

电脑与法律

著作权法对电脑技术提供的保护..... 王桂海 (2)

电脑应用

MACS 企业经营管理与控制系统

的开发与设计..... 钟恩光 (3)

什么叫 MRPII? 刘夏育 (10)

财务管理动态信息的设计与

实现 赖一介 (12)

NEW

电子声音信箱, 一种新型的通信

工具 詹 前 (13)

新型汉卡—星河汉卡 周光发等 (15)

软件介绍

数据加密工具—PC SECURE 张春强 (16)

工具软件极品 RAMinit 2.0 雷 军 (18)

网络与通信

两则行之有效的小改动 赖宏宇 (20)

高级 UNIX 连网实用技术讲座 冯家宁 (21)

IDEA

修改 DOS 命令名带来的妙处 周国均 (21)

防文件型病毒有新招 肖清雷 (22)

大学生之页

能使软盘翻修的小程序—800 杨海军 (23)

Windows 底层结构剖析 王传君 (24)

使用与维修

爱华 AH1488D 单色显示器行扭及

无显示故障的排除 赵明生 (26)

键盘的一般故障处理 刘 江 (27)

竞赛与考试

广东省将于 5 月 9 日举行计算机

操作人员水平联合考试 (27)

广东省青少年信息学竞赛信息 (68)

电脑辅助教学

标准化试卷的评判和程序 罗 君 (28)

中学天地

拷贝特大文件的方法 林志钢 (30)

编读往来

来信照登 (30)

投稿须知 (39)

把握时机施展才华实现理想 (65)

游戏乐园

游戏程序的消声方法 胡术等 (31)

病毒防治

Auto-copy 病毒的检测免疫与

消除 张 研 (33)

Tree 病毒 祝向权 (34)

单片机与单板机

高频教育电视有线网的信号监

测方案 叶惠文 (35)

一个大型可编程控制器电路

分析 徐巨善 (40)

电脑用户

西文状态下的汉字彩色显示 周苏峡 (44)

CGA 显示 25 行 CFOXBASE 2.10

的改进 彭 禾 (46)

怎样用好 CCED 的计算功能..... 王 彬 (47)

如何在软盘上使用 CCED3.0 ... 翁锦玉等 (47)

C 语言下拉式菜单程序的常见失

误及改进 王水等 (48)

“后台”演奏音乐 吴建林 (51)

CGA、EGA/VGA 卡间图形

数据转化 符云清等 (53)

VGA 图形信息的压缩存储和

恢复 郭新平 (54)

《面向十六位字节流的加密工具》改错

与改进 龙相明 (55)

鼠标, 你也能控制 冯志宏 (55)

利用 MSC5.0 对 CGA、EGA (VGA)

的图形画面进行存屏取屏的方法 ... 耿卫东 (63)

汉字放大显示工具库的发现 李 智 (66)

广稿索引 (68)

著作权法对电脑技术提供的保护

王桂海

在国内外对著作权法进行讨论的文章中,往往把它的保护对象分为三个不同的类型,即艺术性作品,纪实性作品和功能性作品。某些电脑技术成果也包括在这三个层次之中。但并不是三个类型的知识成果都由著作权法来提供保护。著作权所保护的,仅是能用某种特定形式表示出来的那一类成果,而且也仅仅保护其形式而不保护思想内容。作为保护知识成果这一类无形财产的专有权的法律,除著作权法外,还有专利法,商业法,商业法以相应的法律、条例、公约等等。美国国会技术评估办公室(OTA)曾就基于知识的产品和享有著作权的作品的关系,用一个图表示(图1),很有参考价值。

从这个图可以看到

1. 电脑有关的程序(系统软件和应用软件)、计算机数据库、微程序码段、指令集、计算机创作(作曲,作图)等等,都是受于著作权法所保护的
2. 计算机算法,计算机线路,半导体芯片掩膜,

都不属于版权保护的对象。至于它们应当获得什么样的法律保护,要根据具体情况去办理。因为版权可以由作品自然获得,专利权和商标专用权,则要经过严格的申请办理手续。

3. 与版权有关的计算机成果都划分在功能性作品范围,少数与艺术性作品有覆盖,也就是说,这些作品多数和艺术性作品不同,它们创作的自由度相对较少,形式上出现雷同的可能性较大。因此,在确定是否侵权时,要和艺术性作品有所区别。在这个图中尤其值得注意的是,计算机应用软件是放在相互覆盖的边沿,而系统软件则完成被看作是功能性作品,可见提供这个资料的作者作已经充分注意这两类软件在创作的自由度上区别。许多国家一方面把计算机软件看作和一般作品一样,纳入著作权法的保护范围,另一方面又注意到软件自身的特殊性,专门作出一些特殊的条例,也正是基于这种原因。 (50)

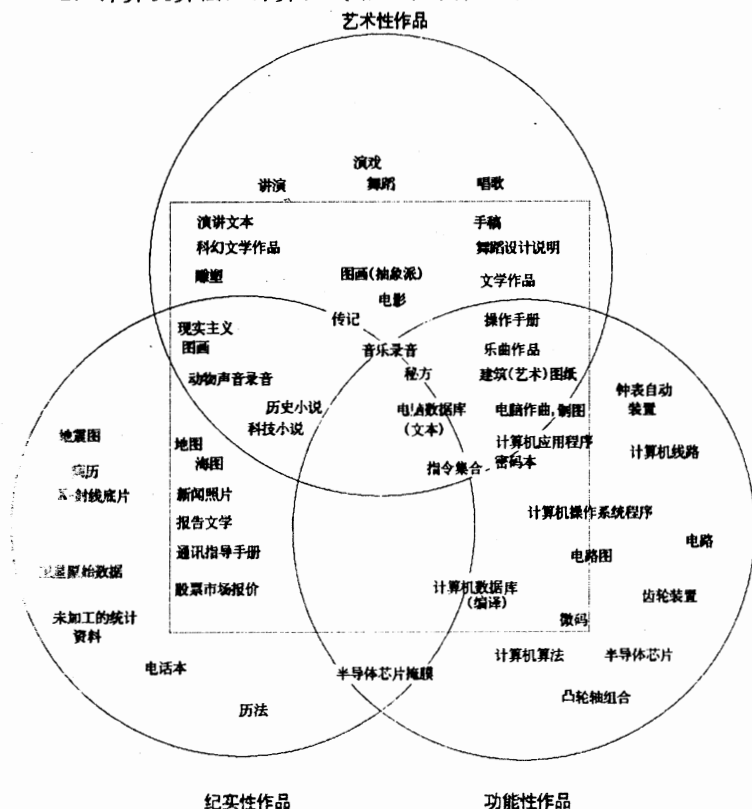


图1 基于知识的产品类型

在虚线表示的方形内,是属于享有版权的作品

资料来源: OTA: Intellectual

Property Rights.1986 年

主编: J.H.GIBBONS

MACS

企业经营管理与控制系统 的开发与设计

万宝电器集团公司 钟恩光

【内容提要】本文试图探讨当前推行市场经济的新形势下,企业计算机管理信息系统的开发如何向高层次发展,以及我国企业经营管理计算机应用系统的开发方向、管理模式,并结合由本人主持研究、开发的,在多个企业成功应用的“MACS 经营管理与控制系统”的实践来论述这种高级系统的设计技术及方法。

当前,中国的企业正面临着市场经济与“关贸总协定的严峻挑战”,同时企业正经历着一场深刻的变革,这个变革的重要特征就是企业正迅速地向着信息化、高科技、高生产力、高效益的方向发展。因此,从事计算机应用的科技人员如何迅速地研究、开发、设计出能使企业适应市场经济与“关贸总协定”的计算机管理系统已经是当务之急。

许多企业家已经意识到,在企业进步的各种机会中,最有希望的机会就是全面实现企业系统的计算机集成化、综合化的管理与控制。许多大中型企业为在激烈竞争中立于不败之地,为完成信息化、高科技、高生产力、高效益的变革,都不约而同的选择了以开发、建立一个高效能的企业计算机管理信息系统为突破口。

然而,目前国内的一些计算机应用系统,在其管理模式上,大多不是建立在市场经济和国际先进管理方法的基础之上,大多偏重于信息处理而忽略了经营管理的控制,在设计思想上,大多是传统的 MIS 系统,而很少或没有采用国际先进的 MRP II、JIT、BPCS 等包含着现代管理思想的方法。

MACS (Management and Control System) 即“企业经营管理与控制系统”是根据当今世界最为流行的 MRP II、BPCS 先进的计算机管理方式为基础的,很具特色的企业经营管理计算机系统软件包。它将计划、制造、采购、分销、库存、财务高度集成于一体,为企业提供一个完整的解决方案。经数个企业应用,确能有效地提高企业生产效率、降低成本、显著改善企业的管理,提高经营效益,使企业整个经营活动得到有效的控制,称它是现代企业经营管理的有效工具和得力帮手。

1 MACS 经营管理与控制系统开发的策略原则

集成化高级管理信息系统软件的开发策略涉及的四

面很多,可从各个角度去描述,且有许多成熟的理论和经验。这里仅从与传统认识有所不同的、而又认为是最基本的三个方面去研究,一是软件开发(应用)的基点,是要计算机管理软件去适应企业现行管理,还是企业现行管理去适应先进的管理软件;二是目前开发管理软件的关键技术,是计算机技术还是管理技术;三是系统实施(应用)的主导力量,是计算机专业人员还是各级管理人员。对于第一个问题,通常的做法都是要求软件的开发去适应并满足本企业的管理现状;对于第二个问题,一般是注重于计算机技术,而严重忽视管理技术,甚至让不懂或完全不懂管理的计算机人员来承担开发任务;而对于第三个问题,许多企业都把管理信息系统实施的主导力量推到计算机专业人士身上。实际上,上述三种认识都是片面的,而且这正是当前企业管理信息系统水平低,效益差的关键所在。笔者认为:

1.1 管理软件开发策略的基点应是企业的现行管理必须要适应先进的管理软件

计算机企业管理信息系统传统的开发原则,都十分强调按用户的需求来进行系统设计,要求计算机软件适应本企业的管理现状,这样开发出的软件只能是现有管理系统的翻版,因此,设计出来的系统水平不高,不能从根本上提高企业管理的水平。我们知道,劳动生产率、管理水平落后,人员素质差是我国制造业普遍存在的严重问题,采用计算机技术对我国企业进行技术改造是提高我国生产力水平的根本出路。如果要计算机应用去适应落后的企业管理现状,那么就会出现一种畸形的现象,即用先进的计算机硬件,采用低水平的软件去控制落后的管理水平的企业。

由此可知,进入信息时代的今日,必须改变那种强调软件要适应企业管理现状的传统原则,而代之以强调软件按先进管理来设计,企业管理要适应软件的要求的

新观念。这就是说,企业管理信息系统的开发策略,必须超越企业落后管理水平的现状,按先进的管理理论,按现代化管理的方法和管理程序去开发高水平的软件,必须按先进企业管理经验的共性和精华设计出一个标准管理软件系统,然后要求企业的管理纳入该标准,按它来实施企业的管理,只有这样才能达到利用先进的计算机硬件、科学的管理软件,去提高企业的管理水平。

实际上,目前国外先进成熟的管理软件一般都是按国际先进的普遍适用的管理理论和管理经验的共性,并严格按某种标准设计开发的,这种系统与企业管理的具体方式无关、与企业结构无关、与产品结构无关。因此,管理软件的开发必须注重采用国际先进的管理概念、管理理论和现代化管理方法,然后,要求企业遵循先进管理软件中规定的管理方法去改造、完善、提高我们的管理,从而全面地提高企业的素质,实现企业高生产力、高效益、高科技的变革。只有这样,才能逼着我们去下决心去改变落后的管理现状,真正达到用先进的计算机去管理先进的企业的境地。

1.2 开发应用软件的技术关键首先是管理技术,其次是计算机技术

在计算机选型及开发环境(硬件、语言、通信等)确定以后,应用软件的开发的水平及其适用性的关键技术就在于管理技术,计算机软件开发人员懂不懂管理,尤其是懂不懂现代管理理论、管理技术、先进的管理方法(如MRP II、JIT、CIMS、BPCS)就成为软件开发成败的关键因素。

我们都知道,企业管理信息系统的开发具有很强的综合性,它不仅仅是计算机软、硬件系统的一体化,而且是一个软技术(包括思想、观念、方法、组织等)和硬技术(包括生产、工艺等)的集成过程,它是一个综合运用系统论、信息论、控制论、行为科学、管理科学、计算机及通信技术的过程,而首先是系统论、控制论、管理科学起主导作用。

企业管理信息系统的开发又具有整体性,系统开发决不是仅限于企业内部某个单项管理业务或职能子系统的开发,也不是单项应用或子系统开发的简单组合,而是根据不同管理职能之间的内在规律,把企业作为一个有机整体,以信息系统为主线,从提高企业管理的整体功能出发对系统进行开发,如没有很强的管理技术来支撑,系统亦难于成功。

由此可见,如果,计算机专业人员对管理技术了解甚少,不仅工作困难很大,而且开发出来的系统,即使计算机编程技巧可能很高明,但在管理技术方面就不可能有创新。这在国际上也是有沉痛教训的。美国企业采用计算机的初期,也出现过一个方向性错误,即过高地

评价计算机专业知识的重要性,崇尚计算机技巧,而忽视经营管理理论,因此,使美国的管理信息系统的开发走了一段弯路。

因此,计算机开发人员当前的一个重要任务就是要认真学习先进的管理理论、管理方法,认真研究如何将计算机技术和管理技术完美地结合。

1.3 计算机管理系统的实施和应用成败的关键在于管理人员而不是计算机专业人员

对于计算机管理系统实施中谁是占主导地位的问题,传统的情况都是以计算机专业人员为主,重担主要落在计算机应用人员上,领导和管理人员很少去关心这个问题。这种情况与计算机应用的实践效果来看是恰恰相反的,事实证明没有管理人员参与的管理信息系统一定是失败的系统。

我们都知道,企业计算机管理信息系统是支持企业生产经营总目标的实现,其重要任务就是要在规定的时间内把正确的信息提供给各层管理人员及生产人员手中,实现对企业内外部信息的动态管理,因此,企业管理信息系统的实施是涉及企业内部各个方面的一场改革,上述任务的实现主要都是由领导者、管理者来完成的,而计算机只是一种先进的手段,计算机专业人员是无法代替的。因此,能否真正按先进软件的管理方式去改善现行的管理模式,落实到实处,也主要取决于管理人员。

大量事实都证明,企业领导及管理人员在企业管理信息系统的实施中的贡献比例是系统成败的关键所在,必须树立“第一把手”原则。

2 企业经营管理与控制系统开发目标

确定企业管理信息系统的开发目标,是决定该系统的结构、功能、应用范围及企业经营运作、管理方式的重要问题。对大中型企业来说,管理信息系统的开发目标必须是显著提高企业的质量效益、经济效益、工作效益和管理效益,实现管理与控制相结合的高效能、综合化、集成化目标。

2.1 系统必须能显著改善企业的管理水平,改变企业的经营方式,促使企业管理体制的改革,适应市场经济。

MACS 不仅应是一个技术先进、结构合理、功能齐全、使用灵活、方便、可靠的计算机软件,更重要的是 MACS 应是一种先进管理方式,这种方式比软件本身更为重要,它的成功之处应在于能使企业的工作方式得到重大改变,有效促使管理体制的改革,使企业从计算机管理系统中直接得到效益。

MACS 系统应是建立在一个先进的、符合国际惯

例的、具有一系列现代化方法的管理模式上,是以先进管理为核心的、管理技术与计算机技术完美结合的成果。

只有这样才能使管理信息系统投入运行后,能使企业经营管理方法、管理水平有一个质的变化,以计算机应用、软件应用来促企业运行机制的变革,以软件应用来理顺企业的管理关系,通过把企业管理各方面如生产、销售、财务等全面地纳入计算机的严密控制与管理上,使企业整个管理上一个新台阶,全面提高企业的管理效益,从而提高企业的经济效益。

2.2 系统必须能显著提高企业的经济效益和质量效益

过去一段时间,由于管理信息系统的低水平重复开发,其应用效果主要表现在减轻管理人员的工作强度方面,被一些人讥讽为“报表系统”,而在经济效益方面,只能从间接效益方面来体现,由于这个原因,长时间来企业的高层及中层领导很难认识到计算机管理信息对他们的极端重要性,因此,当前特别要研究管理信息系统如何直接提高企业经济效益的问题。

笔者认为,可从以下方面来考虑:

① MACS 系统应能将经营管理的各环节的高度集成化处理、向实时化方向发展,强化“第一线”原则与“第一时间”原理,即所有数据从经营管理“第一线”采集,从原始数据开始实时地、多系统、多模块的联机、联网处理,又实时地将所有信息“第一时间”反馈给“第一线”,并实时地控制“第一线”的经营活动。

这就要求 MACS 管理信息系统能管理到科室、车间、班组一层,能直接从生产第一线、经营第一线、仓库第一线、销售第一线、成本第一线上采集数据,从而全面提高整个计算机管理系统的信息服务的及时性、准确性和可靠性,使系统能实时地“第一时间”采集生产、经营、管理第一线的数据,能实时反馈信息给第一线,并实时控制第一线。因此,“第一线”原则是直接提高经济效益的关键。

② 管理信息系统必须向管理与控制一体化方向发展,强化“控制”功能。当前,管理信息系统对经营管理过程的控制功能的多寡已成为管理信息系统水平高低的重要标志之一,这也是提高管理信息系统经济效益的关键技术。这种系统必须具有对经营管理过程的各主要关键业务具有实时的有效的控制功能,这种控制是由管理信息通过信息收集、处理,并与其它环节比较后,再用信息系统去控制经营管理过程,只有把计算机用于管理和控制有机结合起来,才能使管理工作上升到更加科学的水平,从而有利于提高产品质量和降低消耗,提高经济效益。此外,计算机管理控制一体化,不仅能做到局部优化控制,还能做到全过程的优化控制,使企业生

产、经营、销售各方面都处在信息化良性运转状态,从而使企业从分散、独立的传统管理模式从根本上提高到一个整体和系统的水平。

2.3 系统必须向实现经营管理与生产控制的集成化、综合化方向发展

当今时代的企业,从产品设计、生产制造到市场销售的整个经营活动已经成为一个紧密相联的有机体,任何一个环节失控,都会造成企业巨大损失,正是由于这个原因,计算机应用也已经渗透到本来是各自独立,互不联系的计算机应用的各个领域。

因此,根据当代企业生产经营管理的整体性和计算机应用的多分支性,当代信息系统的开发目标应明确它的集成化、一体化功能的发展方向,这种集成化思想的集中点就是将管理与控制,经营与生产,前台与后台集成在一个大系统中统一管理,这就是世界上一些发达国家相继出现的“计算机综合制造系统(CIM)”,“企业自动化系统(OA)”以及“企业计划与控制系统(BPCS)”

由上述可知,确立系统集成化目标除了能使更多部门交换更多的数据,能全面考虑影响最佳实现功能的所有因素,进行完整的信息加工外,它是对企业经营机制、运行体制、组织机构及管理方法的重大变革,使整个企业发生根本的变化。

3 企业经营管理与控制系统的设计要求

3.1 在系统设计上采用模块化、标准化、网络化技术,程序适应性广,可改性强、维护性好。屏幕格式、操作方法、代码方案、数据结构都采用一致的标准,具有一致的风格。

3.2 人机界面友好,联机帮助功能强,在各种查询、录入、修改等功能下,均可得到明确的帮助信息和提示信息,在有限的屏幕中,给出多库、多业务的相关信息,在所有运行过程中,系统的运行状态、出错信息、操作提示等都不断地给出用户,系统透明度高。

3.3 灵活性强,MACS 系统既可运行于微机网络多用户环境,又可单机运行,各功能模块的信息既可联机、多模块、交叉实时地动态处理,又可可不联机、进行信息的集中处理。

3.4 具有良好的数据安全性和信息保密性。各子系统、各模块、各种功能的使用权限均可通过“系统安全管理模块”预先设置,能实现企业各类信息的保密和数据的安全。

4 MACS 的设计技术

4.1 系统资源与系统功能的统一管理

由于企业经营管理与控制系统是一个相当庞大而又复杂的管理系统,因此,整个系统极其系统本身的管理

就显得尤为重要。本系统是采用统一管理的方式,主要是:

4.1.1 属于系统级统一管理的功能

4.1.1.1 系统配置管理

① 系统运行环境设置:如,公司名称;是否网络或单机运行等;

② 系统运行参数设置:如,有关模块是否联机运行,出错信息是否写入黑匣子等;

③ 系统公共变量管理:凡属系统级的全局变量均统一地一次性设置;上述管理,均在一个称为配置文件的程序(CONFIG.PRГ)中设置。

4.1.1.2 系统路径管理

① 主路径管理:对于不同的用户或同一用户的不同部门系统主文件的运行路径可能是不同的,只需改动路径文件便能适应多种情况;

② 模块运行路径管理:对于各模块及其程序运行路径的设置,形成系统的目录树;上述管理均在一个称为路径管理文件的程序(PATHGL.PRГ)中实现。

4.1.1.3 系统安全性管理 所有的系统安全、数据安全、信息保密、用户权限等均在系统安全管理模块中统一管理和实施。

① 用户登记管理:所有访问本系统的用户均须事先进行用户登记,进行用户等级分级,分配或设置用户口令。

② 用户权限管理:所有已登记的合法用户对不同模块、不同功能所拥有的权限的定义和设置,如,访问权、查询权、录入权、修改权、删除权、编辑权等。

4.1.1.4 系统标准管理

对全系统的信息代码、信息分类、运行提示、操作帮助等均在系统标准管理模块中统一管理,使全系统具有一致的标准。

① 全系统代码管理:全系统各模块的各类信息的代码均在一个具有同一结构的数据库(BZDM.DBF)中存放,均在同一管理程序中增加、定义、修改,全系统信息代码均在单一的代码库中调用;

② 全系统提示管理:对于各模块的录入提示、修改提示、查询提示、及其它提示均在同一提示管理程序(BZDMS.PRГ)中实现。

③ 全系统帮助管理:对于各模块的使用说明、帮助内容等均在一个帮助管理文件中实现。

4.1.2 属于子系统级统一管理的功能

4.2.2.1 文件管理(WJGL.PRГ) 各子系统内运行文件的定义、调度、调用、备份、删除等管理均在一个子系统的文件管理文件统一管理及实现。

4.1.2.2 维护管理(WH.PRГ) 各子系统的维护,

如,数据库索引、数据库建立、文件建立等均在各自子系统的系统维护文件中统一管理及实现。

4.1.2.3 屏幕显示格式管理(FMT.PRГ) 对于各模块、各功能的屏幕显示格式均在同一文件中定义与描述,在统一文件中管理,以统一方式调用。

4.1.2.4 记录显示格式管理(FGT.PRГ) 对于各模块、各功能的记录显示格式均在同一文件中定义与描述,在统一文件中管理,以统一方式调用。

4.2 系统的模块化、标准化设计

系统模块化设计是使系统具有可维护性、可修改性,灵活性大,适应性强的一项关键技术。

4.2.1 通用功能程序管理 所有的公用功能程序均在一个通用过程文件中管理(TYPRO.PRГ),提供给其它所有模块调用,主要通用过程程序有:

① 公共变量管理程序;

② 全屏幕记录显示程序;

③ 屏幕卷动、记录移动管理程序,即,光标移动键的管理

④ 键盘操作按键管理:显示按键、判断有效键、等待按键等

⑤ 信息提示管理:提供标准提示格式,提供信息显示格式,提供子菜单显示,提供标准警告、响铃提示等。

⑥ 日期时间管理:提供日历显示、时间显示、计算日期时间,提供日期输入格式并判断其合法性,等。

⑦ 通用打印程序:提供各类打印格式、打印控制、打印参数设置。

⑧ 通用窗口管理:提供各类窗口格式、窗口显示、屏幕刷新等。

⑨ 其它通用功能。

4.2.2 按处理功能划分模块,集中管理,如

① 文件管理(WJGL.PRГ) 所有运行文件的集中管理,定义、调度

② 维护管理(WH.PRГ) 所有子系统的维护集中管理,库索引、库建立

③ 屏幕显示格式管理(FMT.PRГ) 所有屏幕显示格式集中管理

④ 记录显示格式管理(FGT.PRГ) 所有记录显示格式集中管理

4.2.3 标准处理程序的设计

① 标准查询程序 提供各模块分类查询、组合查询、组配查询;

② 标准打印程序 提供各模块打印格式、打印方式、打印设置;

③ 标准统计程序 提供各模块统计功能、统计期

设置

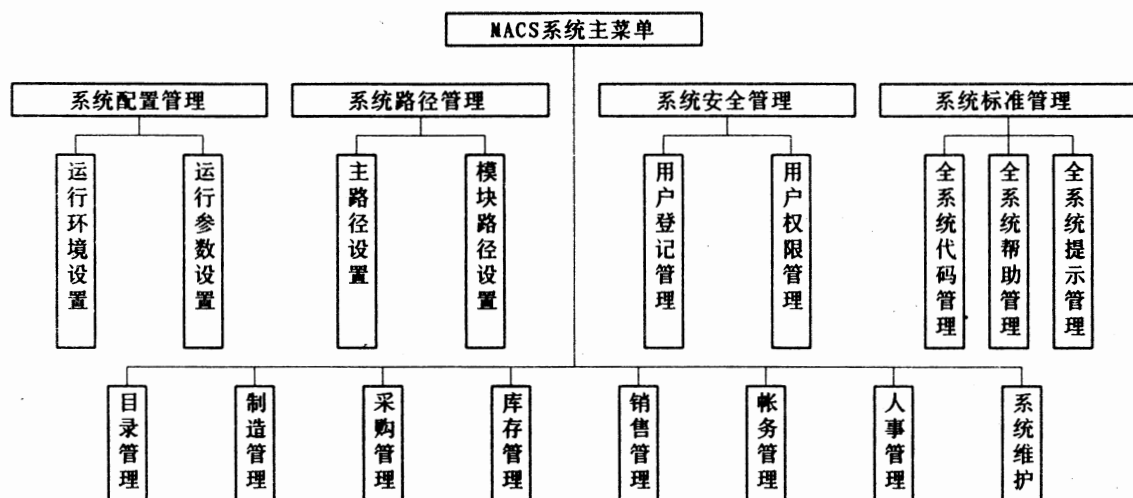
④ 标准窗口程序：提供各模块各类窗口格式、窗口生成等；

上述程序，实际上是一个半自动程序生成器，各模块仅须稍作改动便能生成应用程序，可以大大节约开发时间，并保证格式与标准的一致性。

5 MACS 的结构设计

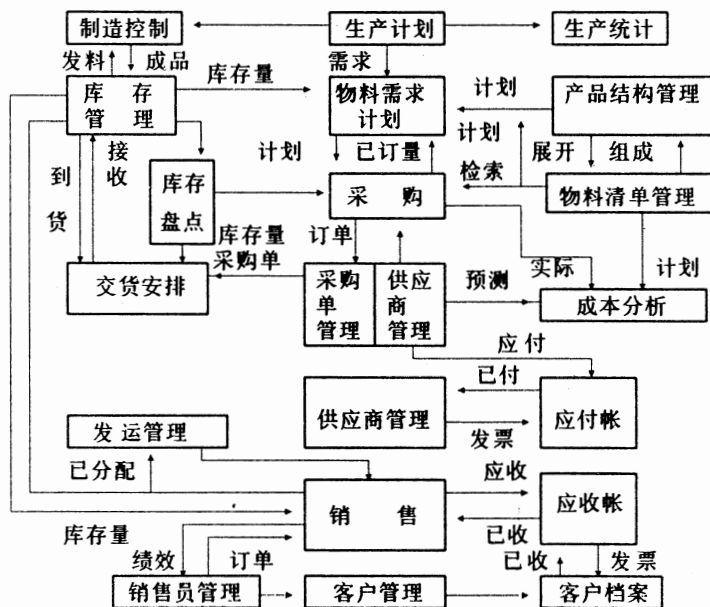
5.1 MACS 的系统结构

MACS的系统结构



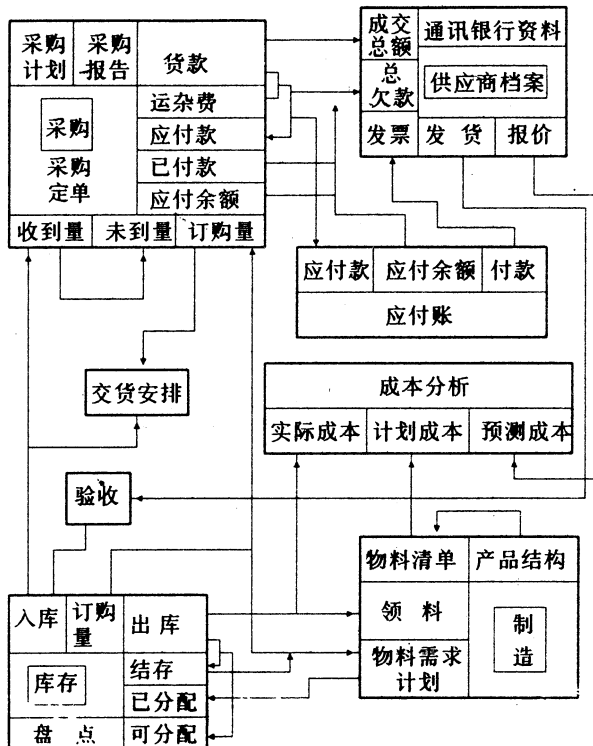
5.2 MACS 的功能结构:

MACS功能结构

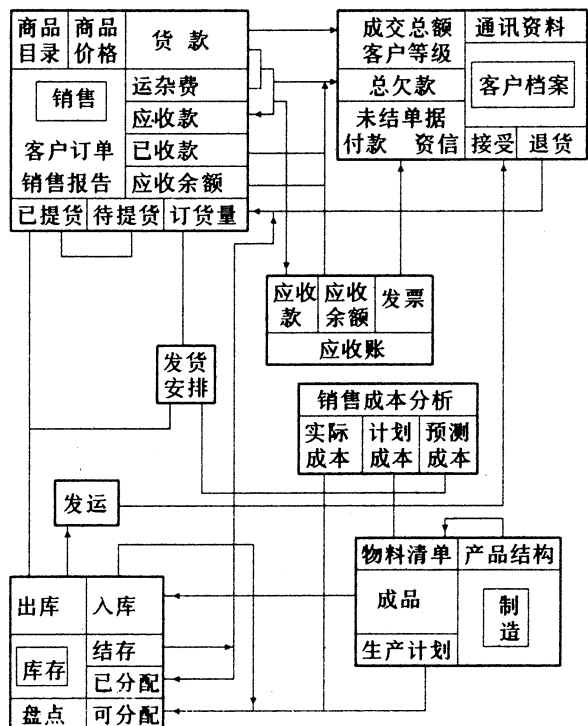


5.3 MACS 多模块信息交叉联机处理

MACS ‘采购- 制造- 库存- 应付账’ 子系统



MACS ‘销售- 库存- 应收账’ 子系统



5.4 数据传递、信息处理

开出销售单

- 库存信息:
 - “出库数” 不变
 - “库存数” 不变
 - 加“已分配数”=销售数量
 - 减少“可分配数”=销售数量
- 库存状态: $(+, 0, -)/(+, 0, -)$
- 账务信息: 增加“应收款余额”=货款
- 提货信息: 增加“欠货款数量”=销售数量
- 客户信息:
 - 增加“应收款总额”=货款
 - 增加“未结单据数”=+1
 - 增加“欠货单据数”=+1
- 客户状态: $(+, 0, -)/(+, 0, -)$
- 单据状态: 置为 +/+ , M1/S0/C0

提货单录入

- 存信息:
 - 增加“出库数”=提货数
 - 减少“库存数”=提货数
 - “已分配数” 不变
 - “可分配数” 不变
- 存状态: $(+, 0, -)/(+, 0, -)$
- 务信息: 不变 “应收款总额”
- 货信息: 减少“欠货款数量”=提货数量
- 户信息:
 - 不变“应收款总额”
 - 不变“未结单据数”
 - “欠货单据数”=+1, -1, 0
- 户状态: $(+, 0, -)/(+, 0, -)$
- 据状态: 置为 +/(+, 0, -) , M1/S0/C1

录入收款单

- 库存信息:
 - “出库数” 不变
 - “库存数” 不变
 - “已分配数” 不变
 - “可分配数” 不变
- 库存状态: $(+, 0, -)/(+, 0, -)$
- 账务信息: 减少“应收款余额”=收款数
- 提货信息: “欠货款数量” 不变
- 客户信息:
 - 减少“应收款总额”=付款数
 - 增加“未结单据数”=+1/0/-
 - “欠货单据数” 不变
- 客户状态: $(+, 0, -)/(+, 0, -)$
- 单据状态: 置为 (+, 0, -)/+ , M1/S0/C0

6 MACS 的功能设计

● 产品结构

- (1) 增加 / 修改 / 删除产品部件
- (2) 生成 / 修改 / 删除产品结构
- (3) 生成 / 查询 / 复制产品部件清单
- (4) 生成 / 查询 / 复制产品物料清单
- (5) 生成 / 查询 / 复制产品结构清单
- (6) 打印产品结构清单 / 部件清单
- (7) 核算 / 统计 / 查询产品组成(重量,用量)
- (8) 核算 / 统计 / 查询部件组成(重量,用量)
- (9) 核算 / 统计 / 查询部件组成(消耗)
- (10) 产品部件分类查询
- (11) 产品部件合并
- (12) 产品结构成批替换

● 物料清单

- (1) 增加 / 修改 / 删除物料
- (2) 生成 / 修改 / 删除物料清单
- (3) 生成 / 查询 / 复制部件物料清单
- (4) 生成 / 查询 / 复制产品物料清单
- (5) 打印物料清单 / 物料参数清单
- (6) 核算 / 统计 / 查询物料(重量,用量)
- (7) 核算 / 统计 / 查询部件组成(重量,用量)
- (8) 核算 / 统计 / 查询物料消耗定额
- (9) 产品、部件、物料分类查询
- (10) 物料清单成批替换

● 制造管理

- 生产计划
- (1) 输入生产计划
 - (2) 修改生产计划
 - (3) 批准生产计划
 - (4) 处理生产计划
 - (5) 取消生产计划
 - (6) 打印生产计划
- 物料需求计划
- (7) 生成 / 修改物料需求计划
 - (8) 编辑 / 打印物料需求计划
 - (9) 批准 / 确认物料需求计划
 - (10) 生产计划缺料报告
 - (11) 预计缺料报告
 - (12) 打印物料情况报告(提料前)

● 采购

- 采购管理
- (1) 采购订单管理
 - (2) 物品采购管理

MACS 系统的主要管理功能

- (3) 交货安排管理
- (4) 供应商报价单管理
- (5) 采购需求报告
- (6) 采购计划管理
- 供应商管理
- (7) 供应商通讯资料
- (8) 供应商银行资料
- (9) 供应商往来资料
- (10) 供应商订单资料
- (11) 供应商编码管理
- (12) 供应商档案记录

● 销售

- 销售管理
- (1) 销售订单管理
 - (2) 商品销售管理
 - (3) 交货安排管理
 - (4) 商品发运管理
 - (5) 商品目录管理
 - (6) 商品价格管理
 - (7) 销售计划管理
- 客户管理
- (9) 客户通讯资料
 - (10) 客户银行资料
 - (11) 客户往来资料
 - (12) 客户订单资料
 - (13) 客户编码管理
 - (14) 客户档案记录
 - (15) 客户报告
- 销售报表
- (1) 客户报表
 - (2) 销售价格报表
 - (3) 未结订单报表
 - (4) 发运报表
 - (5) 未完全发货订单报表
 - (6) 待发订单报表
 - (7) 打印销售订单报表
 - (8) 打印商品销售报表
 - (9) 打印销售统计报表
 - (10) 打印商品销售统计报表
 - (11) 销售成果综合报表

- (3) 物品已分配 / 可分配 / 结存
- (4) 物品 / 物料超欠储
- (5) 物品 / 物料 ABC 分类
- (6) 物品 / 物料盘点
- 库存查询
- (7) 出入库单据查询
- (8) 库存日报表查询
- (9) 超欠储商品查询
- (10) 盘点情况查询
- (11) 库存分析查询
- (12) 库存报警查询
- 库存报告
- (1) 打印物料接收、检查报告
- (2) 库存物料报告
- (3) 物料传输报告
- (4) 物料信息报告
- (5) 物料数目报告(概括)
- (6) 物料数目报告(详细)
- (7) 打印盘点表
- (8) 打印库存日报表
- (9) 打印各部门综合库存表
- (10) 打印库存分析表
- (11) 打印欠储报警表
- (12) 打印超储报警表

● 账务

- 应收账
- (1) 科目管理
 - (2) 凭证录入
 - (3) 账务生成
 - (4) 财务核算
 - (5) 催款管理
 - (6) 应收报告
- 应付账
- (1) 科目管理
 - (2) 凭证录入
 - (3) 账务生成
 - (4) 财务核算
 - (5) 欠款管理
 - (6) 应付报告
- 成本
- (1) 材料成本
 - (2) 单位成本分析
 - (3) 总成本分析
 - (4) 成本报告
- * 财务报告

什么叫MRP II?

郑州市第四棉纺织厂计算机室 刘夏育

自大工业社会化的产业革命以来,企业的生产管理问题日益重要。企业的生存和发展不再单单依赖于工艺设备技术水平的提高,而更重要的是管理水平的提高。按照一般的归纳,一个企业管理上比较突出的有以下几点:

- ①原材料的及时供应
- ②零部件的准确配套
- ③库存积压
- ④资金周转期

上述几点集中起来就是“对物料需求的控制”。早期的管理专家曾提出过所谓“Order Point Method”——即按照过去的库存经验来预测未来的物料需求。在此基础上就有了MRP (Material Requirement Planning 物料需求计划)。这种方法的执行需要大量繁杂的计算。当个人计算机普及后,这种方法得到了迅速发展。MRP的最终目标可以用日本丰田公司的“Just In Time”(刚好及时)原则来形容。到了八十年代,MRP兴起了“计划→实施→评价→反馈→再计划”的闭环模式。在这以后,MRP的含义逐渐演变为Manufacturing Resources Planning (制造资源计划),称为MRP-II以示区别。这里的制造资源不仅限于物料,还囊括了资金、机器设备、能源、人力、时间等资源,从而实现了对于生产经营活动的全面管理。

在信息产业比较发达的国家,MRP-II的开发与应用已有十几年的历史。对于中小型计算机的制造厂商来说,MRP-II已成为支撑市场占有率的必备软件包之一。根据美国生产与库存控制协会(APICS)的统计,使用MRP-II之后企业的库存周转率可望提高30%,准时交货率提高25%,提前交货率提高15%,采购提前期缩短50%,等工待料减少60%,制造成本降低12%,管理人员减少10%。

改革开放以来,我国一些大型企业相继引进了国外的MRP-II软件产品。还有一些企业自行开发与与科研单位联合开发了自己的信息管理系统。就引进来说,大多是国外70年代产品,软件开发环境落后,有的是文件系统,有的是网状数据库,有的甚至使用机型专用语言。另外还由于管理习惯、市场机制、所有制环境的不同,这些引进软件只有大约三分之一可以照搬使用,其余的都要经过大量艰苦的修改,有些部分根本不能使用。在发达的资本主义国家,生产的社会化、专业化水平很高。而我们诸如铸造、锻造、机械加工、热处理、电镀、装配等等所有的工序往往要在一个企业内完成。在原材料和外购件上,外国企业可以将采购期定得很短。而我们的原材料和一些关键配套产品长期短缺,处于卖方市场。有些主要生产资料如有色金属、钢材、煤炭、石化原料等为了保障供给,甚至由国家统一采取半年定货的方法。因此国外的MRP-II不仅产生自制件的计划,同时产生采购计划。而我们却要物资供应计划专门设置一个模块。另外,我们的全民所有制对企业有一系列特定的政策规章、财务成本税收的核算办法。这都是中国企业管理所特有的,它决定了我们应该在吸收外国先进经验的基础上,根据自身实际进行MRP-II开发。

MRP-II中的一些基本概念:

- 1、Material—原材料、原器件、零部件、成品或半成品、自制或外购件等。
- 2、Independent Demand Items—不依赖企业内其他需求项的需求。
- 3、Dependent Demand Items—由其他需求项决定的需求。
- 4、Gross Requirement—未扣除在库量时的物料需求量。
- 5、Net Requirement—扣除在库量和在途量以后的物料需求量。
- 6、On-Order Quantity—已采购未到货或已加工未完成的物料数量。
- 7、Allocated Quantity—库存中已被分配用途的物料量。
- 8、Planned Order—作为需求的订单。
- 9、Firm Planned Order—经过确认的计划订单。
- 10、Horizon Days—在规定的采购或加工期限内,除去必不可少的时间外可供调节的天数。
- 11、Release Date—将计划订单、经过确认的计划订单或采购单/加工令指定成正式采购单/加工令的日期。
- 12、Due Date—采购指定的货物购回日期或加工令指定的生产入库日期。

一个MRP-II的功能结构一般应包括:制造工程、市场销售、物料管理、计划管理、生产控制和财务管理等子系统。

制造工程是整个MRP-II的基础部分,其实质是制造数据库的维护。它定义和维护生产对象(如原料、零部件、最终产品以及工具设备等)、产品构成、制造过程、能力资源等等。它可以分为“产品结构”、“工艺/工作中心”两个部分。同时它还向其他应用系统提供必要的数据。

MACS 经营管理与控制系统

- 专为企业适应市场经济而最新设计的计算机系统软件包
- 设计论文获得广东省计算机用户协会 92 年会优秀论文
- 最适应工业企业、商业企业等进行计算机企业经营管理
- 可在微机单机、微机网络或 AS/400 小型计算机可靠运行

MACS(Management and Control System) 是根据当今世界最为流行的 MRPII / BPCS 先进的计算机管理方式设计, 最具特色的企业经营管理计算机系统软件包。它将计划、制造、采购、分销、库存、财务高度集成于一体, 为企业提供一套完整的解决方案。使用该软件能有效地提高企业生产效率、降低成本, 显著改善企业的管理, 提高经营效益, 使企业整个经营活动得到有效的控制, 是现代企业经营管理的有效工具和得力帮

手。

MACS 不仅是一个技术先进、结构合理、功能齐全、使用灵活、方便、可靠的计算机软件, 更重要的是 MACS 是一种先进管理方式, 这种方式比软件本身更为重要, 它的成功之处在于使您的企业工作方式得到重大改变, 有效促使管理体制的改革, 使企业从计算机管理系统中直接得到效益。

成果推广与交流 针对性开发 企业管理计算机应用咨询

广东省计算机用户协会 AS/400 专业委员会

联系人: 钟恩光 地址: 广州江燕路 288 号

邮编: 510280 电话: 4419977

《电脑》杂志社

地址: 广州石牌华南师范大学内

邮编: 510631 电话: 5516911-3273

市场销售部分的主要功能是向销售人员提供市场信息、客户情况、合同状态等, 帮助销售人员做好合同管理, 掌握市场动态, 提高工作效益。

物料管理可由“库存控制”、“采购管理”两个部分组成。目标是控制库存水平, 反映库状态, 加速库存周转, 保证供应, 降低采购费用。

计划管理向企业各个管理层提供辅助计划工具。可分为“需求预测”、“主生产计划”、“物料需求计划”、“作业和能力计划”四个部分。

生产控制可分为“生产订单控制”和“车间控制”两个部分。前者是生产投入的控制, 后者是生产过程的控制。

财务管理可分为成本、应收帐、应付帐、总帐四个部分。系统将提供多种成本模拟功能, 标准成本和实际成本的比较, 实际成本跟踪等。在制造、销售、采购、库存等活动中产生相应的财务更新总帐, 确保财务信息能够及时准确地反映实际生产和经营状况。

在建立 MRP-II 前, 企业必须具备一些基本条件: 应有一份主计划生产表和正确的产品结构表, 物料的统一编码、库存及产供销信息, 每个加工项目的加工过程等等。

执行 MRP-II 的基本过程:

1. 建立完善的数据库资料。包括①物料基本资料(物料编码、名称、类别、单位、单价、库存量、在途量、已分配量、采购或加工提前期、安全库存量、采购/加工的标准批量、标准成本等)。②产品结构资料。③加工过程和加工中心的资料。④预测数据资料。⑤客户及供应商的资料。等等。

2. 提供完整的需求资料。包括物料/产品的预测需求和未完客户订单的需求量。

3. 将采购单/加工令更新为标准形式。

4. 执行主生产计划。提出需求计划和建议信息, 产生最新的生产方案。

5. 对新的生产方案进行生产能力的模拟分析。

6. 根据模拟结果调整生产方案, 再模拟直至满足要求。

7. 由生产方案得到的资源需求计划, 产生各类需求计划单和建议信息。

MRP-II 是一种先进的企业管理辅助工具。随着改革开放热潮的再次兴起, 我国的市场机制逐步得到建立和完善。以往那些不讲效率、不思进取的企业将失去生存条件。积极采用计算机技术来提高企业的管理水平已经成为大势所趋。

财务管理动态信息的设计与实现

华南理工大学 赖一介

【摘要】本文介绍了计算机管理信息系统中针对财务管理中工资数据的多变性而设计的方案与实现,从而动态地处理了工资数据管理、计算、统计、检索、查询等工作。

一. 引言

在当今改革开放的时期,随着四个现代化发展的需要,企、事业单位中的计算机管理越来越普遍。在财务方面的计算机管理信息的应用,给财务部门的工作效率、工作质量带来了很大的提高。

然而,财务的数据管理,有其特殊的地方,就是多变性。一个计算机管理系统建成投入使用后,往往由于许多外部原因,例如:上级新的政策的下达,新的规定、新的要求的实施等等,最初设计的程序为适应新的要求而不断修改,甚至连某些关键的数据库的结构也要修改,从而造成了大量的维护工作,对于一些计算机管理水平不太高、维护力量不强的单位,就感到了许多不便之处。

因此,在计算机管理系统的开发研制中,设计好一个便于使用,便于维护的系统,将会大大增强企、事业单位推广使用计算机管理的信心,提高企、事业单位的管理水平。

本人就是基于上述观点,在财务管理中,对使用得比较早,应用得比较普遍的工资管理方面,进行了一些新的设计,以求适应工资数据不断变化,程序及数据结构不断修改,维护工作量大的特点,满足用户的需要。

二. 设计思想及实现

通常的工资管理模块,最关键的数据文件是基本工资数据文件。(一个或多个),这个数据文件通常包括三个部分:

1. 该单位各类人员的工资识别信息。例如:姓名、人员编号、部门等等。
2. 工资的应发细项。例如:基本工资、工龄工资、浮动工资、各种补贴、各种特殊津贴等等。
3. 工资的扣除细项。例如:水电费、房租、电话费等等。

以往的做法是:该文件根据用户的要求直接设置,文件的字段名用英文字母或中文表示,确定了该子段的数据内容。如下:

		基本工资数据文件				扣除细项	
		应发细项					
		M 项				N 项	
姓名	编号	部门	基本工资	工龄工资	浮动工资...	水电	房租...
张三	A0101	工会	204.00	10.00	30.00 ...	22.50	9.50...
李四	B0001	长办	124.50	7.00	20.00 ...	15.30	4.00...

该文件一旦确立,许多围绕着该文件的各种计算、统计汇总、数据管理、打印报表等工作就可以展开。

如果工资的应发细项或扣除细项发生变化,增加、减少或从新组合,或数据宽度超出原设计宽度,基本工资数据文件的结构就要修改,从而一系列的程序也要做相应的修改,给工资的计算机管理维护带来许多麻烦。影响其实用性。

为了避免上述弊病,使工资管理更容易适应多变的情况,在开发过程中作了如下设计:

不首先确定基本工资数据文件的结构,而是建立一个辅助文件,该文件存放基本工资数据中的每一应发部分的细项以及扣除部分的细项的数据内容(中文提示),数据宽度、小数点后位数,文件的每条记录,对应着工资的每条细项。提供维护模块,由用户根据需要填入具体内容。例如:

		辅助文件	数据宽度	小数位
		字段名	数据内容	
M 项 应发 细项	F1	基本工资	10	2
	F2	工龄工资	8	2
	F3	浮动工资	8	2
N 项 扣除 细项	K1	水电费	8	2
	K2	房租	8	2

F1, F2, F3, ..., Fm 约定为基本工资数据文件的应发细项的字段名, K1, K2, ..., Kn 约定为基本工资数据文件的扣除细项的字段名,用户可以根据需要改变 Fi 或 Ki 对

电子声音信箱

一种新型的通信工具

詹 前

现代化的社会是个信息交流的社会。商场就象战场，及时地获得各种信息，就可以把握时机和机会，以比较少的时间、成本和努力去争取成功。相反，就会坐失良机，一事无成。因此，信息是无价之宝，必须及时地获得，交换各种最新的信息，才能在瞬息万变的现代社会里生存和发展。

随着科学技术的进步和突破，用来传递信息的通信工具也取得了突飞猛进的发展。从电话、BB机、手提电话到下面介绍的电子声音信箱，愈来愈成为人们生活和工作的必不可少的通信工具。

电子声音信箱是一种新型的通信工具。它与电话、BB机、手提电话等相辅相存，且具有更大的信赖性和灵活性，特别适应在通信需求量大，经济活跃的国家 and 地区。电子声音信箱在美国、日本、新加坡已经得到了广泛的应用，相信也会在经济正在起飞的我国发挥它的巨大威力。

电子声音信箱有一个或多个电话号码。电子声音信箱的每个用户有自己的信箱号码（7位数字）和一个只有本人知道的密码（4位数字）。用户只要利用任何一部电话给电子声音信箱打电话，按照电话里的录音指示，按入信箱号码和密码，就可以打开自己的“声音信箱”，听到来自朋友和各方面的信息。朋友们也只要知道了彼此的信箱号码，就可以非常方便地相互留言了。因此，用户可以把自已的声音信箱号码印在名片上，联系起来更加方便。

电话、BB机、手提电话是现代社会重要的通信工具。在中国，电话线路昂贵，功能单一。有电话来时，如果没有人接，就会失去信息。即使安上昂贵的录音电话，也只有等回到家里，拿起电话时才可以听到信息。并且，有的信息常常被别人听去。BB机只是呼叫对方，不能马上留言（目前自动寻呼BB机可有10秒钟的留言）。打了BB机以后，必须守在电话机旁，不得离开。被BB机呼叫的人可能一时找不到电话，等找到电话打来时，电话线又往往处于占线中。到联系上时，会花去很多时间和精力。手提电话很时兴，有气派，当然价格也是最高的。手提电话只能在一定的范围内使用，走出使用范围，只是一个中看不中用的玩具。

电子声音信箱可以克服电话、BB机和手提电话的不足之处。事实上，电子声音信箱就是以克服上述问题为目的的新型的通信工具。它把电话网络同电子计算机相结合，克服了单一的电话、BB机和手提电话的不足，提供各种方便、及时、安全保密的通信服务。电子声音信箱一天24小时，可以随时为用户效劳。

用户只要到信箱管理公司登记，交纳注册和使用费，就可以获得一个信箱号码和密码。用户可以在自己的信箱里录上问候语，如：“我是张小明，从1月5日到1月15日到广州出差，有事请留言，我会随时听取留言。谢谢”。朋友们留言时，会首先听到问候语，然后按录音的指示留下信息。因此，用户可以随时随地利用电子声音信箱交换信息。与普通的信箱的不同之处是，

应的数据内容，数据长度，可以增加记录或删除记录。

根据该文件的描述而设立相应的基本工资数据文件如下：

		基本工资数据文件						
		应发细项				扣除细项		
		M 项				N 项		
姓名	编号	部门	F1	F2	F3	...	K1	K2...
张三	A0101	工会	204.00	10.00	30.00	...	22.50	9.50...
李四	B0001	长办	124.50	7.00	20.00	...	15.30	4.00...

该文件的数据结构是随着辅助文件的记录的改变而改变，由计算机根据用户在辅助文件的描述，自动修改基本工资数据文件的结构以及工资数据的追加，同时，对工资管理中的工资计算、统计、工资数据的输入、修

改、数据的检索、查询等功能，也都以辅助文件的描述为依据。

程序实现上采用数据库旋转技术结合数组的合理使用，尽量动态性的处理数据信息，将原来固定的处理方式改变为动态的处理方式。其中，基本工资数据的数据维护、数据检索、查询等功能，在屏幕上均实现自动以中文名提示数据内容。

该方案的优点：通用性强，适用范围广，动态性地处理数据信息，可大大减少软件维护的工作量。

该方案已应用在《广东省火电安装公司计算机网络管理信息系统》财务管理中，采用汉字 FOXBASE+ 2.1 实现。

普通信箱看得见摸得着，邮递员每天光顾一次，从发信到送到对方需要几天或更长时间。电子声音信箱是无形的，它的信箱号码对应普通信箱的地址。只要有人留言，信息就会以电波速度瞬时传到指定的信箱里。电子声音信箱的特点：方便、迅速、安全。

下面是电子声音信箱发挥巨大作用的典型例子：

▲一天，按惯例，小刘下班前，给女朋友小张单位打电话，想一起去看一部台湾电影。但不巧，小张去一个公司送货还没回来，小刘又要急着去买电影票。因为正是下班时间，传话也难传到小张那里。怎么办？此时，小刘想起了“电子声音信箱”，很多朋友都在用，是一种自动、快速、方便、准确的新型通信工具。如果自己和小张也租用它，就可以在小张的信箱里留下：“小张，我们今晚去海滨电影院看台湾电影《悲情城市》。我先去买票，7点半在电影院门口相见，不见不散”。小张在送货途中为联系不上小刘着急，也想起了朋友们都在议论、利用，引以为荣的“电子声音信箱”。如果有了信箱，可以就近找一个公用电话，打开信箱，就可以听到小刘的留言。小张在听完留言后，可以马上回答：“我会准时到达”，信息就会自动存入小刘的信箱。小刘只要找空看一下自己的信箱，就会知道小张的回答。他可以放心地在电影院附近的小商店里悠闲自在地为小张挑选一件礼物。到了7点半，给小张一个惊喜。看看，电子声音信箱，在最关键的时刻起了如此巨大的作用。男女恋爱，常有约会。有了电子声音信箱，不但可以及时联络，也不用频频打电话到对方的公司或单位。有了电子声音信箱，还可以为恋爱生活增加浪漫色彩。可以利用声音信箱，说说情话。别人听不见，当面又不好意思说。利用电子声音信箱，可以做到只有天知、地知和两人知道。

▲电子声音信箱还可以为外出办事的公司职员之间提供联系的方便。某公司的小陆和小施外出办事，小陆去海关和邮局，小施去外贸局和运输公司。相约谁先办完谁就去机场接人。但事前没法知道谁先办完，办完后又如何通知对方？此时，电子声音信箱就可以发挥它的巨大的作用。小施办完时，在附近找了一部电话，给小陆的信箱的留言：“小陆，我已办完事，到机场接人去了，请不要着急，办完事后直接回公司，准备接待客人的事就行了”。小陆正在邮局排队，心里很着急，因为离去机场接人的时间已剩下不多了。小施是否办完事了？自己是否应该去机场？此时，他想起了电子声音信箱。他急忙打开“信箱”，有小施的信息！顿时，小陆安心了。

▲电子声音信箱还可以与 BB 机相结合，加强 BB 机的功能。某公司李经理外出，去几个地方办事。临下

班前他的秘书小丽接到来自李经理的老朋友吴先生的电话，要李经理晚上去他家，说有一个过去生意上的朋友从外地来，有事要见李经理。小丽呼了李经理的 BB 机后，过了 20 多分钟还不见经理打电话回来，心里很着急。小丽要与男朋友去观看一个著名歌唱家的晚会，同时也想到李经理可能在车上或其它地方，一时找不到电话或没有时间打电话。怎么办？有了电子声音信箱，小丽只要往经理的信箱里留下：“我是小丽，4 点钟吴先生打来电话，要经理今天晚上去他家，说有外地来的朋友有事相见”。好了，李经理只要在谈话中找空去查看自己的信箱，就可知道一切。电子声音信箱就象一位忠于职守的秘书。有了声音信箱，就会获得各种发财的机会。在 BB 机可以自动交换的地方，有 BB 机的用户可以在信箱里登录上 BB 机号码。有信息来时，电子声音信箱还可以自动接通 BB 机。被 BB 机呼叫的人只要打开自己的信箱，就可以获得信息。电子声音信箱和 BB 机结合，使用起来更加方便。

▲电子声音信箱在国外常用于小组成员之间的联系，在我国更有它的使用价值。某公司有几千人，每个部门有经理、副经理，加总经理、副总经理，少说也有一二十人。平时，秘书通知各部门经理们开会时，常常是通知了张三，联系不上李四，因为经理们常常外出办事，很少坐在办公室里。后来，公司为经理以上的干部租用了电子声音信箱，给大家带来了很大的方便。当总经理理想召集经理们开会或下达简单的命令时，秘书只要利用电子声音信箱里的“小组成员之间的服务”的项目，就可以一次性地同时给多个经理传达开会的通知或总经理的指示，不用一一给他们打电话。方便，准确，及时。大大提高了工作的效率。

▲电子声音信箱还可以用来为公司作广告宣传。除了个人信箱服务以外，电子声音信箱还提供公共信箱服务。每个公共信箱有一个主题，例如：“XX 公司的保健系列产品”，里面可以容纳 50 条各种产品的录音广告。人们还可以利用公共信箱来交流信息。例如：在主题为“找工作”的公共信箱里人们可以把自己想要找的工作的种类，希望的工资，联系地址等留下。在主题为“招员工”的公共信箱里，各公司可以将计划招收的员工种类，报酬，工作场所和联系地址等留下。青年男女也可以利用公共信箱介绍自己寻求理想的对象。

电子声音信箱必将在我国发挥它的巨大的作用，电子声音信箱可以大大地改变电话线路紧张，通信落后的现状。中国经济的起飞，对通信事业的要求越来越高。电子声音信箱作为一种新型的通信工具，以其方便，迅速，准确，安全等独特的优点，一定会在我国得到广泛的应用和发展。

新型汉卡——星河汉卡

安徽星河机电技术研究所 周光发

中国科学技术大学自动化系 刘振安 张培仁

1 引言

目前在 PC 机上所流行的汉字操作系统,大致可分为三类:一类是以中国电子工业部第六研究所推出的 CCDOS 为代表的纯软件汉字系统。这一类汉字系统的不足之处早已众所周知,软字库若驻留内存占用内存过多,将字库放在磁盘上效率又太慢,而且对磁盘的损耗也较严重。这一类系统仅仅修改了 BIOS 的有关内容,使得大量的直接写屏的西文软件(如 NORTON、TURBO、网络软件等)不能使用,这就使得中文用户与西文用户相比,可用软件资源少得可怜。虽然有一些单位和个人对一些常用的西文软件(如 WORDSTAR、DBASE 等)进行汉化并开发了一些中文应用软件,但也不过是杯水车薪。汉化西文软件不仅存在着版权纠纷,也浪费大量人力和物力。

另一类是将字库固化形成汉卡,再配上软件而构成的汉字系统。这一类汉字系统较多,如西山 SPDOS 等。这一类汉字系统虽然因字库固化,解决了上述汉字系统的字库问题,但由于基本设计思想与前一类系统并无本质区别,适用性差依然很差,也没有解决软件资源少的问题。

还有一类汉卡不但固化了字库,而且支持直接写屏功能,所以从根本上解决了前两类系统的缺陷。但这类系统本身又存在一些新问题:其一是价格高,一般用户难以接受(尤其是在微机价格不断下降的今天,这一问题越发明显);其二是没有解决西文软件中的西文制表符与汉字的冲突,造成西文软件虽然能用,但由于将西文制表符显示成汉字或半个汉字,使原本精心设计的人机界面变得混乱不堪。还由于功能键设置不合理,从而使得一些西文软件在运行时发生冲突。

此外,上述三类系统均不同程度的存在着诸如输入方法比较简单,要扩充一个输入法相当困难,即可扩充性差、支持的打印机种类少,并且每一种打印机都需要一个驱动程序和不能适应 DOS V5.0 等不足。

我们研制的汉字操作系统——星河汉字系统,已经解决了上述问题。

2 星河汉卡的基本性能

星河汉字系统的主要性能和特点概括起来有以下几个方面:

① 适用性强

可以适应于 V3.0 以上的 PC-DOS、MS-DOS 和

DR-DOS 不同操作系统。打印字库可以放在任意硬盘的任一子目录中。

② 占用内存少

一方面由于字库固化在汉卡上,另一方面各种输入模块采用覆盖方式使用一内存空间,使得同时使用八种输入方法仅用 50K 内存空间。

此外,在 DOS 5.0 或 DR-DOS 6.0 下,还可以将系统驻留在内存高端,从而不占用 640K 基本内存空间。

③ 支持直接写屏、智能判别制表符

大多数西文软件均可以不经汉化直接使用,如 Novell 网络软件、Turbo 系统软件、西文 Dbase、Foxbase、Pctools、Norton 系统软件等等。智能判别西文制表线、背景和阴影,显示正确美观。

④ 光标形状可变、支持鼠标

系统支持可变光标,但不改变系统时钟(故不支持光标闪烁)。

系统还支持鼠标,字符形或图形鼠标均可以得到正确显示。

⑤ 多种输入方法、可扩充性强

现有区位码、拼音(缩拼/全拼)、五笔字表、表形码和电报码五种输入方法。

留有程序接口,用户可以方便地设计自己的输入方法;也可以使用输入模块生成程序自动生成用户所需要的输入方法。

⑥ 词组量大、使用灵活

系统配有拼音词和通用词库。

拼音词库有 37865 条双字词,装入内存后,可在拼音输入法下使用组输入,并可以动态进行词组的增加、删除、精简等操作。

通用词库有 37865 条双字词、663 条三字词、873 条四字词和少量多字词语。该词库装入内存后,五笔字型、表形码以及使用输入模块生成程序生成的输入方法均可以进行词组输入,从而达到一库多用的目的。

系统配有通用词库生成程序,用户可以方便地根据自己的需要生成自己词库或是对原词库进行增删。

词库可以根据用户的需要随时驻留或撤出内存。

⑦ 两组功能键,避免冲突

系统设置了两组或功能相同的功能键,一组功能键与市面流行的汉字系统完全相同,以保证用户使用方便,一组功能键用于避免与西文软件的功能键发生冲突。两组功能键可以由用户根据需要随时选择。

数据加密工具—PC SECURE

石家庄市职业财会学校计算机室 张春强

许多朋友为了防止自己的程序和数据文件被他人非法使用而绞尽脑汁,但是面对具有丰富 DOS 知识的专业人员,道道防线都能被轻易突破。使用美国 CentralPoint Software 公司的 PC 工具箱—PC TOOLS V7.0 中的 PC SECURE 进行数据加密,效果不凡。

PC SECURE 是保护用户程序和数据不被非法使用的一个专用工具,由以下三个文件组成:

- (1) PCSECURE.EXE——程序文件
- (2) PCSECURE.HLP——帮助信息文件
- (3) PCSECURE.CFG——文件加解密设备文件

如果用户想查看帮助信息,还需调用文件 CPSHELP.OVL。

PCSECURE 有两种方法,一种是在系统提示符下运行该命令文件,加入功能参数,进行加、解密操作。

下面提供的参数使得用户能够灵活、快速地运行 PCSECURE,对文件进行保密和解密。

参数 功能说明

- /D 对指定的文件进行解密
- /F 对指定的文件进行全面加密(压缩)
- /Q 对指定的文件进行快速加密(压缩)
- /C 对文件加密时不作压缩处理
- /P 对文件加密

/Kxxx 用 xxx 作为密码(xxx 可以是 5~32 个任意字符或数字)

以上参数可以任意组合,进行操作,但用户必须牢记密码,否则将无法使用该文件。其用法举例如下(有底划线者为用户输入的内容):

例 1.对驱动器 C 中子目录 WPS 中所有扩展名为 .WPS 的文件进行快速压缩加密,密码是 ZHANG。

```
C> PATH C: \PCTOOLS
```

```
C> CD \WPS
```

```
C> PCSECURE / Q / KZHANG *.WPS
```

例 2.对当前目录下的所有文件进行全面加密。

```
C> PCSECURE / F / P *.*
```

⑨ 适应面广的打印驱动程序

系统提供了一个通用 24 点阵汉字打印驱动程序生成器,除可以生成 OKI 系统、Epson 系列、AR-3240 系列、M-2024/M-1724 系统、NEC 系列、TH-3070 系列、NK-3824 系列、AR-2463 和 CPK-5240 等打印机的驱动程序外,还可以由用户提供一系列参数后,生成任一 24 针打印机的驱动程序。

⑩ 性能价格比高,适用范围广

由于该系统有众多独特之处,而价格又明显低于一般的汉字系统,其性价比之高也就格外突出。该系统可广泛适用于网络工作站、仿真终端、中文录入和中文管理的不同需要,对于从事中文软件开发人员、学校以及家庭更是汉字系统之首选。

3 星河汉字系统运行环境

① 运行星河汉字系统的硬件环境

任何 PC、XT、AT286、386 兼容机

256KB 基本内存

以下显示器之一

Monochrome 720×350 单色图形显示器

VGA 640×480 单色或彩色图形显示器

硬盘、软盘及兼容键盘

星河 I 型汉卡

任何 24 针点阵式打印机

② 星河汉字系统的软件环境

星河汉字系统的低层软件为 PC-DOS 或

MS-DOS,版本在 3.0 以上。星河汉字系统本身由以下几部分组成:

●基本系统模块

系统启动模块 (BOOT.COM)

显示驱动模块 (VEDIO.DRV)

字库管理模块 (FONT.DRV)

键盘管理模块 (KEYBD.DRV)

拼音词管理模块 (EXTSTR.DRV)

●输入法模块,现配有以下几个:

拼音(缩/全)输入法 (HANZI2.OVL)

五笔字型输入法 (HANZI5.OVL)

表形码输入方法 (HANZI4.OVL)

电报明码输入法 (HANZI8.OVL)

●词组库

拼音词组库 (PYWORDS.LIB)

拼音词组库装入程序 (WL.COM)

通用词组库 (WORDS.BIN)

通用词组库装入程序 (LOADLIB.COM)

●24 点阵四体(宋、黑、仿、楷)打印字库 (CCLIB24)

●各种实用程序

系统安装程序 (SETUP.EXE)

输入模块生成程序 (KEYMAKE.COM)

通用词组库生成程序 (LIBMAKE.COM)

24 点阵打印程序生成程序 (PRT.COM)

INPUT KEY: ZHANG

VERIFY KEY: ZHANG

或 C>PCSECURE / F / KZHANG *.*

例 3. 对当前目录下的所有文件进行快速无压缩加密, 密码是 ZHANG.

C>PCSECURE / Q / C / KZHANG *.TXT

例 4. 对 LHARC.EXE 文件用密码 ZHANG 进行解密.

C>PCSECURE / D / KZHANG LHARC.EXE

另一种用法是进入 PC SECURE 主屏幕后, 在菜单方式选择进行的操作, 在系统提示下进行加、解密操作.

在 PC SECURE 中, 设置了两重保密. 当第一次使用 PC SECURE 时, 将提示用户使用一个主控词 (Master key). 它就象是一把万能钥匙, 即使用户忘记了某个具体文件的密码, 也可以用它打开任何加密文件. 主控词由 5~32 个字符 (包括数字) 组成, 也可以用 8 个十六进制数码组成. 在输入十六进制数码前, 必须按下功能键 F9. 为了保密, 主控词在输入过程中, 只显示 "*" 号. 主控词输入后, 屏幕马上显示一个要求用户核对主控词的窗口 (如果主控词是用十六进制数码, 则不要求核对). 如果两次输入的主控词不同, 则要求重新输入, 直至二者完全一致. 需要用户注意的是, 用户必须牢记主控词, 不可忘记, 否则后果严重.

PC SECURE 的第二重保密是密码 (Password). 密码与主控词不同. 主控词是一把万能钥匙, 可以解开任何文件, 而密码是具体文件的口令, 只对某一个或几个文件有效. 同主控词一样, 它也由 5~32 个字符 (或数字) 组成, 密码在输入过程中只显示 "*" 号, 输入完毕后, 立即要求核对. 如果两次输入的密码不同, 则要求重新输入, 直到二者相同.

在 PC SECURE V7.0 主屏幕顶端, 有三个功能项, 即:

1. File (文件操作)

2. Options (选择项)

3. Help (帮助信息)

主屏幕下方提供了 5 个功能键, 其含义如下:

F1——HELP (帮助信息)

F2——INDEX (帮助信息索引)

F3——EXIT (退出)

F4——ENCRYPT (文件加密)

F5——DECRYPT (文件解密)

用户可直接以功能键选择操作, 也可以键入功能键开关字母, 在菜单方式下选择操作内容.

功能项 File 中包括: ENCRYPT (文件加密) 和 DECRYPT (文件解密) 两项内容, 用户按照提示操作即可; 功能项 Help 中包含各类信息, 供用户查看; 而选择项 Options 主要用于设置 PC SECURE 的各种状态, 包括压缩、密码、文件属性、文件读写、加密方

式等等.

在 PC SECURE 主屏幕上, 选中 Options 选项, 这时屏幕即显示出 Options 下拉菜单:

Full DES Encryption
✓Quick Encryption
✓Compression
✓One Key
Hidden
Read Only
✓Delete original File
Expert Mode
Save Configuration

Full DES Encryption 表示全面加密方式, 它与 QuickEncryption (快速加密方式) 只能二者取其一, 系统默认值为 QuickEncryption 方式, 在 Options 菜单中, 把光标覆盖处回车, 或者直接键入开关字母键, 即进入该功能键. 命令左侧出现 "✓" 为处于 ON 状态, 否则为 OFF 状态. ON 状态表示该功能项起作用.

Compression 选项表示压缩选择, 其处于 ON 状态时, 可以使大多数文件压缩 30%~60%, 从而减少占用空间.

One key 选项表示选择单一的密码对所有的文件进行加解密操作, 当该项处于 ON 时, 加解密所有文件可以使用同一密码; 当其处于 OFF 状态时, 每加解密一个文件, 都要输入一个密码.

Hidden 选项表示把加密后的输出文件置为隐含属性, 加深保密程度.

Read Only 选项表示把加密后的输出文件置为只读属性, 避免误删除.

Delete original File 选项处于 ON 状态时, 加密文件将覆盖原文件, 加深保密程度. 如果处于 OFF 状态, 则加密后产生一个与原文件名相同的文件, 其扩展名为 ".SEC", 而原文件仍然存在, 实际上起不到保密作用.

Expert Mode 选项处于 ON 状态时表示加密文件时采用专家模式, 即文件加密时主控词失去作用, 这样可加强保密程度; 如果处于 OFF 状态, 则主控词有效, 即使遗忘了加密文件时使用的密码, 使用主控词仍可对其进行解密操作.

以上设置确定后, 使用 Save Configuration 项, 将其存入 PC_SECURE.CFG 文件.

事实证明, PC SECURE 既具有加密和解密的双重功能, 而且使用方便, 在对磁盘文件进行保密的同时还可进行压缩处理, 因而既可有效地利用磁盘空间, 又提高了程序及数据的安全性.

以上方法在 SUPER PC、AMI 286 和 AST 386,486 上通过.

工具软件极品 RAMinit 2.0

雷军

许多驻留软件运行后无法撤除，再运行其他程序时内存不够，只有关机重启动。RAMinit(简称 RI)可以帮助您清除驻留程序，同时还可以帮助您解决很多实际问题。RI 1.0 从 1989 年推出以来，受到许多用户好评。但 V1.0 还存在一些问题，如无法释放 XMS, EMS, UMB, 无法清除 cache 和网络驱动程序等，作者最近抽出四天时间，重写了 RAMinit，正式推出 V2.0。希望大家多提宝贵意见，同时也希望大家会喜欢她！

RAMinit 2.0 版权归黄玫瑰软件公司所有。如果觉得好用，请给予信心支援，并推荐给更多的人使用，这样能使 RI 变得更好！

一、RAMinit 功能

如果您运行了 UC DOS 1.0，现在您想再用 SPDOS，您只有重新启动机器。如果您拥有 RI，可以直接热键清除 UC DOS，不必重启动。如果您驻留了 SideKick，而内存不够，RI 可以帮您清除一部分驻留程序。

RI 是电脑玩家的好帮手，工具软件中的极品！

二、RAMinit 特色

1. RI 驻留后，能够清除其后驻留的各种各样的程序，如所有的汉字系统，SideKick 工具程序，Novell 驱动程序，PC-CACHE 磁盘缓冲程序和 GB4 游戏克星等。

2. RI 能够中止当前正在执行的程序，如许多无法退出的游戏程序。程序死机时，可以按热键中止退出，勿需重新启动。

3. RI 能够释放驻留程序分配的所有常规内存，恢复环境参数块，键盘 NumLock, Caps Lock, Scroll Lock 的状态，恢复时钟频率，关闭打开的文件，恢复系统时间，关闭扬声器，重新初始化 8259，复位磁盘机，复位监视器，复位打印机和串并口。

4. RI 能够释放其后分配的 EMS 和 XMS 的内存块。

5. RI 能够用热键，命令行参数和中断功能调用激活，形式灵活多样。

6. RI 热键设计巧妙，可以识别 101 键左右 CTRL, ALT 键，方便设置。如设置出左 CTRL 加右 CTRL 加 'X' 这样的热键。

7. RI 能够始终把握键盘控制权，同时又能与 SideKick 这类抢夺控制权的程序兼容。

8. RI 能够驻留多份，可以逐次清除其后驻留的程序，也可一次清除第一份 RI 以后驻留的全部程序。可选择是否清除 RI 本身。

9. RI 保留系统大量的信息，全部采用压缩存储。第一次驻留占 2500 个字节，第二次驻留占 1K 字节左右。

10. RI 可以驻留在 UMB 中，同时还可以释放其后分配的 UMB 块，支持 386MAX 的 UMB 功能。

三、RAMinit 安装与使用

● 键入 "RI / ?" 可得到简要的帮助。

C:> RI / ?

RAMinit Version 2.0 Copyright (c) 1989-1992 Yellow Rose Software Co.
Programmed by Mr. Leijun Nov 1989

Usage: RI [options]
/ H, / ? Display this screen
/ CLS Removes all TSR programs after current RI
/ RET Removes TSR programs include current RI
/ NEW Residents a new data copy of current environment
/ ALL Removes all RI copies and all other tsr programs
/ Sxyy.. Define Hotkey x = AuxHotkey yy.. = shift status
x = auxiliary hotkey (default is "X")
x equ "1" means need AuxHotkey
yy.. = shift status [CASCas]',0dh,0ah
C: Left Ctrl A: Left Alt S: Left Shift
c: Right Ctrl a: Right Alt s: Right Shift

Example: "RI / S1c" means Hotkey is Right Ctrl+X
"RI / S0Cc" means Hotkey is Left Ctrl+Right Ctrl
"RI / CLS" equals simply press hotkey',0dh,0ah
"RI / RET" Removes all TSRs after current RI and this RI
Contact me for RAMinit problems: (01)2010088 Call 4166

● 直接键入 "RI"，RI 就驻留了，同时显示系统当前的热键

RAMinit Version 2.0 Copyright (c) 1989-1992 Yellow Rose Software Co.
Activate with: Right Ctrl + Right Alt + X

系统提供用户的默认热键是右 CTRL 加右 ALT 加 'X' 键。如果使用的是 88 键的键盘，默认的热键是 CTRL 加 ALT 加 'X'。(这是因为 88 键键盘无法识别左右 CTRL, ALT)

● 激活 RI 的方法有多种：

- (1) 直接按热键，就可清除其后驻留的程序；
- (2) 键入 "RI / CLS"，等价直接按热键；
- (3) 键入 "RI / RET"，就可清除 RI 以及其后驻留

的程序;

(4) 键入 "RI / ALL", 就可清除第一份 RI 以及其后驻留的全部程序;

(5) 功能调用:

INT2F Func

AX = C0D7h Return RI segment in AX

AX = C0D8h Removes all TSR programs after RI

AX = C0D9h Removes all TSR programs include RI

AX = C0DAh Removes all RI copies

● 热键设置:

键入 RI / Sxyy..

x: 为 '1' 表示需要 'X' 辅助热键;

为 '0' 表示不需要 'X' 辅助热键;

y: 表示 Shift 键的状态, 可接任意多个 Shift 键的状态

'S' 表示左 SHIFT

's' 表示右 SHIFT

'C' 表示左 CTRL

'c' 表示右 CTRL

'A' 表示左 ALT

'a' 表示右 ALT

如: 设置热键为: Left Ctrl + Right Ctrl + 'X' 的命令为:

RI / S1Cc

如果设置成功, 当前驻留在内存里 RI 的热键也被修改。

注意: 有的热键无法识别, 如 Left Shift + Right Shift + 'X'。不要设置这样的热键。如果设置了, 请用命令行参数清除。

● 驻留多份 RAMinit 的方法

驻留一份 RI 后, 如果不驻留新的程序, RI 不能再次驻留, 除非使用 / NEW 参数。如果 RI 之后驻留了新的程序, 再运行 RI 时, RI 会提问是否继续驻留。

RI 活动时, 屏幕左上角会显示一个数字, 这个数字表明当前驻留了几份 RI。

需要清除所有的 RI 时, 请用命令 "RI / ALL"。

RI 第一次驻留, 占 2.5K 内存, 以后驻留时, 只驻留数据, 约占 1K 字节的空间。RI 的所有数据均压缩存储, 故占内存省。

四、RAMinit 使用注意

虽然, RI 能清除 CACHE 程序, 但最好不要在 RI 之后驻留 CACHE 程序; 同样, 我们建议您在 RI 之前驻留网络驱动程序。您的入网程序可以这样编写:

RI / ALL

IPX

NETX

F:

LOGIN

RI

用 RI 清除驻留程序, 一般不会出错。有时使用 RI 后, 发现系统有问题了, 最好能重启动机器。

五、RAMinit 使用示范

; 驻留第一份 RI

C:> RI

RAMinit Version 2.0 Copyright (c) 1989-1992 Yellow Rose Software Co.
Activate with: Left Ctrl + Right Ctrl

; 驻留 SideKick

C:> SK

; 驻留第二份 RI

C:> RI / NEW

RAMinit Version 2.0 Copyright (c) 1989-1992 Yellow Rose Software Co.
Activate with: Left_Ctrl + Right_Ctrl

Already installed!

For Help, type "RI / ?".

; 驻留 GB4

C:> GB4

; 驻留第三份 RI

C:> RI

RAMinit Version 2.0 Copyright (c) 1989-1992 Yellow Rose Software Co.
Activate with: Left Ctrl + Right Ctrl

Already installed!

For Help, type "RI / ?".

Residents a new RAMinit copy [y / n]? y

OK, RI No.3 residents successful!

; 显示内存印象

C:> MI

Memory Info V8 (c)1992 Central Point Software, Inc.
Addr. Total bytes Program or device driver

025Dh	1,088	Device = HIMEM Attr = A000h Name = XMSXXXX0
02A2h	1,184	Device = RAMDRIVE Attr = 0800h Drive E
0437h	2,624	COMMAND
04EAh	2,560	RI; 第一份
058Ch	84,224	SK
1A1Eh	1,120	RI / new; 第二份
1A66h	11,696	GB4
1D43h	1,152	RI; 第三份
1D8Dh	534,320	<largest free area>

655,360 bytes (640k) total DOS 5.00 conventional memory.

534,320 bytes (522k) largest executable program.

0 bytes Extended (AT / 286 / 386) memory, reported by BIOS.

7,360k bytes XMS 3.0 (3.09) memory: 1,274k used + 6,086k free. DOS in HMA.

C:> RI / RET; 清除第三份 RI

C:> RI / CLS; 清除 GB4

C:> RI / ALL; 清除第一份以后所有的驻留程序及 RI 本身
按左 Ctrl 和右 Ctrl (默认的热键), 等价于 RI / CLS.

对 RI 使用有疑问, 发现 BUG 或不足之处, 请与作者联系。

两则行之有效的小改动

广州市计算机应用研究所 赖宏宇

人们在使用软件的过程中，经常会发现一些不便，甚至错漏之处。问题可能是软件编写者考虑不周所致，也可能是使用者改变了软件原设想的运行环境所致。然而，对于这些软件，往往有不少只需稍微改动，即可获得满意效果。以下两则小改动，就属于这类情况。

一、网络工作站之间的汉字信息传接

当今 NOVELL 局域网的网络操作系统最新版本为 NetWare 386 V3.11。运行这套软件时，网络工作站可以使用一个网络命令—SEND，把简短信息从本站发送到其它工作站，而其它工作站就会把接收到的信息显示在监视器的底行。籍此功能，可以实现异地网络操作员的交谈和事务通知。在我国，计算机通常都要与中文系统相伴，完成汉字信息处理。然而，当我们在汉字系统下用 SEND 命令发送信息到另一台工作站时，该信息没有显示出来。而我们在接收方按键盘使之转入纯西文显示方式时，信息立即显示出来。因而有必要查一下工作站的启动程序，追究其原因，使之在汉字系统下也能显示接收到的信息。我们都知道，DOS 系统下的工作站启动程序为：

```
IPX
NET3 (假定在 DOS3.XX 状态下)
F:
LOGIN
```

我们只需对 IPX.COM、NET3.COM 和 LOGIN.EXE 三个程序进行分析即可。在分析过程中，我们发现在 NET3.COM 内有这么一段程序：

```
1418:890B 50      PUSH  AX
1418:890C 53      PUSH  BX
1418:890D B40F    MOV   AH,0F
1418:890F CD10    INT   10
1418:8911 2E      CS:
1418:8912 88269088 MOV   [8890],AH
1418:8916 2E      CS:
1418:8917 A28F88  MOV   [888F],AL
1418:891A 2E      CS:
1418:891B 883E8D88 MOV   [888D],BH
1418:891F 5B      POP   BX
1418:8920 58      POP   AX
1418:8921 C3      RET
1418:8922 E8E6FF  CALL  890B
1418:8925 30E4    XOR   AH,AH
1418:8927 2E      CS:
1418:8928 A08F88  MOV   AL,[888F]
1418:892B 3C03    CMP   AL,03
1418:892D 760E    JBE   893D
```

```
1418:892F 3C07    CMP   AL,07
1418:8931 740A    JZ    893D
1418:8933 3C0A    CMP   AL,0A
1418:8935 7406    JZ    893D
1418:8937 3C0B    CMP   AL,0B
1418:8939 7402    JZ    893D
1418:893B B4FF    MOV   AH,FF
1418:893D 08E4    OR    AH,AH
1418:893F C3      RET
```

INT 10H 的 0FH 调用，能返回当前显示模式值到 AL。稍为懂一点 IBM PC 显示模式的人都知道，显示模式 1、2、3、7 等均为文字方式。该程序判断到非 1、2、3、7、A、B 显示模式时，就把 AH 置为 FFH，而对于 1、2、3、7、A、B 显示模式就置 AH 为 00H，因而我怀疑是否这个 FFH 值屏蔽了中文状态下的信息显示。因为中文状态下屏幕的显示模式为图象方式，原程序完全有可能只允许文字方式时才显示接收信息。我们试着把 NET3.COM 里 893C 单元的 FFH 改为 00H，重新启动工作站，结果一举成功。而且，无论工作站采用的是 CGA、EGA、VGA 还是 Monochrome 监视器，进入任何中文操作系统，均可在屏幕最底行显示其他工作站送来的简短信息。这些短文既可是汉字，也可是任何可显示的 ASCII 码字符。在纯西文状态下，被改程序也能完全正常运行。

二、2.13H 的屏幕打印程序改造

CCBIOS2.13H 是当前非常流行的中文操作系统，它功能强，使用方便，受到微机用户的一致好评。然而在屏幕打印功能方面，尚有待于改进。

熟悉 2.13H 的人士都知道，它配有两个屏幕打印程序：一个是 SGP.COM，是用于 CGA 显示方式的；另一个是 SEGP.COM，是用于 EGA 和 VGA 显示方式的。本文仅以 SEGP.COM 为例，而 SGP.COM 的道理相同。

启动过 SEGP.COM 程序之后，只要按下 SHIFT+PRTSC 键，打印机就会把屏幕上的图象打印出来。其结果是：当屏幕的背景为黑色时（一般均是这种情况），屏幕上的字或线条，在打印纸上变成了黑块中的白线条，很难才分辨出来。打印出来的图象，就好象黑白照片的底片一样。为此，有必要修改这个程序，使屏幕上的字或线条，在打印纸上能以黑色打出，而背景则转为白色。作这样的修改也不难，只要在程序中找

防文件型病毒有新招

江西财经学院计算中心 肖清雷

在病毒与我们同在的环境中,病毒确实给我们带来不少麻烦。反病毒斗争中,防为上策,杀之为下策。这是因为病毒的检测,只能在病毒出现之后,根据病毒特征代码研制出相应杀毒软件,总处于一种“滞后”状态,倘若某个系统或文件能够抵御病毒侵入,我们就称它具有免疫能力。

在病毒家族中,由于引导型病毒要占领引导区,目标暴露极易被发现。对付它的措施就很多了,比如不用来历不明的软盘启动,对带有硬盘的机器杜绝用软盘启动,也可以在启动盘的 AUTOEXEC.BAT 文件运行 BOOTSAFE.EXE,时刻监视着对引导区的动作。而对付文件型病毒目前还没有很有效措施。

难道文件型病毒真的就无法预防吗?看下文便知。

众所周知,文件型病毒主要攻击目标是.COM 和 .EXE 为后缀的可执行文件,抓住这点,就可实施免疫。主要思想:通过修改系统文件中的命令解释程序(即 command.com),使在该系统环境下的可执行文件的后缀不为.com 和 .exe,比如采用下文中的.con 和 .ese 后缀。这种被改后后缀的可执行文件,不但能正常运行,而且免于文件型病毒的攻击。

这里就着重讲讲,这种系统的获得。

取一张干净启动盘,它必包括 IBMBIO.COM、IBMDOS.COM、COMMAND.COM 三个系统文件。

首先,将这三个系统文件改成 CON 为后缀。这用 PCTOOLS 很容易做到。由于启动时,引导程序会检测其中的 2 个隐含系统文件。这样改名后,必然会使该盘启动不成功。可用 PCTOOLS 选择磁盘操作菜单,用 E 命令将引导区内容显示出来,选择第二页(即引导块的下半部)。在屏幕的右边 ASCII 码显示区,可看到“IBMBIO.COM、IBMDOS.COM”字符串,按 F3 键对其修改,改成“IBMBIOS.CON、IBMDOS.CON”就可以了。

第二,在文件操作菜单下,选择 IBMBIO.CON

文件进行“F”操作,在其中查找“COMMAND.COM”字符串,找到了,选择 E 命令使之修改为“COMMAND.CON”

第三,修改命令解释程序。选择 COMMAND.CON 文件进行“F”操作,在其中查找“COMMAND.COM”串,将所有的改为“COMMAND.CON”,在其中查找“COM、EXE、BAT”串,将其修改为“CON、ESE、BAT”,于是,命令解释程序修改完毕。

这样一张能防文件型病毒的系统盘就做成了,对于有 IO、SYS 和 MSDOS、SYS、COMMAND 的系统,只要修改 IO.SYS 中的“COMMAND.COM”串为“COMMAND.CON”,以及 COMMAND.COM 如上述修改就可以了。

对于用户的可执行文件,只要将其改为 CON 或 ESE,就具有免疫能力。这种免疫能力不同于施加病毒感染标志的免疫,那种免疫只能针对某一种病毒。而这种免疫是针对所有的文件型病毒的。

据说有的文件型病毒还会攻击.OVR 或.OVL 后缀的覆盖文件。我们同样可以通过其后缀名的修改达到自然保护的目的,并修改加载程序内的文件名字符串处,就能正常运行。

这种防病毒方法,简便易行,一劳永逸。读者不妨一试。

最后要补充的是,使用外部命令 FORMAT B: /S 和 SYS b: 所获得系统是原先的系统。我们只要将这 2 个程序中的“IBM BIO COM”,“IBMDOS COM”,“COMMAND COM”串修改在.CON 后缀,然后运行 FORMAT B: /S 或 SYS b: 所获得系统就是被作了上述修改的系统。

有些软件,在运行时会首先检测自己文件名是否被修改过,改过的运行会死机,针对这种情况,只要通过分析和修改也不难使在该系统下正常运行。

能使软盘翻修的小程序——800

北京理工大学 71891 班 杨海军

随着 286 及其兼容的机的迅速普及和大型软件的推广,低密盘越发显得容量不足。既然已有了 1.2M 驱动器,若能充分利用已有资料,将用户手头的低密软盘容量扩大无疑是极有诱惑力的。尽管不少杂志都提供了能将 360K 软盘翻倍的程序,但在具体实现的兼容性等方面还存在许多具体问题,笔者曾用一份这样的程序格化了数张低密盘,但只隔了几个小时,盘上的内容均已无法读出,只好放弃了扩充容量的打算。

而笔者前不久得到一份效果一份效果极佳的小程序:800。它从 BIOS 级给了我们新的选择,该程序是由一位意大利人编写的。这里就介绍一下的该程序,与广大计算机爱好者共享这一成果。

“800”是一个内存驻留程序(TSR),短小却很实用。尽管大小不到 1K,驻留后只占用微不足道的 864Bytes 内存,对软硬件环境要求都不高,但它给我们提供了多达 8 种的软盘格式,只要是 IBM PC 或兼容机, DOS 版本不低于 3.30,这些格式均安全有效。

如果机器配有 1.2M 和 360K 两个 5" 软驱,你可使用的格式有 6 种,低密盘的容量也可扩至 800K;倘若配置了 1.44M 的软驱,你还可以放心地使用容量达 1.6M 之多的 3" 软盘。

在 800 驻留内存之后,你就可以运行 DOS 系统下的 FORMAT 程序格化软盘了(800 并未自带格式化软盘的功能,也无此必要)。命令格式如下:

容量	磁盘类型	适用软驱	命令参数
360K	DD	5" (360K & 1.2M), 3" (720K & 1.44M)	/T:40/N:9
400K	DD	5" (360K & 1.2M), 3" (720K & 1.44M)	/T:40/N:10
720K	DD	5" (1.2M), 3" (720K & 1.44M)	/T:80/N:9
800K	DD	5" (1.2M), 3" (720K & 1.44M)	/T:80/N:10
1.2M	HD	5" (1.2M), 3" (1.44M)	/T:80/N:15
1.36M	HD	5" (1.2M), 3" (1.44M)	/T:80/N:17
1.44M	HD	3" (1.44M)	/T:80/N:18
1.60M	HD	3" (1.44M)	/T:80/N:20

显而易见,上述格式中有四种是标准化的格式(360K, 720K, 1.2M 和 1.44),这些格式的软盘在读写操作时最快,且不受软硬件环境的影响。同时,还解决了一个让人头疼的难题:不同类型的磁盘之间的整盘拷贝操作。众所周知,5" 盘和 3" 盘之间是无法实现 DISKCOPY 之类操作的,但现在只需将两种盘按同一种格式 FORMAT 一下,这个问题便迎刃而解(建议使标准格式,以缩短读写操作时间)。

上述八种格式中除 1.6M 以外均可做成启动盘,只需在 FORMAT 时加上 "/S" 开关即可。当然,在用非

标准格式的盘启动时最好在 AUTO EXEC.BAT 中及时调用 800,否则非标准格式的软盘将无法正常工作。

下面列出 800 的部分命令开关:

/off 撤除 /on 安装 /0 未装驱动器(A:)
/00 未装软驱(B:) /36 5" 低密软驱 /12 1.2M 软驱
/72 720K 软驱 /14 1.44M 软驱 /? 寻求帮助
/C0 兼容模式 /Ke 保存环境

如果在运行某些特定程序(如 COPPIER)时遇到问题,可先在 DOS 环境下打入 800/off 命令撤出驻留。需要使用时只需再次打入 800/on 即可。

在安装时,800 会自动识别驱动器类型并将之列出,如果与实际配置不同,你可以在安装时加上相应参数来指定驱动器类型(例如"800/12/14"意为 A 驱为 1.2M, B 驱为 1.44M)。

对于一些并不完全兼容的 BIOS,你在 1.2M 的驱动器上使用 720K 或 800K 软盘或许会遇到麻烦,只需在安装时加上 "/C0" 参数;如果在多任务环境下使用 800,最好在安装时加上 "/Ke" 参数,否则,有时可能会发生“内存分配错”之类的错误。

而"/?"参数则可以向使用者提供帮助。

800 是从 BIOS 级别上支持新的软盘格式的,但它并未代替 BIOS,它对磁盘的操作是完全开放的(INT 13H),因此最好能保证机器的 BIOS 兼容。该程序可以随时安装(如果它不能正常工作),将其它驻留内存的程序移走试一试)。

注意:在 800 支持下,你还可以使用其他软盘格式,这通过 FORMAT 命令中的 /T: n 参数 n 来达到,此处 $1 < n < 85$ (1.2M) 对于 360K 驱动器 $n < 43$ 。尽管 $n > 40$ (低密)或 $n > 80$ (高密)格式均可由 800 支持,41 和 81 格式也较安全,但超过这一值的磁道的可靠性要低得多。因此,希望使用者不要片面地追求容量,最好采用 720K 格式,在保证效率和可靠性的前提下扩充容量;另外,对于诸如 DISKCOPY 和 DISKCOMP 之类的功能,最好用 DOS 中的外部命令完成,因为有些工具软件包不支持新的磁盘格式。

你还可以充分利用该程序的功能,采用非标准格式做一些软盘,这此盘只有在 800 驻留内存时可以使用,亦不失为软件加密和保护的一条有效途径。

尽管看来在使用上有许多要注意的地方,但实际上你可以放心地使用 800,它可以向你提供高效安全的服务,有疑问时用"/?"参数查看一下帮助便可得到解答。但愿广大爱好者能从该程序中受益。

Windows 底层结构剖析

北京清华大学 王传君

Microsoft Windows 是一个在微机上广泛流行的多任务环境。它在覆盖了 DOS 功能的同时, 提供了丰富的图形用户界面。高收的应用程序编程接口和功能强大的软件开发工具箱 SOK, 实现了数据动态交换和模块动态连接, 自动内存管理等功能。从而进一步发挥了微机的潜力。

从内部结构上看 Windows 由三部分组成: Windows 内核、用户界面和图形设备接口 GDI。在功能上, Windows 是对 DOS 的强有力的扩展。往往许多 Windows 应用程序的设计者对 Windows 底层的软硬件环境及其对 DOS 的扩展功能不十分了解。一方面, 这是因为有关这些底层结构功能的文档只在 Windows 设备驱动程序工具箱 DDK 中提供, 而未在常见的 Windows 用户手册、编程指南中详述; 另一方面, 由于 Windows 提供了极为丰富的应用程序编程接口 API, 经常给人以目不暇接的感觉, 许多编程人员将注意力集中在 SDK、API 上, 对 Windows 底层结构及运行机制不太了解也无可非议。然而, 由此而产生的许多问题, 诸如内存管理中有关扩展内存的使用, 32 位平伸内存模式的实现和使用, 不同地址方式之间的切换等等, 都是一些常常令 Windows 应用程序开发人员头痛的问题。本文试图通过对 Windows 底层软硬件环境的分析谈谈一些目前大多数 Windows 参考手册中很少提及的问题, 以加深对 Windows 编程环境的理解。

一、Windows 的软硬环境

根据系统配置的不同, Windows 有三种运行模式: 实模式、标准模式、386 增强模式。实模式主要用于兼容 Windows 老版本下开发的应用程序, 因为这此程序中的大部分都与后面两种模式下运行的 Windows 内存管理不兼容。在实模式下运行时, Windows 不使用扩展内存, 功能上与 DOS 差别不大。而标准模式和 386 增强模式则对 DOS 进行了许多扩充, 尤其表现在存贮管理部分, 突破了无法使用扩展内存的限制。标准模式一般适用于 80286CPU 在保护方式下的 16 位段地

址寻址方式, 原理仍比较简单, 类似前者。而 386 增强模式则与这两者在内部机制上存在较大差别, 一般适于在页式 32 位地址方式下运行的 386 或 486。

介绍 386 增强模式首先有必要描述一下 80386 的体系结构和运行方式。80386 具有全 32 位的体系结构, 其指令数据, 内部数据通路, 外部数据总线及地址总线都是 32 位结构的, 在硬件上支持虚拟的段页式存贮管理方式 (由 MMU 实现)。为与 8086、80286 兼容, 80386 设置了三种运方式, 实地址方式, 保护方式和 V86 方式。实地址方式下的 80386 相当于一个具有扩展指令集 (80386 专用指令) 和寄存器的高速 8086。在保护方式下, 80386 既可以象 80286 那样以 16 位段地址工作, 又可以在 32 位地址方式下工作。V86 方式则综合了上面两种方式。在此方式下, 8086 的任务同 80386、80286 的任务可同时运行。实际上, V86 方式是一种动态方式, CPU 可在 V86 方式与保护方式之间快速重复切换。为建立起 V86 I 工作环境, 386 操作系统要运行一个或多个 V86 任务, 以建立起一个或多个虚拟机 (VM)。

虚拟机既包括硬件又包括软件。硬件提供虚存, 虚寄存器并直接执行涉及这此寄存器和地址空间的所有指令; 软件统一协调控制虚拟机的 I/O 接口、中断、异常、模拟仿真 I/O 指令。由此看来, 虚拟机既不是一个仿真器, 也不是一个象实机器那样有效的真实机器。

为了支持多个虚拟机, 在所有 VM 的底层, 有一个虚拟机管理程序 VMM, 它管理 386 机器的资源, 包括内存、I/O 设备、特权级、甚至 CPU 时间。VMM 产生并管理多个 VM, 有条不紊地将 386 资源分配给 VM。但需要说明的是 VMM 并不是操作系统的一部分。

具体实现一个 Windows 虚拟机时, VMM 在内存为该 VM 开辟一块存贮区域, 一部分用于保存处理机状态, 如一些重要寄存器的内容, 另一部分用作系统控制块, 实际上是一些内部数据结构, 用来仿真 PC 机的其它重要方面, 诸如端口, I/O 控制、DMA 控制、

视频接口、中断向量表等。这样，VM 就保存了对应用程序运行环境状态的全面描述。对 Windows 应用程序而言，由于 Windows 虚拟机的作用，它们执行起来就象在 Windows 实模式下运行一样。

尽管 80386 的全 32 位的体系结构和存贮管理部件 MMV 从硬件上为 Windows 应用程序提供了直接使用 32 位平伸内存模式的可能性。但实际上，仅用 32 位保护方式来实现 Windows 应用程序是不可能的，因为 Windows 本身就依赖于 16 位段地址的分段内存模式。当 Windows 在 386 增强模式下启动时，系统就自动为其创建一个 Windows 虚拟机，以后 Windows 的台面板程序就在这个 VM 的控制下运行。因此，任何与 Windows 有关的应用程序必须使用至少一个 16 位段地址的代码段。而不能仅用平伸内存模式的 32 位偏移地址方式。尽管有这个限制，我们还是想办法解决这一矛盾，使 Windows 应用程序主要使用 32 位的代码段和数据段。Windows 在 SDK 的动态连接库 DDL 中包含了 Winmen32.DLL 为了编程者提供了一种使用平伸内存模式的标准方法。它所提供的功能有：分配、再分配和释放 32 位内存块，把 32 位地址转换为可被 Windows 直接使用的 16 位段地址加 16 位偏移地址形式；把数据段更名为代码段，这样，在装入 32 位的数据段之后实际产生的是 32 位的可执行代码段。

二、Windows 的底层结构与运行机制

虚拟机的概念对一般按实模式方式编写的 Windows 应用程序是透明的。复杂一点的增强模式下运行的应用程序可以通过调用中断 int 2fh，明确系统的运行模式，内存配置以及其它一些关于系统本身的信息，从而可以调整自身的操作行为以更好地在多任务的环境中运行。更复杂一些的应用程序还可以知道它在一个虚拟机下运行，进而利用虚拟机的某些特性。例如，可以通过调用中断 int 33h 访问 DOS 保护方式接口 DPML，或者调用 int 2fh 利用增强模式下的其它一些功能。通过访问 DPML，Windows 应用程序可以切换到保护地址方式下。

Windows 的虚拟机管理程序 VMM 和虚设备驱动程序 VxD 都在 80386 的 32 位地址保护方式下运行，特权级为 0 级（最高特权级）。用户可以利用 DDK 中提供的 32 位地址方式下的汇编，连接程序开发自己的虚设备驱动程序以完成一些特定功能。Windows 本身同 VMM，I/O 端口，设备的驱动程序以及一些低层硬件设备的仿真程序一起，构成了一套完整的 386 增强模式下的运行系统。下面解释一下这个系统在 386 增

强模式下的运行过程。

当 Windows 在增强模式下启动时，VMM 和 SYSTEM.INI 中定义的虚设备被加载列内存，同时，系统初始化所有的 VxD 和 VMM，建立起 Windows 虚拟机并对其初始化。随后，外层虚设备驱动程序（shell VxD）开始运行 Windows，通过调用 V86MMGR 功能建立起多个 VM。为使一个 DOS VM 开始工作，shell VXD 首先调用 V86MMGR-Allocate-V86-Pages 功能为这个 VM 分配内存页面，建立此虚拟机的内存映象；接着调用 V86MMGR-Set-EMS-XMS-Limits 为 EMS 分配内存，并将这些内存页面与 EMS、XMS 的中断服务程序连接起来，这样，就形成了一个功能全面的 DOS 虚拟机。

每一个虚拟机在全局内存有一个映象，在此映象的局部存贮区中，包括中断向量表，视屏缓冲区及该 VM 的局部应用程序代码和数据。这时，除了 VM 映象外，全局内存中还包括：MS DOS 设备驱动程序，Windows 启动前加载的网络程序和常驻内存程序（TSR、Windows/386 的一部分（在保护方式下运行的部分）以及 ROM 中的 BIOS。

VM 的代码段一般在最低特权级（3 级）下运行，但 VM 在系统内存区域中有一个控制块和一个堆栈处于最高特权级（0 级）。控制块中的数据供系统 VxD 使用，用以协调控制所有虚拟机的硬设备状态。多个 VM 在 VMM 的控制下运行，每个 VM 运行的时间片由 VMM 按动态优先级的调度策略进行分配。

当一个在 VM 下运行的应用程序产生一个软中断或异常时，将引起特权级的转换，这借助于 386 的保护机构实现。控制转移给 VMM，VMM 再将中断转发给相应的 VxD 或转发给虚拟机的操作系统。例如，一个应用程序请求一个 DOS 调用（int 21h），这个软中断将引起特权级从 3 级变到 0 级，转入 VMM，由 VMM 对这个 DOS 调用进行合法性检查，然后再将此中断转发给虚拟机的 DOS，完成中断处理。

上面所涉及的许多概念和原理不仅适用于 Windows，也适用于其它在 80386 或 80486 上运行的多任务环境。

Windows 386 增强模式尚有许多其它特性区别于标准模式和实模式，上面只谈到了 Windows 底层的一些基本特征，用户可以使用增强模式下运行的 debugger 跟踪 DDK，以进一步弄清楚 386 增强模式下 Windows 运行机制的细节。

爱华AH1488D单色显示器

行扭及无显示 故障的排除

吉林白城林业学校计算机室 赵明生

一、行扭曲故障

1. 故障现象

屏幕上的文字沿行的方向扭曲, 出现类似“S”形光栅。

2. 分析与检查

根据故障现象分析, 本机采用桥式整流方式, 假如桥式整流中有一臂二极管正向电阻变大, 使波纹接近50HZ, 光栅就可能出现“S”扭曲, 一般还要伴随着扫描线稀疏的变化。其原因是纹波电压作用到垂直扫描系统后, 使光栅上一部分扫描线稀疏、另一部分稠密。另外, 滤波电容容量变小或损耗增大, 经该电容滤波后的直流电压作用到水平扫描系统中, 也会产生此类故障(此时波纹频率为100HZ, 每屏扭曲次数应为前述的2倍)。本机经替换滤波电容后故障仍然存在, 在检查整流二极管时, 发现确有一个二极管D501(IN5401)正向电阻接近70K Ω , 经更换该二极管显示恢复正常。

二、无显示故障

1. 故障现象

计算机启动后无光标和字符显示, 显示器指示灯不亮, 并烧保险丝。

2. 分析与检查

拆开显示器检查发现, 交流保险丝已熔断, 更换新的保险丝后开机又迅速熔断。经分析原因, 其一是由于负载过重而造成; 其二可能是交流回路(变压器初级回路)的阻抗太小所致。为了确定故障所在位置, 先将电源变压器次级插头拔下, 换一个新的保险丝, 开机仍烧保险, 说明交流回路一定存在问题。拆下电源变压器初级接线端, 用DT-830型数字表测量变压器初级直流

电阻为13.5 Ω , 可以肯定电阻值太小; 因为测量正常的显示器的变压器初级电阻为72.8 Ω 。这说明电源变压器初级线圈有短路现象, 使阻抗减小, 导致初级电流增大而烧保险。

因为电源变压器初级线圈在开始时可能有较少一部分发生短路, 初级电流照正常有所增加, 变压器本身热量随之增加, 逐步使漆包线的绝缘层老化, 使短路匝数越来越多, 最终使初级电流超过保险丝的容量而熔断, 使显示器无法供电而造成不显示的故障。

由于在市场上很难买到原配变压器, 所以只好买一代用品(代用品要考虑初、次级的电压及变压器的功率, 初级电压为220V, 次级电压为18V, 功率为25~30W, 同时还要考虑体积和安装位置)。更换后开机试之, 基本正常; 但显示器又出现行扭(类似上例故障)的现象, 据了解本机在出现故障前无此现象, 所以多数是因为电源变压器没有屏蔽, 使该变压器泄漏50HZ电磁干扰造成的。在通电状态下观察, 用手移动变压器使之逐步远离显示器主电路板, 此现象逐步消失。说明行扭曲正是变压器屏蔽不良所引起。解决的办法: 把原变压器的屏蔽罩拆下, 套在新变压器上, 并将开口处朝向机壳方向(向外), 再开机试之, 故障消除。

综上所述:

1. 该机造成无显示、烧保险故障的原因是由于电源变压器初级短路造成的。

2. “S”形扭曲一般有以下三种可能:

- ① 滤波电容容量减少或损耗增大;
- ② 桥式整流中有一臂二极管正向电阻变大;
- ③ 电源变压器屏蔽不良。

64

键盘的一般故障处理

云南省路西县农业银行 刘 江

键盘在使用过程中出现故障的原因可能是多方面的。除按键本身的机械性故障外，还有逻辑电路故障、焊点和金属化孔的故障。这些故障形式多种多样，需要进行分析判断，并作出处理。现将一般的处理方法介绍如下。

一、某些字符不能输入

若有某一个键符不能输入，则可能是按键失效或焊接点失效。检查时，先打开键盘，用万用电表电阻档测量接点的通断状态。若键按下时始终不导通，则说明按键失效，需修理或更换；若键按下时，接点通断正常，则说明有虚焊、脱焊或金属化孔失效。要找到故障点，可沿着印刷电路板上的印刷线路走向逐段测量，找出失效点补焊。若是金属化孔失效，如金属化孔完全脱落，则可另加引线短接。

若有多个既不在同一行，也不在同一列的字符都不能输入，则可能是列线或行线某处失效或是门电路故障（这时该列或该行的所有字符均不能输入）。此时可使用高频示波器（如 100MH）进行检测，找出故障器件或失效焊点。

二、输入与显示的字符不一致

此种故障可能是由于电路板上短路现象造成的。

它通常表现为显式变为同一列的其它字符。这时可先用万用表测量，若还不能确定故障点，则可用示波器观察。

三、按键时字符乱跳或按下一个键产生一串多种字符

此种现象通常是键盘逻辑电路故障造成的。先始终选中某一列字符，若是包含 ENTER（回车）键的一列，则产生键符乱跳且不能最后进入系统；如果是不含 ENTER（回车）键的某列某行，则可能产生多个其它字符现象。此时最好用示波器检查，找出故障的芯片。

四、光标停不住，字符输入不连续

造成这种故障的原因通常是 SPACE 空格键复位弹簧疲劳失效所致。它不断产生空格，以致于其它键不能输入，属于机械故障。遇到此情况只要用手指捏紧键帽将空格键轻轻往上拔起，此现象即可消失。此时应拿下键帽，再用螺丝刀把键座地盖片撬起来，取出在按键中起回复作用的弹簧丝，更换已失去弹性的弹簧丝，或者把弹簧丝整形恢复，重新安装好，问题即可解决。

65

广东省将于 5 月 9 日举行计算机操作人员水平联合考试

根据《中国计算机软件专业技术资格和水平考试暂行规定》的有关精神，1993 年广东部分城市计算机操作人员水平联合考试定于 5 月 9 日（星期天）举行。今年是第三次水平联合考试，经考试实施办公室、华南理工大学、华南师大、暨南大学等单位的工程师、教授研究，对原考试的有关规定，要求与《考试大纲》作了适当的修改。现将《1993 年广东部分城市计算机操作人员水平联合考试大纲》刊出，供要考取上岗合格证的在职人员或在校学生（包括普通中学、职业中学、中专、中技）参考。有关辅导资料可到报名点购买。

今年水平考试中心考场设在广州市职业技术教育中心，各市满 30 名考生可设分考场。（大纲见 93 年第 1 期）

报名时间：2 月 8 日～4 月 15 日

报名地点：广州市小北下塘西路 41 号

广州市职业技术教育中心

联系人：刘春贞、徐夏帆

电话：3314800 3314981

邮政编码：510050

广东部分城市计算机操作人员水平

联合考试实施办公室

一九九三年元月

66

标准化试卷的评判和程序

兰州电力技工学校 罗 君

随着考试向科学化、标准化的方向发展,人们正在寻找更先进、更科学的方法,以加强对考试的正确评价和分析判断。

为了使考试能及时正确地反映教学效果,必须要有一套切合实际的试题,而标准化试题就是目前国家正在尽力推广应用的方法之一。由于标准化考试涉及面广,考评因素多,统计复杂,数据量大,若用人工统计的方法则很难快速、准确的完成全部工作;然而,对于一般学校来说,不可能具备专门用于批阅试卷的机器,即是你有能力组织起来,那也只能是每年使用一、二次,不可能经常频繁的使用,造成了资金上不必要的浪费;若你不具备这种机器,我们为之推广的标准化考试题将受到阻碍。那么,能不能利用现行计算机来完成这一特定的工作呢?

以此为出发点,笔者首先着眼于目前比较普及流行的 IBM/PC 机,用 BASIC 语言编写了一个专门用于标准化试题的小批量试卷评判程序。这个程序简单明了,要求环境不高,便于推广应用,特别对于学校任课教师是非常实用的,它不但可以帮助你很短的时间内做完繁杂的判卷、打印个人成绩单等工作,而且还能打印出按成绩排序,或者按学号排序等总成绩单,以及对你所出试题的总评价直方图等几项工作。

该程序是面向不懂或略懂计算机人员的,所以,只要你拥有该程序,那么,任何人都可以按照计算机所给的提示,顺利的进行你的工作,即便是你不小心输错了,计算机也能及时地帮助你纠正错误。该程序经过近几年来的使用,效果反映良好,不但减轻了我们的劳动强度,提高了工作效率,而且,能对考试作出较为科学的、全面的评价,为且以后的教学工作打下坚实的基础。

下面是程序清单:

```
10 REM 计算机判卷程序(PJCX.BAS)
20 REM 罗君编于 1987 年 10 月,修改移植于 1990 年 10 月 18 日.
30 CLEAR
40 INPUT "请输入班级名字:";BM$
50 INPUT "请输入该班人数:";H
60 INPUT "请输入课程名字:";KM$
70 INPUT "请输入题数:";N
80 INPUT "请输入标准答案:";Y$
90 A1=LEN(Y$)
100 IF A1<>N THEN PRINT "你刚输入的标准答案与题数不符,请
```

```
重新输入!";GOTO 80
110 GOSUB 1510
120 DIM X$(H),Z(H),C(N),E(H)
130 V=0
140 FOR J=1 TO H
150 PRINT BM$;"班第";J;"个输入学生的学号是:";INPUT X$(J,0)
160 PRINT TAB(20);"成绩单"
170 PRINT TAB(18);"-----"
180 PRINT "答案比较:"
190 PRINT "题号:";
200 PRINT " 1234567890123456789012345678901234567890"
210 PRINT "标准答案:";Y$
220 INPUT "考生解答:";D$
230 A2=LEN(D$)
240 IF A2<>N THEN PRINT "该生题数不够,请重新输入!";GOTO 220
250 S=0
260 FOR I=1 TO N
270 A$=MID$(Y$,I,1)
280 B$=MID$(D$,I,1)
290 IF A$=B$ THEN C(I)=C(I)+1;GOTO 310
300 S=S+1
310 NEXT I
320 Z(J)=INT(10000*(1-S/N)+.5)/100
330 V=V+Z(J)
340 GOSUB 1510
350 GOSUB 1560
360 PRINT BM$;"班";KM$;TAB(44);"No.";J;7370 LPRINT BM$;"班";KM$;TAB(44);"No.";J
380 PRINT TAB(16);"个人成绩单"
390 LPRINT TAB(16);"个人成绩单"
400 PRINT TAB(14);"-----"
410 LPRINT TAB(14);"-----"
420 PRINT TAB(6);"学号";TAB(22);"答对题数";TAB(40);"成绩"
430 LPRINT TAB(6);"学号";TAB(22);"答对题数";TAB(40);"成绩"
440 PRINT TAB(6);X$(J);TAB(24);(N-S);TAB(40);Z(J)
450 LPRINT TAB(6);X$(J);TAB(24);(N-S);TAB(40);Z(J)
460 LPRINT
470 PRINT "标准答案:";Y$
480 LPRINT "标准答案:";Y$
490 PRINT "你的答案:";D$
500 LPRINT "你的答案:";D$
510 GOSUB 1510
520 NEXT J
530 GOSUB 1560
540 PRINT "正在按成绩排序,请稍候....."
550 GOSUB 1280
```

```

560 FOR I=1 TO H-1
570 FOR J=I+1 TO H
580 IF Z(I) > Z(J) THEN 610
590 M1 = Z(I):Z(I) = Z(J):Z(J) = M1
600 M1$ = X$(I):X$(I) = X$(J):X$(J) = M1$
610 NEXT J
620 NEXT I
630 PRINT TAB(14);BM$;"班";KM$;"按成绩排序"
640 LPRINT TAB(14);BM$;"班";KM$;"按成绩排序"
650 PRINT
660 LPRINT " "
670 PRINT TAB(4);"名次";TAB(24);"学号";TAB(40);"成绩"
680 LPRINT TAB(4);"名次";TAB(24);"学号";TAB(40);"成绩"
690 FOR J=1 TO H:PRINT
700 E(J) = J
710 PRINT TAB(4);J;TAB(24);X$(J);TAB(40);Z(J)
720 LPRINT TAB(4);J;TAB(24);X$(J);TAB(40);Z(J)
730 NEXT J
740 GOSUB 1510
750 GOSUB 1560
760 GOSUB 1380
770 GOSUB 1510
780 GOSUB 1560
790 INPUT "是否还要一份这样的成绩单(Y/N)?" :SF$
800 IF SF$ = "Y" THEN 630
810 PRINT "正在按学号排序,请稍候....."
820 FOR I=1 TO H-1
830 FOR J=I+1 TO H
840 IF X$(I) < X$(J) THEN 870
850 M1 = Z(I):Z(I) = Z(J):Z(J) = M1
855 M2 = E(I):E(I) = E(J):E(J) = M2
860 M1$ = X$(I):X$(I) = X$(J):X$(J) = M1$
870 NEXT J
880 NEXT I
890 PRINT TAB(14);BM$;"班";KM$;"按学号打印成绩"
900 LPRINT TAB(14);BM$;"班";KM$;"按学号打印成绩"
910 PRINT
920 LPRINT " "
930 PRINT TAB(4);"学号";TAB(24);"成绩";TAB(40);"名次"
940 LPRINT TAB(4);"学号";TAB(24);"成绩";TAB(40);"名次"
950 FOR J=1 TO H:PRINT
960 PRINT TAB(4);X$(J);TAB(24);Z(J);TAB(40);E(J)
970 LPRINT TAB(4);X$(J);TAB(24);Z(J);TAB(40);E(J)
980 NEXT J
990 GOSUB 1510
1000 GOSUB 1560
1010 INPUT "这里是否需要考试总结(Y/N)?" :SF$
1020 IF SF$ = "N" THEN 1060
1030 GOSUB 1380
1040 GOSUB 1510
1050 GOSUB 1560
1060 INPUT "是否还要一份这样的成绩单(Y/N)?" :SF$
1070 IF SF$ = "Y" THEN 890
1080 PRINT TAB(18);KM$;"命题成功率"
1090 LPRINT TAB(18);KM$;"命题成功率"

```

```

1100 PRINT
1110 LPRINT " "
1120 PRINT TAB(2);"题号";TAB(10);"试题通过直方图";TAB(44);"通过率"
1130 LPRINT TAB(2);"题号";TAB(10);"试题通过直方图";TAB(44);"通过率"
1140 FOR I=1 TO N
1150 G = INT(30 * C(I) / H)
1160 Q = INT(10000 * C(I) / H + .5) / 100
1170 PRINT TAB(2);I;
1180 LPRINT TAB(2);I;
1190 FOR J=1 TO G
1200 PRINT TAB(6+J);" ";
1210 LPRINT TAB(6+J);" ";:NEXT J
1220 PRINT TAB(44);Q;"%":PRINT
1230 LPRINT TAB(44);Q;"%":PRINT
1240 NEXT I:PRINT
1250 GOSUB 1510
1260 GOSUB 1560
1270 END
1280 FOR J=1 TO H
1290 IF Z(J) < 60 THEN F1 = F1+1
1300 IF Z(J) >= 60 AND Z(J) < 70 THEN F2 = F2+1
1310 IF Z(J) >= 70 AND Z(J) < 80 THEN F3 = F3+1
1320 IF Z(J) >= 80 AND Z(J) < 90 THEN F4 = F4+1
1330 IF Z(J) >= 90 AND Z(J) <= 100 THEN F5 = F5+1
1340 NEXT J
1350 ZP = INT(V * 100 / H + .5) / 100
1360 GG = INT(10000 * (H-F1) / H + .5) / 100
1370 RETURN
1380 PRINT TAB(14);BM$;"班";KM$;"考试总结"
1390 LPRINT TAB(14);BM$;"班";KM$;"考试总结"
1400 PRINT "总评成绩为:";TAB(35);ZP
1410 LPRINT "总评成绩为:";TAB(35);ZP
1420 PRINT "其中:100-90分有";F5;"人";TAB(30);"90-80分有";F4;"人"
1430 LPRINT "其中:100-90分有";F5;"人";TAB(30);"90-80分有";F4;"人"
1440 PRINT "80-70分有";F3;"人";TAB(30);"70-60分有";F2;"人"
1450 LPRINT "80-70分有";F3;"人";TAB(30);"70-60分有";F2;"人"
1460 PRINT "60分以下有";F1;"人"
1470 LPRINT "60分以下有";F1;"人"
1480 PRINT "及格率(%):";TAB(35);GG
1490 LPRINT "及格率(%):";TAB(35);GG
1500 RETURN
1510 PRINT
1520 FOR I=1 TO 52
1530 PRINT "-";
1540 NEXT I:PRINT :PRINT
1550 RETURN
1560 LPRINT
1570 FOR I=1 TO 52
1580 LPRINT "-";
1590 NEXT I:LPRINT :LPRINT
1600 RETURN

```

拷贝特大文件的方法

中国科技大学无线电系 88 级 林志钢

随着计算机软件工业的发展,现在在计算机上出现长度达几兆比特的文件已非罕见,特别是使用一些压缩软件如: ICE.EXE, LHA.EXE, PKXARC 软件包等,更会产生一些大文件,它们的长度超出一张软盘的容量,这就给拷贝这些文件带来许多麻烦。目前,人们常用 DOS 提供的 BACKUP 命令来拷贝这些文件。这种方法较麻烦,而且在具有不同版本操作系统的微机间,无法使用 RESTORE 命令拷贝文件,经过 BACKUP 处理的文件,文件名提供的信息也很少。

本文提供了一个程序 CUT.C,该程序利用二进制拷贝,将一个特大文件分割为几个小文件。在 TURBO C 环境下将 CUT.C 编译为 CUT.EXE,执行该文件会出现如下提示信息:

PLEASE INPUT SOURCE FILE NAME:

输入需要分割的文件的文件名后,又会出现下列提示信息:

PLEASE INPUT OUTPUT FILE1 NAME:

输入想生成的小文件的文件名后,接着出现,

PLEASE INPUT THE LENGTH OF FILE1:

这时再输入文件 1 的长度,以比特为单位。下面会接着要求输入将生成的文件 2 的文件名及长度,直到将大文件分割完为止。

若想将这个大文件恢复,只需执行下述 DOS 命令即可:

COPY / B FILE1+ FILE2 (+ FILE3. . .)

SOURCEFILE

源程序 CUT.C 如下:

```
#include "stdio.h"
main ( )
{ FILE * in, * out;
  char s_name[32], o_name[32];
  long int len;
  int num = 1;
  printf("PLEASE INPUT SOURCE FILE NAME:\n");
  scanf ("%s", s_name);
  if ((in = fopen(s_name, "rb")) == NULL)
  { printf("input error\n");
    exit(0);
  }
  while (!feof(in))
  { printf("PLEASE INPUT OUTPUT FILE%d
NAME:\n", num);
  scanf ("%s", o_name);
  if ((out = fopen(o_name, "wb")) == NULL)
```

```
{ printf("cannot open file!\n");
  exit(0);
}

printf("PLEASE INPUT THE LENGTH FILE%d:\n");
scanf ("%ld", &len);
for(len; len > 0; len--)
  putc(getc(in), out);
fclose(out);
num++;
}
printf("CUT IS OVER!\n");
}
```

68

~~~~~

## 来信照登

编辑先生:

您好!

我是贵刊一名忠实的读者。

今天,我迫不及待地打开我向往已久的《电脑》第六期时,却发现刊登在贵刊第 34 页署名为陈红新的《CEC-I 模拟音响》(下称“模拟”)是剽窃本人的作品!

本人的《“CEC-I”音响几则》(下称“几则”),在两年以前就发表在 90 年 10 月 25 日第 20 期的《学生计算机世界》周报上(请看所附复印件)。这令我大为震惊。

“模拟”的前面有几句话完全是“几则”的翻版。在“模拟”中,有警车、发动机、水流、蛐蛐声和“几则”完全一样,一字未改。而救护车只是对“几则”中的警车的参数稍作变动而来。

这大大伤害了我的感情,严重地侵犯了我的著作权,这样的侵权行为是不能令人容忍的。这也是违反我国的“著作权法”的。所以我要求:

1. 请陈红新尽快在贵刊上向我作出公开道歉。
2. 请贵刊索回付给陈的稿酬。

另外,作为一个热情的读者,祝您及贵刊所有工作人员新年快乐,万事如意。

并祝贵刊越办越成功!

急盼回音!

沈凌

92 年 12 月 30 日

69



# 游戏程序的消声方法

四川大学计科系 89 级应用专业 胡 术

四川江油长钢三厂计算中心 周俊晖

在某些情况下,人们希望在“无声”环境下运行游戏,由此提出“消声”的问题。本文所讨论的“消声”,是在不破坏硬件结构的前提下的软件实现手段。

在众多的游戏程序中,有一类游戏本身就有“消声”的功能,这类程序分以下三种:

1. 通过键盘上的逻辑开关达到声音的开启和闭合。由于很多游戏找不到说明书,这些逻辑开关的寻找与经验有关。一般而言 S 键、CTRL+S、ALT+S、F 热键最为常用。《淘金者》即为此类。

2. 游戏自带声音设置功能。如:“sound switch”, “no-sound”, “sound off”, “语音关闭”之类的选项,《三国演义》即为此类。

3. 找出声音信息的数据文件,如将其删除不影响程序运行,即可将其通过改名等方法“屏蔽”起来,达到消声目的。《古巴战士》即为此类,该方法对软件知识要求稍高。

但相当数量的游戏用上述方法无法奏效,目前国内常用 carl burtner 编写的“消声”程序 silence ver1.0 以消声,由于效果不佳,大多数用户已不采用。

以下提供两种比较实用的“消声法”。在此之前,先对扬声器(speaker)的发声原理作一简略介绍,讨论机型为 IBM PC/XT:

PC/XT 的音响系统包括 2.25 寸的扬声器及驱动电路,驱动扬声器发声的波形用以下方法产生:

1. 8253-5 的计数器 2(CNT2)在程序控制下产生一定频率的波形输出。

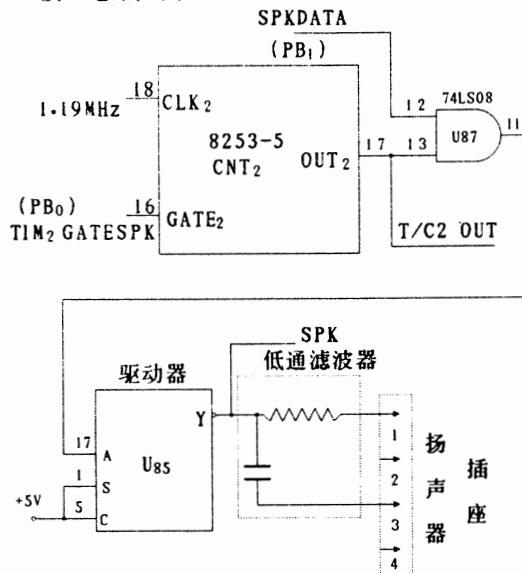
2. 用 8255A-5 的 PB0 控制 8253-5 计数器 2 的选通端 GATE2, 允许或禁止计数器 2 的计数, 控制波形的断续。

3. 控制 8255A-5 的 PB1 状态, 产生波形输出。

设置 8255A-5 的 PB1 为高电平, 8253-5 计数器 2 的输出端 out2 可以通过与门 74S08 (V87); 而当 PB1 为代电平时, 计数器 2 输出的方波不能输出通过与门 V87, 如果计数器 2 的输出 out2 设置成高电平, 则可用程序直接控制 8255A-5 的 PB1 输出, 产生方波通过 V87, 与门的输出端 V87-11 接到驱动器 75477 的输入

端 V85-7 方波通过驱动器 V85 后产生 1/2W 的驱动功率, 再通过阻容代通滤波器, 滤除高次谐波后送扬声器, 扬声器通过插座 P3 与接口电路连接。

接口电路如图:



8253-5 通道 2 口地址为: 42H, 8255A-5 的 B1 口地址为: 61H, 在游戏程序里一般可见 out 42H, AL out 61H, AL 等与声音有关的指令, 一种设想是把程序中的 out 42H, AL(E642), out 61, AL(E661) 全部换为 NOP(90H), 为保持字节数一致, 用两条 NOP 指令替换, 该设想用以下程序实现并运行通过, 用 Turbo C 2.0 编写。

```
#include<stdio.h>
#include<alloc.h>
void main()
{FILE *in,*out;
char first=0,
second=0,
*s;
s=(char *)malloc(19);
printf("Silence ver 1.00");
printf("\n\rCopyright @ Step Lab.Chengdu,China");
printf("\n\r Editor:Hoover & Aahui");
```

```
ss:printf("\n\rplease input source filename:");
gets(s);
in=fopen(s,"rb");
printf("\n\rplease input target filename:");
gets(s);
out=fopen(s,"wb");
if ((in==NULL)||(out==NULL))
    {in=NULL;out=NULL;goto ss;}
free(s);
first=getc(in);
do{second=fgetc(in);
    if ((first=='\xeb'
        && ((second=='\x61')||(second=='\x42'))))
        {fprintf(out,"%c","\x90');
         fprintf(out,"%c","\x90');
         first=getc(in);}
        else{fprintf(out,"%c",first);
         first=second;}
        }while(!feof(in));
}
```

这样就得到一个新的无声程序,该方法对《ZAXXON》,《Load Runner》,《双截龙》,《Block out》等游戏效果较佳。

若程序中对口地址使用 out DX,AL 指令,由于 DX 的内容难以确定,故不能采用上述方法。此外,上述方法对加过密或解密不完全的游戏程序也存在一定局限。

另一个方法是编写一个 08H 中断服务程序,利用 int 08H 每秒 18.2 次的申请次数,每次对端口 61h,42h 作关闭 speaker 操作,达到“消声”目的。为增加 speaker 关闭的相对时间,对 8253-5 的方式选择是其要点。

经对 8253-5 五种工作方式下的波形分析,并经实测,可知方式 2,方式 0 最佳。

TSR 程序清单如下:

```
vector-table segment at 0
    org 4*8h
time-int dd ?
vector-table ends
code segment
    org 100h
assume cs:code,ds:code,es:code
begin: jmp start
    old-08h-vector dd ?
new-08h-interrupt proc far
assume cs:code,ds:nothing,es:nothing
    push ds
    push dx
    push cx
    push bx
    push ax
    push si
    push cs
```

```
pop ds
mov al,00
out 61h,al
mov al,0b4h
out 43h,al
pop si
pop ax
pop bx
pop cx
pop dx
pop ds
jmp old-08h-vector
iret
new-08h-interrupt endp
program-end:
message db 0ah,0dh,'Silence advanced Ver1.10'
        db 0ah,0dh,'Copyright(C) 1992.'
        db 0ah,0dh,'Editor: Hoover,Aahui','¥'
message1 db 0ah,0dh,0ah,0dh,'No-sound program','¥'
start: mov ax,0
        mov es,ax
        mov dx,offset message
        mov ah,09h
int 21h
assume es:vector-table
mov ax,word ptr time-int
mov bx,word ptr time-int+2
mov word ptr old-08h-vector,ax
mov word ptr old-08h-vector+2,bx
cli
mov word ptr time-int,offset new-08h-interrupt
mov word ptr time-int+2,cs
assume es:code
mov dx,offset message1
mov ah,09h
int 21h
mov al,0b4h
out 43h,al
sti
mov dx,offset program-end
int 27h
ret
code ends
end start
```

该程序对《sopwith》、《M1 TANK》、《MOONBUG》、《ROMA》、《决战中国海》、《赛车 1,2,3 版》等游戏消声效果较佳,若将 MOV AL, B4 中 B4 改为 B0 时,对《意大利 90 世界杯》、《三国演义》效果较佳。

由于各种游戏软件中声音的音频与时间常数各不相同,所以任何单一的消声手段都不可能对所有游戏均适用。不过,若将本文提供的三类方法有选择使用,则可极大提高消声覆盖率。

以上探讨较为粗浅,仅供参考。

70



# Auto-copy病毒的检测免疫与消除

吉林电气化专科学校专计 901 班 张 研

最近, 我校的几台 286 微机发现了一种新病毒, 用 CPAV (可查 1009 种) 以及公安部发行的 SCAN3.1 (可查 500 多种) 均不能正确检测和消除。这种病毒的不但感染 .EXE、.COM 文件, 而且感染硬主引导记录, 兼有系统引导型和外壳型病毒的特点, 该病毒感染可执行文件时, 使文件增加 3072 个字节 (3K), 感染速度很快, 感染主引导记录时, 占用硬盘开始的 6 个扇区, 在各权威机构没有正式命名前, 由于病毒程序中有如下一段文字, 故笔者称其为“Auto-copy”病毒。

该段文字如下:

Welcome you!  
Auto-copy Deluxe R3.00  
(C) copy right 1991 Mr yaoi chang sha china  
No one can Beyond me

下面分几部分对 Auto-copy 病毒进行简要介绍

## 一、病毒的引入机制

### ①可执行文件中病毒的引入

当执行带毒的可执行文件时, 病毒首先获得控制权。它调用 21H 中断 54h 功能入口参数 bx = 9064h 如果 AX 返回 9064h 则表明病毒已经驻留内存, 转去执行原程序, 否则调用 4Bh 执行原文件后, 驻留内存。

### ②主引导扇区中病毒的引入

当使用带毒硬盘引导系统时, 病毒程序被硬件读至 0000: 7C00h 开始的 512 个单元, 获得控制权后, 把表示内存容量的 0000: 413 单元中的一个字减 4, 使内存减少 4K, 然后读出病毒的其它部分, 移到内存高端, 从而达到驻留的目的, 之后修改 13h 中断向量, 再读出正常的主引导记录, 并把控制权交给主引导记录。

## 二、病毒的传播机制

Auto-copy 病毒修改了 8h, 13h 和 21h 中断, 以达到控制系统和传播病毒的目的。

病毒修改了 13h 中断, 每次当系统或用户调用 13h 中断时, 都判断是不是对硬盘的操作, 如果是则读出主引导记录, 判断是否染过毒, 如果没有感染过, 则把 3K 的病毒程序写到硬盘的 6 个扇区中, 并且把正常的主引导记录写入 0 头 0 柱 2 扇区。

病毒修改了 21h 中断, 除了修改了上面提到的 54h

功能外, 还修改 34Bh 功能即 DOS 的 EXEC 功能每次执行文件时这段程序获得控制权, 对文件进行检测, 判断是否染毒, 如果已染毒则不再感染, 否则, 不论该文件是何种属性均进行感染。

## 三、病毒的表现机制

当系统时间为 5 月 4 日时, 如果执行文件则病毒发作, 在屏幕上打印如下一段信息, 并且删除这个文件, 这段文字是用密文写的信息如下:

XqR:

Wherever, I love you Forever and ever!

The beautiful memory for ours in that summer time has been recorded in the computer history.

Bon voyage, By dear XqR!

Yours 05121991 in our home

大意如下:

XqR:

无论在哪里, 我永远爱你, 我们在那个夏日的美好回忆已经记录在计算机的历史中了。

一路平安, 我亲爱的 XqR!

1991.5.21 我们的家中

## 四、病毒的免疫

Auto-copy 病毒在每次对可执行文件或硬盘主引导扇区进行传染时, 都对其进行判断, 判断是否传染过, 这此检测是凭借判断特征字完成的。

①在 .COM 文件中特征字在 103h-104h 中的两个字节里 103: 64h 104: 54h

②EXE 文件中的特征字是文件头偏移 1ch、1Dh 两字节 1ch: 4ch、1Dh: 14h

③主引导扇区中的特征字是偏移 EBH、ECH 中的两个字节, 分别为 19h、05h。

只要对上述单元进行填充, 就可以达到免疫的目的。

## 五、病毒的消除

笔者通过对 Auto-copy 病毒的剖析, 用 TURBOC2.0 编写了检测和消除病毒的程序, 其中包括对内存的检测以及对主引导扇区 COM, EXE 文件的检测与消除, 同时对 EXE 文件还有免疫作用。

(程序见本期程序盘)

71

# Tree 病毒

武汉市朴诚电子研究所 祝向叔

Tree 病毒是最近发现的一种恶性操作系统型病毒，该病毒有很强的隐蔽性和破坏性。因此，大家对它应特别注意。

用各类检病毒软件检查，都报告有 2708 病毒，如：SCAN3.0 报告为 Azusa；CPAV 报告发现 Hongkong 病毒（Azusa 和 Hongkong 均是 2708 的别名），用能够清除 2708 的 KILL35.01 去解它，结果一无所获，报告说“一切正常”。经分析后发现，该病毒并非 2708，而是一种新病毒或变种。

Tree 病毒长度为 170H 字节，它具有病毒的三大特性，表现、传染和破坏。

用带毒盘启动系统时，屏幕上有时会出现“Tree 92.3!”的字串，故命名为“Tree 病毒”。该字串显示的条件为：当时系统时钟计数单元（0：46CH）的低 6 位为全零。满足该条件的可能性很小，只有 1/64，这就是 Tree 很难被发现的主要原因。Tree 的这种表现方式同大麻病毒很类似，只不过前者变得更隐蔽了。

Tree 有极强的传染力，可以感染软盘 A、B 以及硬盘 C。感染 360KB 软盘时，Tree 将原 BOOT 程序搬到 0 面 39 道 7 扇区保存，自身藏于 BOOT 区；硬盘感染是在启动系统时进行的，Tree 复盖了原主引导扇区，仅保留了分区表信息，其它数据被丢弃，病毒可以完成硬盘的引导任务。

Tree 对磁盘数据有很大的破坏性，其破坏对象是软盘的目录数据项。病毒进入内存后，截留盗用了系统 13H 中断，监视磁盘的操作，并在病毒体内设置有一计数器，当访问一个已感染过的软盘时，就计数一次，一旦计数达到 64 次，就激活病毒破坏程序。其主要破坏动作有：将病毒体一个扇区的内容写入当前盘面 0 道 6 扇区（逻辑扇区 6）。该扇区恰好是 360KB 软盘根目录第二扇区，这就意味着根目录中第九个到第十六个文件目录项数据被彻底毁坏，从而造成文件数据的丢失，这段程序如下：

```
9FC0: 003F MOV AX, (02B3) ; 判断该盘是否已感染
9FC0: 0042 CMP (00B3), AX
9FC0: 0046 JZ 00A7 ; 已感染，转 A7
:
9FC0: 0058 MOV AX, 0301 ; 写一扇区
9FC0: 005B XOR BX, BX ; 此时 DS=CS，指向病毒头
9FC0: 005D INC CX ; 扇区号加 1
```

```
9FC0: 005E MOV PH, 0 ; 0 面
9FC0: 0060 CALL 0070 ; 调用原中断 13H 写盘
9FC0: 0063 POP SI ; 返回
:
9FC0: 00A7 INC BYTE PTR (00C7) ; 计数一次
9FC0: 00AB TEST BYTE PTR (00C7), 3F ; 达到 64 次?
9FC0: 00B0 JNZ 0063 ; 没有达到，返回
9FC0: 00B2 MOV CX, 0005
; 先指向 0 道 5 扇区，再经 5D 处加 1，变为 0 道 6 扇区
9FC0: 00B5 JMP 0058 ; 写盘
```

Tree 病毒有许多特征，利用它们就可以确定系统是否已染有 Tree 病毒：

- ① 系统启动时，屏幕上显示“Tree92.3!”；
- ② 目录滞留。这一点很象 2708 病毒，就是在高密度驱动器上对不同盘列目录，看到的却是同一目录内容；
- ③ 用 DEBUG 把 BOOT 区调入内存 100H 处，其首条指令为：

```
XXXX: 0100 JMP 01C8
④ 在 XXXX: 01B7 处数据为 (HEX):
B9 9F 88 88 CD D4 DF C3
DE CD CC E7 E7 E0 EA ED
```

这是“Tree 92.3!”中异或加密后的数据，用 0EDH 与其异或便可恢复；

- ⑤ 内存容量单元（0：43FH）被减 1；
- ⑥ 13H 中断向量被改为 MMMM：000B，在 640KB 系统中 MMMM=9FC0H

经确认盘上已感染 Tree 病毒后，应立即用干净系统盘重新启动，并着手清除所有盘的 Tree，否则，病毒一旦发作，将会给你造成惨重的损失。

清除方法：对于 360KB 软盘，用 DEVBG 将 0 面 39 道 7 扇区内容写回 BOOT 区即可；对于硬盘，由于原引导程序被丢弃，因此必须用正常的主引导程序复盖原 Tree 病毒。同时要注意保护好分区表数据，具体修复可参看有关 2708 病毒硬盘的修复方法，或用《电脑》1991 年第 4 期《用 DEBUG 解除所有引导记录型病毒》介绍的通用解毒方法也行。

# 高频教育电视有线网的信号监测方案

华南师大电教系 叶惠文

## 1 问题的提出

随着现代教育技术的发展,在我国采用电视媒体的网络已逐步使用高频有线电视网进行。它的优点是传输线路简单,传送距离远,并可在一线路中进行多路的信号传送。因此,它在教育电视网中得到较快的发展。但是,在多路信号的传送中,为了确保信号的质量,在播放中心需对各信号进行监测。如果对每路信号进行监视,则需要大量的监视器,且在监视器只能靠人的直观感觉判断,没有定量的测试结果。因此,我们设计一利用单片微机进行自动跟踪监视、测试的方案。

## 2 方案的主要设计思想

本方案利用单片微机体积小、控制能力强、具有逻辑判断能力等特点,以其为核心设计一监测系统。系统将高频电视网的输出信号接入高频调谐器,在微机的控制下进行自动选台,被选出的电视信号送中频放大后在检波电路检出电视的场强信号,该信号送至 A/D 转换器转换,将其模拟信号变为数字信息,由单片机读入。最后将测试的结果与被测的电视信号迭加,送到监视器显示。与此同时,被选电视信号的调谐振荡频率经分频送到微机接口,由微机读取还原出原调制频率。

## 3 基本工作原理及微机处理过程

本方案中,系统完成三大功能:① 高频频道选择;② 信号的场强测试及频道的频率测试;③ 监测的结果显示。

### 3.1 频道选择原理

众所周知,现代电视频道的选择都是采用电子调谐进行。本文的方案就是利用现成的电子调谐器,将其手动调谐电压改为单片微机控制。为了方便今后的使用,需预先对网路所使用的频道进行选频记存。其方法是:一旦进入预选频道状态,微机发一选通信号到 CD4042 以选择 I 频段,同时在硬件上输出一个基准调谐调整电压(24V),然后通过键盘操作进行选频,选频时微机将数据送到由两个 74LS373 和电阻组成的 D/A 转换网络(见图 3-1),将输出的数据转换成模拟信号,控制复合调整管 BG1 调整调谐电压。本文方案采用了 12 位 D/A 转换,其可控电压范围在 24-36V,即动态调谐电压范围 12V,故单位量化级的相对变化电压为:

$$12\text{ V} / 2^{12} = 12\text{ V} / 4096 = 3\text{ mV}$$

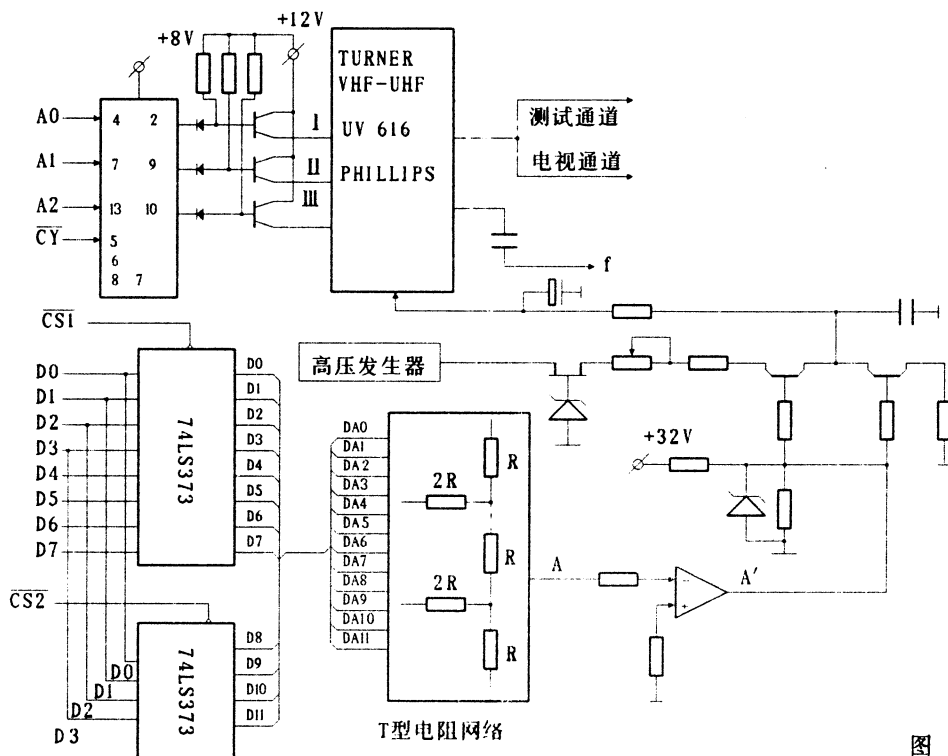


图 3-1

选频时,微机向 74LS373 数据锁存器送出选频控制电压数据,并锁存起来,直至下次更改数据为止。在 74LS373 的输出端,所输出的电压通过电阻网络形成一个调谐选频控制电压。由于输出的数据不同时作用在图 3-1 A 点上的也不同,根据迭加原理,在图中 T 型电阻网络,对其网络的输出 A 点,有:

$$\begin{aligned} V_i &= \Delta V 2^0 B_0 + \Delta V 2^1 B_1 + \cdots \Delta V 2^i B_i \\ &= \Delta V (2^0 B_0 + 2^1 B_1 + \cdots 2^i B_i) \\ &= \Delta V \sum_{i=0}^n 2^i B_i \end{aligned}$$

式中  $V = 5V / 2^{12}$   $B_i$  为 D/A 转换每级的权重系数,取值为 0 或 1。

从上式得到,改变输出到锁存器的数据,即可改变 A 点的电压,从而达到调谐选频的作用。每当微机向锁存器送出一个调谐电压的数据后,微机即开启测试通道测试输出的场强信号和场同步信号。如场强超过 56 dB 且具有场同步信号,自动停止输出调谐控制数据,等用户确认后,将调谐控制电压数据存入 EEPROM(2816A)中,供以后使用。而频段的转换是将测出的频率与每频段值比较,然后微机自动向 CD4042 发出切换命令。

## 3.2 信号场强及接收频率的测试

### 3.2.1 信号场强测试

调谐器接收到电视信号后将其分为两路输出。一路直送图象监视通道,该通道有 AGC 电路,以保证显示的图象的稳定性;另一路送场强测试通道(本路没有 AGC 电路,以保证测试的可靠性)。经中频放大后进行峰值检波,然后将检波电压信号送 A/D 转换器进行模数转换(注:由于该通道与一般场强计中频通道相同,因此原理图略去)。A/D 转换器原理见图 3-2 所示:

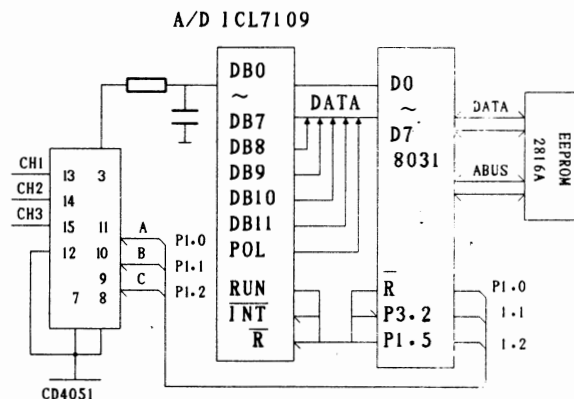


图 3-2

图中 CD4051 IC 是一个八路选择器,在这里作 A/D 转换器的多路转换开关, ICL 7109 是一个 12 位的 A/D 转换器件。根据国际标准常规的电视场强测试仪测试一般为 20~110 dB,因此,本文方案的可测范围类同。

测试时单片机 P1 口的 P1.0-P1.2 接于 CD4051 的 A、B、C 端,作多路开关的选择控制, P1.3 接于 CD4051 的选择允许端。采样开始时,微机的 P1 口输出控制信号开通多路选择器,采样结束后将其关闭。这时 A/D 转换进入了保持阶段,这里用电阻 R 和电容 C 采样保持,保持时间约为 60MS。因 ICL 7109 在 6MHz 的振荡频率时,转换周期为 20MS,所以为确保测试结果的准确性,一次采样进行两次转换,取两次转换结果的平均值作为测试的最后结果。为保证测试数据的可靠,软件设计时采用访问方式控制 A/D 转换。硬件上用微机 P1 口的 P1.5 连接 ICL 7109 的 RUN/LOAD 端,当 P1.5 输出为高电平时 A/D 转换开始, A/D 转换结束后,其 INT 线向微机的 P3.2 发出请求读取数据的信号。这样微机检测该信号后,将 P1.5 口置低电平,使 ICL 7209 处于允许读数据状态,微机即发出读命令将数据读出。

#### 3.2.1.1 数据处理过程

微机从 A/D 转换器读到数据后,进入将毫伏转换为分贝值阶段。为了提高处理速度及编程方便,这里采用查表比较法。此法是将测试范围内的 dB 值与毫伏的数字化的数据列成对应关系制成表存放在 EPROM 里,将从 ICL 7109 读到的数据与表中的数据进行比较,如比较的结果小于前一数据值而大于后一数据值,则认为前一数据对应的 dB 值为所测值。

由于高频头和中放通道有 60 dB 放大(可通过调试中放通道使达到合适放大值),因此当被测信号高于 60dB 时会使通道处于饱和状态,影响测试的准确性,所以将测试范围分为两档,即 20~60dB 档和 60~100dB 档。测试时如 A/D 转换器的溢出标志位为“1”,则微机自动转置 40 dB 衰减再进行测试,还原 dB 值时再加上该系数。

#### 3.2.1.2 毫伏数字化数据与 dB 值关系表的计算方法

据上述,因通道本身具有 60dB 的增益,故从 A/D 转换器读到的数据不是所测实际值,不能直接还原成 dB 值,需转换。如实际场强为 20dB 时,经 A/D 转换送微机的值是 80dB 时的毫伏数据,为此我们制表时将 80dB 的毫伏数据与 20dB 对应,其余照此类推。此外,又因 A/D 转换器是 12 位,本设计中按排满量程为 2820 MV,其量化单位值为:

$$2820\text{mv} / 2^{12} = 2820 / 4096 = 0.7\text{mv}$$

故查表 1(表略)可以算出 90dB 的毫伏数据是:

$$31.6 / 0.7 = 45\text{D}$$

为了提高查表速度,本方案采用单字节存毫伏数据。虽然, A/D 转换器是 12 位,但从表 1 中可发现每 dB 值之间不仅存在非线性且每 dB 值对应的毫伏都有一定的范围,并随 dB 值越大其之间的范围也越大。利用这特点,将小于 256mv 的 dB 值定为一组,而大于 256mv 定为另一组。在制表时对超出 8 位数据,我们去掉低四位,保留次高位和高位,即 4~11 位,并将该 8 位合成在一字节内。而当 A/D 转换结果高出 8 位时微机也自动将低四位去掉,保留高 8 位,合成一字节,例如:

109 dB 对应的毫伏是 282mv

$$282 / 0.7 = 403\text{D} = 193\text{H}$$

110 dB 对应的毫伏是 316mv

$$316 / 0.7 = 443\text{D} = 1\text{BH}$$

去掉低四位后,其结果是: 109dB 对应 19H

110dB 对应 1BH

去掉低四位后,109dB 为  $9 \times 0.7 = 6.3\text{mv}$

110dB 为  $11 \times 0.7 = 7.7\text{mv}$

而这两 dB 值之间的毫伏范围是:  $316 - 282 = 34\text{mv}$ , 而去掉低四位后合成的 8 位数据其之差为  $(1\text{BH} - 19\text{H}) \times 16 = 3.2\text{mv}$ , 因此去与不去低四位数据对测试结果影响不大。就算所去掉的低四位以最大的误差计算其产生的误差是:  $0\text{FH} \times 0.7 = 10.5\text{mv}$

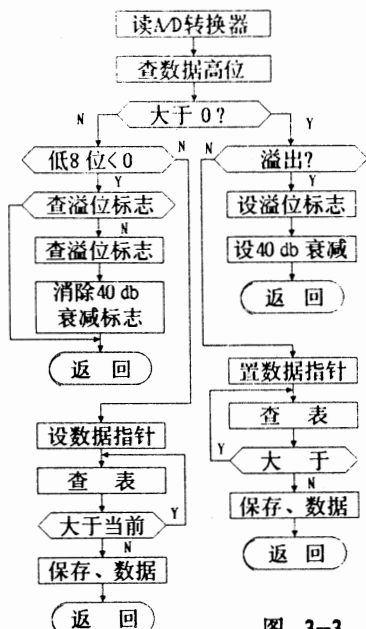


图 3-3

基于上述思想,制出的关系表(表略)

根据上述的关系表,将其固化在 EPROM 中,测试时可根据从 A/D 转换器所读得的数据与 EPROM 里的数据比较。若读入的数据是低 8 位,而高 4 位为 0,则表的指针设定在 20~45dB 范围,反之指针设定在 45~70dB 范围。由于查表分两段进行,且每段的所查的数据范围不大,所以编程采用顺序查找对速度影响不大。

程序处理流程如上:

### 3.2.2 电视射频频率的测试

目前电视信号的射频频率范围一般在 46MHz~899MHz,如用微机直接测试其频率是不可能的。因此需对接收到的高频频率进行分频,以得到适合的频率供微机读测。本文方案采用 PHILIPS 出产的专用高频头,它内部本身带有分频系数为 256 的分频器及其输出口,经它分频后得到的频率为:  $46 \sim 899\text{MHz} / 256 = 0.17 \sim 3.3\text{MHz}$ 。此信号在经 400 分频然后送到单片机读取,因此总的分频系数为:  $256 \times 400 = 102400$ 。分频后的频率信号微机采用 0.1S 的闸门读取,还原时将其读到的频率乘上总分频系数便得到被测信号的本振频率,要得到真正的被测信号的调制射频频率,需去掉差频时的 39MHz 的中频频率,所以微机要进行减 39MHz 中频频率处理。由于频率测试采用分频加运算方式,故测得的频率数据的正确度与分频系数及闸门时间有关。本文方案实际测试证明其误差为:  $\pm 15\text{KHz}$  这对电视信号的监测已经足够(一般有频率测试功能的场强仪测试误差为:  $\pm 100\text{KHz}$ )。分频计数器与微机接口原理图见图 3-4 示。

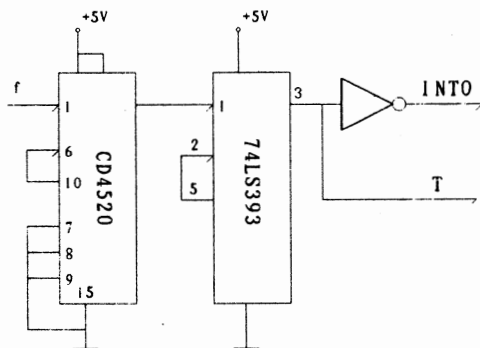


图 3-4

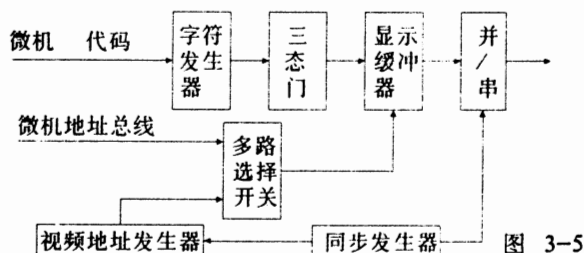
图中高频头送出的分频信号经 CD4520 和 74LS393 组成的 400 分频器分频后被接到单片机的 P3.3 (INT1) 口,该口对输入的信号进行下降沿判

别,一旦信号的下降沿到来,即置定时器和计数器进入工作状态,以保证计数的开始是在信号的上升沿进行。P3.4 置为定时器,为计数闸门定时; P3.5 设定为计数器。计数时定时器开始定时,当定时到 0.1S 时产生中断,设置计数器停止计数。此后微机将计数的数据读出进行运算处理,得出被测电视信号的射频频率。

### 3.3 被测的视频信号与测试结果的混合显示

### 3.3.1 显示数据的视频信号形成原理

为使测试的结果显示在电视屏幕上,需将显示的数据形成视频信号,将微机处理的并行信息变为分时输出的串行信号,其原理见框图 3-5 所示。



图中的字符发生器,可产生 0~9、A~F 等字符供本系统使用,字符点阵为  $16 \times 16$ 。显示时微机将测试结果的数据信息译成对应字符的代码,然后再根据该代码从字符发生器中取出相应字符的点阵数据送到屏幕显示缓冲区。存放在屏幕缓冲区的字符点阵数据由视频地址发生器在视频显示期间将数据读出,送至并/串转换器锁存起来,最后用 8MHZ 的移位信号将显示的象素分时移出形成视频信号与被测电视信号迭加。

### 3.3.2 测试显示信息与被测电视信号的同步

要实现将上述的字符与电视信号同步迭加,需有一同步起止发生器。为使硬件处理简单可行,本文采用被测信号的行、场同步控制视频地址发生器计数与停止,以实现同步。其基本原理见图 3-6。

图中视频地址发生由两个计数器组成: 一个为字地址计数, 由 1/2 74LS393 完成, 每  $1\mu\text{s}$  产生一个地址, 每行产生 16 个。另一个为排地址计数, 由 74LS393 完成每  $64\mu\text{s}$  产生一个地址, 一场产生 64 个。当视频信号经同步分离电路处理后分离出行、场两个同步信号, 通过行、场脉冲边沿对边整形器 74LS74 的 D 端, 其输出分别连接与两个地址计数器的 cr 端, 此外, 行整形信号经反相还送至排地址计数器的 PC 端作计输入脉冲。在行同步到来时刻, IC1 的 D 端为高, 在 8MHz 的脉冲作用下使其输出的 Q 端也为高, 用该信号控制字地址发生器 74LS393 的 CR 端使其处于计数清零和禁止状态, 同步结束时, 74LS74 的 D 端

为低, 尽管其 PC 端脉冲不断输入, 而 Q 端输出还是低, 时序波形如图 3-7 所示。

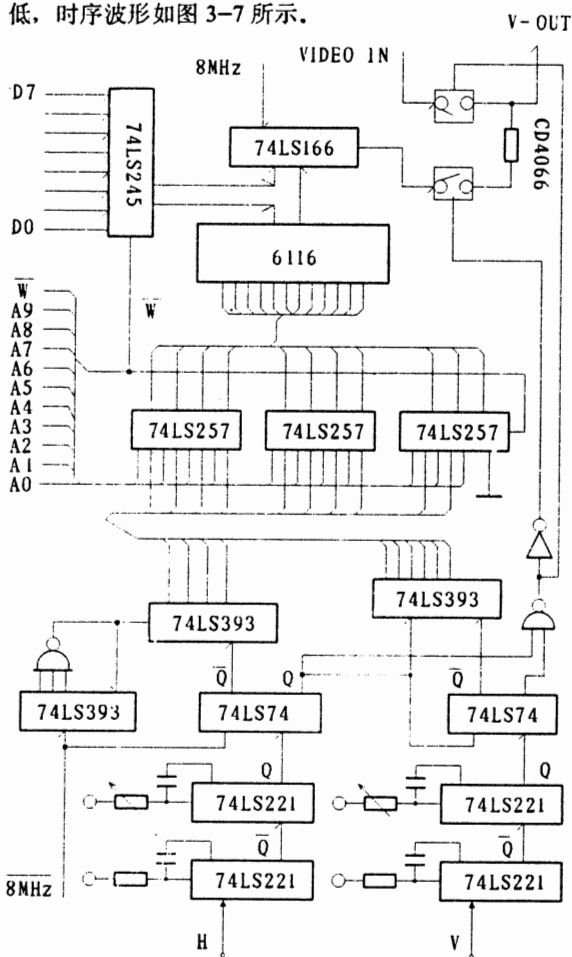


图3-6

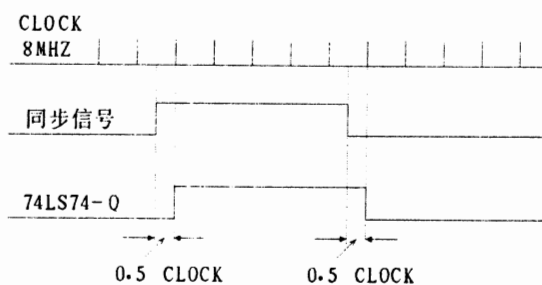


图 3-7

从图看出用该方式同步两像素点在垂直方向产生的边沿误差为 $\pm 0.5$ 时钟周期,即 $0.125\mu\text{s}$ ,因此不会影响字符的正确显示和收看。同样,排地址的同步是将行同步信号作为 IC2 74LS74 的 PC 端输入脉冲,场同步信号送于其 D 端,在场同步到来时,使 Q 端为高,将

排地址计数器清零,待场同步过后才重新计数。

### 3.3.3 字符信息的传送与显示

由于屏幕缓冲区既是微机的内存又是显示缓冲区,故受到两者的访问。为使两者使用时不产生竞争,采用74LS257多路选择器完成数据传送和显示扫描。为保证测试的信息及时到缓冲区,系统给予微机数据传送为优先级,为实现这要求,用微机的写信号作74LS257的选择信号,当有数据送往屏幕时,写信号由高变低,使多路选择器输入接向微机地址总线(如此刻正在输出视频地址都被微机中断);与此同时,用选屏幕缓冲器的片选信号关闭并/串转换器74LS166的输出,使字符信息准确传送屏幕缓冲区,又不干扰屏幕的显示。

### 3.3.4 字符显示窗口的形成

为了使显示的字符清晰,避免白电平干扰,所形成的字符电视信号不是直接与被测的电视信号混合,而是用窗口方式显示。下图是利用行、场同步信号作为触发信号,使用两个74LS221单稳电路形成一个窗口信号,电原理图如图3-8。

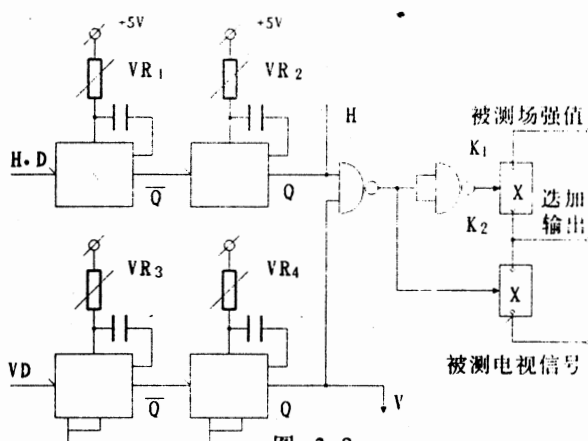


图 3-8

调节单稳的VR电阻,使单稳延迟时间,即可改变窗口的大小,将该信号经反向处理形成一对互补的信号分别控制电子开关CD4066,在窗口信号没来时,开关K1开通,开关K2关闭,被测的视频信号通过,当有窗口信号时,开关K1关闭,开关K2开通,这时如有字符信息则通过该电子开关输出到显示器,从而实现电视信号与测试信息同步叠加,显示于屏幕上。

## 4 结束语

目前我国,除学校建立了高频教育有线电视网外,各乡、镇企业以及县市都建立了大量的有线电视网(台),在这些单位如使用本方制成的监测系统,不仅可以节省大量的设备,而且能定量监测各路信号的场强以及及时了解各频道的偏移,对于提高电视的播放质量有着十分重要的价值。

73

## 投稿须知

随着洪亮的新年钟声,《电脑》杂志步入1993年,《电脑》能有今天的成绩,全靠《电脑》所有作者的辛勤耕耘和广大读者的支持。在读者的赞扬声中,我们感到欣慰。在送旧迎新之际,编辑部代表所有的读者,向我们的作者表示深深的感谢。同时殷切希望他们在新的一年里为我们生产出更好的作品。

为了把《电脑》办得更富有生气,更受欢迎,也为了加快稿件的处理速度,提高工作效率,也为了让广大读者更方便地阅读《电脑》,更快地领会到《电脑》的精髓所在,敬请各位读、作者留意下面几点:

- 1、来稿最好附文本及程序(如文中有程序)磁盘。编辑部将优先采用附有文本和程序磁盘的稿件。
- 2、稿件请注明栏目,如果现有栏目不太合适,可以自拟一个栏目或注明“待定栏目”。建议附上文章题目的英语译文。
- 3、“论坛”主要刊登议论性的文章,文章范围、内容、题材不限。
- 4、凡是新的东西,不管是硬件还是软件,均放在“NEW”栏内。
- 5、“IDEA”则是一些“主意”、“想法”等,可以是成熟的,也可以是“异想天开的”,此栏旨在给人以启发。
- 6、“电脑与法律”主要介绍有关知识产权的法律,软件保护以及侵权界定等知识。
- 7、为了维护读者的利益,请不要一稿多投。来稿恕不退回,请作者自留底稿。投稿两个月后未收到用稿通知,作者可自行处理。
- 8、来稿请注明作者详细通信地址,邮政编码和电话号码。

74

## 小 启

1、为广读者能及时获得本杂志所刊出的程序,从93年第1期起,每一期的《电脑》都配有程序盘一张,程序盘收集了整期杂志的程序文本,需要程序盘的读者,请汇15元(邮费及磁盘成本费)到本杂志编辑部并注明索取哪一期的程序盘及收件人姓名、地址;但请勿把钱夹在信封内寄来。

2、93年第1期起,文章的结束符用  $\odot n$  表示;数字n表示文章的序号和程序清单的文件名。

# 一个大型可编程控制器电路分析

华南师大微电子所 徐巨善

【内容提要】可编程控制器（简称 PC）是专门为工业环境下应用而设计的微型机系统。按照微机内部存储的程序，执行逻辑运算、顺序控制、定时、计数和算术操作等面向用户的指令，并通过数字式或模拟式输入输出，控制各种类型的机械装置或生产过程。PC 特别易于工厂技术人员所掌握使用，并在工厂环境下可靠运行。日本三菱电机公司的 K3N 可编程控制器属于大型 PC，K3N CPU 为其中央处理器模板。下面对 K3N 可编程控制器的总体结构，主要对 K3N CPU 模板的结构原理进行分析说明。

## 一、概述

### 1. K3N 可编程控制器的总体结构

K3N 可编程控制器（简称 K3NPC）主要由 K3N CPU 模板、输入输出模板、编程器、电源和外围设备等组成，如图 1 所示。

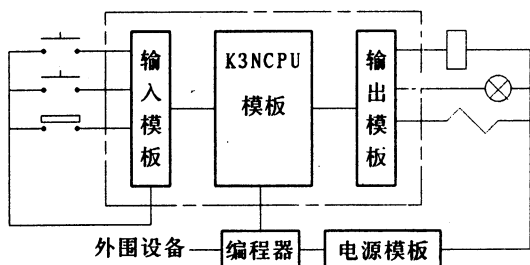


图1 K3N可编程控制器总体结构框图

在工业生产设备及生产过程中，存在着各种开关信号，例如计量槽液位到某一位置，温度在某一数值，水洗一定时间，压力到一定程度，或者机械运动到某位置时，可以用液位开关，温度、时间、压力传感器，或者行程开关等等进行检测，并用其触点的开闭来反映各被测量的变化。这些电器的触点以及控制台上各种控制按钮、开关信号，通过输入点到达 PC 的输入模板。

K3N CPU 模板根据输入条件及生产设备的要求，将输入点的信号，经输入模板各输入通道及输入选择控制电路，读入 K3N CPU 模板，并且按照用户程序进行运算处理，将处理后的结果送输出模板，经输出选择控制电路和输出通道，将信号送到被控对象，如继电器线圈、小型电磁阀和接触器线圈等等，达到控制电机转动、信号灯亮灭、监视并实现预定的生产工艺流程。

### 2. K3NPC 的主要技术参数

根据 K3N CPU 模板的硬件结构，该 PC 的主要技术参数如表 1 所示。

表 1 K3N CPU 主要技术参数

| 参数名称          | 数量   | 表示方式      |
|---------------|------|-----------|
| 输入点 (X 继电器)   | 2048 | X0-X7FF   |
| 输出点 (Y 继电器)   | 2048 | Y0-Y7FF   |
| 中间继电器 (M 继电器) | 1000 | M0-M999   |
| 特殊继电器 (M 继电器) | 24   | MA00-MA23 |
| 定时器           | 256  | T0-T255   |
| 或计数器 (T/C)    | 192  | C0-C255   |
| 标志位 (F)       | 192  | F0-F191   |
| 常数 (K)        | 64   | K0-K63    |
| 数据继电器 (D 继电器) | 1024 | D0-DA23   |

## 二、K3N CPU 模板的结构原理

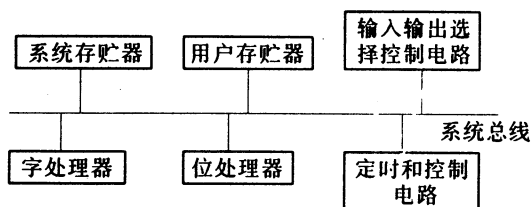


图2 K3N CPU模板基本结构框图

K3N CPU 模板是 K3NPC 的主板，其基本结构如图 2 所示，主要由字处理器、系统存储器、用户存储器、位处理器、定时和控制电路、以及输入输出选择控制电路等组成，它是一种具有字处理器、位处理器和协处理器的中央控制器系统。图 3 为 K3N CPU 的逻辑结构框图。

由图 3 K3N CPU 逻辑结构框图可见，除了虚线框表示输入模板、输出模板和编程器 1、编程器 2 之外，其余都为 K3N CPU 模板的组成部分。根据基本结构框图的组成，现将各部分的结构原理进行分析说明。

### 1. 字处理器

字处理器是 K3N CPU 模板中的主处理器，主要包括中央处理器 Z80CPU、协处理器 C8231 等部分。Z80CPU 用芯片 Z80B，其系统时钟为 5MHz。协处理器是为了提高数据的处理速度，是为 Z80CPU 服务



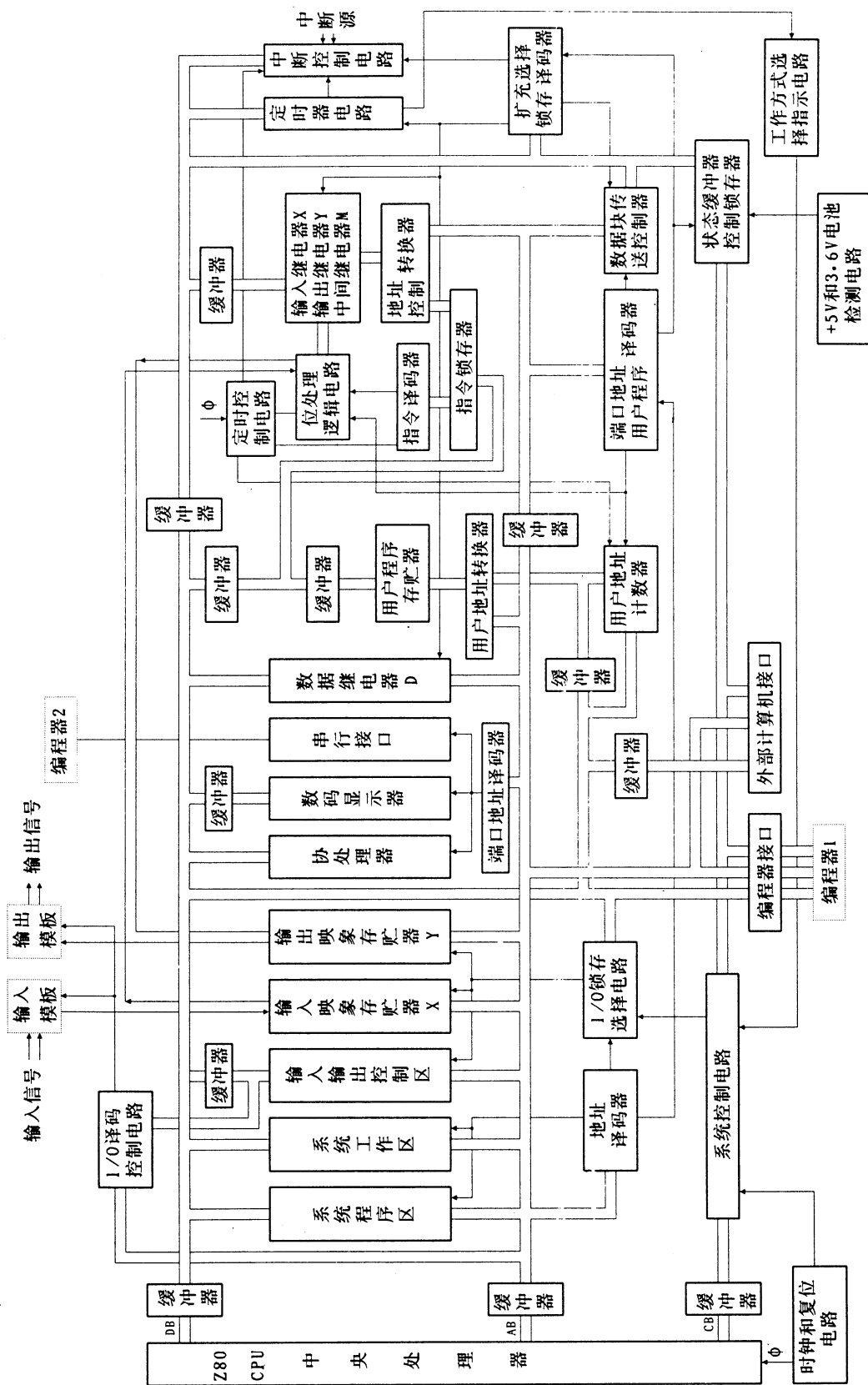


图3 K3NCPU逻辑结构框图

的。位处理器是为执行位处理用户指令的硬件组成部分,也是为了提高 PC 的工作速度。字处理器的功能是统一管理编程器接口、内部定时器、内部计数器、监视扫描时间、处理字节指令、控制系统总线、进行自诊断、协调控制位处理器、协处理器及管理输入输出等等。

Z80CPU 对其所属存贮器 64K 寻址空间,除了地址 2800-37FFH 用作 I/O 端口之外,其余为存贮器寻址空间,其分配如表 2 所示。其中 Z80CPU 对 X、Y、M、D 继电器的寻址由扩充选择锁存译码器来选择,对输入输出映象存贮器 X、Y 和输入输出控制区的寻址由 I/O 锁存选择电路来决定。

Z80CPU 通过检测状态缓冲器,可以检测 K3NCPU 的工作状态为“运行”、“步进”或“停止”;判断执行位处理指令或非位指令中断请求;取得位处理器输出的结果;检测用户存贮器保护状态及电池电压是否正常;同时还实现对 M、D 继电器等数据进行保护,以免系统复位时丢失。

表 2 存贮地址及 I/O 端口地址分配表

| 地址        | 存贮器名称及型号                              |                  |
|-----------|---------------------------------------|------------------|
| 0000-1FFF | 系统程序存贮器                               | 27128            |
| 4000-5FFF | 系统程序存贮器                               | 27128            |
| 2000-27FF | 系统数据存贮器                               | M58725           |
| 7000-77FF | 系统数据存贮器                               | TC5517           |
| 6000-67FF | 用户数据存贮器<br>(X、Y、M、D 继电器)              | HM6116<br>TC5517 |
| 6800-6FFF | 输入输出控制存贮器                             | M58725           |
| 7800-7FFF | 用户数据存贮器<br>(输入输出映象存贮器)<br>(输入输出控制存贮器) | HM6147<br>M58725 |
| 8000-BFFF | 用户程序存贮器                               | 2764             |
| C000-FFFF | (备用)                                  |                  |
| 2800-37FF | I/O 端口地址                              |                  |

协处理器 C8231 用作一台外部设备,在 Z80CPU 的控制下,可独立完成 16 个位定点,32 位定点及浮点算术运算,以及三角、对数等函数运算。

## 2. 系统存贮器

K3NCPU 的系统存贮器主要包括系统程序存贮器(区) 27128 和系统数据存贮器(区) M58725、TC5517。系统程序存贮器的内容主要有三个部分:

①系统管理程序,其功能是运行管理,控制 PC 时间上的分配,即输入、输出、运算、自检及通信;存贮空间管理,生成用户环境,规定各种参数,把程序的有效地址转变为用户可直接使用的元件;系统自监,包括系统出错检验,用户程序语法检验等等。

②用户指令解释程序、编辑程序、K3NCPU 采用助记符,即语句表语言,其位操作指令直接由硬件执

行,而非位操作指令由用户指令解释程序解释执行。语句表通过编程器进行程序的编辑调试,对用户程序进行读出、插入、删除及修改等等。

③系统调用及标准程序模块,这是由许多独立的程序块组成,各自完成不同的功能,有完成输入输出、传输及特殊运算等等。

系统数据存贮器存放用户工作环境,规定的各种参数及程序的有效地址,例如 PC 的工作状态参数,用户数据继电器的选择,数据块传送的首地址及长度,外围模板的类型编码等等,是系统程序软件应用时不可缺少的随机数据。

## 3. 用户存贮器

用户存贮器是存放用户程序和用户数据的存贮器。用户程序是用户针对控制要求的程序。K3NPU 由语句表方式编程,在系统程序控制下顺序执行以完成控制任务。

K3NCPU 用户程序存贮器由两片 2764 构成,程序长度为 4.2K 程序步,占用 8.4K 字节,需要时可扩充到 16K 程序步。Z80CPU 通过编程器可直接对用户程序存贮器进行访问。在正常生产时用户程序的寻址由用户计数器来进行。当某一程序步开始执行时,Z80CPU 将该程序步用户程序指令首地址装入用户地址计数器,其中高 8 位地址的装入命令使用户程序存贮器读出有效,用户程序指令进入指令锁存器。

用户数据存贮器是存贮用户数据的存贮器,主要由输入输出映象存贮器 X、Y (HM6147),输入输出继电器 X、Y (HM6116),中间继电器 M 和数据继电器 D (TC5517) 等组成。用户数据有位数据和字节数据两种。PC 输入输出的控制量为位数据,定时、计数运算中使用的是字节数据。位数据表示开关量或状态等。在执行用户程序前,K3NCPU 将用户数据(输入信号为位数据)经输入模板读到输入映象存贮器 X,再从输入映象存贮器读出并写入 X 继电器,然后执行用户程序。在用户程序执行过程中,即使输入信号发生了变化,输入映象存贮器 X 的数据不会改变。而输入继电器 X 的数据,在用户程序执行时可能发生变化,并可能一次或多次进行,直到用户程序执行完一个周期。同时中间继电器 M 会存贮运算时的中间结果。随着用户程序的执行将控制信号不断送到输出继电器 Y。当用户程序执行完一个周期后,输出继电器 Y 的数据全部连续送到输出映象存贮 Y,进而通过输出模板,将输出信号送到控制器件。所以实际输出控制状态只有在用户程序执行完一个周期后才可能变化,而实际控制状态是否改变,由输入状态是否改变来决定。因而,PC 的控制过程就是输入处理、程序执行处理和输出处理这一系列

操作的不断循环。另外,在定时、计数等运算中,用户所需的字节数据存放在数据继电器 D (TC5517) 之中。

#### 4. 位处理器

K3NCPU 用户指令有字节指令和位指令两类,用户指令执行时,指令译码器译出的既有字节指令,又有位指令,字节指令由 Z80CPU 执行,而位指令由位处理器执行。K3NCPU 位处理器包括位处理逻辑电路、定时控制电路、指令译码器 MB1732 和指令锁存器 74LS74 等部分。位处理逻辑电路和定时控制电路主要由累加器 74LS198、指令逻辑控制门电路和五相时钟发生器 74LS164、指令控制寄存器 74LS74 等组成。

用户程序执行时,首先由 Z80CPU 将用户指令操作码的低 8 位和高 8 位地址送到用户地址计数器,其中高 8 位地址装入命令使指令控制寄存器置位,使用程序存储器读出用户指令操作码,并送到指令锁存器高 8 位。随着定时控制电路中五相时钟的控制,用户指令操作数到达指令锁存器低 8 位。这时,指令译码器将指令操作码译出 14 位控制信号送位处理器的指令逻辑控制门电路。同时译码控制信号和指令锁存器送出地址、控制信号,经地址控制转换器到 X、Y 或 M 继电器。如果用户指令为位操作指令,则位处理逻辑电路将处理后的位数据存入累加器或 Y、M 继电器。同时,定时控制电路准备好继续执行下一条用户指令。如果用户指令为非位指令,则指令译码器的控制信号,经定时控制电路,向中断控制电路 M5L8259 发送中断请求 IR3,并复位指令控制寄存器。Z80CPU 进行中断服务,执行用户指令解释程序,完成非位指令的执行。接着,Z80CPU 重新向用户地址计数器装入下一条用户指令的首地址,再启动指令控制寄存器,继续执行用户指令,直到用户程序执行完毕。

#### 5. 定时和控制电路

K3NCPU 的系统时钟由 10MHZ 晶体振荡器产生,输出 5MHZ 时钟  $\phi$  和  $\bar{\phi}$ ,供 Z80CPU 及系统各部分使用。其中位处理器的五相时钟频率为  $\phi/15$ ,定时器电路为  $\phi/16$ 。

Z80CPU 输出的控制信号,结合工作方式选择指示电路,由系统控制电路产生输出总线响应和非总线响应两组控制信号。总线响应主要为外部计算机接口时使用。通常,在 Z80CPU 控制下,系统控制电路产生读写控制信号  $\overline{RD}$ 、 $\overline{WR}$ 、 $R/\overline{W}$ ,存储器、I/O 请求信号  $\overline{MEMRQ}$ 、 $\overline{IORQ}$ ,中断请求和响应信号  $\overline{INT}$ 、 $\overline{INTA}$  及等待信号  $\overline{WAIT}$  等等,供系统各部分使用。

定时器电路 (D8253),其输出 OUT0 用作中断控

制电路 (M5L8259) 的中断请求信号 IR1, OUT1 作运行等待指示灯闪烁及状态信号。定时器 D8253 和中断控制器 8259 由 IORQ 选择端口所决定。中断源除了非位操作指令中断请求 IR3 和 OUT0 的 IR1 之外,其余为编程器 1 接口、串行接口 D8251 (编程器 2) 及外部中断请求。

#### 6. 输入输出选择控制电路

输入输出选择控制电路主要用来确定输入输出模板的寻址结构以及内部数据的传输。K3NCPU 模板与输入输出模板交换数据,即数据从输入模板出并存入输入映像存储器 X,数据从输出映像存储器 Y 读出送到输出模板;在 K3NCPU 内部,从输入映像存储器 X 读出并存入 X 继电器,及数据从 Y 继电器读出并存入 Y 映像存储器。

K3NCPU 输入输出选择控制电路主要由 I/O 锁存选择电路、输入输出控制区 (M58725)、I/O 译码控制电路及数据块送控制器等部分组成。当 K3NCPU 与输入输出模板交换数据时,I/O 锁存选择电路用来选择输入输出映像存储器 X 或 Y,输入输出控制区数据送 I/O 译码控制电路,选择相应的输入输出模板,并控制数据的输入输出通道。当 K3NCPU 内部传送数据时,I/O 选择控制电路选择输入是映像存储器 X 和 X 继电器,或者选择 Y 继电器和输出映像存储器 Y,并控制数据的流向。数据块传送控制器用来控制内部数据传送成块进行,在地址 7800-7FFF 范围内,调用 Z80CPU 执行空操作指令子程序来实现。

#### 三、结束语

可编程控制器在国际上工业自动化领域中已广泛应用,在我国引进的生产设备中也相当普遍,自行研究、设计和应用也迅速增加。本文介绍了大型可编程控制器的总体结构,特别是对核心部件 K3NCPU 模板提供了详细的逻辑结构框图,并作了具体说明。这对有关工厂技术人员正确了解、掌握和使用可编程控制器有所帮助,而对有关研制中小型、以至大型可编程控制器的科技人员具有积极的参考价值。

本文是作者与刘相安副研究员一起全面解剖分析广宁葡萄糖厂有关 K3N 可编程控制器基础上完成的,现在已有详细的逻辑图纸和分析资料。在此,对刘相安付研究员和有关老师,以及工厂领导、技术人员的支持和帮助深表谢意。

#### 参考文献:

- (1) 钟肇新 彭侃. 可编程控制器原理及应用. 广州:华南理工大学出版社,1991.10-29
- (2) 黄明琪 冯济强. 微电脑可编程序控制器实用原理及程序设计. 贵州:贵州科技出版社,1992.24-40

# 西文状态下的汉字彩色显示

湖北宜昌市财贸学校 周苏峡

**【提要】** 本文介绍了在西文方式下利用 C 语言访问 16×16 和 24×24 点阵字库, 在图形屏幕上显示彩色汉字的一种方法, 从而能在西文方式下方便地对作出的图形进行彩色汉字标注, 解决了 C 标准输出函数只能输出一种 (16 点阵宋体) 单色汉字以及在汉字方式下汉字与 C 作出的图形不能同时显示的不足, 并给出了程序清单。

## 一、引言

读者也许有体会, 用 C 语言标准输出函数 (printf 或 puts 等) 输出汉字有以下不足:

首先, C 语言显示汉字须进入图形方式, 而图形方式不支持 cprintf、cputs 等函数, 所以显示不出彩色汉字, 这对制作彩色图形画面不能不说是一个遗憾。

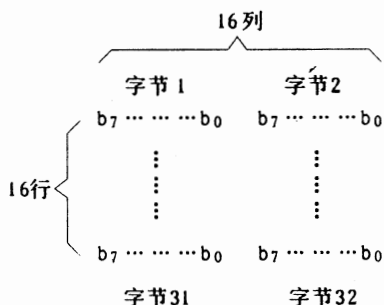
其次, printf 函数的输出位置限于文本坐标 (25×80), 无法按图形坐标在屏幕上任意点输出汉字。另外, printf 函数须在汉字系统下才能显示汉字, 但一些汉字系统对某些西文软件多少有些副作用。如 2.13H 汉字系统下, C 绘制的图形就不能与 printf 输出的汉字同时显示。所以对于只有汉字显示而没有汉字输入的程序有时并不想在汉字方式下执行。而且 printf 只能显示 16 点阵宋体汉字, 字型和大小均不能随意改变, 如想显示出更加漂亮的 24 点阵的楷、宋、仿宋、黑体等汉字字型就力所不及了。

鉴于上述原因, 本文介绍一种利用 C 语言在西文下直接访问汉字库显示汉字的方法。该方法能对作出的图形标注彩色 16 点阵 (宋体) 和 24 点阵 (宋、仿宋、黑、楷体) 汉字, 还可根据需要对它们进行放大或变形等处理, 从而显示出丰富多彩的艺术字型, 为输出的图形画面锦上添花。

## 二、汉字字模的结构

### 1、16 点阵汉字的结构

16 点阵字库中, 每个汉字字模占 32 个字节, 其点阵信息按行顺序存放。图 (一) 为一个汉字字型的 16×16 方阵的排列情况。



图(一) 16×16点阵字模排列

16 点阵库中每个汉字字模的首字节可通过下式定位 (寻址):

$$((\text{区号}-1) * 94 + \text{位号}-1) * 32$$

其中: 区号 = 汉字内码高字节值-160

位号 = 汉字内码低字节值-160

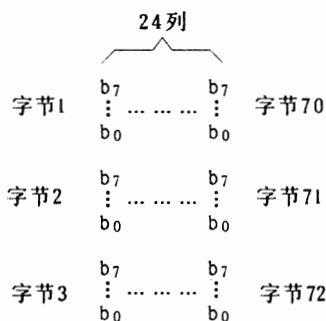
### 2、24 点阵汉字的结构

24 点阵字库中, 每个汉字字模占 72 个字节, 其点阵信息按列顺序存放。图 (二) 为一个汉字字型的 24×24 方阵的排列情况。

24 点阵字库中每个汉字的首字节用下式定位:

$$((\text{区号}-16) * 94 + \text{位号}-1) * 72$$

其中区、位号求法与前面相同。



图(二) 24×24点阵字模排列

## 三、对点阵信息的处理

### 1、16 点阵字模的处理

根据图 (一), 程序中可用三重循环, 外层分别控制 16 个行, 中层控制每行中的两个字节, 内层分别控制每个字节中的 b7-b0 八个位, 判断每位是 1 或 0 决定写点还是不写点, 从而可在屏幕的相应位置上显示出字型。

### 2、24 点阵字模的处理

根据图 (二), 也用三重循环, 外层分别控制 24 个列, 中层分别控制每列中的三个字节, 内层控制每个字节中的八个位。

### 四、往屏幕上写点

在 IBM PC 系列微机及兼容机上, 一般可用两种方法写一个点到屏幕上:

#### 1、利用 ROM BIOS

这种方法的优点是显示速度相对较慢，但显示速度相对较慢。例如下面的函数 `writeln` ( ) 就是用该方法将指定颜色为 `color` 的点写到第 `y` 行、第 `x` 列的位置上。

```
writeln (int x, int y, int color)
{
    union REGS r;
    r.h.ah = 12;
    r.h.al = color;
    r.x.cx = x;
    r.x.dx = y;
    int86 (0x10, &r, &r);
}
```

## 2. 直接访问显示缓冲区

这种方法的好处是显示速度快，但不同显示适配器的缓冲区地址不尽相同，必须先了解采用的屏幕上的坐标与其显示 RAM 地址之间的映射关系，编程相对复杂一些，且要求所用的编译器能支持 `far` 指针。

一般来说，CGA 和 EGA 的显示 RAM 地址都从 B8000H 处开始，每个字节可以存放 4 个点的颜色信息。CGA 把偶数行的点放在 B800H: 0000H 至 B800H: 1FFFH 中，把奇数行的点放在 BA00H: 0000H 至 BA00H: 1FFFH 中。在每个字节内部，各点的信息从左到右存放。也就是说，第 0 个点占用 b7b6 两位，而第 3 个点占用 b1b0 两位。

计算对应地址时，先要把 X 坐标乘以 40，再与 Y 坐标除以 4 的值相加。根据 X 坐标除以 2 的余数，可以确定使用哪个显示区。如果余数为 0，就使用偶数线显示区，否则使用奇数线显示区。点在字节内的位置通过坐标模 4 求出，余数 0, 1, 2, 3 就是该点对应于字节中的位置。例如在 CGA 方式下，函数 `mempointc` ( ) (见附录一) 能将点以颜色 `color` 显示到第 X 行，第 Y 列的位置上。

## 五、例程

附录二中的源程序通过访问 2.13H 的 16 点阵字库 `h2k16`，用 ROM BIOS 中断方法写点，在第 100 行、第 40 列处，以黄色显示“微型计算机应用”。程序在 Turbo C2.0 下通过。如要用直接写显示缓冲区的方法写点，只要将程序中第 57 行中的 `writeln` ( ) 函数用 `mempoint` ( ) 函数取代即可。

如想显示 24 点阵的楷体汉字，可将程序作如下修改：

- ① 第 5 行的 32 改为 72
- ② 第 28 行的 `SEC1-1` 改为 `SEC1-16`
- ③ 第 30 行的 `h2k16` 改为 `h2k24k`
- ④ 第 41 行的 `x+=16` 改为 `x+=24`
- ⑤ 第 42 行的 `x+16` 改为 `x+24`
- ⑥ 第 43 行的 `y+=16` 改为 `y+=24`
- ⑦ 第 52 行的 `i<17` 改为 `i<25`
- ⑧ 第 53 行的 `j<3` 改为 `j<4`

⑨ 第 58 行的 `k3++` 改为 `k4++`

⑩ 第 60 行改为 `k3++; k4=x2;`

如要将汉字字型放大显示，只要把第 57 行的写点函数改为画圆并填充等操作即可，放大倍数（取决于圆半径）可任意，从而可以显示出丰富多彩的美术字型，这里不再详述。

```
/* write disp memory function */
mempoint(int x, int y, int color_code)
{
    union {
        unsigned char bits;
        unsigned i;
    } bit_mask;

    int index, loc;
    unsigned char t;
    char xor;
    char far * ptr = (char far *) 0xb8000000;
    bit_mask.i = 0xff3f;
    xor = color_code & 128;
    color_code = color_code & 127;
    loc = y % 4;
    color_code <= 2 * (3 - loc);
    bit_mask.i >= 2 * loc;
    index = x * 40 + (y > 2);
    if (x % 2) index += 8152;
    if (!xor) {
        t = *(ptr + index) & bit_mask.bits;
        *(ptr + index) = t | color_code;
    }
    else {
        t = *(ptr + index) & (char) 0;
        *(ptr + index) = t ^ color_code;
    }
}
```

## 附录一 直接写显示缓冲区

### 附录二 例程

```
/* display hzk16 program */
1: #include <stdio.h>
2: #include <stdlib.h>
3: #include <dos.h>
4: #include <graphics.h>
5: #define SIZE 32
6: #define X 40
7: #define Y 100
8: #define COLOR 14
9: char buf[SIZE+1], ch[] = "微型计算机应用";
10: char * pl;
11: main( ) {
12: int driver, x, y, mode = 0;
13: driver = DETECT;
14: initgraph(&driver, &mode, "");
15: x = X; y = Y; pl = ch;
16: disp(x, y);
17: getch( );
18: restorecrtmode( );
19: }
20: disp(int x, int y)
21: {
22: char sec1, sec2;
23: unsigned long index;
24: FILE * fp;
```

```

25: while( * pl) {
26: sec1 = * pl-160;
27: sec2 = * (pl+1)-160;
28: index = (sec1-1) * 94+sec2-1;
29: index * = SIZE;
30: if(!(fp = fopen("hzk16","rb"))) {
31: printf("Can't open this file!\n");
32: exit(1); }
33: if (fseek(fp,index,0)) {
34: printf("Seek error!\n");
35: exit(1); }
36: if((fread(buf,1,SIZE,fp))!=SIZE) {
37: printf("EOF readched!\n");
38: exit(1);}
39: fclose(fp);
40: display(x,y,buf);
41: pl+=2; x+=16;
42: if (x+16>getmaxx( )) {
43: y+=16;
44: x=1; }
45: }
46: }
47: display(int x1,int x2,char * buffer)
48: {
49: int i,j,k,k2,k3,k4;
50: char k1;
51: k3=x1;k4=x2;
52: for (i=1;i<17;i++) {
53: for (j=1;j<3;j++) {
54: k1 = * buffer++;
55: k=128;
56: for(k2=1;k2<9;k2++) {
57: if (k1&k)writepoint(k3,k4);
58: k3++; k /= 2;}
59: }
60: k4++; k3=x1;}
61: }
62: writepoint(int x,int y)
63: {
64: union REGS r;
65: r.h.ah=12;
66: r.h.al=COLOR;
67: r.x.cx=x;
68: r.x.dx=y;
69: int86(0x10,&r,&r);
70: }

```

#### 参考文献:

- ① 叶亚明译, 最新 C 语言大全, 中科院希望电脑公司.
- ② 杨志新, C 语言实现美术汉字, 电脑与微电子技术.

76

## CGA 显示 25 行 CFOXBASE 2.10 的改进

四川省涪陵地区卫生局 彭 禾

CCBIOS 2.13 系列 CC25.COM 支持 CGA 25 行显示高分 CFOXBASE, 但在调覆盖文件后即“黑”屏, 需连续按两次 Ctrl-F7, 即经纯西文方式, 再回汉字状态下现“.”提示符。究其原因, 与 OBH 功能调用有关。谨以多用户 FOXBASE V2.10 为例, 介绍回避该功能调用如下:

```

debug mfoxplus.ovl (141296 字节 89-07-04)
-s 100 1000 c6 06 ld 6a 0b
06C0
-u 6c0
06C0 C6061D6A0B MOV BYTE PTR
(6A1D), 0B
06C5 B81C6A MOV AX, 6A1C
:
- 6c4
06c4 0B.0f (改 0B 为 OF)

```

另外, 单用户版本 FOXPLUS.OVL (139904 字节 88-06-28) 用 PCTOOLS V4.21 操作如下:

|               |              |
|---------------|--------------|
| F10           | (取 PATH 功能)  |
| (d:) (CR)     | (选目标盘)       |
| (↓) .....CR   | (择子目录)       |
| (↓) .....CR   | (定文件)        |
| F             | (取 FIND 功能)  |
| F1            | (移十六进制码输入方式) |
| C60699690B CR | (键入寻找字符串)    |
| E             | (进入编辑功能)     |
| →→→→→→→→→→    | (右移光标至“B”字上) |
| F             | (改“B”为“F”)   |
| F5            | (执行修改)       |
| ESC.....ESC Y | (退出)         |

77

## 怎样用好CCED的计算功能

湖南省长沙市供销社 王 彬

CCED2.0是目前应用比较广泛的一种中文字表编辑软件,它将字处理、画线制表与数据加工融为一体,以功能强、操作简便而深受广大用户的青睐,是替代汉字WORDSTAR的一种理想软件。但在使用CCED的“CTRL+C”的计算功能对数列进行计算时,除应按照CCED的说明书进行操作外,还须注意以下几点:

1.参与计算的表格应是规范的二维开式或闭式表。若是开式表,因第一列无左边界线、最后一列无右边界线,因此这两列不能参与计算,计算时的列号应从第二列开始算起,即第n列应定义为Cn-1,如果开式表的第一列和最后一列需参与计算,可先将其制成闭式表,这样这两列就可参加计算,计算完毕,再利用抹线功能键抹去左右边界线即可。

2.作除法计算时,若作为除数的数列中存在数字0或空格,那么按“CTRL+C”后,CCED计算到数字0或空格时就会自动停止,0或空格以下的数列将得不到运算结果。当然在定义行块时,分段定义来避开数字0或空格再分段计算可解决上述问题,但此法过于繁琐,这里有一种更为简便有效的方法供大家参考:先确定一个合适的数X(大小自定,但不能影响到计算结果的精度),然后将X与算式中的除数相加。这样计算时,即使除数列中存在数字0或空格,整个数列仍将计算出正确结果,只不过数字0或空格对应的计算结果是一连串的“\*” (表示溢出),这也正是我们所希望的。例如,表格第八列为第一列与第三列之和除以第四列的值,计算结果取两位小数,令 $X=0.0001$ ,则算式为:  $c8 = \text{round}((c1 + c3) / (c4 + x), 2)$

3.作减法运算时,若差值在-1到+1之间,有时将得不出正确的结果。如 $5-5.01$ 结果应为-0.01,但在计算后,我们会发现CCED计算得出的不是-0.01,而是一连串的“\*”。对这一问题,可采用以下办法来解决:将减数和被减数都乘以 $10^5$ 使减数和被减数都成为整数再除以 $10^5$ ,这样计算得出的结果就是正确的了(X取正整数)。例如:第四列等于第三列与第二列之差,令 $X=2$ ,则算式为:  $C4 = (C3 \times 100 - C2 \times 100) / 100$ 。这时, $5-5.01$ 得出的就是-0.01了。

由于CCED2.0未提供对计算公式的保留功能(CCED3.0有),因此在打印表格前必须删去计算公式,否则计算公式会被打印出来。但是,计算公式能明确数据列之间的相互关系,而且在很多情况下,表格中的数据经更改后需用原公式进行再次计算,因此用户都希望能保留下计算公式。这里,笔者再向大家介绍在CCED2.0中保留计算公式的方法:先在表格的最后空行上输入所有的公式,然后进行计算,在打印表格前,避开最后的计算公式而将表格定义成行块,按“CTRL+T”进行行块打印。这样,计算公式就保留了下来而不会被打印出来。

78

## 如何在软盘上使用CCED3.0

石家庄市职业财会学校 翁锦玉 张春强

众所周知,CCED3.0只能在硬盘上安装、使用,这就给一些有特殊需要的用户带来了不必要的麻烦。有鉴于此,我们经过尝试,成功地将CCED3.0安装到了软盘上,现将过程详述如下。

首先必须准备一张CCED3.0原盘,一张空白软盘。

操作如下:

第一步,将空盘格式化成系统盘。(A: DOS系统盘, B: 空盘)

A>FORMAT B: /S

第二步,将VDISK.SYS和ASSIGN.COM拷至格式好的系统盘的根目录下。

A>COPY A: VDISK.SYS B:

A>COPY A: ASSIGN.COM B:

第三步,建立CONFIG.SYS系统配置文件。

A>COPY CON B: CONFIG.SYS

FILES=30

BUFFERS=30

DRIVER=VDISK.SYS 1 256 8

<F6>

第四步,用刚做好的系统盘重新启动系统(A: 刚做好的软盘)

第五步,在该系统盘上建立CCED子目录。

A>MD CCED

第六步,将CCED3.0原盘上所有文件拷贝至该盘CCED子目录。(A: 刚做好的系统盘, B: CCED3.0原盘)

A>COPY B: \CCED

第七步,将所有对C盘的操作指定为对B驱动器的操作。

A>ASSIGN C B

第八步,执行CCED3.0安装程序。

A>B: CCEDINST (安装过程不再赘述)

第九步,建立用于启动CCED的批处理文件。

A>COPY CON E.BAT

@ASSIGN C A

@CD \CCED

@CCED%1

@CD \

@ASSIGN

<F6>

此时,该盘已经做好,就可以在A驱动器中使用CCED3.0版进行文字编辑了。

79

# C语言下拉式菜单程序的常见失误及改进

## “后台”演奏音乐

南阳大学 王 水

中原机械工业学校 余 光

**【提要】**本文分析了用C语言构造汉字下拉式菜单系统过程中的常见失误，并给出了完善的方法，最后构成一个不依赖于汉字操作系统、可在屏幕任何位置弹出的通用的汉字下拉式菜单系统。

**关键词：**下拉式菜单，动态内存分配，屏幕的保存与恢复，汉字显示。

用C语言构造汉字下拉式菜单系统的常见方法是：根据Herb Schildt所著《C语言高级程序员编程指南》（鸿健编译，北京希望电脑公司出版；另外，尹彦芝等编译，科海培训中心出版的《C语言常用算法与子程序》一书，内容也多取自此书）中第一章有关PULLDOWN菜单的构造方法加以修改，增加其汉字处理功能而实现的。本方法首先在程序起始部分，用一系列对makemenu()函数的调用完成对各个菜单框架的构造，包括为每个菜单在弹出前保存屏幕分配内存空间；然后在主程序中使用pulldown()函数的返回值确定用户的选择。pulldown()函数的基本形式为（为方便起见，本书中所用函数名与原书稍有不同）：

```
int pulldown(int num)
{
    /* num 为菜单号 */
    if(!frame[num].active) /* 若激活标志为零 */
    {
        savevideo(num); /* 保存屏幕 */
        frame[num].active = 1; /* 设置激活标志 */
    }
    displaymenu(num); /* 显示菜单 */
    return getresponse(num); /* 返回用户的响应 */
}
```

恢复屏幕的工作由函数restorevideo()在主函数中完成；整个程序工作于文本方式下，保存和恢复屏幕使用直接读写显示RAM的方法；书中还给出了一个使用下拉式菜单的完整例子。作为一个示范性例子，该程序有以下几点不足：

1. 保存屏幕所用的内存，在makemenu()中已经为每个菜单分配，并在整个程序运行期间均占用此空间，这对于工作于图形方式下的汉字系统来说，是一个很大的开销。

2. 不论是否已经激活（已经显示在屏幕上），进入菜单的时候都要调用显示函数displaymenu()，不仅浪费机时，而且使重入菜单时的显示产生晃动。

虽然，由于原程序工作在字符方式下，以上两点还

没有构成严重的错误；但是，当修改它使成为汉字菜单程序的时候，由于保存图形屏幕所需内存很大，当程序较大，菜单较多的时候，上述第一点就可能构成严重的错误。为了解决这个问题，显然应当采用动态分配内存的方法：在显示菜单(displaymenu())之前，分配内存并保存屏幕；在恢复屏幕(restorevideo())之后，立即释放这部分内存空间。于是有些同志就将对内存的分配和对函数savevideo()的调用放在函数displaymenu()的首部。这时第二点就成为危害很大而且很隐蔽的一个错误：由于下拉式菜单是可以重新进入的，所以用户可能多次进入同一级菜单，每次进入都要重新显示菜单图形，而每次显示又都要进行内存的分配和屏幕的保存，其中屏幕的保存是多余的操作，而对一个没有释放的指针用malloc()函数进行重新分配则可能是灾难性的：在几次似乎“正常”的菜单操作之后，提示“堆栈溢出”并异常结束甚至整个系统挂死。

正确的实现这个过程的一般方法是：将对内存的分配放在函数savevideo()的起始部分，而在restorevideo()的末尾释放已使用过的内存空间，并修改pulldown()函数消除重复的菜单显示。

下面给出用于汉字菜单的结构定义、宏定义及外部变量，注意在结构MENUFRAME中，菜单文本实际形式为：char \* menu[]，写成下面的等价形式是为了消除编译错误；另外，新出现的menucursor指针用于保存一个指示菜单选择的色带，此色带与菜单项进行“异或”操作，产生色带消失和移动的效果；最后的几个参数用于确定菜单的各项色彩：

```
#define ESC 27
#define UP 72
#define DOWN 80
#define MAX_MENUFRAME 10 /* 总菜单数，可根据需要改变 */
typedef struct
```



```

{int startx,starty,endx,endy; /* 菜单的屏幕范围 */
void far * p; /* 保存屏幕的内存指针 */
void far * menucursor; /* 保存菜单杠的指针 */
char * * menu; /* 菜单项文本 */
char * hotkeys; /* 菜单选择热键 */
int count; /* 菜单项数 */
size_t size; /* 保存菜单背景所需内存大小 */
int active; /* 激活标志 */
int currchoice; /* 当前选择 */
int menucolor,backgroundcolor,menubordercolor;
int cursorcolor,cursorbordercolor; /* 色彩 */
}MENUFRAME;

```

```
int CCLIB;
```

```
MENUFRAME frame[MAX_MENUFRAME];
```

下面是需要修改的几个函数，注意它们是如何进行动态的内存分配的：

```

void savevideo(int num) /* 保存菜单背景 */
{
    frame[num].p = malloc(frame[num].size);
    7getimage(&frame[num].startx,frame[num].starty,frame[num].endx,frame[
        num].endy, frame[num].p);
}

void restorevideo(int num) /* 恢复菜单背景 */
{
    putimage(frame[num].startx, frame[num].starty, frame[num].p,
        COPY_PUT);
    frame[num].active = 0;
    free(frame[num].p);
    free(frame[num].menucursor);
}

```

相应的主模块 pulldown(), 注意其运行逻辑与原模块的不同：

```

int pulldown(int num)
{
    int i;
    if(!frame[num].active) /* 若没有激活 */
    {
        savevideo(num); /* 保存屏幕 */
        displaymenu(num); /* 显示菜单 */
    }
    i = getresponse(num); /* 取得用户响应 */
    frame[num].active = 1; /* 设置激活标志 */
    return i;
}

```

构造菜单框架的函数也需作相应修改，以适应图形方式和动态分配：

```

int makemenu(int num,char * menu[],char * hotkeys,int count,int x,int y,
    int menucolor,int backgroundcolor,int menubordercolor,
    int cursorcolor,int cursorbordercolor)
{
    register int i,len;
    int endx,endy;
    size_t size;

    /* 计算菜单大小，假设程序工作在 640 * 480 VGA 下 */
    len = 0;
    for(i=0;i<count;i++)

```

```

        if(textwidth(menu[i])>len) len = textwidth(menu[i]);
        endx = x+len+10;
        endy = y+count * 18+1;
        if ((endx+1>640) (endy+1>480))
        {printf("menu won't fit");
            exit(1);}

```

```

/* 计算保存背景的内存大小 */
size = (size_t)imagesize(x,y,endx,endy);
if (!size){printf("\n menu too big. out of memory.");
    exit(1);}

```

```

/* 形成框架 */
frame[num].startx = x;
frame[num].endx = endx;
frame[num].starty = y;
frame[num].endy = endy;
/* frame[num].p = 动态分配 */
frame[num].menu = (char * *)menu;
frame[num].hotkeys = hotkeys;
frame[num].count = count;
frame[num].size = size;
frame[num].active = 0;
frame[num].currchoice = 0;
frame[num].menucolor = menucolor;
frame[num].backgroundcolor = backgroundcolor;
frame[num].menubordercolor = menubordercolor;
frame[num].cursorcolor = cursorcolor;
frame[num].cursorbordercolor = cursorbordercolor;
return 1;
}

```

下面是显示菜单及取得用户回响的函数，注意它们是如何形成菜单条及利用“异或”方式获得菜单条移动效果的；另外，为具有通用性，汉字显示使用 puthz() 函数，此函数可以在任意指定的坐标（以象素点为单位）显示汉字（或西文）字串，不受一般行、列的限制，也不需要汉字操作系统的存在（后者在工业控制等只需要少量汉字显示的程序非常有用，可以节省汉字操作系统可能占据的大量内存）；为了使用这个函数，必须在程序外预先以低级 IO 方式打开一个 16×16 点阵的汉字库并将返回的句柄赋给外部变量 CCLIB，例如：

```
CCLIB = open("C:\\UCDOS\\CCLIB.DAT","O_ RDONLY O_BINARY");
```

puthz()函数直接从字库中读取字模信息并将汉字“画”在指定坐标处。

```

void displaymenu(int num)
{
    int i;
    /* 产生菜单条 */
    setcolor(frame[num].cursorbordercolor);
    setfillstyle(SOLID_FILL,frame[num].cursorcolor);
    bar3d(frame[num].startx+1, frame[num].starty+1,
        frame[num].endx-1, frame[num].starty+18,0,0);
    frame[num].menucursor = (void far *)malloc((size_t)
        imagesize(frame[num].startx+1,
            frame[num].starty+1, frame[num].endx-1, frame[num].starty+18));
}

```

```

getimage(frame[num].startx+1, frame[num].starty+1,
frame[num].endx-1, frame[num].starty+18, frame[num].menucursor);
/* 写出菜单 */
setcolor(frame[num].menubordercolor);
setfillstyle(SOLID_FILL, frame[num].backgroundcolor);
bar3d(frame[num].startx, frame[num].starty, frame[num].endx,
frame[num].endy, 0, 0);
for(i=0; i<frame[num].count; i++)
    puthz(frame[num].startx+5, frame[num].starty+2+i*18,
        (unsigned char *)frame[num].menu[i],
        CCLIB, frame[num].menucolor);
}
/* 获取用户的回响 */
int getresponse(int num)
{
    union inkey {char ch[2]; int i;};
    int keychoice;
    /* 移菜单条到当前选择项上 */
    if(!frame[num].active)
        putimage(frame[num].startx+1, frame[num].starty
            +frame[num].currchoice*18+1, frame[num].menucursor, XOR_PUT);
    while(1){
        while(!bioskey(1)); /* 等待击键 */
        c.i = bioskey(0); /* 读键盘 */
        keychoice = 0;
        if(c.ch[0]) /* 常规键 */
        {
            /* 检验回车、ESC 和 SPACE BAR */
            switch(c.ch[0])
            {
                case '\r': return frame[num].currchoice;
                case ESC: return -1; /* 撤消 */
                default: /* 移去菜单条 */
                    putimage(frame[num].startx+1, frame[num].starty
                        +frame[num].currchoice*18+1, frame[num].menucursor, XOR_PUT);
                    switch(c.ch[0])
                    {
                        case ' ': frame[num].currchoice++; break;
                        /* check for SPACE */
                        default: /* 检验热键 */
                            keychoice = is_in(frame[num].hotkeys, tolower(c.ch[0]));
                            if(keychoice) frame[num].currchoice = keychoice-1;
                    }
            }
        }
        else
        {
            /* 光标键 */
            putimage(frame[num].startx+1, frame[num].starty
                +frame[num].currchoice*18+1, frame[num].menucursor, XOR_PUT);
            switch(c.ch[1])
            {
                case UP: frame[num].currchoice--; break;
                case DOWN: frame[num].currchoice++; break;
            }
        }
    }
    if(frame[num].currchoice == frame[num].count) frame[num].currchoice

```

```

=0;
if(frame[num].currchoice<0) frame[num].currchoice = frame[num].count-1;
/* 加亮下一选择项 */
putimage(frame[num].startx+1, frame[num].starty
    +frame[num].currchoice*18+1, frame[num].menucursor, XOR_PUT);
if(keychoice) return keychoice-1;
}
return -1;
}

```

最后附上通用的 16 点阵汉字显示模块:

```

void puthz(int x, int y, unsigned char *hz, int handle, int color)
{
    long offset; /* 汉字在字库中的偏移量 */
    unsigned int i, curr_color, f=0, c1, c2;
    int i1, i2, i3;
    char by[32]; /* 存放点阵信息的数组 */
    #define getbit(c,n) ((c)>>(n)&1)
    curr_color = getcolor(0); /* 保存当前前景颜色 */
    setcolor(color);
    settxtstyle(DEFAULT_FONT, HORIZ_DIR, 1);
    while((i = *hz++) != 0) /* 循环到汉字字符串结尾 */
    {
        if(i>0xa0) /* 汉字 */
        {
            if(f==0) /* 汉字第一字节 */
            {
                c1 = (i-0xa0)*94;
                f++;
            }
            else /* 汉字第二字节 */
            {
                c2 = i-0xa0+c1-95;
                /* 汉字记录号计算, 常数 95 随字库而不同, 此是 */
                f=0; /* UCDS 数值; 对 213H, 为 1905 */
                offset = c2*32L;
                lseek(handle, offset, SEEK_SET);
                read(handle, by, 32);
                for(i1=0; i1<16; i1++)
                    for(i2=0; i2<2; i2++)
                        for(i3=0; i3<8; i3++)
                            if(getbit(by[(i1*2)+i2], 7-i3))
                                putpixel((x+i2*8+i3), (y+i1), color);
                x+=16;
            }
        }
        else /* 西文字符处理 */
        {
            outtextxy(x, y+6, (char far *)i); x+=8;
        }
        setcolor(curr_color); /* 恢复原前景颜色 */
    }
}

```

限于篇幅, puthz()的原理这里就不讲了; 另外, 菜单显示部分也可以使用汉字系统本身提供的显示功能和 ansi 命令系列 (通用性不好), 或 BIOS 视频中断 (INT 0x10 功能 9 及 2) (通用性较好, 但只能按行、列定位), 请大家参考有关文章。 (80)

# “后台”演奏音乐

西安交通大学计算机系 吴建林

当计算机 CPU 在“前台”忙个不停时，“后台”能够同时演奏出美妙的音乐。我们知道，DOS 操作系统是没有前后台之分的，那么如何能够实现“后台”音乐演奏呢？

其实，音乐的演奏是通过对 8253 定时器芯片的通道 2 进行编程，使其发出特定频率的信号，此后，只要重新对 8253 芯片编程，就可使其停止发声。而 8253 定时器芯片独立于 8088CPU 运行，所以在其它操作同时进行演奏音乐是很平常的事情。

BIOS 包含一特殊的“伪”中断 (INT 1CH)，这一中断若无相应的例程将不起任何作用，启动此中断后立即返回，但其独特之处是：在 BIOS 定时器中断修改日历计数之后调用该中断的，也就是说，这是每秒 18.2 次自动发生的硬件中断。通过修改 INT 1CH 的中断向量，使之指向音乐演奏中断例程，那么，此程序就会每秒被调用 18.2 次，也就产生了“后台”演奏音乐的效果。

这里给出的“后台”演奏音乐程序附有详细说明，生成过程为：

```
C>masm bkmusic;
C>link bkmusic;
C>exe2bin bkmusic.exe bkmusic.com
C>bkmusic
```

程序在 386 微机上调试通过。

```
;
;本程序完成后台音乐演奏功能
;输入:
;      freq-degree      取值: 0 1 2 .....
;      freq (frequency) 取值: 1 2 3 4 5 6 7 11 12
;                          13 14 15 17 21 22
;                          23 24 25 26 27
;
;      time-base        取值: 正整数
;      time              取值: 1.....32
;      fraction          取值: 1 2 4
;      repeat-times      取值: 1 2 .....
;输出:
;      无
;
title      bkmusic.asm
code       segment para public 'code'
assume cs:code,ds:code,ss:code,es:code
org 100h
start:     jmp      begin
stk        dw       100      dup(?)
stk-top    label     word
```

```
;频率表定义
freqtab    dw       131,147,165,175,196,220,247
           dw       262,294,330,349,392,440,494
           dw       523,587,659,699,784,880,988

;频率代码定义
numtab     db       1,2,3,4,5,6,7
           db       11,12,13,14,15,16,17
           db       21,22,23,24,25,26,27

;频率提高倍数定义
freq-degree db       0
;<<友谊地久天长>>音乐频率代码定义
freq       db       5,11,11,11,13
           db       12,11,12,13
           db       11,11,13,15
           db       16,16
           db       15,13,13,11
           db       12,11,12,13
           db       11,6,6,5
           db       11,16
           db       15,13,13,11
           db       12,11,12,16
           db       15,13,13,15
           db       16,21
           db       15,13,13,11
           db       12,11,12,13
           db       11,6,6,5
           db       11,0

;延时基数定义
time-base  db       4
;<<友谊地久天长>>音符延迟时间定义
freq       db       2,3,1,2,2,3,1,2,2,3,1,2,2,6,2
           db       3,1,2,2,3,1,2,2,3,1,2,2
           db       6,2,3,1,3,1,3,1,2,2,3,1,3,1,6
           db       2,3,1,2,2,3,1,2,2,3,1,2
           db       2,8

;节拍划分定义
;1-32划分,2-16划分,4-8划分
fraction   db       4
which-note db       0 ;频率表下标
sound-now  db       1 ;开始演奏控制变量
first-note db       1 ;音符演奏控制变量
end-note   dw       0 ;结束时脉冲冲数
repeat-times db      1 ;演奏次数
;主程序
begin:
           push     cs
           pop      ax
           mov      dx,ax
           mov      es,ax
           mov      ss,ax
           mov      ax,offset stk-top
           mov      sp,ax
;修改INT 1CH中断向量
           mov      dx,offset melody
           mov      ax,cs
           push     ds
           mov      ds,ax
           mov      ax,251ch
           int      21h
```

```

    pop        ds
    mov        dx, offset pgmend
    mov        cl, 4
    shr        dx, cl
    add        dx, 11h
    mov        ax, 3100h
    int        21h          ;中断例程常驻内存

;中断例程
melody:
    proc        far
    push        ax
    push        bx
    push        cx
    push        dx
    push        di
    push        si
    push        ds
    mov        ax, cs
    mov        ds, ax      ;恢复 DS 值
    cmp        sound-now, 1 ;现在是否发声
    je         play
    jmp        no-now
play:
    cmp        first-note, 0 ;音符是否第一次演奏
    jne        cnt0
    jmp        tm-chk
    mov        first-note, 0
    mov        bx, offset freq
    add        bl, which-note
    adc        bh, 0
    mov        al, [bx]
    cmp        al, 0        ;AL 存放频率代码
    jne        cnt
    jmp        no-mr
cnt:
    mov        si, 0
agn:
    mov        bx, offset numtab ;搜索频率代码
    add        bx, si
    cmp        al, [bx]
    je         nxt
    inc        si
    jmp        agn
nxt:
    add        si, si
    mov        bx, offset freqtab
    add        bx, si
    mov        dx, [bx]      ;DX 存放实际频率
    cmp        freq-degree, 0
    je         no-high
    mov        cl, freq-degree
    shl        dx, cl
no-high:
    mov        si, dx
    mov        dx, 12h
    mov        ax, 34dch
    div        si            ;AX 存放时间常数
    push        ax
    mov        bx, offset time
    add        bl, which-note
    adc        bh, 0
    xor        ax, ax
    mov        al, [bx]
    mov        cl, fraction
    mul        cl

    mov        bx, ax
    add        bl, time-base
    adc        bh, 0
    mov        ah, 0
    int        1ah          ;DX 存放当前时钟脉冲数
    add        bx, dx        ;BX 存放延迟时钟脉冲数
    mov        end-note, bx
    mov        al, 0b6h
    out        43h, al
    ;初始化8253命令寄存器43H
    pop        ax
    out        42h, al
    xchg        al, ah
    out        42h, al
    in        al, 61h        ;打开扬声器
    or         al, 3
    out        61h, al
tm-chk:
    mov        ah, 0
    int        1ah
    cmp        dx, end-note
    jne        no-now
    mov        cl, which-note
    inc        cl
    mov        which-note, cl
    in        al, 61h        ;关闭扬声器
    and        al, 0fch
    out        61h, al
    mov        first-note, 1
    jmp        play
no-mr:
    mov        cl, repeat-times
    dec        cl
    mov        repeat-times, cl
    jz         no-agn
    mov        which-note, 0
    mov        first-note, 1
    jmp        no-now
no-agn:
    mov        sound-now, 0
    ;恢复INT 1CH中断向量
    mov        dx, 0ff63h
    mov        ax, 0f000h
    push        ds
    mov        ds, ax
    mov        ax, 251ch
    int        21h
    pop        ds
no-now:
    pop        ds
    pop        si
    pop        di
    pop        dx
    pop        cx
    pop        bx
    pop        ax
    iret

melody endp
pgmend db        ?
code ends
end start

```

# CGA、EGA/VGA 卡间图形数据转化

重庆大学计算机系 符云清 李国强

## 一、问题提出

我们开发软件时,有时要利用图形库中现成的图形,这些图形都是以数据文件形式存在磁盘上,且是针对某一具体显示卡的,如 CGA 卡,并且其显示的色彩也是固定的。而我們希望能改变其色彩,或能在其他卡上也能显示,如 EGA/VGA 卡等。由于不同卡的硬件结构差异,因而图形数据存储格式也迥然不同。要解决上述问题,就涉及到图形数据的转化。通过转化,能避免重复劳动,使图形数据达到共享的目的。

## 二、图形数据格式

要实现图形数据的转化,首先必须了解不同卡上的图形数据存储格式。本文以 TUBRO C2.0 中 getimage() 的格式为例加以讨论。在 CGA、EGA/VGA 卡上,图形数据的头两个字节表示图形矩形区域 X 方向像素个数,设有 m 个像素,则该两个字节值为 m-1 (从 0 开始计数),紧接两个字节的 Y 方向的像素个数,设为 n,再后便是图形数据。图形数据中 CGA 以 2 个 bit 表示屏上一个像素,故一个字节表示连续 4 个像素。X 方向共用  $(m \times 2 + 6) + 8$  个字节,整个图形用  $(m+3) \times n + 4$  个字节。

而 EGA、VGA 卡图形数据格式则截然不同,其格式为用 4 个 bit 表示屏上一个像素,而且这 4 个 bit 不在同一个字节中,而是位于 4 个字节中,见附录部分图示。

## 三、图形数据转化

理解其存储格式后,便可以很容易实现在不同卡间及在同一卡上不同颜色显示的转化。本文以 CGA 卡到 EGA/VGA 卡转化为例并给写 TUBRO C2.0 编写的通用转化程序。程序的思想是:每次从 CGA 图形数据文件中取  $(m \times 2 + 6) + 8$  字节,以 2 个字节为一小组,每次依次取 2 个 bit,识别出是前景色还是背景色,以转化成 EGA/VGA 上的相应颜色值,然后利用一个 4 重循环,依次将颜色值的对应位写入相应字节中去。为提高速度,我们也可以使用 4 个缓冲区,循环一次即可。这是以空间来换取时间,还是值得。上述介绍的是 CGA 到 EGA/VGA 卡的转化,根据上述分析,读者亦不难编写 EGA/VGA 到 CGA 卡的转化程序,笔者曾用上述方法实现不同卡上数据转化,及同一种卡上的色彩变换,效果相当好,从而实现图形库的共

享及卡间的可移植性。

附程序清单及图形数据存储模式:

(本程序文件内的程序 1 是将 CGA 的图形改变相应的前景和背景,输出仍为 CGA 图形数据)

```
A>type vga.c
#include <stdio.h>
#include <io.h>
#include <dos.h>
#include <conio.h>
char buf[256];
char new_buf[128];
char mask[]={
    0xc0, 0x30, 0x0c, 0x03,
};
char get-bit[]={
    0x08, 0x04, 0x02, 0x01,
};
main()
{
    char source[11], object[11];
    int old-bk, new-bk, old-color, new-color, color;
    short x-len, y-len, x-byte, sum, ch;
    FILE *fp1, *fp2;
    int i, j, k;
    clrscr();
    gotoxy(1, 5);
    printf("CGA graph data convert to EGA/VGA V1.0\n");
    printf("Copyright(C) by Fu YunQing, ChongQing");
    printf("\n");
    printf("input source graph data file name:");
    gets(source);
    printf("input target graph data file name:");
    gets(object);
    printf("origin background color code:");
    scanf("%d", &color); old-bk=(char)color;
    printf("new background color code:");
    scanf("%d", &color); new-bk=(char)color;
    printf("new front color code is:");
    scanf("%d", &color); new-color=(char)color;
    fp1=fopen(source, "rb");
    fp2=fopen(object, "wb");
    if(fp1==NULL||fp2==NULL){
        printf("File Open Error! \n");
        exit(0);
    }
    x-len=getw(fp1), putw(x-len, fp2);
    y-len=getw(fp1), putw(y-len, fp2);
    x-len++, y-len++;
    x-byte=(x-len*2+7)/8;
    while(y-len--){
        fread(buf, sizeof(char), x-byte, fp1);
        for(i=0; i<4; i++){
            for(j=0; j<x-byte; j+=2){
                sum=0;
                for(k=0; k<4; k++){
                    ch=(buf[j]&mask[k])>>(6-2*k);
                    if(ch==0) ch=0;
                    else ch=new-color;
                    ch=(ch&get-bit[i])>>(3-i);
                    sum=sum|ch<<(7-k);
                }
                for(k=0; k<4; k++){
                    ch=(buf[j+1]&mask[k])>>(6-2*k);
                    if(ch==0) ch=0;
                    else ch=new-color;
                    ch=(ch&get-bit[i])>>(3-i);
                    sum=sum|ch<<(3-k);
                }
                new_buf[j/2]=sum;
            }
            /*end of loop j*/
            fwrite(new_buf, sizeof(char), (x-len+7)/8, fp2);
        }
        /*end of loop i*/
    }
    fclose(fp1), fclose(fp2);
}
```

# VGA 图形信息的压缩存储和恢复

江西省地质矿产局程序研究室 郭新平

【内容摘要】VGA 在图形显示方式下，屏幕图形信息的存储需占用大量外存。本文试图通过采用无冗余取样压缩算法，使得在存储屏幕图形信息时，能将信息数据进行压缩存储，在恢复图形时再还原，以节省外存空间。

【关键词】VGA 图形 信息存取 数据压缩与还原

VGA 的图形显示，一般是采用  $640 \times 480$  的显示方式（如 UCDOS、CCDOS2.13H 等），其屏幕图形信息的存取，已多次见诸报刊，采用的方法都是直接对四个位平面进行数据的读出和写入。每个位平面的数据为 38400 个字节，四个位平面共占用 153600 个字节，约为 151K 字节，若想保存一些有用或有趣的图形，需占用大量的外存空间，这就很不经济。为了节省外存空间，本人采用无冗余取样压缩算法对屏幕图形信息进行压缩存储，收到了明显的效果（如存储刚进入 FOXBASE 时显示的图形只需 13830 个字节，不到未压缩时的十分之一）。

数据压缩规则如下：

(1) 在每个位平面的原始数据中，对于连续出现的  $N$  ( $N \geq 2$ ) 个相同的数  $X$ ，只在压缩数据中相继存放  $N$  和  $X$  两个数；

(2) 在每个位平面的原始数据中，对于相邻而又不相等的连续  $M$  个数据，则在压缩数据中先存放一  $M$ ，再连续存放这  $M$  个数；

(3) 对于连续的相同的  $N$  个数或连续的不相同的  $M$  个数多于 127 个时（即  $N > 127$  或  $M > 127$ ），由于用一个字节表示成有符号数时，其范围为  $-128 < N < 127$ ，因此需分多次进行存放，即每次至多存放 127 个；

(4) 压缩数据中数的存放次序与原始数据中数的出现次序一致；

(5) 在存储每个平面的压缩数据之前，先存放该位平面压缩数据的长度。

如原始数据为：

1 2 3 10 10 10 10 10 10 7 8 8 8

则压缩后数据为：

-3 1 2 3 6 10 -1 7 3 8

使用上述算法存储屏幕图形信息，对于一些简单的图形的压缩存取具有明显的效果。一是占用外存少，二是还原显示速度快。即使对于特别复杂的图形（相邻的数据两两不相等）的存储，其四个位平面的压缩数据因

为算法而增加了字节数也不多（至多 1220 个字节）。当然，对于十分复杂的图形，我们可以假定，当每个位平面数据压缩后的长度超过 30000 个字节时，为节省恢复数据的时间，可直接对位平面的原始数据进行存储，不存入压缩数据，且在存放原始数据之前保存长度值 38400；在恢复数据时，如果读入的长度值为 38400，则直接将读出数据写入位平面。有兴趣者可加一段程序实现。

下面的两个程序（程序清单附后），SAVE.ASM 是对屏幕图形进行压缩存储的程序，LOAD.ASM 是还原显示程序，两个程序经汇编连接后生成 COM 文件，可在 DBASE III (FoxBASE) 或操作系统下运行，运行时需带参数，如 SAVE N 或 LOAD N，N 的取值为 0~9，表示两程序可对 PICTURE.000 至 PICTURE.009 十个图形文件进行存取。

这两个程序都在 T&W386 机上 CCDOS2.13H 下运行能过。

；VGA 或者 TVGA 显示方式为  $640 \times 480$  的全屏幕图形压缩保存程序

```
code segment
org 100h
assume cs:code,ds:code
savescrn
proc near
push ds
mov ax,cs
mov ds,ax
mov al,byte ptr ds:[0082h]; 取文件名参数
mov filename[10],al
lea dx,filename ; 打开文件
mov cx,0
mov ah,3ch
int 21h
jc quit1
jmp cont
quit1:
jmp quit
cont:
mov handle,ax
mov dx,3ceh ; 打开四个位平面供读
mov al,4
out dx,al
mov ax,0a000h
mov es,ax
```

```

mov count,3
again: mov al,count ; 选择位平面 (3,2,1,0)
      mov dx,3cfh
      out dx,al
      mov ax,0 ; 以下至标号 store 为该位平面数据并压缩
      mov di,ax
      mov si,offset buff
      mov bx,si
      mov cx,bx
      inc si
      mov nota,al ; nota 为标志变量, 处理相邻两字节
                    ;相同数据时为 0, 不相同数据时为 1
      mov nota,al
      mov al,es:[di]
      mov [si],al
      cmp al,es:[di+1]
      jz one
      inc nota
      inc si
one:   mov dl,1
      inc di
aga:   mov al,es:[di]
      cmp al,es:[di+1]
      jz yequ
      cmp nota,0
      jz equ1
      cmp dl,0 ; 相邻两数不等, 且前面处理不等的情况
      jnz nze
      mov bx,si
      inc si
nzc:   inc dl
      mov [si],al
      inc si
      cmp dl,127
      jb c127
      not dl
      inc dl
      mov [bx],dl
      mov dl,0
      mov bx,si
      inc si
c127:  inc di
      cmp di,38401
      jz store
      jmp aga
equ1:  inc nota ; 相邻两数不等, 但前面处理相等的情况
      inc dl
      mov [bx],dl
      inc si
      mov dl,0
      inc di
      cmp di,38401
      jz store
      jmp aga
yequ:  cmp nota,0
      jz equ2
      dec nota ; 相邻两数相等, 但前面处理不等的情况
      cmp dl,0
      jz nz
      not dl
      inc dl
      mov [bx],dl
      mov dl,1
      inc si
      mov dl,1
      inc si
      mov [si],al
      inc di
      cmp di,38401
      jz store
      jmp aga
equ2:  inc dl ; 相邻两数相等, 且前面处理相等的情况
      mov [si],al
      inc di
      cmp di,38401
      jz store
      cmp dl,127
      jb aga
      mov [bx],dl
      mov dl,0
      inc si
      mov bx,si
      inc si
      jmp aga
store: mov ax,si ; 数据存储
      sub ax,cx
      inc ax
      mov word ptr len,ax
      mov dx,offset len ; 存压缩后一个位平面的字节数
      mov cx,2
      mov bx,handle
      mov ah,40h
      int 21h
      jc close
      cmp ax,2
      jne close
      mov dx,offset buff ; 存一个位平面的压缩数据
      mov cx,word ptr len
      mov bx,handle
      mov ah,40h
      int 21h
      jc close
      cmp ax,word ptr len
      jne close
      dec count ; 转入下一位平面
      jge again1
      jmp close
again1: jmp again
close:  mov bx,handle ; 关闭文件
      mov ah,3ch
      int 21h
      quit: mov dx,3ceh ; 关闭四个位平面
            mov al,3
            out dx,al

```

```

        inc dx
        mov al,0
        out dx,al
        pop ds
        ret
savescrn endp
filename db 'picture.000',0
handle   dw 0
count    db 0
nota     db 0
len       dw 0
buff     db 39000 dup(0)
code     ends
        end savescrn

```

; VGA 或者 TVGA 显示器显示方式 640×480 的全屏幕图形恢复程序

```

code segment
    org 100h
    assume cs:code,ds:code
loadscrn proc near
    push ds
    mov ax,cs
    mov ds,ax
    mov al,byte ptr ds:[0082h] ; 取文件名参数
    mov filename[10],al
    lea dx,filename           ; 打开文件
    mov cx,0
    mov ah,3dh
    mov al,0
    int 21h
    jc quit1
    jmp cont
quit1:   jmp quit
cont:    mov handle,ax
        mov ax,0a000h
        mov es,ax
        mov dx,3c4h           ; 打开四个位平面供写
        mov al,2
        out dx,al
        mov count,8
again:   mov al,count         ; 选择位平面 (3,2,1,0)
        mov dx,3c5h
        out dx,al
        mov dx,offset len ; 读一个位平面数据压缩后的字节数
        mov cx,2
        mov bx,handle
        mov ah,3fh
        int 21h
        jc close
        cmp ax,2
        jne close
        mov dx,offset buff ; 读一个位平面压缩后的数据
        mov cx,word ptr len
        mov bx,handle
        mov ah,3fh

```

```

        int 21h
        jc close
        cmp ax,word ptr len
        jne close
        mov ax,0           ; 数据还原
        mov di,ax
        mov si,offset buff
aga:     mov bl,[si]
        inc si
        mov al,[si]
        cmp bl,0
        jg yequ
        not bl              ; 相邻两数据不相等的情况
        inc bl
nequ:    mov es:[di],al
        inc di
        cmp di,38401
        jz next
        inc si
        mov al,[si]
        dec bl
        jnz nequ
        jmp aga
yequ:    mov es:[di],al     ; 相邻两数据相等的情况
        inc di
        cmp di,38401
        jz next
        dec bl
        jnz yequ
        inc si
        jmp aga
next:    shr count,1        ; 选择下一位平面
        jnz again
close:   mov bx,handle      ; 关闭文件
        mov ah,3eh
        int 21h
quit:    mov dx,3c4h        ; 关闭四个位平面
        mov al,2
        out dx,al
        inc dx
        mov al,0fh
        out dx,al
        pop ds
        ret
loadscrn endp
filename db 'picture.000',0
handle   dw 0
count    db 0
nota     db 0
len       dw 0
buff     db 39000 dup(0)
code     ends
        end loadscrn

```



# 热烈祝贺——

## 电子工业出版社广州科技公司在穗开业

### 电子工业出版社广州科技公司业务简介

电子工业出版社是我国以出版电子科技图书为主的一家直属电子工业部的中央一级出版社。成立十余年来,致力于为我国电子行业的生产、科研、教学、管理人员和广大电子爱好者服务;致力于发展国内外文化、技术的交流与合作,繁荣我国科技出版事业;振兴我国电子科学技术,促进科学技术和生产相结合,促进高新技术研究成果的商品化,加快具有中国特色的社会主义的建设是本社的唯一宗旨。

随着改革开放的进一步深入,为了充分利用珠江三角洲地区信息快、商品新、科技发展迅速的优势,特在广州开设“电子工业出版社广州科技公司”,以繁荣高新技术市场、促进微电子和电子信息等高新技术、产品的交流和推广为宗旨,充分组织以广州为中心的南方地区的科技优势,及时总结科技专家的研究成果,通过资料、期刊、图书的出版以及科技咨询,科技转让,科技服务等,为加快“四小虎”的发展尽微薄之力。

公司下设:编辑出版部;图书发行部;综合经营部;技术开发部。

《今日电子》《电子与电脑》读者服务部;“中国软件行业协会软件出版分会”软件销售中心(广州部)等,已准备设在本公司,筹备工作正在进行。

希望广东乃至全国关心电子科技出版事业发展的作者、读者们给予关照和支持。

我们真诚的希望有更多的作者同我们联系出版事宜,欢迎赐稿。欢迎各书店和广大读者来我公司选购及批发电子版图书、期刊、软件和音像出版物。

公司地址:广州市天河五山路华师大科技服务楼215号

邮 编:510631

联系人:周青峰 陈昊

# 《面向十六位字节流的加密工具》

## 改错与改进

江西省水利水电学校微机室 龙相明

《电脑》杂志 1992 年第五期刊登刊登的《一种面向十六位字节流的程序加密解密方法》一文中的软件工具程序，适用于加密中西文信息的混码文件，具有一定的实用价值。然而，经笔者使用，发现原程序中存在一些明显的错误和不足，现分别介绍如下：

1、用原程序进行一次加密或解密操作，被加密的文件尺寸要扩大一倍。如：某文件长为 100byte 经一次加密解密操作后，其长度被扩大为 400byte。分析原程序后，发现产生这种错误的原因是，原程序中对被加密程序的记录数取值有错误。正确的记录数取值应按下式计算：

记录数 =  $\text{lof}(1) / \text{记录长度}$

由于原程序是面向十六位字节流的，其记录长度应取 2，由此，原程序中的 140、330 号语句应改为：  
 $\text{bot} = \text{lof}(1) / 2$ ，这样修改后，被加密的文件就不会改变长度，而且，由于主循环次数缩小了一倍，运行速度也将近提高一倍。

2、原程序中加密解密子程序仅有个别语句的一个字符（即 210 与 367 号语句中的加减运算符）之差，只要把其中一个语句改成条件转向语句，就完全可以用一个子程序取代。原文作者可能是出于提高程序的运行速度，但仅一个语句之差而雷同于一个几乎完全相同的大段子程序，实在没有必要。

3、逐个输入密钥序列，使用不方便。可以改成一次输入口令，让程序自动产生密钥序列的方式。

经上述纠错改进后，被加密后的文件尺寸不变，大大提高了程序的运行速度，操作更为方便，程序长度压缩了五分之二，显得更为简练。改进后的程序清单如下：

1 REM 改进的面向 16 位字节流的加密解密工具

```

5 DIM J (100): CLS
6 REM 产生密钥
10 INPUT "请输入加（解）密口令 (N=100):"; N$
12 N=LEN (N$)
14 FOR I=1 TO N
16 J (I) =VAL (MID$ (N$, I, 1))
18 PRINT J (I);
20 NEXT: PRINT
22 PRINT "1——加密 2——解密 0——退出!";
23 C$=INPUT$ (1): PRINT
24 IF C$="1" OR C$="2" THEN 30
25 END
30 GOSUB 100
40 CLS: GOTO 10
90 REM 加密、解密子程序
100 INPUT "请输入文件名 ([d:] [path] filename (ext):";
F$
120 OPEN "R", #1, F$, 2
130 FIELD #1, 2 AS A$
140 BOT=LOF (1) / 2
150 B=1
160 FOR I=1 TO BOT
170 IF B<N THEN 180
175 B=1
180 K=J (B)
190 GET #1, I
200 M%=CVI (A$)
210 IF C$="1" THEN M%=M%+K ELSE M%=M%-K
220 LSET A$=MKI$ (M%)
230 PUT #1, I
240 B=B+1
250 NEXT
260 CLOSE #1
270 RETURN
    
```

# 鼠标,你也能控制

香港金山公司 冯志宏

平时大家都喜欢使用鼠标,“大千世界,方寸能容”,一只鼠标几乎可以操作各种软件而不动键盘,令人爱不释手。但在使用鼠标的同时,你可曾想过自己编制一个支持鼠标的程序吗?

近来,笔者参与求伯君先生主编的《深入 DOS 编程》一书的校对工作,发觉其中关于 Mouse 的部分写的极其精彩,不但介绍了最新版的 MicroSoft Mouse 功能调用,而且深入浅出地论述了 Mouse 的具体使用,令笔者茅塞顿开,对鼠标的使用有了清楚的认识,经作者同意,择其精要先行公布,以飨读者。

鼠标的支持需使用鼠标中断服务程序,该功能号为 33H。别看其中功能调用一大堆,与编程关系最紧密的只有几个,即:

## 1. 功能 00: 初始化鼠标

该功能确定是否安装了鼠标;如果鼠标存在,将驱动程序复位,并返回鼠标的按钮数。本功能的调用方法为:

调用寄存器: AX 0000H

返回寄存器: AX 0 未安装鼠标

0FFFF 安装了鼠标

BX 按钮数 (Microsoft 为 2, 有些其他牌子的为 3)

## 2. 功能 01H: 显示鼠标光标

该功能在显示器上显示鼠标的光标。本功能的调用方法为:

调用寄存器: AX 0001H

返回寄存器: 无

## 3. 功能 02H: 关闭鼠标光标

该功能关闭鼠标光标的显示。本功能的调用方法为:

调用寄存器: AX 0002H

返回寄存器: 无

## 4. 功能 03H: 取鼠标位置

该功能返回当前鼠标位置和按钮状态。本功能的调用方法为:

调用寄存器: AX 0003H

返回寄存器: BX 按钮状态

CX X 坐标 (水平)

DX Y 坐标 (垂直)

不论屏幕为什么方式,本功能总是返回 X 坐标 (列, 0 到 639 之间) 和 Y 坐标 (行, 0 到 199 之间)。下表显示了鼠标光标对每种显示方式允许的位置,以屏幕坐标 (象素) 表示。

| 屏幕方式     | 鼠标坐标                         |
|----------|------------------------------|
| 00H, 01H | X = 16 * 列, Y = 8 * 行        |
| 02H, 03H | X = 8 * 列, Y = 8 * 行         |
| 04H, 05H | X = 2 * 屏幕坐标 X, Y = 屏幕坐标 Y   |
| 06H      | X = 屏幕坐标 X, Y = 屏幕坐标 Y       |
| 07H      | X = 8 * 屏幕坐标列, Y = 8 * 屏幕坐标行 |
| 0EH-10H  | X = 屏幕坐标 X, Y = 屏幕坐标 Y       |

鼠标按钮的状态由 BX 返回,只有低位有效。下表说明各位的意义。由于各按钮的动作是独立的,对两按钮鼠标而言该值在 0 到 3 之间,对三按钮鼠标而言在 0 到 7 之间。

| 鼠标按钮状态位    | 含义          |
|------------|-------------|
| 位 76543210 |             |
| .....0     | 左边按钮放开      |
| .....1     | 左边按钮压下      |
| .....0.    | 右边按钮放开      |
| .....1.    | 右边按钮压下      |
| .....0..   | 中间按钮放开 (若有) |
| .....1..   | 中间按钮压下 (若有) |
| XXXXX...   | 未定义         |

## 5. 功能 09H: 置图形鼠标形状

该功能设置绘图方式下所用的鼠标形状。本功能的调用方法为:

调用寄存器: AX 0009H

BX 热点 X 位置 (-16 到 16)

CX 热点 Y 位置 (-16 到 16)

ES:DI 指向视屏掩码和光标掩码的指针

返回寄存器: 无

使用该功能调用,可以在图形屏幕上显示出漂亮的图形鼠标,尽管该功能调用稍烦,笔者还是建议大家好看一看。

在图形方式下,光标是由视屏掩码,光标掩码和热点来定义的。功能 09H 全部做了定义。

这一概念在鼠标驱动程序中最容易混淆的,需详细说明。为了提供一个检查的框架,对软件来说,可见鼠标总是定义为一个 16 \* 16 方块,热点是此方块左上角对应的单一象素地址,这正是正式的光标位置。在方块内,可见的指示形状是由视屏掩码和光标掩码确定的。

当鼠标在图形方式下产生时,视屏掩码同屏幕做与

运算,而光标屏蔽与结果做异或运算。

举个例子,如果视屏掩码字节为 9CH,另一字节为 3BH,把它们作与运算的结果如下:

```

AND   9CH   10011100
      3BH   00111011
-----
      00011000
    
```

如果光标掩码字节为 9CH,而另一字节为 3BH,把它们作异或运算的结果如下:

```

XOR   9CH   10011100
      3BH   00111011
-----
      10100111
    
```

这些操作对图形方式 1 至 6 的实际效果见如下每位:

|       | 视屏掩码位 |   |
|-------|-------|---|
|       | 0     | 1 |
| 光标掩码位 | 0     | 1 |
| 变化    | 反视    |   |

对视屏方式 7 以上,效果如下:

|       | 视屏掩码位 |   |
|-------|-------|---|
|       | 0     | 1 |
| 光标掩码位 | 0     | 1 |
| 黑     | 白     |   |
| 无变化   | 无变化   |   |

在 ES:DX 所指位置存储的掩码是 16 位字的位映象块,位映象的每一个字对应于 16 行光标的一行(从最上面的行开始,视屏掩码在先)。字 0-15 是视屏掩码。字 16-31 是光标掩码,下图显示了这种对应方式。

对应光标位置字节

偏移 0 1 2 3 4 5 6 7 11 12 13 14 15  
掩码

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

第 0 行(字节 0)  
第 1 行(字节 1)  
第 2 行(字节 2)  
第 3 行(字节 3)  
第 4 行(字节 4)  
第 5 行(字节 5)  
第 6 行(字节 6)  
第 7 行(字节 7)

在图形方式 4-6 和 14-16 下,每一光标象素有两位,即使光标大小不同也如此。

视屏掩码暂时擦去光标区域,然后光标掩码将指示字形画到这一空区,这导致鼠标指示字好象覆盖了屏

幕数据。如提供几套掩码,且使用这一功能在其间切换,就可以在程序的不同部分使用不同的指示字形。

热点坐标总是与 16 \* 16 象素区的左上角有关,且这一区域是由两个掩码屏蔽的。热点坐标也定义了一个特定的坐标,每当任何功能要求鼠标信息时,它将作为鼠标位置返回。

设置热点为 (0, 0) 将使图象的左上角成为正式鼠标位置(缺省的箭头图形的尖点),设置为 (8, 8) 将使区域中心为正式位置, (16, 16) 将使正式位置移到右下角。由于 (-16, -16) 是一个合法热点设置,正式位置也可移到实际图象外边——虽然很难设想出好的理由去这样做。

#### 6. 功能 0FH: 设置鼠标移动单位与象素的比值

该功能设置光标物理移动(以 mickey 计算,为 1 英寸的 1/200)和坐标变化(象素)的比值。本功能的调用方法为:

调用寄存器: AX 000FH

CX 水平位置变化 8 个象素时所需 mickey 数,缺省为 8

DX 纵向位置变化 8 个象素时所需 mickey 数,缺省为 16

返回寄存器: 无

在 CX 和 DX 两个寄存器中,高位必须为 0。最小比值为 1。比值愈高,鼠标在屏幕上移动愈慢。

当然,Mouse 的功能调用远不止这几个,其它新增加的功能更加强劲,只是作者给我的 Copyright 只有这么多,想深入了解 Mouse 功能调用的读者只好去跑书店了。

看了这几个功能的简介,不知你明白了多少,幸好作者附书提供了一个例子程序,对这几个功能进行 DEMO。别小看这个 DEMO,你可以 COPY 其中的例程挂进自己的程序中,编出漂亮的软件,如有佳作,到时候别忘了寄给笔者一份。

```

{*****}
{      鼠标支持演示程序      }
{      作者: 求伯君 陈波      }
{*****}
program TestMouse;
uses Dos, Crt, Graph;
{      定义鼠标光标数据类型      }
type
  GraphCursMaskType = record
    Mask : array [0..1, 0..15] of word;
    HorzHotSpot, VertHotSpot : integer;
  end;
const
  BgiPath = '';
  MouseDelay = 250;
    
```

```

{      定义一个手状鼠标      }
StandardShapeCurs: GraphCursMaskType = (
Mask: (( $ 3fff, $ 1fff, $ 0fff, $ 07ff, $ 03ff,
        $ 01ff, $ 00ff, $ 007f, $ 003f, $ 001f,
        $ 01ff, $ 10ff, $ 30ff, $ f87f, $ f8ff, $ fc3f),
        ($ 0000, $ 4000, $ 6000, $ 7000, $ 7800,
        $ 7c00, $ 7e00, $ 7f00, $ 7f80, $ 7fc0,
        $ 7c00, $ 4600, $ 0600, $ 0300, $ 0180, $ 00c0));
HorzHotSpot: -1;
VertHotSpot: -1
);
{      定义一个箭头状鼠标      }
PointingHandCurs: GraphCursMaskType = (
Mask: (( $ e1ff, $ e1ff, $ e1ff, $ e1ff, $ e1ff,
        $ e000, $ e000, $ e000, $ 0000, $ 0000,
        $ 0000, $ 0000, $ 0000, $ 0000, $ 0000, $ 0000),
        ($ 1e00, $ 1200, $ 1200, $ 1200, $ 1200,
        $ 13ff, $ 1249, $ 1249, $ 1249, $ 9001,
        $ 9001, $ 9001, $ 8001, $ 8001, $ 8001, $ ffff));
HorzHotSpot: 5;
VertHotSpot: 0
);
var
  Regs: Registers;
  NumMouseKeys : Byte;
  MousePresent, MKP: Boolean;
  MKey : (None, Left, Right, Both);
  MouseX, MouseY, CellSizeX, CellSizeY: Word;
procedure SetKeyStatus(MStatus : Word);
begin
  Case MStatus Of
    0 : MKey := None;
    1 : MKey := Left;
    2 : MKey := Right;
    3 : MKey := Both;
  end;
end;
procedure ResetMouse;
var
  MouseInt: Pointer;
begin
  MKP := False;
  NumMouseKeys := 0;
  MousePresent := False;
  GetIntVec($ 33, MouseInt);
  if MouseInt <> Nil then begin
    Regs.AX := 0;
    Intr($ 33, Regs);
    if Regs.AX <> 0 then begin
      MousePresent := True;
      NumMouseKeys := Regs.BX
    end;
  end;
end;
end;
procedure VirtualScreenSize;
begin

```

```

  Regs.AH := $ 0F;
  Intr($ 10, Regs);
  CellSizeX := 1;
  CellSizeY := 1;
  Case Regs.AL Of
    0, 1 : begin
      CellSizeX := 16;
      CellSizeY := 8;
    end;
    2, 3 : begin
      CellSizeX := 8;
      CellSizeY := 8;
    end;
    7 : begin
      CellSizeX := 9;
      CellSizeY := 14;
    end;
    4, 5, 13 : CellSizeX := 2;
  end;
end;
procedure ShowMouseCursor;
begin
  Regs.AX := 1;
  Intr($ 33, Regs);
end;
procedure HideMouseCursor;
begin
  Regs.AX := 2;
  Intr($ 33, Regs);
end;
procedure GetButtonStatus;
begin
  Regs.AX := 3;
  Intr($ 33, Regs);
  With Regs Do begin
    SetKeyStatus(BX);
    MouseX := CX;
    MouseY := DX;
  end;
end;
procedure SetGraphicsCursor
(Var Mask: GraphCursMaskType);
begin
  With Regs Do begin
    AX := 9;
    BX := Word(Mask.HorzHotSpot);
    CX := Word(Mask.VertHotSpot);
    DX := Ofc(Mask);
    ES := Seg(Mask);
  end;
  Intr($ 33, Regs);
end;
Function LeftMouseKeyPressed : Boolean;
begin
  If MKP Then Delay(MouseDelay);
  GetButtonStatus;
  MKP := MKey = Left;
end;

```

```

LeftMouseKeyPressed := MKP;
end;
Function RightMouseKeyPressed : Boolean;
begin
  If MKP Then Delay(MouseDelay);
  GetButtonStatus;
  MKP := MKey = Right;
  RightMouseKeyPressed := MKP;
end;
Function BothMouseKeysPressed : Boolean;
begin
  If MKP Then Delay(MouseDelay);
  GetButtonStatus;
  MKP := MKey = Both;
  BothMouseKeysPressed := MKP;
end;
procedure DetectMouse;
begin
  ResetMouse;
  If Not MousePresent Then begin
    Writeln('Mouse or mouse driver not present');
    Halt;
  end;
  VirtualScreenSize;
  Writeln('Mouse present');
  Writeln('Mouse has ', NumMouseKeys:1, ' keys');
  Write('Press RETURN to continue demo ...');
  Readln;
end;
procedure HideAndShowDemo;
begin
  ClrScr;
  Writeln(
    'This demonstrates hiding and showing the cursor');
  Writeln(
    'Press <Left Mouse Key> to hide cursor ...');
  Writeln(
    'Press <Right Mouse Key> to show cursor ...');
  Write(
    'Press <Both Mouse Key> to exit ...');
  ShowMouseCursor;
  While Not BothMouseKeysPressed Do begin
    GetButtonStatus;
    Case MKey of
      Left : HideMouseCursor;
      Right : ShowMouseCursor;
    end;
    GotoXY(1, 23);
    Write('X: ', (MouseX div CellSizeX):3,
      ' Y: ', (MouseY div CellSizeY):3);
  end;
  HideMouseCursor;
end;
procedure GraphicsMouseDemo;
var
  S, S1, S2: String;
  StandardMouse: Boolean;

```

```

GrBkCol, GrFgCol: Word;
GraphDriver, GraphMode, GraphErrorCode: integer;
begin
  GraphDriver := Detect;
  InitGraph(GraphDriver, GraphMode, BgiPath);
  GraphErrorCode := GraphResult;
  If GraphErrorCode <> grOK Then begin
    ClrScr;
    Writeln(
      'Graphics error:', GraphErrorMsg(GraphErrorCode));
    Halt;
  end;
  GrBkCol := GetBKColor;
  GrFgCol := GetColor;
  OutTextXY(0, 0,
    'Press left mouse button to display position');
  OutTextXY(0, 10,
    'Press right mouse button to change cursor shape');
  OutTextXY(
    0, 20, 'Press both mouse button to exit ...');
  ShowMouseCursor;
  GetButtonStatus;
  Str(MouseX:3, S1);
  Str(MouseY:3, S2);
  S := 'X: ' + S1 + ' Y: ' + S2;
  SetColor(GrFgCol);
  OutTextXY(0, 100, S);
  StandardMouse := True;
  While Not BothMouseKeysPressed Do begin
    If LeftMouseKeyPressed Then begin
      SetColor(GrBkCol);
      OutTextXY(0, 100, S);
      Str(MouseX:3, S1);
      Str(MouseY:3, S2);
      S := 'X: ' + S1 + ' Y: ' + S2;
      SetColor(GrFgCol);
      OutTextXY(0, 100, S);
    end;
    If RightMouseKeyPressed Then begin
      StandardMouse := Not StandardMouse;
      if StandardMouse
        then SetGraphicsCursor(StandardShapeCurs)
        else SetGraphicsCursor(PointingHandCurs);
    end;
  end;
  HideMouseCursor;
  CloseGraph;
end;
begin
  DetectMouse;
  HideAndShowDemo;
  GraphicsMouseDemo;
end.

```

# 利用MSC5.0对CGA、EGA(VGA)的图形画面进行存屏取屏的方法

广州市经济研究院 耿卫东

我们经常要在图形方式下,为自己设计的软件作一些解释性、或装饰性的图形画面。美观实用的图形画面,不仅提高了软件的质量,而且增强了软件的用户友好性,受到用户的欢迎。

一般来说,做好一幅图形画面后,都采用存屏方式将之保存在内存或文件中,在使用时,再从内存或文件中读出来,在屏幕上显示。这样能够节省实现图形显示在软件中所占用的代码,也使图形能够以完整的画面一次显示出来,从而使图形显示更美观实用。

笔者运用 MSC5.0 分别针对 CGA、EGA(VGA)编写了完成存屏、取屏功能的程序,以下分别介绍一下对 CGA、EGA(VGA)完成存屏、取屏功能的实现方法及相应的程序。

CGA 的图形方式,可以用 BIOS 的 10H 中断设定,我们知道 CGA 图形缓冲区的起始地址为 B8000000H,当其显示模式被设置为 6H 时,屏幕上的像素点个数为  $200 \times 350$ 。CGA 可以利用调色板功能为每个像素点提供四种颜色的选择,故每个像素点需要两位来描述,这样,显示一幅 CGA 的图形画面至少需要 160000( $200 \times 320 \times 2 / 8$ )个字节。而 CGA 图形缓冲区的长度为 4000H(即  $16 \times 1024 = 16384$ ),尚有富裕的空间。同时,在 CGA 图形缓冲区中的偶数字节和奇数字节相隔 8152 个字节,根据以上对 CGA 图形缓冲区结构的了解,我们编写了分别利用内存或文件的方式,对 CGA 图形画面进行存屏、取屏的程序,见 CGA\_1、CCGA\_2.C。

EGA 显示器的结构及显示方式不同于 CGA,它拥有 4 个 64KB 的图形区域——位平面,分别为红、绿、蓝及加亮区,其图形缓冲区的起始地址为 A0000000H。当 EGA 的显示模式被置为 10H 时,其像素点个数为 224000( $350 \times 640$ ),由于每个像素点可以有 16 种颜色的选择,故需 4 个位来表示一个像素点,这样,每个位平面上的字节数为 22800( $350 \times 640 \times 4 / 4 \times 8$ ),四个不同颜色的位面正好组成 16 种颜色。从 EGA 的图形缓冲区读出屏幕图形

信息时,可以利用 EGA 图形控制器(端口地址:03CEH, 03CFH)中的位平面读出寄存器(索引号为 4)来控制 4 个位面,从而依次将相应的图形信息读出,并写入长度为 112000 的磁盘文件。恢复屏幕时,则首先要将图形信息从文件中读出来,然后利用 EGA 操作定序控制器(端口地址:03C4H, 03C5H)中的位平面屏蔽寄存器(索引号为 2)来控制向哪个位平面写入图形信息。

VGA 显示器的结构及显示方式类似于 EGA,当显示模式被置为 12H 时,它也拥有 4 个 64KB 的位平面,其图形缓冲区的起始地址为 A0000000H,像素点个数为 224000( $480 \times 640$ ),这样,它的每个位平面上的字节数为 38400( $480 \times 640 \times 4 / 4 \times 8$ )所以,只要将显示模式置换为 12H,将每个位平面上的长度置换为 38400,其余代码不变,即可对 VGA 的图形画面进行存屏、取屏。

根据以上对 EGA(VGA)图形缓冲区结构的了解,我们编写了利用文件的方式对 EGA(VGA)图形画面进行存屏取屏的程序,见 EGA\_SCR.C。该程序在 2.13H 汉字系统下运行,读出 2.13H 的 24 点阵字库并显示,然后完成存屏和取屏的功能。

```
/* cga-1.c: 利用内存 对 CGA 的图形 进行存 取屏 */
#include <dos.h>
#include <stdio.h>
unsigned char far buf[200][80];
char far *ptr = (char far *)0xb8000000;
char far *temp;
union REGS in, out;
main()
{
    register int i, j;
    setvideomode(6); /* 置 CGA 的图形方式 */
    for(i=0; i<24; i++)
        for(j=0; j<80; j++)
            printf("B");
    /* 将 CGA 的图形读入内存 */
    temp = ptr;
    for(i=0; i<200; i++)
        for(j=0; j<80; j+=2)
        {
            buf[i][j] = *temp; /* 读偶数字节 */
            buf[i][j+1] = *(temp+8152); /* 读奇数字节 */
            *temp = 0; *(temp+8152) = 0; /* 清屏 */
            temp++;
        }
    get-key();
}
```

```

/* 将 CGA 的图形由内存写入图形缓存区 */
temp = ptr;
for(i=0; i<200; i++)
    for(j=0; j<80; j+=2)
        { *temp = buf[i][j]; /* 写偶数字节 */
          *(temp+8152) = buf[i][j+1]; /* 写奇数字节 */
          temp++;
        }
get-key();
set-videomode(3); /* 恢复文本方式 */
}
/* 置图形模式 */
set-videomode(mode)
int mode;
{
    in.h.ah=0;
    in.h.al=mode;
    int86(0x10, &in, &out);
}
/* 清屏幕指定区域 */
cls(x1, y1, x2, y2, bgcolor, textcolor)
int x1, y1, x2, y2, bgcolor, textcolor;
{
    in.h.ch=x1;
    in.h.cl=y1;
    in.h.dh=x2;
    in.h.dl=y2;
    in.h.ah=0x06;
    in.h.al=0;
    in.h.bh=bgcolor*16+textcolor;
    int86(0x10, &in, &out);
}
/* 读键盘 */
get-key()
{
    in.h.ah=0;
    while(! kbhit());
    return(int86(0x16, &in, &out));
}

/* cga-2.c -- 利用文件 对 CGA 的图形 进行存 取屏 */
#include<dos.h>
#include<stdio.h>
unsigned char far buf[200][80];
char far *ptr = (char far *)0xb8000000;
char far *temp;
union REGS in, out;
main()
{
    char fname[80];
    FILE *fp;
    register int i, j;
    set-videomode(6); /* 置 CGA 的图形方式 */
    for(i=0; i<24; i++)
        for(j=0; j<80; j++)
            printf("B");
    /* 将 CGA 的图形写入文件 */
    if((fp=fopen("cga.pic", "wb"))==NULL)
        { printf("can not open file"); exit(0); }
    temp = ptr;
    for(i=0; i<8152; i++)
        { putc(*temp, fp); /* 读偶数字节 */
          putc(*(temp+8152), fp); /* 读奇数字节 */
          temp++;
        }
    fclose(fp);
    cls(0, 0, 24, 79, 0, 0);
    get-key();
    /* 将 CGA 的图形由文件写入图形缓存区 */
    temp = ptr;
    if((fp=fopen("cga.pic", "rb"))==NULL)
        { printf("can not open file"); exit(0); }
    for(i=0; i<8152; i++)
        { *ptr = getc(fp); /* 写偶数字节 */
          *(ptr+8152) = getc(fp); /* 写奇数字节 */
          ptr++;
        }
    fclose(fp);

```

```

get-key();
set-videomode(3); /* 恢复文本方式 */
}
/* 置图形模式 */
set-videomode(mode)
int mode;
{
    in.h.ah=0;
    in.h.al=mode;
    int86(0x10, &in, &out);
}
/* 清屏幕指定区域 */
cls(x1, y1, x2, y2, bgcolor, textcolor)
int x1, y1, x2, y2, bgcolor, textcolor;
{
    in.h.ch=x1;
    in.h.cl=y1;
    in.h.dh=x2;
    in.h.dl=y2;
    in.h.ah=0x06;
    in.h.al=0;
    in.h.bh=bgcolor*16+textcolor;
    int86(0x10, &in, &out);
}
/* 读键盘 */
get-key()
{
    in.h.ah=0;
    while(! kbhit());
    return(int86(0x16, &in, &out));
}
/* ega-scr.c: 利用文件 对 EGA 的图形进行存屏 取屏 */
#include <stdio.h>
#include <dos.h>
#include <conio.h>
#include <string.h>
#include <stdlib.h>
#include <graph.h>
FILE *fp;
char far *ptr=(char far *)0xa0000000; /* EGA 缓冲区起始地址 */
char far *temp;
unsigned int port, byte-val, memo-len;
union REGS in, out;
main()
{
    char *fname;
    set-videomode(16);
    cursor(0); /* 消除光标 */
    printf("%cC15D0, 349B635, 345]", 14); /* 框线 */
    printf("%cC15D1, 348B633, 343]", 14); /* 框线 */
    printf("%cC15D3, 346B629, 339]", 14); /* 框线 */
    printf("%cV%d, 4, 345, 627, 337, 1, 0, 0]", 14, 2); /* 框线着色 */
    printf("%c[-30]50# %d*9@广州社会经济综合]\n", 14, 2);
    printf("%c[-30]130# %d*12@决策支持系统]\n", 14, 2);
    printf("%c[-300]279# %d*9@A 广州经济研究院 1992.5]\n", 14, 2);
    fname = "gwa.pic";
    ega-save(fname); /* 对 EGA 的图形进行存屏 */
    set-videomode(16); /* 清屏 */
    get-key();
    ega-load(fname); /* 对 EGA 的图形进行取屏 */
    get-key();
    cursor(1); /* 恢复光标 */
    set-videomode(3);
}

/* 对 EGA 的图形进行存屏 */
ega-save(fname)
char fname[20];
{
    register int i, j;
    if (! (fp = fopen(fname, "wb")))
        { printf("cannot open file\n"); return(0); }
    memo-len = 0x6D60;
    temp = ptr;
    for (i=0; i<memo-len; i++)
        { for (j=0; j<-3; j++)
            { /* 位面号控制 */
              port = 0x03CE;

```



# 把握时机 施展才华 实现理想

```

byte-val= 4;
outp(port, byte-val);
/* 选位面号 */
port = 0x3CF;
byte-val= j;
outp(port, byte-val);
/* 存第j个位面 */
putc(*temp, fp);
}
temp++;
}
fclose(fp);
}
/* 对EGA 的图形进行取屏 */
ega-load(fname)
char fname[20];
{
register int i, j;
if ( ! (fp = fopen(fname, "rb")) )
{ printf("cannot open file\n"); return(0); }
memo-len=0x6D60;
temp=ptr;
for (i=0; i<memo-len; i++)
{ for (j=0; j<=3; j++)
{ /* 位面号控制 */
port = 0x3C4;
byte-val= 2;
outp(port, byte-val);
/* 选位面号 */
port = 0x3C5;
if (j==0) byte-val=1;
if (j==1) byte-val=2;
if (j==2) byte-val=4;
if (j==3) byte-val=8;
outp(port, byte-val);
/* 读第j个位面 */
*temp = getc(fp);
}
temp++;
}
}
fclose(fp);
}
/* 置图形模式 */
set-videomode(mode)
int mode;
{
in.h.ah=0;
in.h.al=mode;
int86(0x10, &in, &out);
}
/* 读键盘 */
get-key()
{
in.h.ah=0;
while(! kbhit());
return(int86(0x16, &in, &out));
}
/* 2.13H 的光标控制 */
cursor(n)
int n;
{
union REGS regs;
regs.h.ah=19;
regs.h.al=n;
int86(0x10, &regs, &regs);
}

```

86

电脑杂志社地处华南经济中心—美丽的广州市，《电脑》杂志以珠江三角洲为依托，面向全国电脑用户及计算机厂商。多年来《电脑》深受读者欢迎。

在一日千里的经济大潮里。电脑杂志社的业务量也在飞速膨胀。

您想在广告业方面有一个显示自己才华的舞台吗？您想展示自己超前的艺术构思吗？电脑杂志社广告部可以满足您的表现欲！

您想让全国的读者欣赏自己的编辑天才，您希望从未谋面的读者和自己交流思想，您想扬名天下吗？《电脑》杂志责任编辑的位置可以令您感到满足，感到欣慰。

您有“公关”的特长，您想广交天下朋友。电脑杂志社公关部是您的第一选择。

电脑杂志社需聘责任编辑、广告策划、广告业务员、广告设计与制作，公关、文秘等多名工作人员。你若想在电脑杂志这个大家庭里施展才华，那么请你把自己的履历、学历、血型、相片一张、职务和薪酬要求等资料寄到广州市华南师范大学电脑杂志社。邮编：510631。

有志者快把握时机，来实现自己的理想吧！

87

# 汉字放大显示工具库的发现

交通银行重庆分行 李 智

在众多的汉字放大显示实用程序中, 无论以何种程序语言实现, 其共同之处在于: 直接从汉字库中读取点阵字模, 并以图形画点方式进行显示。在实际应用中, 应用程序为了对某几个汉字进行放大显示, 而不得将整个汉字库随应用程序一道交付用户使用, 如为 2.13 的 24 点阵打印字库, 将占据 480K 以上存储空间, 致使汉字字模使用率极低。造成磁盘资源的巨大浪费。能否事先将应用程序所需的汉字字模拟汉字库中抽取出来, 形成一个供应用程序放大显示汉字使用的专用字库? 笔者使用 TURBO C 实现的 CCDOS 2.13 24 点阵汉字放大显示工具库, 它将使你方便地达到这一目的。

## 一、工具库内容:

汉字放大显示工具库包括以下程序:

### 1、专用字库生成程序 CRTLIB.C

由于汉字在机器内是以内码形式表示, 我们调用以下 C 语言函数计算该汉字在字库中的地址:

```
long int GetAddr(Byte1, Byte2)
unsigned char Byte1, Byte2;
{
    int Qnum, Wnum;
    long int Addr;
    Qnum=Byte1-0xa0;
    Wnum=Byte2-0xa0;
    if (Qnum>15) Qnum=-8;
    else if (Qnum==9) Qnum=6;
    Addr=(Qnum-1)*94+Wnum-1-658;
    return Addr;
}
```

其中: Byte1, Byte2 为汉字首, 尾字节。程序通过交互式汉字输入, 自动将汉字内码转换为字模地址, 从 CC DOS 2.13 24 点阵字库中读出字模, 把点阵信息以图形画点方式显示汉字, 并将内码, 字模组成记录, 写入文件。以此生成一个面向应用程序的专用字库, 其文件扩展名为 .DAT, 字模的使用率将达到 100%。

### 2、专用字库显示程序 DSPLIB.C

在专用字库生成之后, 常需要查询显示专用字库的内容。该程序顺序读取专用字库, 将读出的字模以图形画点的方式显示出来, 每屏 5 行, 每行为 10 个汉字。

### 3、专用字库索引文件生成程序 CRTIDX.C

为使汉字放大显示程序快速查找专用字库中汉字字模, 提高显示速度, 需为专用字库建立一索引文件。它通过读取专用字库, 动态分配一个结构数组, 用以存放

每一个汉字内码及其字模在专用字库中的起始地址, 并使用 TURBO C 中的 qsort() 函数对结构数组进行快速分类排序, 将排序后的结构数组一并写入文件。生成一个与专用字库文件同名, 而括展名是 .IDX 的索引文件。

## 4、汉字串放大显示程序 DSPFONT.C

从命令行参数中取得显示汉字串的内码, 以此作为索引文件键值, 使用 TURBO C 中 bsearch() 函数, 对索引文件进行二分法查找, 获得该汉字字模在专用字库中的起始地址, 以此定位专用字库文件指针, 读取汉字字模, 对字模进行放大显示。它支持如下特性:

- (1) 支持横向及纵向放大;
- (2) 支持汉字串水平及垂直显示;
- (3) 能控制汉字串的显示颜色, 调整字间距。

## 二、工具库的用途

- 1、应用程序界面设计;
- 2、在西文方式下显示汉字;
- 3、由于专用字库的引入, 可在西文方式下开发运行汉字软件, 以解决由于汉字操作系统加载, 内存不够, 无法运行汉字应用软件的问题。

## 三、用法示例

### 1、CRTLIB HZK24K MYLIB <Enter>

以交互方式输入汉字, 从 HZK24K 点阵楷体字库中提取字模, 生成专用字库 MYLIB.DAT

### 2、DSPLIB MYLIB <Enter>

顺序显示专用字库 MYLIB.DAT 的内容。

### 3、CRTIDX MYLIB <Enter>

据专用字库 MYLIB.DAT 的内容, 形成索引文件 MYLIB.IDX。

### 4、DSPFONT MYLIB 100 50 2 1 5 0 "交通银行"

在(100, 50)处, 以横向放大 2 倍, 纵向放大 1 倍, 字间距为 5, 水平显示字串 "交通银行"。

## 四、程序清单

由于篇幅有限, 在此仅附部份源程序。对本工具库感兴趣的读者, 请与笔者联系。

FONTHED.H 汉字放大显示工具库头部文件;

DSPFUNC.C 汉字放大显示工具库公共函数;

CRTLIB.C 专用字库生成程序。

```

/* WORDHED.H :
   汉字放大显示工具库头部文件 */
#define SIZE 72L
#define MASK 0x80
#define BGNX 180
#define BGNX 180
#define BGNX 100
#define ESC 27
#define DOTNUM 24
#define NAMELEN 13
typedef struct idxitem {
    unsigned char    Key[2];
    unsigned long    link;
}    IDXITEM, *PIDXITEM;

#ifdef MAIN
#define LINGAP 26
#define COLGAP 26
#define WORDNUM 10
#define WORDLIN 5
typedef struct rword {
    unsigned char    Key[2];
    unsigned char    Bitmap[72];
}    RWORD;

char    Bits[72];
char    * ErrMsg[]={
    "Can't Open file %s in r+b mode ! ",
    "Fseek fail",
    "Usage : DSPWORD fname",
    "Press ESC key for Exit ...",
    "Press any key for Exit ...",
    "输入汉字 — ",
    "Usage : GETWORD sfname tfname",
    "Can't Open file %s in a+b mode ! ",
    "Usage : CRTLIB sfname tfname",
    "Usage : DSPLIB fname",
    "Usage : CRTIDX fname",
    "Usage : DSPFONT fdat color bgnx bgny
xtimes ytimes chgap direct str",
    };
char Datfile[NAMELEN];
char Idxfile[NAMELEN];
#endif

```

```

/* DSPFUNC.C
   汉字放大显示工具库公共函数 */
#include <stdio.h>
#include <graphics.h>
#include "fonthed.h"
#ifdef NODSP
DspChar(dot, color, bgnx, bgny, xtime, ytime)
int    color, bgnx, bgny, xtime, ytime;
char    *dot;
{
    int    mask, i, j, k;
    int    xt, yt, ch, x, y;
    mask=MASK;
    x=bgnx;
    y=bgny;
    for (i=0; i<SIZE; i+=3) {
        for (xt=0; xt<xtime; xt++) {

```

```

            for (k=0; k<3; k++, i++) {
                for (j=0; j<8; j++) {
                    ch=dot[i]& mask;
                    for (yt=0; yt<ytime; yt++) {
                        if (ch) putpixel(x, y, color);
                        y++;
                    }
                    mask=mask>>1;
                }
                mask=MASK;
            }
            y=bgny;
            x++;
            i-=3;
        }
    }
}

#endif
Comp(PIDXITEM p1, PIDXITEM p2)
{
    return memcmp(p1->Key, p2->Key, sizeof(p1->Key));
}
DspTab(PIDXITEM p, int Num)
{
    int i;
    for (i=0; i<Num; i++) {
        print(p);
        printf("    %5d\n", p->link);
        p++;
    }
}
print(PIDXITEM p)
{
    int i;
    char *str;
    for (str=p->Key, i=0; i<sizeof(p->Key); i++)
        putchar(*str++);
}
Help(char *msg)
{
    puts(msg);
    exit(1);
}

```

```

/* CRTLIB.C
   专用字库生成程序, 以如下方式编译连接:
   TCC CRTLIB DSPFUNC GRAPHICS.LIB */
#include <stdio.h>
#include <mem.h>
#include <graphics.h>
#define MAIN
#include "fonthed.h"
#define NUMBER 100
static    RWORD word[NUMBER];
long    int GetAddr();
main(int argc, char * argv[])
{
    long int num;
    FILE *fp, *tp;

```

```

int driver, mode, element;
unsigned char first, second;
if (argc != 3) Help(ErrMsg[8]);
if ((fp=fopen(argv[1], "r+b"))==NULL) {
    printf(ErrMsg[0], argv[1]);
    exit(1);
}
strcpy(Datfile, argv[2]);
if (! strchr(Datfile, '.')) strcat(Datfile, ".dat");
if ((tp=fopen(Datfile, "a+b"))==NULL) {
    printf(ErrMsg[7], Datfile);
    exit(1);
}
driver=DETECT;
mode=0;
initgraph(& driver, & mode, "");
element=0;
gotoxy(20, 6);
printf(ErrMsg[5]);
gotoxy(20, 20);
puts(ErrMsg[3]);
while(1) {
    gotoxy(40, 6);
    if ((first=getche())<0xa0)
        if (first==ESC) break;
        else continue;
    else second=getche();
    num=GetAddr(first, second);
    if (fseek(fp, num*SIZE, SEEK-SET)) {
        puts(ErrMsg[1]);
        exit(1);
    }
    fread(Bits, SIZE, 1, fp);
    word[element].Key[0]=first;
    word[element].Key[1]=second;
    memcpy(word[element].Bitmap, Bits, SIZE);
    DspChar(Bits, 0x8a, BGNX, BGNX, 1, 1);
    delay(1000);
    DspChar(Bits, 0x00, BGNX, BGNX, 1, 1);
    ++element;
}
fwrite(word, sizeof(RWORD), element, tp);
fclose(tp);
fclose(fp);
restorecrtmode();
}
long int GetAddr(Byte1, Byte2)
unsigned char Byte1, Byte2;
{
    int Qnum, Wnum;
    long int Addr;
    Qnum=Byte1-0xa0;
    Wnum=Byte2-0xa0;
    if (Qnum>15) Qnum=-8;
    else if (Qnum==9) Qnum=6;
    Addr=(Qnum-1)*94+Wnum-1-658;
    return Addr;
}

```

88

## 广东省青少年信息学竞赛信息

今年省青少年信息学竞赛定于 93 年 5 月 16 日举行初赛, 7 月中旬举行复赛, 初赛为笔试, 以 BASIC 语言命题, 也可用其他语言解答, 主要考核学生计算机基础知识和基本的编程的能力, 初赛采用通讯赛形式, 由省统一命题, 复赛是在初赛的基础上, 由省竞赛委员会根据各地计算机活动开展情况及去年竞赛成绩分配名额, 以市为单位组队参赛, 复赛分一试试和二试, 属上机解答题型, 一试试采用多道题目的形式, 全面考核学生计算机程序设计能力, 二试按照国际信息学奥林匹克竞赛和全国信息学竞赛的要求命题, 有一定的难度, 复赛采用全国信息学竞赛指定语言版本 (Q BASIC4.5 版, TURBO PASCAL5.5~6.0 及 TURBO C 2.0 或 C++ 2.0)。

另今年 4 月 25 日举行全省小学生 LOGO 语言程序设计通讯赛, 竞赛采用笔试的形式在各市举行。

(郭) 89

## 广告索引

1. 珠海经济特区科达电源设备厂
2. 广利电脑产品
3. 广州市科教电脑设备厂
4. 华力科技开发公司
5. 广州电子设备公司
6. 赛宝星河
7. 星辰电脑
8. 天河电子
9. 特强 (广州) 电子有限公司
10. 天河新一代电脑技术开发部
11. 清化大学科学馆
12. 艾西显示设备有限公司
13. 华粤电子系统公司
14. MACS 经营管理控制系统
15. 广州市中原电脑科技公司

90

## 更正

1992 年第 6 期广告 14 页的华粤电子系统公司广告中的 "Acer 电脑产品系列一览表" 应改为 "EISA 电脑产品系列一览表"。在此特向华粤电子系统公司致歉。