

计算机应用研究

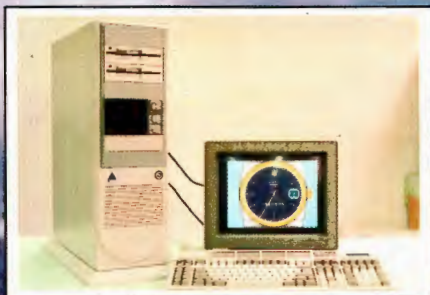
APPLICATION RESEARCH OF COMPUTERS

1994 4

证券交易计算机处理系统
计算机自动播音系统
多媒体应用系统
微机系列软件
新潮系列微机
工业控制计算机
开关电源
传感器
微机皮带秤

新潮电脑

时代新潮



ISSN 1001-3695



四川新潮计算机产业集团公司
中国四川成都新鸿路四号 邮编: 610051
电话: 4448100, 4444670 传真: 4444115

“安”和“稳”的故事……



“凯联”追求稳定的电压和自身的稳定！

我厂专业生产稳压电源，SBW、DBW系列三相、单相全自动大功率补偿式稳压器，是专为电网电压波动而设计的，是国内目前最为先进的节能型稳压电源。

SBW、DBW系列具有：功率大、损耗低、稳压精度高、输入范围宽、能承受瞬间超负载……。它广泛应用于邮电、工业、医院、交通、科研、宾馆……等一切需要电压稳定的场所。也是所有进口设备的必配电源。

主要技术指标及规格：

指标：稳压精度：额定电压 $\pm 1 \sim 5\%$ (可调)
稳压范围：额定电压 $\pm 20\%$ 损耗：1.5% (平均值)
波形畸变：0.1% 应变时间： $\leq 0.1s$ 有过压、故障自动保护
规格：三相 10, 20, 30, 50, 100, 180, 225, 300, 400, 500, 600, 800, 1000 (KVA)
单相 3, 5, 10, 20, 30, 50, 100, 180, 225, 300 (KVA)

“凯联”服务宗旨：上门调试 代办托运 来询即答 终身服务

SBW、DBW——系列 全自动大功率补偿式稳压器



上海凯联稳压设备厂

地址：上海市常德路379号 电话：2537960 2553368 传真：021-2553368 电挂：1454 邮编：200040

经 销 商：
• 山东省机电设备总公司 地址：济南市经五路225号 电话：733723
• 陕西省机电设备总公司 地址：西安市西五路22号 电话：775249
• 湖南省机电设备总公司 地址：长沙市五一中路6号 电话：420645
• 南京市机电设备总公司 地址：南京市太平南路514号 电话：2215084
• 浙江省机电设备公司 地址：杭州市延安路217号 电话：555646
• 闽东电机(集团)公司 地址：福州市五一一路144号 电话：524452

MICROTEK

旋極

領導世界最高水平開發手段之先鋒

——MICE 微處理器開發系統

- 提供完整微處理器系列開發系統
- 採用模塊化設計,只需要更換一塊模塊即可仿真其它 CPU
- 提供豐富的硬件斷點,具有總綫/執行/外部斷點
- 可設定斷點間 AND/OR/THEN 的邏輯組合
- 具有正反向邏輯狀態實行追蹤功能
- 可記錄 2k/32k 周期之各總綫及外部信號
- 提供 256KB/1MB 程序仿真存儲器
- 提供多窗口用戶界面操作
- 提供匯編程序反編譯程序選擇
- 提供萬用符號及高級語言調試程序選擇

提供完整系列軟件支持

- 匯編語言交互匯編程序(X'ASM)
- 萬用符號調試程序(USD)

NEW MICE- I S:

8048 8052 8085 8086/88 80186/188 80286
6502F 6809/E 68HC11 68000/10 Z80

MICE- II:

8086F 80C186/ 188 80C186EB/XL 80286
H8/532 V25 V20/V30 .

68000/EC000 68020/EC020 68030/EC030
68302 64180 Z80

MICE-V 80386 80386SX 80486 80486SX
80586(奔騰)

其它機型:

8096/196 683XX 80C186EA/EC 68HC16
68040/EC40/LC040 TMS 系列 320CXX

- C 語言交互編譯程序(X'COM)
- 高級語言調試程序(HLLD)

本公司欲舉辦研討會,有意參加者請來函來電聯系

不用不知道 用過都說好!

中國地區總代理
北京旋極科技公司

聯系地址:北京學院路皇亭子東源電子樓三樓
電話:2043258 2032211-818
傳真:2043258 郵編:100088

SBW、DBW 系列 三相、单相大功率电力稳压器

SBW·DBW系列三相、单相大功率自动补偿式电力稳压器是我公司引以为自豪的荣誉产品。多年来在引进欧洲技术的基础上，结合我国实际情况，使该产品性能不断完善、质量不断提高，销量不断上升，在强手如林，竞争激烈的市场中立于不败之地，深受广大用户信赖。该系列产品设有报警装置、自复装置、避雷保护、缺相保护、相序指示等多种功能，还可根据用户的特殊要求，三相分调、双速调节、配备隔离变压器，以适应任何国家引进的设备所需电压。欢迎联系，欢迎咨询。



指标：稳压精度：额定电压 $\pm 1\sim 5\%$ (可调)
稳压范围：额定电压Ⅱ型 $\pm 20\%$ (304~456V)
Ⅲ型 $\pm 25\%$ (285~475V)
波形畸变：0.1%
损耗：1.5%(平均值)
应变时间： $< 0.1s$ 有过压、故障自动保护
规格：三相：10~1200kVA
单相：3~300kVA



来函必复·备有现货·上门调试·服务完善

上海仪表(集团)公司系统工程部制造

地址：上海市大名路65号 电话：(021)3562757 3062998 邮编：200080

传真：(021)3230561 电挂：2599

开户银行：工行虹办提分处 帐号：212-04082840

Voltage Stabilizer



MOTOROLA 摩托罗拉半导体

中国代理

香港庆成集团●成都庆成电子有限公司

香港庆成企业有限公司(QCE)是美国摩托罗拉公司半导体产品的中国代理,成都庆成电子有限公司是其合资公司,为您提供全系列MOTOROLA 半导体产品。

CATV 模块放大器

★MHW 系列:40~550MHz 40~1000MHz MFF 系列 CA 系列...

射频器件

★MRF 系列 MRW 系列 TP 系列 TPV 系列 BFR 系列...

功率器件

★MJ 系列 MJE 系列 IRF 系列 MTM 系列...

单片机 MCU

★MC6805 系列 MC68HC05 系列 MC68HC11 系列...

通讯类产品

★全系列各种功能电话机 IC 各类程控交换机 IC...

光耦器件

★TIL 系列 4N 系列 H11 系列 MOC 系列...

传感器

★压力传感器 MPX 系列 温度传感器 MTS 系列...

电视机专用 IC

★彩色、黑白电视全套 IC 梳状滤波器...

有线电视用户机上变换器专用 IC

★MQS3001 MC44802 M3006

总部地址:成都市人民南路四段 11 号附 1 号(省电子所内)

电 话:(028)5212153 5212152 5582666—2018/2087 传 真:(028)5212154

门市部地址:成都市一环路南二段九号附 17、18 号 电 话:(028)5541255

香港总公司地址:香港荃湾青山道 264—298 号南丰中心 1901 室 电 话:(852)4934202(8 线)

电 传:57275 QCC IIX 电报挂号:1297 传 真:(852)4136307

目次

综述评论

无线网的发展及其相关技术.....	张震(1)
美国信息高速公路漫话.....	邓瑞祥(2)
生成真实感图形的四大技术	亢祝真 陆丽娜(57)

研究探讨

层次分类模型的构造及实现.....	王义章(3)
计算机辅助生产流程分析系统 Pro/FA	祝国旺 郑德涛 高健 周济 余俊(5)
正向思维与逆向思维交叉式软件设计.....	梁宗强(7)
汽车零部件失效分析专家系统的研究.....	毕常青 杨开英 张明友(9)
并行与分布式处理的错误调试	袁道华 陶辅周(11)
群集 UNIX 环境下的 ORACLE 并行服务器技术	李强 陶辅周(15)
ORACLE 中 SQL 查询语句的优化	李强 陶辅周(17)

应用实践

一个微型控制器设计中的技巧	张昆 张晓光(21)
内存状态显示方法	梁春成 田慧珍(24)
用软件应答进行串行通信	许青林(28)
微机控制系统的实时多任务调度	金彪(30)
生物组织结构定量分析系统的开发	高云钦(32)
一种通用的决策支持系统生成器	孙永生 许宗芬 王连海 孙军(34)
HD—COPY 的剖析	徐令毅 张烨(36)
滚轮电气传动智能监控系统	彭忆强 周文祥(36)
一个实用的全屏幕数据录入修改程序	王雄基(39)
制做带商标的管理软件封面	午锁平 张海蓉 午毅凯(42)
Foxpro 的记录成组访问技术 Rushmore	瓮正科(44)
Foxpro 键重定义的方法及技巧	侍永新(46)
也谈清屏	乐其斌(47)
Micro—PROLOG 和 DBASE 间的数据传递	李景贵(49)
软件的防拷贝措施与 C 程序介绍	何岭松 吴雅(55)

维护维修

硬盘的管理维护与常见故障处理	郭颖格(50)
M2724 打印机故障诊治一例	陈德明(53)
WPS 常见的故障与处理	周宏甫 宾鸿赞(53)

动态简讯

CAD/Graphics'95 第四届国际计算机辅助设计及图形学学术会议征文通知	(59)
简讯广告	(封1、封2、封3、封4、14、33)

无线网的发展及其相关技术

延安大学数学系计算教研室 (延安 716000) 张 震

摘要 本文叙述了发展无线网的必要性、需要解决的技术问题、现可采用的方案和它的应用方向。

关键词 有线网 无线网 网络协议 扩频传输 SST 和无线分组技术 红外线

九十年代是网络技术大发展的年代,而随着这种技术的发展应用,人们越来越感到现有技术有许多不足之处,迫切要求利用新的方式来连网,以便更有效地利用网络资源,这就是大力发展无线网技术,以扩展和深化网络应用领域。

1 无线电网络发展背景

目前大多数局域网均采用有线方式连接以实现数据的传输,但是,有限网再好,它的覆盖范围也要受到线路的限制。有些场合,不可能在现场提供物理连线,这就影响了网络技术的普及和应用。另外,在比几公里的区域更大的范围内进行计算机连网时,常需使用异步调制解调器通过专用或拨号电话线实现,但受到电话线路及线路质量的制约,数据传输率低。鉴于以上原因,国内外一些厂商先后成功地发展了多种无线网络产品,彻底弥补了有限网络的上述缺陷,丰富了计算网络的内涵,提供了更多的网络通讯实现方案,同时也简化和方便了计算机连网。

2 无线网需解决的技术问题

首先,无线电网络是对有线电网络的补充和进一步发展,因而无线电网络的设备必须适合现有网络协议,如以太网协议、RS232C 标准等,只有这样才可真正支持现流行的网络协议,如 TCP/IP、SPX/IPX、DECnet 等;其次必须解决无线传输过程中的抗干扰、保密及无线频道的管制、天线辐射电信号的自适应等问题;再者应有较高的数据传输率。若不能在以上技术上有创新突破,所谓无线电网络产品就没有生命力。

3. 现采用的无线网络通信技术

在特定的场合下,可使用红外线进行网络通信。红外技术的主要优点是不受任何国家频率管制的限制,成本低,速度快,抗干扰性能好、保密性好。不能穿过墙壁、地板,是红外技术的缺点,因而限制了它的使用范围。如美国 Spectrix 公司开发了一种蜂窝式网络,它使用了有线与红外相结合的技术,可以覆盖整个建筑和一些特定位置的房间。

为了改进红外系统的缺点,现在无线网常采用扩频传输 SST 和无线分组技术。SST 技术代表当代通信的最新技术,其核心为在发送射频之前将其扩宽,使发

送的射频能量分布于更宽的频带,这样会使单位带宽面积的射频能量很小、信噪比很小,因此传输的信号很难捕获,而且不干扰其它正常传输无线电信号;其它无线电信号也干扰不了 SST 传输的信号,故无需申请专用固定频道即可使用。在授权的接收方再将扩频还原成正常的无线信号,并根据校正技术校正接收的信号。整个过程实际上是完成扩频的调制、发送和解调过程。

无线电信号的自适应技术的核心是无线分组和网络路由节点地址自动寻址技术。无线分组技术可同时将无线信号分组多路复合发送,网络寻址解决网络的透明访问,从而保证了高达 1.35M 的传输率,及自动转接和网络寻址的智能化。

采用以上技术的无线网络通信系统产品有:无线以太网板、无线网桥、无线网络 HUB、无线中断器、支持标准 RS232C 的无线接发设备,从而为网络多用户应用提供了更加灵活方便的实现方案。

4 无线网络的应用范围

无线网络现已基本成熟,可以完全实现以太标准的网络系统(如 TCP/IP、NETWARE、DECnet 等)及多用户系统(如 UNIX、VMS、MVS 等),也可以和有线方式混合使用。

由于无线网络具有有线网络所没有的一些特点,故可应用于布线困难、移动通信、抗干扰要求高和高速通信的场合,如军队、医院和商场等,这样可节省布线所需介质的投资和线路维护的成本。

5 结束语

总之,无线网有其优良的特性,故在军用和民用方面有着广阔的应用前景,会备受人们青睐的。特别是在国内,目前我国通信状况较为落后,通信线路紧张,况且有些地方难以修建有线通信网,无疑无线网为我们提供了有力的数据通信手段。因此我们必须重视无线网技术的研究和开发。

参考文献

- 1 薛锋等. 微机通信技术大全
- 2 计算机世界月刊, 1994. 3

美国信息高速公路漫话

四川省华夏电脑信息工程公司 (成都 610061) 邓瑞祥

随着信息量的不断膨胀爆炸,处理信息的设备—电子计算机及外设等不断的改进,现今单一的视、听、数字等信息的传输远远不能满足用户的需要,加上电视电话的飞快兴起,传统的电话线路已成了制约高质量信息传输的瓶颈,为解决上述不足,一种崭新的构想——“信息高速公路”(本文简称“信高路”)由美国科学家提出,并逐步成为美国乃至世界的热门话题。1993年底美国总统正式批准在美国的纽约,华盛顿,洛杉矶,旧金山首先实施“信高路”计划,并计划在今后的10—15年内在全美实施信高路计划。届时全美各地只要有电话线和电话机的地方,人们都享受信高路带来的方便和好处;全国乃至全球性的电脑联网、企业界、政界领导人可在各自的办公室里进行跨国或跨洋的谈判和会晤,学生可在家里“听课”用户可在舒适的家里,根据各自的喜好选物或选看娱乐库里的电视和立体声节目。文职人员可在家里处理文件,并及时与单位联系。各种信息的获取都可能是在键盘上“弹指一挥间”的功夫。电子邮件将会减少或取代传统的邮件,随着人脑通过电脑在信高路上的交流,在公路上会少见运人客车,和来往行人……一句话,“信高路”必将影响美国的经济的发展,而且也标志着当今信息时代在信息交流,以及通信领域的重大革命性飞跃。

这样好的构思,这样伟大的发展战略,一经提出,立即受到世界发达国家和地区的重视。自美国总统宣布搞“信高路”后不到4个月,日本政府和欧州发达国家也相继提出了各自的“信高路”方案,并在技术上提出了一些新的方案。

1 信息高速公路的先行产业—光纤通讯

光缆——光导纤维缆是“信高路”的网,联结光缆的光端机和电脑就是“信高路”的络。理论上,直径不到1.3CM,由32根头发丝粗细的光纤组成的光缆能同时传输5000个频道的电视图像信号和50万路的电话语言信号,但实际上要达到理论数据还得花相当大的力气。目前美国卡泰尔电信工业公司推出的“异步传输码”(ATM)的交换和传输技术是信高路电子的高技术产品,它能在一根光纤上传输60路电视频道信号和数字及语言信号,也就是说它可在一根光纤缆上传输1920路电视信号。在电话通信方面一根32根光纤组成的光缆,可传输983040路电话信号,所以发展“信高路”是以光纤通讯为先导的技术革命,而这种革命已初

步取得成功。

2 超智能网络使信高路“天堑变通途”

围绕着“信息高速公路”的发展,人们不仅仅满足于电视信号和声音信号的传输,数字信号的传输量将越来越大,由于数字信号的传输受到电脑网络发展的阶段性局限,由于一种网络系统和另一种网络系统不能相互兼容而造成网与网之间交流的困难制约着数字通讯的发展。这样就好像有无数江河沟壑在阻碍交通一样,人们就会寻找一种新的方法,最好能容已知的所有网络,而不同的网与网之间的交流通过这种网后可自动地翻译成各网之间固有的技术语言,也用不着投资重建新网络将老网络废弃,在原有网上开发的软件仍可使用,这就象讲各种语言的民族用一种能翻译所有语言的“快译通”一样。据报道一种同Ethernet网类似,能够兼容所有网络系统的一种超智能的网络系统已由美国“微软”研制成功,已经投放市场,这无疑大大推动了“信高路”的数字通讯的发展,据说在北京人民大会堂,以及国家的七个部委都可能使用这一新的网络系统。该网络同时也受到上海、海南省政府和有关单位的青睐!也许在不远的将来世界上将会刮起一阵争上这种新网络的“风”。

3 “信息高速公路”的各终端站和美国“信高路”上使用的微机

有了光纤通讯的干道和分道有了适应各网络的智能高速网。那么在网络上使用的终端站电脑应该是什么样的机器呢?首先让我们来看机关或家庭有什么样的需要;第一、较高层次的机关用机要求能处理各种文件,能满足一些科研设计的要求。第二、能随意查阅文件和调用图书馆和娱乐库里的文件和节目。第三、随着时代的发展,能只换微处理器或加上Overdrive处理器就能升级的微机,处理速度至少是零等待。第四、作为家用,它既是一个可在上面搞开发设计的电脑,又可作电视电话用还可欣赏音乐和电视节目,反正一切信息都能处理存储传输。第五、也是最重要的,它能方便地上网与外界联接。要达到上述要求,就必然是一种486档次的微机,而且应具有声音和图形的处理存储功能,这就是多媒体微机。

据报道:为了满足上述要求美国EVEREX公司已推出一种适应时代要求的新型多媒体微机—EVEREX STEP PLUS,EVEREX公司所(下转27页)

层次分类模型的构造及实现

贵州省科委计算中心 (贵阳 550002) 王文章

摘要 本文提供了一种层次分类模型的构造方法。利用这个模型建立的决策支持系统(DSS)运行表明,框架,规则是可行的,模型是有效的。这个模型也适用于可层次化的复杂系统的分类和辨识。

1 引言

现实生活中大量复杂系统的组成,普遍存在着等级分层现象,或可等级分层分解^[1]。至少有三个理由可以说明何以复杂系统一般都是可层次化的:

1.1 在具有给定复杂性的各种可能存在的系统中,通过其演化过程最可能出现的是由分系统构成的分层等级系统,分层结构的各部份本身都是稳定的系统。

1.2 在给定的复杂性的系统间,分层等级系统各部份间所需的信息传输量要比其它类型系统少得多。

1.3 对分层等级结构来说,一个系统的复杂性,从系统中任一特定位置来观察,几乎与其总规模无关。

的确,具有充分的理由使我们在对复杂问题或系统的分析中,首先考虑人类社会普遍存在的那种组织层次结构或可层次化结构。例如,多目标决策中,涉及的因素往往较多,因素间相互联系又相互制约,使决策问题复杂化。因此,决策问题的科学方法,一般是基于对问题的全面系统分析,将复杂问题分解成相对简单的子问题,然后再将各子问题间的相互影响进行综合分析,同时又将各子问题视为一个较复杂的问题进行分析分解,如此进行分析的细化和深入。可以看出,分析过程本身就是层次化的过程。如果将影响决策的诸多因素视为指标,那么,对多目标决策问题的分析过程,首先就是将决策专家(或部门)的知识通过分析,分解,建立指标体系。这个指标体系可以视为描述决策对象的知识框架。它的最简单,最常用的结构就是层次化的树状结构。

本文针对可层次化的复杂对象的多目标决策,建立一种分层分类模型。众所周知,决策方法也是模式识别的基本方法之一^[2],所以,本文提出的方法也可用于模式识别与分类。

2 预备知识

由于可层次化的复杂系统其知识框架可以用树状结构来描述,它具有如下二个特点:1. 复杂系统由简单系统构成,而简单系统又是一个可层次化系统的递归过程;2. 每一子系统在特定问题背景下,描述了决策专家某方面的知识,有其特定内涵,因而是一棵有序树。

我们可以用一个上下文无关文法 $G(S) = (N, \Sigma, P, S)$ 来形式定义一个分层结构。其中: N 既表示非终结符集又表示非叶结点集; Σ 既表示终结符集又表示叶结点集; P 表示产生式集; S 表示开始符同时也表示树根。

对结点 $N_i i_1 i_2 \dots i_k \in \Sigma^* N$, 引入脚标。脚标个数 k , 表明该结点处于第 kP_i (深度)。每个脚标的值,表明它是其父亲的第几个儿子,且按自左至右顺序。根结点 S , 是第 0 层。

定义 1: 对一棵树 T , 结点 $N_i i_1 i_2 \dots i_k$ 通过脚标, 定义偏序关系“ α ”如下:

$$N_i i_1 i_2 \dots i_k, N_j i_1 i_2 \dots i_l \in N \cup \Sigma$$

$$\text{若 } \sum_{l=1}^k 10^{k-l} > \sum_{l=1}^l 10^{l-1}$$

$$\text{则 } N_i i_1 i_2 \dots i_k \alpha N_j i_1 i_2 \dots i_l$$

有了结点间的序关系,我们可以把一个层次化的知识框架视为一个上下文无关文法 $G(S) = (N, \Sigma, P, S)$ 的派生树^[3] (Derivation Trees) (或推导树)。

定义 2: 派生树的边沿 (Frontier) 是全体树叶结点所组成的字符串 (自左至右顺序)。

定义 3: 派生树边沿所得到的一组取值,称为一个采样。

采样过程是每个叶结点逐一进行,也可从是交互式,某些叶结点的值可以通过方法库中反映专家知识的函数计算得到,视具体问题背景定。

在采样基础上,根据专家知识,自底向上对各层结点进行整合运算,得出各结点的估价函数值,直至对 $N_1, N_2 \dots N_m$ 计算出估价函数值 $a_1, a_2 \dots a_m$ 。整合运算规则以合取范式表示: 对于“与”关系的谓词短语,取子树估价函数值的最大(或最小)者; 对于“或”关系取子树估价函数值的和^{[4][5][6]}。在计算各结点的估价函数值时,若涉及到各子树的权重分配,我们采用 AHP 方法^[7]。记 $\alpha_1, \alpha_2 \dots \alpha_m$ 是以 $N_1, N_2 \dots N_m$ 为根的子树的权, $\sum_{i=1}^m \alpha_i = 1$ 。

定义 4: 论域 $X_1, X_2 \dots X_m, X_i \in R \quad i=1, 2 \dots m$ 。记 $X_1 \times X_2 \times \dots \times X_m$ 形成的空间 Ω , 若 $A_i \in \Omega, i=1, 2 \dots m$ 。 $A_i = \{a_{i1}, a_{i2} \dots a_{im}\}$, 称为 Ω 中的一个样本, $a_{ij} \in X_j, j=1, 2 \dots m$, 称为 A_i 的第 j 个指标(值)或属性(值)。这个空间 Ω , 称为特征空间。

对于每一次采样,通过整合运算,都可以构成空间 Ω 的一个样本,记第 i 个样本为 $A_i = \{a_{i1}, a_{i2}, \dots, a_{im}\}$ 。因此,从采样到形成 Ω 中的一个样本的过程,可以视为是特征抽取的过程。

上面的预备知识,提供了在知识框架确定以后,特征空间中样本生成的算法以及形式检查的基础。

值得指出一个具体问题,层次分解成树状结构一般不是唯一的,但在专家知识参与后,确定一个知识框架是可行的。在分层分解过程中,各子树(指标)间内涵对应相互独立。

从结点序关系的定义,可以看到,是采用宽度优先的序。在实现中^[8],我们利用生长树方法来建立知识框架并组织相应的数据结构,建立方法库的组织。

3 分层分类

设将 Ω 中的样本分成 $k+1$ 类,只需将空间 Ω 作如下 $k+1$ 个部分:

3.1 建立各类质心,共 k 个,记为:

$$Q_1 = \{q_{11}, q_{12}, \dots, q_{1m}\}$$

$$Q_2 = \{q_{21}, q_{22}, \dots, q_{2m}\}$$

$$Q_k = \{q_{k1}, q_{k2}, \dots, q_{km}\}$$

质心的形成方法很多,可以根据问题背景选择。比如采用各种求聚类中心的方法。

3.2 以 $\alpha_1, \alpha_2, \dots, \alpha_m$ 为法矢,过点 Q_1, Q_2, \dots, Q_k , 在空间 Ω 作平行超平面族:

$$C_i: \sum_{j=1}^m \alpha_j (x_j - q_{ij}) = 0 \quad \dots\dots\dots (1)$$

$$i = 1, 2, \dots, k$$

3.3 对每一个样本 $A_i = \{a_{i1}, a_{i2}, \dots, a_{im}\} \in \Omega, i = 1, 2, \dots$ 计算 A_i 到平面族 C 中每个平面的距离。

$$a. \begin{cases} \sum_{j=1}^m \alpha_j (X_j - a_{ij}) = 0 \\ \frac{x_1 - a_{i1}}{\alpha_1} = \frac{x_2 - a_{i2}}{\alpha_2} = \dots = \frac{x_m - a_{im}}{\alpha_m} = \lambda \dots\dots (2) \end{cases}$$

$$\text{记解为 } Y_j = \{y_{j1}, y_{j2}, \dots, y_{jm}\} \quad j = 1, 2, \dots, k$$

$$b. d_{ij} = \|A_i - Y_j\| \quad \begin{matrix} i = 1, 2, \dots, k \\ j = 1, 2, \dots, k \end{matrix} \quad (3)$$

d_{ij} 是 Ω 中两点间的距离,根据问题背景,可从选取不同的测度加以定义。

4. 分类规则

$$d_{ij_0} = \min_j \{d_{ij}\} \Rightarrow$$

j

$\begin{cases} A_i \in j_0 \text{ 类, 若 } j_0 \text{ 唯一} \\ \text{拒分类, 交互式补充知识, 若 } j_0 \text{ 不唯一} \end{cases}$

4 实现

我们利用本文提出的方法,实现了“科技成果辅助评奖系统”,该系统运行两年来,表明框架、规则、方法是可行的。为节约篇幅只对指标体系及系统框图列出^[8]。

N_1 : 技术学术水平

N_2 : 经济效益

N_3 : 社会效益

N_{11} : 创造性

N_{12} : 先进性及科学性

N_{13} : 难度

N_{14} : 论文

N_{15} : 专利

N_{21} : 增产值

N_{22} : 增利润

N_{23} : 增税收

N_{24} : 创汇及转让收入

N_{25} : 偿还情况

N_{26} : 节约投资

N_{27} : 间接经济效益

n_{28} : 同行业效益比较

n_{31} : 提高生产率%

n_{32} : 降低消耗(能源或原材料、贵重金属等)

n_{33} : 改善劳动条件

n_{34} : 标准化程度

n_{35} : 国产化程度

n_{36} : 技术(产品)出口,减少产品进口

n_{37} : 对生态环境的影响

n_{38} : 社会受益面,受影响的深度及实用性

n_{39} : 科技政策因素

n_{111} : 理论上的创造性

n_{112} : 技术上的创造程度

n_{121} : 理论(技术)高度

n_{122} : 实现方法的先进性

n_{123} : 对科学理论和技术发展的作用

n_{124} : 资料及实验数据的完整、可靠

n_{131} : 研究涉及学科的广度

n_{132} : 所用技术或知识的深度

n_{133} : 研制工作的规模

n_{134} : 研制工作的条件

n_{135} : 困难程度评价

n_{141} : 发表论文刊物的最高级别

n_{142} : 宣读论文会议的最高级别

n_{143} : 正式出版物上的引用次数(国内外)

n_{211} : 年增(一次性)产值(万元)

n_{212} : 合计增产值(万元)

n_{221} : 年增利润(万元)

n_{222} : 合计增加利润(万元)

n_{231} : 年增税收(万元)

n_{232} : 合计增加税收(万元)

n_{241} : 创汇

n_{242} : 直接转让收入(万元)

计算机辅助生产流程分析系统 Pro/FA

广东工学院计算机软件开发研究中心 (广州 510090) 祝国旺 郑德涛 高 健
华中理工大学机械一系 CAD 中心 (武汉 430074) 周 济 余 俊

摘要 本文介绍了一个用于成组技术的计算机辅助生产流程分析原型系统的各个主要模块的功能、部分模块实现的算法,并对部分输入、输出结果作了必要的讨论,为借助于软件技术在我国企业中推行成组技术与思想作了有益的尝试。

关键字 生产流程分析 成组技术

1 引言

成组技术(Group Technology)是一种适合于中小批量零件生产的技术,尽管某些零件的材料、形状和尺寸不尽相同,但只要具有相似的工艺过程,便可放在一个加工单元(或机床组)中制造。该单元的机床是按零件工艺相似的规律组合在一起,并装备着规定的刀具,安排成为一个整体。

成组技术被认为是本世纪继流水线生产方式之后的又一重大发现^[1],其关键是机床的成组平面布局、短循环流通控制以及投产顺序计划。其中成组平面布局的主要节约效果之一是来自物料流通系统的简化。为

了实现成组平面布置,从而改善企业物料流系统,通常要运用生产流程分析技术。生产流程分析时要处理大量的诸如机床加工能力、各种零件工艺路线以及许多表格等数据。这些数据对于手工处理来说往往是很复杂和耗时的^[2]。为了加速成组技术在我国企业中的推广和应用,提供良好的实行成组技术的手段,本文作者结合某纺织机械厂开发了一个计算机辅助生产流程分析系统 Pro/FA (computer - aided Production Flow Analysis)。

2 Pro/FA 系统

Pro/FA 系统由四个主要模块组成,图 1 是 Pro/

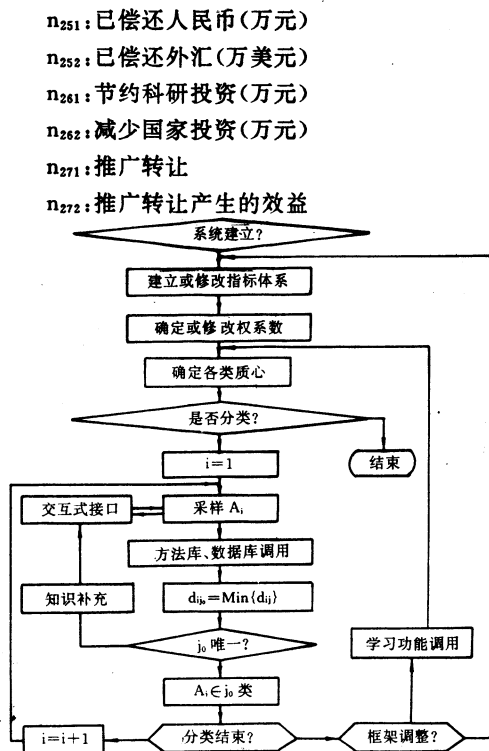


图 1 系统原理图

在实现中,科技水平的进步和发展,以及专家知识

随着认识的深化,知识框架及相应的数据结构、方法库要便于调整,同时在诸如质心选择等方面,引进了学习功能,增强了系统的自适应调整。对于科技成果辅助评奖系统中,我们选择了历年科技成果评定资料作样本,利用学习功能建立了系统质心动态,运行一年后,再调用一次学习功能。在没有历史资料可用的情况,学习功能也为建立动态提供了手段。

参考文献

- 1 H. A. Simon The New Science of Management Decision. Prentice-Hall, INC. Englewood cliffs 1977
- 2 R. C 冈察雷斯 M. G 汤姆逊,句法模式识别,清华大学出版社 1984
- 3 邹明海,形式语言、自动机和语法分析,华中工学院出版社
- 4 Vipin Kumar L. N kanal, A General Branch and Bound Formulation for Understanding and Synthesizing AND/OR Tree Search Procedures, Artificial Intelligence Mar 1983
- Frederick Hayes-Roth D. A Waterman D. B. Lenat, 专家系统建造,山西人民出版社 1986
- 6 A. P. Sage C. C. White, A Knowledge-Based Interactive System for Planning and Decision Support IEEE Trans on System Man & Cybernetics Jan/Feb 1984
- 7 许树柏,层次分析法原理,天津大学出版社 1988
- 8 郝嘉伍,王义章,贵州省科技成果辅助评奖系统技术报告 1990

FA 系统的 IDEFO^[3] 简图。下面分别介绍这四个模块的功能。

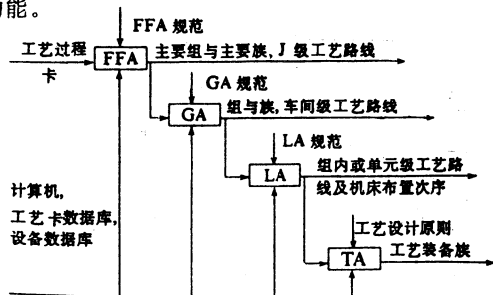


图1 计算机辅助生产流程分析系统IDEFO简图

FFA(工厂流程分析)模块的功能是合并加工单位以形成主要组,以及找出在这些组之间的尽可能简单的流通系统。它实际上是研究工厂中不同生产过程之间的物流系统。用来实现每种生产过程的每张设备组成明细表,被称为“加工单位”(Process Unit)。在必须将不相容的生产过程分开的前提下,工厂流程分析把生产一批相同零件的各个加工单位组成更大的组,叫做“主要组”。通过这一分析所找出的各主要组,包含了最可能协调生产的机床组。这些机床组可以完成各自“主要族”中全部零件的加工,而无需中转其它主要组,或外包生产。这种改革简化了物流系统。凡是不适合于简化过的新物流系统的零件工艺路线,则分别予以审查和修改。

GA(组的分析)模块是要找出设备的组合,即所谓的“组”,以及由这些组所产生的零件明细表,即所谓的“族”。通常,相似的设计特征也要求相似的工艺过程。在组分析时,就是要寻找具有相似的工艺流程的那些零件族,以及使得每个“族”的零件仅仅在一组设备中完成全部加工的设备组。即找出“组”与“族”的一对一关系。

LA(线的分析)模块是要研究物料在组内设备之间的流通。它通过进一步分析零件过程卡中的信息,就能找出布置机床的最佳次序。换言之,这种分析能找出机床的相对位置使之最接近于流水线。

TA(工艺装备分析)是要找出零件的工艺装备族,这些零件在它们的安装调整布局中采用相同的工艺装备,同时还要找出最佳投产顺序。

以上四个主要模块对应了四种不同的功能。在Pro/FA的程序实现中,这四个不同模块且有着极相似的算法。下面以GA模块为例,说明其程序实现方法:

第一步,读取 FFA 输出的“主要组”所含的机床号以及相应的“主要族”里的零件号;

第二步,构造稀疏矩阵;零件以行表示,机床以列表示,在行列交叉处填值1或0。当填1时表示该行的零件在该列所对应的机床上加工;当填0时表示该行的

零件不在该列所对应的机床上加工;

第三步,重新排列稀疏矩阵,使得每行元素值之和按升序(或非降序)排序,每列元素值之和按降序(或非升序)排列;

第四步,以稀疏矩阵表中的第一列为准,凡各行的这一列元素非零的元素则移到元素为零的行之上;

第五步,以下一列为准,凡是这一列元素为1的行均移到这一列元素为0的行之上,但在上一列元素为1的行之下,并重复这一过程直到所有行重新排列为止。如果当前的矩阵表和前一个矩阵表相同,则转第八步(停止),否则转第六步;

第六步,以矩阵表的第一行为准,凡各列的这一行上的元素值为1的列都移到其它元素值为0的列的前面(即左面);

第七步,以下一行为准,凡各列的这一行上的元素值为1的列都移到元素值为0的列的前面(左面),但在上一行元素值为1的行之下。重复这一过程直到所有列重新排列为止。如果当前矩阵表与前一个矩阵表相同,则转8(停止),否则转第四步;

第八步,停止。

四个功能模块都含有自身的打印输出子模块,根据不同的功能输出不同的内容。Pro/FA 由 C-FoxBase2.0 及 BorlandC++2.0 编写,在 386 微型计算机上运行。下面以 GA 模块为例给出其输入及输出结果。(见表1及表2)。

表1:由 FFA 产生的结果(作为 GA 的输入)

机床号	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
零件号	1	0	1	0	0	0	0	0	0	1	1	1	0	0	0
2	0	0	1	0	0	0	0	0	0	0	0	0	1	0	1
3	1	0	0	0	1	1	0	1	1	0	0	0	0	0	0
4	1	0	0	1	0	0	0	0	1	0	0	0	0	1	0
5	0	0	1	0	1	0	0	1	0	0	0	0	1	1	0
6	1	0	0	1	0	1	0	0	1	0	0	0	0	0	1
7	0	1	0	0	0	0	1	0	0	1	1	1	0	1	0
8	0	0	1	0	1	0	0	1	0	0	0	0	0	0	0
9	0	0	0	1	0	1	0	0	1	0	0	0	0	1	0
10	0	1	0	0	0	0	1	0	0	1	1	1	0	0	0

表2:由 GA 产生的结果

机床号	14	9	6	4	1	15	13	8	5	3	12	11	10	2	7
零件号	9	1	1	1	1	0	0	0	0	0	0	0	0	0	0
4	1	1	0	1	1	0	0	0	0	0	0	0	0	0	0
3	1	1	1	0	1	0	0	0	0	0	0	0	0	0	0
6	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	1	1	1	1	1	0	0	0	0	0
5	0	0	0	0	0	1	1	1	1	1	0	0	0	0	0
2	0	0	0	0	0	1	1	1	1	1	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0
10	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1
7	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1

正向思维与逆向思维交叉式软件设计

西安医科大学第一临床医学院计算机室 (西安 710061) 梁宗强

摘要 本文主要描述了思维方式和思维过程对软件设计的影响;正向思维、逆向思维两种软件设计的特点;正、逆向交叉思维方式及其软件设计总原则,试图从另一种角度提高结构化程序设计(SD法)的效率和质量。

关键词 正向思维 逆向思维 软件设计 数据库 知识库

目前所流行的软件设计方法之一是采用自顶而下、自左而右的结构化程序设计方法(即SD法),这在某种程度上讲,大大提高了软件的设计效率,然而,由于自然界事物的多面复杂性,使得我们分析与解决问题时不得不调整思维过程。有时需要顺序思维(即正向思维),而有时需要反向思维(即逆向思维),这就决定了以后软件设计的方法尽可能朝着符合人体思维的过程发展,即在常规过程中能动态处理多种突发事件,正向与逆向自适应事件要求,且系统能易于扩展。

1 正向思维 逆向思维

正向思维是按照事物的顺序发展过程进行事先推理、事务处理、事后总结等一系列步骤,它是常见事物的正常发展处理过程。大多数情况下,正向思维是应用经验知识与理论知识进行普通常规处理,这种思维方式应用于许多学科领域的诸多方面。如医学学科中的

常规疾病诊断与处理;气象学与水文地质的测算分析等等。在医院里,一位病人来看病时,首先要向医生诉其感觉与症状,医生们再应用自己的经验或前人的经验(知识)把这种病情划归为某一类型或某几种类型,然后再进行这种类型疾病诊治方案的筛选,最后进行治疗(即对症下药);如果这个病人是复诊者,则医生再根据上次诊治的情况和近期效果进行总体方案的调整。显然,这其中的每一环节都必须明确的,不能掺杂着模糊不清的概念,这类软件设计过程一般也是按照通常的情况进行处理;亦即自顶而下的模块化设计方法SD。

- 建SD模型
- 建数据库
- 分类分块常规处理

这种顺序处理的优点是简单明了,易于移植,不足

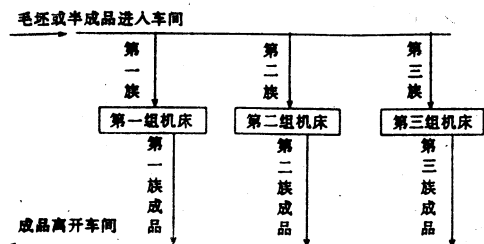


图2 由GA模块产生的物流与机床组布置示意图

由GA计算结果可以看出该“主要组”可以分为三个组,第一组由14号、9号、6号、4号、1号等五台设备组成;第二组由15号、13号、8号、5号、3号等五台设备组成;第三组由12号、11号、10号、2号、7号这五台设备组成。零件的主要族可以分为三个族(又叫生产族),第一族由9号、4号、3号、6号等四种零件组成,第二族由8号、5号、2号这三种零件组成,第三族由1号、10号、7号等三种零件组成。并且第一、二、三族的零件依次分别由第一、二、三组的机床加工。按成组思想可能的物料流与机床布置如图2所示。

每个组内机床的先后秩序的排法有很多种,例如仅第一组就有120种可能的排法(当然有些排法是不合理的)。到底哪一种排法最好,这个任务就由LA模块来完成。LA的功能在前面已谈过,由于篇幅所限,这里不再对LA及TA进行详细讨论了。

3 结论

成组技术是一项能使工业生产产生巨大经济效益的新技术,是本世纪继流水线生产方式之后的又一重大发现。生产流程分析是按成组思想组织生产前必需经历的重要步骤。本文作者开发的计算机辅助生产流程分析系统Pro/FA试图为成功地和有效地推行成组技术提供现代化的分析手段,并已在某纺织机械厂中得到初步运用,相信她会为成组技术在我国企业中的成功运用起到应有的作用。

参考文献

- 1 [英]约翰L·伯比奇著,蔡建国译. 成组技术导论,上海科学技术出版社,1986
- 2 Andrew Kvisan, "A Knowledge-based system for Group Technology", SME Technical Paper, 1989
- 3 漆海卫,祝国旺,钟毅芳,周济, "IDEF方法及其在CIMS系统中的应用", 湖北工学院学报, Vol. 8, No. 2-3(1993), P58-62

之处是无法(或很少)进行异常处理、无法(或很少)进行突发处理,无法反馈/修改,且每步需从头启动处理。

逆向思维正好弥补了正向思维的缺点,它可以直接从某个子过程的结果开始逆向思考、逆向推理、产生出可能的中间结果与条件,再判断与事实的符合情况,它不需要推到事务的开始,只需满足所需的条件即可,即不用推个“水落石出”,它可以中途得出结论,也可以不断添加/修改推理过程,可以加权限制,可以进行升降式搜索。但逆向过程又有机构过于庞大”、“小题大作”的缺点,而且有时会严重影响软件效率。其软件处理过程为:

- 升降式搜索结果。
- 升降式搜索中间结果或条件。
- 对比分析。
- 反馈与修改。
- 确认。

这种思维模式也已应用于多种知识领域。例如,就拿病人看病而言,有时病人已不能言语,或症状描述不清,或描述与某些检查有误差,这就使得医生一时难以按常规方法明确诊治。这时候,如果应用逆向思维,先从存在结果的知识库中搜索各种可能符合的结果(这种搜索可用升降式搜索法),再进一步升级从中间知识库中搜索出可能符合的结果与条件,经分析对比,提出差异,并重复上述过程,最后确认其结果(得什么样病?怎么样处置?)。这种方法能够推理或治疗一个病人身患多种疾病或综合症的情况。软件设计除了按正向 SD 法外,更重要的几点是知识库的搜索、维护与不断补充和更新。因而可以这么说,它的生命力较顽强。缺点是简单事物复杂化,或设计过于庞大,那么,能否提供一种正逆双向交叉方法呢?回答是可能的。

2 正逆双向交叉思维方式及软件的设计

正逆双向交叉思维实质上是人们日常经常采用的方式,它兼备正向思维和逆向思维两者的优点,即对一个问题从前提和结果同时考虑,提出解决问题的方法,然后采用结构化程序设计 SD 法来实现算法。

虽然模块化结构设计能够提高软件的质量,使程序易理解、易维护,并使系统易于多人并行开发,但模块化的效果同划分模块的标准是密切相关的。正逆双向交叉法能够通过多种软件设计法实现。

在整个一个系统中,软件的一个特点是不断地需要修改,在整个软件的生命期中,用户的要求会发生变化,例如数据结构需要改动;另一方面,系统的操作环境也会变化,如机型、存储设备等变化。这些变化都需要程序系统本身作出相应的修改。我们需要考虑的问题是:如何将整个系统适当地划分成若干个模块,使得

上述变化对软件结构的影响尽量小一些。

Parnas 提出了信息隐蔽法,我们可以把 Parnas 的这一原则适当地应用于正、逆向思维的程序设计中。

首先是提问题、摆条件,并罗列出一些可能发生变化的因素,在设计模块时将一些可能发生变化的因素稳含在某个模块的内部,使其他模块与这些因素无关,这样,将来这些因素发生变化时,我们只需要修改一个模块就够了,而其他模块则不受这个因素的影响。也就是说,在设计模块结构时,将一个可能发生变化的因素隔离在某个模块的内部,这个因素的变化就不致于传播到所有模块的边界之外。

我们可以看看下面的一个设计:

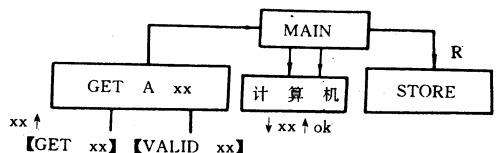
某一程序输入一个实数到变量 XX,对它进行合理性检查,然后进行计算,最后将计算结果存入存储器中。根据实际情况,我们估计到以下三个因素可能发生变更:

a) 输入数据 XX 的格式。目前 XX 的格式为六个数字型字符及其他,其中有符号位、三位整数、小数点、和一位小数,但将来会有变化。

b) 对 XX 作合理性检查的规则。目前假设只有 XX 的数值范围在 -100.0 到 +100.0 之间就认为它是合理的,将来打算采用某些更精确的检查规则。

c) 目前将计算结果存入存储器中,将来要存入其他外存(如磁盘、磁带等等),存储格式也有变化。

我们可以采用隐蔽法将程序设计成如下图示:



其中 GET XX 隐含了上述第一个因素,而模块 VALID XX 隐含了第二个因素,模块 STORE 隐含了第三个因素。对于其他模块则不必了解这些因素。将来一旦某个因素发生变化,仅仅修改一个模块就行了。这些方法正好符合正逆双向交叉思维设计思想。

3 讨论

(1) 按 PARNAS 的原则设计整个系统时要充分考虑双向交叉思维的重点。

(2) 虽然 PARNAS 法思想特别先进,但要全面作到其要求是非常不易的,而且目前还未形成统一而成熟的体系,因而需我们进一步的探讨。

汽车零部件失效分析专家系统的研究

武汉工学院 (武汉 430070) 毕常青 杨开英 张明友

摘要 本文对专家系统技术在汽车零部件失效分析中的应用理论和技术进行了探索和研究。介绍了开发汽车零部件失效分析专家系统的设计思想、实现策略和方法;重点介绍了知识库的开发。

关键词 专家系统 失效分析 故障树

1 概述

专家系统技术是人工智能领域中最成熟的技术之一,许多成功的专家系统已取得明显的效益,并具有深远的影响。

汽车工业是一个综合性的产业,汽车本身结构的复杂性,汽车用材的多样性,汽车零部件服役条件的苛刻性,决定了汽车零部件不可避免的会出现早期失效,也决定了汽车零部件的失效实际是一个动态的复杂系统。由于涉及失效因素的广泛性、失效内容的交叉性,如何对大量观察、测试而获取的不完全精确的数据进行综合分析,从众多的潜在因素中找出导致失效的根本原因,继而有针对性的采取改进措施,是一件难度很大的工作。因此,建立失效分析专家系统辅助决策是必要的,具有现实意义的。

2 汽车零部件材料失效分析专家系统结构和功能

系统由推理机、知识获取和咨询解释等功能模块组成。其结构如图1所示。

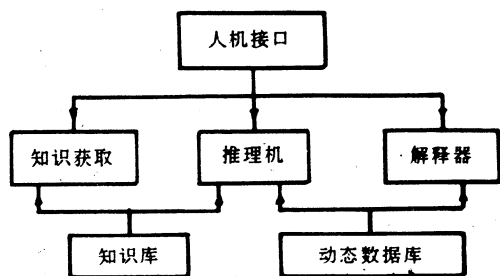


图1 汽车零部件失效分析专家系统结构图

知识获取模块的功能是:用于知识的编辑、编译及知识库的完整性检查、是系统和专家的界面。本系统中知识的获取采用两种方式:一是采用人机对话方式输入知识,直接转化为内部存储形式的知识,二是在编辑工具的支持下用限定方式输入知识,由编译器进行语法检查、分析生成内部形式的知识。对知识库的同化

和顺应操作依赖领域专家完成,系统可为专家提供已建立的知识模型的层次结构图和文本形式下的规则。

推理机:用于选择知识库的知识和动态数据库的事实匹配,控制系统的运行。

解释器:本系统在推理时实时显示被匹配的变量之值,而解释器的功能是显示推理的路径图。

知识库:是所存储的失效分析知识的总称,它由数据文件组成。动态数据库记录了用户咨询信息及推理过程中更新的事实和激活的规则,它既是推理机的工作空间,又是解释的依据。

3 开发原理

3.1 从何处开始——选取典型子集

图2反映了汽车故障诊断的目标轮廓。

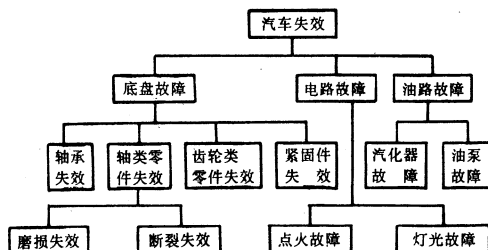


图2 汽车故障诊断的目标轮廓

汽车是一个复杂的机、电、油综合系统,因此,专家系统的开发不是一件容易的事情,其开发工作是渐近递增的过程。在成千上万个汽车零件中我们从失效模式及严重程度分析的角度出发,找出具有典型代表意义的子集,以汽车轴类零件的断裂失效作为系统的原型,确定其开发的起点。

3.2 开发的关键——知识模型的建立

模型是为了理解和描述事物而对事物所作的抽象,知识模型是失效分析中各种知识的抽象形式,这种形式能以精确的表示来实现,并能验证和逐渐加入细节以利于完善。

专家系统的质量在很大程度上取决于它存储的知识完备性、有效性、正确性和可维护性。其次,知识的表达和存储形式还应当有利于推理的检索、匹配。为

此,本专家系统采用了多层次的共性知识和启发性个性知识相结合的建模方法。

故障树分析法是失效分析中行之有效的经典方法。它将系统故障形成原因由总体至各个部分按树枝状逐渐细化,并以逻辑符号联结组成一棵或多棵能描述汽车系统性能故障的逻辑图。为了真实反映汽车零件失效的信息,对复杂的汽车零件的失效事件按不同层次的范畴(如失效类别集合、失效原因集合、失效属性集合、防止失效措施集合)建立若干棵故障树。一个故障树就是一个层次模型,层次模型中的结点对应于一组个性知识,该组知识体现了失效分析领域中长期积累的原理、经验,及行之有效的措施等启发性知识,每条个性知识用产生式规则表示,规则的前提表示引发事件,规则的结论表示引发的结果,规则与结论既可是进入下层结点的入口,又可激活另一棵故障树,从而体现了针对同一失效事件,从不同侧面所建立的故障树之间的联系。从图 2 汽车故障诊断目标轮廓中的轴类零件失效为例描述故障树(知识模型)的建立过程如图 3 所示。

从图 3 中我们可以看到在失效类别集合故障树层次模型中其根结点对应的一组规则可为:

$R_1: IF (n > 2)$

THEN 断裂失效

$R_2: IF (n < 2) \wedge \text{真实尺寸} > \text{原始尺寸}$

THEN 变形失效

$R_3: IF (n < 2) \wedge \text{真实尺寸} \approx \text{原始尺寸} \wedge \text{重量减少}$

THEN 表面损伤失效

其中: \wedge 表示前提事件之间是“与”关系。

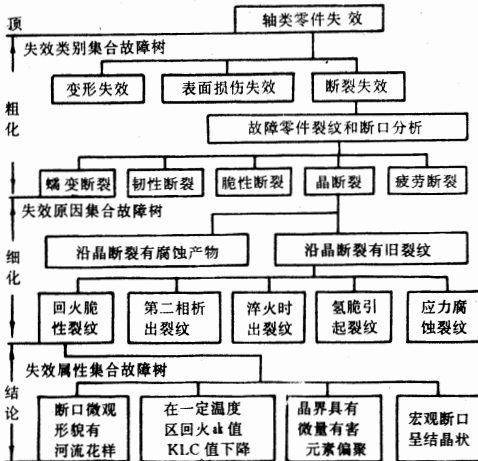


图 3 轴类零件失效故障树

位于上层树的叶结点虽然都进入下层细化的树,但所对应的子树不同,每棵子树中的结点所对应的规则组不同。

3.3 模仿专家工作——推理过程

选择和使用知识是人类专家能力的一个重要方

面,推理的任务是模拟人类专家控制推理的执行和选择推理方式。

采用故障树分析法建立汽车零件失效分析专家系统的知识库是以最小割集(MCS)为基础的,而最小割集为我们描绘出了处于故障状态的系统所必须要改进的基本故障,指出了系统中的最薄弱的环节。因此,在最小割集中人们以最有效的方式存储了有关系统状态和操作专家的经验,并对各种信息进行离线前处理系统的推理就是模仿人类专家从顶事件开始分析的过程,系统采用的是数据驱动式正向推理方法。基于产生式规则的正向推理的基本过程是从问题已有的事实(或设想的顶事件)出发,首先和根结点的规则进行检索匹配,遇到匹配的规则就更新动态数据库,并将其结论(中间事件)作为新的前提继续匹配,如果动态数据库提供的事实不足,将提示用户加入新的事实,如此循环往复,直到底事件为止。

其推理过程如图 4 所示:

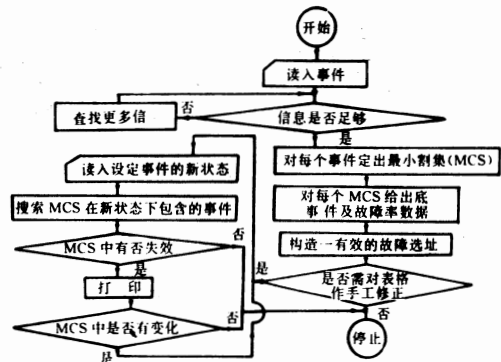


图 4 用于诊断的流程图

4. 实现、验证及改进

4.1 实现

本系统是用 C 语言在 IBMPC 及兼容机上实现的。知识的层次模型、规则、及规则的前提结论均可定义结构数据类型表示。

基于知识的组织和表示,推理过程是树的搜索和顺序检索的结合。

人机界面采用下拉和弹出式菜单方式,各级均有提示帮助信息,以方便操作使用。

4.2 验证

装入汽车轴类零件失效分析知识后,对实际案例进行了测试。结果表明其知识模型适应性好,表现能力较强,对原问题的分解可建立不同的知识库、扩充成实用系统。

由于知识模型的层次性,用户可以从模型的任一棵树的根结点启动推理,以适应具有不同经验水平的用户的需求。

4.3 改进

并行与分布式处理的错误调试

四川大学计算机系 (成都 610064) 袁道华 陶辅周

摘要 本文分析了并行与分布式处理系统可能发生的各种错误情况,通过一个具体的分布式调试系统讨论了并行与分布式处理的错误调试。

1 引言

分布式计算系统是一组通信和协作进程的集合,它们一起工作以获得一个共同的系统目标。这种目标可能提供给用户一个数据库管理系统或一个办公信息系统。术语分布式计算系统可以指系统硬件、系统软件或两者都包括。在这种情况下,系统进程分布在不同的相距较远的处理机上,这些处理机通过一个具有可共享二级存储器的特殊的网络配置松散地耦合在一起。并行处理指的是,一个程序中的各个进程并发地执行在一组紧密耦合的处理机上,这些处理机位于同一个系统中,具有可共享的主存储器。因此,并行和分布式处理都可能出现竞争情况(Race condition)、顺序错误(Sequencing bug)、和死锁等问题。

大多数程序员仍然使用传统的调试方法,如自底向上和两阶段技术(two-phase)。这些技术能够检测若干同步错误,然而在某些情况下,它们不能准确地定义错误的原因,这种缺陷是由于不恰当地管理监控下面的进程所获得的信息。本文将要介绍的分布式调试系统 DPDP 是一个源级的交互式调试工具,它适用于用 C 编程的并行和分布式应用的调试,使程序员能够交互式地监控、检测和修正一个系统。该系统能够检测包括运行时刻、逻辑上和所有同步故障在内的各种应用错误。

2 并行与分布式处理中的错误类型

本节通过四个例子,讨论了几种不同类型的同步错误。

2.1 竞争情况

程序中加锁原语的位置放置不当,导致不正确的同步进程,引起竞争情况的发生。这可以通过下面的例子来说明:

设有两个并行进程,它们分别具有常量值 15 和 20,另外文件 max __ file 中有一当前的最大值 10。现

在两个进程试图从其常量值和文件中的当前最大值间产生一个新的最大值,进程的执行如下:

```
Const __ value = { 15  进程 1
                  20  进程 2
step1: Open max __ file
step2: Read maximum __ value
step3: If (const __ value > maximum __ value) Do
        lock max __ file
        write const __ value
        unlock max __ file
        print the new maximum __ value
      Else
        close max __ file
      End
```

本例中,两个进程同时读取 max __ file。在进程 1 读取老的最大值 10 之后,加锁 max __ file 之前,进程 2 写入新值 20;在进程 2 解锁 max __ file 之后,进程 1 写入新的最大值 15,从而文件 max __ file 中具有不正确的最大值。为了修正这个例子,程序员需在两个进程中把加锁原语放在第 2 步之前。

2.2 顺序错误

当并行进程之间以不同于所期望的次序交互作用时,引起顺序错误。下面的例子包括 3 个进程,每个进程运行在网络的一个结点上(见图 1)。进程的执行如下:

```
进程 1: step1: 计算 X 值
        step2: 发送 X 值到进程 3
进程 2: step1: 计算 Y 值
        step2: 发送 Y 值到进程 3
进程 3: step1: 接收一封信用
        step2: 把收到的信存放在变量 X 中
        step3: 接收一封信用
        step4: 把收到的信存放在变量 Y 中
        step5: 计算 Z = X - Y, 输出 X、Y、Z
```

很明显,信件发送的次序可能引起顺序错误。如果进程 2 在进程 1 发送 X 之前发送 Y,进程 3 将收到具有 Y 值的第一封信,把它存在变量 X 中,然后接收第 2 封信,信中含有进程 1 传来的 X 值,把它存在变量 Y 中。由于在 X 值之前发送 Y 值不是进程 3 所期望的,

收稿日期:1994 年 3 月 20 日

目前,系统中对汽车零部件分类建立的知识库是互相独立的,因此,不能从整车性能水平上进行失效分析。但故障树分析法本身是可以适应复杂系统的,有待解决的问题是将各个知识库视为底事件,进一步合成到顶事件——整车失效。

其次,知识库中存储的是文本形式的知识,还不能直接表示对零件的微观测试所产生的图象信息中所包含的知识规则,这是一个更广阔的领域,有待于今后的扩充。

因此结果 $Z=X-Y$ 是不正确的。

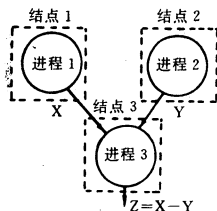


图1 顺序错误

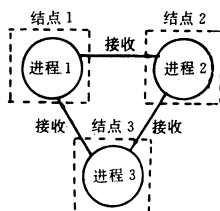


图2 通信死锁错误

2.3 资源死锁错误

这里我们给出一个例子说明在两个进程和两个资源之间的死锁情况。假定有两个进程，每个进程运行在应用网络的一个结点上，两个进程的算法如下：

进程 1:

```

step1: 打开文件 R1
step2: 加锁文件 R1
step3: 读 value1
step4: 打开文件 R2
step5: 加锁文件 R2
step6: 读 value2
step7: 计算 result=value1+value2
step8: 解锁 R1, 解锁 R2
step9: 关闭 R1, 关闭 R2
step10: 输出 result

```

进程 2:

```

step1: 打开文件 R2
step2: 加锁文件 R2
step3: 读 value2
step4: 打开文件 R1
step5: 加锁文件 R1
step6: 读 value1
step7: 计算 result=value1 * value2
step8: 解锁 R2, 解锁 R1
step9: 关闭 R2, 关闭 R1
step10: 返回结果

```

每个进程先分别加锁了一个资源，然后试图加锁已被对方封锁了的另一个资源（失败），这样任何一个进程都不能继续执行，从而产生了死锁（无限等待）。

2.4 通信死锁错误

图 2 给出了进程之间的通信死锁情况。下面的例子由三个进程组成，它们互相接收信件：

进程 1:

```

step1: 计算  $v=X1+X2$ 
step2: 接收一封信
step3: 把收到的信存在 d 中
step4: 计算  $z=d/v$ 
step5: 打印结果

```

进程 2:

```

step1: 计算  $r=X1 * X2$ 
step2: 接收一封信
step3: 把收到的信存在 V 中
step4: 计算  $z=r * v$ 
step5: 打印结果

```

进程 3:

```

step1: 计算  $d=X1/X2$ 
step2: 接收一封信
step3: 把收到的信存在 r 中
step4: 计算  $z=d * r$ 
step5: 打印结果

```

上面的例子中，进程 1 试图接收从进程 2 送来的值，同时进程 2 希望接收进程 3 送来的值，进程 3 试图接收进程 1 发来的信件，因此每个进程都进入了一个无限等待情况，从而产生了通信死锁。

3 DPDP 分布式调试系统

3.1 系统结构

分布式计算系统是一组相互协作的应用进程的集合，它们分布和运行在系统的若干结点上。DPDP 是一个分布式的调试系统，各调试模块以主—从（服务员—顾客）模式进行交互和通信。DPDP 由一个主控制器（master controller）和若干并发从调试器（Slave debugger）组成，图 3 是该系统的功能结构图。

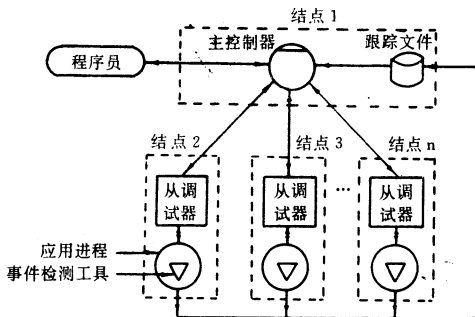


图3 DPDP 系统结构

主控制器管理和控制分布在不同网络结点上的从调试器的操作，主控制器可以位于任何结点上。每个从调试器监控正被调试系统的一个应用进程的执行。从调试器是交互作用的，并支持若干调试设施，其中包括跟踪（trace）、单步（single step）和设置/清除断点（set/clear breakpoint）等。另外，每个从调试器提供了一个工具，用来检测正被监控的进程中所有的重要事件，并把它们记录到跟踪文件（trace file）中去。为简便起见，检测和记录事件的工具被插入到应用进程里面。假定跟踪文件位于主控制器所在的结点，通过主控制器，程序员就能在一个交互式环境中以整体方式来调试一个分布式计算系统，获得执行期间的所有调试信息。因此，程序员能够准确地定位出进程中引起错误的确切行，并动态地通过跟踪文件检查进程间的交互作用。

3.2 “事件—动作”模型

DPDP 系统采用“事件—动作”（event—action）模型来描述并发系统。系统事件被分为下面几种类型：

- ① 进程启动（进程号），进程终止（进程号）
- ② 发送（源地，信息类型，目的地），

- 接收(源地,信息类型、目的地)
- ③文件打开(文件名),文件关闭(文件名)
- ④文件访问(文件名,读/写,项目)
- ⑤资源加锁(资源号),资源解锁(资源号)
- ⑥加锁失败(资源号)
- ⑦用户定义事件

事件可以被看作一个单一的“原子”,即它或者被执行或者不执行。事件是进程中一个重要的活动,它具有原子性。

如果几个进程并行执行,那么其结果必须与它们按某种次序串行执行的结果相同。因此每个进程中的事件,在执行期间应该是串行化的。

动作是跟事件相联系的,当执行期间识别到一个特定的行为时,动作被执行。动作可能涉及程序变量、调试器变量或输入、输出、中断这样的函数。

3.3 跟踪文件

在DPDP中,相关事件被记录在一个可共享的跟踪文件中,跟踪文件被用来检测各种同步错误。

通常,可以给每个事件附上一个时间,给事件附时间邮戳(time-stamp)对应于给事件分配一个数值,该数值由一个物理时钟确定,表示该事件发生的时间。跟踪文件可以被看作一个关系(表),关系的属性有:

- 源进程:启动一个事件的进程号
- 事件类型:代表检测的事件
- 目标进程:如果有接收者,表示将接收该信件的进程号,否则表示源进程
- 时间邮戳:唯一地标识一个事件发生的时间,时间邮戳提供了系统事件的一个全局定序关系。

因此,跟踪文件是目标程序中已经发生的事件所构成的文件,它描述了被调试程序的行为。下面是一个样本跟踪文件,它指出了一个死锁错误:

源进程	事件类型	目的进程	时间邮戳
i	打开文件 Ri	i	17:46:18.42
i	加锁 Ri	i	17:46:19.68
j	打开文件 Rj	j	17:46:19.45
j	加锁 Rj	j	17:46:20.78
i	打开文件 Rj	i	17:46:20.83
j	打开文件 Ri	j	17:46:20.95
i	加锁失败 Rj	i	17:46:21.44
j	加锁失败 Ri	j	17:46:21.71

3.4 主控制器

主控制器是一个控制进程,它管理和控制一组并发调试设施,这些设施包括跟踪、单步、浏览事件

(View event)、设置/清除断点。在任何一个设施获取控制之前,各应用进程必须在相应的各结点上启动。主命令例程接收程序员的命令输入,转入相应的调试设施执行,在每个设施成功地执行之后,控制返回到主命令例程,以选择另一个调试设施或从系统退出。下面是主控制器的C程序框架:

```
main()
{
    Initialization(); /* 初始化 */
    for(;;){
        gets(programmer command); /* 接收程序员命令 */
        if(command=='T') trace(); /* 跟踪控制器 */
        if(command=='S') single_step(); /* 单步控制器 */
        if(command=='V') view(); /* 浏览事件 */
        if(command=='sbp') set_break(); /* 设置断点 */
        if(command=='cbp') clear_break(); /* 清除断点 */
        if(command=='C') continue(); /* 继续例程 */
        if(command=='Q') quit(); /* 退出系统 */
    }
}

Initialization()
{
    Input number of application processes;
    Input processes names or id;
    Initialize master controller files;
    Initialize master and slaves windows;
}
```

(1)跟踪控制器

为了启动对所有应用进程的跟踪,跟踪控制器向所有从调试器发送跟踪信件,告诉它们启动对应用进程的跟踪,然后等待接收从调试器发来的信息。这些信息在执行完源代码的每一行后被发送,它们有:

• 错误信息:<错误类型,进程号i,行号j>表示进程i在第j行发生了某种类型的错误。一旦跟踪控制器接收到了从任何从调试器发来的错误信息时,即停止所有从调试器的操作,这通过设置一个公共的挂起标志来实现。在执行了源代码的每一行后,所有的从调试器读取该标志。

• 断点信息:<Break,进程号i,行号j>表示监控进程i的从调试器在第j行检测到一个断点,当接收到该信息时,跟踪控制器挂起所有从调试器的操作,控制返回到主命令例程。

• PTN信息(进程正常终止):<PTN,进程号i>通知主控制器进程i已经完成并正常终止。

• Done信息:<Done,进程号i,行号j>表示进程i的第j行已经成功地被执行。当接收到该信息时,跟踪控制器修改监控进程i的从调试器所对应的窗口中的指针位置,即将指针移到第j+1行。

(2)单步控制器

除了跟踪外,程序员可能希望一次执行应用进程的一条指令(行)。单步控制器向所有从调试器发出单

步信件,告诉它们执行每个进程的一行,然后等待接收从每个从调试器发来的一个信息,该信息可能是错误信息、PTN 信息或 Done 信息。之后,单步控制器返回控制到主命令例程。

(3) 浏览事件

在执行期间的任何时候程序员都可以浏览和检查应用进程的跟踪文件。浏览事件例程在一个单独的窗口显示跟踪文件。

(4) 设置/清除断点

为了在任何应用进程中设置断点,设置断点例程需要通知适当的从调试器对某个特定的行设置一个断点标志。类似,为了清除一个断点,清除断点例程发送一个清除断点信件以清除断点标志。

(5) 继续例程

继续例程的目的是恢复一个断点后的操作,因此,它复位挂起标志并调用跟踪控制器。

3.5 从调试器

从调试器是执行和监控一个应用进程的一组调试设施的集合。首先,从调试器初始化断点表,应用进程的每一行在断点表中存在一个记录<行号,标志>。然后从调试器启动错误检测模块,检查所有的运行错误,如溢出、被零除等。如果存在任何错误,从调试器即向主控制器发送错误信息。当检测到进程结束时,则向主控制器发送一个 PTN 信息。在源代码的每一行被执行后,应用进程调用从调试器。下面是从调试器的 C 程序框架:

```
main()
{
    if(Initialization flag != 1){
        Initialization(); /* 初始化 */
        Initialization flag=1;
        slave_com(); /* 从命令例程 */
    }
    if("Error"){
        send Error message to master controller;
        slave_com();
    }
    if (End of process){
        send "PTN" message to master controller;
        slave_com();
    }
    if(last command=="trace"){
        send "Done" message to master controller;
        trace();
    }
    if (last command=="single step"){
        send "Done" message to master controller;
```

```
slave_com();
    }
}
slave_com()
{
    Receive a message; /* 接收来自主控制器的命令 */
    store the received command in last command;
    if (command=="trace") trace();
    if (command=="single step") single_step();
    if (command=="set break") set_break();
    if (command=="clear break") clear_break();
    if (command=="quit") exit();
}
Initialization()
{
    Count process lines;
    Allocate a memory for break-points table;
    Initialize Errors detecting module;
}
```

3.6 用户界面

DPDP 用户界面使用多个窗口和屏幕页,该界面包含下面的窗口:

(1)主控制器窗口:该窗口位于屏幕的底部,用来显示程序员命令或任何可能发生的错误信息。

(2)从调试器窗口(多个):这些叠加的跟踪窗口用来显示执行期间各应用进程未被修改的源代码。每个窗口有一指针用来指示当前执行的行。

(3)工作区窗口:该窗口位于一个单独的屏幕页中,用来显示应用进程的输出。

(4)跟踪文件窗口:该窗口也是位于一个单独的屏幕页中,用来显示跟踪文件的内容。

参考文献

- 1 P. H. Enslow, What Is a Distributed Data Processing System? IEEE Comp. (1978)
- 2 H. G. Molina, F. Hermano, and W. H. Kohler, Debugging a Distributed Computing System, IEEE Trans. Software Eng. SE-10(1984)
- 3 J. Gait, A Debugger for Concurrent Programs, Software Pract. Exp. (1985)
- 4 D. C. Marinescu, J. E. Lumpp, and T. L. Casavant, Models for Monitoring and Debugging Tools for Parallel and Distributed Software, J. Par. Dist. Comp. 9, 171-148(1990)
- 5 M. Zaki, etc., DPDP: An Interactive Debugger for Parallel and Distributed Processing, The Journal of Systems and Software, Vol. 22, No. 1, 1993. 7

本刊启事

1. 原预计 94 年 8 月份出版的《计算机故障维修 1000 例》因故改期出版,请读者勿再寄预订书款,可只报预订册数。
2. 本刊人员分工如下:张钢主管广告通联、技术资料专著(译)出版;唐大利主管来稿登录编号、作者读者服务;邓傢文主管期刊发行(含报刊交换)、出版档案;哲明负责来稿初审、综合协调、纵横沟通。有关业务,请按照分工,对口联系,以便及时处理。
3. 今后作者来稿,请务必注明作者姓名、所在单位、通讯地址、邮政编码、中英文标题、摘要、关键词、参考文献、收稿人姓名,以便及时妥善处理。
4. 经正式批准,本刊为中国计算机学会会刊,特再次公告。

《计算机应用研究》编辑部

群集 UNIX 环境下的 ORACLE 并行服务器技术

四川大学计算机系 (成都 610064) 李 强 陶辅周

摘要 本文介绍了 ORACLE 的并行服务器技术如何运用于类似 VAXcluster 这样的松耦合多处理机环境,使其具有可调节性和高度的可用性。使用这项技术,ORACLE 能在一个群集环境的各节点上并行运行同一系统映象。同时文章还讨论了为什么这项技术可以很快运用于 UNIX 环境。

引言

随着数据库应用的发展,今后几年,大幅度提高数据库服务器性能的呼声将会更大。目前,性能的提高往往是通过用更强大的系统替代现有系统来实现的。但这种方法从性价比上看有明显的局限:用更强大的机器来替代现有机器是昂贵的,因为随 MIPS 数的增长,每 MIPS 的开销也在增加。另外,无论一台计算机有多强大,它也很难满足日益增长的数据库服务器的性能需求。松耦合结构是目前计算机硬件中最新提出的一种计算机结构。在松耦合系统中,虽然诸多机器各不相同,但它们被连在一起且能直接访问所有数据。ORACLE 并行服务器技术利用了这种计算机结构的优点,充分挖掘了松耦合系统的潜力,使性价比达到前所未有的水平。

松耦合系统

有两种方法可以提高对一组数据的处理能力:一是提高 CPU 能力;二是增加 CPU 数目。前者优势在于它一般不需要对现有系统结构做任何改变;其不足在于费用方面,随 CPU 的 MIPS 数增长,每个 MIPS 的花费也要增加。后者的优势在于它不需要引入昂贵的新 CPU,但它需要一些额外的工作将各个 CPU 连接起来,使它们协调工作。

连接 CPU 的方法有三种:第一是使用一般的网络,这样连起来的 CPU 仅访问它自己的数据和内存(RAM),用一般的通讯与其他 CPU 联系。第二种是紧耦合环境,这样连起来的 CPU 将能访问同一组数据,且它们使用同一个内存,即内存由 CPU 共享。第三种是松耦合环境。这样连起来的 CPU 虽然仍能访问同一组数据,但它们的内存是独立的,非共享的。

对于简单网络,由于每个 CPU 都有它们自己独立的数据,因此需要数据库有分布功能。ORACLE 并行服务器重点不在分布式数据库上,因此本文不讨论分布式数据库的问题。

在紧耦合环境中,由于所有 CPU 都直接访问同一组数据,所以只需一个数据库就足够了。但由于各 CPU 的协调方式,一个紧耦合系统仅能将很有限的

CPU 连接起来。在紧耦合系统中,各 CPU 是通过共享内存来相互协调的。当一个 CPU 要修改一个数据项时,它首先在共享内存中放入必要的信息以防其它 CPU 在同一时刻也来修改该数据项。当许多 CPU 同时都想修改某一组数据时,从各 CPU 到共享内存的信息流就会增加。这些信息流要通过内存总线传送,而内存总线往往具有固定带宽。要提高内存总线带宽是非常昂贵的,这和提高单个 CPU 能力一样,会产生性价比不佳的问题,因而这意味着可同时访问共享内存的 CPU 数目是有限的。

与紧耦合系统一样,在松耦合系统中所有 CPU 都直接访问同一组数据,由于各 CPU 并不靠共享内存来通讯,因此大多数松耦合系统的通讯通道的速度和吞吐量都是相当高的。尽管与紧耦合系统的内存总线相比,这些通讯线路的效率要低些,但大多数松耦合系统能做到少用通讯线。这样松耦合系统可以比紧耦合多连接更多的 CPU,而且每个 MIPS 的费用要低得多。为提高松耦合系统的能力,不必将 CPU 以新换旧,而仅仅增加 CPU 个数就可以了。

由于松耦合系统的通讯性能不高,实现一个保证所有 CPU 都要访问单一数据库的 DBMS 是困难的。ORACLE 并行服务器可以使松耦合系统中各 CPU 间的通讯要求减至最小,并使该系统具有可调节的性能。

ORACLE 并行服务器

在介绍 ORACLE 并行服务器之前,让我们回顾一下 ORACLE 数据库在单 CPU 和紧耦合系统上是怎样提高性能的。

实现一个最简单的 RDBMS 的方法是:当需要数据时,将数据从盘上读入内存;一旦该数据不再有用就立即写回盘中。因此,每个访问数据库的用户进程都用它自己的私用内存来保存它正操纵的数据。此方法的优点是简单,易于实现,但其性能极大地受限于读写盘的 I/O 数量。

提高性能的一种办法是用共享内存来缓存最近访问的数据。在这类数据库系统中,当一用户进程访问数据时,它首先将数据读入位于计算机主存中的共享高速数据缓存中。当用户完成对该数据的修改后,将之写回磁盘,不过仍将该数据的备份留在共享高速缓存中

以便其他用户可以不经过再次读入 I/O 操作就可以访问该数据。高速缓存是所有用户共享的,这可以减少不必要的读盘 I/O 数量。不过该方法仅能减少读盘次数却并不减少代价昂贵的写盘次数,因为所有的修改都要立即写盘。ORACLE V5 用的就是这种方法。

为了铲除这最后一个瓶颈,ORACLE V6 提供了“快速提交”和“延迟写”功能。这允许数据库系统在每次数据更改后并不立即写盘,而是将该数据留在共享高速缓存中,这样,必须用一个单独的 REDO 日志来记录用户对数据库所做的各种修改,并且,写日志的操作是被集中后成批执行的,它有更高的效率。快速提交是指写日志允许的某一事务提交,“延迟写”是指被修改数据的写盘操作被延迟到整个事务提交时才进行。

但是要将上面特点引入到松耦合系统中并非一件容易的事。一个数据库系统性能的好坏取决于它是否能将常用数据放在高速缓存中以减少 I/O 次数。在松耦合系统中,每个 CPU 都有它自己的内存,且各 CPU 间没有任何可被共享的内存。那么我们在实现一个数据库系统时如何才能摆脱这明显的障碍呢?

ORACLE 并行服务器由于使用了分布式高速缓存技术从而突破了单机上数据库性能不佳的障碍。运行 ORACLE 并行服务器时,松耦合系统中每台 CPU 都有其自己的高速缓存。这些高速缓存中的数据都来自盘上同一个数据库。现在的问题是如何实现快速提交和延迟写以便 ORACLE V6.0 能用分布高速缓存保证高的性能,且不使松耦合方式连接的各 CPU 之间通讯线太繁忙。ORACLE 并行服务器通过创造性使用一个被称作“分布封锁管理器 DLM”的单独软件产品来解决上述问题。DLM 是一个通用的软件程序,许多松耦合系统都用它来协调各 CPU 间的工作。DLM 是用来管理分布锁的,它通过发送结由通讯线的消息来确保各 CPU 不会并发地以不相容的方式使用同一分布锁。

ORACLE 并行服务器使用 DLM 来传送这样的信息:每个高速缓存有权访问哪些数据。当一台 CPU 上的 ORACLE 要访问数据库的某一子集时,它首先获得一个或多个 DLM 锁来宣布它正在访问该子集。其它 CPU 上的 ORACLE 将不允许去访问这一子集,除非它获得了该 DLM 锁。

需要指出的是:DLM 锁与行级锁是两个不同的应用程序。行级锁由 ORACLE RDBMS 内部管理,用来协调不同用户对数据的访问。当某一用户对数据库中的某些行进行访问和修改时,它仅同行级锁来保证其它用户不会并发地来修改这些行。而 ORACLE 用 DLM 锁来协调各高速缓存的工作。

还需指出,DLM 锁是由各高速缓存特有的。也就是说当一个用户访问完他所需的数据后,DLM 锁并不被释放,直到其它高速缓存明确提出要其中的某些

DLM 锁之前,该高速缓存一直持有这些 DLM 锁。有了 DLM 锁,ORACLE 并行服务器能通过使用快速提交和延迟写技术保证 I/O 次数是最少的,并使其能充分利用松耦合系统中各 CPU 来完成数据库工作。

ORACLE 并行服务器的特点

与单机数据库系统相比,ORACLE 并行服务器具有更多的特点:

- 比任何单机系统的性能高得多
- 具有可调节的性能
- 高可用性
- 数据库管理更方便
- 应用开发和调度更方便

高性能

ORACLE 并行服务器比其它 RDBMS 有更多的性能。在一个具有四台 VAX6000/560 的 VAXcluster 环境上,进行具有工业标准的 TPC-B 基准测试,结果表明 ORACLE 并行服务器达到 425.7tpsB。在一个拥有 64 个 CPU 的松耦合环境 nCUBE 上,ORACLE 突破了 1000tpsB 大关。

在当今市场上还没有哪个 RDBMS 能达到上述性能指标中的任何一项,而且在不远的将来,ORACLE 并行服务器肯定会更出色。

另外,ORACLE 并行服务器的性价比也是很突出的。在 VAXcluster 上,每 tpsB 花费仅在 17K 美元至 18K 美元之间。而需在传统大型机上进行类似的测试,其每 tpsB 的花费要高得多。在 nCUBE 上,每 tpsB 的花费仅在 2K 美元至 3K 美元之间。

可调节的性能

ORACLE 并行服务器的性能不仅很高,而且是可调节的。只要在松耦合系统中加入新的 CPU,ORACLE 并行服务器就可以充分利用它。也就是说,当 ORACLE 并行服务器达到一个松耦合系统的性能极限时,所需做的仅仅是向系统中添入新的 CPU,那么性能上的提高立即就会表现出来。这种方法比将现有 CPU 更换为更大、更快或更贵的作法要便宜得多。

高可用性

ORACLE 并行服务器提供了高可用性的优势在一个松耦合系统中,若有一台或多台 CPU 由于某种原因不再可用时,ORACLE 并行服务器将保证整个系统能继续运转。比如,假设 ORACLE 并行服务器工作于四台 VAX 系列机的 VAXcluster 上,突然有一台 VAX 机出了故障不再可用,这时 ORACLE 并行服务器将自动地完成必要的恢复工作,且通过群集环境中的其它三个可用节点来保证对数据库的任何访问工作不受任何影响,即使系统中只剩下了一个可用节点,ORACLE 并行服务器仍保证对数据库的任何访问不受任何影响。也就是说,ORACLE 并行服务器能自动地、透明地完成硬件故障的保护工作。(下转 23 页)

ORACLE 中 SQL 查询语句的优化

四川大学计算机系 (成都 610064) 李 强 陶辅周

摘 要 本文概述了 ORACLE 中 SQL 查询语句的优化策略,包括索引/谓词的记分(ranking)和驱动表(driving table)的选择。通过列举一系列从简单查询到复杂查询的例子,详细说明了如何将 ORACLE 的查询优化策略运用到数据库应用设计中,从而提高数据库性能。所举例子中有些是为说明某一问题而专门设计的,有些是 ORACLE 数据库应用设计过程中所遇到的实际案例。

1 引 言

从第 5 版开始,ORACLE 采用了人工智能查询优化技术来改进查询性能。它使系统始终具有确定最佳存取路径的能力。并充分利用所有可用的索引和聚簇实现其优化。

ORACLE 查询性能的调整要依据如下经验准则:

- 借用 ROWID 的查询效率最高;
- 借用唯一索引的查询要快于借用非唯一索引的查询;
- 借用完整说明索引的查询要快于借用部分说明索引的查询;
- 等条件查询快于范围条件或不等条件的查询;
- IS NULL 条件的查询不能使用索引;
- 尽量避免查询中出现全表的扫描。

我们可以用 ORACLE 的 EXPLAIN 工具,求出每条查询语句的查询路径记分表,根据该表并结合上面的经验准则来优化 SQL 查询语句,提高数据库的查询速度。

同时我们还可以使用一种画查询图的技术来优先连接查询。连接查询是相当重要的一类查询,同时也是极难优化的一类查询。画查询图的第一步是把查询中涉及的表按其出现在 FROM 子句后的次序,依次列在一页纸上,然后,依据 WHERE 子句中的连接条件把各表用线连接起来。同时依据 WHERE 子句中形如〈列名〉〈比较符〉〈常数〉的表达式,将表与常数用线连起来。最后,在每条线上标出一个索引类型,该索引类型代表了优化程序所用来的存取的最佳索引类型。

索引类型用如下的缩略符号表示:

S 或 X——单属性索引或多属性索引(组合索引);

U 或 R——唯一索引或非唯一索引;

C 或 P——完整说明索引或部分说明索引。

根据上述经验准则,SUC 索引(即单属性、唯一、完整说明索引)的查询效率最高,XRP 索引(即多属性、非唯一、部分说明的索引)的查询效率最低。

下面我们列举一系列的例子来说明如何采用 EXPLAIN 工具和画查询图技术并依据优化准则来优化

SQL 查询语句,从而提高数据库查询效率。

2 数据库中所用的表和索引

下面的语句用于建立我们所要用到的测试中的表和索引。

```
CREATE TABLE EMP(  
    EMPNO NUMBER(4) NOT NULL,  
    ENAME CHAR(10),  
    JOB CHAR(9),  
    MGRNO NUMBER(4),  
    HIREDATE DATE,  
    SAL NUMBER(7,2),  
    COMM NUMBER(7,2),  
    DEPTNO CHAR(4));  
  
CREATE UNIQUE INDEX EMP _ U _ EMPNO ON  
EMP(EMPNO);  
  
CREATE INDEX EMP _ MGRNO ON EMP  
(MGRNO);  
  
CREATE INDEX EMP _ DEPTNO ON EMP  
(DEPTNO);  
  
CREATE TABLE DEPT(  
    DEPTNO CHAR(4) NOT NULL,  
    DNAME CHAR(14),  
    LOC CHAR(13));  
  
CREATE UNIQUE INDEX DEPT _ U _ DEPTNO ON  
DEPT(DEPTNO);  
  
CREATE TABLE SKILLS(  
    SKILL CHAR(3),  
    SKILL _ DESCR CHAR(20));  
  
CREATE UNIQUE INDEX SKILLS _ U _ SKILL ON  
SKILLS(SKILL);  
  
CREATE TABLE EMP _ SKILLS(  
    EMPNO NUMBER(4) NOT NULL,  
    SKILL CHAR(3) NOT NULL);  
  
CREATE UNIQUE INDEX EMP _ SKILLS _ U _ EMPNO _  
SKILL ON EMP _ SKILLS(EMPNO,SKILL);
```

3 简单查询

3.1 最简单的情况

请看下面的单表查询,其 WHERE 子句只有一个谓词:

```
SQL>EXPLAIN PLAN SET STATEMENT _ ID='E1'  
FOR
```

```
SELECT * FROM EMP
WHERE DEPTNO=10;
```

```
SQL>@XPLAIN E1
```

```
QUERY PLAN
1.0 TABLE ACCESS FULL EMP
```

查看一下表和索引的建立,知道在 EMP 表的 DEPTNO 上确有索引 EMP_DEPTNO。但是为什么此索引未被派上用场呢?原因在于上述查询中隐含着数据类型的转换。DEPTNO 列定义为字符型,但在 WHERE 子句中却让其与一数值匹配,ORACLE 的处理是把等号左边的列转换为数字型,由此使得 EMP_DEPTNO 索引不能被使用。

若把上述 WHERE 子句改写如下,则索引即能被利用上。

```
SQL>EXPLAIN PLAN SET STATEMENT _ID='E2'
FOR
```

```
SELECT * FROM EMP
WHERE DEPTNO='E10'
```

```
SQL>@XPLAIN E2
```

```
QUERY PLAN
1.0 TABLE ACCESS BY ROWID EMP
2.1 INDEX RANGE SCAN EMP_DEPTNO NON_UNIQUE
```

此时查询计划的选择是,先扫描非唯一索引 EMP_DEPTNO,然后由 ROWID 存取表。

3.2 等值与不等值的查询

我们知道 ORACLE 能够为 WHERE 子句中的等值条件使用索引,但对不等值条件(如大于、小于、不等于)会是怎样呢?下面给出三个例子说明:

```
SQL>EXPLAIN PLAN SET STATEMENT _ID='E3'
FOR
```

```
SELECT * FROM EMP
WHERE DEPTNO! ='D10';
```

```
SQL>@XPLAIN E3
```

```
QUERY PLAN
1.0 TABLE ACCESS FULL EMP
```

```
SQL>EXPLAIN PLAN SET STATEMENT _ID='E4'
FOR
```

```
SELECT * FROM EMP
WHERE DEPTNO<>'D10';
```

```
SQL>@XPLAIN E4
```

```
QUERY PLAN
1.0 TABLE ACCESS FULL EMP
```

```
SQL>EXPLAIN PLAN SET STATEMENT _ID='E5'
FOR
```

```
SELECT * FROM EMP
WHERE DEPTNO>'D10' OR DEPTNO<'D10';
```

```
SQL>@XPLAIN E5
```

```
QUERY PLAN
1.0 CONCATENATION
2.1 TABLE ACCESS BY ROWID EMP
3.1 INDEX RANGE SCAN EMP_DEPTNO NON U-
```

NIQUE

2.2 TABLE ACCESS BY ROWID EMP

3.1 INDEX RANGE SCAN EMP_DEPTNO NON_UNIQUE

上述三例的查询结果相同,但却有不同的查询计划。头两例中,ORACLE 对不等值条件未使用索引。这是因为优化程序假设该查询结果将占该表总行数的 10—15%,故使用索引并不合算。但若实际情况并非如此,希望能强制使用索引时,则可以把查询改写为第三例的形式,在第三例的查询计划中,存取了 EMP_DEPTNO 索引两次,分别为 WHERE 子句的两个谓词所用。这样查询结果即为两次存取的并 (CONCATENATION)。

3.3 简单连接

简单连接是很常见的一种查询形式,它是指连接条件为两属性值相等的两表连接。例如,在测试库中,若要查出有关所有雇员的信息包括全部部门信息如部门名称及地址,则可有如下的简单连接:

```
SQL>EXPLAIN PLAN SET STATEMENT _ID='E6'
FOR
```

```
SELECT * FROM EMP,DEPT,
WHERE EMP.DEPTNO=DEPT.DEPTNO;
```

```
SQL>@XPLAIN E6
```

```
QUERY PLAN
1.0 NESTED LOOPS
2.1 TABLE ACCESS FULL DEPT
2.2 TABLE ACCESS BY ROWID EMP
3.1 INDEX RANGE SCAN EMP_DEPTNO NON_UNIQUE
```

Query Diagram
SRC SUC

EMP.....DEPT

这里优化程序选择了 DEPT 表作为驱动表。这是由于 DEPT 列在 FROM 子句的最右端,而且对 EMP 表可用索引来存取。对这一查询几乎没什么改进的余地,因为 DEPT 表是一个较小的表,而且是我们要选作全表扫描的表。

若上述查询不是针对所有雇员的,而是针对某部门内的雇员的,则连接情况会是怎样?对新的查询要求可以给出多种查询形式,因此便存在一个谁优谁劣的问题,首先看如下的近乎冗余的查询语句:

```
SQL>EXPLAIN PLAN SET STATEMENT _ID='E7'
FOR
```

```
SELECT * FROM EMP,DEPT
WHERE EMP.DEPTNO=DEPT.DEPTNO
AND EMP.DEPTNO='D10'
AND DEPT.DEPTNO='D10';
```

```
SQL>@XPLAIN E7
```

```
QUERY PLAN
1.0 NESTED LOOPS
2.1 TABLE ACCESS BY ROWID DEPT
```

```
3.1 INDEX UNIQUE SCAN DEPT _U _DEPTNO U-
NIQUE
```

```
2.2 TABLE ACCESS BY ROWID EMP
```

```
3.1 INDEX RANGE SCAN EMP _DEPTNO NON _U-
NIQUE
```

QUERY Diagram

```

SRC      SUC
EMP.....DEPT
: SRC      : SUC
:          :
: 'D10'    : 'D10'
```

此查询不管如何执行,都有可用的索引路径。此处优化程序选择 DEPT 作为驱动表,并利用唯一索引 DEPT _U _DEPTNO 存取该表。这或是因为 DEPT 表处于 FROM 子句的最右端,或是因为在查询路径记分表中唯一索引优于非唯一索引。下面改动 FROM 子句中表的次序:

```
SQL>EXPLAIN PLAN SET STATEMENT _ID='E8'
FOR
```

```

SELECT * FROM DEPT,EMP
WHERE EMP.DEPTNO=DEPT.DEPTNO
AND EMP.DEPTNO='D10'
AND DEPT.DEPTNO='D10';
```

```
SQL>@XPLAIN E8
```

```

QUERY PLAN
1.0 NESTED LOOPS
```

```
2.1 TABLE ACCESS BY ROWID DEPT
```

```
3.1 INDEX UNIQUE SCAN DEPT _U _DEPTNO U-
NIQUE
```

```
2.2 TABLE ACCESS BY ROWID EMP
```

```
3.1 INDEX RANGE SCAN EMP _DEPTNO NON _U-
NIQUE
```

可见此时优化程序仍选 DEPT 表作为驱动表,同时可以说明这一选择是基于唯一索引优于非唯一索引。

现在若对上述的冗余查询,从 WHERE 子句中去掉一些不必要的条件,又会出现什么情况?

```
SQL>EXPLAIN PLAN SET STATEMENT _ID='E9'
FOR
```

```

SELECT * FROM EMP,DEPT
WHERE EMP.DEPTNO=DEPT.DEPTNO
AND EMP.DEPTNO='D10'
```

```
SQL>@XPLAIN E9
```

```

QUERY PLAN
1.0 NESTED LOOPS
```

```
2.1 TABLE ACCESS BY ROWID EMP
```

```
3.1 INDEX RANGE SCAN EMP _DEPTNO NON _U-
NIQUE
```

```
2.2 TABLE ACCESS BY ROWID DEPT
```

```
3.1 INDEX UNIQUE SCAN DEPT _U _DEPTNO U-
NIQUE
```

QUERY Diagram

```

SRC      SUC
```

```
EMP.....DEPT
```

```

: SRC
: 'D10'
```

这里优化程序选择 EMP 作为驱动表,并由非唯一索引 EMP _DEPTNO 存取该表。

消除查询中冗余条件的另一途径是将 EMP.DEPTNO=DEPT.DEPTNO 条件去掉,而只保留两个 DEPTNO='D10' 条件。

```
SQL>EXPLAIN PLAN SET STATEMENT _ID='
E10'FOR
```

```

SELECT * FROM EMP,DEPT
WHERE DEPT.DEPTNO='D10'
AND EMP.DEPTNO='D0'
```

```
SQL>@XPLAIN E10
```

```

QUERY PLAN
1.0 NESTED LOOPS
```

```
2.1 TABLE ACCESS BY ROWID DEPT
```

```
3.1 INDEX UNIQUE SCAN DEPT _U _DEPTNO U-
NIQUE
```

```
2.2 TABLE ACCESS BY ROWID EMP
```

```
3.1 INDEX RANGE SCAN EMP _DEPTNO NON _UNIQUE
```

QUERY Diagram

```

EMP      DEPT
: SRC      : SUC
: 'D10'    : 'D10'
```

虽然这里的两表没有明显的连接,但所使用的索引路径与开始的冗余查询中的路径相同。从以上几种对等连接的改写方案中看出,形如(列名)二常数的条件是与由优化程序选择查询计划有关的。

4 复杂查询

4.1 外连接

外连接限制优化程序的选择。任何情况下带外连接符(+)的表不能用作驱动表。下面是一个使用了外连接的查询,它将返回所有雇员的信息,不管其部门号是有效还是无效,也不管其值有还是没有。

```
SQL>EXPLAIN PLAN SET STATEMENT _ID='
E11'FOR
```

```

SELECT * FROM EMP,DEPT
WHERE EMP.DEPTNO=DEPT.DEPTNO(+);
```

```
SQL>@XPLAIN E11
```

```

QUERY PLAN
1.0 NESTED LOOPS OUTER
```

```
2.1 TABLE ACCESS FULL EMP
```

```
2.2 TABLE ACCESS BY ROWID DEPT
```

```
3.1 INDEX UNIQUE SCAN DEPT _U _DEPTNO U-
NIQUE
```

这里优化程序只能选 EMP 作为驱动表,并对之作全表扫描。这一查询没有可改进的余地。

4.2 三表连接

下述的查询将给出所有雇员及其技能的清单,其中包括雇员姓名和技能描述。由于我们给出的测试库

是规范化的,因此这需要在表 EMP EMP_SKILLS 和 SKILLS 之间作三表连接。

```
SQL> EXPLAIN PLAN SET STATEMENT _ID = '
E12' FOR
```

```
SELECT EMP.EMPNO,ENAME,DEPTNO,EMP
_SKILLS,SKILL,SKILL_DESCP
FROM EMP,EMP_SKILLS,SKILLS
WHERE EMP.EMPNO=EMP_SKILLS.EMPNO
AND EMP_SKILLS.SKILL=SKILLS.SKILL;
```

```
SQL> @XPLAIN E12
```

```
QUERY PLAN
```

```
1.0 NESTED LOOPS
```

```
2.1 NESTED LOOPS
```

```
3.1 TABLE ACCESS FULL EMP_SKILLS
```

```
3.2 TABLE ACCESS BY ROWID SKILLS
```

```
4.1 INDEX UNIQUE SCAN SKILLS_U_SKILL U-
NIQUE
```

```
2.2 TABLE ACCESS BY ROWID EMP
```

```
3.1 INDEX UNIQUE SCAN EMP_U_EMPNO U-
NIQUE
```

```
QUERY Diagram
```

```
SUC XUP SUC
```

```
EMP.....EMP_SKILLS.....SKILLS
```

这里优化程序选择了 EMP_SKILLS 表作为驱动表。这是很自然的事情,因为此时其它两表的存取都可利用其相应的索引路径。查询机制中,EMP_SKILLS 表要作全表扫描。对每一返回行通过 SKILLS_U_SKILL 索引从表中 SKILLS 取得技能描述信息,再通过 EMP_U_EMPNO 索引从 EMP 表中取得雇员信息。

对这一查询,可以进行一定的改进。首先 EMP_SKILLS 表的行可能多于 EMP 表,因此需要交换 FROM 子句中表的次序。其次,若仔细观察,会发现 EMP_SKILLS 表存在一个我们所要存取的两属性的组合索引 EMP_SKILLS_U_EMPNO_SKILL。若优化程序使用这一索引,则避免再去存取表 EMP_SKILLS。

下面是简单地交换查询中 FROM 子句中 EMP 表和 EMP_SKILL 表的次序后所产生的结果:

```
SQL> EXPLAIN PLAN SET STATEMENT _ID = '
E13' FOR
```

```
SELECT EMP,EMPNO,ENAME,DEPTNO,EMP
_SKILLS,SKILL,SKILL_DESC
FROM EMP_SKILLS,EMP,SKILLS
WHERE EMP.EMPNO=EMP_SKILLS.EMPNO
AND EMP_SKILLS.SKILL=SKILLS.SKILL;
```

```
SQL> @XPLAIN E13
```

```
QUERY PLAN
```

```
1.0 NESTED LOOPS
```

```
2.1 NESTED LOOPS
```

```
3.1 TABLE ACCESS FULL EMP
```

```
3.2 INDEX RANGE SCAN EMP_SKILLS_U_EMP-
NO_SKILL UNIQUE
```

```
2.2 TABLE ACCESS BY ROWID SKILLS
```

```
3.1 INDEX UNIQUE SCAN SKILLS_U_SKILL U-
NIQUE
```

此时 EMP 表作为驱动表作全表扫描。这是比较好的选择。另外 EMP_SKILLS 表不再被存取,而代之以使用 EMP_SKILLS_U_EMPNO_SKILL 索引。

4.3 WHERE 子句的追加和性能改善

对上面的例子,要使优化程序既不必存取 EMP_SKILLS 表,又可以用上 EMP_DEPTNO 索引而避免对 EMP 表作全表扫描,我们可以通过采用在 WHERE 子句中追加新的条件表达式来实现。

```
SQL> EXPLAIN PLAN SET STATEMENT _ID = '
E14' FOR
```

```
SELECT EMP.EMPNO,ENAME,DEPTNO,
EMP_SKILLS.SKILL,SKILL_DESCP
FROM EMP_SKILLS,EMP,SKILLS
WHERE EMP.EMPNO=EMP_SKILLS.EMPNO
AND EMP_SKILLS.SKILL=SKILLS.SKILL
```

```
AND DEPTNO LIKE 'E%'
```

```
QUERY PLAN
```

```
1.0 NESTED LOOPS
```

```
2.1 NESTED LOOPS
```

```
3.1 TABLE ACCESS BY ROWID EMP
```

```
4.1 INDEX RANGE SCAN EMP_DEPTNO NON _U-
NIQUE
```

```
3.2 INDEX RANGE SCAN EMP_SKILLS_U_EMP-
NO_SKILL UNIQUE
```

```
2.2 TABLE ACCESS BY ROWID SKILLS
```

```
3.1 INDEX UNIQUE SCAN SKILLS_U_SKILL U-
NIQUE
```

但是,若经考察知此时用索引还不如全表扫描来得快,而查询中必须要 DEPTNO LIKE 'E%' 这个条件,则需要限制索引的使用。可用一个函数对 WHERE 子句中的 DEPTNO 做一个修饰。例如用 NVL。由 NVL 函数适用于所有数据类型,因此该方法比较通用。当然还可以用如下方法:字符属性用加“11”修饰 (<列>11''=字符串),数值属性用加“0”修饰 (<列>+0=值)。

```
SQL> EXPLAIN PLAN SET STATEMENT _ID = '
E15' FOR
```

```
SELECT EMP,EMPNO,ENAME,DEPTNO,EMP
_SKILLS,SKILL,
SKILL_DESCP FROM EMP,EMP_SKILLS,
SKILLS
```

```
WHERE EMP.EMPNO=EMP_SKILLS.EMPNO
AND EMP_SKILLS.SKILL=SKILLS.SKILL
AND NVL(DEPTNO DEPTNO) LIKE 'E%'
```

```
SQL> @XPLAIN E15
```

一个微型控制器设计中的技巧

山东工业大学计算机系 (济南 250014) 张 昆
中国矿业大学 张晓光

用单片微机或微处理器和其它集成电路芯片组成微型专用控制器,实现对机器设备的自动控制,具有许多优点,它比继电器接触式控制器小巧、工作可靠、寿命长,功能的扩充灵活,价钱也具有相当竞争力;与使用成品计算机相比,由于其专用性强,能充分简化设计,价钱便宜,更重要的是它占有空间大小和形状较灵活,可以嵌入其它设备内部,特别适合于机电一体化。因此研究这种微型控制器的设计技巧,具有重要意义。本文阐述的是一具体设计实例中的技巧,希望它能体现设计中的一些规律。

这是为弯管机设计的一种数控系统,机器的工作性能类似于瑞典产康泰尔(KANTHAL)万能弯管机(KJG—4型)。但新的机器不只用于弯曲管材而且可用于弯曲钢筋一类的棒材。与康泰尔弯管机控制系统不同,新系统采用计算机控制,为控制功能的扩充留有较大的余地,输入更为方便,显示更为醒目,价格则便宜得多。数控系统的功能为:对工件的整个弯曲加工过程,可用按钮预置几种弯曲角度,对每种弯曲角度可预置重复次数。弯曲角度可在 $0\sim 210^\circ$ 间任选,重复次数可在 $0\sim 9$ 次间任选。预置的数字可即时通过数码管显示出来。加工时按照规定的顺序(由显示位置表示)执行每种角度和相应次数的弯曲加工动作。整个加工过程完成后,计件器自动加1。

1 控制面板设计

控制面板包括显示器和按钮两部分,是人——机间的接口,面板设计不但要考虑整个控制功能,操作方便,而且要考虑控制器在机器上的位置、几何空间。该面板镶嵌在弯管机前面,占有约 $200\times 150\text{mm}^2$ 面积。

收稿日期:1994年3月5日

几乎全部显示和按钮都在面板上。面板布置如图1所示。

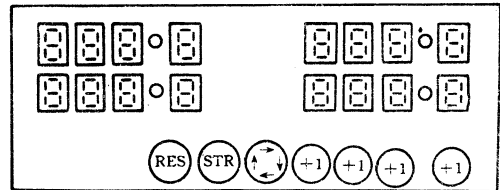


图1 面板布置

1.1 显示

用数码管显示,共有四组,从左上角开始沿顺时针方向,依次代表按相应顺序工作的四种弯曲角度和次数。每组前三位代表弯曲角度,第四位代表相应的弯曲次数。前三位与第四位之间设置发光二极管,当发光二极管亮时,表明该组为当前工作组或被设置组,当在工作状态时,该显示表明弯曲角度和次数的目标数,当在设置状态时显示相应按钮置入的数。

1.2 按钮

面板上共有七个按钮。从左至右,前四个钮分别用于每组四位预置数,分别与4位显示管相对应,第一个至第三个钮分别用于弯曲角度的百位,十位,个位的置数,第四个钮用于弯曲次数置数。置数按增1方式进行,每点动一次按钮,相应数增1,当持续压下按钮时,相应数每隔0.2秒自动加1。当数字达到最大值时再加1则回零。置入数的变化即时在相应数码管显示出来。第五个钮用于移动当前组,亦可用点动或持续方式输入,随着该钮的按动,四组的发光二极管可按顺时针排成的顺序依次发光,哪个发光则哪组为当前组。第六

QUERY PLAN

1.0 NESTED LOOPS

2.1 NESTED LOOPS

3.1 TABLE ACCESS FULL EMP_SKILLS

3.2 TABLE ACCESS BY ROWID. SKILLS

4.1 INDEX UNIQUE SCAN SKILLS _U _SKILL U-NIQUE

2.2 TABLE ACCESS BY ROWID EMP

3.1 INDEX UNIQUE SCAN EMP _U _EMPNO U-NIQUE

5 结束语

从以上讨论看出,同一查询的不同SQL语句写法会给查询性能带来显著的差异。象EXPLAIN和

TKPROF之类的工具能提供很有用的信息并免去很多推测帮助用户有效地调整自己的语句。作为设计者,唯一要掌握的是诸如表的大小和索引的选择率等数据统计信息。因此切实搞清统计信息如何影响性能是至关重要的。比如索引的使用并不是任何时候都能改善存取数据的效率。在ORACLE七版中,提供了基于统计信息的查询优化,免除设计人员了解统计信息和不断根据库情况改写SQL语句之苦。不管怎样,数据库调整(data base tuning)是一个实践问题,只有在实践中不断摸索,吸取教训,才能取得最佳途径。

个钮用于启动弯管机工作。第七个钮用于数控系统复位,复位后加工从头开始执行。如果某组加工次数置为0,则该角度不被加工。

1.3 几点说明

由于置数不是经常发生的,仅在加工一种管形之前进行,因此不要求快速,为减少按钮数,采用移位和加1方式置数。

为了减少移位次数,每次移动一组。当前置数组,亦为当前工作组,置数和加工两个过程不发生在同一时刻,因此互不矛盾,可用同一标示信号。

移组信号,采用LED管的形式,和采用数字闪烁形式相比,避免了数字本身闪烁给人带来的不适感。虽然增加了4个LED管,但不增加别的任何元件,LED仅占用了锁存及驱动器的第8位;LED不但起光标作用,而且起区分弯曲角度和次数显示的作用。

与采用移组方式相对应,4个置数按钮位置与4个数位相对应,适合于人的操作习惯,且总键数仍不超过8个,仍可采用直接一次读入全部键状态的方式。移组和LED非闪烁标示方式,在编程上亦便于处理。

2 系统硬件配置

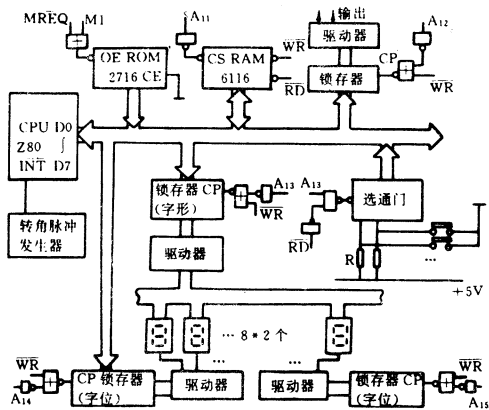


图2 系统硬件逻辑图

2.1 处理器用Z80 CPU

虽然有许多单片机可以采用,但Z80 CPU在不比单片机增加更多器件的条件下,能完成各项功能要求。其价格却比单片机便宜。

2.2 按键输入采用定时扫描直接读入方式

由于键的总数仅有7个,因此按图示接法,直接通过数据总线从相应键盘编码地址中取回一个字节,即可得按键状态象,由图示接法可知,当某键按下时,相应数据位为低电平,其余各位为高电平。

2.3 16位数码显示采用扫描方式

每位字形由数据总线轮流送入同一锁存器中存放,每2ms换一次,各位数码管的共阴极电平也是经数据总线分两次分别送入两个8位锁存器的相应位中保存,每次所送电平仅有一位为低,16位轮流送低电

平,也是2ms换一次。当哪一位共阴极为低电平时,该位数码管才可能有电流通过,才能按照字形锁存器中的数据显示出字形。由于一位数码管一次显示时间为2ms,16位轮流一遍需32ms,因此每位数码管一秒中可显示31次,人眼感觉不出闪烁。为保持足够亮度,需对显示管提供较大脉冲电流,为此对字形和字位都采用驱动器。这种显示方法比每位数码管单独采用字形锁存器的显示方法节约了器件和空间。

此外,采取扫描方式,即使偶尔发生一、二次送显示错误,亦不致造成影响。由于需显示的数和字形都放在内部存储器中,这部分处在系统的核心部位,能采取较好屏蔽措施,不易受到干扰发生错误,而输出部分则比较易于受到干扰。若不断从内存中取数和字形送至显示,则偶尔一次送字形错误,在继之而来的再次送字形中,可得到纠正。因此不大可能因干扰发生持久的显示错误,短时错误,可以不被人所感觉。

2.4 转角计数脉冲接入处理器中断请求端INT

由于计数脉冲间隔时间较长,可以采用中断的方法对脉冲进行计数以及其它处理。这种方法省却了单独的CTC计数器。

2.5 对完成工件数的累计采用由脉冲电流驱动的机械式计数器,不但价钱便宜,而且可防止电源断电时计件数的丢失。

2.6 各输出信号状态平时都在内存中保存,各占一位,组成一个字节,任何一位需输出时都重新将整个字节输出。

2.7 地址译码的简化

由于ROM和RAM仅占用A₀~A₁₀共11位地址线,Z80CPU尚有A₁₁~A₁₅高5位地址线可用于译码。考虑到与CPU进行数据交换的部件共有7部分:ROM、RAM、输出信号锁存器、显示字形锁存器和2个选位锁存器、键输入电路选通门,没有必要进行全译码,而是分别用A₁₁~A₁₅中的单根地址线,配合CPU的WR、RD及M1等信号,就可单独选中各部分。如图2所示,A₁₅与WR共同作用产生右边选位锁存器的打入脉冲,这时数据线数据被送入该锁存器。同理A₁₄=1,其余高4位地址为0时,选中左边的选位锁存器;利用A₁₃=1和WR信号选中显示字形锁存器,利用A₁₃=1和RD信号选中键电路的选通门;利用A₁₂=1和WR信号选中输出信号锁存器;利用A₁₁=1选中RAM;利用M1和MREQ信号选中ROM(产生指令读出信号),由于仅在读指令周期才同时有M₁和MREQ信号,因此不会与其他部分的读出信号混淆。

以上接法实质上是把输入输出电路与存储器统一编址,与此相对应数据传送指令也都用存储器数据传送指令。图示接法各部分编址如下:RAM地址空间为0800H~0FFFH,ROM地址空间为0000H~07FFH中的任一个,字形锁存器地址和选通门地址可用

2000H~27FFH 中的任一个,左、右两个数显字位选择锁存器地址分别可用 4000H~47FFH 和 8000H~87FFH 中的任一个。顺便说明,字形锁存器和选通门虽地址重叠,但由于前者为只写,后者为只读,因此可由读、写信号区别。

3 控制程序

经常要做的工作放在主程序中完成,随机出现的工作则放在中断程序中完成。

3.1 主程序 图3为主程序逻辑框图。

主程序要做的工作为送显示和按键处理,这两部分工作穿插进行,互不重叠。每送一位显示,延时 2ms 接着送下一位显示。如果前后送了 5 次显示,延时恰好 10ms,则检查有无键按下。如果键按下,再将键码与上次(10ms 前)取回的键码比较,若二者不同,则暂存键码,而后转延时 2ms 及送下一位显示;如果键码与上次相同,表明此键确实按下,则转键处理程序段。对于数字键及移位键,若按键时间不足 1000ms,则作一次

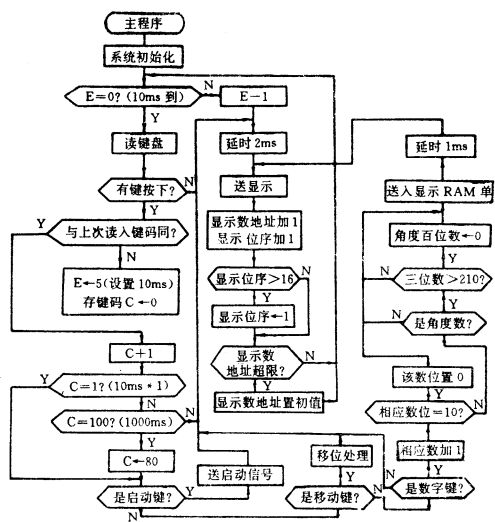


图 3 主程序逻辑图

点键处理,对数字键作相应数加1处理,对移位键作移一组位置处理;如果按键超过1000ms,则作持续按键处理,每隔200ms就加1(或移一组位),直至检查到此键已放开,以此加快输入速度。对于置数键,若置入角度超过210°,则将百位数置零,此外对于十位数和个位数若加1后够10,则置零。由于三位数总值不得超过210°,显然百位数不会超过2。在键处理完后,考虑到执行这部分程序,约占用1ms,则再执行1ms延时程序,而后转入送下一位数码显示,这样,一位数码的一次显示时间仍为2ms。延时1ms或2ms,皆利用执行循环减1程序实现,而不必应用硬件计时器,由于处理机任务不重,这种软件延时方法不影响正常任务的完成。至于更长的延时,则是利用累计2ms延时次数而实现,执行这部分程序仅用几条指令的时间(微秒级)。也由于显示和键防抖措施不需严格定时,这种方法才得以应用。

3.2 中断服务程序

由于仅这一个中断源,采用中断方式 1,可直接转入这部分程序。这部分程序完成的任务是:(1)对转动角度计数。(2)计数值达到预置值时,发出制动信号。(3)对弯曲次数计数。(4)对完成工件数计数。程序框图如图 4 所示。

3.3 仅从程序上加以改变,这一控制器能适用于多几倍的弯曲角度。相应的显示和置数可采用“分页”的方法,即依次在面板上显示各页。

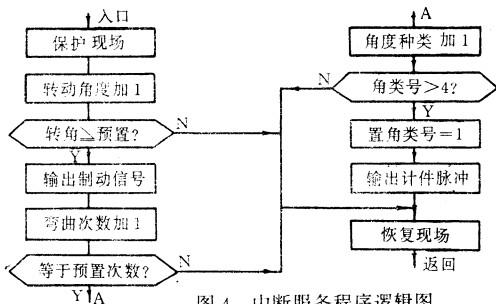


图 4 中断服务程序逻辑图

(上接 16 页)

数据库管理

在一个松耦合系统中使用 ORACLE 并行服务器时,并不需要让系统中的每台 CPU 都持有一个独立数据库。相反,由于所有 CPU 可以通过 ORACLE 并行服务器来共同访问一个单一数据库,因此所有的数据都可以放入该单一数据库中。这就大大减轻了数据库管理员 (DBA) 的工作负担。

应用开发和调度

使用 ORACLE 并行服务器后,所有数据都可放入一个单一库中,所有 CPU 都可以直接访问该库。这样,应用开发工作变得与单机系统上的一样简单。应用开发时可以不再去考虑该应用将在哪台 CPU 上运行,也不

用考虑部分系统出现故障后的保护方法。由于 ORACLE 并行服务器允许所有 CPU 直接访问整个数据库, 因此象分布数据库环境下将数据按 CPU 静态分配的做法不再需要。还有, 松耦合系统中某一 CPU 发生故障时, 并不会导致整个数据库的某一部分不再可用。

结束语

ORACLE 并行服务器是九十年代数据库技术的方向。它完全冲破了单机环境的束缚,使得充分利用松耦合硬件结构的数据库服务器的性能大大提高。ORACLE 还直接影响了计算机硬件厂商们对要生产什么样的计算机所做出的决策。ORACLE 并行服务器做好了充分准备来迎接数据库服务器上日益增长的性能需求,并且它正迅速成为九十年代唯一的“超级服务器”。

内存状态显示方法

中国矿业大学选矿工程研究室(徐州 221008) 梁春成 田慧珍

摘 要 针对 TSR 程序广泛使用,本文叙述了利用覆盖技术拷贝原始中断向量的方法,介绍了查找内存块与区分环境内存控制块及文件内存控制块的方法,并以上述为基础,说明如监测中断向量的变化和显示内存使用情况。

关键词 内存驻留 中断向量 内存控制块

内存空间是计算机四大硬件资源中最关键而且又是最紧张的“瓶颈”资源,对它的管理是操作系统的主要功能之一,近年来内存驻留程序(TSR)广泛使用,由于没有相应的标准规范,自由使用 TSR 的结果,使 DOS 对其管理也处于乏力的地位。

TSR 作为后台服务,通常在 DOS 的基础上增加或删除功能,同时占用相应内存空间,使内存状态发生变化,这些变化主要表现在如中断向量的改变;内存驻留目录;内存占用等方面,及时了解这些变化对安全运行具有实际意义。笔者编制的实用程序 MDIR.COM 包括了如下几方面的功能:记录原始中断向量;显示被修改的中断向量;显示 TSR 的路径名称;显示环境内存控制块(EMCB)、文件内存控制块(FMCB)及相应的内存分配块(MAB)的大小。

1 中断向量监测

中断向量共有 256 个,它储存于内存地址 0000:0000H 以后的 1KB 单元中,每个向量占用 2 个字,按中断号顺序排列,低位字是偏移量,高位字是段值。TSR 通常利用修改后的中断向量作为启动管道。监视中断向量变化即可对内存状态作概括性了解。

要完成监测中断向量变化功能,应首先拷贝原始中断向量表,考虑到拷贝功能只能使用一次,在程序中使用了覆盖技术,开机后在自动批处理中第一个运行该程序,在获得中断向量表的同时,拷贝程序段将被“消灭”,自此以后更换操作系统会使显示信息没有价值。

覆盖的方法是在程序的第一条指令上使用 Jmp,跳转的长度为 1029 字节,然后新的位置上又用一条 Jmp 指令向前跳执行拷贝,由于拷贝程序段长度不足,可在第一条指令后设立数据区补足,在第 2 个 Jmp 后填上几个 NOP 指令会使连接更加可靠。以后运行该程序,将利用 INT 21H 35H 子功能读取当前各中断向量,并与原始拷贝对比,不同则表明中断向量被修改过。

在 80 列屏幕上,每行显示两个中断号,并显示新

的中断向量和原始中断向量。每个中断号的显示格式为:

```
INT XX NEW XXXX:XXXX OLD XX  
XX:XXXX
```

2 内存块核查

MS-DOS 对内存空间的使用采取内存块链方法,内存块由内存控制块(MCB)和内存分配块(MAB)组成,MCB 又分为环境内存控制块(EMCB)和文件内存控制块(FMCB)。MCB 共 16 个字节,首字节为标识字符,“E”表示为最后一个内存块,“M”表示为非最后一个内存块,1~2 字节是配主程序 PSP 的段值,该字若为 0 表示为自由块,3~4 字节为 MAB 的大小,以节为单位,5~0FH 保留。系统开工之后,第一个内存块的段值随 DOS 版本不同被登记于一个固定的内存单元中,偏移量固定为 0。下一个内存块的段值,由当前 MCB 段值+MAB 的值+1 后得出,如此构连成链^[1]。

内存块的核查在不固定 DOS 版本的情况下,可采用顺序查找法,假定内存低位上的某一字为首块 MCB 的段值,检查该段址上的第一个字节是否为“M”或“E”,如为“M”根据上述下一个内存块段值的计算公式继续查找,直到“Z”出现表明查找成功。如过程中出现差错说明假定的段值是错误的,更换新的存储单元继续查找。查找的范围是 640K 空间。

3 环境控制块与文件控制块的区分

在内存控制块中并没有明显的环境内存控制块(EMCB)和文件内存控制块(FMCB)标志,但 DOS 在加载 .EXE 和 .COM 型文件时,会给程序分配一个 EMCB 和 FMCB,因为它们都附属于同一程序,必具有相同的 PSP 段值,DOS 把 EMCB 放在前面,FMCB 放在后面,只要相邻的两个 MCB 有相同 PSP 段值,就可区分 EMCB 和 FMCB。

对那些要更深隐藏的程序,它们不仅修改程序段前缀的内容,而且会修改 MCB 中的 PSP 段值为 0,所以多个自由块的存在也可能意味着有 TSR 已在其中。

4 路径文件名

查找 TSR 一般情况下可以通过程序自身的 PSP 内容,即+18 偏移处开始的 20 个字节句柄文件打开

表获得已打开文件的数量,系统文件控制块(FCB*)号等内容,进而通过文件控制块(FCB)查知相应的文件名及有关信息,但有些 TSR 在完成初始化后覆盖了 PSP,甚至可能修改了 FCB,使利用这种方法的查找工作失败。

内存块是 DOS 控制内存空间的途径,内存块的存在是无法变更的,是否在内存块中可查知路径文件名呢?答案是肯定的。环境分配块(EMAB)中的内容是一组 ASCII 字符串,以标志 0000H 字表示组结束,紧跟一个字的进程计数后,就是我们要找的路径文件名。EMAB 的段值比 EMCBC 的段值大 1,所以顺序查找组结束标志字 0000H,并把偏移量再增加 2 就是路径文件名的存储位置,路径文件名的结束标志是 00H。

5 内存状态显示

中断向量的显示如前所述,内存使用情况的显示方法是依次显示每个内存块的段址,然后显示 MAB 的长度,单位为节,对自由块在路径文件名位置上显示“FREEMCB”。最后一条信息将显示本程序使用内存的情况,当见到显示信息后,该块已被释放,我们可以从中计算出真正的自由空间值。

在本文结束之前提醒大家注意,不要随意删除 TSR,因为 TSR 不仅修改中断向量,而且可能修改操作系统的其它部分,使用的手法和形式多种多样,在不了解程序结构情况下,贸然删除,很可能造成微机系统的软、硬故障,最可靠的删除方法就是重新启动机器。

本文上述问题可参考下面给出的源程序进一步了解。

*****MDIR.ASM*****

;内存状态显示程序,使用 MASM,命令:MDIR

prog segment 'code'

assume cs:prog,ds:prog

org 100h

start: jmp start0

flg db '\$ \$'

mem db 898 dup(?)

start1: mov ah, 3dh

mov dx, offset itself

mov al, 2

int 21h

mov handle, ax

mov ax, 4200h

mov bx, handle

mov cx, 0

mov dx, 5

int 21h

push ds

mov bx, handle

mov ax, 0

mov ds, ax

mov dx, 0

mov cx, 400h

mov ah, 40h

int 21h

pop ds

mov ah, 3eh

mov bx, handle

int 21h

mov dx, offset wbuffer

mov di, dx

mov byte ptr [di], 90h

;为覆盖预留空间

;中断向量表起始地址标志

;1KB 空间补足数据区

;以读写方式

;打开自身

;磁盘文件

;

;保存文件句柄

;移动文件

;指针

;5 个字节

;

;修改 DS 准备

;装入句柄

;修改 DS

;为 0000H

;设定偏移量为 0000H

;设定写字字节数 1024

;拷贝中断

;向量表

;

;关闭

;当前

;文件

;建立写缓冲区

;地址指针

;移入

mov byte ptr [di+1], 90h

mov byte ptr [di+2], 90h

mov byte ptr [di+3], 90h

mov ax, 3d02h

lea dx, itself

int 21h

mov bx, ax

mov handle, ax

mov cx, 0

mov dx, 405h

mov ax, 4200h

int 21h

mov bx, handle

mov dx, offset wbuffer

mov cx, 4

mov ah, 40h

int 21h

mov ah, 3eh

mov bx, handle

int 21h

mov ah, 4ch

int 21h

start0: jmp start1

NOP 覆盖

jmp start2

rbuffer db 65 dup(?)

wbuffer db 256 dup(?)

handle dw 0

coun db 0

pagenum db ?

ads dw ?

ado dw ?

pmps dw ?

vmcb db 'VMCB:', '\$'

vmab db 'VMAB:', '\$'

fmcb db 'FCMB:', '\$'

fmab db 'FMAB:', '\$'

freemcb db 'FREEMCB:', '\$'

itself db 'MDIR.COM', 0

intchr db 'INT'

intnum db '00'

newchr db 'NEW'

newaddr db '0000'

dchr0 db ', '

newaddro db '0000'

oldchr db 'OLD'

oldaddr db '0000'

dchr1 db ', '

oldaddro db '0000'

intover db ', '\$'

overp db 0dh, 0ah, '\$'

start2: mov ah, 0fh

int 10h

mov ah, 0

int 10h

start3: mov ah, 35h

mov al, coun

int 21h

mov al, coun

call ht

lea dx, intchr

mov di, dx

mov byte ptr 4[di], ah

mov byte ptr 5[di], al

mov ax, es

xchg ah, al

call ht

mov byte ptr 12[di], ah

mov byte ptr 13[di], al

mov ax, es

call ht

mov byte ptr 14[di], ah

mov byte ptr 15[di], al

mov ax, bx

xchg ah, al

call ht

mov byte ptr 17[di], ah

mov byte ptr 18[di], al

mov ax, bx

;4 个

;NOP

;指令

;再次

;打开

;文件

;存储

;句柄

;移动

;文件指针

;到最后

;

;设定写缓冲区

;设定写字字节数

;调用

;写文件中断

;关闭

;当前

;文件

;拷贝中断向量结束

;正常退出

;1KB 结束处, 此处被

;

;程序第二次及以后运行

接续点

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

call ht	; 低位字节	cmp byte ptr es:[0],5ah	; 是最后一个内存块?
mov byte ptr 19[di],ah	; 转换为 ASCII	jz sta5	; 是转
mov byte ptr 20[di],al	; 存储	mov ax,cs	; 不是
mov al,coun	; 取中断向量号	mov bx,word ptr es:[3]	; 块段值+MAB 的值
xor ah,ah	; 乘 4 后加 2	add ax,bx	; 再加 1
mov cl,2	; 作为查找中断	inc ax	; 到下一内存块
rol ax,cl	; 向量的	jmp sta3	; 继续查找
add ax,2	; 附加偏移地址	sta5:nop	; 开始 EMCB 和 FMCB 的
mov si,ax	; 指针		区分
mov ax,word ptr flg[si+2]	; 取原始中断向量	mov pagenum,0	; 清自由块标记
xchg ah,al	; 段址取高位字节	mov bx,0	; 设定附加偏移值为 0
call ht	; 转换为 ASCII	mov di,bx	
mov byte ptr 27[di],ah	; 存储	lea bx,rbuffer	; 取偏移地址
mov byte ptr 28[di],al		staa0:cmp pagenum,1	; 前一个 PSP 段值为 0?
mov ax,word ptr flg[si+2]	; 取原始中断向量段址	jnz staa00	; 不为 0 转
call ht	; 低位字节转换为 ASCII	mov pagenum,0	; 是,清标志
mov byte ptr 29[di],ah	; 存储	sub di,2	; 附加偏移值减 2
mov byte ptr 30[di],ah		staa00:mov ax,word ptr[bx+di]	; 取块段值
mov ax,word ptr flg[si]	; 取原始中断向量	mov es,ax	; 入 ES
xchg ah,al	; 偏移地址高位字节	mov handle,ax	; 并保存
call ht	; 转换为 ASCII	mov ax,word ptr es:[3]	; 取 MAB 的值
mov byte ptr 32[di],ah	; 存储	mov ads,ax	; 并保存
mov byte ptr 33[di],al		cmp word ptr es:[1],0	; 该块为自由块
mov ax,word ptr flg[si]	; 取原始中断向量	jz staa2	; 是转
call ht	; 低位字节转换为 ASCII	mov pagenum,1	; 不是设立标志
mov byte ptr 34[di],ah	; 存储	mov ax,word ptr es:[1]	; 取 PSP 段值
mov byte ptr 35[di],al		mov psp,ax	; 并保存
push ds	; 恢复	mov ax,word ptr es:[3]	; 取 MAB 的值
pop es	; ES 寄存器	mov ads,ax	; 并保存
lea dx,newaddr	; 装入当前	inc di	; 附加偏移地址
mov si,dx	; 中断向量地址指针	inc di	; 指针增 2
lea dx,oldaddr	; 装入原始	mov dx,di	; 查回
mov di,dx	; 中断向量地址指针	cmp coun,dl	; 最后一个块完成?
mov cl,9	; 设定检查字符数	jz staa1	; 是转
start4:cmpsb	; 比较	mov dx,word ptr[bx+di]	; 不是,取下一个段值
jnc start5	; 不等转	mov es,dx	; 到 ES
dec cl	; 相等计数减 1	cmp word ptr es:[1],0	; PSP 段值为 0?
cmp cl,0	; 比较完成否	jz staa2	; 是转
je start7	; 完成转	mov ax,word ptr es:[1]	; 不是,取 PSP 段值
jmp start4	; 否则继续比较	cmp psp,ax	; 与前一个 PSP 段值比较
start5:mov ah,9	; 显示中断向量	jz staa4	; 相同转
lea dx,intchr		mov pagenum,0	; 不同,清标志
int 21h		jmp staa0	; 继续比较
start7:add coun,1	; 中断号码增 1	staa1:jmp stab0	; 退出本次运行
jc start8	; 已完成 256 个转	staa2:lea dx,vmcb	; 显示“EMCB”
jmp start3	; 否则继续核对	call mcbd	
start8:nop	; 以下进入内存块查找	mov ax,handle	; 显示 EMCB
lea dx,overp	; 首先进行	call dseg	; 段址
mov ah,9	; 两个回车与换行	lea dx,vmab	; 显示“EMAB”
int 21h	; 区别两种	call mcbd	
int 21h	; 不同的显示内容	mov ax,ads	; 显示 EMAB
mov coun,0	; 清内存块计数器	call dseg	; 的值
mov ax,0	; 从段地址 0	lea dx,freeemb	; 显示“FREEMCB”
mov ads,ax	; 偏移地址 1	call mcbd	
mov ado,ax	; 开始寻找首块	lea dx,overp	; 回车换行
sta0:ad4 ado,1	; 内存控制块段地址存储单元	call mcbd	
jc stal	; 超出 64K 转	sta3:inc di	; 附加偏移地址
jmp sta2	; 否则转	inc di	; 指针增 2
stal:mov ado,0	; 重设偏移地址为 0	mov dx,di	; 检查全部完成?
mov ax,ads	; 段地址	cmp coun,dl	
add ax,1000h	; 增加 1000H	jz staa1	; 是转
mov ads,ax		jmp staa0	; 否则继续
sta2:mov ax,ads	; 取查询段地址	staa4:mov ax,word ptr es:[3]	; 取后一内存块 MAB 的值
mov es,ax	; 入 ES	mov ado,ax	; 并保存
mov bx,ado	; 取偏移地址为指针	lea dx,vmcb	; 显示“EMCB”
mov ax,es,[bx]	; 取假设内存块段值	call mcbd	
sta3:mov es,ax	; 到 ES	mov ax,handle	; 显示 EMCB
cmp byte ptr es:[0],4dh	; 这个段的第一字节为“M”?	call dseg	; 的段值
jz sta4	; 是转	lea dx,vmab	; 显示“EMAB”
cmp byte ptr es:[0],5ah	; 不是,第一字节为“Z”?	call mcbd	
jz sta4	; 是转	mov ax,ads	; 显示 EMAB
mov coun,0	; 不是,清内存块计数器	call dseg	; 的值
jmp sta0	; 转入下一地址继续查找	call dfn	; 查路径文件名并显示
sta4:mov al,coun	; 取当前内存块数	lea dx,fmcb	; 显示“FMCB”
xor ah,ah	; 作为附加	call mcbd	
mov di,ax	; 偏移地址指针	mov ax,es	; 显示 FMCB
lea bx,rbuffer	; 取偏移地址	call dseg	; 的段值
mov ax,es	; 存储	lea dx,fmab	; 显示“FMAB”
mov word ptr[bx+di],ax	; 已找到的段值	call mcbd	
add coun,2;	更新指针		

mov ax,ado	;显示 FMAB	lea dx,wbuffer	;调用
call dseg	;的值	mov ah,9	;显示
lea dx,overp	;回车换行	int 21h	;中断
call mcdb	; ;	lea dx,intover	;回车
jmp staa3	;进行下一个循环	mov ah,9	;换行
mcdb:mov ah,9	;显示中断调用	int 21h	; ;
int 21h	; ;	pop es	; ;
ret	; ;	pop di	; ;
dseg:push bx	;显示字子程序	pop si	; ;
push ax	;入口 AX,出口无	pop dx	; ;
xchg ah,al	;取字高位	pop cx	; ;
lea bx,oldaddro	;取存储偏移地址	pop bx	; ;
call ht	;转换为 ASCII	pop ax	; ;
mov byte ptr[bx],ah	;存储	ret	; ;
mov byte ptr 1[bx],al	; ;	stab0:push ds	;恢复
pop ax	;取字低位	pop es	;ES 值
call ht	;转换为 ASCII	mov ax,4c00h	;正常退出
mov byte ptr 2[bx],ah	;存储	int 21h	; ;
mov byte ptr 3[bx],al	; ;	ht proc near	;16 进制——ASCII 转
push dx	; ;		换子程序
lea dx,oldaddro	;取显示偏移地址	push bx	;入口 AL,出口 AX
mov ah,9	;调用显示中断	push cx	; ;
int 21h	; ;	push dx	; ;
pop dx	; ;	push si	; ;
pop bx	; ;	push di	; ;
ret	; ;	mov ah,al	;取前 4 位
dfn:push ax	;查找路径文件名子程序	and ah,0f0h	; ;
push bx	; ;	mov cl,4	; ;
push cx	; ;	ror ah,cl	;右移 4 位
push dx	; ;	and al,0fh	;取后 4 位
push si	; ;	lea dx,hchr	;取对照表偏移地址
push di	; ;	mov di,dx	; ;
push es	; ;	mov bl,ah	;移位后字节为
mov ax,handle	;取 EMCB 段值	xor bh,bh	;附加偏移地址指针
inc ax	;增 1	mov ah,byte ptr[dx+di]	;取前 4 位的 ASCII
mov es,ax	;送入 ES	mov bl,al	; ;
mov bx,0	;设定偏移地址为 0	mov al,byte ptr[dx+di]	;取后 4 位的 ASCII
dfn1:cmp word ptr es:[bx],0	;环境段组结束?	pop di	; ;
jz dfn2	;是转	pop si	; ;
inc bx	;否则偏移地址增 1	pop dx	; ;
jmp dfn1	;继续查找	pop cx	; ;
dfn2:add bx,4	;跳到路径文件名开始处	pop bx	; ;
lea dx,wbuffer	;设定偏移地址	ret	; ;
mov di,dx	;指针	hchr db '0123456789ABCDEF'	; ;
dfn3:mov al,byte ptr es:[bx]	;取路径文件名	ht endp	; ;
cmp al,0	;取完?	prog ends	; ;
jz dfn4	;是转	end start	; ;
mov byte ptr[di],al	;否则移动	end	; ;
inc bx	;更新		
inc di	;读写地址指针		
jmp dfn3	;继续读写		
dfn4:inc di	;写地址指针增 1		
mov byte ptr[di],24h	;写入“\$”		

参考文献

- 1 张昆藏 . DOS 内核的奥秘 . 北京科海培训中心, 1990. 10

(上接 2 页)70%以上被美国军队和政府定购或推荐使用,“信息高速公路”上使用它可得到绝对的可靠性,加上它超凡的设计;将局部总线、多功能卡、SCSI-11、声音卡,以太网卡显示卡都设计在一个主板上,可扩至 64 兆内存,完全满足信息高速公路上各终端机的需要。就它的设计功能看,①音卡可将模拟信号转换成数字信号,因此它具有收音、录音、放音功能加上配置的两只 Sony 柱形喇叭,能够收到 HIFI 的声音效果、闲时还可作音乐欣赏用机。

②它所配置的 SCSI-11 卡可带动几十兆光盘,而且内存可扩至 64 兆,加上图像卡这就使图像的实时传输成为切实可行,电视电话中对方的图像就可逼真地显示出来以致使双方察颜观色,洞悉双方对事物的看法,促进信息交流,开创出全新的影像通讯时代。

③特殊的 ethernet 网的硬件功能,加上微软公司专为信

息开发出的超智能网络软件,使得各网之间壁垒大开,互相开放,促进信息共享。

④该机本身的微处理器属 486 档次,微处理器可单芯片升级,这种档次的微机、本身就是一台功能强大的计算机、可满足于一般数据、文字处理、打印、设计等要求,加上微软的 WINDOWSNT, SUNsoft, OS2 energy AT&T 软件都能兼容,这种机器既可满足于“信高路”微机通信方面的要求,也可满足一般单位、家庭的工作需要甚至娱乐要求,难怪在美国政府宣布信息高速公路计划中采用 Ethernet 硬件,该公司的股票从 1:9\$ 猛增至 1:150 多\$, EVEREX 公司的股票也稳中有升,微软的股票更不用说,从 92 年底的 1:1000 涨至 93 年底的 1:1500\$,且今年还有上升。这标志着“信息高速公路”的重要性和迫切性,也标志着现代社会中信息业的繁荣兴旺和信息业经久不衰创造高产值高效益的倍增效应。

用软件应答进行串行通信

江汉石油学院信息系 (沙市 434102) 许青林

摘 要 本文简化两台计算机通信连线到最简的三线连接,用软件完成信号应答,避免了硬件造成的诸多通信错误,具有极高的可靠性和便于控制的特点,对于多机控制和管理系统的数据传输均具有很高的参考价值。

关键词 通信 缓冲 BIOS 软中断

1 概 述

在多机管理系统及控制系统中都存在着上位机和下位机或管理计算机和现场计算机之间的通信。一般来说,在这种情况下,传输的数据量不大,速度也不要太高,但要求有高可靠性,便于控制。因此可以利用 RS-232 串行口来进行通信。

RS-232 的 D 型连接器有 25 个引脚,其中 20 个引脚已被 RS-232 标准所定义,但一般的微机系统只使用其中的 9 个,并且现在越来越多的微机采用 9 针插座。表一列出了这 9 个引脚的含义及两种插座的对照表。

表一 RS-232 主要引脚定义

25 脚脚号	9 脚脚号	信号名称	缩写名
2	3	发送数据	TD
3	2	接收数据	RD
4	7	请求发送	RTS
5	8	清除发送	CTS
6	6	数据装置就绪	DSR
7	5	信号地	GRD
8	1	载波检测	DCD
20	4	数据终端就绪	DTR
22	9	振铃指示	RI

在不使用 Modem 的情况下,两台计算机的连接大多采用图 1 的方式。许多微机并不严格遵循 RS-232 标准,这种连线有时不能正常工作,为了减少对硬件的依赖性,可以用软件进行应答。在图 1 的七线连接方式中有两对线是为数据发送/接收服务的信号线,若采用软件应答方式,可以取消这两对线。即把各自的 4 和 5 连在一起,6 和 20 连在一起(见图 2),把自己发出的请求信号当作对方发来的回答信号接收,因此硬件

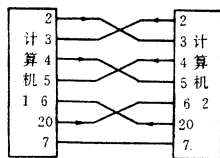


图 1 两台计算机的七线连接

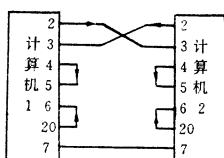


图 2 两台计算机的三线连接

线路一直是畅通的。如果此时没有软件进行协调,仅靠硬件这样连接就会出现数据超越错,即接收方计算机还没有来得及从其串行口寄存器中将数据读出就被下一个发来的数据所冲掉。下面以文件传送为例说明如何解决上述问题,如何用软件进行应答。

2 软件应答

软件应答的工作原理如下:发送计算机一方发送第一个字节,等待接收计算机返回应答字节,一旦收到应答字节,发送者送出下一个字节再继续等待应答字节;接收计算机一方,循环测试串行口,直到有数据到达,从串行口缓冲寄存器中读出数据,并送出一个应答字节。发送方和接收方的此过程一直到文件传完。

上述发送和接收过程可以用伪 C 语言形式地描述如下:

```
SEND() /* 发送方 */
{
    while ! feof() {
        发送一个字节;
        等待应答信号;
    }
}

RECEIVE() /* 接收方 */
{
    DO {
        接收一个字节;
        发送一个应答信号;
    } while ! feof();
}
```

用这种方法,可以避免数据超越错,使两台速度不同的计算机有条不紊地进行数据传输。

3 具体实现

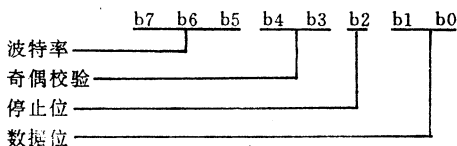
BIOS 可支持 4 个串行通信口,但一般只使用前两个,即 0 号和 1 号口,它们对应的逻辑设备名是 COM1 (或 COM) 和 COM2。

BIOS 提供的软中断 INT14H 用于串行通信服务,它具有四个功能,描述如下:

0 号功能:初始化串行口

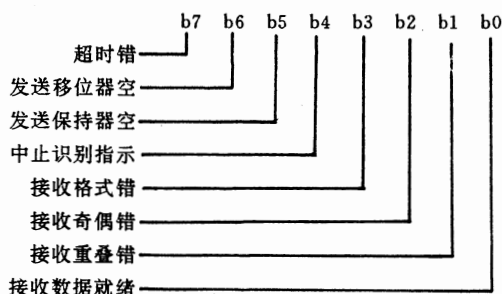
输入 DX = 串行口号

AL = 初始化参数,该参数编码成一字节



输出 AH=线路状态
AL=Modem 状态

在这里,我们没有使用 Modem,所以只讨论线路状态,在通信过程中线路可能出现多种状态,AH 的每一位表示其中的一种状态。



利用上述介绍的软件应答方式进行通信,可以避免硬件造成的许多错误。以上线路状态中的大多数可以不必关心,但是有两条状态是必须考虑的,它们是第7位和第0位,即“超时错”和“接收数据就绪”,在接收方要检查“接收数据就绪”信号,以确定何时端口已接收一字节。

中断 14H 的 1 号功能是向 DX 中指定的串行口传送一个字节,该字节数据必须在 AL 中,返回时,AH 中存有线路状态。

中断 14H 的 2 号功能是从串行口读一字节,所读字节放在 AL 中,线路状态在 AH 中。

中断 14H 的 3 号功能是检查串行口状态,在这里主要是检查 AH 的第 7 位是否置 1,即是否出现“超时错”。

利用上述的方法就可以实现数据的串行通信。但是,由于串行通信是逐个字节地进行的,即读一个字节,传送一个字节,接收计算机接收并向盘中写一字节,若这样进行,读/写磁头的磨损是相当厉害的。因此在具体实现中,在发送和接收计算机端各开辟了一个缓冲区,工作原理如图 3 所示。

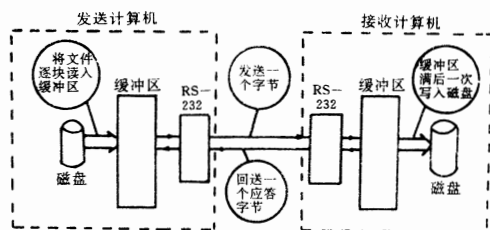


图3 带缓冲区的串行通信原理图

加了缓冲区后,由逐个字节读写变为逐块(这里定义每块为 512 个字节)读写,这样既可以减少磁头的磨损,又可以提高传输速度。

加了缓冲区后,发送过程的伪 C 源代码形式如下:

```
NUM_OF_BLOCK=LENGTH_OF_FILE/512+1;
SIZE_OF_BLOCK=512;
FOR(I=0;I<NUM_OF_BLOCK;I++)
```

```
{
    IF (NUM_OF_BLOCK==1)
        SIZE_OF_BLOCK=LENGTH_OF_FILE % 512;
    /* 最后一块可能不足 512 个字节 */
    依次从文件中读出 SIZE_OF_BLOCK 个字节到缓冲区;
}
```

```
FOR (I=0;I<SIZE_OF_BLOCK;I++) {
    发送一个字节;
    等待回答信号;
}
```

接收过程的伪 C 源代码为:

```
NUM_OF_BLOCK=LENGTH_OF_FILE/512+1;
SIZE_OF_BLOCK=512;
FOR(I=0;I<NUM_OF_BLOCK;I++)
```

```
{
    IF (NUM_OF_BLOCK==1)
        SIZE_OF_BLOCK=LENGTH_OF_FILE % 512;
    FOR (I=0;I<SIZE_OF_BLOCK;I++) {
        从串行口读一字节到缓冲区;
        发送一个应答字节;
    }
    将缓冲区中的 SIZE_OF_BLOCK 个字节写入文件;
}
```

4 结束语

利用这种方法编写的数据通信程序,不仅可以传送文本文件(包括西文和中文),还可以传送图形、图像、执行文件(EXE 和 COM 文件),执行文件传过来后可以直接运行。我们在多次使用这个通信程序中,没有出现传送错误,效果很好。当然高可靠性是由于牺牲速度换来的,在传输速度要求不太高,距离较近的情况下,这种方法是很有参考价值的。(需要该程序者,请直接与作者联系)

参考文献

- 1 张昆藏 编. 微型机(PC 系列)系统功能教程,清华大学出版社,1992 年 8 月

微机控制系统的实时多任务调度

铁道部资阳内燃机车工厂 (资阳 641301) 金 彪

摘 要 本文概述了微机实时控制中多任务调度的基本概念及一般用法,并介绍了一种实用的多任务调度方法。

关键词 实时控制 多任务调度 重入 派送 堆栈

随着计算机主机系统运行速度的不断提高,在计算机进行比较大型的控制中,用一块 CPU 主机芯片同时执行多个不同的任务,如控制,数据采集,屏幕显示,人机对话,参数记录等等,是非常实用的。和多主机控制系统相比,它不仅简化了硬件结构,而且避免了多主机之间的通讯问题。但由于各任务的实时性要求,例如,有些任务(特别是一些对外设的控制任务)完成一次循环要占据 CPU 比较长的运行时间,而另一些任务必须立即加以处理。因此,简单的将各任务顺序排列,不分优先级,循环执行,一个任务执行完毕,再转去执行下个任务,是不一定能满足所有任务的实时要求的。这就需要计算机系统软件中,有一套多任务调度系统,以调配各任务之间的切换,保证所有任务都能及时、迅速地得到处理。

计算机控制领域中的任务,是指完成某些功能的单独的程序模块,它们可以分别编译,以后连接成整个程序。每个任务有其相应的状态,如激活状况、休眠状况。系统据此决定是否执行该任务。多任务之间的调度可以用以下几种方法:

1. 一个任务需要服务时,它即进入准备就绪状况。各就绪的任务按先后次序排队,先就绪先执行。各任务不分优先级。这种方法比较简单。但它的实时性不佳。因为任何任务执行后都必须等其他就绪任务统统执行一遍才能再次执行。

2. 预先指定各任务模块的优先等级,任务调度从准备就绪的任务队列中选择优先级最高的任务执行,若该任务的优先级高于现行的任务,中断现行任务的执行。这种方法能保证对时间要求较高的任务能及时处理。缺点是这种方法可能产生很多现场保护,且不能保证有些优先级低的任务被执行。

3. 预先指定各任务模块的执行时间,一个任务执行时间到达后,即中止其执行,转去执行下一个任务。

4. 预先指定各任务模块的挂起时间和优先级。一个任务执行完毕后处于挂起状况。在预定的挂起时间到达后处于等待状况。所有处于等待状况的任务按优先级排队,这种任务调度考虑了任务的优先级,通过配置各任务的挂起时间,还可以使优先级低的任务有机会得到执行。但它处理起来比较复杂,要占用较多的处

理时间。

对一个任务调度系统的评价,既应考虑到它必须完成各个任务的要求,又要简单易行,任务切换本身应尽量少占用 CPU 时间和少占用内存。一种比较简单实用的方法,是将各任务顺序排列,不分优先级,循环执行。但不是一个任务执行完毕,再转去执行下个任务,而是在各任务模块中设置一个或多个切换点,保证每个任务每次运行只占用 CPU 足够短的时间,即转向执行下一个任务。各任务快速循环执行。切换点一般选在任务中需要等待的地方。例如,在查询等待某个状态时,若该状态不满足条件,即切换到下个任务,在下次循环到达时再次查询该状态。这样,该任务模块在等待这个状态时,并不占用 CPU 的时间。又比如,当某一任务的运算需要占用 CPU 较多的时间,而可能影响到其它任务的执行时,也可将切换点插入该任务的运算过程中。以下主要介绍这种任务调度方法。

每个任务有一个采用下面的结构表示的目录区:
(用 C 语言编写)

```
struct TASK {
    int status ;
    int (*entry) ;
    char *info ;
};
struct TASK TASK_LST [TASK_MAX];
```

另外,为每个任务分配一块长度为 SIZE 的堆栈区:

```
int stack [TASK_MAX * SIZE];
```

其中, TASK_MAX 表示任务总数。

该结构 TASK_LST 为各个任务模块在内存分配一个目录区,其中 int status 表明该任务处于何种状况,是激活状况,还是休眠状况。

每个任务首次执行时的入口地址或重入时的返回地址存放在自己的堆栈区中。TASK_LST 中 int (*entry) 指向该存放点。另外, char *info 是该任务运行结果的信息,该项是可选择项,它不影响多任务调度系统。

程序运行时,首先对各任务进行初始化处理,即在 TASK_LST 中给定各任务的初始入口地址,设置其初始状况。其后便进入任务派送 dispatch,执行第一个任务,程序从第一个任务的起始地址开始运行。在第一个任务中至少有一个重入 reentry,程序一运行到这里,就转入下一个任务,在下一个循环到来时,即当所

有其他任务被完成后返回该任务时,直接回到这个转换点继续运行。

每个任务分为一个一次性运行部分和一个主循环体。一次性运行部分主要用于指定该任务中变量的初始值。这部分程序只运行一次。以后程序一直在主循环体内运行。在主循环体内至少有一个 reentry,以转向下个任务。在整个程序的任何地方,都可使某个任务处于休眠或激活状况,这样可随时中止或激活某些任务的执行。

重入 reentry 用于任务之间的切换。它包括二个部分,一是保护现场,二是任务派送 dispatch。保护现场是将所有通用寄存器、变址寄存器、基址寄存器、以及状态寄存器压入该任务的堆栈区,将此处的堆栈指针写入该任务目录区中。这里的堆栈指针有两个作用。第一,它指示该切换点处的栈顶位置。第二,在调用 reentry 即 call reentry 时,由于 call 功能将切换点的指令指针 IP 压入堆栈内,所以,找到了这里的堆栈指针,也就找到了该点的程序指针,即任务的返回地址。任务派送 dispatch 就是转入下一个任务。dispatch 首先在任务目录区中找出下个任务的状况,若这个任务处于休眠状况,则不执行该任务,而执行再下个任务。若这个任务处于激活状况,就在任务目录区中找出该任务的堆栈指针,并具此弹出其上次运行时压栈的内容,即恢复现场。最后执行调用返回 ret。该指令将此处堆栈指针指向的内容(即这个任务切换点的程序指针 IP)弹入 IP,执行该任务的运行。

这里说明一下多任务调度中堆栈的作用。每个任务都有它自己的堆栈区。程序在调用 reentry 时转向下一个任务后,也将堆栈指针转为下个任务的堆栈指针。这样,各任务之间互相独立,不产生任何干扰。该任务的所有内部变量,执行过程中的所有堆栈操作等都在这个区域。用高级语言编制的程序,在调用功能时,将产生很多压栈操作。所以,要选择好堆栈区的大小。太大则占用过多的内存,太小会产生溢栈,破坏整个任务调度系统。各任务堆栈区采用数组变量的方式定义,即它占用的是变量的内存,所以不会和操作系统产生地址冲突。

这种任务调度无需任何操作系统的支持。以下程序适用于 8086 系列 CPU 的工作。为了保证任务之间的转换速度,这部分程序用汇编语言编制。

```

dispatch    proc        near
cli;
dirspal:
mov ax,NXT_TASK;
mov AKT_TASK,ax;
mov dx,ax;
inc ax;
cmp ax,MAX_TASK;
jl  dispa2;
mov ax,0;

```

```

dispa2:
mov NXT_TASK,ax;
lea si,TASK_LST;
add si,dx;
add si,dx;
add si,dx;
add si,dx;
add si,dx;
add si,dx;
add si,dx;
mov al,[si];
or al,al;
jnz dispal;
inc si;
inc si;
mov bl,[si];
inc si;
mov bh,[si];
or bx,bx;
jz dispal;
mov sp,bx;
popf;
pop ax;
pop cx;
pop dx;
pop bx;
pop bp;
pop si;
pop di;
sti;
ret
dispatch    endp
reentry     proc        near
push di;
push si;
push bp;
push bx;
push dx;
push cx;
push ax;
pushf;
mov bx,sp;
lea si,TASK_LST;
mov dx,AKT_TASK;
add si,dx;
add si,dx;
add si,dx;
add si,dx;
add si,dx;
add si,dx;
inc si;
inc si;
mov [si],bl;
inc si;
mov [si],bh;
jmp dispatch;
reentry     endp

```

上述程序中,AKT_TASK 代表当前的任务号, NXT_TASK 代表下一个任务号。

参考文献

- 1 沈兰荪 编著. 数据采集技术, 中国科学技术大学出版社。

生物组织结构定量分析系统的开发

华西医科大学(成都 610044) 高云钦

摘要 本文介绍了作者在访美期间开发的一种生物组织结构定量分析系统。该系统的分辨率和精确度都很高,并且几乎可以测量二维图像中应该测得的所有参数。该系统结构简单,操作简便直观。实践证明,该系统是成功的。

关键词 生物组织结构/定量分析

1 引言

生物医学工程与人类健康有着很密切的关系,用定量分析方法来研究生物组织的微观结构,将对生物组织结构的微观认识提高到一个新的水平。定量地观察在各种疾病的情况下与之有关的生物组织结构的微观变化,有助于寻找预防和医治人类各种疾病的更有效方法。

从生物组织学的研究中已知,生物组织结构的主要特点是:组成成分的种类很多,体积很小,且许多种类的形状极不规则;某些组成成分的数量很大,分布不均匀,且呈现动态变化的情况等。因此,要求定量分析系统至少应具备如下一些功能:

1. 系统的分辨率要求较高。
2. 图像采集系统的测量单位应小到以埃(angstrom,一埃= 10^{-7} mm)为单位;
3. 系统应能自动识别待测物体的形状的边界,不允许操作员用光笔描绘,以免失真太严重;
4. 待测的原始参数,如面积、周长、直径、角度、和计数等应由系统自动测量、采集和记录;
5. 系统应能对所测参数进行必需的统计分析。

本文将介绍作者在访美期间所开发的一种生物组织结构的定量分析系统。

2 图像采集硬件系统

系统组成如图 1 所示:

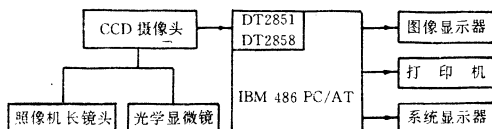


图 1 图像采集硬件系统组成框图

若是用生物组织切片,需用光学显微镜进行高倍放大;若用电镜片,则采用照相机的长镜头即可。

CCD 4810 摄像头是美国 COHU, Inc 产品,具有 754(H)×488(V)像元的高分辨率能力和高灵敏度(to 0.007 fc/0.07 Lux),足以满足识别生物组织成分的要

求。

IBM 486 PC/AT, RAM 4MB, HD 200MB 适应于图像采集的高速度和大存储用量的要求。

用于图像采集和辅助处理的 DT2851 和 DT2858 插件板是由美国 Data Translation, Inc. 提供的产品。它们具有灰度等级为 0—255(8 位),帧尺寸为 512×512(8 bit)的分辨率,图像采集速度为每帧 0.04 秒,并能存储两帧 512×512×8 bit 图像。8 bit A/D 输入, 16 bit D/A 输出。

图像显示器是日本 SONY 公司的产品,分辨率很高。每帧 200 线,每线 640 像素元。

生物组织图像由 CCD 摄像头摄取后,在 DT2851 暂存,并经 DT2851 和 DT2858 进行一些预处理后送入 IBM 486 PC/AT。然后在 SONY 图像显示器上显示图像,并在计算机的显示器上进行有关操作和处理。

3 图像采集和处理的软件系统

此软件系统的主要组成部分有:

- MS-DOS V5.0 或以上版本操作系统
- WINDOWS V3.0 窗口软件
- OPTIMAS 图像处理软件
- STATWORKS 数据统计和处理软件
- 专用程序

OPTIMAS 是美国 BIOSCAN 公司专为进行图像采集和分析而开发出来的商业图像处理软件,在美国生物工程界享有很高的声誉。该软件能自动识别屏幕物体,并可自动采集二维图像中的许多参数,例如:面积、周长、直径、直线长度、角度,和点的 X—Y 坐标值等。

STATWORKS 是美国 Cricket Software, Inc. 为 Macintosh II 计算机开发的,功能很强而操作容易的数据统计和处理软件包。

专用程序主要是用于根据 OPTIMAS 软件采集的参数计算出其他的有关参数。比如,用 OPTIMAS 采集的胶原纤维横切面积(椭圆)的长径和短径计算出该胶原纤维的轴线与横切面的倾斜角度和方向。然后,将 OPTIMAS 采集的数据进行格式和文体转换,以适于 STATWORKS 软件之用。

此软件系统在操作上的最大特点是充分利用了窗口技术和鼠标技术,使整个操作过程直观和容易。

4 一些技术问题的处理

4.1 定标问题

图像采集系统的定标问题是生物组织结构定量分析的核心问题。它包括两个方面:

(1) 尺寸定标问题

本系统采用了闭环的尺寸定标方式。首先将标尺置于切片或电镜片应放的位置上。然后由 CCD 摄像头摄入,并显示于图像显示器上。最后,操作鼠标器将标尺长度作为标尺在屏幕上显示的长度值而输入图像采集系统中。

(2) 灰度定标和阈值值的设定

本系统将该系统在当前状态下的屏幕物体的实际灰度值作为该系统自动识别屏幕物体时的灰度阈值。从而排除了诸多因素对确定屏幕物体灰度值的影响。

需要指出的是,由于生物组织的复杂性,同一类型的组织成份(如胶原纤维)在同一屏幕上的屏幕物体灰度值往往是不一样的。因此有时需要采用将屏幕划分成若干个小区域分别进行测量的方法;在每一小区内又可随机选取若干个屏幕物体测量和计算其平均灰度值,以此作为系统自动识别屏幕物体时的灰度阈值。

4.2 从二维图像中计算纤维物体的空间位置和方向。请参阅作者在本杂志 94 年第 3 期上的文章“如何从二维图像中得到主动脉中胶原纤维的空间位置?”

4.3 数据的文本格式转换

美国 Apple 公司生产的 Macintosh II 计算机是美国最普及的个人计算机。因而,有着许多软件公司为它开发了许多功能很强的软件。美国生物工程研究中的许多统计分析工作都是由 Macintosh II 计算机完成的。但在生物组织的图像采集和处理方面,IBM PC/AT 计算机却占据着绝对多数的地位。因此,在美国通常形成的格局是用 IBM PC/AT 进行图像处理和采集参数,然后,在 Macintosh II 上进行数据统计分析。为此,美国的一些电子公司设计了一种将 IBM PC/AT 数据盘转换成 Macintosh II 数据盘的装置,解决了两个不同机种的数据转换和共用问题。

4.4 本系统的尺寸测量单位可以小到埃。

假设所用的电镜片是生物组织结构的 10,000 倍,在电镜片放置的地方采用一个 1mm 长的标尺。实际上,在摄像机前的 1mm 长度只为生物组织结构实际尺寸的 $1\text{mm}/10000=1000$ 埃。当 1mm 长的标尺被摄入,并显示在图像显示器上时,假设该屏幕物体的尺寸是 100mm,显然,屏幕物体的 1mm 就为生物组织结构实际尺寸的 10 埃。

高分辨率的图像显示器,每帧 200 线,每线 640 点,因此,其测量单位小到埃是毫无问题的。

5 应用举例

生物组织结构的定量分析是生物医学工程界深入研究生物医学实质问题的重要方法之一。作者在美国 University of California, San Diego 访问期间,在美国工程科学院院士,冯元桢博士的指导下,开发了本文所介绍的系统,并应用于课题“大白鼠心血管组织结构的定量分析研究”之中。该课题主要对主动脉组织的内膜、中膜、外膜及其胶原纤维、弹性纤维、平滑肌细胞、成纤维细胞和它的组成成份进行了定量分析。并写出题为“Monphometry of Collagen, Elastin, Smoth Muscle, and other components in Arterial wall”的论文,送美国 American Journal of Physiology: Heart and Ciculotory Physiology 杂志上发表。实践证明,这一系统是成功的。

参考文献

- 1 O. D. Faugeras and G. Toscani, The Calibration Problem for Stereo, Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition, pp. 15—20, 1986
- 2 R. Hoffman and A. K. Jain, Segmentation and Classification of Range Image, IEEE Trans., Vol: PAMI—9, pp. 608—620, 1987
- 3 M. A. Eshera and K. S. Fu, A Graph Distance Measure for Image Analysis, IEEE Trans., Vol: SMC—14, PP. 398—408, 1984
- 4 L. C. Junqueira et al; Basic Histology. LANGE Medical Publications 1977

本刊启事

为了更快地推动、促进我国计算机产业的迅猛发展,及时地为计算机科研、开发、生产等部门提供计算机行业的最新发展动态,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》杂志社现正在开展优惠出版各类计算机专业技术资料业务,欢迎广大作、译者踊跃赐稿。具体出版业务欢迎来函来电来人商议。联系人:张钢(邮编:610041)

《计算机应用研究》杂志社启

起,同时又保持相对独立。系统结构的框架,如图2所示:

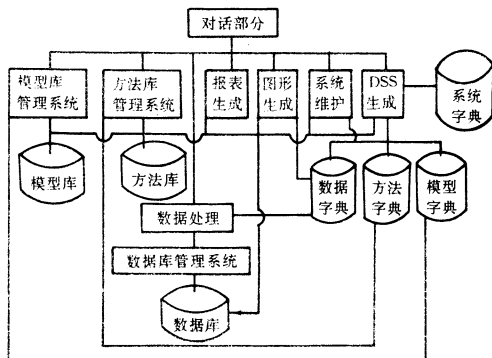


图2 系统结构框图

这个框架既反映出系统的内容,又表示出了系统各个部分之间的联系。由于DSS是以模型驱动的,因此,模型库系统在整个系统中占主导地位,是系统的核心部分。模型库系统与对话部分交互作用,可使用户控制对模型的操作、处置和使用;它与数据部分交互作用,以便提供各种模型所需要的数据,实现模型输入、输出和中间结果存取自动化;它与方法部分交互作用,实现目标搜索、装载运行自动化。

4 用户接口

用户接口是用户与系统进行联系的窗口,以会话方式实行用户与系统的交互,以达到用户对系统的操纵,以及系统对用户的服务和支持。其功能是:提供使用系统的手段和辅助信息。它主要由会话命令及解释程序组成,借助这些命令可转入各子系统。用户以“点菜单”的方式使用这些命令,系统将按照用户的选择,调用相应的命令模块完成其工作。

会话命令有两类:静态和动态。

• 静态命令:是系统定义的,通用命令,按其功能的逻辑关系,被分别组织在不同的静态窗口中。

• 动态命令:是用户定义的,专用命令,这些命令被装入一个动态窗口中。该窗口有一个充分大的虚屏支持,系统根据窗口的大小自动分屏,可按用户的选择进行滚动。

该系统提供一种静动结合,全屏幕、多窗口、分布式菜单。对于用户某一操作流程,相应的窗口按其层次关系分布在物理屏幕上。如图3所示:

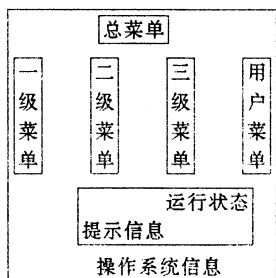


图3 菜单窗口布局

为此,我们采用了回溯技术,将因全屏幕操作而抹掉的菜单窗口予以保留,以待恢复,使光标定位于原来位置。

5 系统特点

该系统具有以下特点:

• 功能齐全界面友好:系统面向决策支持、事务处理和专用DSS(MIS)开发,提供了所需要的各种功能和一个静动结合、全屏幕、多窗口、分布式菜单,界面友好,使用方便。

• 安全可靠:我们采取分别授权,分层保密的保护措施,以保证资源不被侵害。系统程序通过严格的测试,并在实际应用中得到检验和证实。系统还提供了自动检测功能,以保证系统的可靠性。

• 标准化与通用性:我们将系统程序模块化,模型规范化,方法标准化,而且在系统程序与所处理的对象之间,以及“三库”之间,建立了标准的接口,使得系统程序独立于所处理的对象,模型独立于方法,方法独立于数据。使得专用DSS(MIS)的开发变得十分容易,用户只需给出专用DSS(MIS)定义,系统会自动生成专用DSS(MIS)。对开发者来说,几乎没有编程任务,除非他想在输出生成时,做某种特殊处理,在那里为特殊处理程序留有接口。

• 集成性与开放性:我们以支持决策过程为目标,将许多相关的功能有机地组合在一起,构成一个集成系统。另一方面,由于该系统可与操作系统进行通讯,调用外部程序,从而可把所需功能软件集成到系统之中。系统提供了标准的模型、方法和功能。同时,又为用户构造模型、建立方法、定义功能提供手段和支持,并把用户定义的模型、方法和处理功能与标准的有机地结合起来,构成一个开放性系统。

• 自动化与自适应性:我们利用自动生成技术,提高系统的自动化程度和应变能力。面对决策环境不断的变化,采用循环设计方法建立一个自适应系统。使得在短期运行中,系统允许在相对窄的范围内寻求答案;在中期水平上,通过修改和扩充它的能力,让系统“进行学习”;在长期运行情况下,将演变成自适应系统。

6 结束语

该系统面向决策支持、事务处理、专用DSS和MIS开发,可帮助决策者提高决策能力和决策水平,最终实现提高决策质量和效果。同时,适用于专用DSS和MIS类应用系统的开发,可提高系统开发效率和质量,降低开发费用。目前,已有一些单位和部门在使用,取得了显著的经济和社会效益。

该系统已通过省级技术鉴定。与会专家一致认为:该系统设计思想新颖,具有特色,填补省内空白,居国内领先水平。

参考文献

- 1 王亚芬等,决策支持系统. 陕西科学技术出版社,1988:1~215
- 2 孙永生、许宗芬、王连海、孙军,关于模型库系统的探讨. 山东科学,1993;(3):1~7
- 3 R. T. Plant, Expert System Development and Tesing: A Knowledge's Perspective. J. SYSTEMS SOFTWARE, 1992: 19: 141~146
- 4 孙永生、许宗芬、曾凡平、王健美,《信息处理通用程序系统》的设计与实现. 山东科学,1989;(1):6~11

滚轮电气传动智能监控系统

西南交通大学 牵引动力国家重点实验室 (成都 610031) 彭忆强 周文祥

摘要 本文介绍了滚轮电气传动智能监控系统的软、硬件研制过程。该系统能实时监测电气传动系统中各参数的变化,并以图形方式显示系统的运行状态,为操作人员提供了良好的操作及监视条件。

关键词 监控系统 电气传动 滚动振动试验台

1 前言

牵引动力国家重点实验室的滚动振动试验台是进行机车车辆动态模拟试验的关键设备。其中,试验台电气传动系统的作用是对滚轮进行电传动控制,为机车车辆或其它零部件的室内试验提供速度、牵引力或制动力的动态环境。

这种多功能的动态环境对电传动提出了较高要求:

收稿日期:1994 年 4 月 5 日

(1)四个传动机组有电动轻载、电动重载、反接制动、再生制动四种工况,并能灵活转换;

(2)每个机组均能独立运行,并能分组,如 1[#]、2[#] 机组电动运行,而 3[#]、4[#] 机组制动运行,以满足某些特殊试验要求;

(3)既要能控制滚轮的力矩,也要能控制其转速。

同时,因为主电路设备的功率很大,相应的辅助冷却设备多达 20 余个,所以操作步骤多;此外,由于在传动系统中,接入了齿轮箱,对系统的电气参数产生了一些影响,增加了运行过程中人为判断的难度。

HD-COPY 的剖析

浙江工业大学 (杭州 310014) 徐令毅 张 烨

作为一个普通的计算机用户,您也许常常为磁盘间拷贝速度太慢,或者三英寸软盘与五英寸软盘不能直接互相复制而感到烦恼,特别是目前许多大型应用软件的安装盘多为三英寸软盘,对于一些仅有二只五英寸软驱的用户,造成了很大的麻烦。现在有了全新的磁盘拷贝工具 HD-COPY:它兼有 DUP 和 800K 的功能,将帮助你解决难题。

当你第一次拿到 HD-COPY 1.7@版时,会发现它相当精致。运行后出现一个非常友好的界面,同时具有源盘、目标盘、菜单、设置信息、操作及出错信息共五个窗口,当前工作窗口边框加亮,支持鼠标,操作极其简单。据设计者称,用此软件作盘间拷贝只要短短 32 秒,在 OPTION 中选择 FAT SELECTION 为 ON 状态,速度将快于任何拷贝软件。用户有了 HD-COPY,完全可以淘汰 DUP。

HD-COPY 最引人入胜的地方是它强大的格式化功能,选择菜单选项 Format destination,可将磁盘格式化成 1.2MB-1.72MB 等几种不同规格,由于受到驱动器的限制,我们一般只能用到 1.476MB 左右。HD-COPY 最大的优点是扩容后的软盘,在任何一台机器上都可以脱离软件使用,无须任何软件驻留内存。只要第一次读盘失败后重新启动,这台计算机系统就

具备了对这种软盘的识别能力。就笔者使用的经验来看,五寸软盘一般做到 1.44MB 为宜,一方面实现了与三寸盘的通用,一方面因为磁盘扩容的原理主要是降低磁头转速,以便划分出额外的磁道,转速过低,就会降低读盘的可靠性,也会降低读写速度。当格式化完成后,HD-COPY 自动写上一个卷标,以示区别。另外在全盘拷贝时,它会自动识别软盘规格,并制作相同的目标盘,三寸盘和五寸盘完全通用。读入一次,可写多次,避免了复制多份时,要反复读源盘的麻烦。

使用 HD-COPY 时也应该注意:1)程序运行时要在 C 盘根目录下建立一个临时文件 HDCPTMP. \$\$\$,需要不少于 1.37MB 的空间。如果需要将临时文件放在其它路径之下,只要在 HD-COPY.EXE 文件中寻找字符“C:”,改成其他路径名即可。2)退出时不要把软盘留在软驱内,以便系统更新缓冲区。3)建议采用 Safe 的 User's Mode 防止不小心造成的数据损失。

此外 HD-COPY 还有许多附加的功能,如使用清洗盘功能可避免磁头反复读零磁道位置,使清洗盘充分发挥作用。对特别重要的磁盘在复制后可以进行校验,确保质量。HD-COPY 还兼有解密软件的特征,可把缓冲区的数据写入文件。这些不常用的功能就不一一叙述了,对此软件有兴趣的朋友可与浙江工业大学 #312 信箱徐令毅、张烨联系。

收稿日期:1994 年 4 月 6 日

所研制的监控系统能对电气传动系统运行中的主要设备开关状态及主要参数进行监测和记录。而应用窗口技术设计的用户界面,以及彩色大屏幕显示的系统运行模拟图,为操作人员提供了良好的操作、监视环境。

2 系统构成

智能监控系统主要由工控微机、显示器、打印机、信号采集板和相应的系统软件构成。

系统的硬件配置如图1所示。

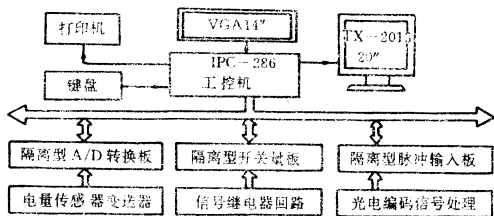


图1 监控系统的构成

主控计算机采用“积木式”的工控微机,其CPU为80286,主频:16MHz。根据电气传动系统的具体情况,对I/O通道进行了如下配置:16个模拟量输入通道;4个脉冲量输入通道;96个开关量输入通道。完全满足了实际应用的需要。

模拟量输入通道主要用于监测电网电压以及电机的电流、电压等;开关量输入板用于监测各主要设备的开关状态;而脉冲量输入板则用于测量电机的转速。

2.1 模拟量的测量

本系统中需测量的模拟量及其数值范围为:

- (1)电枢电压:±850V,DC,0~3通道;
- (2)电枢电流:0~1000A,DC,4~7通道;
- (3)励磁电流:0~200A,DC,8~9通道;
- (4)励磁电压:0~300V,DC,10~11通道;
- (5)电网电压:19~29KV,380V,AC,12~13通道。

对于(1)~(4)项,采用新型的霍尔电流电压传感器。这些传感器的一次侧接入主回路,从与之电隔离的二次侧得到一个与被测电压或电流成正比的电流信号。

对于第(5)项则采用2只TD-1000系列交流电压变送器。将一个变送器的原边并入380V电网,而另一个变送器则并入25KV电压互感器的次边。与前面相似,在它们的次边可得到一与网压成正比的电流信号。

总共14个通道的电流信号用约10m长的双芯屏蔽电缆送到模拟量输入接口板,由该板上的电位器变为10V以下的电压信号,供测量使用。

所有信号,均为差分方式输入。

2.2 开关量的测量

96个开关量的状态检测分两种情况:一种是利用被监视的开关有多余的辅助触点,利用该触点使24V直流电源开断,从而控制信号继电器。一种是没有辅助触点,因而信号继电器线圈只能与辅助触头的其它负载如指示灯并联。信号继电器装在控制室内,利用信号继电器触头的开合,即可使接入接口板光耦原边的24V电源接通或分断。这样,开关的通断信息就转化为光耦的工作与否,进入开关量接口板。

我们把开关量分为中断型开关量和非中断型开关量。

在系统中某些重要设备的开关状态的变化可能引起系统的误动作,产生不良后果,我们将这些开关量归入中断型开关量。在试验过程中实时监测其变化,如发生异常,则引起中断,产生报警信号,使操作人员能及时采取措施。

2.3 脉冲量的测量

脉冲量输入板用于测量电机的转速。

将转速传感器——光电编码器固定于电机转子的一端,随电机的转动,产生300个/转的电脉冲,经过60多米的双绞屏蔽线送至75115接收器,经处理,再送入脉冲量测量板。

电机转速的测量采用M测速法。其原理如下:由脉冲发生器送来的脉冲信号由计数器计数,定时器每隔时间T秒向CPU发出一次中断请求,CPU响应以后从计数器中读出计数值,并立即将计数器清零。根据计数值的大小可求出对应的转速测量值。

设脉冲发生器每转输出的脉冲数为p,在时间T内共发出m个脉冲,则转速为:

$$n = (60 \cdot m) / p \cdot T \quad (\text{转/分})$$

在该系统中: $p = 300, T = 1$ 秒

$$\therefore n = \frac{60m}{300 \times 1} = \frac{1}{5}m \quad (\text{转/分})$$

$$\text{分辨率 } Q = \frac{60}{pT} = \frac{60}{300} = 0.2 \quad (\text{转/分})$$

2.4 系统的抗干扰措施

由于建筑布局,在监控系统周围10m内存在着种类较多的干扰源,例如:可控硅整流柜,开关电源,这些设备与计算机共用一个电源,同时,1000伏左右、1000安以上的脉动电压、电流等,都是不可忽视的。在系统设计时,都作了认真考虑。

针对上述情况,我们主要采取以下措施:

- (1)使用精密净化交流稳压电源为监控系统供电;
- (2)所有信号均通过隔离再进入计算机,有的信号甚至经过两次隔离,如继电器+光耦,传感器+隔离放大器;
- (3)使用双绞屏蔽线进行信号传输;
- (4)在软件中,对输入信号进行数字滤波。

从目前运行情况看,这些措施是十分有效的。

3 系统软件设计

整个系统软件由主控、运行模拟图显示、数据处理、转速信号测量、故障检测五个功能模块构成。后两个模块以中断服务程序的形式编写,保证了系统的实时响应。

主控程序用于控制系统的操作进程:运行模拟图模块用于显示整个电气传动系统的模拟图;数据处理模块用于试验完成后或故障发生后所记录数据的显示及打印。

而转速信号测量模块则是由时钟触发的中断服务程序。在此程序中完成转速信号的采集,并具有超速报警的功能;故障检测程序则是当电气传动系统中某些主要设备的工作状态发生异常变化时而引起的中断。当此类中断发生时,首先判断是由何种设备引起的,然后在故障显示窗口内报告有关信息,并产生报警信号。

下面介绍部份模块的功能。

3.1 主控程序

主控程序框图如下。

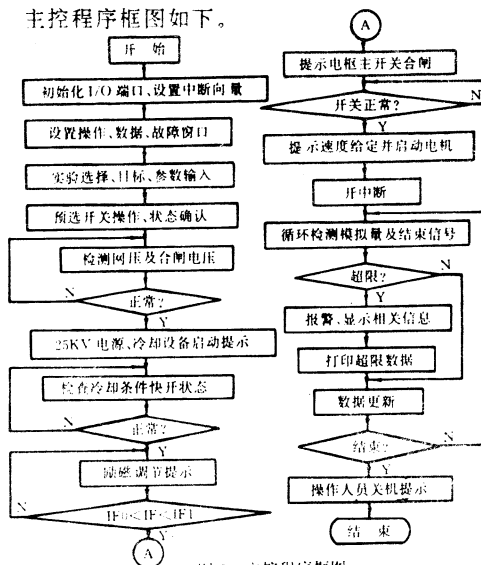


图 2 主控程序框图

在 主 控 程 序 中,首 先 初 始 化 各 I/O 端 口,设 置 中 断 服 务 程 序 的 中 断 向 量,显 示 模 拟 运 行 图 的 初 始 状 态;然 后 在 VGA 显 示 器 上 设 置 操 作 提 示、数 据 显 示 和 故 障 显 示 三 个 窗 口,如 下 图 所 示。

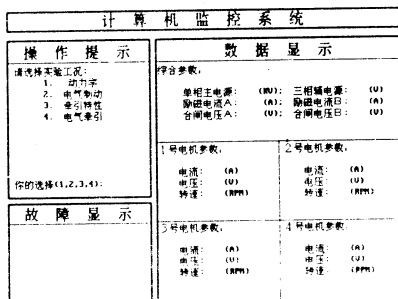


图 3 图形窗口用户界面

操作提示窗口是人机对话窗口,它提示操作人员按一定的步骤操作控制台面板,进行实验选择、给定目标速度等。

而数据显示窗口则实时显示电气传动系统的一些主要参数,如:电网电压、励磁电流、电机电压、电流、转速等;如在运行过程中,系统出现故障,则在故障显示窗口中报告当前系统状态,指出可能的原因。

在确认了所有预选开关的状态后,如网压正常则提示操作人员合闸,并启动冷却设备;如冷却设备工作正常,则提示操作人员调节励磁电流。当励磁电流稳定在一定范围内后,自动提示操作人员给定电机速度,最后启动电机,完成开机过程。

电机启动后,主控程序就循环检测、显示各种模拟量信号,如有超限发生,则产生报警信号,并显示及打印相关信息。

当收到实验结束信号后,主控程序提示操作人员按一定的顺序关闭各种设备,结束程序运行。

3.2 电气传动系统运行模拟图显示

监控系统中的彩色大屏幕图形显示器用于显示整个电气传动系统中各功能环节的图形。随着操作过程的进行,在显示器上,以各种不同的色彩或闪烁光,表示设备的不同运行情况,以形象逼真的方式显示出了系统的运行状态。同时,还实时显示了电机的转速和转速的变化曲线。从此屏幕上,操作人员可以很清楚地了解当前的实验工况,系统的工作状况等。

运行模拟图如下。

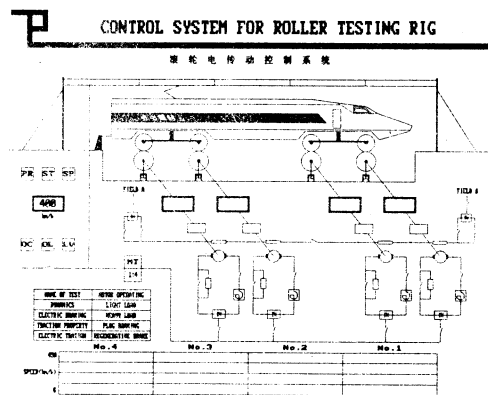


图 4 电气传动系统模拟图

3.3 转速信号测量模块

如前所述,在本系统中转速信号的测量采用 M 法测速。因此,以中断方式进行转速信号的测量。

程序框图如图 5。

此中断程序由一时钟信号触发,每秒钟产生一次中断。进入中断程序后,立即读取脉冲计数值,计算出电机转速,如电机超速则产生报警信号,并显示相关信息;在中断结束前,将计数器清零,打开中断,开始下一次脉冲计数。

一个实用的全屏幕数据录入修改程序

甘肃省计算中心 (兰州 730030) 王雄基

1 问题的提出

用 DBASE 语言设计一个数据录入修改程序,一般都要设计一个格式文件。如果不在程序上做一个特殊处理,则每录入下一条记录,上一条记录就不再保留在屏幕上;修改时也是一样,屏幕上只能显示要修改的哪一条记录。这样就使得在数据的录入修改过程中,前后录入的数据没有联贯性、参照性。如果一个记录的字段很多,每录入一条记录,必须占用一次整屏,这样的设计则是无可非议的。但如果一个记录的段不多,也就是说一个屏幕上同时可容纳很多条记录,这时如果仍用格式文件,采用一般方法录入修改数据就显得不够完满。为了解决这种不足,我们不用格式文件而用命令文件。

2 解决问题的方法

解决上述不足的基本设想是最大限度地利用屏幕的有效面积,使之能够同时容纳更多的记录,所有记录能够前后滚动,能够在屏幕上随时显示第一条记录直至最后一条记录。程序运行时能随意录入数据,录入时后一条记录跟在前一条记录之后,满屏之后,前面的所有记录依次前移一行,新录入的记录处在最后一行。在录入时且能随时进入修改状态。一旦进入该状态,即可在键盘的控制之下,查找库文件中的任意一条记录,方便直观地完成记录的插入、删除及各字段的修改,如不需修改则可随时退出。在上述处理

中,要求屏幕上始终显示出包括需要修改、删除或者插入处记录在内的一段记录,这样就可以给用户提供一个类似于 BROWS 命令的功能且又强于它的数据录入维护功能,使用户在录入维护数据时,根据显示出的一段记录前后关照,上下联系,提高库中记录的准确性。这样的一种设计在某些场合的实际应用中是很有必要的。如在会计记帐凭证的录入维护中,这样的设计会给用户带来很大的方便。

3 程序设计

为了实现上述设想,笔者设计了下述几个程序。

1. 数据录入程序(SJLR.PRG),此为主程序,完成数据的录入功能。

2. 数据修改程序(SJXG.PRG),它由主程序调用,完成记录的查找、修改、插入、删除功能。为了使整个程序具有较强的通用性,笔者设计了三个单独的小程序。它们分别是 PMGS.PRG(屏幕格式)、LRWZ(录入位置)、XSWZ.PRG(显示位置)。它们分别由录入程序和修改程序调用。数据的录入格式、显示位置以及屏幕容纳记录的多少完全取决于这三个小程序。只要根据库结构及各字段的含义改写这三个程序,就可以完成任意一个数据库(每条记录中录入数据的字段长度总数不超过八十列(包括字段间的间距))的数据录入维护工作。

收稿日期:1994年3月6日

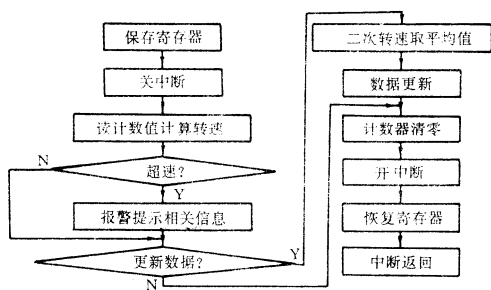


图5 转速信号测量

为使操作者能够清晰地观察到转速的变化,每中断二次,进行一次数据及曲线更新。

4 结束语

智能监控系统的研制成功,为保证电气传动系统

的正常运行提供了良好的监测手段。

首先,它将整个电气传动系统中 100 个左右的开关状态以直观、生动的方式集中在图形显示器上显示,完全由机器监测其变化,大大减轻了操作人员的工作负荷;其次,对所监测的参数均有超限报警功能,使操作人员可以及时处理突发事件,避免事故的发生;此系统另一个突出的优点是具有一定的记忆功能,能将故障前后系统的状态及参数记录下来,为操作人员进行故障诊断及检修提供了可靠的依据。

此外,也为完全实现电传动系统的计算机控制打下了良好的基础。

参考文献

- 1 庄梓新, 测量与控制核心系统手册, 北京: 航空工业出版社, 1989
- 2 苏彦民, 电力拖动系统的微型计算机控制, 西安: 西安交通大学出版社, 1988

4 程序

随文附上全部程序,供感兴趣的同仁参考。

```
* sjxg.prg
para tc
do while .t.
set color to g+
@ 24.1 say space(80)
@ 24.14 say'退出'      向后查找;
    向前查找 选中'
set color to r+
@ 24.14 say'[Q]'
@ 24.25 say'[B]'
@ 24.40 say'[P]'
@ 24.55 say'[Enter]'
set color to g+
set cons off

wait to cx
cx=uppe(cx)
set cons on
if cx='Q'
tc='Y'
exit
endif
if cx='B'
if xgh>qsh
@ xgh.zzwz say''
xgh=xgh-1
set color to W+
@ xgh.zzwz say'-->'
set color to g+
jls=jls-1
else
if jls-1>=jl
jls=jls-1
k=qsh
xgh=qsh
go jls
do while .not. eof().and. k(=zzh
do xswz with k
k=k+1
skip
enddo

set color to W+
@ xgh.zzwz say'-->'
set color to g+
endif
endif
endif
if cx='p'
go bott
ss=recno()
if jls(ss
if zzh)xgh.and..not. eof()
@ xgh.zzwz say''
xgh=xgh+1
jls=jls+1
set color to W+
@ xgh.zzwz say'-->'
set color to g+
else
if jls+1(= dqjl
jls=jls+1
```

向后查找;

```
go jls
k=zzh
do while k(=qsh
do xswz with k
k=k-1
skip-1
enddo
xgh=zzh
@ xgh-1.zzwz say''
set color to W+
@ xgh.zzwz say'-->'
set color to g+
endif
endif
endif
if cx='
set color to g+
@ 24.1 say space(80)
@ 24.20 say'      修改;
    退出 插入'
set color to W+
@ 24.20 say'[Enter]'
@ 24.35 say'[D]'
@ 24.48 say'[I]'
set color to g+
do while .t.
set cons off
wait to cx1
cx1=uppe(cx1)
set cons on
do case
case cx1='D'
if dqjl=jls
go dqjl
dele
pack
dqjl=dqjl-1
jls=jls-1
k=xgh
do while k(=zzh
@ k.1 say space(80)
k=k+1
enddo
xgh=xgh-1
set color to W+
@ xgh.zzwz say'-->'
set color to g+
exit
else
go jls
dele
pack
dqjl=dqjl-1
k=xgh
go jls
do while .not. eof().and. k(=zzh
do xswz with k
skip
k=k+1
enddo
if k-1(zzh
do while k(=zzh
```

```
@ k.1 say space(80)
k=k+1
enddo
endif
set color to W+
@ xgh.zzwz say'-->'
set color to g+
* pmgs.prg
* 屏幕格式
m='.....';
.....;
.....;
bm='项目名称 一季度;
二季度 三季度;
四季度'
set color to r+
@ 1.23 say'==数==据==录==入;
==屏==幕==';
set color to g+
@ 2.3 say m
@ 23.3 say m
@ 3.4 say bt
qsh=4
zzh=22
* qsh:记录在屏幕上的起始行
* zzh:记录在屏幕上的终止行
retu

C)
* lrwz.prg
para h0
* 记录指针位置
zzwz=1
* 记录录入位置
@ h0.4 get b01
@ h0.21 get b02
@ h0.36 get b03
@ h0.50 get b04
@ h0.64 get b05
read
@ h0.4 say b01
@ h0.21 say str(b02,12,2)
@ h0.36 say str(b03,12,2)
@ h0.50 say str(b04,12,2)
@ h0.64 say str(b05,12,2)
* b01,b02,b03,b04,b05 为 z303.dbf;
中的字段名
retu

C)
* xswz.prg
* 记录显示位置
para hh
@ hh.1 say space(80)
@ hh.4 say b01
@ hh.21 say b02
@ hh.36 say b03
@ hh.50 say b04
@ hh.64 say b05
retu

C>
* sjlr.prg
SET TALK OFF
```

```

else
    clear
    close all
    set color to
    retu
endif
enddo
    exit
endif
    case cx1='1'
if xgh<zzh
    go jls
        inse blan
        jls=jls+1
        dqjl=dqjl+1
        xgh=xgh+1
        k=xgh
        go jls
do while ~not. eof() .and. k<=zzh
    do xswz with k
    k=k+1
    skip
enddo
    @ xgh-1,zzwz say ''
    set colo to w+
    @ xgh,zzwz say '->'
    set colo to g+
    @ xgh+1,zzwz say ''
    go jls
    do lrwz with xgh
    exit
else
    go jls
        inse blan
        jls=jls+1
        dqjl=dqjl+1
        go jls
        k=zzh
        do while k>=qsh
        do xswz with k
        skip -1
        k=k-1
        enddo
        set colo to W+
        @ zzh,zzwz say '->'
        set colo to g+
        go jls
        do lrwz with xgh
        exit
endif
case cx1=''
    go jls
    do lrwz with xgh
    exit
    othe
    loop
endcase
enddo
endif
enddo
retu

```


制做带商标的管理软件封面

甘肃省计算中心 (兰州 730030) 午锁平 张海蓉

甘肃省人防办公室 (兰州 730000) 午毅凯

摘要 本文介绍了用 Foxbase 和 C 语言共同开发一种带有商标的软件封面,即在屏幕上既有商标图案又有标题、版本的封面,使软件产品向商品化迈进了一步。

关键词 商标 图形 图案 商品化软件

1 综述

用 Foxbase 语言设计管理软件非常方便灵活、且容易实现,再配上 2.13 系统下的放大汉字显示很漂亮。但对于某些公司、某特定的用户想将自己的商标或特定标志同时显示在屏幕上,仅用 Foxbase 语言是不容易实现的。我们可以利用 C 语言弥补这个不足,使同一屏幕上既显示文字又显示图形或图案装饰,以起到点睛的作用。以下我们以我们为甘肃省医药集团奥博公司设计的软件封面为例,来说明这种制做方法。

2 系统环境的配置

2.1 设备的配置

硬件:普通 286,386,486 以及兼容机和外设

软件:3.0 以上 Dos 版本

汉字系统:2.13H

应用语言:Foxbase 2.1, Turbo-C 2.0

2.2 安装配置

```
config.sys
DEVICE=\dos\vdisk.sys 1024 512 112/e
DEVICE=ANSI.SYS
FILES=20
BUFFERS=24
213.bat
ECHO Off
cls
cd\213
if not exist d:\hzk16 copy hzk16 d:
file3 d2
CCCC
cv26
int10f
prta
file16b
file24a lsfhk
ZF24 3
yx1
lx1
CD\
```

设置虚盘不仅是为在使用 2.13 汉字时提高读取速度,同时也是使 C 语言在 2.13 汉字下可以显示图形。如果虚盘是 d 则在 213.bat 中用 d:,如果虚盘是 e 则在 2.13.bat 中用 e:

3 商标设计

比如要设计此标志:



(甘肃省医药集团奥博公司标志)

笔者使用 Turbo-C 2.0,选择 graphics()函数,然后逐步画出此标志。

例子程序:

```
/* ob.c */
#include<dos.h>
#include<stdio.h>
#include<graphics.h>
#include<stdlib.h>
int x,y;
main ()
{
    int i,driver,mode;
    driver=VGA;
    mode=VGAHI;
    initgraph(&driver,&mode," ");
    setcolor(15);
    setcolor(4);
    /* draw OBC */
    circle(20,30,14);
    circle(50,30,14);
    circle(80,30,14);
    setfillstyle(1,4);
    sector(20,30,90,270,14,14);
    sector(80,30,90,270,14,14);
    bar(36,5,50,44);
    setfillstyle(1,0);
    bar(13,5,14,44);
    bar(73,5,74,44);
    bar(42,5,43,44);
    bar(93,5,95,44);
    /* DRAW i */
    setfillstyle(1,4);
    bar(85,48,88,51);
```

```

bar(85,53,88,62);
/* DRAW o,o */
fillellipse(35,55,7,7);
fillellipse(64,55,7,7);
setfillstyle(1,0);
fillellipse(35,55,4,4);
fillellipse(64,55,4,4);
/* DRAW b */
setfillstyle(1,4);
bar(44,48,49,62);
fillellipse(50,52,4,4);
fillellipse(50,58,4,4);
setfillstyle(1,0);
bar(44,51,50,52);
bar(44,57,50,58);
/* draw a */
line(15,62,26,62);
line(16,61,25,61);
line(17,60,24,60);
line(27,62,22,48);
line(26,62,21,48);
line(25,62,20,48);
line(24,62,19,48);
line(17,53,14,62);
line(18,53,15,62);
line(19,53,16,62);
line(20,53,17,62);
/* draw s */
setfillstyle(1,4);
sector(81,48,180,270,8,9);
sector(78,58,270,360,4,4);
sector(78,58,0,90,4,4);
setfillstyle(1,0);
setcolor(0);
sector(81,48,180,270,3,6);
sector(78,59,0,360,1,1);
setfillstyle(1,4);
bar(73,59,79,62);
return(0);
}

```

其中程序中须调用图形初始化函数 initgraph (&driver,&mode," "),但不使用 closegraph ()函数。否则,在后面的程序中(另一程序)要使用汉字显示时,屏幕显示为西文方式(按 CTRL-F7 方可回到汉字显示方式),无法显示汉字。故不使用 closegraph ()函数,其显示方式不变。

设计上述字母时有一定的难度,如变形字母“A”、“B”或“S”设计时,是将它们分解成几个小块,然后组合在一起,最后设计出此标志。

4 Foxbase 和 C 合用的组合屏幕

4.1 可以直接用 Turbo-C 选用汉字放大技术设计一屏幕,再套用 Foxbase 菜单使用(此程序方法简单略)。

4.2 可以使用 2.13H 的汉字放大与画直线技术设计封面,将 Turbo-c 编译的标志文件 ob.exe 嵌入到 Foxbase 的程序中,且需要在什么地方运行此程序,就

将此 *.exe 插入到什么地方。

例, Foxbase 的封面程序:

* 这是一封面程序

@ 0,0 say chr(14)+'Ro']'

* 运行商标程序

! ob

set colo to w+/b+

@ 4,0 clear to 24,79

@ 0,13 clear to 5,79

@ 0,0 say chr(14)+'C3D1,450B638,450']'

* 0,0 say chr(14)+'V6,1,478,638,476,1,0,4']'

@ 0,0 say chr(14)+'[_ 290|80 * 15 # 9@A 商流软件 V1 * 00']'

@ 0,0 say chr(14)+'C7D290,110B264,1']'

@ 0,0 say chr(14)+'C7D290,115B264,1']'

@ 0,0 say chr(14)+'C15D128,238B394,63']'

@ 0,0 say chr(14)+'C7D126,240B398,67']'

@ 0,0 say chr(14)+'V2,128,237,394,61,1,0,4']'

@ 0,0 say chr(14)+'[_ 135|184 * 8 # 2@D 商品流转处理系统']'

@ 0,0 say chr(14)+'[_ 30|320 * 15 # 1@A 甘肃医药集团 管理技术应用公司']'

@ 0,0 say chr(14)+'[_ 175|320 * 5 # 1@A 奥博']'

@ 0,0 say chr(14)+'[_ 477|320 * 2 # 1@A 版权所有']'

@ 0,0 say chr(14)+'C8D30,350B578,1']'

@ 0,0 say chr(14)+'C8D30,355B578,1']'

a=0

c='8'

x=1

do while a=0

do case

case x=1

c='1'

case x=2

c='6'

case x=3

c='5'

case x=4

c='8'

case x=5

c='4'

endcase

@ 0,0 say chr(14)+'[_ 135|184 * 8 # 2@D 商品流转处理系统']'

if c='1'

c='2'

endif

if c='8'

c='7'

endif

@ 0,0 say chr(14)+'[_ 175|320 * 5 # 1@A 奥博']'

x=x+1

if x>5

x=1

endif

a=inkey()

enddo

5 运行结果

Foxpro 的记录成组访问技术 Rushmore

新疆电子计算中心 (乌鲁木齐 830011) 瓮正科

摘 要 本文较为全面地介绍了 Foxpro 的一项专利技术 Rushmore 的概念,使用规则和应用实例。

关键词 Foxpro 数据库 数据访问 检索

Rushmore 技术是一种数据访问技术,允许高效地进行成组记录访问,其访问速度和单记录索引查询相当。该技术之所以叫 Rushmore,是因为发明者在前一天晚上,浏览了一本 Hitchcock 的“North By North-west”书之后,选择的内部项目名称。

使用 Rushmore 技术,一些复杂的表/数据库操作运行速度和以前的操作相比,可以提高成百甚至成千倍。Foxpro 2.5 可以使得个人计算机可以处理含上百万个记录的表/数据库,而且速度可以与主机(即过去所说的大型机)数据库系统相媲美。

Rushmore 利用标准的 B-树索引,不需要新的任何文件类型和索引。它使用 Foxpro 的索引:标准 (IDX)索引,压缩索引 (CDX)或者组合索引 (CDX)。

实际上,Rushmore 技术并不依赖于新的压缩索引,包括 (CDX)和 (IDX)格式。利用压缩索引的文件,其文件大小只是原来索引的 1/6 大小。压缩索引处理速度快的原因是因为它的文件尺寸,那么,内存和磁盘之间的交换就比较少,从而在使得索引的大部分内容留在 Foxpro 的内存缓冲中。尽管 Rushmore 利用压缩索引文件可以得到许多益处,但它利用老的索引格式也同样有效。为了说明 Rushmore 技术的应用,下面所讨论到的数据库字段名称是这样:这样一个数据库。

CUSTNO	FIRSTNAME	LASTNAME	HIREDATE	ADDR	STATE
顾客编号	名	姓	订货日期	地址	省
012022	gress	jun	01/02/93	no taasd	New York

1 索引、数据库和 Rushmore

Rushmore 技术是用来处理大型数据库的一项技术,对比较少记录的数据库,其优势不是很明显。下面

收稿日期:1994 年 4 月 18 日

是 Rushmore 和索引,多库和单库之间一些关系。

(1)索引和 Rushmore

当 Rushmore 以优化查询索引关键字时,它忽略逻辑操作符 NOT (NOT 或!)。这样大多数类型的表达式也得到优化。例如:

* 索引省名

COUNT FOR state='0H' && 优化

COUNT FOR NOT (state='0H') && 还可以优化

* 索引 on DELETED()

COUNT FOR DELETED() && 优化

COUNT FOR! DELETED() && 还可以优化

Rushmore 不使用在索引表达式中有 NOT 或者! 的索引表达式。当在处理一个大型数据库时,假如机器小,没有足够的内存。这时屏幕将出现一个警告信息: Not enough memory for optimization (对优化没有足够的内存),此时,命令处理的速度将会和以前一样。尽管数据没有丢失,程序也是正确的,但从 Rushmore 中没有得到任何好处。因此,假如处理大型数据库,建议用 Foxpro 的扩展版本。一般数据库记录大于五十万个记录,都建议使用扩展版本。

(2)多数据库 Rushmore

在使用多个数据库时,可以充分发挥 Rushmore 技术。一般对多库查询都使用 SQL 命令,而 SQL 命令中是自动进行优化处理的,Rushmore 技术是 SQL 的一个基本工具。在用 SQL 命令时甚至不需要做打开数据库操作。如果需要索引时,SQL 可以自动建立一个临时索引,供查询时使用。


(3)单表/数据库 Rushmore

在使用单库时,可以在由 FOR 子句的操作中充分

此运行结果以浅蓝色为底色,大红色商标,再配上循环变色的标题和白色的版本,显得很醒目。此时系统处于等待状态,按任一键则进入运行。

参考文献

- 1 叶欣,童长志,张国锋,唐晓菲编译.《Turbo-C++面向对象程序设计》,中国科学院希望高级电脑技术公司,1991,4
- 2 李锐平,“为软件配置良好的用户界面”,《计算机应用研究》,NO3,1993
- 3 《CCBIOS2.13H 汉字系统用户手册》,1990.2



商流软件 V1.00

商品流转处理系统

甘肃医药集团 奥博 管理技术应用公司 版权所有

发挥 Rushmore 的优势。Rushmore 处理查询数据的速度与记录的个数成正比,记录数越多,Rushmore 的优势越明显。与 Rushmore 技术有关的命令如下:

AVERAGE	BROWSE	CALCULATE	CHANGE	COUNT	COPY TO
COPY TO ARRAY	COUNT	DELETE	DISPLAY	EDIT	EXPORT
INDEX	LABEL	LIST	LOCATE	RECALL	REPLACE
SCAN	SORT	SUM	TOTAL		REPORT

在使用优化的 FOR 子句表达式时,命令中要包含范围子句 ALL 或者 NEXT,这样才能充分发挥 Rushmore 的优势。在使用 Rushmore 技术时,缺省的范围子句是 ALL。

在优化时,Rushmore 可以使用除了被滤波和唯一索引的任何打开的索引。为了优化的目的,不要设置数据库的任何次序,可以用 SET ORDER TO 命令去掉数据库的次序。另外在大型数据库时,最好设置 SET DELETED OFF,以提高速度。

2 基本优化表达式

Rushmore 技术依赖于 FOR 子句中基本优化表达式的存在。所谓的基本优化表达式,可定义如下:

基本优化表达式:

〈索引表达式〉〈关系操作符〉〈常数表达式〉

或者

〈常数表达式〉〈关系操作符〉〈索引表达式〉

其中:

〈索引表达式〉必须是和建立的索引表达式完全匹配且不含有别名。

〈关系操作符〉必须是其中之一:〈,〉,=,〈=〉,=,〈〉,≠,!=。

〈常数表达式〉必须是一个表达式,包括与表/数据库无关的内存变量或者字段。

例如,已对 FIRSTNAME, CUSTNO, UPPER(LASTNAME), HIREDATE, ADDR 字段进行索引,则下列是可优化的基本表达式。

FIRSTNAME='Fred'

CUSTNO >= 1000

UPPER(LASTNAME)='SMITH'

HIREDATE<{12/30/90}

如用一个命令 STORE 'WASHINGTON AVENUE' TO X,则下列基本表达式也是可优化的。

ADDR=X

ADDR=SUBSTR(x,8,3)

3 组合基本优化表达式

Rushmore 的数据抽取优化依赖于 FOR 子句表达式。假如 FOR 子句表达式是优化的,无论是用简单或者复杂的 FOR 子句表达式,数据抽取的速度都可以被加速。如果基本表达式是优化的,那么用逻辑操作符 AND,OR 和 NOT 与基本表达式进行组合而形成的复杂 FOR 子句表达式也是优化的。用优化的基本表达式

组合的表达式是完全优化。所用的一个或者多个基本表达式不是优化的,则在构成的复杂表达式可能是部分优化,或者是不优化的。下面的一套规则确定,用基本优化表达式或者非优化表达式组合的一个表达式是否是完全优化,部分优化或者不优化情况,并给出相应的例子。

组合基本表达式规则

NO	基本表达式	操作符	基本表达式	查询结果
1	优化	AND	优化	完全优化
2	优化	OR	优化	完全优化
3	优化	AND	不可优化	部分优化
4	优化	OR	不可优化	不可优化
5	不可优化	AND	不可优化	不可优化
6	不可优化	OR	不可优化	不可优化
7	--	NOT	优化	完全优化
8	--	NOT	不可优化	不可优化

下面是基本表达式组合的一些例子。

例 1:

FIRSTNAME='FRED'AND HIREDATE<{12/30/89}

优化 AND 优化=完全优化

例 2:

FIRSTNAME='FRED' OR HIREDATE<{12/30/89}

优化 OR 优化=完全优化

例 3:

FIRSTNAME='FRED'AND 'S' \$ LASTNAME

优化 AND 不可优化= 部分优化

例 4:

FIRSTNAME='FRED'OR 'S' \$ LASTNAME

优化 OR 不可优化 = 不可优化

例 5:

'FRED' \$ FIRSTNAME AND 'S' \$ LASTNAME

不可优化 AND 不可优化=不可优化

例 6:

'FRED' \$ FIRSTNAME OR 'S' \$ LASTNAME

不可优化 OR 不可优化=不可优化

例 7:

NOT FIRSTNAME='FRED'

NOT 优化 = 完全优化

例 8:

NOT 'FRED' \$ FIRSTNAME

NOT 不可优化=不可优化

除了上面的这些例子,还可以用括号将基本表达式进行分组,上面的这些规则仍然适用。

4 复杂表达式的组合

用复杂表达式组合的更复杂的表达式可能是完全优化的,部分优化的或者是不可优化的。进一步更复杂的表达式组合的表达式又可能是完全优化的,部分优化的或者是根本不优化的。下表是复杂表达式组合的规则,这些规则同样适合于带括号的表达式。

复杂表达式的组合规则

Foxpro 键重定义的方法及技巧

徐州矿务局物资供应管理处 (徐州 221006) 侍永新

键重定义在 Foxpro 编程中有着及其重要的作用,只有灵活运用键重定义功能,才能充分利用 Foxpro 新的特色。例如,Browse 命令是 Foxpro 最为新颖、最为引人注目的特色之一,但若利用 Browse 命令编制出生动活泼的程序,不进行键的重定义是难以做到的。Foxpro 键重定义命令有下面几条:

```
on key Label <key Label ><command>
on key <command>
on ESCAPE <command>
on key = <EXP><command>
```

其中 ON ESCAPE 命令与 Foxbase 功能基本一样,而 on key = 基本上是为了与 Foxbase 兼容

才保留的,该命令已很少使用了。在 Foxpro 编程中使用最多的还是上面两条,下面分别讨论。

1 on key Label 命令的运用

该命令为指定的多个键或鼠标按钮设置陷阱,当任一指定键或鼠标按钮被按下时,陷阱即被触发,程序暂时中断,执行该命令指定的(<command>)。该命令可同时定义多个键的陷阱。该命令主要在 edit、change、browse、read 前使用,有两种使用方式。一是定义当某键按下时执行指定的功能。例如在 browse 命令前定义:

```
on key Label F1 do Deleworker
```

则在 browse 命令期间,若按下 F1 键,则中断当前工作去执行 Deleworker 过程,删除当前记录对应的职工,然后再返回 browse。另一种方式

收稿日期:1993 年 11 月 20 日

NO	表达式	操作符	表达式	结果
1	完全优化	AND	完全优化	完全优化
2	完全优化	OR	完全优化	完全优化
3	完全优化	AND	部分优化	部分优化
4	完全优化	OR	部分优化	部分优化
5	完全优化	AND	不可优化	部分优化
6	完全优化	OR	不可优化	不可优化
7	——	NOT	完全优化	完全优化
8	部分优化	AND	部分优化	部分优化
9	部分优化	OR	部分优化	部分优化
10	部分优化	AND	不可优化	部分优化
11	部分优化	OR	不可优化	不可优化
12	——	NOT	部分优化	部分优化
13	不可优化	AND	不可优化	不可优化
14	不可优化	OR	不可优化	不可优化
15	——	NOT	不可优化	不可优化

下面是组合复杂表达式的一些例子。

例 1:

```
(FIRSTNAME='FRED' AND HIREDATE<{12/30/89})OR(LASTNAME=''AND HIREDATE>{12/30/88})
```

完全优化 OR 完全优化 = 完全优化

例 2:

```
(FIRSTNAME='FRED' AND HIREDATE<{12/30/89})AND 'S' $ LASTNAME
```

完全优化 AND 不可优化 = 部分优化

例 3:

```
(FIRSTNAME='FRED' AND 'S' $ LASTNAME) OR (FIRSTNAME='DAVE');
```

AND 'T' \$ LASTNAME)

部分优化 OR 部分优化 = 部分优化

例 4:

```
('FRED' $ FIRSTNAME OR 'S' $ LASTNAME)OR ('MAIN' $ STREET;OR 'AVE' $ STREET)
```

不可优化 OR 不可优化 = 不可优化

5 Rushmore 技术失效和禁止

在少数情况下,Rushmore 技术对增加数据抽取操作是无效的,在这种情况下,执行的速度和 Foxpro 早期的版本是一样的。Rushmore 技术失效的情况主要是如下几点:

①FOR 子句的表达式不是优化的;

②有 WHILE 子句;

③内存较小的情况,Rushmore 技术不能处理优化命令;

④标准版的数据库记录数大于 50 万个,扩展版没有这个限制。

Rushmore 技术是一个成组记录访问技术,当调用一个利用 Rushmore 技术的命令时,这种技术就首先将满足 FOR 子句表达式的记录组合在一起,然后,再由这个命令去操作这一组记录。显然,它是将大型数据库变成了一个小数据库,对小数据库的操作当然要快。另外,这种技术在访问记录时,先把记录已经成组,假如操作的命令修改了 FOR 子句中的索引关键字,这时 Rushmore 的记录集合可能是过时的,所以在这种情况下,应禁止 Rushmore 的作用,以保证当前数据库的时效性。

在一些高级命令中,有 NOOPTIMIZE 关键字,表示禁止 Rushmore。在整个系统中禁止或者允许 Rushmore,可以通过命令 SET OPTIMIZE 进行设置。命令 SET OPTIMIZE OFF 表示禁止 Rushmore,命令 SET OPTIMIZE ON 表示允许 Rushmore。缺省是允许 Rushmore。

也谈清屏

成都飞机工业公司动力科 (成都 610092) 乐其斌

贵刊于 93 年第 1 期刊登的《一个丰富多彩的清屏程序》(简称《清屏》)中讲述了 8 种有趣的清屏方法。笔者读了该程序之后,便使用 Turbo C 提供的丰富的字符屏幕功能函数及系统中断调用功能函数 `int86()` 编写了另一个丰富的清屏程序,该程序除了实现《清屏》中的所有功能外,还增加了将字符屏幕清成斜边,由此可将屏幕清成任意多边形,还可由此用逼近的方法将其清为圆形(程序清单附后),如果再加上直接写视屏,或将 `gettext()` 函数与 `puttext()` 函数联合使用,则这种清屏将给人一种电视屏幕的感觉。但该程序应在字符屏幕方式下运行,如要在图形状态下实现该程序中的功能,则建议选用 Turbo C 提供的图形函数 `setviewport()` 函数定义窗口,然后用图形函数 `clearviewport()` 清除该窗口即可。该程序在 CGA 与 EGA 上均已通过,如是 VGA,则将 `mode(3)` 改为 VGA 的显示模式即可。

附:程序清单

```
#include <dos.h>
#define BLUE 1
#define GREEN 2
#define RED 4
```

```
#define INTENSE 8
#define BLUE_BACK 16
#define GREEN_BACK 32
#define RED_BACK 64
#define BLINK 128
void mode(), color_puts(), palette(), read_cursor_xy(
), goto_xy();
main()
{ mode(3);
  play1();   cls1();
  play1();   cls2();
  play1();   cls3();
  play1();   cls4();
  play1();   cls5();
  play1();   cls6();
  play1();   cls7();
  play1();   cls8();
  play1();   cls9();
  getch();
}
void color_puts(s,color)
char *s,color;
{ union REGS r; char x,y;
  read_cursor_xy(&x,&y);
  while(*s){
    if(*s=='\n'){printf("\n");s++;x=0;y++;
    continue;}
    r.h.ah=9;      r.h.al=*s++;
    r.h.bl=color;  r.h.bh=0;
    r.x.cx=1;      int86(0x10,&r,&r);
```

收稿日期:1994 年 1 月 12 日

是将某些组合键定义成功能键或其它单键,以便于用户操作。例如 `browse` 的退出命令是 `Ctrl+W`,在程序方式下用此键退出既不方便又不利于用户掌握,若作如下定义:

`on key Label F2 keyboard chr(27)`

则在 `browse` 时可按 `F2` 键存盘退出。需注意的是该命令定义后就一直有效,因此,在不需时应用 `on key Label <key Label >` 命令撤消。

与该命令密切相关的另一条命令是 `keyboard` 命令,它的作用是将命令中的键值放到键盘缓冲区中,就如同从键盘敲入的一样。`Foxpro` 中允许以一种新的方式将一些组合键放入键盘缓冲区中,即如下所示:

`keyboard "{A+B}"`

其中 A、B 分别是键标,这些键标除 `LeftArrow`、`RightArrow`、`UpArrow`、`DnArrow`、`PgUp`、`PgDn`、`Mouse`、`LeftMouse`、`RightMouse`、`SpaceBar` 外,其它基本与键名相同。例如在在定义带多选项的弹出式菜单时,若先定义:

`on key Label SpaceBar Keyboard "{Shift+SpaceBar}"+"{DnArrow}"`

则在菜单弹出时用空格键可选择其中一条菜单,并且光带下移一条,这样的效果比用户用 `Shift+SpaceBar` 选择要好得多。

2 on key 命令的运用

该命令可为任意一键设置陷阱,在程序运行过程中,当有任意一键按下时,陷阱即被触发,控制转移,执行该命令指定的 `(Command)`。这在程序编制时也是很有用的,例如我们在一小窗口中显示人员的编号和姓名,一旦人工欲检索的项出现在屏幕上,则任按一键显示该职工的详细信息,如下程序所示:

```
Set Talk off
Use Worker
Define Window ABC From x1,y1 to x2,y2 in screen
Activate Window ABC
on key Do Disp _Detial
Scan
? WorkerNo+" "+Name
Enscan
```

以上讨论了 `Foxpro` 键重定义的方法及其技巧。只要灵活运用 `Foxpro` 的键重定义功能,就一定能编出丰富多彩的应用程序来,读者不妨一试。

```

x++; goto_xy(x,y);
}
void read_cursor_xy(x,y)
char *x,*y;
{union REGS r;
r.h.ah=3; r.h.bh=0;
int86(0x10,&r,&r); *y=r.h.dl;
*x=r.h.dh;
}
void palette(int pnum)
{
union REGS r;
r.h.bh=1; r.h.bl=pnum;
r.h.ah=11; int86(0x10,&r,&r);
}
Void mode (int mode_code)
{
union REGS r;
r.h.al=mode_code; r.h.ah=0;
int86(0x10,&r,&r);
}
void goto_xy(int x,int y)
{
union REGS r;
r.h.ah=2; r.h.dl=x;
r.h.dh=y; r.h.bh=0;
int86(0x10,&r,&r);
}
play1()
{
register int i,j;
cls(); goto_xy(0,0);
color_puts("This is an interesting clearing
screen program\n",BLUE|RED|BLINK);
for(i=0;i<22;i++){
for(j=0;j<79;j++){
printf("%c",i+'a');
printf("\n");}
}
cls() /* 清屏 */
{
union REGS r;
r.h.ah=6; r.h.al=0;
r.h.bh=7; r.h.ch=0;
r.h.cl=0; r.h.dh=24;
r.h.dl=79; int86(0x10,&r,&r);
}
cls1() /* 清除四个角 */
{
cls(); clsb();
cls(); clsd();
}
cls2()
{
union REGS r; register int x,y;
for(x=0;x<24;){
for(y=79;y>54;){
r.h.ah=6; r.h.al=0;
r.h.ch=x; r.h.cl=y;
r.h.dh=24; r.h.dl=79;
r.h.ah=5; int86(0x10,&r,&r);
x++; y--;
}
}
clsb()
{
union REGS r; register int x,y;
for(x=24;x>0;){
for(y=0;y<24;){
r.h.ah=6; r.h.al=0;
r.h.ch=0; r.h.cl=0;
r.h.dh=x; r.h.dl=y;
r.h.bh=5; int86(0x10,&r,&r);
x--; y++;
}
}
clsd()
{
union REGS r; register int x,y;
for(x=0;x<24;){
for(y=54;y<=79;){
r.h.ah=6; r.h.al=0;
r.h.ch=0; r.h.cl=y;
r.h.dh=x; r.h.dl=79;
r.h.bh=5; int86(0x10,&r,&r);
x++; y++;
}
}
Cls2()
{
union REGS; register int x,y;
for(x=0;x<=39;){
for(g=79;y>=39;){
r.h.ah=6;r.h.al=0;
r.h.bh=7;r.h.ch=0;
r.h.cl=0;r.h.dh=24;
r.h.dl=x;int86(0x10,&r,&r);
r.h.ah=6;r.h.al=0;
r.h.bh=7;r.h.ch=0;
r.h.cl=y;r.h.dh=24;
r.h.dl=79;int86(0x10,&r,&r);
x++;y++;
}
}
cls3() /* 由中间向左右清 */
{
union REGS r; register int x,y;
for(x=39;x>=0;){
for(y=39;y<=79;){
r.h.ah=6; r.h.al=0;
r.h.bh=7; r.h.ch=0;
r.h.cl=x; r.h.dh=24;
r.h.dl=39; int86(0x10,&r,&r);
r.h.ah=6; r.h.al=0;
r.h.bh=7; r.h.ch=0;
r.h.cl=39; r.h.dh=24;
r.h.dl=y; int86(0x10,&r,&r);
x--; y++;
}
}
cls4() /* 由中间向上下清 */
{
union REGS r; register int x,y;
for(x=12;x>=0;){
for(y=12;y<=24;){
r.h.ah=6; r.h.al=0;
r.h.bh=7; r.h.ch=x;
r.h.cl=0; r.h.dh=12;
r.h.dl=79; int86(0x10,&r,&r);
r.h.ah=6; r.h.al=0;

```


Micro-PROLOG 和 DBASE 间的数据传递

吉林省计算机技术研究所 (长春 130012) 李景贵

Micro-PROLOG 和 DBASE 都是功能很强的实用软件,它们各有自己的优点和不足。为了编写更好的实用程序,往往要把它们结合起来一起使用。

在正文介绍之前我们先看几个语句。

1. READ

从文件中读入一项。

? ((READ ABC X)(PP X))

2. OPEN

打开文件

3. WHRITE

4. 下面是如何把 DBASE 库中的数送 PROLOG 中来。

我们希望把 DBASE 库中的数 22,33 送到 PROLOG 中的变量 X。首先在 DBASE 下建立一个库 LLL.DBF;并把 22,33 输入 LLL.DBF 中。之后以 SDF 格式拷入 PPP.TXT 中去。

再退出 DBASE,改名 PPP.TXT 为 PPP.LOG;

• CREA LLL

• COPY TO PPP SDF

>REN PPP.TXT PPP.LOG

>PROLOG 启动 PROLOG.

&LOAD MICRO 装入 MICRO 模块。

? ((OPEN PPP)) 打开 PPP.LOG

? ((READ PPP X)(PP X)) 把数据读入 X,并显示。

22

? ((READ PPP X)(PP X))

33

&

5. 4.5WRITE

与 PP 功能同,只是 WRITE 将 T1..TK 写入指定文件。

? ((WRITE ABC (1 2 3 4)))

如何送 PROLOG 数句到 DBASE:

>PROLOG

&? ((CREATE D))

建文件 D.LOG

&? ((WRITE D(3333 4444 5555)))

3333 4444 5555,其长(=字段长)。

&? ((OPEN D))

打开 D.LOG

&? ((READ D X)(PP X))

读一数看是否已写

入。

3333

&

&QT

退出 PROLOG.

.QT

>

>CD\DBASE

>COPY\PRO\D.LOG

把 D.LOG 拷到 DBASE 子目录。

>REN D.LOG D.TXT

换名 D.LOG 为 D.TXT

>DBASE

• CREA KJ

建 KJ 库

• DO PRO-DB

•

PRO-DB 的内容是:

use kj

注意:kj 中的字段长(=写入时的长度。

append from d. txt sdf

从 D.TXT 追入数据到

KJ.DBF

收稿日期:1994 年 5 月 3 日

```
r. h. bh=7;    r. h. ch=12;
r. h. cl=0;    r. h. dh=y;
r. h. dl=79; int86(0x10,&r,&r);
x--; y++;}
}
cls5() /* 由上下向中间清 */
{
union REGS r; register int x,y;
for(x=0;x<=12;){
for(y=24;y>=12;){
r. h. ah=6;    r. h. al=0;
r. h. bh=7;    r. h. ch=0;
r. h. cl=0;    r. h. dh=x;
r. h. dl=79;   int86(0x10,&r,&r);
r. h. ah=6;    r. h. al=0;
r. h. bh=7;    r. h. ch=y;
r. h. cl=0;    r. h. dh=24;
r. h. dl=79;   int86(0x10,&r,&r);
x++; y--;}
}
}
cls6() /* 由上向下逐行清 */
{
register int i,x; union REGS r;
for(x=0;x<=24;){
r. h. ah=6;    r. h. al=0;
r. h. bh=7;    r. h. ch=0;
r. h. cl=0;    r. h. dh=x;
r. h. dl=79;   int86(0x10,&r,&r);
for(i=0;i<50;i++){x++;}
}
}
cls7() /* 由下向上逐行清 */
{
```

```
register int i,x; union REGS r;
for(x=24;x>=0;){
r. h. ah=6;    r. h. al=0;
r. h. bh=7;    r. h. ch=x;
r. h. cl=0;    r. h. dh=24;
r. h. dl=79;   int86(0x10,&r,&r);
for(i=0;i<50;i++){x--;}
}
}
cls8() /* 由左向右逐列清 */
{
register int i,x; union REGS r;
for(x=0;x<=79;){
r. h. ah=6;    r. h. al=0;
r. h. bh=7;    r. h. ch=0;
r. h. cl=0;    r. h. dh=24;
r. h. dl=x;    int86(0x10,&r,&r);
for(i=0;i<50;i++){x--;}
}
}
cls9() /* 由右向左清 */
{
register int i,x; union REGS r;
for (x=79;x>=0;){
r. h. ah=6;    r. h. al=0;
r. h. bh=7;    r. h. ch=0;
r. h. cl=x;    r. h. dh=24;
r. h. dl=79;   int86(0x10,&r,&r);
for(i=0;i<50;i++){x--;}
}
}
```

参考文献:

- 1 DOS/BIOS 使用详解. 电子工业出版社,1989 年 4 月
- 2 Turbo C 使用大全. 北京科海培训中心,1990 年 4 月

硬盘的管理维护与常见故障处理

安徽省计算中心 (合肥 230031) 郭颖格

随着计算机技术的飞速发展,硬盘驱动器的发展也十分迅速,新的硬盘层出不穷。硬盘是计算机系统中系统资源和信息资源的重要存储设备,为了避免这些资源遭到破坏或丢失,应高度重视对硬盘的管理。这对于公用计算机尤为重要。硬盘故障的维修日益被重视,并对维修工作者提出了更高的要求。笔者根据自己多年来维护硬盘的经验,试图从硬盘管理的有关知识、常见故障的出现与排除、如何正确使用硬盘、管理硬盘等方面进行阐述,供同行们参考。

1 硬盘管理的有关知识

硬盘存储系统通常由硬盘驱动器、硬盘控制卡及连接电缆组成。目前使用的多为温盘,又称为固定盘。硬盘驱动器由头盘组件(HDA)与印刷电路板组件(PCBA)两部分组成。硬盘驱动器是一高精密部件,大部分组件都密封在一个金属腔体中,用户在使用中不需作任何调整。硬盘与主机之间建立联系的界面是硬盘控制卡(又称硬盘控制适配器)。控制卡有这些功能:对硬盘的辨认;磁头定位;写入信号的编码和读出信号的数据分离与译码;格式化处理;资料区段的识别;读出数据的校验及自动纠错;坏磁道的替代及坏扇区的标记等。硬盘驱动器有不同的接口形式,硬盘控制卡也就有不同的接口,配置时必须使两者一致。

常用的硬盘有 5.25 英寸、3.5 英寸、2.5 英寸及 1.8 英寸等。其容量一般为十兆位至几百兆位。接口类型有 ST506 接口、ATBUS 接口(又称为 IDE 接口)、ESDI 接口及 SCSI 接口等数种。前两种使用较多。ST506 接口多见于 5.25 英寸盘,是较早的接口。ATBUS 接口多见于 3.5 英寸盘,只能用于 AT 机而不能用于 XT 机。硬盘的性能指标主要有:盘径、接口类型、磁头数、每磁道扇区数、数据传输率、柱面数、磁盘转速、电源、重量等。国内常见的硬盘有 Seagate 系列硬盘、型号为“ST”开头,例如:ST125、ST138……;Conner 系列硬盘,型号为“CP”开头,例如:CP3000、CP3024……;Nec 系列硬盘,型号为“D”开头,例:D3142、D3861……。它们的型号不同,容量、尺寸、高度、平均存取时间、接口均不同。

硬盘的接口有:ST506/412 接口,最多支持四个硬盘,最大空间为 150MB。PC/XT 中使用这种接口(例 5150 控制卡)。PC/XT 中使用 5280 卡,缺点是容量难以做大。它的传输速率为每秒 5MB。该接口由 34 根命令线和 20 根数据线两部分组成。ST506 接口的硬盘控制器有两种形式,分别用于 AT 机和 XT 机。用于 AT

机的还兼作软盘驱动器的控制器。由于两者的插座有所不同,连接时应当注意。ST506 接口的硬盘一般有四组插座。(1)电源插座。是一 4 针插座,供给电源,反向是插不进去的。(2)控制线插座。是一 34 针插座,与扁平线连接,一般带有颜色的一边对应于“1”脚。(3)类型设置跳线槽。有四至八对,用于设置工作方式,如设主硬盘和从硬盘,若要改变设置,需参考硬盘手册。(4)数据线插座。是 20 针插座,数据线扁平电缆。控制线电缆有两种形式,一种是接一个硬盘,另一种是连接两个硬盘的。ATBUS 接口是当前新型硬盘驱动器普遍采用的一种接口。传输速率比 ST506/412 高,可达每秒 7.5MB 以上,注意 DOS 版本不能低于 3.3。这种接口只有三组插座,电源插座(4 针)、控制线及数据线插座(40 针)和类型设置跳线槽。连接两个驱动器时,主从关系由驱动器上的跳线决定。ESDI 接口,为增强型小设备接口,传输速度每秒 10MB,直接接到硬盘驱动器上,并控制基本操作,如寻道和磁头选择等。最多可支持四个硬盘驱动器,硬盘空间最大 600MB,由 34 根命令线和 20 根数据线组成。SCSI 接口,是一种系统级的接口,可以同时接到各种不同设备的任何一种(如硬盘驱动器、光盘驱动器、磁带驱动器、扫描仪和打印机等)上,并通过高级命令与它们进行通讯,传输速率达每秒 10MB,最多支持 32 个硬盘,硬盘空间可达 1GB,由 50 根命令线和数据线组成。

2 硬盘的参数设置和格式化

安装一个新的硬盘,需要对主机进行硬盘类型的设置,此外,当计算机发生某些故障时,也需重新进行硬盘类型的设置。因此维护人员必须知道所用机器的硬盘型号、容量、磁头数、柱面数及每磁道扇区数。

XT 类型的计算机,我们单位有一批,其硬盘参数是由硬盘控制器上的跳线或 DIP 开关控制的,所以平时不需重新设置。286 以上的计算机,硬盘参数保存在主机板上的 CMOS 存储器中,需要用 SETUP 程序设置。大多数计算机的 ROMBIOS 中均含有 SETUP 程序,可以在启动计算机时用 Ctrl+Alt+Esc 键、Del 键(不同牌号微机的此键不相同,可查看主板使用说明)等,进入 SETUP 状态,根据所用硬盘的参数(柱面数、磁头数、每磁道的扇区数)选择一个适当的类型即可。用高级诊断程序的 SETUP 功能也可进行这一工作,有 47 个硬盘号。其中 1 至 15 为 IBM PC/AT 规定的标准硬盘类型号,兼容机均与此相同。15 号以上的类型号,不同厂家有不同的定义,SETUP 程序或说明

书有提示。类型 47 可由用户定义,输入所用硬盘参数,如选择的类型号所规定的硬盘参数小于实际参数时,硬盘仍可正常工作,只是容量达不到额定容量而已。类型 16 以后没有统一的标准,所以不同的机器虽然类型一样,可能代表不同的硬盘,因此知道硬盘的参数是一个关键。读者要找出被安装机器与其匹配的类型,每个类型的参数机器不一定都能显示,需查阅该机器手册。笔者在为一个用户安装 AST386 机上 120 兆硬盘时,就是用这种方法,用 DM 安装成功的。AST P I 自行设置的类型是 88 或 89,不是 47。大容量硬盘现在出来比较多,硬盘本身带有自己的硬盘管理软件,可用该软件管理,如果没有,只好自己寻找。低版本 DOS(比如 Dos3.3)对于 200M 硬盘大于 1024 个柱面就不能全部管理,只能管理 1024 个柱面,约 159M,需用较高的磁盘管理软件,例如 ST 公司的 4.2 版本的 DM,因为这种磁盘管理软件中包含扩展的硬盘 BIOS。

新硬盘及发生严重故障或被某些病毒感染而不能工作的硬盘必须经过低级格式化、分区和 FORMAT 格式化后方能工作。低级格式化的作用是设置硬盘的工作参数,如交叉因子、坏块表等,同时它还将清除硬盘中原有的所有数据。分区的作用是将一个物理盘分成一个或多个逻辑盘,以适应 DOS 对硬盘容量的限制和不同操作系统的需要。FORMAT 格式化的作用是清文件登录表(FAT)和检查坏的扇区,做低级格式化必须使用专门的格式化软件,如 LOWFORM、DM、QAPLUS 及 IBMAT 高级诊断程序等,某些计算机的 ROMBIOS 中含有低级格式化程序,也可使用。低级格式化软件使用起来都类似,例如 IBMAT 高级诊断盘插入 A 驱动器后启动,屏幕显示出菜单,选择 0,SYSTEM CHECKOUT,回车后显示当前配置清单,键入 Y 后,回车屏幕显示,选择 0,RUN TESTS ONE TIME,回车后屏幕显示菜单,选择 17 后回车,屏幕显示硬盘诊断菜单,这时选择 7,FORMAT MENU 回车后屏幕出现格式化菜单,选择 2,UNDITIONAL FORMAT,接着屏幕显示将破坏硬盘上原有数据,按 Y 键后,再次提醒用户,询问是否输入硬盘上有缺陷的磁道。ST506 接口的硬盘在制造过程中往往会产生几个坏磁道,其磁道号都记录在硬盘外壳的标签上。若需要输入,则键入 Y,然后按屏幕提示将这些坏磁道一一输入,以便低级格式化时将其剔除。近几年生产的 AT EMBEDDED 硬盘则一般无此问题。所以可检查一下硬盘的外壳,若无标签或标签空白,则键入 N,回车后低级格式化开始。根据硬盘容量、速度不同,低级格式化需几分钟至十几分钟或更长。完成后可用 DOS 软盘启动系统,对硬盘进行分区。

分区将硬盘分为几个不同的存储区域,每一个区域称为一个分区。目前 DOS 系统所能支持的硬盘容量是有限制的。例 DOS3.3 最多只能支持 33MB。超过

33MB 的硬盘必须通过分区将其分为多个逻辑盘(每个逻辑盘均不超过 33MB)后才能使用,对于小于 33MB 的硬盘,只是分为一个区。硬盘分区后,可将不同的分区用于不同的操作系统,给用户使用带来方便。分区可用 DOS 的 FDISK 软件完成,也可用 DM 等软件完成,过程都类似。例将含有 FDISK.EXE 的系统软盘插入 A 驱动器,启动后键入 FDISK,回车后出现菜单,选择 1,即建立 DOS 区。这时出现 1 为建立基本 DOS 分区,即用于启动系统的分区(C 盘),2 为建立扩展 DOS 分区,扩展 DOS 分区可用来建立逻辑盘。先选择 1,建基本 DOS 分区,问(Y/N)……? 一般硬盘在 30MB 以下时,回答 Y,即 DOS 分区使用尽可能大的空间,否则回答 N,接着输入所希望的柱面数,这样基本 DOS 分区已建立,按 ESC 退出,回到主菜单再键入 2,便可建立扩展 DOS 分区,如要把剩余的硬盘空间全部作为扩展 DOS 分区,可直接回车,否则输入分区大小后回车。建完扩展 DOS 分区,屏幕显示再回车用扩展 DOS 分区建立逻辑盘,这时输入所需逻辑盘的大小或直接回车可建立一个或多个逻辑盘,且盘号分别为 D、E……对于这时分区完后,还要选择主菜单中第二项,将基本 DOS 分区设置为活动分区。这时可用 DOS 软盘重新启动计算机,并用 DOS 的 FORMAT 程序对各逻辑盘逐一格式化,只要键入格式化命令:FORMAT [d:.]。对于基本分区(C 盘),由于需要用其启动系统,应装入 DOS 系统文件,所用命令:FORMAT c:/s 对各逻辑盘进行格式化后,机器便可启动使用。它的作用是对硬盘初始化,使其记录格式能为 DOS 所接受;检出硬盘有缺陷的磁道;建立目录文件分配表;装入 DOS 系统文件(对其本分区)。

3 硬盘故障的分析与排除

由于受病毒感染,有时计算机需做低级格式化,但硬盘经格式化后,感觉读写速度较慢,经分析,是因没有正确选择交叉因子引起的。笔者在对兼容 386 机作低级格式化时,菜单中第四项选择交叉因子(CHANGE INTERLEAVE)对硬盘工作性能有很大影响。硬盘的物理扇区在磁道上连续排列的,而格式化后的扇区(逻辑扇区)是交叉排列的,连续物理扇区是对应着不连续的逻辑扇区,交叉因子即每两个连续逻辑扇区所间隔的物理扇区数。硬盘读完一个逻辑扇区后,需等待主机将该扇区的数据处理完后才能再读下一个逻辑扇区,在等待过程中,硬盘可能已转过了几个物理扇区。如果交叉因子选择较小,则代表第二个逻辑扇区的物理扇区已转过磁头,需等磁盘再转一圈后才能将其读出,如果交叉因子选择较大,则代表第二个逻辑扇区的物理扇区还未转到磁头处,需继续等待。因此选择合适的交叉因子可使待读的逻辑扇区之间不产生等待时间,明显提高硬盘的读写速度。笔者在格式化前选择不同的交叉因子,试验比较格式化时间的长短,

即可确定。一般 286 以上的计算机可选择较小的数值 (2 或 3), XT 机一般为 6。

一例硬盘卡壳现象。我单位一台 LC0530 机, 使用步进电机定位的 ST251 42MB 硬盘, 故障发生时硬盘不能启动, 经拆机分析发现硬盘的主动马达不转动, 马达启动阻力大, 卡壳。小心地将硬盘侧立, 清除灰尘, 用手轻轻拨一下马达转盘, 再启动马达转起来了, 读写正常。原因是用步进电机作磁头定位的硬盘机, 无复位弹簧, 断电前一瞬间磁头处于什么磁道位置, 断电后磁头仍然停在老地方, 下次启动时, 磁头则在原来的磁道上, 经过一段时间与盘片摩擦之后才能浮起来, 这种硬盘机, 为保护好“00”道, 必须借助于软件 Park 等, 使磁头复位, 另外, 硬盘要保持清洁, 否则硬盘容易卡壳。现在用音圈电机或力矩电机作磁头驱动的硬盘, 只要一断电, 在复位弹簧的作用下, 磁头立刻快速转移到陆区, 陆区为数据区之外最靠近主轴盘片表面区域, 此区域线速度较低, 摩擦不剧烈, 加之不记录数据, 不影响正常使用。硬盘引导失败。故障多半是因感染了病毒, 被病毒所破坏或其它软件故障引起的。首先检查机器的参数设置是否被误操作丢失, 或改变过, 若不符应将其重新按正确设置改正。如设置参数正确, 则有可能是“00”道信息被破坏。在格式化时, 若发现有不适合于数据记录的“坏块”, 就将“坏块”的有关参数(磁道号)、相对索引的位置以及“坏块”的长度制成表格存放在“00”道上。另磁盘操作系统的系统文件、用户文件目录也放在“00”道上, 系统启动后, CPU 将“00”道上的信息调入内存, 以备随时使用。如“00”道信息受病毒感染严重时要作格式化处理。也可能用户把软盘系统拷入硬盘, 造成与原 DOS 版本不同, 微机启动时必须读取的三个文件: 两个隐含文件 IBMBIO.COM、IBMDOS.COM 和一个一般属性的文件 COMMAND.COM 与硬盘格式化时装入的这三个文件版本不同, 造成不能启动, 显示: Incorrect Dos Version。这时只要将原正确的 DOS 版本两个隐含文件重新传送至硬盘, 即: A>SYS C: 回车, 然后将 COMMAND.COM 文件重新拷入硬盘即可。另外在出现故障时, 用户还应细心地检查, 如看看硬盘根目录下 COMMAND.COM 文件是否存在, 是否被删除或被修改(病毒感染时, 字节数会加长), 用 FDISK 看看硬盘 DOS 分区激活状态正确否, 若出现多个呈激活状态的分区, 也会出现故障。为了防止病毒破坏, 用户平时可备份一下正确的主引导区(硬盘的 0 面 0 柱 1 扇区), 当硬盘引导区出现故障时, 重新装入即可恢复。还有一种情况, 可能是 IBMBIO.COM 和 IBMSYS.COM 文件不在硬盘数据区的最前端。可用

PCTOOL 软件查看, 并将硬盘数据区最前端的其它文件删除, 然后再进行传送系统文件工作。若仍不能启动, 则只有进行 FORMAT 格式化。如果硬盘用 ADM 或 DM 等软件分区的, 则应检查 C 盘根目录中的 CONFIG.SYS 文件中是否有“DEVICE=ADM.SYS”或“DEVICE=DM.BIN”等语句, 并保证根目录中确有 ADM.SYS 或 DM.BIN 等文件。

4 合理使用硬盘

合理使用硬盘, 重在加强管理和维护。硬盘是集精密机械、微电子电路、电磁转换为一体的比较娇贵的外存储装置, 为避免资源遭到破坏, 应重视管理, 对公用计算机机房更为重要, 应制定一套硬盘管理的规章制度, 严格执行。

在公用机房, 当硬盘已装有 DOS 系统软件时, 不要再用系统软盘启动计算机, 更不要用与硬盘、版本不同的系统软盘启动, 以免造成混乱, 使计算机不能正常运行。对公用计算机, 各人应在硬盘中建立自己专用的子目录, 所有工作尽可能在各自的子目录下进行。对于硬盘中公用软件(如一些应用软件), 也应分别建立子目录。这样可以避免根目录下文件太多, 不便管理。同时也可避免破坏他人的文件或公用软件。应禁止非管理人员随意向硬盘内拷贝系统文件或对硬盘进行格式化, 禁止随意删除硬盘内文件, 以免造成系统资源的混乱或破坏硬盘内的文件。禁止玩游戏, 因游戏盘往往带病毒多, 一旦病毒感染, 破坏后果严重。公用计算机还应建硬盘文件目录档案, 并经常清理, 删除已不用的文件, 以免积累文件过多。否则存储空间被浪费, 不便管理。对硬盘中的文件应经常用软盘备份, 以免万一因硬盘故障而造成损失, 使用软盘备份可用 DOS 的 BACKUP 命令。从软盘恢复到硬盘上用 DOS 的 RESTORE 命令。在使用计算机时, PC 机系统的故障 40% 是硬盘, 而硬盘关键要保护好“00”道(TRK“00”)。虽然采用了温彻斯特技术, 可靠性指标有了突破性提高, 断电后磁头能迅速回到陆区, 以保证“00”道不在使用中遭破坏, 但 40MB 以下的硬盘, 每次关机前一定要用 Park.COM 命令锁好磁头。

另外使用环境要保持卫生, 减少空气中的含尘量, 必要时将电路板拆下, 清除板上及主轴电机转子部分的灰尘。硬盘的外壳一般有接地插片, 切不可加电启动后随意插拔。在拆装过程中, 不要用手接触印刷电路板的焊点, 因人手上会有一定的静电。有的橡皮或塑料垫板工作台上有数百伏静电, 会损坏器件。工作时或关机后, 主轴电机尚未停转之前, 严禁搬动机器, 以免磁头与盘片撞击, 擦伤盘片表面磁层。

WPS 常见的故障与处理

华中理工大学 (武汉 430074) 周宏甫 宾鸿赞

摘要 本文概括了 WPS 排版系统使用中的一些常见故障问题,提出了 WPS 排版系统中的一些常见故障的处理方法。

关键词 计算机 排版 WPS 系统

1. 缺文件:

1)File Spdos com not found(金山汉字的显示模块找不到)

处理:运行正确的 spdos.com 或拷入 spdos.com 后,再运行 spdos.com 程序。

2)Super-CCDOS 汉卡没有安装(缺汉卡或文件 SPLIB.COM 或 CHLIB.COM)

处理:安装汉卡或运行 SPLIB.COM 或 CHLIB.COM

3)文件 WPS1.ovl 找不到(覆盖文件 WPS1.ovl 找不到)

处理:拷入 WPS1.ovl

4)文件 WPS2.ovl 找不到(覆盖文件 WPS2.ovl 找不到)

处理:拷入 WPS2.ovl

5)源文件找不到(文件服务时源文件名找不到)

处理:键入正确的源文件名

6)文件不存在(在 WPS 操作里读文件时找不到文件)

处理:键入正确的文件名

2. 版本不兼容

1)WPS 所要求的 CCDOS 版本与当前的 CCDOS 版本不兼容

处理:拷入与当前的 CCDOS 版本一致的 WPS 程

收稿日期:1994年2月26日

M2724 打印机故障诊治一例

国防科技大学二系 (长沙 410073) 陈德明

故障现象 一台 M2724 中英文打印机,打印文件时,打出一些乱字符,与欲打印的文本有些相符,有些则不符。

故障分析 这种问题应出在硬件上。可能的部件是:打印机本身;打印卡;接口电缆。

故障诊治 先关机并使打印机脱离计算机,在按住打印机面板上的“选择”开关的同时,开通打印机电源,使打印机自检,结果发现,自检打印的样张正确无误,表明故障不在打印机本身,而在接口部件。M2724 打印机有一个十六进制打印功能,能打印从计算机发送的数据的十六进制码,可以方便地用来检测从计算机传来的数据是否正确。关掉打印机,联上计算机后开启计算机,再在按住打印机面板上的“选择”开关和“换行”开关的同时开通打印机电源,使打印机进入十六进制打印方式,从计算机向打印机发送数据,打印一个 ASCII 码文件,该文件内容是 ASCII 码从 33 到 125 的连续西文字符,结果如下:

```
21 22 23 24 25 26 27 20 21 22 23 24 25 26 27 30 !"# $%&'()*+,-./:
31 32 33 34 35 36 37 30 31 32 33 34 35 36 37 40 123456701234567@
```

```
41 42 43 44 45 46 47 40 41 42 43 44 45 46 47 50 ABCDEFG@ABCDEFGP
51 52 53 54 55 56 57 50 51 52 53 54 55 56 57 60 QRSTUVWQRSTUWV'
61 62 63 64 65 66 67 60 61 62 63 64 65 66 67 70 abcdefg' abcdefgp
OD OA 71 72 73 74 75 76 77 70 71 72 73 74 75 OD ..qrstuvwprstu.
OA
```

显然,打印机接收到的数据与发送数据不符。用

DEBUG 显示的文件数据如下:

```
21 22 23 24 25 26 27 28-29 2A 2B 2C 2D 2E 2F 30 !"# $%&'()*+,-./:
31 32 33 34 35 36 37 38-39 3A 3B 3C 3D 3E 3F 40 123456789,;<=>? @
41 42 43 44 45 46 47 48-49 4A 4B 4C 4D 4E 4F 50 ABCDEFGHIJKLMNOP
51 52 53 54 55 56 57 58-59 5A 5B 5C 5D 5E 5F 60 QRSTUVWXYZ[\]^_`
61 62 63 64 65 66 67 68-69 6A 6B 6C 6D 6E 6F 70 abcdefghijklmnop
OD OA 71 72 73 74 75 76-77 78 79 7A 7B 7C 7D OD ..qrstuvwxyz({}
OA 1A 20 20 20 20 20 20-20 20 20 20 20 20 20 ..
```

对比打印机收到的数据与发送数据的差别,可以发现一个规律,每 16 个字符中就有 8 个字符不正确,且不正确的 8 个字符错成前 8 个字符。对比 ASCII 码值可知,每个字符对应字节的倒数第 4 位(即 D3 位)总是为零。问题很可能出在数据传输通道的 D3 位数据线。关机后拆开接口电缆的接头,发现靠计算机的一头的一根线的焊点已脱落,这可能是因频繁拔插和搬动造成的。重新焊好以后试机,故障已排除。

收稿日期:1994年4月2日

序

要求: Super CCDOSS V5.0 配 WPS2.0

Super CCDOSS V5.1 配 WPS2.1

Super CCDOSS V6.0 配 WPS3.0

2) WPS 的覆盖程序与 WPS.EXE 不兼容

处理: 拷入与 WPS.EXE 一致的 WPS1.com 和

WPS2.com

3. WPS 内存错误

内部错误时退出 WPS, 并显示错误地址

内部错误的原因:

1) 被编辑的文件包含有非法的字符或非法的控制

符

2) 被编辑的文件为 WPS 格式文件

3) 行长超出指定范围

4) WPS 设计错(如左边界大于右边界)

处理: 将 .BAK 文件改名后调入, 抛弃有内存错误的文件, 或重新拷入正确的 WPS 系统程序。

5) WPS 系统程序破坏, 如 FAT 表中的某些簇丢失

4. WPS 被非法终止

死机时压 CTRL+REAK 造成

处理: 再重新进入 wps, 并将临时文件%A%与%B%读入

5. 内存不够(Program too big to fit in memory)

出现在 WPS 启动, 增加窗口, 模拟显示及打印。

内存不够时可用 CHKDSK 命令检查内存。如对

DOS3.30 版本

640K 内存空间检查, 其结果为:

655360 内存空间 可用空间

开机后 5958496

装字库后 358720

装打印程序后 232224

反复模拟显示后 84032(提示内存不够)

处理:

1) 增加计算机内存量

2) 减小 config.sys 中的 buffers 与 files 的数值

3) 重新拷贝 wps 文件, 由于 wps 的文件头破坏, 使 wps 开辟的内存空间太大

4) 内存中有病毒程序驻留, 消灭病毒

5) 除去汉字输入法模块

6. 建立文件错

产生于 wps 的启动, 增加窗口或文件服务功能

原因与处理:

1) 磁盘空间不够, 用 CHKDSK 检查磁盘, 若确实为磁盘空间不够, 将备份文件与不必要的文件删掉, 以增加磁盘空间。

2) DOS 引导时, config.sys 文件所规定的 files 数

值太小或根目录下无 config.sys 文件, 可将 files 改为 files=20, 或根目录下建立 config.sys 文件。

3) 文件名及路径, 驱动器非法, 检查后键入正确的驱动器名, 路径名与文件名。

7. wps 运行太慢

由磁盘碎头与病毒程序引起

处理: 将硬盘备份后, 格式化, 再重新拷入备份盘

8. wps 非正常退出时文件的恢复

非正常退出由于停电、CTRL+BREAK 中断, 死机后造成。

处理: 检查临时文件, %A%, %B% 与后备文件, BAK 换名后, 在 wps 下用读文件的方式与原文件连接。

9. 死机

1) 显示金山汉字版本时死机

原因: .SPDOS.COM 破坏

.SPDOS/参数, 键入的参数不正确, 如 MON 键或 VGA

处理: 覆盖 SPDOS.COM 文件或键入正确的参数

2) 显示 wps 提示时死机

原因: wps.com 文件破坏或 WPS1.OVL, WPS2.ovl 破坏

处理: 拷入正确的 wps 系统文件

3) wps 文件编辑中死机

原因:

· 键入了非法的字符

· 系统硬件不稳定

· 系统软件或 wps 文件被破坏

· 键盘的 5 芯 DIN 接口有故障

处理: 中断后再进入并插好键盘接口

10. 模拟显示不正确

原因: 汉字打印字库不全

处理: 拷入完整的打印字库

11. 打印机故障

(1) 打印机电源开关打开时, 指示灯不亮

原因: 打印机电源线未连接好

解决办法: 检查电源线

(2) 能自检但不能联机打印

原因: 通常为电缆线断开

解决办法: 重插电缆线或用万用表检测电缆线

(3) 能打印英文但不能打印中文

原因: 打印机驱动程序损伤

· 打印机驱动程序感染病毒

· 打印机驱动程序未装入内存

· 打印机驱动程序选错

解决办法: 运行正确的打印驱动程序

(4) 打印有控制符的中文时出现空走格

原因: 字库不全

软件的防拷贝措施与C程序介绍

华中理工大学机械学院 (武汉 430074) 何岭松 吴雅

在软件的开发工作中,为了保护设计者的合法权益,应当对软件进行防拷贝处理,阻止非法用户的随意复制。下面就介绍一点笔者在从事这方面工作中的经验,并给出三个用C语言编制的程序例子供参考。

对软盘的防拷贝,比较简单有效的方法是采用磁道接缝技术,它通过修改软盘基数表和自行设计的磁道格式化、写、读程序,可以对软盘进行特殊操作,并在其上制做出难以复制的指纹。下面是一个笔者按该方法设计的5英寸1.2M高密盘指纹制做程序,该程序在79道格式化时将磁道上最后一个扇区ID标志中的扇区长度N由常规的2(512字节)改为3(1024字节),从而在该道上制做出磁道接缝,另外程序还在正常的0—79道之外格式化出第80额外磁道,用于保存79道磁道接缝的密文数据。

```
#include <dos.h>
#include <stdlib.h>
main()
{int i,d;union REGS r;struct SREGS s;
unsigned char far *p,pl[12],b[1500];
d=getdisk();setdisk(0)
/* 设定A驱 */
r.h.ah=0x00;r.h.dl=0x00;int86(0x13,&r,&r);
/* 磁盘初始化 */
r.h.ah=0x17;r.h.al=0x03;int86(0x13,&r,&r);
/* 设定1.2M格式 */
r.h.ax=0x351e;intdos(&r,&r);segread(&s);
/* 取磁盘基数表地址 */
p=MK_FP(s.es,r.x.bx);
for(i=0;i<=10;i++) pl[i]=p[i]
/* 保存磁盘基数表 */
p[0]=0xdf;p[1]=0x02;p[2]=0x25;p[3]=0x02;
/* 修改磁盘基数表 */
p[4]=0x0f;p[5]=0x2a;p[6]=0xff;p[7]=0x54;
p[8]=0xf6;p[9]=0x0f;p[10]=0x08;
for(i=0;i<15;i++)
/* 79道格式化参数 */
{b[i*4]=79;b[i*4+1]=0;
b[i*4+2]=i+1;b[i*4+3]=2;
}
b[14*4+3]=3;
/* 设定磁道接缝 */
biosdisk(5,0,0,79,1,15,b);
```

```
/* 79道格式化 */
biosdisk(5,0,1,80,1,15,b);
for(i=0;i<15;i++)
/* 80道格式化参数 */
{b[i*4]=80;b[i*4+1]=0;
b[i*4+2]=i+1;b[i*4+3]=2;
}
biosdisk(5,0,0,80,1,15,b);
/* 80道格式化 */
biosdisk(5,0,1,80,1,15,b);
p[3]=0x03;biosdisk(2,0,0,79,15,1,b);
/* 读79道磁道接缝 */
for(i=0;i<512;i++) b[i]=b[i+512];
/* 接缝数据反转加密 */
p[3]=0x02;biosdisk(3,0,0,80,1,1,b);
/* 在80道生成密文 */
for(i=0;i<=10;i++) p[i]=pl[i];
/* 恢复磁盘基数表 */
setdisk(b);
/* 返回原驱动器 */
}
```

将一张格式化好的5英寸高密盘插入A驱中,然后运行一次该程序就可以在盘上生成软指纹。这时该盘就可以做为加密防拷贝盘使用,在所设计的被加密程序中只要挂接一个校验软盘79道接缝数据与80道中保存的密文数据是否一致的函数就可以探知该盘是原盘还是非法拷贝盘,若是原盘程序正常运行,若是拷贝盘则程序异常终止,这样就可以起到防止非法拷贝的目的。

软件运行时每次都要校验软盘的合法性是不方便的,也容易造成原盘损坏。为此笔者设计了一个安装程序,允许用户将软件拷贝到硬盘上去使用。系统仅在第一次安装时校验软盘指纹,通过后则提取硬盘的磁盘I/O参数表和当前安装子目录的目录项信息,然后反转加密后生成硬盘指纹密文文件。由于这些硬盘参数具有较好的唯一性,利用它们可以有效地防止非法用户从硬盘向外进行程序拷贝。下面是一段笔者设计的硬盘指纹制做程序,它先校验安装盘指纹的正确性,若通过则提取硬盘特征参数并反转加密后生成硬盘指纹密文文件供其它工作程序使用。

收稿日期:1994年5月23日

· 汉字驱动程序不正确

解决办法:

· 用RESTORE将汉字字库拷全

· 运行正确的汉字驱动程序

(5)打印不清晰

解决办法:换色带

(6)信号线断掉后能自检,信号线插上后不能自检

解决办法:检查或更换信号线与打印机适配器

(7)打印汉字时漏笔画,即漏针现象

原因:(1)打印针受阻

(2)打印机头驱动线圈损坏

(3)打印针损坏

处理:(1)用酒精清洗打印针

(2)用万用表测量各线圈的引线与地线间的电阻,若各线圈的电阻值相近,则线圈良好,若少数线圈电阻值相差较大,则为线圈故障。

处理:换线圈

(3)换打印针头或连动片

参考文献:

1 吴万锋等编著.计算机病毒的分析诊断与防治,海洋出版社,1993年2月。

采用上述方式只要在硬盘上开辟个子目录,并将所有文件拷入其中,然后运行上面的安装程序生成硬盘指纹。以后其它软件工作时只要校验当前盘的特征参数与密文文件的内容是否一致就可以知道软件的合法性。这样既起到了软件防拷贝的目的,又避免了对软盘指纹的反复校验,从而极大地方便了用户,同时也提高了加密系统工作的可靠性。下面是一段被加密的工作程序的例子,其中挂接了一个硬盘指纹校验函数,若通过则程序进行正常工作,否则立即终止。

```
#include <stdio.h>
#include <dos.h>
main()
{unsigned char gap[64];
long int t1,t2;union REGS r;
t1=biostime(0,0)
get_hard(gap);
t2=biostime(0,0)
if ((t2-t1)>50) int86(0x19,&r,&r);
/* 抗动态跟踪 */
/* 软盘合法性校验 */
soft_pass();
save_hard(gap);
/* 生成硬盘指纹 */
}
soft_pass();
{int i,j,k,d; union REGS r;struct SREGS s;
unsigned char far *p,b[1500],bl[512];
d=getdisk(); setdisk(0);
r.h.ah=0x00;r.h.dl=0x00;int86(0x13,&r,&r);
r.h.ah=0x17;r.h.al=0x03;int86(0x13,&r,&r);
r.h.ax=0x35le;intdos(&r,&r);segread(&s);
p=MK_FP(s.es,r.x.bx);
p[3]=0x03;biosdisk(2,0,0,79,15,1,b);
/* 读 79 道接缝 */
for(i=0;i<512;i++) b[i]=b[i+512]
p[3]=0x02;biosdisk(2,0,0,80,1,1,bl);
/* 读 80 道密文 */
for(i=0;i<512;i++)
if(b[i]!=bl[i])
/* 指纹校验 */
{printf("Soft disk mark error!");exit(0);}
setdisk(d);
}
get_hard(unsigned char g[64])
{int i,j,k,l,d,nf,sf;
unsigned char a[40],f[16][32]
getcurdir(0,a); l=strlen(a);
/* 取当前子目录 */
d=getdisk();absread(d,1,0,f);
/* 读当前盘 BOOT 区 */
for(i=0;i<32;i++) g[i]=f[0][i]
/* 保存磁盘 I/O 表 */
nf=(g[17]+(g[18]<<8))/16;
sf=g[14]+(g[15]<<8)+(g[22]+(g[23]<<8))*
g[16];
for(i=0;i<nf;i++)
{absread(d,1,sf,f)
/* 读目录区 */
for(j=0;j<16;j++)
if(strncmp(a,f[j],1)==0)
```

```
{for(k=0;K<32;k++)g[32+k]=f[j][k];
/* 保存当前目录项参数 */
return;
}
}
save_hard(unsigned char g[64])
{int i;FILE *fp;
for(i=0;i<64;i++) g[i]=g[i];
/* 硬盘指纹反转加密 */
fp=fopen("HARD. DAT","wb");
/* 生成密文文件 */
fwrite(g,sizeof(char),64,fp);
fclose(fp);
}
#include <stdio.h>
#include <dos.h>
main()
{hard_pass();
/* 硬盘指纹校验 */
/* work part */
printf("OK,SUCCESS!\n");
}
hard_pass()
{unsigned char gl[64],g2[64]; FILE *fp;
long int t1,t2;union REGS r;int i
t1=biostime(0,0) get_hard(gl);
/* 读硬盘参数 */
t2=biostime(0,0)
if ((t2-t1)>50) int86(0x19,&r,&r);
/* 抗动态跟踪 */
fp=fopen("HARD. DAT","rb");
fread(g2,sizeof(char),64,fp);/* 读密文文件 */
fclose(fp);
for(i=0;i<64;i++)
if(gl[i]!=g2[i]指纹校验
{printf("Hard disk mark error!");exit(0);}
}
get_hard(unsigned char g[64])
{int i,j,k,l,d,nf,sf;
unsigned char a[40],f[16][32];
getcurdir(0,a);l=strlen(a);
d=getdisk(); absread(d,1,0,f);
for(i=0;i<32;i++)g[i]=f[0][i];
nf=(g[17]+(g[18]<<8))/16;
sf=g[14]+(g[15]<<8)+g[22]+(g[23]<<8))*g
[16] for(i=0;i<nf;i++)
{absread(d,1,sf,f);
for(j=0;j<16;j++)
if(strncmp(a,f[j],1)==0)
for(k=0;k<32;k++) g[32+k]=f[j][k]
return;
}
}
```

为了提高整个加密系统的可靠性,在上述程序中还可以加入一些抗动态跟踪和静态分析的手段,防止非法用户的解密,有关这方面内容可参阅参考文献。

参考文献

- 1 林宜雄主编. 磁盘加密解密实用技术. 西安交通大学出版社, 1992. 6

生成真实感图形的四大技术

西安交通大学计算机系(西安 710049) 亢祝真 陆丽娜

摘要 本文系统地分析了生成计算机真实感图形的四大技术:浓淡处理,光线跟踪,纹理效应和阴影处理,归纳总结了它们的主要算法思想,研究起源和发展动态。

关键词 浓淡处理 光线跟踪 纹理效应 阴影处理 真实感图形

1 引言

在计算机生成图像的研究领域,生成高度真实感图形是一项极富挑战性的工作。

所谓真实感图形,主要指图像的效果在屏幕上显示时能迷惑观察者,使他认为这是极其逼真的真实图景。真实感显示,又有两种含义:

(1)广义的:指显示或给出的图形具有某种真实感。实行的方法有:对物体进行轴测投影,透视投影,对物体消除隐藏线和隐藏面,等等。

(2)狭义的:使计算机能产生像照片那样的黑白图像和彩色图像,甚至产生真正三维的立体图像。这种显示又称高度真实感显示,产生的是仿真效果。它的物质基础在于高度发展的光栅显示技术。

在七十年代早期,随着存储器价格的下降,光栅技术有了新的发展。在当时流行的是笔划式显示器,它只能画出一定数量的直线,颜色也十分有限。早期的真实感图形研究集中在随机扫描显示器上画出的对象物体上消除隐藏线,显示出来的物体尽管并不具备真实感,但包含了许多特定的信息,以为计算机辅助设计等工作服务。八十年代以来,由于大规模集成电路允许整个被填充的屏幕区域的颜色具有相当广泛的不同的亮度和颜色的浓淡,这使得真实感图形的研究更加得心应手。

实际的需要使得真实感图形的显示在应用中具有重要的意义,仿真显示在作战模拟、仿真训练方面起到不容忽视的作用;真实感动画的制作可以生成灵活生动的各种故事和场景;美观的产品造型、包装设计图案、艺术广告、装饰画等等;在计算机辅助设计和制造中,对图形的高度真实感的要求也越来越多。

生成高度真实感图形通常有以下几个步骤:消除隐藏面,光线跟踪,浓淡处理,造阴影,生成纹理效应,等等。由于消隐算法是最常用的且已相当成熟,下面介绍浓淡处理、光线跟踪、纹理效应和阴影处理四种技术。

2 浓淡模型

三维景物图形产生的逼真性很大程度取决于能否成功地模拟浓淡或明暗效应。我们对物体的处理从观

察变换到进行消除隐藏线和隐藏面后,就要用光照模型来计算用以显示可见面的亮度和颜色。

浓淡处理有两种经典的算法,分别是 Gouraud 方法和 Phong 方法。

(1). Gouraud 方法: H. Gouraud 对在 1971 年提出曲面的连续性生成浓淡的方法。将曲面用多个多边形来逼近,并保存多边形的各顶点处的真正表面的法向量。当多边形被渐次地转换为像素时,在每个顶点处取的正确颜色被计算,这些值沿多边形进行线性的插值。Gouraud 法有较好的浓淡效果,但是对多边形的逼近情况会强烈地影响由镜面反射而形成的高光反射形状。

(2). Phong 方法, Bui T. Phong 和 F. C. Crow 在 1975 年提出了曲面多边形的改进绘制算法。将沿扫描线光强作线性插值的 Gouraud 方法改进为:沿扫描线对法向量作线性插值,然后按光照模型计算各点的光强。Phong 的浓淡处理方法计算量大,但可解决 Gouraud 法所遇到的许多问题。

Gouraud 法和 Phong 法也通过由多面体逼近曲面物体的思想求得了曲面的仿真显示。

对透明体的浓淡处理还需采用专门的方法,因为经典的光照模型和浓淡处理方法都假定到考虑的物体表面是不透明的,入射光线在透明体上发生反射和透射,透射也分为规则透射和漫透射。由 Newell 和 Sancha 提出的透明算法不考虑折射的影响,是一个最简单的透明算法:对透明的多边形或表面注以标记,若可见面是透明的,则取它与同它相距最近的另一表面光强的线性组合,并将所得光强值写入帧缓冲器,并递归地进行此算法直至遇着一个不透明的面或背景。

3 光线跟踪

许多研究集中于光学效应和光线跟踪上,研究方向主要包括透明性造型,折射造型,反射造型,及使用透镜的照像机模型。

光线跟踪是由 Whitted 在 1980 年为浓淡显示提出的改进照明模型中引进的,是对 Appel 和 Goldstein, Nagel 在三维可视仿真问题中使用的光线投射的改进。在光线投射中,光线从眼中出发,经过像

素进入世界,光线和世界中的物体最近的交点决定了可见的面,而 Whitted 在交点处又加了两条光线路径,分别沿反射方向和折射方向,这样可以精确地为反射和折射面的变形进行造型。

光线跟踪的算法基本思想是:观察者能够看见景物是由于光源发出的光照射到物体上的结果,其中一部分光到达人的眼睛引起视觉。到达观察者眼中的光可由物体表面反射而来,也可通过表面折射或透射而来,若从光源出发跟踪光线,因只有少量的光能到达观察者的眼睛,处理的效率太低,因此按反途径进行光线跟踪,即按从观察者到景物的方向。在简单的光线情形下,假定画面中的景物均已变换到图像空间,光线从观察者出发,通过光栅中像素的中心进入画面,然后沿光线路径进行跟踪以决定它与画面中的哪一物体相交。每一光线均需与画面中的每一物体进行比较,如果相交,则需求出该光线与物体的所有可能的交点。具有离观察点最近的交叉对应的面为此像素的可见面,再由光照模型可确定该像素处的显示值。完整的光线跟踪算法中用到的光照模型,除了要考虑直接来自光源的光线外,还要考虑由于其它物体的反射或折射传来的光线。

光线跟踪的两大缺点在于费用昂贵和有走样现象,对光线跟踪的研究主要有四个方向。(1)对不同的物体求取交点的算法。(2)扩展光学效应的范围,用光线跟踪捕获更多的光学效果。(3)反走样技术。(4)降低测试交点的次数。早期研究都集中于迅速在光线和复杂物体之间求取交点。

4 纹理效应

纹理映射是由 Catmull 在 1974 年对计算机显示曲面的细分算法中提出的,来显示复杂物体的真实性。事实上,这是一种对存在的多边形“造墙纸”的方法。一个被进行纹理处理的多边形的每一个顶点都包含了在二维空间的坐标。当一个像素被进行明暗处理时,进行纹理坐标的插值并执行对包含了纹理的颜色的二维数组的查找。这个二维数组的值被用作多边形在特定像素的颜色,从而产生了“墙纸”。这个过程其实就是在纹理多边形上的像素被转换到纹理空间,在被转换的像素边界进行的纹理采样生成了屏幕上的像素的颜色。

由于纹理效应包含的频率、高,易于产生反走样。而且,使用纹理效果多边形可能只占屏幕的很小一部分,使得同一个像素上有许多纹理点。

Blinn 和 Newell 通过使用一个标志滤波器改进了上述技术。Feibush, Levoy 和 Cook 则推出了 Gaussian 滤波器,但是计算很昂贵,Wiuiams 提出的滤波机制,虽然并不如其他方法精确,但计算纹理是最快的了。他在不同的分辨率下保存纹理的备份并采用最合适的一个。Crow 使用一张预计算表来存整个纹理映射,使得

滤波能在常效时间内完成,Norton, Rockwood 和 Skolmoski 提出把纹理映射作为一个解析函数,预先计算光谱频率并无需改变任何高频地去重构此解析函数,从而快速计算纹理,这很适于模糊物体如云、波浪、原野等。纹理在这些物体上并不需要包括许多频率因素。

沿着一个复杂的三维物体,很难在其上包裹一个二维的纹理,这需要定义一个函数,把这个物体的空间坐标映射到三维纹理空间并使用三维空间,不管物体的形状如何,在它表面上的纹理是一致的,当纹理模型与制作物体的材料是一致时,特别有用,因为需要额外的大量的空间来存一个像素的三维数组,Peachey 和 Perlin 提出使用“过程纹理”,但是过程纹理很难进行反走样。

纹理无需包括一个表面上的颜色。存储其他表面特性也可帮助纹理映射增强性能,Blinn 提出应把表面法矢扰动存储于纹理映射中,又称颠簸映射。这可以为有折皱的表面产生逼真的模拟(通过高光正确地与表面的颤动匹配的方法)。Gardner 使用解析纹理来修整椭球体的表面边界,以模拟树和云的形状。纹理技术在环境映射中使用也较多,如果物体是闪闪发亮的,那它的表面可以反射其它同在环境中的物体,计算这些反射是昂贵的,但纹理技术就可以把从一个视点看去的环境的近似图景投射到球体、圆柱体或长方体上,并把这投影存在一个纹理映射中。然后,当我们为一个点造浓淡时,在反射的方向进行纹理采样即可。

5 阴影处理

阴影显示在计算机合成图像上表示真实性的过程中起到了重要的作用。当观察方向与光源方向不重合时,就会出现阴影,它使人感到画面上景物的远近深浅,从而极大地增强了画面的真实性。

阴影算法的复杂程度与光源的模型有关。这包括光源的类型和位置的情况。阴影处理的算法很容易和光线跟踪模型结合为一体,假定光源(或多个光源)向量 L 在其路径上不被任何物体阻挡,如果 L 被阻挡,则当前表面点处于阴影中要确定其他物体是否处于 L 的路径上,方法是建立一条从当前点到光源的附加光线,这条光线称为阴影测试线。如果一个完全不透明的物体是处在阴影路径上,则把光线交点的亮度值减少至环境光照值,如果相交物体是半透明的,则计算出光线交点的亮度值,这时仍把 L 当作一个常向量(光源在无限远处)已不合适。所谓阴影测试线只是其方向就是 L 方向的那些光线,如果阴影测试线与半透明物体相交,则它应被折射。但是,计算方法比较复杂,阴影测试线在初始阶段是作为表面交点与光源之间的直线计算的,这项计算较简单,但要从这一点到光源的光线进行跟踪并考虑折射效果,就相当困难了。

随着光源数量的增加,用于阴影测试计算开销会很快占主要地位。因为,如果有几个光源,则每个光线

与表面的交点会产生几条阴影测试线,对一条阴影测试线的物体交点的开销点对一条主光线的开销是完全相同的,为了减少阴影测试的计算量 Haines 和 Greiburg 在 1986 年提出一种使用光线缓存区作为阴影测试加速器的新方法,用这个方法,可使用阴影测试次数减少至 1/4 至 1/30。但它只能处理点光源,且需要很大存储量。

参考文献

1 陈振初、蔡宣平. 计算机图形显示原理(软件). 国防科技大学出版社,1991.6月版

2 Realism in Computer Graphics: A Survey, J. Amannickes. IEEE CG&A, January 1987

3 H. Gouraud. "Continuous Shading of Curved Surfaces", IEEE Trans. Computers, June 1971

4 T. Whitted, "An Improved Illumination Model for Shaded Display"

5 F. C. Crow, "Shadow Algorithms for Computer Graphics", Computer Graphics (Proc. SIGGRAPH77), July 1977

CAD/Graphics'95

第四届国际计算机辅助设计及图形学学术会议征文通知

1995 年 10 月 22—25 日,华中理工大学,武汉,湖北,中国

主办单位:中国计算机学会,华中理工大学

协办单位:美国计算机学会(ACM),日本信息学会(IPSJ),中国自动化学会,中国工程图学学会,IEEE 北京分会,湖北计算机学会

征文范围:

* 计算机图形学	* 工程数据库/知识库
* 计算机辅助几何设计	* 智能 CAD
* 计算几何	* 计算机图形硬件及工程工作站
* CAD/CAM 技术及应用	* 科学计算可视化
* CAD/CAPP/CAM 集成	* 仿真技术
* IC 计算机辅助设计	* 人机交互技术
* 测试、诊断、容错技术	* 多媒体技术

重要日期:1994 年 12 月 1 日截稿,1995 年 2 月 1 日录用通知,1995 年 4 月 1 日终稿交递。

论文要求:所投论文应符合征文范围且从未发表过。论文须用英文书写,包括插图在内不得超过 5000 字。论文须用 A4 纸隔行和单面打印。除论文正文外,所投论文还必须包括不超过 200 字的论文提要、8 个以内的关键字、作者姓名、职务和技术职称、所在单位、通讯地址、邮编及电话、传真号码。

论文请寄:武汉华中理工大学 CAD 中心 高峰

(邮编:430074,电话:027 7861881,传真 027 7866053)

APPLICATION RESEARCH OF COMPUTERS

Vol. 11. No. 4(Total 60)

CONTENTS

SURVEY

The Development of Radio Net and Its Interrelated Technology	Zhang Zhen(1)
In Trodution to Information High—speed Way of America (NII of America)	Deng Rui xiang(2)
Four Technologies of Generating Sense of Reality Graphic	Kang Zhu zhen et al. (57)

RESEARCH

Structuring and Model on Level Classification and Realizing	Wang Yi zhang(3)
The Analyzing System Pro/FA of Computer—aided Production Flow	Zhu Guo wang et al. (5)
Designing Software of Cross Forward Thought and Reverse Thought	Liang Zong qiang(7)
The Study of Expert System of Failure Analysis for Automobile Spares	Bi Chang qing et al. (9)
Handling Error Debugging by Parallel and Distribution	Yuan Dao hua et al. (11)
ORACLE Parallel Servers Technique on Cluster UNIX Environment	Li Qiang et al. (15)
Optimizing the ORACLE SQL Query Statement	Li Qiang et al. (17)

APPLICATION

A Technic in Designing Microcontroller	Zhang Kun et al. (21)
A Method of Displaying Memory State	Liang Chun cheng et al. (24)
Responding to Serial Communication by Software	Xu Qing lin(28)
Controlling System of Real Time Multitasking by Microcomputer	Jin Biao(30)
Development of the Quantitative Analysis System for Biological Histology	Gao Yun qin(32)
A Universal Generator of Decision Support System	Sun Yong sheng et al. (34)
Analyzing HD—COPY	Xu Ling yi et al. (36)
An Intelligence Monitoring System on Roller Electrical Drive	Peng Yi qiang et al. (36)
A Practical Update Routine of Logging Data on Full Screen	Wang Xiong ji(39)
Making Front Cover of Management Software Having Trade Mark	Wu Suo ping et al. (42)
Foxpro Record Group Reference Technology of Rushmore	Weng Zheng ke(44)
Method of Redefining the Key of Foxpro and Technic	Shi Yong xin(46)
Talking about Clear Screen as Well	Le Qi bin(47)
Data Transmission Between Micro—PROLOG and DBASE	Li Jing gui(49)
Measure of Preventing Software Copy and Introducing C Program	He Ling song et al. (55)

MAINTENANCE

Management Maintenance of Hard Disk and Handling Common Fault	Guo Ying ge(50)
An Example about Diagnosing Faults on Printer of M2724	Chen De ming(53)
Common Problems of MPS System and Its Correcting	Zhou Hong Fu et al. (53)

INFORMATION

The Solicit Articles Circular of International Academic Meeting	(59)
News in Breif & Advertising	(Cover1,Cover2,Cover3,Cover4,14, 33)

新技术、新产品荟萃

1. 博大(BD)系列电脑电视字幕机——用于电视录相及广告制作

BD 系列电脑电视字幕机由微计算机、标准 PAL 内外同步器、彩底发生器、PAL 彩色合成器、视频合成器、高精密中西文字库及字幕、图形编辑、创作软件组成。主要功能有：文字字幕编辑、动画/图形创作、数字图像处理特技、外设备同步、编辑文件存储等功能。主要型号如下：

● BD—1：同屏 16 色 11 矢量字库 286 主机 高分彩显

● BD—2：同屏 256 色 11 矢量字库 三维动画 386 主机(带协处理器) 高分彩显

● BD—3：同屏 32768 色 11 种矢量字库 386 主机 高分彩显 视频质量达广播级标准

● BD—4：同屏 32768 色 11 种矢量字库 数字特技 三维动画有渐变色 486 主机 高分彩显 视频质量达广播级标准

本系列产品荣获 1988 年度四川省科技进步二等奖。产品遍布全国，已推广千余台。

2. SMD—6420 三维制鞋辅助测量辅助设计系统——用于各种鞋类设计、样片制造及生产管理

本系统是制鞋业计算机辅助制鞋测量和辅助制鞋设计而研制开发的系统。系统提供的辅助测量及辅助设计功能覆盖了制鞋设计全过程各个阶段，能满足皮鞋、布鞋、胶鞋、旅游鞋等各类男、女、童鞋设计需要，使用该系统设计准确度高、快捷高效、简便易学，是目前国内功能最完整、最实用、性能价格比最优的一套系统。该系统已在数拾家制鞋企业中推广应用。

硬件：主机：名牌 486 微型计算机 外设：三维制鞋面坐标测量仪、1280×1024 高分辨率图形显示器、TMS34X 系列图形加速卡 A3 幅面平面数字化仪、A3 或更大幅面 XY 绘图仪。

软件：全套软件包括：辅助制鞋测量，全制鞋面展平，平面、立体帮、底样设计及帮、底样片扩缩，单位优化排料，面积测量计算，样片切割，制鞋档案资料管理和设计数据管理等功能。

该项目已获 90 年度四川省科技进步二等奖，92 年起被列为国家重点推广项目。

3. MD 系列机器设备状态监测与故障诊断系统

MD 系列状态监测与故障诊断系统以国内外这一领域最新理论、技术及方法为基础，应用当代先进的 ASIC 技术、网络技术的优势，率先在全国推出最高水平、最优性价比的 MD 系列产品。该系列产品其性价比优于进口产品，系替代进口的最佳选择。主要用于工业生产工艺流程中的各关键设备(机组)的运行监测、故障预报、故障分析诊断和指导维修，进行单机及多机组同时在线监测。该系统是保证生产安全运行，减少维修费用，提高经济效益的不可或缺的手段。本系统已在航空燃气轮机研究中心及四川化工厂等重要现场中运行。

92 年 12 月，本系统获得四川省重大科技成果三等奖。93 年列为四川省重点科技推广项目。

● MD³⁹⁰⁵₃₉₀₆ 系列机器设备状态监测与故障诊断网络系统——用于工业生产工艺流程中各关键设备(机组)的运行监测、故障预报、分析诊断和指导维修，实行多机组同时在线监测。

● MD3902 系列机器设备状态监测与故障诊断系统——用途同于 MD3905、3906 系统，区别在于适用于单个机组的在线监测，如企业自备电厂汽轮机系统。

● MD8500 系列便携式数据采集/分析仪——现场数据采集及预处理。适用于设备的定期巡检、重点监测及现场事故处理使用。

● MD9500 系列便携式旋转机械故障诊断仪——用于化工、机械、冶金、电力、轻工等工业现场对旋转机械进行现场监测和现场分析。

4. BMP—C 系列彩色脑地形图处理系统——用于诊断脑部疾病

BMP—C 型彩色脑地形图处理系统是诊断脑部疾病和研究大脑神经活动的重要仪器。它利用人体自发产生的脑电图，经频谱分析处理为直观的地形图，反映出病灶在脑部的位臵和范围。我们在总结分析几十万临床病例的基础上，率先在国内开发出 BMP—C 型脑地形图处理系统和性能优异的 STAR9316 16 道脑电信号采集器，并可配置成为一套完整的一体化系统。BMP—C 型对于脑肿瘤、癫痫、脑炎、脑外伤、脑血管、脑萎缩等颅脑疾病的准确率达 70% 至 92%。尤其是独特的癫痫分析模式的研制成功，标志着 BEAM 技术进入新的里程。同时 BMP—C 型带有性能优良的十六道脑电信号放大器和大容量记录仪，独家推出脑电全息处理模式，成为当今最为理想的 EEG/BEAM 系统。本系统现已在全国推广近 500 套。

5. CBM 系列心电图体表电位标测系统——无创性心电图检查，早期诊断心脏疾病。

体表电位标测是一种新型的无创性心电图检查方法，它通过安放在人体躯干上的 64 个电极获得心动周期中各瞬间身体表面心电图的空间分布，经计算机处理，以体表等电位图等多种形式表达心脏各部分兴奋与恢复的过程，它反映的是心脏电生理活动的空间变化规律，提供的信息比传统的心电图更多、更全面，因而更能客观准确地反映心脏的电生理活动。

该系统图象清晰、失真小、重复性好、操作简便，在诊断心肌梗塞、预激综合症、束支阻滞、心室肥大等病方面比常规心电图灵敏、准确性高，为心脏疾病的早期发现、定位诊断提供了先进设备。本产品已于 92 年通过省级技术鉴定，并以最优的性能价格比(价格仅为国外同类产品的二十分之一)，在国内数家著名大医院获得推广应用。

6. KRL 矿热炉微机控制系统——用于黄磷、铁合金、电石行业的矿热炉控制

矿热炉是广泛应用于化工、冶金、有色金属行业的重要冶炼设备，也是能耗大户。KRL—Ⅰ 矿热炉微机控制系统就是为降低设备的能耗，提高经济效益而研制的高科技节能产品，该系统用于黄磷、铁合金、电石冶炼和工业硅过程控制，投资少，见效快，其可靠性大大超过同类产品。绵竹黄磷厂统计，使用该系后可增产黄磷 6.3%，节电 7.86%，电极消耗下降 6.1%，焦炭消耗下降 9.98%，年直接经济效益在 25 万元以上，使用后三个月至半年即可收回投资。现在该系统已经在数拾台矿热炉上成功地运行。92 年已被列入四川省重点推广项目。

7. WCY 型电网调度微机远动系统——用于地方电网的监测、调度指挥

WCY 微机远动系统由调度机与厂站端机构成。以 1:N 的循环调度模式,实现对各分散厂站端的遥测、通信、遥控三遥功能。整个系统具有统一的时钟,组合数台电视形成大屏幕代替老式模拟屏,用高精度电压、电流可变换器取代传统的变送器,系统可靠性高、运营简便,保证地方电网有效合理地分配利用能源。本产品适用于地县级电网调度自动化及大厂矿电力管理。

8. DZJ-5G 电能综合计量仪——电压的监测、电能分时计量

专门用于电能的分时计量、电压缺相检测、电压合格率统计计算。它是电能分时段计量、监测电能质量的专用智能仪表。经省级技术鉴定和中国测试技术研究院测试,各项指标均属国内先进水平。可广泛应用于发电厂、供电局变电站、电力用户、小水(火)电、电力公司,是电力现代化管理最理想的仪器,已在全国推广 300 余台。

9. 电子防盗、电视监控报警保安系统

本系统由闭路电视监视、电子防盗报警、内部有线对讲、进出门管理四个子系统组成。该综合系统将报警、监视、指挥、录像功能融为一体,根据各种对象,不同规模和独特的系统要求。可以选择单个子系统独立使用,或者组成综合系统。使用

可靠,功能丰富,性能价格比优。本系统广泛应用于银行、宾馆、车站、机场、大型综合商厦、交通枢纽、博物馆、展览馆、停车场、重要库房、军事设施,各企事业单位财务室、档案室以及别墅、普通家庭等。

10. DNX 型电脑成像机——用于旅游点电脑画像

DNX 电脑成像机通过了四川省省级鉴定,获四川省科技进步三等奖。此机广泛适用于旅游点、宾馆、游乐场、科技馆、照像馆等为人画像。

该机由电脑主机、摄像头、快速高密度打印机、监视器组成,经电脑处理,可在 35 秒左右打印出像片。这种像片既有一般照片的逼真性,又有素描画像的艺术性,别具风格,可永久保存。若备专用色带和调温熨斗,还可将像片转印到混纺织物旅游纪念品上,深受旅游部门和群众的欢迎。

电脑成像机是在一般白纸上印象,取材就近便利,且结构简单,操作方便直观,一般人经过半小时学习可熟练掌握。

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

《计算机应用研究》杂志办刊单位

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

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

山东省计算中心
河南省计算中心
青海省测试计算中心
武汉工学院

《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 闫长荣 王升亮
李天健 郑国基 叶大卫 乔中南
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 曾光初 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
刘启茂

责任编辑:邓像文

1994 年第 4 期(总第 60 期)

出版日期:1994 年 7 月

计算机应用研究(双月刊)

JI SUAN JI YING YONG YAN JIU

(公开发行)

刊 号: CN51-1196/TP(国内)
ISSN 1001-3695(国际)

邮发代号: 62-68(国内)
BM4408(国外)

主 编:张执谦

副 主 编:李泽民

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

电 话:(028)5582666—2055

印 刷:成都市新都华兴印务有限公司

订 阅 处:全国各地邮政局

国内总发行:成都市邮政局

国外总发行:中国国际图书贸易总公司

广告经营许可证:川蓉工商广字 050 号

每册定价:2.00 元