机状态下。

(2)按字体锁定按钮,使发光二极管选定在你所希望的位置。

(3)按字间空距锁定按钮,使发光二极管选定在你所希望的位置。

(4)把打印机调到联机状态下,按高密/高速按钮,使其定在你所希望的位置,再脱机。

(5)先按下字体锁定按钮不放;再按字间空距锁定按钮等蜂鸣器响后,打印机状态被锁定。在这里要说明的是字体按钮是下沿边触发;字间空距按钮是上沿边触发。所以要先按字体按钮,后按字间空距按钮来锁定。

其中步骤 2、3、4 是可选的,步骤 5 是锁定。锁定之后每次开机就自动进入锁定的状态。

《计算机应用研究》杂志联(协)办单位

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

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

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

《计算机应用研究》杂志社董事会

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

《计算机应用研究》杂志编辑委员会

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

1997年第1期(总第75期)

出版日期:1997年1月31日

责任编辑:邓傢文

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

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

邮发代号:BM4408(国外)
62-68 (国内)

主 编:张执谦 副 主 编:李泽民

主管单位:国家科委、四川省科委

主办单位:四川省电子计算机应用研究中心

编辑出版:《计算机应用研究》杂志社

通讯地址:成都市人民南路4段11号附1号

邮 政 编 码:610041 传 真:5221946

电 话:(028)5210177 或 5220666 转 2055

印 刷:成都市新都华兴印务有限公司

订 阅 处:全国各地邮政局

国内总发行:成都市邮政局

国外总发行:中国国际图书贸易总公司