

现代计算机

'95.10

总第 45 期

原名 《电脑与微电子技术》

现代计算机杂志社出版

MODERN COMPUTER



40M²LED 全彩色电视广告屏 矗立在中国出口商品交易会门前

广州中鸣公司隆重推出

公司地址：广州中山大学东北区 368 号
董 事 长：张苑岳 传真：4185464

电话：4184847
邮编：510275



广州市科教电脑设备有限公司

Guangzhou Science & Education Computer Equipment Co. Ltd.

COMPAQ 高档笔记本专卖 美国康柏电脑

不
求
价
格
第
一

- * COMPAQ LTE Elite 4/75(486DX4 - 75)
8M, 510M, 1.44M TFT
- * COMPAQ LTE Elite 4/50(486DX2 - 50)
8M, 510M, 1.44M TFT
- * 8M RAM For COMPAQ Elite
- * COMPAQ Smart Station Base For LTE Elite 4/40/50/75cx
- * COMPAQ 15" SVGA 151FS
- * COMPAQ Keyboard
- * COMPAQ Mouse
- * Kingston PCMCIA Fax/Modem 14.4K B/S
- * Xircom PCMCIA Ethernet II PS TP

但
求
服
务
最
好

地址:五山路华师科技大楼 157—159 号, (邮政信箱 1233 号, 邮编:510630)

电话:7549981、7549982、7549983、7549984、7549985、7549986、7549987、7549988 Fax:7549989

展销部:广州天河体育东路 39 号天宝大厦二楼新一代电脑城 A200 室 Tel:7548818

科教电脑多媒体中心

集 各种声霸卡、CD—ROM、音箱……

供 各种解压卡、视霸卡、TV 卡……

售 14" TOPCON 彩显、LQ—1600K、LQ300K 打印机……

欢迎批发、联销、合作!

地址:广州五山路科技街二栋二楼 222 号 Tel:7548485、5510446(Fax)

中山大学 电脑照排科技中心

博采众长 自成一格
技术先进 服务周到

经营： 高档精密照排系统
普及型轻印刷排版系统
电脑多媒体网络系统
书报刊排版印刷

承接： 印刷厂技术更新改造
新老排版系统换代升级
信息管理系统软件开发
电脑排版技术培训

我们的客户遍布茂名、珠海、湛江、深圳、阳江、广州
和省内外，欢迎垂询，欢迎更多的朋友加入我们的行列

诚聘： 电脑技术人员 2 名。条件：男性，大专文化以上，
年龄 30 岁以下。有意者请寄来自述简历及半身脱
帽相片一张，听候约见。一经录用，待遇从优。

地址：广州中山大学东北区 368 号

邮编：510275

经理：梁 勇

电话：4186300—1999

现代计算机

'95.10
总第 45 期

主 办 中山大学
编辑出版 现代计算机杂志社
主 编 张纬铮
联系地址 广州新港西路 135 号中山大学
邮政编码 510275
电 话 4186300—6540
刊 号 ISSN 1007-1423
CN 44-1415/TP
广告许可 粤 010329 号
广告总代理 广东省广告公司
直通电话:7752254 传真:7778225
印 刷 中山大学印刷厂
'95 发行 广东省报刊发行局
'96 发行 现代计算机杂志社
每期订价 2.00 元 零售价 3.00 元

公开发行 1995 年 10 月 20 日出版

本刊图文版权所有
未经允许不得转载

注重实用性、知识性、知识性
面向科研、面向生产、面向管理、面向用户、面向教学

明年本刊自办发行

- 不用到邮局去查找邮发代号
- 欢迎新订户来函索取明年的订单
- 老订户的订单上期已随刊奉送
- 欢迎投稿
- 欢迎刊登广告

目 录

☆ 研制·开发·应用 ☆

- 户外 LED 全色大屏幕视频显示系统 张苑岳②
- LED 显示屏多媒体编播平台 METS 的设计与实现 余 阳⑤
- LED 群显系统 左明 李军 张苑岳⑧
- 图象形态法在棉纱线黑板条干图像处理中的应用 蒋忠仁⑭
- SLPC 可编程调节器在锅炉自动控制中的应用 隋景明⑮

☆ 软件纵横 ☆

- FOXPRO2.5 源程序精华点滴 顾建飞⑮
- C 语言汉字平滑移动的设计示例 黄焕如 王 玲⑲

☆ 网络与通信 ☆

- 如何提高用户局域网的性能 孙迎春⑳
- SSU-12 数字程控用户交换机计费系统的二次开发 郑翔东㉕
- NOVELL 远程自举工作站上制作系统盘一法 饶拱维㉙

☆ 编程技巧 ☆

- 通用开放式下拉菜单自动生成器 狄振强 孟广盈㉙
- 用 FOXPRO 制作信息活动窗口 刘全胜㉚
- 在 CLIPPER 中的图形界面设计 周昭权㉜

☆ 经验与交流 ☆

- 也谈用五笔字型为 AutoCAD 输入汉字 张会福㉟
- 应用软件中的汉字输入方法 兰文祥㊲
- GW-200 彩显特有的故障及其修复 刘应琪㊴
- FoxPro 中使用数组必须注意的几个问题 车光宏㊶

☆ 动态与信息 ☆

- 欢迎订阅《环境技术》杂志 ㉚
- 欢迎订阅《电子天府》(双月刊) ㉜
- 欢迎订阅《电机电器技术》杂志 ㉞
- 欢迎订阅《电子文摘报》与《家庭电子》月刊 ㉟

户外 LED 全色大屏幕视频显示系统

中山大学电子机械研究中心 张苑岳

摘要:本文简要介绍户外 LED 大屏幕显示系统的工作原理、特点和发展情况,并详尽地阐述了它的设计要点。通过应用实例的引证使读者对该系统的优点有更深刻的了解。

关键词:LED 户外 全色大屏幕 视频显示系统

一、引言

到目前为止,户外全色大屏幕视频显示系统几乎都属于真空泛束管型或平板 CRT 象素管型。它们具有亮度高、彩色好、观看角度宽以及图象质量高等优点。但存在功耗大、工作电压高、造价贵、寿命短以及易破碎等缺点。

八十年代后期,由于半导体发光器件(LED)制作技术的提高,高亮度 LED 的问世和市场价格的大幅度下降,使以固体发光的 LED 器件取代传统的真空型发光器件制作户外大屏幕视频显示系统成为可能。尽管当时按三基色原理,只有红、绿二种基色的 LED,只能制作要求不高的户外多色 LED 大屏幕显示屏。但由于它具有低功耗、低电压、低成本、长寿命、不易破碎等固有优点,还是受到广大用户青睐。为了弥补缺乏蓝色 LED 不能制作全色大屏幕的不足,九十年代初有人利用小型化的蓝色真空低压萤光管(VFD)作补充,制作成 LED 和 VFD 混合型全色视频显示系统。使屏幕的色彩得到了很大的提高。但是,由于 VFD 管的功耗大,系统散热费用高以及寿命短不能与 LED 相匹配等,实际使用上仍存在一定的困难。

直到 1994 年初,InGaN 高亮度蓝色 LED 推出市场后,全固化、全彩色户外大屏幕视频显示系统的梦想才成为现实。中山大学电子机械研究中心和所属广州中鸣显示技术工程有限公司于 1994 年中率先研制成功我国第一块 5 平方米的户外 LED 全色大屏幕视频显示系统。而且,还于 1995 年 4 月上旬,在广州中国出口商品交易会门前竖起我国第一块 40 平方米具有 76,800 只象素的全天候户外 LED 全色大屏幕视频显示系统。该系统从 95 年 4 月 15 日现场直播第七十七届交易会盛况以及此

后全天候地播放新闻、经济讯息和商业广告至今,历时半年,经受了 95 年广州夏季强台风及其带来的狂风暴雨和高温酷暑的考验。证明户外 LED 全色屏在亮度、色彩和图象清晰度等参数与传统的真空管型大屏幕显示系统相当。但在寿命、功耗、运作环境要求、制作成本和维护保养费用等方面的优势,是后者远远无法与之比拟的。充分显示出户外 LED 全色大屏幕视频显示系统的优点和生命力,以及它的应用前景和海内外市场潜力。

诚然,不同的显示器件构成的显示屏幕具有不同的特点,都有自身设计规律。一个好的户外 LED 全色屏的基本要求首先要有高的亮度、好的色彩、高的图象清晰度、大的观看角度和合乎要求的显示功能。但在设计上往往不能同时达到。因此,必须对其性能作全面考虑,在相互制约的因素间进行调整和取舍是在所难免的。

二、户外 LED 全色大屏幕显示系统的工作原理

户外 LED 全色大屏幕视频显示系统(如下简称 LED 显示屏)的工作原理如图 1 所示。图中显示屏由若干块显示模块组成。每一模块则由若干只象素组成。而且每一象素依白场要求,由若干粒红、绿和蓝色 LED 构成。

如图 1 所示,来自电视、录像或储存在硬盘或 CDROM 中的视听信息,经多媒体设备处理后,以数字信号方式从计算机的 I/O 口输进 LED 显示屏的数据处理系统和控制单元以及音响系统。经处理的视频信息在控制单元信号控制下准确地传送和分配到对应的显示模块的接收电路,驱动 LED 发光。当每一粒 LED 的驱动电流同样的情况下,每一 LED 发光点给人眼所感受到的亮度完全由驱

动时间所决定。换句话说,对于某帧的图象对应该点图象的辉度(或灰级)完全由驱动脉宽所决定。这样,对应不同的象点有不同的辉度,在屏幕上便把该帧的图象在LED显示屏上重现出来。

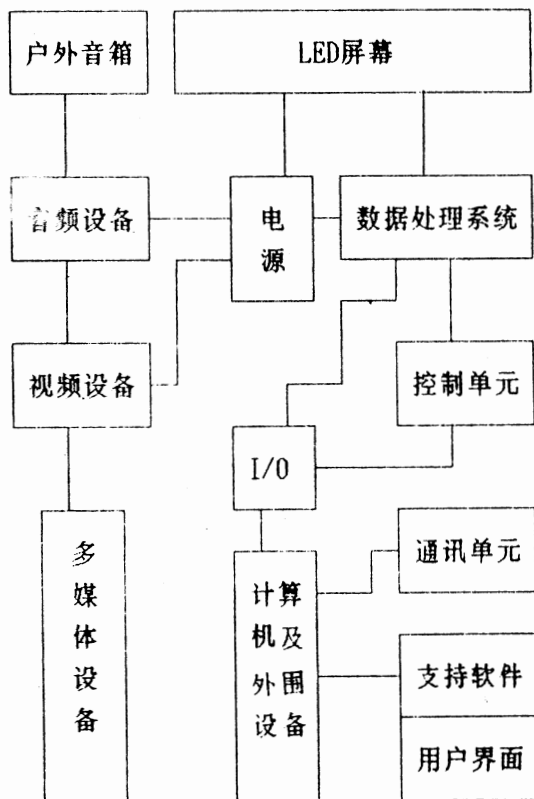


图1 户外LED全色大屏幕视频显示系统工作原理

三、户外LED全色大屏幕视频显示系统的设计要求

户外LED全色大屏幕视频显示系统主要用于在环境亮度极高的环境下进行图象和图文显示。因此,在设计上必须解决的是提高屏幕的亮度和图象清晰度。

1、亮度

由于LED管的发光区域主要集中在P—N结区,更确切地说它集中在P—N结的周边。通常LED芯片为 $20\mu\text{m}$ — $30\mu\text{m}$ 的小方块,几乎可以把它看成是点光源。因此,如何才能最有效地利用LED的光能是设计上必须考虑的。例如对于HLMP-3950 LED管,当其锥形发射角为 $\pm 24^\circ$ 时的亮度为120mcd,而在 $\pm 32^\circ$ 和 $\pm 45^\circ$ 时,则分别为55mcd和45mcd[1]。显然,想要取得高的亮度,

必然会减少观看角度。因此,在设计上可根据观看环境和屏幕大小,在亮度和观看角度间进行选择,以确定LED封装时的发射角。目前LED生产厂家已设计出一种椭圆形的封装模具,造成LED产生椭圆锥形的发射角。这样,已可使观看角度要求不大的垂直方向的发射角尽量小,而在水平方向有较大的发射角。比较好地解决了LED亮度和观看角度间的矛盾。

另一方面,LED的发光效率如图2所示[2]。它随着工作电流的增加而下降。从特性要求知20mA为佳。因此,在电路上为保证有最佳的观看

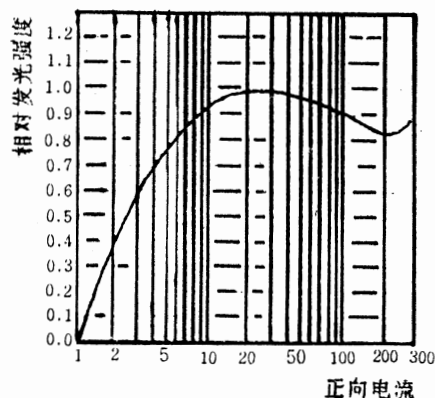


图2 相对发光强度与正向电流之关系

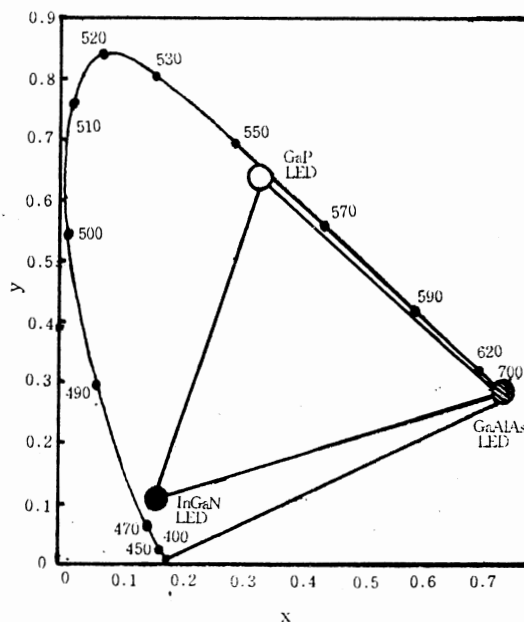


图3 GaAlAs, GaP 和 InGaN LED 的色度座标

亮度,一般采用静态驱动代替常用的扫描驱动电路。以保证最高级辉度级时LED的光发射时间接近一帧图象的时间,即1/50或1/60秒。

2、色彩

目前市面上可供用作红、绿、蓝三基色的LED,峰值波长分别为635nm、560nm和450nm,其光谱半宽度分别为40nm、30nm和70nm[3]。三种颜色在色度图上的座标位置如图3所示[4]。表1分别列出LED和显象管的三基色和白场的色度座标[5]。

图中,△ABC是PAL制彩色电视的色域,

△AFD是目前LED全彩屏的色域,△AGD则是我们修正绿基色后所形成的色域。为了配出绿基色G,我们使用了F(黄绿)和E(蓝绿)作为色素。显然,修正后的LED三基色所形成的色域△AGD绝不逊色于彩色电视机,因此,它对彩色电视信号具有很好的视觉效果还原能力。

必须指出,上述颜色的调配在环境亮度较低的情况下是令人满意的。在太阳光极强的环境下,除了象素外形结构设计上尽量减少环境光的干扰外,保证显示屏色彩的最有效办法,当然是增加LED的显示亮度。

表1 LED和显象管的基色和白色的色度座标

色 座 标	LED色度座标				我国彩色电视标准				NTSC制标准			
	R	G	B	C白	R	G	B	C白	R	G	B	C白
x	0.71	0.33	0.16	0.40	0.64	0.29	0.15	0.31	0.67	0.21	0.14	0.31
y	0.29	0.63	0.11	0.33	0.33	0.60	0.06	0.316	0.33	0.71	0.08	0.316

3、清晰度

增加屏幕象素和象素灰级,可以提高显示屏的清晰度。为兼容计算机和NTSC电视制式,象素结构可选择640(列)×480(行)或320(列)×240(行)。在灰级选择上有16、32、64、128和256级的选择。从理论上来说灰级越多清晰度越高。但在使用上往往会增加数据的处理和传送速度,而且在户外强烈环境光下工作,低灰级的差别往往会为环境光所掩盖,况且灰级增加使数据处理和传送时间增加,从而减少静态显示时间,降低亮度。因此,选择64灰级基本上是合适的。

四、应用实例

由我们制作竖立在广州中国出口商品交易会门前40平方米的户外LED全色大屏幕显示系统,主要由LED显示屏和播出平台组成。

播出平台用于节目的制作、编辑和播出,其硬件包括计算机、多媒体卡、视频设备、音频设备和专用I/O卡等。软件包括三维动画制作、播出设备管理、播出节目预编辑和用户界面等。

显示屏由320(列)×240(行)=76,800象素阵

列组成。其中每一象素由2粒红色、6粒含双绿色芯片和1粒蓝色LED安装在16mm×16mm的方形象素壳体内部构成;以16×16只象素均匀相嵌在一块352mm×352mm正方形的模块上,象素中心间距为22mm。每一模块都是单独的显示单元,带有256只三基色象素、接收电路、驱动电路和供电电源。屏幕由20(列)×15(行)=300块模块以积木方式构成。数据处理系统和控制单元安装在屏幕侧背上,通过信号线与播出平台连接。

测试结果如下:

- 1、白场亮度: >4000Cd/m²
- 2、颜色: 三基色
- 3、灰级: 每一色64级
- 4、帧频: 50/60
- 5、可视角: ±25°
- 6、最大功耗: 70千瓦
- 7、平均功耗: 30千瓦
- 8、功能: 显示视频图象、计算机图文
- 9、工作环境: -10℃—40℃全天候
- 10、寿命: >60,000小时

(转第7页)

LED 显示屏多媒体编播平台 METS 的设计与实现

中山大学电子机械研究中心[510275] 余 阳

摘 要: 本文论述了将新兴的多媒体技术应用于 LED 显示屏控制的方法。详细阐明了多媒体编播平台(METS)的软硬件组成、结构、功能特点及关键的实现技术。

关键词: 多媒体 编播系统 LED 显示屏 自动播放

一、METS 的研究背景与设计目标

目前,LED 显示屏由于亮度高、色彩鲜艳、寿命长、价格适中等特点,正越来越多地运用于广告、证券、银行、车站等公众信息发布领域。从控制方式来看,LED 屏的第一代产品是图文电脑屏,仅能显示由电脑控制的图文信息;第二代视频显示屏,对视频信号采用独立的解码、数字化部件,不通过电脑,直接驱动显示,虽然通过切换可兼容图文屏的能力,但在控制上是独立的,可编程及自动化能力很差,在显示效果上,缺乏综合显示能力;随着多媒体技术的日趋成熟、逐渐用于 LED 屏的控制,它对声音、图像(包括视频)、动画、图形、文字等多种媒

体信息的综合显示效果和连接控制能力,使 LED 屏的显示效果更加多彩,控制更加自动化,形成了第三代产品——多媒体 LED 屏。

由于 LED 屏在我国的开发应用毕竟刚刚开始,各单位所研制的 LED 屏与多媒体电脑的接口并无标准化。因此,目前市面上虽有多种国内外多媒体制作平台,却无法满足不同 LED 屏的独特要求,充分发挥它们的潜力,也无法满足很多用户对自动控制的要求。电视台的自动播出系统,虽在自动播出方面功能较强,但设备昂贵,同时,现有系统也没有考虑 LED 屏的特点。市场的需求,促使我们开始为 LED 屏度身定做一个多媒体编播平台 METS。

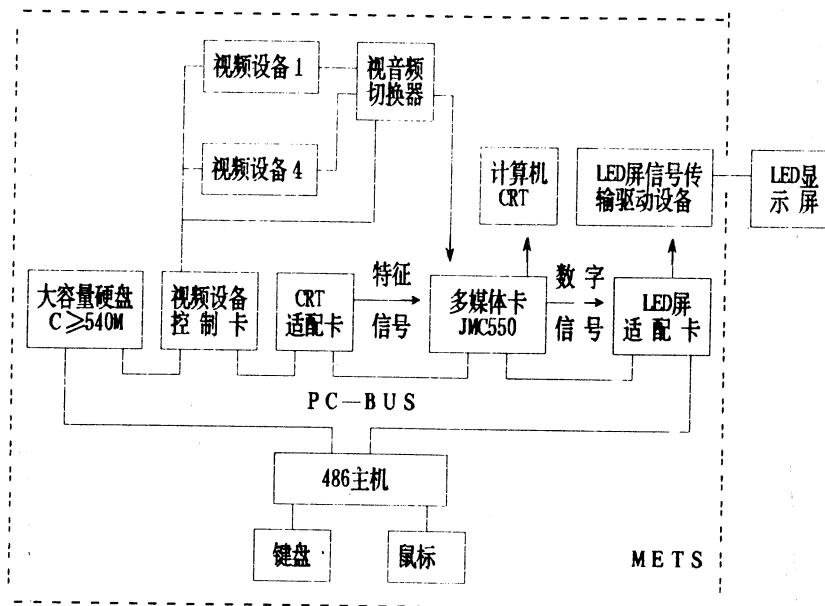


图1 METS 硬件系统结构

我们的设计目标是:首先在输出质量和价格上,METS 应适合 LED 屏的需要,它应继承一般多媒体制作平台对各种媒体综合显示,连接组合的能力,同时要努力满足无人值守、定时插播、全天候播出的需求。

二、METS 的系统结构

1、硬件环境及组成

METS 主要由 486 微机、多媒卡(JMC550)、视频设备控制卡、视音频切换器、LED 显示适配卡、LED 屏信号传输驱动设备、视频设备(家用录像机、影碟机、摄像头等)等部分组成,其结构见图一。

486 微机主频 60 兆以上,8 兆内存,540 兆以上硬盘。

多媒体卡采用银河公司的 JMC550 视频实时压缩/回放卡,该卡能将视、音频信号实时压缩于硬盘,且可实时同步回放,同时也可直接将视频信号与 VGA 信号叠加输出。

视频设备控制卡是我们针对松下家用录像机、影碟机等视频设备开发的控制卡,通过它计算机可以完成原来的手工操作,如快进、倒带、暂停、播放等功能。由于 JMC550 卡只有一组视、音频输入口,故我们还设计了一个视、音频切换器(4 选一),也由该卡控制。

LED 显示屏的适配卡和信号传输驱动设备负责计算机与 LED 屏间的控制及数字信号传送,可控制 LED 屏的整体亮度、开屏、关屏、定格等功能。

2、METS 软件系统的结构

如图二所示,METS 软件系统是由文件服务、节目编辑、播出等功能模块组成的。文件服务完成文件存取等功能;节目编辑包括顺序节目和定时节目的编辑;播出模块首先将对用户直观的程序描述转换成 MEL 语言描述(MEL 语言是我们为多媒体编播设计的专用语言),再交由 MEL 的解释执行系统具体实施显示播出。

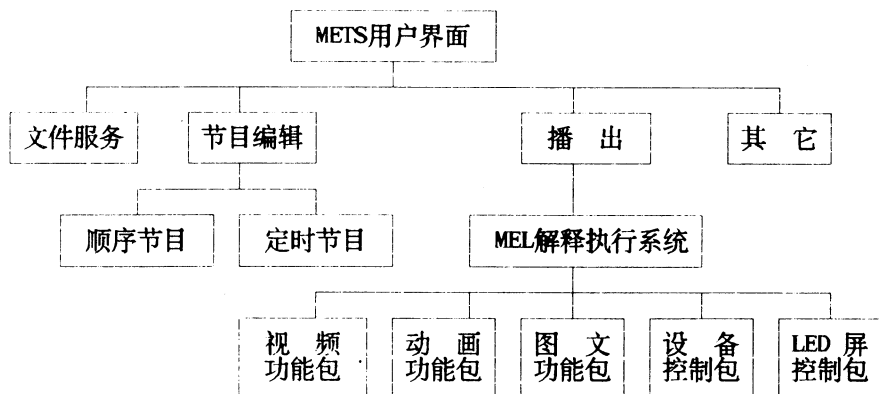


图 2 METS 软件系统结构

三、METS 的主要功能及特点

1、丰富多彩的多媒体综合显示效果

METS 不仅能单独显示图形、文字、图像、动画等信息,并且能将它们叠加以产生丰富的播出效果。如视频字幕功能,图形、动画与视频的叠加产生的简单视频特技效果等等。图文信息有几十种出图方式,支持 45 种可平滑无级放大的矢量汉字。

2、完备多变的播出控制

METS 的播出单位是一个“节目”,一系列的节目可构成顺序节目单和定时节目单,顺序节目的顺序可任意调整,还具有循环、子程序等控制结构,这

两类节目可单独使用,也可结合使用。如:在广告系统中,可将客户提供的视频广告通过 JMC550 压缩到硬盘,由于广告一般只有几十秒长,所以 540 兆硬盘即可满足一般广告公司的要求。另外计算机还可制作动画、图文广告,这时,如果顺序节目由录像机、影碟机等视频设备播放的故事片、电视剧、音乐会、球赛等组成,而把广告、报时、天气预报,财经信息等作为定时节目,进行定时定频率插播,则使整个播出内容更加多彩引人,就像一个小型的电视台。另外,METS 还能控制压缩视频、动画、图文信息的播放速度、暂停/运行等。

3、无人值守

当一个节目单排定, METS 即可自动播放, 对于录像机等视频设备, 也能自动控制它的倒带、暂停、播放、快进、几个视频信号的切换等, 只要不换带或碟就不再需要人工操作, 一般使用 2—3 台视频设备即足以达到丰富播出内容的目的, 大大减轻人的劳动强度和人为出错的机会。

4、全天候适应环境播出

LED 显示屏具有 4 级整体亮度调整功能, 计算机可通过 LED 屏适配卡控制, 结合 METS 的定时功能, 可根据一天内日光强弱的变化、及时自动调整 LED 屏的整体亮度, 既保证了画面清晰、又不至于太亮刺眼, 保证了播出效果, 同时达到节能、延长器件寿命的目的。

METS 的其它功能及特点因篇幅所限, 不再详细列出。

四、METS 的主要实现技术

1、视频设备控制卡

由于家用录像机等视频设备与微机并无标准化控制接口, 因此, 当前的多媒体电脑只能被动、盲目地去接收来自它们的视音频信号。限于这种情况, 我们也只能针对特定的松下系列机进行改装和设计。为了不修改视频设备内部线路, 我们对这些设备的遥控器进行了改装, 使之与视频设备控制卡相连, 使计算机可以模拟按键, 完成各种功能, 这一技术是实现“插播”和无人值守的重要保证。

2、软件系统的分层设计和 MEL 语言

整个 METS 软件系统可分为两大层次: 用户界面、MEL 解释执行系统。MEL 语言试图提供控制多媒体演播的各种功能“原语”, 同时又具有条件、循环、子程序、定时任务等控制结构。

简单的“原语”可组合出变化多端的图文特技和控制方式。使用 MEL 语言作为界面将 METS 分层, 不但降低了各层开发的复杂度、提高了各层的独立性, 便于开发人员分工协作, 而且以 MEL 语言作为开发多媒体演播系统的公用基础, 使得开发其它的应用非常简便, 例如开发火车站的信息显示系

统, 只需更改用户界面即可, 这极大地提高了此类软件的生产效率。

3、动画接口及矢量字库接口

当前制作三维动画最著名的软件是 Autodesk 公司的 3D studio, 它产生的动画文件格式 FLI/FLC 得到了众多软件的支持。为使播放 FLI/FLC 动画的能力与 METS 的控制特点一致, 我们分析了 FIL/FLC 文件的结构, 写出了—个软件库, 它可方便地截取动画的任何一部分, 在屏幕任何一个显示窗口播放, 并且可调整播放速度、控制暂停/运行。

我们与得力软件公司合作, 得到了他们支持 3Dstudio 的 45 种字库的结构, 并开发了读取、变形、还原矢量汉字的软件包, 使 METS 的字幕, 文字广告、二、三维动画在字体方面保持了一致性。

4、前后台双道程序的运行与通信

MEL 解释执行系统的运行采用双道程序前后台运行, 前台负责解释执行 MEL 命令, 后台负责监控定时任务和任务执行时间, 后台可向前台发出请求“中断”信号, 前台在每个 MEL“原语”之间响应这些“中断”, 它为定时播放提供了技术基础。

五、小 结

MEL 自今年初研制成功, 已在数家 LED 屏广告公司使用, 得到客户好评, 目前, 我们正接收用户反馈意见, 考察新技术, 以求研制出更高性能价格比的多媒体编播平台。

参考文献:

- [1]刘武、梁妙园等,《多媒体演播系统的设计与实现》,全国第二届多媒体会议论文集,1993.10,P117
- [2]余阳,《MEL 语言及其解释执行系统》,本单位内部资料,1995.3
- [3]北京银河电脑公司,JMC-550 用户手册,1993—1994 版
- [4]北京银河电脑公司,MDK3.0 多媒体制作平台用户手册,1993—1994 版

(本文收稿日期:1995.10.8)

(接第 4 页)

五、结 论

实践证明全固体型户外 LED 全色大屏幕视频显示系统具有亮度高、彩色好、功耗低、寿命长、造价低,以及具有播放视频图象和计算机图文的功能,适用于交通、广告、股票金融、公共事业等领域播放交通信息、商业广告、国内外新闻、金融信息等。是一种理想的户外全色视频显示设备。

参考资料:

- [1]Optoelectronics Designer's Catalog. p.3—17.
- [2]Optoelectronics Designer's Catalog. p.3—19.
- [3]Optoelectronics Designer's Catalog. p.3—74.
- [4]刘寿听编著,黑白电视与彩色电视. p.139.
- [5]Tokushima, SID 95 DIGEST. p.713

(本文收稿日期:1995.10.8)

LED 群显系统

中山大学电子机械研究中心 左 明 李 军 张苑岳

摘要:介绍了一种利用无线寻呼台作为中心信息发射台,多个 LED 显示屏为接收分站的 LED 群显系统,以及实现 LED 显示屏分站的软硬件设计。

关键词:LED 显示屏 无线寻呼 微处理器

一、引言

随着社会经济的高速发展,产品竞争日趋激烈,各厂家和企业为了提高其产品竞争力,大力加强对产品的广告宣传。据统计,欧美各国近年用于产品广告宣传方面的费用每年高达 2000 多亿美元,可见产品的广告宣传已成了人们促销其产品不可缺少的手段。现有的广告宣传工具虽然形式多种多样,但各自都存在着不足之处,归纳起来主要有:广告信息量少,显示方式呆板,造价高、覆盖面小、管理不便。因此,如何充分利用现有的科学技术,设计制造出广告形式美观多样、信息容量大、广告宣

传覆盖面大、造价低廉、易于管理的广告宣传工具就成了广告工具制造者们的一个有意义的课题。LED 群显系统是由一个中心信息发射台通过无线信号管理多个 LED 显示屏的显示系统,它具有显示形式多样、信息容量大等特点。由于是无线管理,因此可在有效通讯范围内安装 LED 显示屏,覆盖面大,管理方便。

二、系统设计及电路实现

1、设计原理

LED 群显系统的组成可分为中心信息发射台和 LED 显示屏接收站。其结构框图如图 1。

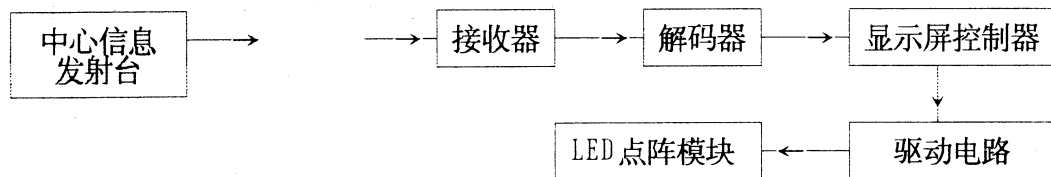


图 1 系统结构框图

中心信息发射台为无线寻呼台,它向各 LED 显示屏发送信息的过程与无线寻呼 BP 机的过程相同,只是寻呼 BP 机为个人资料,而向各显示屏发送的资料为广告信息。中心信息发射台的无线信号编码采用 POCSAG 码,即无线寻呼国际标准一号码。各 LED 显示屏接收站的接收器接收由中心台发送的信号,经放大、解调和滤波后,送至解码器,由解码器对信号进行解码处理,而后将解码后的数据送至 LED 显示屏控制器的微处理器,再由单片微处理器对所得数据进行处理,最后将相应的

信息在 LED 显示屏中显示出来。每个显示屏接收站都有一寻呼地址和独立地址,寻呼地址是所有 LED 显示屏占用中心发射台的一个寻呼号的地址,发射台根据寻呼地址管理这一系统内的所有 LED 显示屏,独立地址用于中心发射台发送信息给该系统内的某指定的 LED 显示屏。此外还有一个公共显示和单个显示的判定地址(或判定位)。

2、各部分电路的作用及设计

① 信号接收器:它的作用是完成对信号的接收、放大、解调、滤波,该电路与 BP 机接收电路相

同。

② 解码器:其功能是对接收器送出的信号进行采样、纠错和校验、信号提取。本系统采用 80C31 单片机作解码器,通过软件编程来实现对信号的解码。其硬件电路较为简单,在此不详述了。根据 BP 寻呼机无线信号编码的特点,解码器编程每次累积采样 32Bit,而后根据 POCSAG 码特性进行代码检测,为保证数据发射的可靠性,可采用对同一信息内容发送二次或二次以上的方式以确保接收无误。为使中心信号站能将信息准确而快捷地发送给各 LED 显示屏接收站,有必要给每个显示屏命名,也

即是赋予每个显示屏各自独立的地址,如 A 显示屏、B 显示屏、C 显示屏等。因此在同一时间内,中心信息发射台发出的信息只为指定的某一显示屏所接收,而不会影响到其它显示屏的工作。由于在很多情况下,要求所有显示屏显示同样的广告信息,则又有必要设置一个公共显示地址,用以判定是所有显示屏共同显示的内容还是单独某个 LED 显示屏独自显示的信息。这样就可以节省中心信息发射时间,提高发射效率。解码器的解码程序流程框图如图 2。

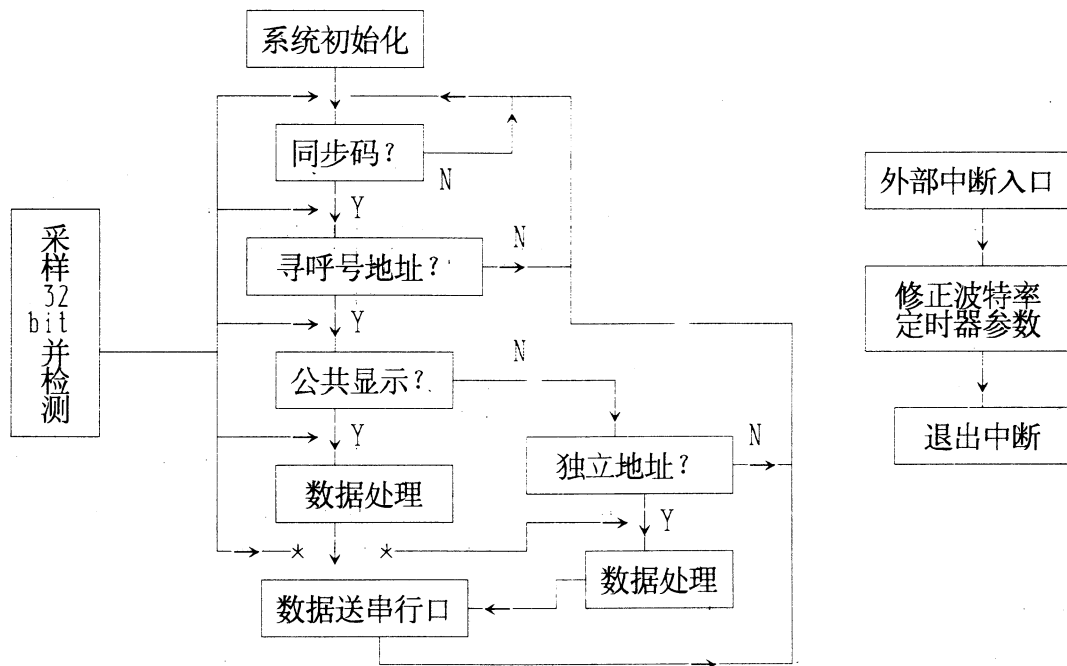


图2 解码程序流程框图

外部中断子程序的作用是用于修改定时器参数。由于是利用软件解码,采样的定时是由微处理器用软件设定定时器的方法来确定,难免有时间的累积误差,即在某一时间内产生一个 Bit 的漂移,为了消除这一累积误差,利用接收器送出的信号下降沿作为外部中断信号源,在中断子程序中对采样定时器进行时间参数校正,以达到准确采样的目的。

③ LED 显示屏:其功能是通过微处理器串行口接收解码器送来的数据,并根据得到的数据从字库中取出相应字符,经微处理器作存储空间编排处理,而后将信息送 LED 显示屏显示。其电路原理框图如图 3,该电路主要由单片微处理器、显示缓冲存储器、扫描信号发生器、数据同步、移位信

号转换器、数据转换器、缓冲器、二级点阵字库存储器、LED 驱动电路、LED 点阵屏幕等部分组成。单片微处理器是 LED 显示屏的核心部分,本系统采用美国 Dallas 半导体公司生产的高速单片微处理器 80C320,该微处理器的工作晶振可达 25MHz,其工作运行速度比 8032 平均提高 2.5 倍(同样工作晶振频率下)。显示屏微处理器的工作过程如下:微处理器通过响应其本身串行口的串行输入中断,获得从解码器送来的数据,按照其显示顺序将其字库代码存放于显示缓冲区的字符映象区;在两行数据传送的间隔期间内,微处理器判定是否要改变显示屏的显示内容,如要则根据字符映象区内相应字符代码从二级汉字点阵字库获取相应的字符点阵字模,并将字符点阵字模写入显示缓冲区的点阵映

象区,实现屏内容的刷新。LED 显示屏微处理器的软件包括一个主程序,一个时间中断程序和一个

串行入口中断程序,均存放于单片微处理器的外接 EPROM 内。软件的结构框图如图 4 所示。

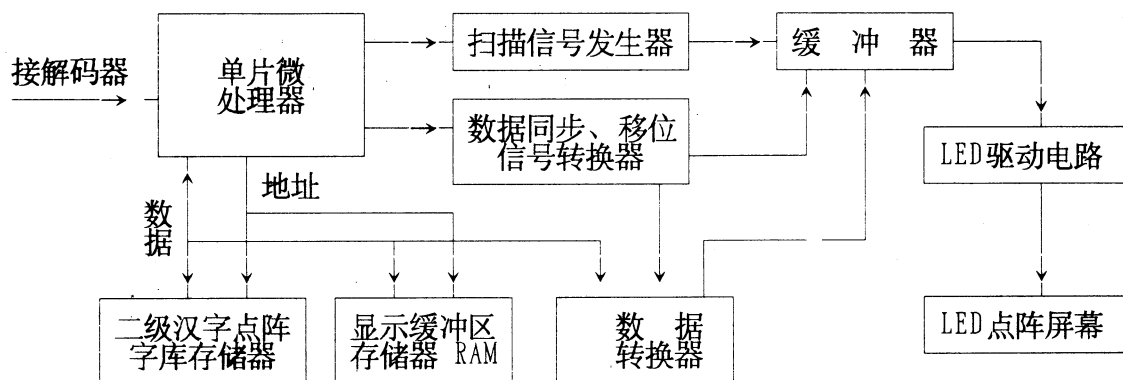


图 3 电路原理框图

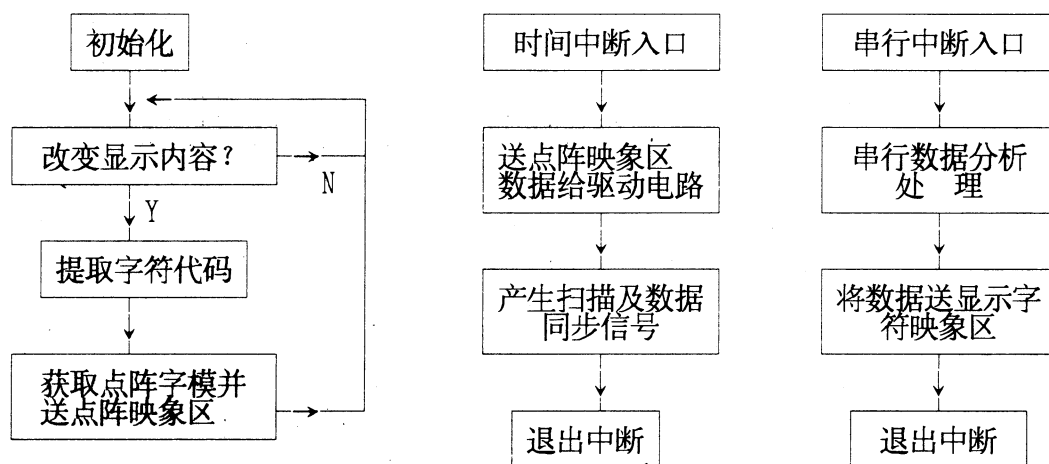


图 4 显示屏微处理器的软件结构框图

显示缓冲区 RAM 的存储空间分为字符映象区和点阵字模映象区两部分,当 LED 显示屏点阵面积确定之后,点阵字模映象区的存储容量就确定了,在这种情况下,RAM 的存储容量大小就决定了 LED 显示屏所能显示的广告(信息)篇幅数目,也即一次所能接收中心信息发射台发射的最大信息容量。对于点阵面积为 480(点)×64(点)的显示屏(双色),如果 RAM 为 64K 字节,则除去点阵映象区所占存储空间,字符映象区所能存储的广告(信息)内容大于 120 幅。二级汉字点阵字库选用 16×16 点阵字模或 24×24 点阵字模,所需 EPROM 的存储容量都大于 64K 字节,而系统所用单片微处理器的寻址范围只有 64K 字节,因此需

采用分段寻址的方法,以 64K 字节为一段。

三、群显的容量、显示编排

从理论上说,LED 群显系统所能容纳的 LED 显示屏数目是无限的,因其独立地址是可以由用户自己根据需要来定义其位数的,即使按 BP 机寻呼号的方式用一个代码来定义独立地址,可用的地址数也多达上百万个。因此,在实际应用中,LED 群显系统的容量是不受限制的。而当系统建立之后,用户若要增加 LED 显示屏接收站,只需中心信息发射台新增独立地址即可,其扩展是非常方便的。

群显系统中的各个 LED 显示屏,它们显示的
(转第 14 页)

图象形态法在棉纱线黑板条干图像处理中的应用

上海工程技术大学纺院 蒋忠仁

摘要:本文介绍一种图像处理的应用技术,将图象的形态变换与形态分析用于棉纱线的图象处理及质量检测。之前,厂家把被测对象纱线平行地绕在小黑板上,呈现所谓“黑板条干”图,然后我们把该图输入电脑,作图象处理。

关键词:数学形态学 图象形态法 自相关 腐蚀 膨胀 孤岛 疵点 棉结

一、引言

微机图像处理技术是一门新兴的学科。它作为后起之秀,目前已被广泛地应用于信息处理等领域,我国在图像处理技术方面的应用也日益广泛。

二、技术背景

1、纱线检测

细度是纱线的重要指标。纱线细度不同,性能和风格也不同,它用单位长度重量来表示。纱线质量检测具有其必要性,纱线作为半制品,不仅在很大程度上决定纺织品的外面和内在质量,而且还对纺织生产过程的稳定性有着重大的影响。另外,正确评定纱线的质量也是商业和贸易的需要。

纱线的细度不均,是指沿纱线长度方向(径向)的粗细不均,也是评定纱线质量的重要指标。测量细纱不匀率的方法有多种,分为定性和定量两大类。目前我国采用的主要有测长称重法,目光检测法,乌斯特均匀度仪法等。目前在世界主要纺织生产国中,黑板条干仍保留着广泛的应用领域。

检测标准和评级检验沿用国标 GB9996-88,标准中规定:如有严重疵点,条干品级定为二级,如有严重规律不匀,评为三级。否则将黑板样照比较,根据粗节粗细及数量,阴影深浅及面积进行评定。

2、硬件概况

黑板中大量的不匀信息的记录和处理,采用286以上微机;将不匀信息通过扫描仪和采样卡输

入到计算机,这些,统称输入设备。其中扫描仪可选手持式或平板式,分辨率可选400 DPI。

按照纺织学规定,在采样黑板条干时,先盖上一块衬片,衬片是一块黑色的硬纸板,上面开有对称的5个50×50mm的窗口,每个窗口应有20条纱线。

三、技术方案

形态法用于图像处理的技术方案可以下面的框图作概括。

1、图像预处理

为了使对图像的处理更具准确性,在对图像正式处理之前,必须对二值化图像进行预处理。所谓预处理,包括以下四种处理:1、纱线断裂处需要进行弥补。2、填补纱线内部的空隙。3、切除那些会造成疵点误判的毛纤。4、除去孤岛。

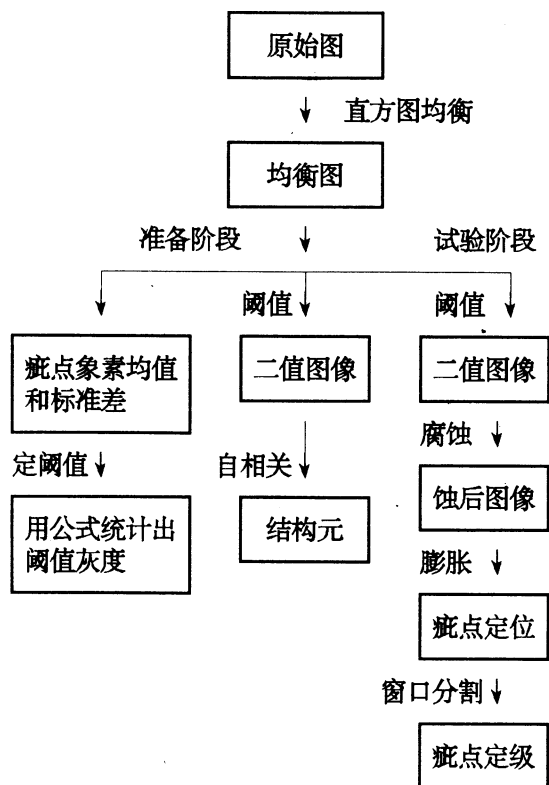
2、直方图均衡和图像二值化

直方图均衡法对像素的灰度来一个再分配,使图像分布更趋一至性,在许多场合,直方图等分法给出的图像细节对人眼的感受要比原始图更易分辨,而原始图毕竟亮度的梯度较小。

3、自相关法

自相关技术能合并一幅图像的所有部分,也可用来表征重复结构,在该过程中我们用自相关来确定织物图像中经纬纱的重复块的大小。

我们可以用自相关法或平均法来计算纱条的平均直径。根据平均直径可以确定下面所要进行的腐蚀和膨胀的结构元。



4、数学形态学

数学形态学 (Mathematical Morphology) 首先是由 Mathrom 和 Serra 于 1982 年提出并用于图像处理的, 现在已成为数学图像处理中十分有力的工具之一。该方法运用形态变换的基本理论, 把数学图像看成是一种点的集合, 选用不同的结构元, 对此集合进行不同的集合运算, 从而达到对数学图像的各种处理。这种方法具有运算简单, 结构元选取方便, 可高速并行处理等特点。因而得到人们普遍的关注。

数学形态学的基础和表达语言是集合理论。数学形态学中的集合可描述二值图像或灰度图像中显示出的形态, 它是一种基于图形的数字图像处理方法。

如果原始图像是一个二值信号, 则形态学变换将其直接看作一个集合, 并用集合运算进行处理, 如果原始图像是一个灰度图像, 则形态学变换先将其变换在函数的本影, 然后对其得到的集合进行集合运算, 最后由处理后的本影推出处理后的函数。

利用数学形态法对纱线图像进行处理和分析, 使课题更具实用性和推广性。

5、腐蚀和膨胀

腐蚀和膨胀是运用上面所确定的结构元对图像进行处理的两种运算方法, 一条纱条经过腐蚀后, 那些小于平均直径的部分即被腐蚀了, 只留下那些大于平均直径的部分。然后经过膨胀后, 由于小于平均直径的部分已被腐蚀了, 因此只对大于平均直径的那部分进行膨胀运算, 结果疵点被还原, 从而我们可以按照留下的疵点进行评价。

6、疵点定级

经腐蚀膨胀处理后的图像, 只剩下了疵点, 这些疵点中包括粗节和棉节。直径大于 2 倍平均直径, 长度大于 5mm 的为棉结。直径大于平均直径, 小于 2 倍平均直径, 而长度大于 5mm 的为粗结。

四、形态处理

对预处理过的图像进行处理, 具体包括三个主要步骤: 1、对原图像文件进行解码; 2、对图像进行形态操作; 3、输出一系列目标文件。

1、结构开闭

这里所讲的处理是指对图像进行形态运算, 如腐蚀, 膨胀和结构开 (先腐蚀, 后膨胀), 和结构闭 (先膨胀, 后腐蚀), 该过程包括简单地对一个二值图像的像素作叠加或搬移。依据的法则要看邻个的像素是什么类型。

数学形态学的四个基本算符是: 腐蚀 (扩张)、膨胀 (侵蚀)、开启、闭合。

以 Z^2 表示所有有序元素对 (a, b) 的集合, a, b 是实整数 Z 集合里的整数。假设目标 A 和结构元素 B 代表 Z 中的集合, 让 (B) 代表 B 的平移, 若其起始位置在 X , 而 \bar{B} 代表 B 的映象, 这样有: $\bar{B} = \{X | X = -b, b \in B\}$; A 被结构元素 B 腐蚀记作 $A \ominus B$, 定义为所有满足 (B) 被包含在 A 里面条件的点 X 的集合, 那就是 $A \ominus B = \{X | (B)_x \subseteq A\}$; A 被结构元素 B 膨胀记作 $A \oplus B$, 定义为所有满足 (B) 和 A 具有一个非空交集条件的点 X 的集合, 那就是 $A \oplus B = \{X | (\bar{B})_x \cap A \neq \Phi\}$, 这里 Φ 代表空集。

开启 $(.)$ 和闭合 $(.)$ 是一对算: 数据集 B 和结构元素 K 的开启运算可表达为 $B \cdot K = (B \ominus K) \oplus K$ 数据集 B 和结构元素 K 的闭合运算可表达为 $B \cdot K = (B \oplus K) \ominus K$ 。

腐蚀, 从数学上来说, 就是对结构元和图形取交集。从几何上来说: 就是一个图形从它的轮廓开始向图形内部 (一个连通区域内) 进行缩小, 其具体

缩小的程度由结构元的尺寸和大小来决定。

膨胀,从数学上来说,就是对结构元和图形取并集。而从几何上来说,就是一个图形从它的周边轮廓开始向图形以外进行同等程度地放大,每次放大的量也是由结构元的大小决定的,由此可以看出结构元的选取是十分重要的。

2、结构元

在数学形态法中选择结构元素的形状和大小是关键性的一步,我们实现这一方案的目标是一方面所选的结构元必须足够大以腐蚀那些与普通织物结构大同小异的窗口,同时必须足够小以保留经腐蚀之后的图形里的重复斑点。既然自相关法能够借用来定普通织物重复单元的尺寸和形状,(该单元的大小和形状依次引导了结构元素大小形状的选择)因此也能用它来选择结构元的尺寸和形状。

要对一个图像进行腐蚀和膨胀,要得到好的效果,还有一个因素必需考虑:那就是在已知结构元的大小的情况下,究竟是用较小尺寸的结构元进行多次作用呢,还是选用较大的结构元进行一次作用,以及结构元的中心取在何处。这些都必须根据经验得出,经过多次试验而定。

另外还有一点需要强调:那就是腐蚀和膨胀只有对图像的边界才能起作用,在图像的内部不起作用。

在本课题中,纱条样板是由机器按一定的间距,用同样大小的拉力绕制而成,因此纱条的分布是等间距的垂直平行分布,所以根据自相关定义可知,对于等间距的垂直平行分布的纱条,在其X方向上是具有相关性的。对条干来讲,研究Y方向的相关性意义不大,只有X方向上的相关性才具有明确的物理意义。具体方案确定时,我们选用的结构元是一条直线,其宽度等于纱条的平均直径,其形态可以象纱线那样自由弯曲,结构元与图象作形态运算的起点不选在图象的中心,而选在纱条的左上角。

3、目标文件的生成

可供用户选择生成的目标文件大致有:a)各种预处理得到的系列文件;b)各次腐蚀得到的文件;c)经过“结构开”处理后得到的文件。

限于篇幅,只对典型程序作介绍,下面选择对图像形态进行腐蚀的fs()这个函数作介绍,其程序如下:

```
void fs(dfs,p)
int dfs;
char huge * huge * p;
```

```
{
    FS1 fs1;
    int i,bnum;
    char s0,s1;
    char huge * mp;
    unsigned char value,temp;
    for(i=0;i<depth;i++){
        mp = * (p+i);
        s0=1,temp=0x80,bnum=0;
        do{
            value = ( * (mp+bnum))&temp;
            if(value==temp) {
                s1=1;
                if(s0!=s1) {
                    fs1.mask=temp,fs1.bnums=bnum;
                    fs1.ds=dfs,fs1.ps=mp;
                    invert(dfs,&fs1);
                    s0=1;
                    temp=fs1.mask,
                    dfs=fs1.ds,bnum=fs1.bnums;
                }
            }
            else{
                if(temp==0x01) bnum++,temp=0x80;
                else temp>>=1;
            }
        }
        else{
            s0=0;
            if(temp==0x01) bnum++,temp=0x80;
            else temp>>=1;
        }
    } while(bnum<bytes);
}
```

```
void invert(di,pi)
int di;
FS1 * pi;
{
    int n;
    unsigned char mask1;
    n=0;
    do{
        mask1=~(pi->mask);
        * ((pi->ps)+(pi->bnums))
        = ( * ((pi->ps)+(pi->bnums)))&(mask1);
        n++;
        if(pi->mask==0x01)
            pi->mask=0x80,pi->bnums++;
        else pi->mask>>=1;
    } while((n<di)&&(pi->bnums<bytes));
}
```


}

其中,在腐蚀时需经常重复将连续的若干个白点变成黑点,所以设立了一个 invert() 的函数,它的输入变量为 di,pi。di 是代表需要连续作转换的点数,pi 则是指向一个如下结构的指针:

```
typedef struct {
    unsigned char mask;
    char huge * ps;
    int ds;
    int bnums;
} FS1;
```

这个结构正是用来完成 fs 到 invert 的参数输入和输出的。它的每一个域的定义如下: unsigned char mask 是一个屏蔽字,因为在本图像中,只有黑白两个值,1 个 bit 代表 1 位,所以对点的处理就变成对 bit 的处理,因此在处理之前必须将该位取出来,然后才能处理。在完成后,该屏蔽字右移一位,处理下一个 bit。

char huge ps; ps 是一个指向某一图段的第一个数据地址的指针。要完成从特定点开始连续若干个点的转换,必需首先告之该点的数据所在的图段的地址,以及在每个图段之中的偏移量, bnums 就是表示这个偏移量的。

int ds 代表需要连续变换的点数; int bnums

需要做连续转换的第一个白点在该图段中的偏移量。

膨胀和腐蚀的程序极为相似,也有一个用来将黑点转换为白点的函数 hbb, 用来传递参数的结构也十分相似,只是在 hbb 中,每处理完成一个 bit,必须将屏蔽位向左移一位。(其程序略)

最后是回写 Tiff,与解 Tiff 相对应,回写 Tiff(packtiffwrite()) 主要分三部分。第一部分是写文件的头 writetiffheader(); 第二部分是写文件的数据 writetiffline(); 第三部分是写文件目录项的 writetiffdic()。

五、结束语

图像处理是一门崭新的学科,它在越来越广的范围内占据着越来越重要的地位。但如何将图像处理技术与具体的某一行业很好地结合起来,发挥其优势,仍需我们认真研究、不断努力。

参考文献:

- [1] 吴敏金著,《图象形态学》,上海科技出版社,1991 年
- [2] 马建波著,《C 语言图像处理程序集》,海洋出版社,1992 年

(本文收稿日期:1995.9.27)

(接第 10 页)

信息内容即可能相同,也可能不同,同一个显示屏中不同的信息内容显示的时间也是不相同的。依靠独立地址可将信息内容发送至该系统中指定的 LED 显示屏,公共显示地址可使中心发射台一次将同一条信息发送到该系统下的所有 LED 显示屏。中心信息发射台所发送的每幅显示数据包括显示字符代码、显示方式、显示时间,LED 显示屏的微处理器就是根据这些信息来完成一幅屏幕内容的显示。可见,它在显示内容和显示时间上是非常灵活的。显示时间包括:信息在什么时间显示及显示的时间长短。每幅信息显示时间的判定是由微处理器的软件时钟来确定,为了保证系统内所有显示屏在时间上的一致性,可对系统内的软件时钟进行校正,即在一定时间间隔,由中心信息发射台将标准时间以公共显示数据的方式发送出去,各显示屏接收站收到这一信息后,自动将其内部的软件时钟的时间校正。

四、结束语

LED 群显系统是一种价廉、可靠、实用性强、易于管理的宣传工具,它不仅适用于广告宣传,而且也适用于交通无线管理,利用显示屏来表明各路段的突发情况。而且对于区域较大,显示屏分散的指示系统,它也是一个理想的显示系统。它在显示编排上的灵活性及系统管理和扩展上的方便性,必将为它的应用提供广阔的前景。

参考文献:

- [1] 万先觉等编,无线电寻呼——BB 机,航空工业出版社,1993
- [2] 田昕编,BP 机使用与维修手册,人民邮电出版社,1993
- [3] 孙涵芳等编,单片机原理及应用,北京航空学院出版社,1987
- [4] 杨爵等编,实用纠错编码,中国铁道出版社,1988
- [5] [美] Dallas 半导体公司,PRODUCT DATA BOOK (增补本),1992—1993

(本文收稿日期:1995.10.8)

SLPC 可编程调节器在锅炉自动控制中的应用

山东省莱芜钢铁总厂自动化部 隋景明

摘要:本文介绍用可编程调节器 SLPC 控制工业锅炉汽包水位,实现汽包水位三冲量自动调节。

关键词:SLPC 可编程调节器,三冲量自动调节

1、前言

SLPC 可编程调节器具有组态灵活,适用范围宽,智能性强,用户可多次编程,硬件通用性好,现场安装调试周期短。已在日本横河技术现场应用十多年,具有很高的可靠性。因此,选用一台 SLPC 可编程调节器,配上相应的用户程序,通过现场信号调节给水调节阀,实现锅炉汽包水位的定值精确调节。

2、方案的确定

我厂动力部的 2 号蒸汽锅炉是一台额定负荷 75 吨/小时的蒸汽锅炉,运行负荷平均 60 吨左右/小时。水、汽进出量较大,因此汽包水位的变化比较敏感。就要靠人工调节维持水位的基本不变。应用 SLPC 可编程调节器,考虑高压、高温情况蒸汽的流量补正,协调蒸汽流量与给水量的平衡,并以此克服水位测量由高温、高压而产生的虚假特性。通过二级 PID 调节的串级控制,实现水位的自动,调节达到水位的相对稳定。系统的串级 PID 调节控制方案示意图见附图 1。

3、控制方案具体实施

有关工艺情况如图 1 所示。汽包液位范围为一 320~+320mm。蒸汽流量为 0~100 吨/小时,给水流量为 0~100 吨/小时。由于出口压力、蒸汽温度作为参考信号,对出口蒸汽流量实行在线实时温压补正,补正后参与串级控制中副回路的 PID 调节。这样只用一台 SLPC 可编程调节器就可实现常规仪表 2 台调节器或者固定程序调节器(也称为智能调节器)2 台才能实现的串级控制。该系统既考虑了蒸汽流量与给水量不平衡对液位的影响,同时,又考虑了高压、高温的蒸汽对蒸汽流量测量精度的影响,构成了一个较完善的多冲量水位自动调节控制系统。

依方案设计,用 SPRG 编写了 42 步的用户程序,具体程序清单略。

4、系统运行的几种状态

系统运行时有三种状态:自动状态、手动状态、异常状态。

自动状态为正常运行状态。根据事先设定调控给水阀,维持水位相对稳定在一定精度范围内。异常状态为非运行状态。系统出现异常故障时,自动转为硬件后备状态,并维持以前状态输出,同时点亮前面故障指示灯,提醒操作人员立即采取措施。手动状态类似(外作用)于手操点动输出,其原理相当于一个单冲量点动控制,通过释放输出到手操或操作器。

5、结束语

本系统应用在我厂动力部 2 号 750t/h 锅炉上,自 95 年 2 月份投入运行以来,大幅度的提高了水位的调节品质,由原来的手动调节的 $\pm 75\text{mmH}_2\text{O}$,提高到 $\pm 15\text{mmH}_2\text{O}$ 的范围内。给锅炉生产提供了可靠的运行条件,提高了锅炉的生产能力以及安全系数。从投运至今,运行情况一直稳定可靠,各项控制功能和指标符合生产工艺要求,无需专人看守,解除了生产工人精神上的紧张心理和劳累之苦。该系统是同类锅炉应用较为成功的一例。

(本文收稿日期:1995.9.16)

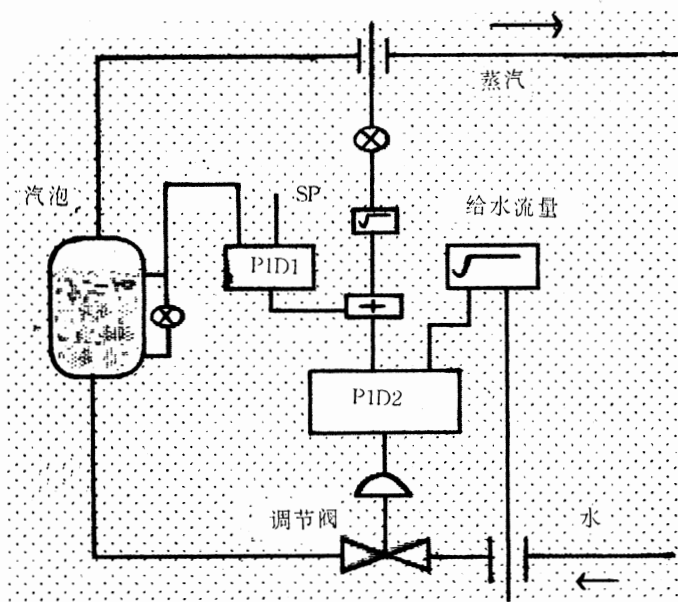


图 1

FOXPRO2.5 源程序精华点滴

哈尔滨理工大学管理系 顾建飞

如何掌握比较好的软件开发方法,形成好的程序风格,这是一个重要的研究课题。几乎所有搞软件开发的人都是在经过大量的实践逐步积累一些经验,慢慢地养成自己的一套开发方法和风格。至于他的方法和风格是否最好,只有他自己知道,笔者认为剖析优秀的软件系统是学习开发大系统软件的最捷径的方法。事实上国外若干大公司的软件人员都是花大价钱去买源程序,通过读源程序,并基于源程序之上开发实用软件包。这就是外国人软件水平高的一个秘诀。

FOXPRO 为了充分体现其开放性,其菜单生成器和屏幕生成器的源程序是用 FOXPRO 本身编写的。其参数是 DBF 格式的数据库装的(屏幕数据库.SCX 和.SCT,菜单数据库.MNX 和.MNT)。读者欲详细剖析生成器,可参阅文献[1]。笔者在这里仅抛砖引玉。下面所列程序大多来自生成器源文件,读者领会后,可直接应用于自己的程序中。

一、文件名处理模块

一个文件由盘符,路径和基本文件名和后缀四部分组成,有时只需基本文件名,有时只需盘符和路径,有时需要后缀,那么就需要一些程序模块来处理相应的文件名字符串。

```
* BASENAME (返回文件名)
FUNCTION basename
PARAMETER m. filename
RETURN strippath(stripxt(m. filename))
*
* STRIPEXT (去掉扩展名) *
FUNCTION stripxt
PARAMETER m. filename
PRIVATE m. dotpos, m. terminator
m. dotpos = RAT(".", m. filename)
m. terminator = MAX(RAT("\", m. filename), RAT(":", m. filename))
IF m. dotpos > m. terminator
    m. filename = LEFT(m. filename, m. dotpos-1)
ENDIF
RETURN m. filename
```

```
*
* STRIPPATH (去掉文件名的路径)
FUNCTION strippath
PARAMETER m. filename
PRIVATE m. slashpos, m. namelen, m. colonpos
m. slashpos = RAT("\", m. filename)
IF m. slashpos > 0
    m. namelen = LEN(m. filename) - m. slashpos
    m. filename = RIGHT(m. filename, m. namelen)
ELSE
    m. colonpos = RAT(":", m. filename)
IF m. colonpos > 0
    m. namelen = LEN(m. filename) - m. colonpos
    m. filename = RIGHT(m. filename, m. namelen)
ENDIF
ENDIF
RETURN m. filename
*
* STRIPCR (去掉回车换行)
FUNCTION stripcr
PARAMETER m. strg
strg = ALLTRIM(strg)
i = LEN(strg)
DO WHILE i >= 0 AND INLIST(SUBSTR(strg,i,1),
CHR(13),CHR(10))
    i = i - 1
ENDDO
RETURN LEFT(strg,i)
*
* ADDBS (给文件名加上反斜杠)
FUNCTION addbs
PARAMETER m. pathname
PRIVATE ALL
m. pathname = ALLTRIM(UPPER(m. pathname))
IF ! (RIGHT(m. pathname,1) $ '\:') AND ! EMPTY
(m. pathname)
    m. pathname = m. pathname + '\'
ENDIF
RETURN m. pathname
*
* JUSTFNAME (只返回无路径的文件名)
FUNCTION justfname
PARAMETERS m. filename
PRIVATE ALL
```



```

IF RAT('\',m.filename) > 0
    m.filename = SUBSTR(m.filename,RAT('\',m.
filename)+1,255)
ENDIF
IF AT(':',m.filename) > 0
    m.filename = SUBSTR(m.filename,AT(':',m.filename)
+1,255)
ENDIF
RETURN ALLTRIM(UPPER(m.filename))
*
* JUSTPATH (只返回给定文件的路径)
FUNCTION justpath
* Return just the path name from "filename"
PARAMETERS m.filename
PRIVATE ALL
m.filename = ALLTRIM(UPPER(m.filename))
IF '\ $ m.filename
    m.filename = SUBSTR(m.filename,1,RAT('\',m.
filename))
    IF RIGHT(m.filename,1) = '\' AND LEN(m.filename)
> 1 ;
        AND SUBSTR(m.filename,LEN(m.filename)-1,1)
<> ':'
        m.filename = SUBSTR(m.filename,1,LEN(m.
filename)-1)
    ENDIF
    RETURN m.filename
ELSE
    RETURN ""
ENDIF
*
* FORCEEXT (强制加上一个扩展名)
FUNCTION forceext
PARAMETERS m.filename,m.ext
PRIVATE ALL
IF SUBSTR(m.ext,1,1) = "."
    m.ext = SUBSTR(m.ext,2,3)
ENDIF

m.pname = justpath(m.filename)
m.filename = justfname(UPPER(ALLTRIM(m.
filename)))
IF AT(':',m.filename) > 0
    m.filename = SUBSTR(m.filename,1,AT(':',m.
filename)-1) + ':' + m.ext
ELSE
    m.filename = m.filename + '.' + m.ext
ENDIF
RETURN addbs(m.pname) + m.filename

```

二、进度计处理模块

软件进度计是用来描述程序执行过程的动态情况的。整个过程的任务为100%。当执行到一部分,就根据完成任务的百分之几来象温度计的方式

显示出来,这样就一目了然,进度计在屏幕上设计一个窗口,表示进度计窗口,动态显示完成任务的情况。本文还给出了调用进度计程序的范例。

```

* ACCTERM(激活软件进度计)
PROCEDURE acttherm
PARAMETER m.text
PRIVATE m.prompt
* #DEFINE c—dlgface      "MS Sans Serif"
#DEFINE c—dlgface      "黑体"
* #DEFINE c—dlgsize      8.000
#DEFINE c—dlgsize      10.000
#DEFINE c—dlgstyle      "BI"
public m.g—thermwidth ,m.g—graphic,m.g—outfile
m.g—thermwidth=0 && Thermometer width
m.g—outfile = "12345" && Output file name
m.g—graphic = .F.
* m.nblocks = (m.percent/100) * (m.g—thermwidth)

DO CASE
CASE AT("WINDOWS",UPPER(VERSION())) <> 0
    m.g—thisvers = "WINDOWS"
    m.g—graphic = .T.
CASE AT("MAC",UPPER(VERSION())) <> 0
    m.g—thisvers = "MAC"
    m.g—graphic = .T.
CASE AT("UNIX",UPPER(VERSION())) <> 0
    m.g—thisvers = "UNIX"
    m.g—graphic = .F.
CASE AT("FOXPRO",UPPER(VERSION())) <> 0
    m.g—thisvers = "DOS"
    m.g—graphic = .F.
ENDCASE

if m.g—graphic
    m.prompt = LOWER(m.g—outfile)
    IF TXTWIDTH(m.prompt,c—dlgface,c—dlgsize,c—
dlgstyle) > 43
        DO WHILE TXTWIDTH(m.prompt+"... ",c—
dlgface,c—dlgsize,
c—dlgstyle) > 43
            m.prompt = LEFT(m.prompt,LEN(m.prompt)-1)
        ENDDO
        m.prompt = m.prompt+"..."
    ENDIF

DEFINE WINDOW thermomete ;
    AT INT((SROW() - ((5.615 * ;
    FONTMETRIC(1,c—dlgface,c—dlgsize,c—dlgstyle
)) / ;
    FONTMETRIC(1,WFONT(1,""),WFONT(2,""),
WFONT(3,"")))) / 2), ;
    INT((SCOL() - ((63.833 * ;
    FONTMETRIC(6,c—dlgface,c—dlgsize,c—dlgstyle
)) / ;

```

```

FONTMETRIC(6, WFONT(1,""), WFONT(2,""),
WFONT(3,"")))) / 2);
SIZE 5.615,63.833;
FONT c—dlgface, c—dlgsize;
STYLE c—dlgstyle;
NOFLOAT;
NOCLOSE;
NONE;
COLOR RGB(0, 0, 0, 192, 192, 192)
MOVE WINDOW thermomete CENTER
ACTIVATE WINDOW thermomete NOSHOW

@ 0.5,3 SAY m. text FONT c—dlgface, c—dlgsize
STYLE c—dlgstyle
@ 1.5,3 SAY m. prompt FONT c—dlgface, c—dlgsize
STYLE c—dlgstyle
@ 0.000,0.000 TO 0.000,63.833;
COLOR RGB(255, 255, 255, 255, 255, 255)
@ 0.000,0.000 TO 5.615,0.000;
COLOR RGB(255, 255, 255, 255, 255, 255)
@ 0.385,0.667 TO 5.231,0.667;
COLOR RGB(128, 128, 128, 128, 128, 128)
@ 0.308,0.667 TO 0.308,63.167;
COLOR RGB(128, 128, 128, 128, 128, 128)
@ 0.385,63.000 TO 5.308,63.000;
COLOR RGB(255, 255, 255, 255, 255, 255)
@ 5.231,0.667 TO 5.231,63.167;
COLOR RGB(255, 255, 255, 255, 255, 255)
@ 5.538,0.000 TO 5.538,63.833;
COLOR RGB(128, 128, 128, 128, 128, 128)
@ 0.000,63.667 TO 5.615,63.667;
COLOR RGB(128, 128, 128, 128, 128, 128)
@ 3.000,3.333 TO 4.231,3.333;
COLOR RGB(128, 128, 128, 128, 128, 128)
@ 3.000,60.333 TO 4.308,60.333;
COLOR RGB(255, 255, 255, 255, 255, 255)
@ 3.000,3.333 TO 3.000,60.333;
COLOR RGB(128, 128, 128, 128, 128, 128)
@ 4.231,3.333 TO 4.231,60.500;
COLOR RGB(255, 255, 255, 255, 255, 255)
m. g—thermwidth = 56.269

SHOW WINDOW thermomete TOP
ELSE
m. prompt = SUBSTR(SYS(2014, m. g—outfile), 1, 48)
+;
IIF(LEN(m. g—outfile)>48, "...", "")

DEFINE WINDOW thermomete;
FROM INT((SROW()-6)/2), INT((SCOL()-57)/
2);
TO INT((SROW()-6)/2) + 6, INT((SCOL()-
57)/2) + 57;
DOUBLE COLOR SCHEME 5
ACTIVATE WINDOW thermomete NOSHOW

```

```

m. g—thermwidth = 50
@ 0,3 SAY m. text
@ 1,3 SAY UPPER(m. prompt)
@ 2,1 TO 4, m. g—thermwidth+4 &g—boxstrg

```

```

SHOW WINDOW thermomete TOP
ENDIF
RETURN

```

```

* UPDTHERM(修改进度计)
PROCEDURE updtherm
PARAMETER m. percent
PRIVATE m. nblocks, m. percent

```

```

ACTIVATE WINDOW thermomete
* Map to the number of platforms we are generating for
m. percent = MIN(INT(m. percent), 100)
m. nblocks = (m. percent/100) * (m. g—thermwidth)
IF m. g—graphic
@ 3.000,3.333 TO 4.231, m. nblocks + 3.333;
PATTERN 1 COLOR RGB(128, 128, 128, 128, 128,
128)
ELSE
@ 3,3 SAY REPLICATE(" ", m. nblocks)
ENDIF
RETURN

```

```

* DEACTTHERMO(终止进度计,并释放进度计窗口)
PROCEDURE deactthermo
IF WEXIST("thermomete")
RELEASE WINDOW thermomete
ENDIF
RETURN

```

```

* ACTTEXAM.PRG(软件进度计使用示例)
SET PROCEDURE TO USEAGE
DO acttherm WITH "程序生成进度计正在工作,请等
待....."
NUM=1
DO WHILE NUM<=100
DO updtherm WITH NUM
j=1
do while j<=1000
j=j+1
enddo
NUM=NUM+1
ENDDO
DO deactthermo
SET PROCEDURE TO

```

文献:[1]《FOXPRO2.5 菜单和屏幕生成器剖析与应用》
(翁正科)学苑出版社

(本文收稿日期:1995.9.24)

C 语言汉字平滑移动的设计示例

江西拖拉机发动机厂 黄焕如 湖南师范大学物理系 王 玲

内容提要: 本文详细介绍了利用 C 语言设计的汉字平滑移动技术, 所列举的示例程序由浅入深、循序渐进、界面漂亮, 很容易扩充和修改。

关键字: C 语言、平滑移动、立体窗口

在 C 语言的动画制作、封面和菜单设计中, 汉字的平滑移动能给人以较强的动感和较鲜明的提示, 应用十分广泛。在实际应用中通常采用下述两种方法, 一种方法是将汉字的平滑移动程序驻内存, 使得后续的程序在运行时不能复盖汉字的移动; 一种方法是将汉字的平滑移动和后续程序结合成一个程序。前者不仅仅需要占用一部分内存, 而且还存在诸如 DOS 的不可重入等较复杂的技术问题, 而后者相对来说实现比较简单、修改和扩充都比较容易, 同时汉字的移动画面质量也很高, 本文仅讨论第二种方法。

为了给初学者提供一个快速掌握汉字平滑移动的技术, 本文采用循序渐进、逐步实现的方法, 希望能给读者起到抛砖引玉的作用。

首先看看如何实现西文的移动, 详见程序一。该程序内的核心是移动函数 `movs()`, 其参数 `int x`, `int y` 为屏幕显示坐标, `shw` 为字符串, `len2` 为字符串长度。其设计思想是将字符串的前一“字节”移到字符串最后面, 形成一个新的字符串, 复盖原字符串重新显示, 造成整个字符串移动的效果。

如果把程序一中的英文字符串改成汉字, 运行时不难发现, 因为汉字由两个字节组成, 每次移动一个字节在字符串首尾会产生半个汉字, 影响汉字移动的效果。将搬运“字节”改成搬运“字”, 上述问题即迎刃而解, 详见程序二。请注意, 该程序和程序一运行的结果还有一点不同, 字符串向相反的方向移动, 读者可稍加比较和分析。

至此, 汉字移动的问题已基本解决, 但是屏幕上的汉字移动是按“字节”或“字”作为基本单位搬运的, 产生的移动效果有较强的跳动感觉。显然如果将搬运的基本单位由“字节”或“字”改成“像素”, 即屏幕上的“像素”坐标, 才能真正使得汉字平滑移动。这很自然使我们想起了 C 语言的屏幕图形存储和释放函数。

```
void far getimage(int left, int top, int right, int
```

```
bottom, void far * buf)
```

其中图形的左上角用坐标 (`left, top`), 右下角 (`right, bottom`) 表示, 这些区域的图象将拷贝到由 `buf` 所指向的内存。使用该函数前, 应使用 `imagesize()` 函数指定所需的存储器字节数, 并由 `malloc()` 函数得到内存区的首指针。

```
void far putimage(int x, int y, void far * buf,
int op)
```

该函数将先前由 `getimage()` 函数存储的, 用 `buf` 所指向的内存中的图象释放在起始位置 (`x, y`) 屏幕上。其 `op` 值决定图象的释放方式, `COPY_PUT(0)` 表示复制; `XOR_PUT(1)` 表示与图象取“异或”后复制; `OR_PUT(2)` 表示与图象取“或”后复制; `AND_PUT(3)` 表示与图象取“与”后复制; `NOT_PUT(4)` 表示与图象取“反”后复制。

程序三利用了这类函数实现了汉字的平滑移动。程序中使用 `windows()` 函数产生立体窗口, 以便衬托在限定范围内汉字平滑移动的美感。函数 `strhz()` 用来显示汉字, 其中 `row, col` 是行、列值, `length` 是字符串总长度, `s` 是字符串内容, `attr` 是颜色属性。

在实际应用中, 显然不希望同时出现两行相同内容的汉字, 程序四对程序三进行了修改, 使得储存和释放汉字在同一行进行, 值得注意的是, `putimage()` 函数的释放方式, 一般不宜使用 `NOT_PUT`, 而应使用 `COPY_PUT`, 否则在汉字移动时产生闪烁, 影响汉字平滑移动的质量。

```
/* 程序一 */
```

```
#include "stdio. h"
#include "conio. h"
#include "string. h"
```

```
char movs(int x, int y, char shw[40], int len2)
{
char tms[40], str[40];
```



```

int len1,df=0;
len1=strlen(shw);
while(! kbhit())
{gotoxy(x,y);
strncpy(str,shw,(len1>len2 ? len2-1: len1-1));
printf(str);
if(! df)
{strcpy(tms,shw);
tms[1]='\0';
strcat(shw,tms);
strcpy(tms,&(shw[1]));
strcpy(shw,tms);}
else
{strcpy(tms,shw);
strcat(tms,shw);
tms[len1*2-1]='\0';
strcpy(shw,&(tms[len1-1]));}
delay(400);}
exit(1);
}

```

```

main()
{ clrscr();
movs(30,10," WELLCOME YOU!",15); }

```

/* 程序二 */

```

#include "stdio. h"
#include "conio. h"
#include "string. h"

```

```

char movs(int x,int y,char shw[40],int len2)
{
char tms[40],str[40];
int len1,df=0;
len1=strlen(shw);
while(! kbhit())
{gotoxy(x,y);
strncpy(str,shw,(len1>len2 ? len2-1: len1-1));
printf(str);
if(df)
{strcpy(tms,shw);
tms[1]='\0';
strcat(shw,tms);
strcpy(tms,&(shw[1]));
strcpy(shw,tms);

strcpy(tms,shw);
tms[1]='\0';
strcat(shw,tms);
strcpy(tms,&(shw[1]));
strcpy(shw,tms);}
}

```

```

else
{strcpy(tms,shw);
strcat(tms,shw);
tms[len1*2-1]='\0';
strcpy(shw,&(tms[len1-1]));

strcpy(tms,shw);
strcat(tms,shw);
tms[len1*2-1]='\0';
strcpy(shw,&(tms[len1-1]));}
delay(400);}
exit(1);
}

```

```

main()
{ clrscr();
movs(30,10," 欢迎您使用!",15); }

```

/* 程序三 */

```

#include "stdio. h"
#include "graphics. h"
#include "stdlib. h"
#include "conio. h"
#include "bios. h"
#include "dos. h"
#include "string. h"

```

```

char movs(int xx,int yy,char shw[400],int len2);
void windows();
void * buf; int j=0;

```

```

main()
{
int gd=VGA,gm=VGAHI;
int size;
initgraph(&gd,&gm,"");
cleardevice();
setfillstyle(SOLID_FILL,CYAN);
bar(37,428,588,454); windows(37,428,588,454);
setfillstyle(SOLID_FILL,GREEN);
bar(37,28,588,54); windows(37,28,588,54);

size=imagesize(64,432,559,449);
buf=malloc(size);
movs(25,9, "请比较:汉字的一般移动和汉字的平滑移"
"动的效果。其根本原因是“字节”或“字”的"
"搬运容易产生跳动,而“象素”的搬运显得"
"比较平滑和漂亮",68*2);

free(buf);
closegraph();
exit(0);
}

```

```

}

void windows(x1,y1,x2,y2)
int x1,x2,y1,y2;
{ int i;
  setcolor(WHITE);
  for(i=1;i<3;i++) line(x1-i,y1-i,x2+i,y1-i);
  for(i=1;i<3;i++) line(x1-i,y1-i,x1-i,y2+i);
  setcolor(DARKGRAY);
  for(i=1;i<3;i++) line(x2+i,y1-i,x2+i,y2+i);
  for(i=1;i<3;i++) line(x1-i,y2+i,x2+i,y2+i);
}

```

```

strhz(char row,char col,char length,char *s,char attr)
{

```

```

  union REGS regs;
  int i; char row0,col0;
  for(i=0;i<length;i++)
  {
    row0=row; col0=col+i;
    row0=row0-1;col0=col0-1;
    regs.h.ah=2;
    regs.h.dh=row0;
    regs.h.dl=col0;
    regs.h.bh=0x00;
    int86(0x10,&regs,&regs);
    if(*s!=NULL)putat(*s++,attr);
    else putat(' ',attr);
  }
  return 1;
}

```

```

putat(char ch,char attr)
{

```

```

  union REGS regs;
  regs.h.ah=9;
  regs.h.al=ch;
  regs.h.bh=0;
  regs.h.bl=attr;
  regs.x.cx=1;
  int86(0x10,&regs,&regs);
}

```

```

char movs(int xx,int yy,char shw[300],int len2)
{

```

```

  char tms[300];
  char str[300];
  char df=0;
  int len1;
  len1=strlen(shw);
  while(! kbhit())

```

```

{
  strncpy(str,shw,(len1>len2?len2-1:len1-1));
  strhz(xx,yy,62,str,(LIGHTGRAY*16)|RED);

```

```

  if(! df)
  {
    strcpy(tms,shw);
    tms[1]='\0';
    strcat(shw,tms);
    strcpy(tms,&(shw[1]));
    strcpy(shw,tms);

```

```

    strcpy(tms,shw);
    tms[1]='\0';
    strcat(shw,tms);
    strcpy(tms,&(shw[1]));
    strcpy(shw,tms);
  }

```

```

  else
  {
    strcpy(tms,shw);
    strcat(tms,shw);
    tms[len1*2-1]='\0';
    strcpy(shw,&(tms[len1-1]));

```

```

    strcpy(tms,shw);
    strcat(tms,shw);
    tms[len1*2-1]='\0';
    strcpy(shw,&(tms[len1-1]));
  }

```

```

  while(j<=15){ getimage(64+j,432,559,449,buf);
    putimage(64,32,buf,NOT_PUT); j++; }
  j=0;
}
return(0);
}

```

```

/* 程序四 */
#include "stdio.h"
#include "graphics.h"
#include "stdlib.h"
#include "conio.h"
#include "bios.h"
#include "dos.h"
#include "string.h"

```

```

char movs(int xx,int yy,char shw[400],int len2);
void windows();
void *buf; int j=0;

```

```

main()

```

```

{
int gd=VGA,gm=VGAHI;
int size;
initgraph(&gd,&gm,"");
cleardevice();
setfillstyle(SOLID_FILL,CYAN);
bar(37,428,588,454); windows(37,428,588,454);

size=imagesize(64,432,559,449);
buf=malloc(size);
movs(25,9, "在实际应用中,显然不希望同时出现两
行相同内容的汉字,本程序使得储存和释
放汉字在同一行进行,请注意本程序和程
序三的不同和异同之处.",68*2);

free(buf);
closegraph();
exit(0);
}
/*..... windows(),strhz(),putat()等函数和程序三相
同 */

```

```

char movs(int xx,int yy,char shw[300],int len2)
{
char tms[300];
char str[300];
char dirflg=0;
int len1;
len1=strlen(shw);
while(! kbhit())
{
strncpy(str,shw,(len1>len2?len2-1:len1-1));
strhz(xx,yy,62,str,(LIGHTGRAY*16)|RED);

```

```

if(! dirflg)
{
strcpy(tms,shw);
tms[1]='\0';
strcat(shw,tms);
strcpy(tms,&(shw[1]));
strcpy(shw,tms);

strcpy(tms,shw);
tms[1]='\0';
strcat(shw,tms);
strcpy(tms,&(shw[1]));
strcpy(shw,tms);
}
else
{
strcpy(tms,shw);
strcat(tms,shw);

```

```

tms[len1*2-1]='\0';
strcpy(shw,&(tms[len1-1]));

strcpy(tms,shw);
strcat(tms,shw);
tms[len1*2-1]='\0';
strcpy(shw,&(tms[len1-1]));
}
while(j<=15) { getimage(65,432,559,449,buf);
putimage(64,432,buf,COPY_PUT); j++; }
j=0;
}
return(0);
}

```

参考文献:

- [1] 袁津生,《C 语言实用技巧》,北京师范大学出版社
1993 年 9 月
- [2] 庄粤盛,《TURBO C 绘图程序设计实例》,学苑出版社
1993 年 10 月 (本文收稿日期:1995.9.26)

欢迎订阅《环境技术》杂志

《环境技术》原名为《环境条件与试验》,自 1992 年第 1 期起更名。它创刊于 1983 年,是跨部的中级科技期刊,国内统一刊号:CN44—1325/X,国际标准连续出版物号:ISSN1004—7204,公开发行。广告经营许可证号:粤 010235。

本刊旨在报导国内外有关电工电子产品环境适应性、环境条件、环境试验、测试设备、环境防护、计算机应用、产品可靠性、IEC 动态、国外考察、企业管理等内容,以及国内外有关电工电子产品环境技术科研、试验、标准化、行业活动等。

本刊系双月刊,电脑激光编排,16 开本,每期约 8 万字。全年订价 21.00 元(含邮费)。欢迎单位和个人订阅。1996 年的征订工作现已开始,请订阅者与广州电器科学研究所本刊编辑部联系。订款可由银行信汇或邮局汇至广州电器科学研究所。

地址:广州市新港西路 204 号

银行帐号:中行新港西路办 256—517002—13

邮政编码:510302

★欢迎刊登广告·本刊竭诚为客户服务★

如何提高用户局域网的性能

山东莱芜钢铁总厂自动化部信息中心 孙迎春

摘 要:本文从可靠性和响应时间两个方面,介绍了如何提高用户局域网的性能,并引入了客户机、服务器以及族的概念

主题词:客户机 服务器(Client/Server)结构族(Cluster)

局域网技术的发展至今已有十几年的历史,目前这项技术已较为成熟。主要有以太网、令牌总线网、令牌环形网、光线数据分布接口局域网等。局域网提供了分布式信息处理应用的环境。衡量局域网的性能的主要指标是为网上用户提供服务质量,这种服务涉及两方面的内容:一是用户能够享受到的整个网络的平均响应时间,二是用户可得到的可靠、安全的服务。

1、响应时间是指网络用户向服务器提出请求到服务器开始响应用户请求的时间区间。影响响应时间的因素主要有三个方面:①源结点的处理;②传输介质的时间消耗;③服务器的性能指标。

当然网络的响应时间还与网络的拓扑结构,规模,冲突处理方式及重传输机制习习相关。但相比而言,如何选好整个系统的网络体系结构,对系统的响应时间的影响,最为重要。因此,本文着重探讨一下有关适应具体用户需求的分布式网络系统体系结构的性能和特点。

分布式网络系统提供了一种分布自治—合作方式的最佳途径,具有很大的灵活性。分布式系统和其他系统相比有以下特点。

①分布式系统具有很强的适应能力,可以满足用户的不同需求。

②分布式系统是并行处理的理想选择。

③分布式系统为各用户提供尽量均匀的分布,且系统具有很强的扩充能力。

近几年来,应用日益普遍的客户机/服务器结构,就是分布式处理的例子。客户机/服务器(client/server)结构是在网络基础上对硬件和软件进行合理配置,并使之协同工作的一种体系结构。用户在本机工作站上开发的应用产品,可以不加修改地运行于整个网络系统。对于包

含多种网络操作系统、多种数据源的复杂系统,用户的开发过程和单机一样简单。在客户机/服务器结构中,所有的用户处于相同的优先级,对网络上任一节点的服务可以通过多路径选择避免冲突,多服务器选择缩短响应时间。用户可以到网络系统中的任一服务器登录,所有数据的存取过程对开发者来说是透明的,使用户的开发工作更多地着重于最终的用户需求。

本文以 Decnet 为例,说明客户机/服务器结构的特点。网络组成如图 1。

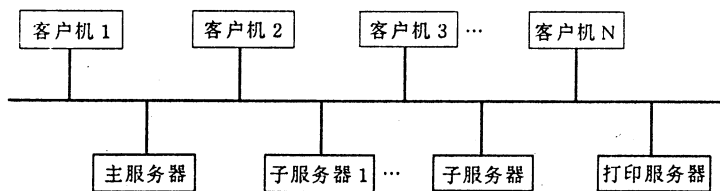


图1 网络组成

该系统有一台主服务器,若干台从服务器,一台打印服务器,若干台客户机。主服务器提供网络中任何地方的用户服务、网络及安全管理。从服务器为用户提供数据、通信方面的服务。打印服务器负责整个网络的打印工作,自动为用户的打印队列分配打印机,提供打印服务。网络中的客户机,不但分享系统资源,同时又可作为主机终端。客户机具有一定的存储空间,大量的数据操作有客户机完成,只有要到中心库中存取数据时,才和服务器建立逻辑连接。客户机可以到网络中的任何服务器登录,这样服务器可为各客户节点提供均匀分布的处理,减少了传输碰撞(Collision),缩短了传输部分的响应时间。这种结构对源节点的响应时间,传输部分的响应时间以及目的节点的响应时间都有所考虑,是缩短整个网络响应时间的最优方案。

建立好的客户机/服务器结构,用户要根据自己的业务需求和未来的发展趋势来选择产品。选择服务器要根据网络的终端数目、业务量、用户对响应时间的要求,考虑机器的处理能力、内存、主频等。客户机的选择尽量和服务器配套。同时选择高速的网络适配器、数据传输率高的传输介质,以提高网络的整体性能,使客户机/服务器体系结构的优势发挥的淋漓尽致。

2、提高网络的可靠性主要包括以下两点:

①传输路径的可靠性是指保证源节点发送的数据准确到达接收节点。对数据的冲突处理,数据的传输校验格式,各种网络操作系统都有相应的规定,且网络操作系统本身都有容错、保密、纠错功能,足以满足传输可靠性的要求。

②服务器的可靠性。服务器是网络的中心部件,它管理数据存储、共享、打印、通信、系统安全。因此网络可靠性的关键是服务器。如果服务器出现故障,整个网络系统将处于瘫痪状态。为了避免这类问题的发生,对于可靠性要求不太高的场合可考虑双机备份或双硬盘结构服务器。但对于可靠性要求高的场合如银行、股票交易所等可以考虑双处理机系统。连接如图2。

两台 Dec-7710 通过计算机连接(CI-Computer interconnect)形成一个簇(cluster)结构,作为主服务器,对用户而言,这种组合是一台大硬盘的服务器。两台处理机没有主次之分,其中一台出现故障,另一台仍可继续工作,完成整个服务器的任务,用户感觉不到这种变化。

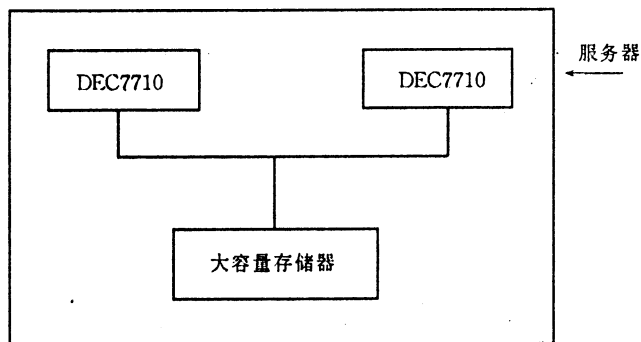


图2 双处理机服务器连接图

CI是一种连接处理机和智能终端的高速的、全双工的总线。簇(cluster)结构是Dec公司的一种

新技术,这种技术的在可靠性方面的优越性,已为许多计算机同仁接受。

结论:随着网络技术的进一步发展,人们对提高用户网络性能的研究越来越深入,高可靠性、短响应时间的要求一定能实现。但高可靠性、短的响应时间是以高性能价格比为代价的,这一点提醒广大用户注意。

(收稿日期:1995.9)

首届四川省十佳科技期刊 连续三届荣获优秀科技期刊一等奖 欢迎订阅《电子天府》(双月刊)

时代性·资料性·专题性·系统性·实用性

地址:①60017/成都市育婴堂街20号

②610017/成都市桂王桥西街66号

电话:(028)5560990、6750444

●是电子行业中唯一由26个省、区、市电子学会联合主办(编辑)的一本实用性的电子技术刊物。

●融生动、求实、启发于一炉,集“时代性”、“资料性”、“专题性”、“系统性”、“实用性”于一册。

●企业家开发产品的参谋 技术员更新知识的教材 爱好者拓宽视野的窗口 维修者排除疑难的助手

●刊物特点:“时代性”反映了内容的先进性、“专题性”满足了读者的求知欲、“实用性”、“系统性”和“资料性”增强了刊物的实用价值。

●栏目精彩:消费指南、新技术园地、小特集、技术讲座、专家新论、电路图说、电路详解、电路分析、机型剖析、难题图纸、维修精华、维修园地、专家会诊、IC数据、功能开发、检修技巧、维修实例精选、维修指南、速修卡片、彩电新电路、最新IC应用、资料邮购服务、报刊题录、电子新书邮购等等。

●'96年每期定价:7.00元,邮购价:8.05元。全年一次订阅,免收邮费。一次汇款,全年可得:42.00元/年,欢迎来函索取订单。

●合订本邮购价:'95年(49.50元)、'94年(41.40元)、'93年(32.20元)、'92年(28.70元)、'91年(28.70元)。

●本刊发行部长期经营销售全国各家电子、科技报刊及计算机、实用电子技术图书,品种达千种以上,欢迎函索详细书目并惠顾。

●邮局汇款:610017/四川省成都市育婴堂街20号电子天府发行部 银行汇款:开户行:工商银行成都红星中路分理处 帐号:25908902294 户名:电子天府编辑部

SSU—12 数字程控用户交换机计费系统的二次开发

山东省寿光市化工总厂微机室[262700] 郑翔东

一. 前言

上海贝尔电话设备制造有限公司生产的 SSU—12 型数字程控用户交换机, (以下简称 SSU12) 以其先进的技术、灵活的配置、方便的维护, 在国内获得了广泛的应用, 其计费系统是通过连接在 SSU12 的一个串行口的 CP80 串行打印机来完成的, 它实时记录呼叫公共电话网的分机的号码、公共电话网的被叫用户号码, 进行呼叫的日期及时间、通话占用的时间, 这些信息的统计、汇总、查询必须依靠用户人工完成, 给用户对内部分机的管理工作带来了很大的不便。

有鉴于此, 我们在 SSU12 所配置的简单的计费系统基础上, 将原输出到串行打印机的有关计费信息转储到一台 PC 机中, 利用 BASIC 语言自行开发了一套明细计费系统软件包来完成计费信息的汇总、统计以及多种形式的查询功能, 为用户对电话费用的管理工作提供极大的方便。

二. 系统二次开发的设计思想

SSU12 的通用同异步接收发送器 (USART) 8251A 输出的数据经 PC 机的 USART 接受后, 以字节为单位存放在 dos 通讯缓冲区中, PC 机的 USART 与通讯缓冲区之间的数据传输是由 DOS 和 BASIC 系统完成的, 通讯缓冲区的大小可以由用户根据数据传输数率来指定。由于采用了 BASIC 的通信陷阱技术, 该缓冲区采用缺省的 256K 已经足够了。

在 BASIC 系统中, 异步通讯适配器是被当作一个设备文件来使用的。其使用前必须利用下列语句打开:

```
OPEN "COM n:[波特率] [, 效验] [, 数据位]
    [, 停位] [, RS] [, CS[n]] [,
    DS[n] [, CD[n]] [, LF] [,
    PE]" AS# <文件号> [LEN
    =<记录长>]
```

本文使用三线多环回型结构的串行接口 RS—232—C 电缆, 因而可以简单的使用下列语句打开通讯

口:

OPEN "COM n:波特率, 效验位, 数据位, 停止位" AS# 文件号具体原理见硬件配置部分。

为触发该适配器, 必须先执行语句: COM(n) ON.

ON COM(n) GOSUB line 语句对字符进入通讯缓冲区时产生陷阱, 其中 line 为通信陷阱程序的入口行号。

n 为异步通讯适配器的编号。

通信适配器经以上语句的初始化之后, 已经被触发并建立通讯陷阱。则其后每执行一条语句之前, BASIC 都要检查是否有字符进入指定的通信适配器,

若有, 则将转而执行通信陷阱程序。当然, BASIC 不执行程序时, 陷阱是不会发生的。

执行 COM(n) STOP 语句适配器将不发生陷阱, 但记住被接收的任何字符, 以便当适配器重新打开时立即发生陷阱。

对进入通讯缓冲区的数据, 在通讯陷阱程序中利用语句 INPUT 接收。

在该系统中, 一共建立了 13 个数据库文件, 其中 12 个作为备份库文件, 分别保存一年中 12 个月的数据, 另一个作为实时库文件, 实时接受进入通讯缓冲区的数据。

备份库文件用于完成用户对数据的汇总、统计、查询的需要, 在用户进行这些操作时, 不影响实时库文件对数据的接收。系统根据机器时钟, 在每月初以顺序输入方式打开当前月的备份库文件, 删除旧数据, 使系统保持 12 个月的数据量, 当然, 用户亦可根据需要和 PC 机的容量大小来确定系统需保持的数据量。

实时库文件除用于接收数据外, 还便于用户查询当天的数据, 满足个别用户的特殊需要。在每日凌晨, 系统将自动完成实时库文件到备份库文件的数据传输, 并更新实时库文件。

三. 系统硬件的重新配置

SSU12 带有四个串行通讯(SIO)接口,采用 D 形 25 引脚连接器,原 CP80 打印机通过串行通讯电缆与其中的一个连接。

在系统的新配置中,我们将 PC 机作为数据终端设备(DTE),SSU12 作为数据通讯设备(DCE),更改原通讯电缆的物理接口方式,采用了三线多环回型结构,具体引脚分配见下表。

2	2
3	3
4	4
5	5
6	6
7	7
8	8
.	.
20	20
.	.

上图中未列出的引脚悬空,下面是所使用的引脚的说明

引脚	说 明
2	发送数据
3	接收数据
4	请求发送
5	允许发送
6	数据装置就绪
7	信号地
8	接收线路信号检测器
20	数据终端就绪

该联结方式的原理是:通过把数据终端就绪跨接到数据装置就绪,一旦 DTE 建立其数据终端就绪线,便完成数据就绪阶段。在 DTE 建立请求发送时,由于请求发送直接跨接到接收线信号检测器,由于请求发送跨接到允许发送,故请求发送所包含的信道准备就绪工作完成了,随后即可进行数据的发送了。

为节约起见,未另行购置汉字打印机,而是利用原来配置的 CP80 串行打印机,通过三线经济型结构的通讯电缆与 PC 机的一个未用的串行接口连接,完成各类数据的报表输出。具体连接方式见图 1、图 2。

四. 程序说明

系统软件包主要包括以下几个主要模块。

1. 主模块

主要完成通讯端口的设置,数据文件的打开方

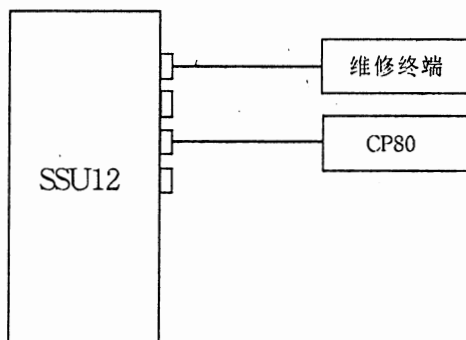


图 1 原 SSU12 与 CP80 的连接

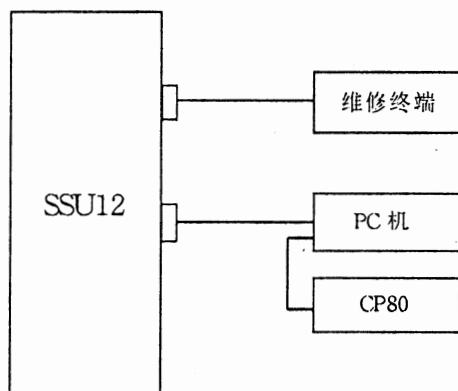


图 2 现 SSU12 与 PC 机及 CP80 的连接

式,通讯陷阱和错误陷阱的启动,以及系统参数和变量的设置等一系列初始化工作。另外,系统功能主菜单及动态时钟的演示及日期的刷新功能亦在该模块中完成。其简要流程图见图 3。

2. 通讯陷阱处理模块

接收通讯缓冲区中以字节为单位的数据,经分段整理形成呼叫日期及时间,分机号码,被叫用户号码,通话时间等五个字段,并写入实时库文件中。

3. 汇总统计处理模块

根据用户的要求,对一个时间段的数据进行统计,汇总报表输出。报表分两种格式:

(1)以一个时间段为关键字。对该时间段内的数据汇总,详细分列出各分机对公共电话网呼叫的类型(如市话、农话、国内长途、国际长途等)及次数,单位时间(一般以每分钟为一计时单位,市话以三分钟为一计时单位)的单价,以及该时间段内的总费用。此表便于用户对各分机的管理。

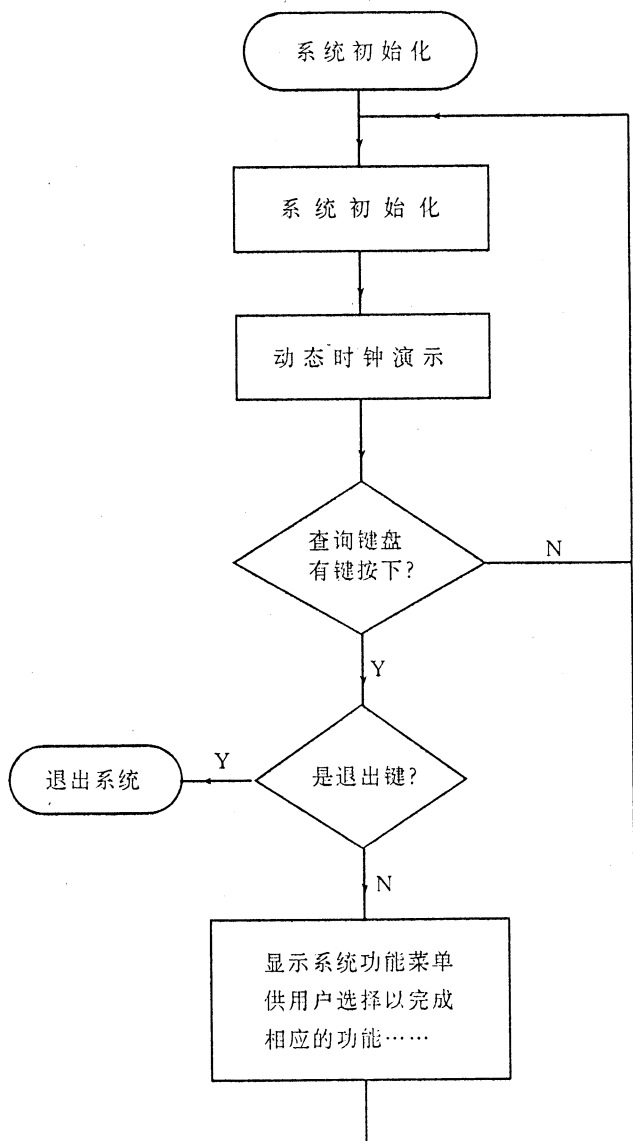


图3

(2)以分机号码为关键字。集中输出一个时间段(一般以月为单位)内各分机的费用。此表用于用户收取电话费。同时,亦可根据用户要求输出各分机的收费单。

4. 信息查询处理模块

该模块有四种数据的查询方式

(1)以一时间段为关键字,详细列出用户要求的分机每次对公共电话网的呼叫类型,呼叫时间,单位时间单价以及该次呼叫的费用,以便于分机用户对汇总费用提出疑问时进行查询。

(2)以呼叫类型为关键字,详细分列一时间段内进行过该类型呼叫的分机的详细数据。

(3)以用户提供的被叫用户号码为关键字,详

细分列出一时间段内分机对该用户号码进行呼叫的日期、时间、通话时间及费用。

(4)以用户提供的一组数据为关键字进行模糊查询。该组数据可以是分机号码、被叫用户号码、时间、日期、通话时间的有机组合,以方便用户的特殊需要。但查询速度比较慢。

5. 备份处理模块

该模块完成实时库文件到备份库文件的数据备份。在该模块开始时,先使用COM(n) STOP 暂停通讯陷阱但记忆其后发生的通讯活动,在数据备份完毕后利用语句COM(n) ON 立即发生陷阱。在完成数据的备份后,实时库文件中的数据将被删除,以便接收新数据。

6. 错误处理模块

利用语句 ON ERROR 启动错误陷阱后,系统发生任何错误将导致系统转入该模块进行处理。进入该模块的处理,系统根据用户的要求将:

(1)排除系统的错误,(如打印机电源未打开等),继续完成功能操作。

(2)关闭系统初始化时打开的系统设备和文件,退出系统。

7. 通讯数据处理模块

由于目前国内通讯事业的发展较快,各地区使用的区号变化频繁,因此专门建立该模块,以便于用户对国内各地区的区号及费率进行添加、修改和删除,保证对分机的收费符合国家标准。

8. 即时查询处理模块

该模块的处理是针对实时库文件的操作,查询分机当天的数据,立即结算电话费并打印出收费单,以方便用户对外来人员征收电话费。

9. 其它处理模块

系统还增加了一些动画及后台音乐伴奏、警笛等小模块,以方便用户,增加系统界面的友好性。

五. 结束语

本系统运行于MSDOS3.0以上的系统中,选择ccdos及ucdos两种广泛流行的汉字系统作为中文平台,具有较好的兼容性。同时要求使用286以上的PC兼容机。系统经过两年多的实际运行,工作稳定,使用效果良好,数据安全性较高,未发生数据丢失的现象。另外,由于PC机与SSU12之间采用单信道的串行通讯,对SSU12的正常运行不会产生任何不良影响,而且也避免了SSU12由于CP80打印机故障而引发的报警。同时,系统具有较强的通用性,稍加修改即可适用于其他类型的交换机。有兴趣的同行请与我们联系。

(本文收稿日期:1995.10.2)

NOVELL 远程自举 工作站上制作系统盘一法

广东梅州嘉应大学计算中心 饶拱维

我们知道,使用 FORMAT/S 格式化磁盘时,FORMAT 命令会把系统文件从启动盘调出并写入格式化盘。对 NOVELL 远程自举工作站而言,自举是利用服务器硬盘上的远程自举映像文件完成的,因此,在这类工作站执行带参数/S 的 FORMAT 命令时,会首先提示用户在 A 驱动器中插入系统盘,以便从 A 驱动器的系统盘中取出系统文件复入新盘。另外,在网络环境下,不借助于系统盘,也无法用 SYS 命令传送系统文件,因为 SYS 命令不能用于网络盘。这样一来,为了做张启动盘,就必须借助于现有系统盘才能完成,这给用户带来不便。目前,许多学校的微机室都已安装了 NOVELL 局域网,其工作站大多采用无盘工作站(一般都保留软驱,供学生练习使用)。因为 FORMAT 及 SYS 命令的不足,给学生上机带来麻烦。那么,能否借助于网络上的 MS-DOS 系统文件制作系统盘呢?答案是肯定的,实现方法是使用一个自建的 NWFORMAT.BAT 文件。具体方法介绍如下。

一、准备工作

为了能使用 NWFORMAT.BAT 文件,必须完成以下工作:

1、在网络上的 DOS 目录下建立 NWFORMAT.BAT 批文件,文件内容为:

```
@echo off
format %1/b%2
attrib %1\*. * -h -s
copy f:sysfile %1>nul
attrib %1\*.sys +h +s +r
echo System transferred.
echo on
```

2、在 DOS 目录中建立 SYSFILE 子目录,把系统文件 IO.SYS、MSDOS.SYS 及 COMMAND.COM 拷入 SYSFILE 子目录。拷入之前应先去掉 IO.SYS、MSDOS.SYS 的文件的隐含属性和系统属性(可使用 ATTRIB 命令)。

3、若网络上含有多种版本的 DOS,为了能在

不同版本的 DOS 下使用 NWFORMAT 命令,应在各版本的 DOS 目录下建立 NWFORMAT.BAT 文件及相应的 SYSFILE 子目录。

二、使用 NWFORMAT.BAT

文件的格式及说明

1、使用格式

```
NWFORMAT <盘符> [/4][/V[:label]]
[/U][/Q][/F:size]
```

2、说明

(1)在 NWFORMAT 命令的格式中,<盘符>后至少应留一个空格,<盘符>后的各参数之间则不能含有空格。

(2)使用 NWFORMAT.BAT 文件时,请进入当前版本的 DOS 子目录。

(3)执行 NWFORMAT.BAT 时,回答"Format another (Y/N)?"提示应键入 N,不要键入 Y;若要再制作另一张系统盘,则须重新执行 NWFORMAT 命令。

(4)屏幕显出"System transferred."信息时,表示系统文件传送成功,格式化完毕。

本法已在 NETWARE 3.11 及 MS-DOS5.0、6.20、6.21 上通过。(本文收稿日期:1995.10.4)

欢迎订阅《电机电器技术》

《电机电器技术》杂志是机械工业部广州电器科学研究所主办的科技期刊,主要介绍有关日用电器、中小型电机、特殊电机、自动控制装置、计算机检测装置、电工测量仪器、晶闸管元件、绝缘材料及金防技术装置等方面的科研成果及国内外动态。

每期定价 3.50 元,全年订价 14 元(含邮资),请与该刊编辑部联系订阅。地址:(510302)广州市新港西路 204 号广州电器科学研究所,银行帐号:广州中行新港西路办 256—517002—13。

通用开放式下拉菜单自动生成器

山东曲阜师范大学物理系[273165] 狄振强 孟广盈

摘 要 本文介绍一种通用、开放式的下拉菜单自动生成系统,它通过对话框的形式编辑菜单内容,并可对菜单进行纵横缩、放的控制。生成的菜单系统,对用户是开放的,即可随时再对菜单进行编辑修改。

关键词 下拉菜单 菜单项 内存变量 菜单信息 功能键

一、引 言

下拉菜单是目前数据库管理软件中普遍使用的菜单驱动形式,由于 FOXBASE 系统提供的命令 MENU BAR 生成的下拉菜单,在汉字状态下屏幕显示效果不理想,程序设计者往往自己为软件编制下拉菜单。令程序设计者费事的是要产生一个菜单就得编一个程序,在调试过程中要修改菜单又得修改程序,这些工作,耗费了编程人员大量的时间和精力。

为了克服这一不足,笔者设计了一个通用的、开方式下拉菜单自动生成器。通过交互方式,用户直接在屏幕上编辑、修改菜单,使原来由编程才能完成的任务变成了轻松的敲键操作。它提供给用户的不是一个一成不变的菜单环境,而是允许用户随时进行编辑修改,对于用户是一个开放的菜单生成、编辑工具。

二、功能介绍

本程序有两大功能:一是菜单编辑功能,以对话框方式接受用户输入的有关信息,并将定义信息存入一内存变量文件中。二是菜单驱动功能,从生成的内存文件中提取菜单定义信息,把下拉菜单调出,并接收菜单选项的键盘控制,它为用户提供一个菜单使用环境。系统可在“菜单编辑状态”和“菜单驱动状态”之间切换。

三、控制功能键

1、菜单编辑功能键

当初次启动系统时,首先进入菜单编辑状态。在此状态下,能够定义和修改系统(主标题)名称、主(子)菜单项的数目、名称和菜单的数目、宽度、名称。有关菜单的参数自动存放在一内存变量文件中。

菜单的底行是各功能键的操作提示信息,功能如下。

F2——标题,它定义系统(主标题)名称、设置主菜单项的数目、名称。

F3——宽度,用于设置子菜单项的宽度,可通过光标左右移动键任意调整。

F4——长短,用于确定子菜单项的数目,通过光标上下移动键任意调整。

F5——光带,用于输入子菜单项名称。

F6——定义,定义菜单选择项调用执行的模块名程。

另外,当整个菜单编辑完成后,按 CTRL + END 可退出“菜单编辑功能”,转换至“菜单驱动功能”,并保存用户定义的信息。

2、菜单驱动功能键

在系统启动时,若已存在用户定义的菜单信息(内存变量)文件,则进入菜单驱动状态,即用户使用的菜单环境。也可由“菜单编辑状态”转换至该状态。

在该状态下可通过四个光标移动键进行菜单项的选择,按 ENTER 键确定,并调用执行相应的功能模块。

任何时候系统均可通过 CTRL + B 键切换到“菜单编辑状态”,以便对当前菜单重新进行编辑修改。

四、程 序

完成上述功能的 FOXBASE 程序如下。(限于篇幅,省去了封面设计、立体效果修饰及二级菜单等的处理部分。)在 FOXBASE + 2.0、2.1 版本上通过。

```
SET PROC TO TYXLCD
SET TALK OFF
SET ESCAPE OFF
SET SCOR OFF
ON ERROR DO ERR
```

```

IF FILE('XCD.MEM')
  RESTORE FROM XCD
  MTG=. F.
  CS=2
ELSE
  CS=1
  DIMENSION CDM(7),ZCD(7,16),XMS(8),XMW
(7),cdw(7),CDN(7),CK(7)
  MTG=. T.
  ZHUB=SPACE(24)
  ZHUBT='['+ ZHUB+']'
  ZHCDS=7
  ZCDS=8
  NUM=SPACE(2)
  SPCDW=INT(76/ZHCDS)
  DO CDCSH
ENDIF
DO QCZH
ac=1
ha=1
do while .t.
  HA=1
  DO ZCDXS
IF MTG
  do while .t.
    SET COLO TO W+/B
    @ 24,16 SAY "操作提示:F2 标题  F3 宽窄  F4 长短
      F5 光带  F6 定义"
    CLEAR TYPEaHEAD
    Pk=inkey(0)
    do case
      CASE pk=-4      &&. 光带修改过程
      DO WHILE HA<=XMS(AC)
      SET COLO TO G+/3+,gr+/RB+
      @ 3+HA,5+(AC-1)*SPCDW-CK(AC) GET
ZCD(AC,HA)
      READ
      if CS=2 . AND. ( readkey()=15. or. readkeY()=
271). OR. HA=XMS(AC)
        exit
      ELSE
      HA=HA+1
      ENDIF
    ENDDO
    CASE pK=-3
    DO WHILE .T.
    OK=INKEY(0)
    DO CASE
    CASE OK=24
    IF XMS(AC)>=16
      loop
    ENDIF
    XMS(AC)=XMS(AC)+1

```

```

  PHA=XMS(AC)
  SET COLO TO 0/0
  @ 4+XMS(AC),4+(AC-1)*SPCDW-CK(AC)
  CLEAR TO 5+XMS(AC),7+(AC-1)*SPCDW+
  XMW(AC)-CK(AC)
  SET COLO TO W/6
  @3+XMS(AC),3+(AC-1)*SPCDW-CK(AC)
  CLEAR TO 3+XMS(AC)+1,6+SPCDW*(AC-1)+
  XMW(AC)-CK(AC)
  zcd(ac,pha)=space(len(zcd(ac,1)))
  set colo to w+/3+
  @ 3+XMS(AC),4+(ac-1)*SPCDW-CK(AC)
  SAY '['+ZCD(AC,PHA)+']'
  CASE OK=5
  if xms(ac)=1
    loop
  endif
  XMS(AC)=XMS(AC)-1
  PHA=XMS(AC)
  SET COLO TO /W
  @ 6+XMS(AC),4+(AC-1)*SPCDW-CK(AC)
  CLEAR TO 6+XMS(AC),7+(AC-1)*SPCDW-CK
  (AC)+XMW(AC)
  @ 5+XMS(AC),2+(AC-1)*SPCDW-CK(AC)
  CLEAR TO 5+XMS(AC),4+(AC-1)*SPCDW-CK
  (AC)+XMW(AC)
  SET COLO TO 0/0
  @ 5+XMS(AC),4+(AC-1)*SPCDW-CK(AC)
  CLEAR TO 5+XMS(AC),7+(AC-1)*SPCDW-CK
  (AC)+XMW(AC)
  SET COLO TO W/6
  @ 4+XMS(AC),4+(AC-1)*SPCDW-CK(AC)
  CLEAR TO 4+XMS(AC),6+SPCDW*(AC-1)+
  XMW(AC)-CK(AC)
  CASE OK=13
  DO ZCDXS
  EXIT
  ENDCASE
  ENDDO
CASE pK=-2
DO WHILE .T.
  ROK=INKEY(0)
DO CASE
CASE ROK=4
IF 7+(AC-1)*SPCDW-CK(AC)+XMW(AC)>77
  LOOP
ENDIF
XMW(AC)=XMW(AC)+1
SET COLO TO 0/0
@ 4,6+(AC-1)*SPCDW-CK(AC)+xmw(ac)
CLEAR TO 5+XMS(AC),7+(AC-1)*SPCDW-CK
(AC)+XMW(AC)
SET COLO TO W/6

```

```

@3,5+(AC-1)*SPCDW-CK(AC)+xmw(ac)
CLEAR TO 3+XMS(AC)+1,6+SPCDW*(AC-1)+
XMW(AC)-CK(AC)
EWQ=.T.
DO GDXG WITH EWQ
CASE ROK=19
If xmw(ac)=10
    loop
endif
SET COLO TO 6/w
@ 3,6+(AC-1)*SPCDW-CK(AC)+XMW(AC)
CLEAR TO 5+XMS(AC),7+(AC-1)*SPCDW-CK
(AC)+XMW(AC)
XMW(AC)=XMW(AC)-1
SET COLO TO 0/0
@ 4,7+(AC-1)*SPCDW-CK(AC)+xmw(ac)
CLEAR TO 5+XMS(AC),7+(AC-1)*SPCDW-CK
(AC)+XMW(AC)
SET COLO TO W/6
@3,6+(AC-1)*SPCDW-CK(AC)+xmw(ac)
CLEAR TO 3+XMS(AC)+1,6+SPCDW*(AC-1)+
XMW(AC)-CK(AC)
EWQ=.F.
DO GDXG WITH EWQ
CASE ROK=13
DO ZCDXS
    EXIT
ENDCASE
ENDDO
CASE pK=-1
zhub=space(18)
set colo to /w
    @ 3,2 clear to 21,78
    set colo to g+/3+,n/5
    @ 4,15 clear to 15,55
    @ 4,20 say '系统名称'
    @ 4,30 GET ZHUB
        read
        zhub=trim(zhub)
        zhubb='【'+zhub+'】'
@ 5,20 say '主菜单数:'GET ZHCDS RANGE 2,7
    READ
SPCDW=INT(76/ZHCDS)
    ew=1
DO WHILE ew<=ZHCDS
@ 5+EW,15 SAY CDN(ew) GET CDM(ew)
    READ
    CDM(ew)=TRIM(CDM(ew))
    ew=ew+1
ENDDO
AC=1
do qczh
do zcdxs

```

```

case Pk=-5
NUM=LTRIM(STR(AC))+LTRIM(STR(HA))
    PG&NUM=SPACE(10)
SET COLO TO W+/B+,W+/B+
@ 19,30 SAY '功能模块:' GET PG&NUM
    READ
SET COLO TO N+/W+
@ 19,25 clear to 19,50
DO ZCDXS
CASE PK=23
    MTG=.F.
    DO QCZH
    DO ZCDXS
    SAVE TO XCD
    EXIT
    CASE pK=5    &&↑
DO ZCXG1
    HA=IIF(HA=1,XMS(AC),Ha-1)
    DO ZCXG2
    CASE pK=24    &&↓
DO ZCXG1
    HA=IIF(HA=XMS(AC),1,HA+1)
DO ZCXG2
case pk=19    &&←
    DO CDXG
    ac=iif(ac=1,ZHCDS,ac-1)
    DO ZCDXS
case pK=4    &&→
DO CDXG
    AC=iif(ac=ZHCDS,1,ac+1)
DO ZCDXS
endcase
enddo
ENDIF
do while .t.
SET COLO TO W/B+
@ 24,16 say '操作提示:→←↑↓——选择菜单 Enter——
—选中 Esc——退出'
CLEAR TYPEAHEAD
key=inkey(0)
DO CASE
case key=2
    MTG=.T.
    EXIT
case key=27
    SET PROC TO
    return
CASE KEY=5    &&↑
    DO ZCXG1
    HA=IIF(HA=1,XMS(AC),Ha-1)
    DO ZCXG2
CASE KEY=24    &&↓
    DO ZCXG1

```



```

    HA=IIF(HA=XMS(AC),1,HA+1)
    DO ZCXG2
case key=19      &&←
    DO CDXG
    ac=iif(ac=1,ZHCDS,ac-1)
    exit
case KEY=4      &&→
    DO CDXG
    ac=iif(ac=ZHCDS,1,ac+1)
    exit
case key=13
    NUM=LTRIM(STR(AC))+LTRIM(STR(HA))
    do &pg&.num
    do qczh
    do zcdxs
ENDCASE
ENDDO
enddo
PROC ZCDXS
SET COLO TO 7/N
@ 1,2+(AC-1)*SPCDW sAY CDM(AC)
IF 7+(AC-1)*SPCDW+XMW(AC)>78 . AND. 7+
(AC-1)*SPCDW+XMW(AC)>CK(AC)+78
    CK(AC)=7+(AC-1)*SPCDW+XMW(AC)-78
ENDIF
SET COLO TO 0/0
@ 4,4+(AC-1)*SPCDW-CK(AC) CLEAR TO 5+
XMS(AC),7+(AC-1)*SPCDW+XMW(AC)-CK
(AC)
SET COLO TO W/6
@3,3+(AC-1)*SPCDW-CK(AC) CLEAR TO 3+
XMS(AC)+1,6+SPCDW*(AC-1)+XMW(AC)-CK
(AC)
    I=1
    DO WHILE I<=XMS(AC)
    set colo to W+/3+
    @ 3+I,4+(ac-1)*SPCDW-CK(AC) SAY '['+
ZCD(AC,I)+'']'
    I=I+1
    ENDDO
    ha=1
SET COLO TO W+/B+
@ 3+ha,4+(ac-1)*SPCDW-CK(AC) SAY '['+ZCD
(AC,ha)+'']'
SET COLO TO W/6,N/W
PROC ZCXG1
set colo to 7/3+
@ 3+ha,4+(ac-1)*SPCDW-CK(AC) clear to 3+ha,
4+(ac-1)*SPCDW+XMW(AC)-1-CK(AC)
@ 3+ha,4+(ac-1)*SPCDW-CK(AC) SAY '['+ZCD
(AC,ha)+'']'
PROC ZCXG2
set colo to W+/B+

```

```

@ 3+HA,(AC-1)*SPCDW+4-CK(AC) SAY '['+
ZCD(AC,HA)+'']'
PROC CDXG
SET COLO TO 5/3
@ 1,2+(AC-1)*SPCDW SAY CDM(AC)
SET COLO TO n/w+
@3,3+(AC-1)*SPCDW-CK(AC) CLEAR TO 3+
XMS(AC)+2,7+SPCDW*(AC-1)+XMW(AC)-CK
(AC)
proc qczh
SET COLO TO 2+/6
@ 0,0 CLEAR TO 24,79
@ 0,25 SAY ZHUBT
    PR=1
set colo to 5/3
@ 1,1 clear to 1,78
DO WHILE PR<=ZHCDS
    @ 1,2+(PR-1)*SPCDW sAY CDM(PR)
    PR=PR+1
ENDDO
    set colo to 7/7
    @ 2,1 clear to 21,78
proc cdcsh
zh=1
do while zh<=ZHCDS
    CDM(ZH)='【'+ltrim('菜单')+'0'+LTRIM(STR
(ZH))+ltrim('】')
    CDN(ZH)=CDM(ZH)
    xmw(ZH)=10
    xms(ZH)=ZCDS
    CK(ZH)=00
    zc=1
    do while zc<=16
        ZCD(zh,zc)=space(xmw(zh))
        zc=zc+1
    enddo
    zh=zh+1
enddo
PROC GDYG
PARa EWQ
EQ=1
DO WHILE EQ<=XMS(AC)
    IF EWQ
        ZCD(AC,EQ)=ZCD(AC,EQ)+SPACE(1)
    ELSE
        ZCD(AC,EQ)=LEFT(ZCD(AC,EQ),LEN(zcd(ac,
eq))-1)
    ENDIF
    EQ=EQ+1
ENDDO
PROC ERR
@ 18,3 say '未定义此功能模块'
    WAIT

```

(本文收稿日期:1995.9.24)

用 FOXPRO 制作信息活动窗口

河北人民警察学校 刘全胜

近年来,FOXPRO 语言正在我国迅速流行起来,已有逐步替代 DBASE Ⅲ、FOXBASE 的趋势。它易学易用,功能强大,我在实践中用 FOXPRO2.5 编制过一种信息活动窗口,将它用于一些软件中,效果不错,所以深深体会到 FOXPRO 在制作各种窗口中的优越性,现将它介绍一下。

此窗口形式是这样的,它是一个宽一行长二十列的带状窗口,首先出现在屏幕最右边的中部,然后自动向左移动,同时窗口中显示信息,信息是先显示一个窗口长的内容,然后信息逐渐向左卷动,以便显示出剩余内容,窗口到达屏幕中间时会停下来,在完整地显示一遍信息内容后,再继续向左移动,继续卷动式显示信息,到达屏幕最左边时窗口自动消失,在移动过程中因为是以窗口形式出现,不会将屏幕上背景内容抹掉,所以很安全。此窗口的大小,出现的位置可以随意改变,但在它们改变的同时,与它们相关的控制到达位置的变量,也应作相应的改变。以下是此程序内容:

```
* 信息活动窗口: FLY()
* 调用形式: ? FLY("信息内容")
PARAMETERS MESSAGE
PRIVATE START,XTOP,YTOP,XSIZE,YSIZ,YY
START=1      && 信息字符串显示时的起点
YY=0        && 控制窗口到达位置的变量
XTOP=8      && 窗口左上角的行坐标
YTOP=78     && 窗口左上角的列坐标
XSIZE=1     && 窗口行方向上的大小
YSIZE=IIF(LEN(MESSAGE)%2=1,21,20)
            && 窗口列方向上的大小要根据信息长
            && 度来定,以免显示时出现半个汉字
DEFINE WINDOW FLYWIN AT XTOP, YTOP SIZE
XSIZE, YSIZE COLOR SCHEME 4 SHADOW
SET TALK OFF
SET STATUS OFF
SET CURSOR OFF
ACTIVATE WINDOW FLYWIN
DO WHILE YY < 80
DO WHILE START < LEN(MESSAGE)
```

```
* 如果窗口未处于屏幕中间,则移动窗口
IF YY<>50
MOVE WINDOW FLYWIN BY 0,-2
YY=YY+2
ENDIF
A=SUBSTR(MESSAGE,START)+SPACE(2)
    && 截取信息时尾部加上两个空格,以免
    && 显示汉字时拖尾
SET COLOR TO+GR/W
@ 0,0 SAY A
=INKEY(0.2)
START=IIF(YY<=YSIZE,START,START+2)
ENDDO

START=1
* 如果窗口到达屏幕中间,则停下来完整显示一遍信息
IF YY=50
DO WHILE START < LEN(MESSAGE)
A=SUBSTR(MESSAGE,START)+SPACE(2)
@ 0,0 SAY A
=INKEY(0.2)
START=IIF(YY<YSIZE,START,START+2)
ENDDO
YY=YY+2
ENDIF

ENDDO
DEACTIVATE WINDOW FLYWIN
RELEASE WINDOW FLYWIN
SET CURSOR ON
RETURN
```

此窗口仅仅是一个最基本的形式,我们还可以将它改为活动时带铃声,多个位置出现,多个方向移动,窗口多次激活,窗口消失可控等多种形式,而且它不仅可以作为传达信息的窗口,还可以改制成有趣的软件封面,很有意思,希望以此与各位同行交流经验,敬请指正。

(收稿:1995.7.14,修回 9.8.)

在 CLIPPER 中的图形界面设计

东莞理工学校 周昭权

提要:本文分析 CLIPPER 同 C 语言的接口,并用 C 语言实现图形用户界面的设计。

用 CLIPPER 编写的程序可以编译成 EXE 文件,其代码效率比较高,且文件执行速度也不错,因此很多 MIS 开发者都乐意采用它。但它提供的用户界面是基于字符方式,在图形界面流行的今天就显得比较单薄。

但在中文状态下,我们就可以改变这种状况,由于中文系统中文系统下实际上是工作在图形方式之下,一般是采用 16 色 640×480 图形模式,横向 640 点分成 80 列(每列 8 点),纵向 480 点 25 行(每行 18 点,其它 30 点作提示行)。因此,我们在程序中加上图形界面,如图形边框,图形窗口,按钮等。但是,CLIPPER 本身并不支持图形操作。

实际上,CLIPPER 虽不能直接进行图形操作,但它允许用其它如 C 语言等来编写函数,并提供了一组扩展函数来完成参数传递。这样,我们可以用 C 等其它语言编写图形操作函数。实际上只要有画横竖线的函数,我们就可以很简单地实现如图形边框,图形窗口,按钮等图形界面。因此我们先来用 C 来解决画线的问题。画线等最好要通过直接写屏来完成。

当然,用 C 语言编写 CLIPPER 的用户自定义函数是有一定的格式和要求。

主要有以下几点:

- 1、C 语言函数应声明为 CLIPPER 类型。
- 2、函数参数传递和返回参数需用 CLIPPER 的扩增函数。
- 3、一定要在 C 函数中调用 CLIPPER 的返回扩增函数,若没有返回值,则可调用 _ret() 返回空值。
- 4、应把 extend.h 包括在 C 函数中。

笔者在工作中,编写了几个简单的函数,用起来较方便,可以达到一些成功软件的界面效果。

```
/* mline.c */
#include "extend.h"
#include <dos.h>
```

• 34 •

```
#include <mem.h>
CLIPPER mline() /* 画横线 */
{
    char far * buf;
    char far * buf1;
    char buff, buff1;
    int yy, m, x, y, len, mm, mode, color;
    char poin[] = {0xff, 0x7f, 0x3f, 0x1f, 0x0f, 0x07, 0x03,
        0x01, 0x00};
    x = _parni(1);
    y = _parni(2); // 起点坐标
    len = _parni(3); // 长度
    mode = _parni(4); // 上下偏移
    mm = _parni(5); // 左右偏移
    color = _parni(6); // 画线颜色
    yy = y;
    y = x * 18 + mode;
    x = yy * 8;
    FP_OFF(buf) = 80 * y + x / 8;
    FP_SEG(buf) = 0xa000;
    FP_OFF(buf1) = 80 * y + x / 8 + len - 1;
    FP_SEG(buf1) = 0xa000;
    for(m = 0; m < 4; m++)
    {
        outportb(0x3ce, 4);
        outportb(0x3cf, m);
        buff = * buf;
        buff1 = * buf1;
        if((color >> m) & 1)
        {
            buff |= poin[mm];
            buff1 |= (0xff - poin[8 - mm]);
        }
        else
        {
            buff &= (0xff - poin[mm]);
            buff1 &= (poin[8 - mm]);
        }
        outportb(0x3c4, 2);
        outportb(0x3c5, 8 >> (3 - m));
    }
}
```

```

_fmmemset(buf, buff, 1);
_fmmemset(buf1, buff1, 1);
outportb(0x3c4, 2);
outportb(0x3c5, 0x0f);
}

FP_SEG(buf) = 0xa000;
FP_OFF(buf) = 80 * y + x / 8 + 1;
for(m = 0; m < 4; m++)
{
    outportb(0x3c4, 2);
    outportb(0x3c5, 8 >> (3 - m));
    _fmmemset(buf, 0xff * ((color >> m) & 1), len - 2);
}
outportb(0x3c4, 2);
outportb(0x3c5, 0x0f);
_ret();
}

```

```

char px(char y)
{
    char i, s = 1;
    for(i = 0; i < y; i++)
        s *= 2;
    return(s);
}

```

```

CLIPPER mvline() /* 画竖线 */
{
    int x, y, len, mode, mm, color;
    char far * buf;
    char buff;
    int m, i, yy;
    char poin;
    x = _parni(1);
    y = _parni(2); // 起点坐标
    len = _parni(3); // 长度
    mode = _parni(4); // 左右偏移
    mm = _parni(5); // 上下偏移
    color = _parni(6); // 画线颜色
    yy = y;
    y = x * 18 + mm;
    x = yy * 8;
    len = len * 18 - 2 * mm;
    poin = px((char)(7 - mode));
    for(i = 0; i < len; i++)
    {
        FP_OFF(buf) = 80 * (y + i) + x / 8;
        FP_SEG(buf) = 0xa000;
        for(m = 0; m < 4; m++)
        {

```

```

            outportb(0x3ce, 4);
            outportb(0x3cf, m);
            buff = * buf;
            outportb(0x3ce, 4);
            outportb(0x3cf, 0);

            if((color >> m) & 1)
                buff |= poin;
            else
                buff &= (0xff - poin);

            outportb(0x3c4, 2);
            outportb(0x3c5, 8 >> (3 - m));
            _fmmemset(buf, buff, 1);
            outportb(0x3c4, 2);
            outportb(0x3c5, 0x0f);
        }
    }
    _ret();
}

```

程序中用到了 CLIPPER 的扩展函数来完成参数传递:

_parni(int[, int]) : 它从 CLIPPER 中获取一个整数。

_ret() : 向 CLIPPER 返回空值。

程序中应包括 EXTEND.H 来对上述函数声明。

以上程序在 Borland C3.1 下编写, 稍加修改库函数名称可适用于 MS C。

把上面的程序编译成 .obj 文件, 再和 CLIPPER 程序一起连接, 即可调用 mhline(), mvline() 来画横线和竖线。编译时应注意:

在 Borlandc 下:

```
bcc -c -ml mline.c
```

在 MS C 下

```
cl /c /AL /Gs /Zl /Oalt /Fpa mline.c
```

有了这两个函数, 我们就可以在 CLIPPER 中定义以下函数:

```

/* 画方框 */
/* x1, y1, y2, y2: 方框左上右下坐标 */
/* mode: 0-7 偏移点数 一行 18 点 一列 8 点 */
proc whorx
para x1, y1, x2, y2, mode, color
mhline(x1, y1, y2 - y1 + 1, 17 - mode, 8 - mode, color)
mhline(x2, y1, y2 - y1 + 1, mode, 8 - mode, color)
mvline(x1, y1, x2 - x1 + 1, 7 - mode, 17 - mode, color)
mvline(x1, y2, x2 - x1 + 1, mode, 17 - mode, color)

```

```

return

/* 画外方框 */
proc mebox
para x1,y1,x2,y2,color
x1=x1-1
y1=y1-1
x2=x2+1
y2=y2+1
wbox(x1,y1,x2,y2,0,color)
return

/* 窗口 */
proc window
para x1,y1,x2,y2,str /* x1,y1,x2,y2:左上及右下坐标
*/
set colo to 7+/1 /* str: 窗口标题 */
@ x1,y1 clea to x1,y2
@ x1,y1+(y2-y1-len(str))/2 say str
set colo to 0/7
scroll(x1+1,y1,x2,y2,0)
@ x1,y1 say '-'
mhline(x1,y1,y2-y1+1,17,0,0)
mvline(x1,y1+2,1,0,0,0)
x1=x1-1
x2=x2+1
y1=y1-1
y2=y2+1
wbox(x1,y1,x2,y2,0,0)
wbox(x1,y1,x2,y2,1,7)
wbox(x1,y1,x2,y2,2,7)
wbox(x1,y1,x2,y2,3,0)
return

/* 按钮 */
proc mbar
para x1,y1,x2,y2,mode,size /* x1,y1,x2,y2:左上及右
下坐标 */
priv i,c1,c2 /* mode 1:上凸 */
i=0 /* size 凹凸边缘宽度 */
if mode=1
c1=15
c2=0
else
c1=0
c2=15
endif
do while(i<size)
mhline(x1,y1,y2-y1+1,i,0,c1)
mhline(x2,y1,y2-y1+1,17-i,0,c2)
i=i+1
endd

```

```

i=0
do while (i<size)
mvline(x1,y1,x2-x1+1,i,i,c1)
mvline(x1,y2,x2-x1+1,7-i,i,c2)
i=i+1
enddo
return

/* 类 WINDOWS 的提示窗口 */
proc block
para x1,y1,x2,y2,st
priv i
i=1
set colo to 7+/1
@ x1,y1 clea to x1,y2
@ x1,y1+(y2-y1-len(st))/2 say st
set colo to 7+/7
@ x1,y1 say '-'
scroll(x1+1,y1,x2,y2,0)
set colo to 0/7
mhline(x1,y1,y2-y1+1,17,0,0)
mvline(x1,y1+2,1,0,0,0)
mhline(x1-1,y1-1,y2-y1+3,17,8,15)
mhline(x2+1,y1-1,y2-y1+3,0,8,0)
mvline(x1-1,y1-1,x2-x1+3,7,17,15)
mvline(x1-1,y2+1,x2-x1+3,0,17,0)
do while (i<5)
wbox(x1-1,y1-1,x2+1,y2+1,i,1)
i=i+1
enddo

/* 提示窗口 */
proc wwait
para info1
priv x1,y1,x2,y2,len1,key,w
len1=len(info1)
x1=9
y1=24
x2=16
y2=55
w=savescreen(x1-1,y1-1,x2+1,y2+2)
set colo to /0
scroll(x1+1,y1+2,x2+1,y2+2,0)
bar(x1,y1,x2,y2,'系统提示')
mbar(x1,y1,x2,y2,0,1)
@ x1+3,y1+(y2-y1+1-len1)/2-2 say '【'+info1+'】'
,
@ x1+5,y1+7 say '按任意键继续...'
tone(800,5)
key=inkey(0)
restscreen(x1-1,y1-1,x2+1,y2+2,w) (转第41页)

```


也谈用五笔字型为 AutoCAD 输入汉字

湘潭矿业学院机械工程系 张会福

摘 要: 本文用 AutoCAD 11 版提供的 ADS 功能, 为 CAD 提供一个能在西文下输入并标注汉字的函数。

关键字: 码表, 型文件, ADS

1、引 言

最近读了贵刊总第 42 期('95. 7) 兰文祥先生《用五笔字型为 AutoCAD 输入汉字》一文, 觉得该文提出的方法很有实用性。但是文章中的程序却是用 LISP 语言开发的, LISP 语言有如下的不足之处: (1) LISP 语言是一种解释语言, 当字码比较完全 (有上万条) 时检索很慢; (2) 使用的资源有限, 不能对操作系统提供的功能进行利用; (3) 程序用户界面无法设计得美观; 等等。本人也曾用 C 语言开发过功能相同的函数, 通过在 CAD 上使用这段时间来看, 效果很好。该函数把字码表放在磁盘文件上, 因此可使用各种自编码表, 只要求字码格式为“字码汉字”, 码的长度小于程序中的 CODE_LEN, 字词长度不限, 如:

```
a 工
b 了
pygl 福
aawt 工作
rssf 技术要求
fiwq 未注倒角
```

等等, 有如 WINDOWS 系统中的通用码表输入法, 码表的行数不受限制, 本人使用的是从 UC DOS 中提取的五笔字型码表字词两万多行, 运行时并不觉得慢; 函数能在 AutoCAD 中汉字直接写屏, 输入的汉字和输入情况以及重码都可在提示行中看到, 运行函数一次最多可输入 TXT_BUF_LEN / 2 个汉字, 且输入方式与一般的汉字系统输入法相同。下面是程序清单。

2、程序清单

```
/* File name: ctext.c 张会福 1995. 3. 17
   compile: wc386p ctext -fpi287 -3s -s -oailt -zq
   link: wlinkp system ads file ctext library wcads90 option
   quiet */
#include "string.h"
#include "stdio.h"
#include "graph.h"
```

```
#include "adslib.h"
#define TEXT_BUFFER_LEN 200
#define CODE_LEN 5
#define ELEMENTS(array) (sizeof(array)/sizeof((array)
[0]))
char key_buf[CODE_LEN], txt_buf[TEXT_BUFFER_LEN];
int txt_num; /* 当前汉字个数 */
struct videoconfig VC;
void main (int argc, char * argv[]);
int loadfuncs (void);
int ctext(void);
int get_key(void); /* 取键盘输入到 key_buf */
int get_txt(void); /* 从 key_buf 取值到 txt_buf */
int show_txt(char * buf);
/* 从 txt_buf 中得到汉字并显示它们 */
void dispcc16(int x, int y, char * cc); /* 显示汉字串 */
void getcodefromstr(char * from_buf, char * to_buf);
/* 取码子程序 */
int setmode(void); /* 设置 CAD 文本窗口的图形模式 */
FILE * ccfont, * inputcode;
typedef int (* MYFUNC) (void);
/* function pointer */
struct f_item /* single function info */
{
/* function name — function pointer */
char * f_name;
MYFUNC f_pointer;};
/* declare function table and supply values */
struct f_item f_table[] =
{
/* function name — function pointer */
{"ctext", ctext},};
/* MAIN — the main routine */
void main(argc, argv)
int argc;
char * argv[];
{
int stat, f_number;
short scode = RSRSLT;
/* This is the default result code */
```

```

ads_init(argc, argv); /* Initialize the interface */
while ((stat = ads_link(scode)) >= 0)
{
    scode = RSRSLT; /* Default return value */
    switch (stat)
    {
        case RQXLOAD: /* load functions */
            scode = loadfuncs() ? RSRSLT : RSERR;
            break;
        case RQSUBR:
            /* determine which function to run */
            if ((f_number = ads_getfuncode()) >= 0)
            { (*f_table[f_number].f_pointer)();
              scode = RSRSLT; }
            else
            { ads_abort("Error, function is not in function
table");
              scode = RSERR; }
            break;
        default:
            scode = RSRSLT;
            break;
    } /* end switch */
} /* end for loop */
printf("\nLoader: bad status from ads_link() = %d\n", stat);
printf("Link fail !");
fflush(stdout);
} /* end main */
/* LOADFUNCS—Define functions with ads_defun() */
static int loadfuncs()
{
    int i;
    char func_buf[30];
    func_buf[0] = '\0';
    ads_printf(" \nThis programe define following
functions:");
    strcpy(func_buf, "C:");
    /* include STRING.H for STRCPY */
    for (i = 0; i < ELEMENTS(f_table); i++)
    {
        strcpy(func_buf + 2, f_table[i].f_name);
        ads_printf("\nfunc_name%i = %s", i, func_buf);
        if (! ads_defun(func_buf, i))
        { ads_abort(" \nError defining function: %s", func_
buf);
          return RSERR; }
    }
    setmode();
    return RSRSLT;
}

```

```

int ctext(void)
{
    int i;
    if ((ccfont = fopen("hzk16", "rb")) == NULL)
        {ads_printf("\nCan't found clib16j. dot ! \n");
         return RSRSLT;
        }
    if ((inputcode = fopen("wbzx.txt", "rb")) == NULL)
        {ads_printf("\nCan't found input_code file! \n");
         return RSRSLT;
        }
    ads_graphscr(); /* 输入汉字前需先进入图形方式 */
    for (i = 0; i <= TEXT_BUFFER_LEN; i++)
        txt_buf[i] = 0; /* clear the text buffer */
    _setcolor(1);
    _rectangle(_GFillInterior, 0, VC.numypixels -
40,
              VC.numypixels, VC.numypixels);
    _setcolor(12);
    dispcc16(180, 0, "汉字标注 V1.0a");
    dispcc16(10, VC.numypixels - 17, "五笔字型:");
    txt_num = 0;
    while(1) /* 无限循环 */
    {
        get_key();
        if(key_buf[0] == 13) /* 只有空一个回车才结束 */
            break;
        if(key_buf[0] != 0)
            get_txt();
        _setcolor(1);
        _rectangle(_GFillInterior, 10, VC.numypixels -
40,
                  120 + TEXT_BUFFER_LEN * 17, VC.numypixels -
20);
        _setcolor(12);
        dispcc16(10, VC.numypixels - 38, txt_buf);
    }
    fclose(ccfont);
    fclose(inputcode);
    _setcolor(1);
    _rectangle(_GFillInterior, 0, VC.numypixels -
40,
              VC.numypixels, VC.numypixels);
    _setcolor(12);
    show_txt(txt_buf);
    return RSRSLT;
}
int get_key()
{ /* 键盘输入模块,输入四个字符送入 key_buf */
    int i;
    char c;

```

```

for(i=0;i<=CODE_LEN;i++) key_buf[i]='\0';/*
清除键输入缓冲区 key_buf */
i = 0;
while(i < CODE_LEN - 1) /* 调用一次输入一个字 */
{
    c=getch();
    switch (c)
    {
        case 8:
            if (i > 0)/* there are key in key_buf */
            { i--;
              key_buf[i]='\0'; }
            else
            if(txt_num >= 2)
            /* 从 txt_buf 中删除一个字(如果那儿有字的话) */
            {
                if(txt_buf[txt_num - 1] > 0xa1)/* 是个汉字 */
                { txt_buf[txt_num - 1]='\0';
                  txt_buf[txt_num - 2]='\0';
                  txt_num-=2;
                  i = CODE_LEN; }
                else/* 是个字母 */
                { txt_buf[txt_num - 1] = '\0';
                  txt_num -= 1;
                  i = CODE_LEN; }
            }
            else if (txt_num >= 1)
            { /* 只有一个字符键 一定是字母 */
              txt_buf[txt_num - 1] = '\0';
              txt_num -= 1;
              i = CODE_LEN; }
            break;
        case 13:
            if (i > 0)/* 回车则废除刚才输入 */
            { for (i = 0;i < CODE_LEN;i++)
              key_buf[i] = '\0';
              i = 0; }
            else/* 或结束汉字输入 */
            {key_buf[0]=13;
             i=CODE_LEN; }
            break;
        case 27:
            exit(0);
        case 32:
            if (i > 0)/* 已输入完一个字 */
            i=CODE_LEN;
            else/* 输入一个空格符到 txt_buf 中 */
            {txt_buf[txt_num]=' ';
             txt_num+=1; }
            break;
        default:
            if(islower(c))
            {key_buf[i]=c;
             i+=1; /* if a lower send to key_buff */
            }
            else if(isgraph(c))/* 是可打印字符 */
            {
                if(i==0)/* the first char */
                { txt_buf[txt_num]=c;
                  txt_num += 1;
                  txt_buf[txt_num]=0;
                  i=CODE_LEN; }
                }
            break;
    } /* switch */
    _setcolor(1);
    _rectangle(_GFillINTERIOR,80,VC.numypixels -
17,120,VC.numypixels);
    _setcolor(12);
    _setcharsize(16,8);
    _grtext(80,VC.numypixels-17,key_buf);/* display
input code */
} /* while */
return RSRSLT;
} /* end function get_key */
int get_txt(void)
{
    int i,polycode,buflen;
    char c;
    char key[CODE_LEN],read_buf[TEXT_BUFFER_
LEN];
    char ccbuf[TEXT_BUFFER_LEN];
    polycode = 0;
    for(i=0;i<TEXT_BUFFER_LEN;i++)
    { ccbuf[i] = 0;
      read_buf[i] = 0;
    }
    for(i=0;i<CODE_LEN;i++)
    key[i] = 0;
    fseek(inputcode,0,0);/* 从头开始 */
    while ((fgets (read_buf, TEXT_BUFFER_LEN,
inputcode))!=NULL)
    { /* 找到最后一个字 */
      getcodefromstr(read_buf,key);
      if(strcmp(key,key_buf)==0)/* 找到字码 */
      {strcat(ccbuf,read_buf);
        /* 把找到的字加入到 ccbuf 中去 */
        polycode += 1; }
    }
    if(polycode >= 2)/* 有重码 */
    {
        _setcolor(1);

```

```

    _rectangle(_GIFILLINTERIOR,120,VC.numypixels -
17,
    VC.numypixels,VC.numypixels);
    _setcolor(12);
    dispcc16(120,VC.numypixels-17,ccbuf);
    /* 显示所有重码 */
    c=getch();
    if (isdigit(c)) /* 选择第几个 */
    { txt_buf[txt_num] =ccbuf[(c-49) * 2];
      txt_buf[txt_num+1]=ccbuf[(c-49) * 2+1];
      txt_num+=2; }
    else
    { txt_buf[txt_num]=ccbuf[0];
      txt_buf[txt_num+1]=ccbuf[1];
      txt_num+=2; }
    polycode = 0;
  }
  else
  { setcolor(1);
    _rectangle(_GIFILLINTERIOR,120,VC.numypixels -
17,
    VC.numypixels,VC.numypixels);
    _setcolor(12);
    dispcc16(120,VC.numypixels-17,ccbuf);
    strcat(txt_buf,ccbuf);
    txt_num+=strlen(ccbuf); }
  return RSRSLT;
} /* end function get_txt */
void dispcc16(int x,int y,char * cc)
/* 显示汉字或字母子程序 */
{
  unsigned char qm,wm,buf[32],C_upper[2];
  int locate,i,j,Ccline;
  C_upper[1] = 0;
  while(*cc)
  {
    if(*cc > 0xa1) /* 是汉字 */
    {
      qm=*cc++ - 161;
      wm=*cc++ - 161;
      locate=qm*94+wm;
      fseek(ccfont,locate*32L,SEEK_SET);
      fread(buf,32,1,ccfont);
      for (j=0;j<16;j++)
      {
        Ccline=(buf[j*2]<<8)+buf[j*2+1];
        _setlinestyle(Ccline);
        _setcolor(12);
        _moveto(x,y+j);
        _lineto(x+15,y+j);
      } /* for */
    }
  }

```

```

    x+=17;
  } /* if */
  else /* 是字母 */
  {
    C_upper[0]= *cc++;
    _setcharsize(16,8);
    _grtext(x,y,C_upper);
    x+=9;
  }
} /* while */
_setlinestyle(32767);
} /* end dispcc16 */
void getcodefromstr(char * from_buf,char * to_buf)
/* 取码子程序 */
{
  int i,codelenth,buflenth;
  buflenth = strlen(from_buf)-2;
  /* 回车换行也读进来了 */
  for(i=0;i < strlen(from_buf);i++)
  {
    if(from_buf[i] > 122)
      break;
    to_buf[i] = from_buf[i];
  }
  to_buf[i] = '\0';
  codelenth = i;
  for(i=codelenth;i < buflenth;i++)
    /* 最后多 codelenth 个字符,去掉 */
    from_buf[i-codelenth] = from_buf[i];
  from_buf[i-codelenth]=0;
}
int setmode(void)
{ /* 设置文本窗口成图形方式,以便汉字直接写屏 */
  ads_graphscr(); /* 转入图形窗口 */
  _getvideoconfig(&VC); /* 取显示模式 */
  _setvideomode(VC.mode); /* 设置成图形模式 */
  _getvideoconfig(&VC); /* 取图形模式的最大点数 */
  _setcolor(1); /* 以下恢复弄乱了图形屏幕 */
  _rectangle(_GIFILLINTERIOR,0,0,VC.numypixels,
VC.numypixels);
  ads_command(RTSTR,"redraw",0);
  ads_menuscmd("S=SCREEN");
  ads_textscr();
  ads_graphscr();
  return RSRSLT;
}
int show_txt(char * buf)
{
  struct resbuf vall;
  ads_point pt1;
  ads_real txt_hight,txt_angle;

```

```

ads_initget(1);
if ((ads_getpoint(NULL,"TEXT begin point:",pt1))!
=RTNORM)
{
ads_fail("\nError input point.");
return RSRSLT;
}
ads_initget(1+2+4);
if ((ads_getreal("Input TEXT height:",&txt_height))!
=RTNORM)
{
ads_fail("\nError input TEXT height.");
return RSRSLT;
}
ads_initget(1);
if ((ads_getreal("Input TEXT angle:",&txt_angle))!
=RTNORM)
{
ads_fail("\nError input TEXT angle.");
return RSRSLT;
}
val1.restype=RTSHORT;
val1.resval.rint=0;
ads_setvar("CMDECHO",&val1);
ads_setvar("BLIPMODE",&val1);
ads_command(RTSTR,"text",RTPOINT,pt1,
RTREAL,txt_height,
RTREAL,txt_angle,RTSTR,buf,0);
val1.restype=RTSHORT;

```

```

val1.resval.rint=1;
ads_setvar("CMDECHO",&val1);
ads_setvar("BLIPMODE",&val1);
return RSRSLT;
}

```

3. 程序使用注意事项

(1) 本程序用 WATCOM C 9.01 编译成 CTEXT.EXP 文件,在“COMMAND:”提示符下用“(XLOAD”CTEXT”)”命令装入,然后在“COMMAND:”提示符下用“CTEXT”命令调用函数,就可定义码表输入汉字了;

(2) 当前目录中必须有字库文件“HZK16”和码表文件“WBZX.TXT”;

(3) 使用前先设置当前字型使包含汉字型“HZTXT”;

(4) 出现重码时再输入 1、2 等数字选择重码字;

(5) CAD 的屏幕如果不是设置成标准 EGA 或 VGA,则会使屏幕底色发生变化,但并不影响操作;

(6) 标注时亦可输入特殊字符,如%%D(度符号“°”)等;

(7) 空回车结束汉字输入。

(8) 本程序已在 AutoCAD 11.0 和 12.0 下测试通过。

(本文收稿日期:1995.9.23)

(接 36 页)有了以上的过程,则在程序中即可直接调用它们。

例如:

```

/* kc.prg */
set colo to 7/3
clea
x00=3
y00=6
window(x00,y00,x00+18,y00+66,'系统主菜单')
mhline(x00+2,y00,67,0,0)
set key 24 to down_key
set colo to &deskcol,&mcurcol
@ x00+1,y00+1 prom "系统管理"
@ x00+1,y00+11 prom "仓库设置"
@ x00+1,y00+21 prom "单据管理"
@ x00+1,y00+31 prom "库存查询"
@ x00+1,y00+41 prom "报表打印"
@ x00+1,y00+51 prom "辅助功能"
menu to m00
if m00<>0
wwait("此功能尚未建立")
endif

```

以上程序在中文 CLIPPER5.01 下编写。读者

还可以通过调用画线函数来定义一些实现其它图形界面的函数。

应注意的是:

1. 由于 CLIPPER 是工作在字符模式下,SAVE SCREEN 等只是保存屏幕的字符及其属性,并不保存你用写屏画的图形,因此在 RESTORE SCREEN 时,写屏所画的图形也将消失,解决的办法是重新画上。这就要求图形不要过于复杂,以免影响软件的运行速度。

2. 汉字系统应选用支持直接写屏的如 UC DOS, CXDOS, TWAY 等。

3. CLIPPER 应是已汉化版本。

4. 本文中的画线只是针对 VGA 方式,且 VRAM 不得少于 512k。

参考书目:

- [1] 廖斌 张林编著,《CLIPPER 高级编程技巧与实例》,北京希望电脑公司
- [2] 陶伟 张东等编著,《EGA VGA TVGA 高级微机图形编程指南与实例》,科学出版社

(本文收稿日期:1995.9.18)

应用软件中的汉字输入方法

空军导弹学院机械基础教研室[713800] 兰文祥

摘 要 本文提出了一种应用软件中的汉字输入方法,即可使用五笔字型,也可使用区位码输入汉字,简单实用。

关键词 五笔字型,区位码,汉字输入

应用软件中有时需要输入汉字,例如我们编制的《机械制图题库管理系统》中就需要解决汉字的输入问题。汉字输入如采用汉字系统,则系统庞大,且界面不好控制。为此我们利用汉字的点阵字库、五笔字型编码、区位码,解决了应用软件中的汉字输入问题。程序如下,供参考。(本程序用 Turbo 2.0 编程,在长城 386 上运行通过,汉字库为 UC DOS3.1 的 16 点阵字库 hzk16)

```
#define ESC 27
#define BELL 7
#define RETURN 13
#define BACKSPACE 8
#define SPACEBAR 32
#define hzsize 30 /* 收集的五笔字型编码个数 */
#include <graphics.h>
#include <stdio.h>
#include <dos.h>
#include <ctype.h>
#include <string.h>
struct wbxhzindex {
    char *wbxcode;
    char *wbxhz;
};
struct wbxhzindex wbxstr[hzsize]=
{"PV","安","RPV","按","PVS","案","MMGD","
凹","WTY","八","RDC","拔","RCN","把","
RRRR","白","RLFC","摆","GYTG","班","TEMC","
般",NULL,NULL};

int maxx,maxy,row=0;
FILE *ccfp16;

void setgrmode()
{
    int gd=VGA,gm=VGAHI;
    initgraph(&gd,&gm,"c:\\tc");
    maxx=getmaxx();
```

```
maxy=getmaxy();
}

void setpromptport()
{
    setviewport(0,maxy-100+10,maxx,maxy,1);
}

void setgraphport()
{
    setviewport(10,10,maxx-19,maxy-100-20,1);
}

void frame()
{
    rectangle(0,0,maxx,maxy-100-11);
    line(0,maxy-100,maxx,maxy-100);
}

void out_cc_text(int x,int y,char *str)
/* 输出汉字串 */
{
    unsigned int zcode,bcode,hzcode,ccmemloc;
    int i,j,k,rec,CCcolor;
    long int len;
    char buf[32],textstr[2],*zimo;
    CCcolor=getcolor();
    while(*str){
        if ((*str&0x80)&&(* (str+1)&0x80)) {
            zcode=(*str-0xa1)&0x07f;
            bcode=(* (str+1)-0xa1)&0x07f;
            memcpy(&hzcode,str,2);
            rec=zcode*94+bcode;
            len=rec*32L;
            fseek(ccfp16,len,SEEK_SET);
            fread(buf,1,32,ccfp16);

            for(i=0;i<16;i++)
                for(j=0;j<2;j++)
```



```

for (k=0;k<8;k++)
if (buf[i*2+j]>>(7-k)&1)
putpixel(x+j*8+k,y+i,CCcolor);
x=x+16+2;
str=str+2;
}
else {
textstr[0]=*str;
textstr[1]='\0';
outtextxy(x,y,textstr);
x=x+16+2;
str=str+1;
}
}
setpromptport();
return;
}

int get_char() /* 读键盘 */
{
int c,flag,mx,my;
static union REGS rg;
while (1) {
if (kbhit()) {
rg.h.ah=0;
int86(0x16,&rg,&rg);
if (rg.h.ah==0)
continue; /* function key */
else {
c=rg.h.ah;
return c;
}
}
}
}

void prompttext(int code,int lx,char *prstr)
/* 输出提示信息 */
{
setpromptport();
if (((row%3)==0) && (code==0))
clearviewport();
if (code==1) {
row=0;
clearviewport();
}
out_cc_text(lx,3+(row%3)*18,prstr);
row=row+1;
return;
}

```

```

int textlen(char *textstr)
/* 计算汉字西文混合字符串个数 */
{
char *textstrp;
int len=0;
textstrp=textstr;
while (*textstrp) {
if ((*textstrp & 0x80) && ((*textstrp+1) &
0x80)) {
textstrp=textstrp+2;
len=len+1;
}
else {
len=len+1;
textstrp=textstrp+1;
}
}
return len;
}

void open_cc_lib()
{
ccfp16=fopen("c:\\ucdos\\hzk16","rb");
if (ccfp16==NULL) {
prompttext(1,0,"ucdos 3.1 hzk16 file not found");
prompttext(0,0,"program aborted.");
getch();
exit(1);
}
}

void close_cc_lib()
{
fclose(ccfp16);
}

```

```

void getstr(char *promptstr,char *pstr,int len)
/* 读字符串 */
{
int i=0,ch=0,j,x,y,tempcolor;
char ps[1],s[2],str[30];
tempcolor=getcolor();
x=textlen(promptstr)*18;
prompttext(0,0,promptstr);
s[1]='\0';
str[1]='\0';
ps[1]='\0';
y=3+((row-1)%3)*18;
setpromptport();
for(;;) {
j=i*16+2;

```

```

    if (i==len)
        break;
    ch=get_char();
    if ((ch==RETURN) || ((len==4) &&( ch==
SPACEBAR)))
        break;
    if (ch==BACKSPACE) {
        if (--i<0) i=0;
        j=i*16+2;
        setcolor(0);
        sprintf(ps,"%c",str[i]);
        outtextxy(x+j,y,ps);
        setcolor(tempcolor);
    }
    else if (isgraph(ch)) {
        str[i]=ch;
        s[0]=ch;
        setcolor(tempcolor);
        outtextxy(x+j,y,s);
        i++;
        j=i*16+2;
    }
}
j=i*16+2;
str[i]='\0';
strcpy(pstr,str);
}

```

```

void ptstr_to_num(char * ps,int * val1,int * val2)
/* 位置字符串转换成数据 */

```

```

{
    char vps1[20],vps2[20];
    int i1=0,i2=0;
    while(ps[i1]!='(',')') {
        vps1[i1]=ps[i1];
        i1=i1+1;
    }
    vps1[i1]='\0';
    i1=i1+1;
    i2=0;
    while(ps[i1+i2]) {
        vps2[i2]=ps[i1+i2];
        i2=i2+1;
    }
    vps2[i2]='\0';
    * val1=atoi(vps1);
    * val2=atoi(vps2);
}

```

```

void rd_point_loc(char * ptstr,int * ptx,int * pty)
/* 读点的坐标 */

```

```

{
    char txtptstr[20];
    setpromptport();
    getstr(ptstr,txtptstr,19);
    ptstr_to_num(txtptstr,ptx,pty);
}

```

```

int wbxhz(char * wbx)
/* 检索是否收集该五笔字型编码 */
{
    int i;
    for (i=0;i<hsize;i++) {
        if (strcmp(wbx,wbxstr[i].wbxcode)==0)
            return (i);
        if (wbxstr[i].wbxcode==NULL)
            return (-1);
    }
    return (-1);
}

```

```

void quicktext(char * textstr)
{
    unsigned int qw;
    char qwstr[5],asctext[30];
    int txtxs,txtys;
    int wbxstatus=1,wbxhnum,len;
    char text[100];
    setpromptport();
    rd_point_loc("start point :",&txtxs,&txtys);
    strcpy(textstr,"");
    while(1) {
        if (wbxstatus==0)
            getstr("区位码 :",qwstr,4);
        else
            getstr("五笔型 :",qwstr,4);
        strupr(qwstr);
        if (strcmp(qwstr,"TEXT")==0) {
            getstr("Text :",asctext,29);
            setgraphport();
            out_cc_text(txtxs,txtys,asctext);
            txtxs=txtxs+textlen(asctext)*18;
            strcat(textstr,asctext);
            continue;
        }
        if (strcmp(qwstr,"QWM")==0) {
            wbxstatus=0;
            continue;
        }
        if (strcmp(qwstr,"WBX")==0) {
            wbxstatus=1;
            continue;
        }
    }
}

```

```

}
if (strcmp(qwstr,"EXIT")==0) {
return;
}
switch(wbxstatus) {
case 0: /* qwm */
qw=atoi(qwstr);
len=strlen(textstr);
* (textstr+len+0)=qw / 100+0xa0;
* (textstr+len+1)=qw % 100+0xa0;
* (textstr+len+2)='\0';
setgraphport();
out_cc_text(txtxs,txtys,textstr+len);
txtxs=txtxs+18;
wbxstatus=1;
break;
case 1: /* wbh */
if (strlen(qwstr)==0) {
return;
}
wbxhznnum=wbxhz(qwstr);
if (wbxhznnum>=0) {
strcat(textstr,wbxstr[wbxhznnum].wbxhz);
setgraphport();
out_cc_text(txtxs,txtys,wbxstr[wbxhznnum].wbxhz);
len=textlen(wbxstr[wbxhznnum].wbxhz);
txtxs=txtxs+textlen(wbxstr[wbxhznnum].wbxhz)*
18;
}
else {
prompttext(0,0,"这个编码没收入,请改用区位码输入.");
putchar(BELL);
putchar(BELL);
}
break;
default : break;
}
}

main()
{
char textstr[100];
setgrmode();
clearviewport();
setbkcolor(BLUE);
setcolor(YELLOW);
open_cc_lib();
frame();
settextjustify(LEFT_TEXT,TOP_TEXT);
settextstyle(DEFAULT_FONT,HORIZ_DIR,2);
quicktext(textstr);
setgraphport();
clearviewport();
out_cc_text(10,100,textstr);
close_cc_lib();
getch();
closegraph();
}

```

(本文收稿日期:1995.10.1)

《电子文摘报》集国内外百余种报刊之精华(维修、资料、制作、知识、消费)于一体,是国内唯一的一份实用电子技术文摘类报纸。

第一版:知识·信息·消费·评论版

第二版:AV(视听)·电脑·办公自动化版

第三版:维修技巧专版

第四版:实验制作·实用资料版

本报为周报,每周五出版,四开四版,每期

定价0.18元,全年定价9.36元。邮发代

号:61-87。免费订阅报纸,还

能获得其他收获,请见

以下优惠办法。

《家庭电子》杂志以其新颖独特的实用文章而独树一帜,是面向亿万家庭的实用电子科普刊物。设有:消费天地、电子生活、AV天地、电脑园地、家电维修、个人通信、跟我学电子、资料汇编、每期一图、点题台(回音)、海外窗口等栏目。本杂志为月刊、每月15日出版,16开,32页,每期定价1.40元,全年定价16.80元。邮发代号:62-189。免费订阅杂志,还能获得其他收获,请见以下优惠办法。

家电消费者的良友·家电维修者的助手·电子爱好者的天地·电子工作者的参谋

●凡订阅全年报纸或杂志的读者(厂商除外),将免费获得本报刊社出版的与订报或刊等价的实用科技书籍或资料(请全年订阅报刊的读者将订报凭证原件寄本报编辑部,并将详细地址、邮编写清楚)。

●订阅全年报刊者,除享受免费赠送的书籍或资料外,还将有机会报销报纸的订费(限1万人,截止时间一九九五年十二月一日,以邮戳先后秩序额满为止,须寄订订全年报刊凭证原件)。

●订阅全年报纸或杂志的厂商,可获得一次免费在报缝(面积15cm²)或杂志内文(信息专栏)刊登广告或信息的机会(手续必须符合《广告法》的有关规定);个人可刊登专利消息或转让信息。

●如您不甚了解本报刊,不妨来函免费试阅两期。

●通讯员及撰稿者请寄回订订发票,予以报销。

★报刊社地址:四川省成都市抚琴东南路10号二单元7楼(市118信箱)

电 话:(028)7741629

邮编:610031

免费赠阅·新的起点

GW—200 彩显特有的故障及其修复

广西桂林铁路电务段[541001] 刘应琪

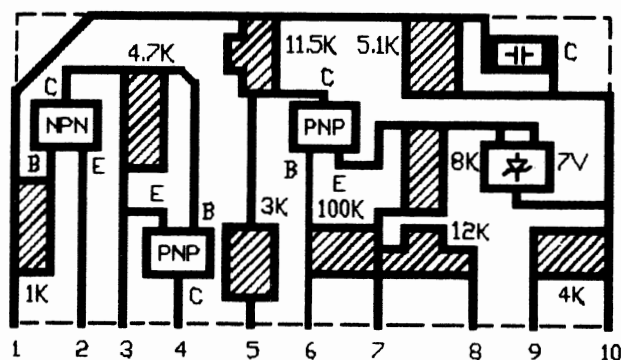
GW—200 是长城 EGA 彩显,使用至今已进入维修期。该设备电源部份故障率颇高,动不动就烧保险和开关调整管。为此笔者作了一番分析研究,发现一个普遍规律:输入电压允许范围整个偏低,在 75—220 伏之间。是什么原因引起的呢?外接自耦调压器试验,在上述电压范围内稳压输出正常。由低到高继续调压,到 210 伏时比较放大管开始饱和,超过 210 伏则输出电压随之升高,失去稳压作用。由此判断电路基本正常,可能有元件变质导致电路特性变化。然而检查所有元器件都符合标称值,更换厚膜块也无用。又反复检查电路板及各焊点,均未发现问题。

山穷水尽疑无路。我们再从原理上分析:取样比较放大后加到开关调整管基极一个负电压,该负电压与开关变压器反馈到基极的正电压相互作用控制开关调整管。前面所做的工作表明取样比较放大部份是好的,很可能是正反馈过强所致。试将正反馈回路电阻 R335 换成 $47\Omega/2W$,故障消失。输入电压调至 250V 仍然保持输出不变。

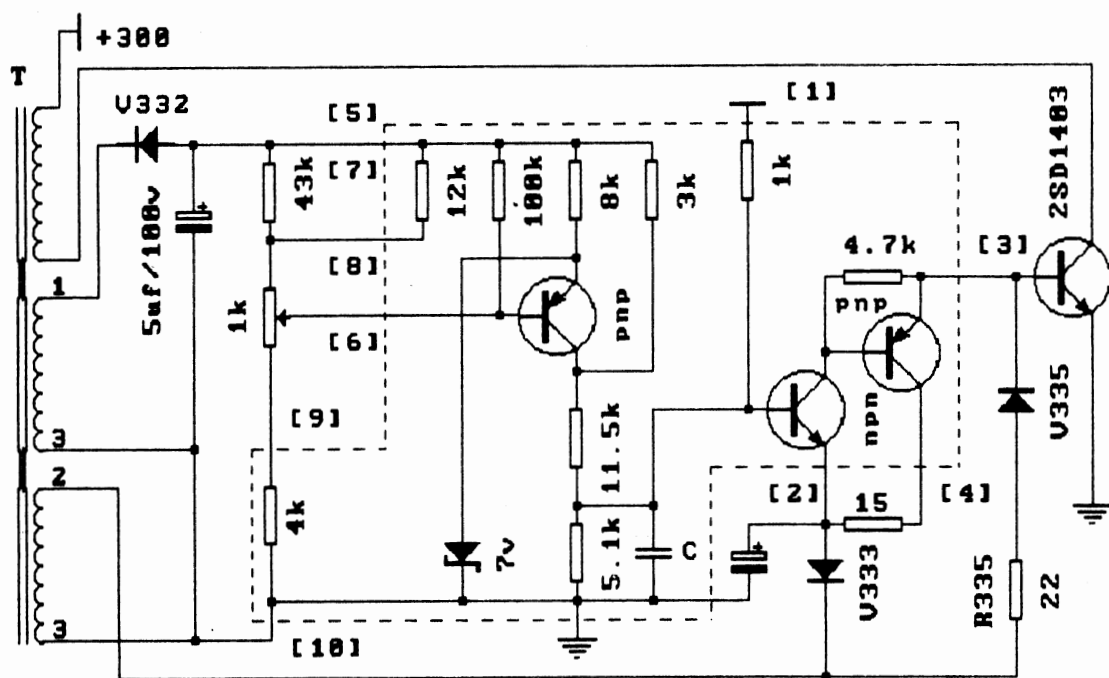
笔者用此法修复了多台 GW200 显示器,可见此故障有一定普遍性。笔者推测该显示器原是在 110V 电源工作的,国产化时忽略了过压试验,使用日久特性稍稍变差就会出问题。

附上实测绘制的原理图及厚膜块图各一份,供同行参考。

(本文收稿日期:1995.9.17)



电源厚膜块元件布置图



GW200 彩显开关电源原理图(部份)(T——开关变压器;[1]—[10]——电源厚膜块引脚;虚线框内是电源厚膜块)

FoxPro 中使用数组 必须注意的几个问题

安徽财贸学院[233041] 车光宏

摘 要:本文说明了 FoxPro 中 COPY TO ARRAY 命令的正确用法和过程(用户函数)中数组参数的使用问题。

关键词:FOXPRO 数组参数

数组作为一种十分有用的数据结构,FoxPro 予以了充分的支持,不仅提供了丰富的数组操作函数,还允许数组与数据库之间的数据传送以及使用数组参数来实现过程之间的数据传送。但是,FoxPro 的语法要求不象 PASCAL、C 等语言那样严密,因此在 FoxPro 程序中使用数组时,其正确性和合理性都要由用户程序自行负责。如不注意,极易出错。本人开始使用 FoxPro 编程时,就曾在这上面交过“学费”。由于到目前为止,所见到的各种关于 FoxPro 的著作和计算机期刊上还没发现有关数组使用问题的详细说明,现在就把自己的一点体会献给读者,以免重走我所走过的弯路。

一、COPY TO ARRAY 命令的二义性

COPY TO ARRAY <数组名> 命令如下两种不同的功能:

(1)执行此命令时,若<数组名>指定的数组不存在,则建立此数组并把当前数据库中若干记录的数据复制到数组中。所建数组的大小恰好能容纳要复制的数据。

(2)若执行此命令时<数组名>指定的数组已经存在,则依数组定义的方式又分有如下两种情况:

如果数组是由本命令前次执行时产生的或者是由 DIMENSION 语句定义的二维数组,则其作用与(1)类似,即把若干记录的数据复制到数组中。不同的是,要复制的数据个数多于数组元素数时,数组元素数不增加,多出的数据被忽略;要复制的数据个数少于数组元素数时,数组元素数也不减少,数组后面的元素保持其原来的值。

如果数组是由 SCATTER 命令执行时产生的或者是由 DIMENSION 语句定义的一维数组,则其作用与 SCATTER 命令类似,即把一个记录的

若干字段的数据复制到数组中。不同的是,此命令复制的是所给范围内符合所给条件的第一个记录(不一定是当前记录!)中的数据。

由上述可见,当希望将数据库中的数据复制到数组时,最好用 RELEASE 命令将数组清除后再复制(以保证数组中的数据恰好是我们所需要的)。若复制若干记录,使用 COPY TO ARRAY 命令;若复制某一个记录,则采用先定位,然后使用 SCATTER 命令的方法为好(因为使用 COPY TO ARRAY 命令完成此任务必须要保证数组已存在而且是一维的)。

二、数组参数不可按传值方式使用

过程调用时,如果把实参数组按传值方式使用,FoxPro 虽然不报告出错,但是它并不把实参数组各元素的值传送到过程。所以按传值方式使用的实参数组的存在也就失去了意义。程序 TEST1 的运行结果表明,DO SUB1 WITH (A)语句中实参数组 A 的存在是没有意义的。

```
* TEST1. PRG
SET TALK OFF
DIMENSION A(2)
A=1
DO SUB1 WITH (A) && 强迫传值
? A(1),A(2)
RETURN
PROCEDURE SUB1
PARAMETERS X
DIMENSION X(2)
? X(1),X(2)
X=2
RETURN .T.
```

执行程序 TEST1 显示的结果为:

.F. .F.

1 1

显然,实参数组 A 各元素的值未能传入过程 SUB1。所以应当注意,数组作过程参数时,切莫按传值方式调用。

三、过程可以改变实参数组的元素个数

在 FoxPro 语言中,过程中说明的形参数组的维数、元素数均可与调用语句中的实参数组不同。如果按传址方式调用,则当形参数组元素数少于实参数组元素数时,实参数组只有前面若干个元素的值被传入过程。返回调用过程时,实参数组只保留和过程中的形参数组一样多的元素,多余的元素被释放,而且其维数也变得和过程中的形参数组一样;当形参数组元素数多于实参数组的元素数时,实参数组的所有元素均被传入形参数组的前面若干个元素中,而形参数组后面的各元素被初始化为 .F.。返回调用过程时,实参数组的维数、元素数均和过程中的形参数组一样。由此可见,按传址方式使用数组参数调用过程时,过程可能对数组的维数、元素个数产生副作用。因此,使用数组作为过程参数时应特别小心,尽量避免这种副作用可能造成的程序错误。程序 TEST2 演示了形参数组和实参数组维数、元素个数不等时的情况。

* TEST2.PRG

SET TALK OFF

DIMENSION A1(2),A2(2)

STORE 1 TO A1(1),A2(1)

STORE 2 TO A1(2),A2(2)

DO SUB2 WITH A1,A2

? ' 返回后: '

DISPLAY MEMORY LIKE A?

RETURN

PROCEDURE SUB2

PARAMETERS X1,X2

DIMENSION X1(1),X2(2,2)

X2(2,1)=3

X2(2,2)=4

? ' 过程中: '

DISPLAY MEMORY LIKE X?

RETURN .T.

执行此程序显示的结果如下:

过程中:

X1 Priv @ A1

X2 Priv @ A2

返回后:

A1 Priv A

(1) N 1

A2 Priv A

(1 , 1) N 1

(1 , 2) N 2

(2 , 1) N 3

(2 , 2) N 4

可见,返回后数组 A1 的元素少了一个,数组 A2 的元素多了两个而且变为二维的了。所以过程中使用数组参数时,应特别当心这种副作用。

(本文收稿日期:1995.9.27)

Main Contents

- ② Outdoor full color LED screen for video display system
- ⑤ Multimedia Edit & Broadcast System for LED Screen
- ⑧ Multiple LED Display Screen Radio—paging System
- ⑪ Applying the Morphology approach to the Image Process of White Yarn Stripes Wound on Blackboard
- ⑮ One way of making system disk in NOVELL remote boot workstation
- ⑯ Cream drop of the source program for foxpro 2.5
- ⑲ The Example for Moving of Chinese Word
- ⑳ How to improve the informance of user local area network
- ㉕ Second development of accounting system For SSU—12 digital programming control customers exchanger
- ㉙ An Automatic Generating System of Current Open Pulldown Menu
- ㉛ Graphic surface design in CLIPPER
- ㉞ Input Method of Chinese Character for Application Software
- ㊱ Some Noticeable Problem in Using Array in Foxpro

· 代邮 ·

编辑部有
请作者王革、秦
旭、李侨生告知
新址及发往原
址的赠刊有无
收到。

· 预告 ·

本刊今年
最后两期将
以合刊出版。