

电脑与法律

- 对计算机软件法律保护的几点看法
..... 王桂海 (2)

电脑应用

- 电脑在广告设计中的应用..... 何晓田 (4)
高校学籍管理系统的设计..... 严婉庄 (7)

软件纵横

- MIS 开发方法的新设想——系统界面法
..... 吕廷振 (8)

网络与通信

- 高级 UNIX 连网技术讲座 第三讲
NFS 和 RPC..... 冯家宁 (11)
谈如何提高 XENIX 系统的安全性
..... 林荣庆 (16)
UNIX 系统使用终端时可能出现的问题
..... 王连军 (18)
本地硬盘的备份与恢复 宋 捷 (19)

书讯

- 全国电子报刊联合征订简明目录表 (20)

使用与维修

- NDD、DISKEDIT 修复软盘一例
..... 潘 敏 (21)
东海 0520C 软驱维修一例 杨海宁 (22)
LQ1600K 打印机纸尽检测器故障
..... 柯 纯 (22)
微机开关电源的维修方法与实例
..... 冯晓沁 (23)
用 7805 稳压块巧修四通 MS-2401 打字机
..... 陈丁君 (24)

IDEA

- 巧用 FOX 函数简化程序提高效率
..... 符卜成 (25)
谈 PCTOOLS 对深层子目录的加密
..... 储汉宾 (26)
删除子目录的简捷方法 杨 扬 (26)

大学生之页

- DOS 状态下的菜单制作 徐富军等 (27)
利用递归方法绘制分形图案 徐志凌 (28)

中学天地

- 中华学习机 A,B,T 类文件的相互转换
..... 翁元祥 (30)

竞赛与考试

- 1993 年广东省青少年奥林匹克信息学竞赛试题
分析及参考解答..... 邬家炜 (31)

NEW

- 谈谈 MS-DOS5.0 新增命令 DOSKEY
..... 唐银红 (34)
新一代驱动器——光磁软盘系统 陈明忠 (38)

游戏乐园

- 波斯王子二代攻略秘诀 梁宇舫 (39)
三国志游戏数据存放 陈 涛 (40)
《神州八剑》游戏攻略心得 曹宇杰 (40)

病毒防治

- 神秘的“1991”病毒 胡向东 (41)
一种新型的计算病毒-V300E 颜岳军 (43)

单片机与单板机

- 单片微机与模糊控制讲座 第五讲 自组织
模糊控制方法 余永权 (45)

电脑用户

- CPAV 口令密码字的破译 曹小忠等 (49)
GW386 上的初始配置信息文件 彭起顺 (50)
鼠标激活内存驻留程序方法 姜金友 (51)
Turbo C 中使用汉字 陈兆前等 (53)
文本方式下的汉字显示 刘树东等 (55)
赋值运算符“=”在 ORACLE 数据库的
PRO * C 程序接口中的妙用 陈 实 (57)
在 MSDOS5.0 下为 2.13H 和 SPDOS 合理
地配置 CONFIG.SYS 文件 陈君阳 (58)
自适应各种汉字系统的屏幕保存 / 恢复程序
..... 罗 辉 (60)
一种简单有效的 C 程序加密方法
..... 尹敬东等 (61)
一类链表丢失的文件修复方法 程江勇 (61)

编读往来

- 广告索引 (52)

对计算机软件法律保护的一点看法

华南师范大学 王桂海

我国著作权法第三条(八)款规定,计算机软件属于作品,受到该法保护。中国国务院根据著作权法第五十三条,于1991年6月4日公布了《计算机软件保护条例》(以下简称“条例”)。《条例》从1991年10月1日施行到现在,已经快两年了。这两年来,软件法律保护工作取得了相当的成绩,推动了软件产业的发展。但是,到目前为止,这仍是一件新事,还需要花很大力气去宣传普及。因为在过去相当长的一段时期内,软件抄袭的风气很盛,软件工作者守法意识还不够强,不可能在短时间里就对著作权法和“条例”有较深入的了解。现在,不少人还认为这只是法律界的事情;认为软件情况特殊,“条例”只是应付门面的事,起不到应有的法律作用;认为“条例”执行难度很大,不会有多大效果等等。总的来说,就是还不知道这一法律可以保护自己,也不知道这一法律正在保护他人。不懂得主动地运用著作权法这一法律武器维护自身权益,或者触犯著作权法而不自知,甚至盲目地以身试法。这后果将是严重的。这和我国已参加伯尔公约,并且根据1992年签署的中美知识产权谅解备忘录,要把软件按照该公约定义的文学、艺术作品来保护这一事实极不相称的。所以,是主动地运用著作权法去保护自己的权益,去约束自己的行动;还是出了问题再去找法律甚至受制于法,仍然是一尖锐问题。

一、中国软件法律保护的特殊性

由于计算机软件是以特定文字(符号或计算机专用语言)来描述的,和一般作品有共同之处,许多国家经过反复考虑,把它置于版权法保护之下。这是成功的做法[1]。但是,与其它作品相比,软件有其独特的地方:它所用的符号与语言是普通人难以读懂的,抄袭者的伪装较容易实现;它功能性很强,能直接产生效益;它易于复制(在机器支援下只需很短的时间);它所反映的是科学规律,作品的共性很大,创作自由度要比一般的文学作品少,作者的独创性不易通过其形式来表现;它受到硬件的限制,由硬件的相似而导致软件表达形式的相似易于引起纠纷等等。这使软件的诉讼案处理起来难度很大,在立法之后,仍然急切呼唤一套新的科学的判案准则和方法。

对中国来说,除了上述的问题之外,还有一些特殊情况:

(一)兼容性软件的合法地位。我国计算机硬件走国际兼容的道路,产品兼容程度一般到达逻辑兼容以上。这样,系统软件必然和原有的外国机器(称为目标机)兼容,而且事实上十分接近。这便很容易被指控为与目标机

系统软件有“实质相似性”(substantial similarity,这是指控某一作品侵权的必要条件)。所以如何解决兼容软件的合法地位是一个迫切的问题[2]。

(二)我国较长时间没有软件保护法,不少软件工作者长期在消化、分析国外软件中开展工作,常常发表文章,公布成果,而较少注意这些软件的来源及版权归属,这便等于承认与国外各类软件有接触(access),而这是指控作品侵权的另一必要条件。与前一条件合一起来,便构成被指控侵权的充分条件。怎样处理“接触”是另一个重要问题[3]。

(三)我国计算机软件发展晚,除了汉字信息处理技术及其它某些领域方面处于独创,领先地位外,大量是国外原版软件的衍生物,这方面的版权保护范围也是一个敏感的问题。

(四)要发展我国软件产业必然使软件出口创汇,面对的是一个复杂的知识产权斗争环境。以我国如此短浅的不完备的软件保护经验对付这个局面,条件是远不如日本、韩国、菲律宾、匈牙利、保加利亚、澳大利亚等等国家的。

因此,对于软件法律保护问题应加倍研究而绝不能掉以轻心。

二、对策、措施、建议

一提到对策、措施,很容易产生这样的反问:这一套人家(国际上)承认不承认,有没有用?其实,这是本末倒置。首要的问题是,针对国际上知识产权之争,我们有没有一套适合中国国情的对策和措施。只有有了自己的对策和措施,而且不断丰富完备,才有被国际上承认的可能。目前,世界上虽然有了公约,但处理具体案件时,并没有一成不变的模式,法庭判案的“弹性”很大。要是没有自己的整套观点、方法,办起案来,便只好沿用别人的做法,结果是受制于人。而如果我国能通过深入研究,形成自己的理论和方法,形成“一家(中国)之言”,在国内通过实践,逐步取得共识,得到认可和支持,便有了发言权,有一份在国际上“争雄”的力量,起到保护自己,制约他人的作用。在此基础上,才谈得上争取国际的承认。

另外还要认识到解决软件保护,不仅是个法律问题,而且还是关系到计算机科学技术的问题,这有点象“法医学”。既有“法”的一面,又有“医”的一面,“医”一面要为“法”这一面提供科学的依据。计算机技术也要为法庭对软件判案提供科学的根据。希望有关方面能把软件保护研究列为自然科学的重要课题,开展深入的专门性的研究。

当前比较迫切要解决的几个问题是:

在理论上

(一) 软件的表达与思想的分界线。著作权保护作品的表达形式,但不保护作品的思想。一般文学作品对“思想”与“形式”比较易于划分,但在软件方面,国际上则尚有许多争议,即使是把软件纳入著作权法保护多年的国家,如美国,现在对此都没有定论。八十年代中期,美国通过一些判案,把软件表达形式扩大到序列(sequence),结构(structure)和组织(organization)的非文字层次上(即所谓S.S.O.准则)。这个判案被多次引用,但在美国、日本、中国也都引起了异议。而且,也有不按此原则处理的案例(4)(5)。对这个关键问题,我国是要认真研究和提出看法的。

(二) 对于“可观与可感”(Look & Feel)即人机界面,在程序中应看作是创作的形式还是属于标准化或公知范围的引用,要进行研究,因为这也是目前美国争议较多的问题(6)。

(三) 对于一个程序的“创新性衍生产品”;对于运行在同一硬件上的兼容软件,如果其模块的“接口”、“断点”(有点象文学作品的章节)相同,但语言和编码不同,能否取得版权等等。由于这些问题与我国关系密切,要有充分反映我国实际状况的观点。

当然,还有其它的值得在理论上探讨的问题,如专利和著作权法的分工,标准化引用,如何反对垄断等等。

在方法上

(一) “实质相似性”的差别标准及其量化。作者曾提出了VML法和它的一些方法,且用于判定国产紫金计算机的磁盘操作系统(ZIDOS)的版权,但这仅仅是初步的认识,仍有待于进一步完善[7]。

(二) 在“接触”的判别上要解决我国净室(cleanroom)建立的条件、规程、提出法定的模式。

(三) 要对著名案例(目前主要目标是美国、美日之间、欧美之间的判案)作出及时的评论(做舆论基础)。鲜明的提出既符合国际版权公约,又反映中国立场的观点。我们已参加伯尼尔公约,大可以对国外判案在科学上,法律上“评头品足”。现在国内苦于信息不灵,不易取得判案的原始材料,应进一步加强国际交流。

在机构上

(一) 软件登记机构,国内已设立,并正常工作。最近(8月5日),北京市高、中级人民法院正式建立知识产权审判庭,其审理范围包括计算机软件著作权纠纷,这是又一件有重大意义的事件。

(二) 关于仲裁机构

前面谈到,由于软件有其复杂性,亟需有仲裁机构来提出相应的科学论据。这个机构应当是合法的,有权威性的。组成人员是计算机科学工作者和法律工作者(以前者为主)。而且,有些争议、纠纷,可以由仲裁机构而不必由法院来解决,美国IBM对富士通的软件纠纷,便根据1983年双方协定,在美国仲裁协会(American Arbitra-

tion Association,简称A.A.A.)作出最后仲裁后得到解决(8)。初步设想,仲裁协会的早期任务是:

- 判断告诉的软件与原作相比是否已有“实质相似性”
- 对安全机构(Secured Facility SF,净室的隔离层)的审定
- 对进出SF信息的判断(是否只牵涉思想,不涉及形式)
- 对争议双方提出合理要求(包括费用)
- 对敏感性问题(思想与表达的同一体性;表达形式的唯一性,作品的创作自由度限制,等等)作出研究或裁决

仲裁机构设想置于版权局的领导之下,可以是兼职的、民间的,权威性不是基于行政方面而在于仲裁的科学性、准确性。经费可以由争议的双方提供。

三、对《条例》的一点商榷

笔者认为,我国的软件保护条例第31条规定:“因下列情况之一而引起所开发的软件与存在的软件相似,不构成对已经存在的软件的著作权的侵犯:

- (一) 由于必须执行国家有关政策、法律、法规和规章;
- (二) 由于必须执行国家技术标准;
- (三) 由于可供选用的表现形式种类有限。”

有几个地方似可商榷:

(1) 软件相似并不是侵权的充分条件,根据著作权的特性,相似本身并不一定构成侵权,所以,可否改为:“...不构成与已存在的软件有实质相似性”。

(2) 对于上引条文的(一)、(二),可否改为由于“引用公知范围和标准化”。因为某个国家制定有关政策是什么,很难说。假如某国制定一项可以抄录他国软件的政策,另外的国家会天真地认为这不是侵权吗?

(3) 对于上引文的(三),表现形式种类有限,可否改为“表现形式是唯一的或在常识范围内可认定是极少的”。因为“有限”数可以是很大的,一个是有限,一万,十万,也还是有限——这样的提法会使法官为难的。

主要参考文献

- [1]、[3] 应明 计算机软件的版权保护 北京大学出版社 1991年
- [2] 王桂海 略论计算机软件版权保护与兼容软件(上),(下) 计算机世界 1990(23)(24),3
- [4] 应明 美国对软件版权纠纷判处的新发展 著作权 1993年(1)
- [5] 金渝林 从美国计算机程序著作权判例法的发展探索程序著作权保护的界限 著作权 1993年(1)
- [6] R.C.Johnson Spreadsheet copyrights may prove hard to get EET.Sept.4, 1989
- [7] 王桂海 软件相似判别准则之讨论 计算机科学 1992.(5)
- [8] 中国软件登记中心等编 计算机软件保护工作手册 电子工业出版社 1993年P.187-191

电脑在广告设计中的应用

广东省广告公司创作部 何晓田

一、概述

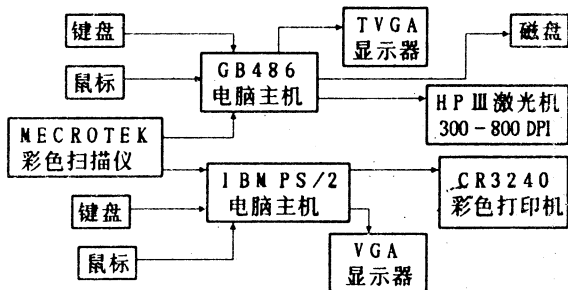
计算机图形图像处理技术是近一二十年以来随着半导体集成电路技术和计算机科学技术的迅速发展而产生的一门新兴学科，并且取得了巨大的进展。到80年代，数字图形图像处理进入成熟阶段。CPU处理器技术的发展使计算机的运算速度大大提高，大规模集成电路（LSI）和超大规模集成电路（VLSI）技术的发展，使计算机的内存、外存和高速缓存（Cache）及显示卡存储器的容量都得以大幅度提高。这使计算机图形图像处理能力大为增强，在国民经济和国防建设的各个领域得到了广泛的应用。比如：卫星遥感、医学诊断、公安管理、文字识别、计算机辅助设计等等。

计算机辅助设计（CAD）包括用计算机辅助进行的各种设计，如：建筑设计、机械设备设计、工业产品设计、集成电路设计、服装设计、平面广告设计、三维动画设计等等。其中平面广告设计在近些年里得到迅速的发展，涌现出一批优秀的设计软件，大大丰富了广告设计的创意手法和表现力，提高了设计的精度和速度。本文是笔者在平面广告设计工作实践中的一点粗浅体会。

二、设计系统的硬件构成

目前在平面设计领域，有两大系列的个人电脑：即 Apple 公司的 Macintosh 系列机和其它众多公司的 IBM PC 及其兼容机系列。Macintosh 系列机的特点是专业化，不论是图形图像处理的质量、功能和速度等方面均有较高水准，深受专业设计公司的欢迎。不足之处是它的汉字处理功能较差，造价也较高。而 PC 系列机虽然在图形图像处理的质量上略低于 Macintosh 机，但它具有很强的通用性，世界大多数公司生产的和我们通常使用的都是 PC 系列及其兼容机。随着高速处理 CPU、大容量内存外存设备、高分辨彩色显示、高精度输出设备的发展；随着 Windows 操作环境和一大批应用软件问世，改变了 PC 机传统的操作方式，使 PC 系列机完全具备了处理图形图像的强大功能。而且国内和海外开发的各样汉字系统大都可以配备在 PC 系列机上，包括高精度的矢量或轮廓汉字输出。这使得 PC 系列机在很多方面的性能接近或超过了 Macintosh 系列机，具有较高的性能价格比，而且可以在原有的 PC 系列机的基础上通过增加一些设备或附件来构成较高档次的设计系统。因此 PC 系列机正越来越多地为一般设计部门所采用。

我们创作部设计系统的硬件部分，是由前两年所购置的一台 IBM PS/2 SX 55 型机、HP III 激光打印机和 CR3240 彩色 24 针打印机组成的。在 Windows 3.0 和其它设计软件的支持下，可进行一般的设计工作。后来由于应用范围的扩大、设计软件的增多和版本的更新，原有系统在处理速度和存储容量方面已远远不能满足应用的要求；另外越来越多的设计人员开始应用电脑进行设计。因此我们又增加了一台 GB486/33 电脑，总线结构为 EISA 位总线，内存加至 8MB，硬盘容量为 200MB，显示卡为 TVGA，显示存储器加至 1MB，可显示 1024×768×256 色。这样使主机的处理图形图像的能力大大加强，运行 Windows 和设计软件的速度明显提高。另外在外设方面，我们从香港购进了 WinJet800 激光机分辨率增强卡，使 HP III 激光机的输出精度从原来的 300DPI 提高到 800DPI。最近又新购了一台分辨率为 1200DPI 的 MECROTEK 台式彩色扫描仪。从而组成了一套完整的电脑设计系统。系统硬件结构如下图：



系统硬件结构图

从上图可以看出，左侧的输入设备是键盘、鼠标和扫描仪；中间处理设备是电脑主机；右侧的输出设备是显示器、激光打印机、针式彩色打印机以及磁盘文件（磁盘文件用于发外作 1000~3000DPI 高精度输出）。

三、系统的软件环境与 Windows 系统

在平面设计中已有很多优秀的设计软件问世，大部分是在 Windows 操作环境下运行的。我们所选配的基本常用软件有：MS-DOS5.0 与 Windows3.1 操作系统、CorelDRAW 绘图设计软件、PageMaker 文字排版软件、PhotoStyler 图像处理软件以及尖端中文系统、华光中文

系统等。这些组成了一个较为完整的设计软件平台，其中最基本的支撑软件就是 Windows 系统。

自 Windows 3.0 推出两年多来已广泛应用于整个 PC 界。它以友善的用户图形界面、强大的内存管理、文件管理、任务管理以及丰富的应用程序而受到人们的喜爱。去年又推出了 Windows 3.1 版本，功能更加完善、运行更稳定、工作速度也更快。Windows 系统摆脱了传统的 PC 操作方式，采用鼠标器，进行全交互式操作，非常直观、简便，这一点特别适合于平面设计。我们可以在 Windows 窗口下进行任意的创意绘图、编辑文字以及方便地变形修改。Windows 采用 PostScript 或 TrueType 最新字型技术输出，可做到真正的“所见即所得”（What You See Is What You Get），显示和打印的字体多达 150 多种，而且字形非常美观、漂亮。

Windows 提供一组基本的操作，包括文件管理、任务管理、控制板管理、打印管理等等。Windows 自身带有一批基本软件，如 Write 文书软件、Paintbrush 绘图软件、以及记事本（Notepad）、计算器（Calculator）、卡片文件（Cardfile）、记录器（Recorder）等小工具软件。这都给用户的使用带来极大的方便。Windows 3.1 版还提供了多媒体功能，可以与话筒、录音机、录像机、CD 光碟机等设备联结，用来播放音乐、动画影片、卡拉 OK 等等。可利用听觉、视觉等多种方式制作各种妙趣横生的软件。这可以用来丰富广告作品的表现形式。现在越来越多的软件开发厂商，开发或移植了大量 Windows 版本应用软件，Windows 不仅在图形处理和文字处理方面独占鳌头，而且在数据处理和科学计算等领域也得到越来越多的应用。Windows 已经成为新一代软件标准，在全世界的用户已超过一千万。

四、CorelDRAW 绘图软件

目前 PC 界涌现出许多优秀的绘图软件，如 Ventura 公司的 Paintbrush、Aldus 公司的 FreeHands、Adobe 公司的 CorelDRAW 和 Micrografx 公司的 Designer 等等。虽然它们各有特点、各有长处，但我们认为 CorelDRAW 是其中一种功能最强大、使用又很方便的绘图软件，适合于平面绘图、广告设计等领域。1992 年推出的 CorelDRAW 3.0 最新版本功能更加完善，获得巨大成功，在美国和其它国家受到很大欢迎，国内也有越来越多的用户开始使用。

如果将 CorelDRAW 3.0 源盘全部解压安装在硬盘上，那么大约需要 33MB 左右的硬盘空间，会产生一组绘图软件，包括：

- CorelDRAW：主要的绘图软件、
- CorelCHART：绘制数据图表软件、
- Corel PHOTO-PAINT：彩色影像处理软件、
- CorelTRACE：将点阵图转换为向量图的转换软件、
- CorelMOSAIC：图形文件管理软件、

CorelSHOW：简报展示软件；

以上应用软件也可以选择安装，其中最基本、最常用的是 CorelDRAW 绘图软件，下面给予简单地介绍。CorelDRAW 绘图软件的功能很多，最主要的功能是选取工具箱提供的基本工具，利用鼠标器全交互式地绘制任意的直线、曲线、矩形、椭圆等基本图形，然后对基本图形进行平移、旋转、复制、放大/缩小、拉长/压扁、投影、倾斜等各种变化。还可以利用端点编辑功能，对图形中的任意一点或任意一段进行修改变形。图形中的每个端点有两个控制点，改变控制点的位置，可以控制端点附近图形的形状，而端点可以任意地移动、增加或删除。使用端点编辑功能可以方便快捷地画出各种复杂的图形，这是 CorelDRAW 的特色之一。图形画好之后，可以对图形填充任意的颜色（包括渐变的颜色）、预定的模型，或由用户自己设计的图形。也可以设定任意粗细、任意变化、任意颜色的边框。CorelDRAW 所支持的颜色系统为 24Bit 真彩色格式，颜色数最多可达 16, 777, 216 种，并分为多种配色系统，其中常用的包括：RGB（红、绿、蓝）、HLS（色彩、亮度、饱和度）、CMYK（青、品红、黄、黑，印刷业使用）、Name（按颜色名称选择、或自定义颜色名称）等等，也可以用鼠标在调色区直接选择颜色。强大的颜色处理功能是 CorelDRAW 的又一个特色，也是我们进行彩色广告设计的有力工具。

CorelDRAW 的另一个主要功能是文字处理。CorelDRAW 提供 150 多种 TrueType 精密字体，对所输入的文字，可以自动进行拼写检查。也可以直观地任意改变字距、行距、粗细、大小、长扁，可以象图形一样填充任意颜色、边框和其它各种变化。还具有分栏、分区、左齐、右齐、居中、左右都对齐等排版功能。可以将文字按任意路径摆放。

除以上主要功能以外，CorelDRAW 还可以对物件（图形或文字）做出类似透视的效果（允许在平面上有一个透视点或两个透视点）；从一个物件渐变到另一个任意物件的特殊效果；类似 3D 投影的特殊效果（可选择光源、深度、旋转等）；以及其它很多功能。

五、PageMaker 文字排版软件

由美国 Aldus 公司开发的 PageMaker 是目前在微机上运行最完善、功能最强大的页式排版软件之一。85 年首次在 Macintosh 上发行，后又移植到 IBM AT 及兼容机和 IBM PS/2 计算机上。PageMaker 因发动桌面印刷革命而闻名。桌面排版印刷系统是指采用个人计算机来生产高质量的正文和清晰的图形，并可以将正文和图形溶于同一页中，然后在高分辨率打印机或排字机上整页输出。目前广泛使用的是 PageMaker 4.0 版本。

当我们从 Windows 进入 PageMaker 后，首先要创建一个新的排版文件。设定页面的大小、正文的边空、文件的页数、是单面还是双面排版等基本页面参数。然后进入

页面排版编辑状态,可对任一页面进行文字或图形的输入,也可对全部页面进行统一处理。PageMaker引入了主页的概念。设置在主页上的正文图形等内容都将同时出现在文件的各个页面上。主页的使用为按统一格式编排书刊带来极大的方便。

PageMaker的文字输入有三种方式:第一种可以在页面上直接输入文字。第二种允许在另外一个编辑程序(如WordStar等)将文字打好,然后用Place命令灌入PageMaker页面中(又分为手动、半自动和自动三种正文流入方式)。第三种使用PageMaker的故事编辑(Edit Story)。这时进入一个故事编辑窗口,可以快速方便地输入正文,并有各种编辑、查找、替换以及检查拼写错误等功能。文字输入后,可以方便地设定或改变字体、尺寸、字距、行距、分栏数以及自然段的各种属性。文字块可作任意移动、旋转、缩放、复制等变化。对某一页的编辑修改,会自动引起有关页面的调整。

PageMaker也可以画任意的直线、曲线、矩形、椭圆等图形;可以为正文和图形加上任意的颜色;还可以接收扫描照片图片而产生的图像文件;从而生成图文并茂、色彩丰富的排版文件。然后在激光打印机或精密照排机上输出。PageMaker还有许多其它的功能,在此不作细述。

六、PhotoStyler图像处理软件

PhotoStyler是在PC及其兼容机上运行的Windows所支持的完整彩色图像编辑处理软件,具有与Macintosh机上的PhotoShop所类似的强大功能。它主要处理由扫描仪输入的图像,如:照片、幻灯片、图片、绘画、标志、文字等。作为Windows的应用程序,PhotoStyler的下拉菜单、功能图案、交互窗口、在线查询的直观界面为用户的使用提供了极大的方便。PhotoStyler能够输入多种格式的文件,也能够以多种文件格式进行转换和输出。能同时显示多幅图像,同时进行编辑处理。

PhotoStyler具有强大的彩色分解与合成功能,配色系统为24bit真彩色格式,包括RGB、HLS、CMYK等多种配色系统。在广告设计和印刷中常常用到的分色处理,都可以用PhotoStyler来完成。PhotoStyler可将一幅完整的彩色图像分解为CMYK四种颜色,输出为四色胶片,可直接拿到印刷厂印刷。对图像中的某一块颜色可进行任意调整编辑。如物体的背景、人物的衣服可换为其他颜色,如有不满意的地方还可以放大后,进行逐点编辑。

PhotoStyler可以用铅笔或喷笔绘图,可以在图像内或图像之间进行拷贝和粘贴,也可以对图像或选定的区域做几何变化,如倾斜、旋转、对称等。PhotoStyler还可以产生如平滑、模糊、钝化、锐化、涂污等多种变化,产生漩涡、浮雕、马赛克、波浪、球状和柱状变形等特殊效果。总之,PhotoStyler的图像编辑和调整功能应有尽有,任你发挥;各种特殊效果处理,可使平淡图像增添趣味,是平面广告设计的重要工具。

七、中文处理与尖端字库

以上设计软件功能强大、使用方便,可这些软件大都是美国公司开发出来的,直接处理的是英文或其它西方文字。再好的软件不能处理中文,它的使用就会受到很大的限制。我们原先用于设计的中文系统是华光IV排版系统(普及版),虽然它具有强大的中文排版功能,特别适合于国内一般书报的排版,输出的矢量汉字字形也比较漂亮,但它采用的是批处理方式,命令多、规则多,格式复杂,难于掌握和记忆。更重要的是它不能像Windows及其应用程序那样,采用鼠标直观地交互式地操作,图形和彩色处理功能也较弱。怎样才能既可以处理中文,又能利用西文软件的强大功能呢?人们通常采用汉化西文软件的办法,虽然这样可以在西文软件中直上中文,但它们大都是点阵字库,一般办公或数据处理还可以,但用于广告设计就不行了。国内外不少公司都在致力于这方面的研究。由美加一群华人电脑工程师组成的顶尖公司经过多年的研究发展,终于开发成功一套独创的结合东西方文化的中文字库——尖端中文字库。

尖端字库(TopFonts)采用首创的编码技术,采用最先进的动态字型,可以让所有在Windows下执行的西文软件(如CorelDRAW、PageMaker等)象使用西文一样使用中文,从而产生拉长、压扁、倾斜、空心、反白、旋转、立体、滚动、网纹、彩色等千变万化的文字创作。并可以在各种打印机上输出优美的PostScript中文字型,真正做到了What You See Is What You Get! 尖端字库采用贝塞尔曲线来描述中文字的轮廓,比一般的矢量字库在品质上还要高出一筹。我们所购的尖端字库共有中文字型32种,包括22种繁体体和10种简体(以后还可以扩充),基本上已能满足我们广告设计的应用。尖端字库既可以接受台湾BIG-5内码,也可接受大陆GB 2312-80内码。关键都在于它有一个“桥(Bridge)”转换软件,使用尖端字库,首先需要在某一编辑程序下建立所要转换的文字文件。你可以选用你所习惯的任意的汉字输入法,如拼音、五笔字型、自然码或者是仓颉码等。如果是大陆内码,在文件头部加\G命令。然后用Bridge软件进行转换,转换后的字型文件命名为Top_fn0.wri。接着就可以进入Windows应用软件,将Top_fn0.wri引入。在CorelDRAW中,用Import命令;在PageMaker中,用Place命令。文件引入后,屏幕上即出现中文,以后就可以象处理英文一样进行任意的排版、变形了。通过我们几个月的实践表明,尖端字库确实较好地解决了西文软件不能处理精密汉字的问题,输出的中文轮廓字型美观、漂亮。不愧是“连接东西方文化的桥梁”,成为我们广告设计中的不可缺少的工具。

目前计算机事业正在飞速发展,新技术、新产品不断涌现。我们在电脑平面设计中的应用还只是刚刚起步,我们愿意同电脑业、广告业各位同行一起,为我国电脑平面设计事业的发展而共同努力。

高校学籍管理系统的设计

广东行政学院 严婉庄

一、引言

学籍管理是根据国家对高校学生德、智、体全面发展的要求, 制定规章、制度、办法并按一定的程序对高校学生从入学到毕业在校期间的学籍取得与更移、教学管理制度、校风校纪三方面实施的管理。这种管理, 起着维护高校的教学秩序, 促进提高教学质量, 保证培养目标实现的重要作用。

当前许多院校在管理学籍时, 仍以传统的手工操作, 工作量大, 就以学籍管理中成绩登记这一项为例: 如一个大学专科学院在校学生有 1000 人, 按在学两年开设二十三门课程计算, 从学籍卡、原始登分册、成绩一览表、学生成绩通知单到一式两份的毕业成绩表, 共要重复用手工抄 6 次, 合计登分 138000 个, 工作量很大, 而管理学籍远不止一个项目, 所以用传统手工操作的方法来管理学籍, 就显得落后了。

二、系统功能与设计

学籍管理的内容包括: (1) 入学与学籍; (2) 成绩考核; (3) 升级、留级、转学、退学和休学; (4) 纪律、考勤; (5) 奖励和处分; (6) 毕业、结业和肄业。

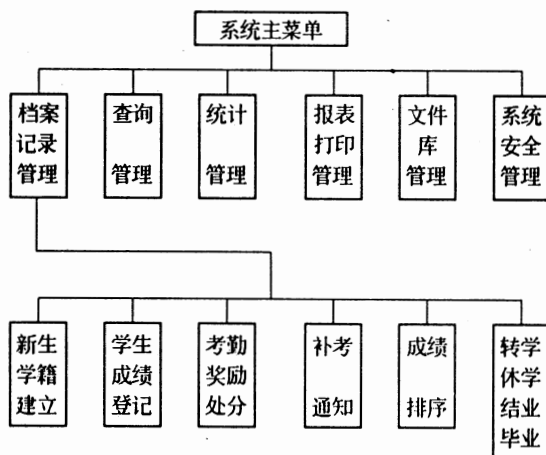
根据学籍管理的内容该系统的主要功能有:

- 1、档案登记 新生的学籍建立。包括个人资料, 所修课程等。
- 2、档案维护 对档案登记内容进行修改、删除, 如登记成绩、考勤、奖惩、升留级等等。
- 3、档案查询 有较强的查询功能、可随意组合查询条件、查出满足条件的学生或项目, 并打印出来, 也可进行分类查询、查同一专业、同一年级、同一成绩的学生情况等等。
- 4、统计汇总 学生的年龄、各年级各专业的成绩、考勤、奖惩、升留级、转退休学、毕业结业等情况的统计并有各百分比、直方图等显示和打印。
- 5、报表输出 每学期均有各学生成绩单, 补考通知等, 并可输出不同报表多种。
- 6、文件库 建立上级管理部门的有关学籍管理

文件库, 建立索引, 方便登记、查询和修改工作, 处理问题时能方便快捷找出根据。

系统设计

系统模块见图



三、系统特点

1、系统设计采用模块化、树形结构、可改性强、维护性好、屏幕格式、操作方法、代码方案、数据结构都采用统一的标准具有一致的风格。

2、人机界面友好, 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, 在所有运行过程中, 系统的运行状态、出错信息、操作提示等都随时提供给用户, 系统透明度好。

3、具有良好的数据安全性和信息保密性, 各模块各功能的使用权限均可通过‘系统安全管理模块’预先设置, 实现各类信息的保密和数据安全。



MIS 开发方法的新设想——系统界面法

广州市流花宾馆 吕廷振

【摘要】本文通过宾馆酒店开发 MIS 的实践，提出系统界面法的设想及一些具体的设计的方法，我们在 MIS 开发中，深切地体会到系统与用户之间的协调始终是当前开发工作的难点。经过几年的探索，我们提出系统界面法作为提高 MIS 开发方法。

一、系统界面方法的提出

信息系统是由人、设备、程序和数据所构成的统一体，能对信息进行收集、处理、存储、管理、检索和传输，能向有关人员提供有用信息的一种系统，信息系统本身是一个独立的系统，但同时它又是其所属社会经济组织的一个子系统，在企业内部，它从属管理系统是为管理和控制工作服务的。有些信息系统开发后并不能很好地应用，主要原因在于系统与用户的界面没有处理好，系统开发研制人员与用户之间关系没有协调好，很多人认为 MIS 开发应以计算机专业人员为主，忘记或忽视了 MIS 的真正使用者——企业各级管理人员的积极参与和支持。虽然强调了开发系统方法的本身和规范化即忽略了用户及具体软件使用人员的意见，从实际开发情况看，用户对管理及本职工作非常精通，但他们最大的困难是不了解计算机能为他们干什么，常常会定出过高的目标或过低的要求。例如，有些管理人员要求计算机专业人员，研制的软件管理功能十分齐全，什么都能做根本不用人参与，日常什么工作也不用做或做甚少工作，到了月尾，管理人员所需要报表统统都有，或者只要计算机打印一些简单报表等等。而一方面系统研制人员对计算机和软件方法精通，而对具体业务不懂，对管理过程或管理问题缺乏鉴别力，他们往往用计算机的数据运行速度，程序的结构，软件水平等衡量他们的成功，同时也由于调查分析不充分对企业内部业务过程理解上的偏差。由于上述的种种原因，导致新系统不能满足用户要求使系统的性能不能长期适应企业要求。因此，我们需要探讨一个新的开发方法，使得不同专业的人员为了一个共同目标进行合作，在他们之间建立一个稳定的环境，确定相对稳定的系统界面。这就是提出系统界面法的指导思想。所以，在开发初期我们就十分注意这些问题，我们曾有过这方面的教训，例如：我们在楼层结帐界面设计时，当初没有邀请各方面的有关人员参加，只要靠软件设计人员研制，及部分人员片面地追求数据运行速度，而没有注意本部门具体实际情

况，因此在设计结帐界面时，不分长包房或是旅行团亦或散房都一次性结账，这样做成一些该结未结，未结帐已结帐，如长包房也没有显账务应不得超过规定时间结帐，哪些帐务可以月底结等。又例如旅行团一次只能结一个房间帐务，一个旅行团一般都十几二十人住一个晚上，这时候结帐就会出现几十张帐单，服务员就感到十分麻烦，并没有显示出电脑的优越性，设计人员把这一辛辛苦苦研制出来的软件拿到楼层实际应用时，使用者十分不满意，要求重新设计出结合自己本部门有实际使用价值的软件；设计人员在接受了这次教训之后，深入调查，认真分析，邀请各方面有关人员、经理、部门主管、有代表性服务人员等人员与设计人员共同商榷共同参与研制。根据我馆地理环境和实际情况，例如入住我馆的人员大部分是散客，客人流量大，乘坐飞机火车人多等特点，进行重新的设计，使得更加结合实际使用情况，因此现时的结帐这界面可以根据客人的需要，灵活多变。可以一人结多项账，只结单项费用，提前结账，只结房租，房租分单，团体结帐，长包房还显示出那些超过五天未结帐房号，现时使用者感到使用十分方便，很满意。这样例子很多，不一一列举，从上面这些例子都说明界面设计是十分重要的。

二、系统界面方法的探讨

建立信息管理系统是为用户服务，以便提高用户的社会效益和经济效益。所以用户并不需要关心具体软件技术，只重视系统的功能，例如：楼层管理人员，只要知到你设计的功能，是否适合我的工作需求，是否给工作带来方便，工作效率是否有提高，差错是否减少了，是否给客人带来方便，经济效益是否提高了等，仓库管理人员，他们也是只知道你设计的功能是否适合本部门等等，信息系统界面法的用户一方面应该对业务和管理较为精通，并有一定的预见性，例如将来企业发展前途，企业的业务的发展，系统的扩展，计算机通讯网的

升级,软件的优化,这样,在确定界面时才不会停留在现行的业务性水平上。在信息系统设计时,必须首先考虑用户的要求,只有这样信息系统的设计才能更适应用户的需求。

1、定义系统界面法

用户可看到的管理信息功能及数据结构,它包括对信息进行收集、处理、存储、管理、检索和传输,能向有关人员提供有用信息的一种系统。如各种类型报表,日报表、营业报表、各种管理信息功能,有楼层客人管理信息功能,财务收支信息功能,仓库进出信息管理功能,客人查询、动态营业情况等等。用户和软件人员双方一起通力合作充分调查协商的基础上确定一个相对稳定的系统界面这个界面,是在共同充分理解和合作以及满足用户的基本要求,使其能达到实用性较强,而且功能具体,一旦界面确定之后,就成为双方各自的制约。这也是系统验证的标准。例如楼层管理界面就必须有相应的管理人员,服务员与开发人员共同订出楼层管理界面。

2、采用系统界面法的原则。

确定以用户为主的界面,管理信息系统的成功与否,是和用户分不开的,他们最了解实际情况,最了解本质的信息需求,例如,楼层管理人员最了解楼层的管理情况,电话计费系统话务员最清楚话费的业务情况等。用户要求通常包括功能要求,性能要求,可靠性要求,安全保密要求,以及防病毒要求有费用,开发周期等。因此系统界面应该是以用户为主导,在充分合作的基础上确定。由于计算机的使用与传统的工作方法有着很大差异,因此,流程应以系统开发人员设计,在征得用户接受的情况下确定。为用户提供一个全面流程的和友好界面。

3、系统界面法的主要内容

(1)、系统界面法是针对用户和系统来确定的,它的主要内容包括五方面:

系统功能界面:管理信息系统应用范围非常广泛,但不论哪个系统,大致上包括着系统初始、数据输入、加工计算、汇总统计、查询、输出以及系统维护等工作。因此,系统功能从一开始这些不同功能界面进行工作。例如前台管理功能界面,有预订分房、楼层服务(饮料、送餐、洗衣)、个人、团体、长包房等,各类客人的记帐、直拨电话及人工接的长途电话费及记帐。旅客建档及查询、餐厅、商场以及各类仓库管理。还有后台信息管理功能有收款处理。记帐凭证自动生成,帐务处理和固定资产折旧,财务收支状况等财务处理,商场仓库、食品仓库、五金仓库等各种仓库管理。各类统计报表、日报表、月报表、营业报表、经济分析与预测等。各个界面都必须十分协调好,稍有偏差就会使整个

系统不能正常运行,辛辛苦苦设计出来的系统软件就会没有人使用。所以,我们在进行界面设计时都十分注意,各方面管理人员及工作人员的合作,制定出各个合作的界面,由于改革不断深入,有些界面已不适应今天使用,所以有些界面又要进行改进和修改,如楼层管理和财务界面。

(2) 用户输入界面设计

输入设计是指软件内部和数据存贮设备之间的数据传递格式设计,也就是确定哪些数据以什么方式存贮在什么地方,后备的存贮、分配。由于我们三十多工作台同时做数据传输工作;往往造成通信量拥挤死锁成断网,从而使整个系统瘫痪而无法工作,大大影响工作效率。因此建立一个网络,首先应考虑线路长度也即路径问题,最短路径算法起着关键性的作用,它们既可以直接用确定在某种费用或距离的度量下最佳固定路径,由于它既能改变传递速度以及有关节省经济。确定性即固定路径选择方法根据网络放设,网络的设计用以安排连接网络节点的链路和链路容量,这是网络合成的关键阶段,其部分原因是由于路径选择,流量控制以及信息量和它的性能设计算法大都取决于所给定的网络敷设,敷设设计也具有若干性能和经济上的含义,节点位置,链路连接以及链路速度将直接经网络中转时间,敷设计算法,通常由两部分组成;第一部分是由初步的敷设,第二部分则反复加以改进,特别是这种算法开时只取一个可行的敷设结构。通常所提出的算法以满足所给的某种最佳函数,如网络时延链路费或链路容量,第三部分则力图将网络敷设最佳化,用早先提出的算法来检查及改进可行性,重复最佳步骤,直到不能进一步改善为止,这时的算法输出就是最终网络设计,这就是我们最后决定安装方案。我们还采用多个专用服务器,根据数据的存贮量而合理安排专用服务器的容量。

(3) 用户输出界面设计

所谓用户输出界面设计,是指用户所需要的信息。计算机采用什么方式,用什么格式供给用户例如酒店管理经营的各类日报表,月报表,经济预测数据待等,各种经济管理信息,如楼层客人管理信息,财务管理信息,仓库、电话计费等各种管理信息。我们借助输入输出来确定用的系统界面。一般地来说,事务管理系统需要处理的数据比较大。因此,占用存贮空间问题比较突出。对于数据存贮的策略好坏直接影响系统开发的成败。所以在使用专用服务器存贮数据一定要做到合理安排,把所有数据存贮都均匀地分配好各个专用服务器上,就能避免分配不均造成数据拥塞而导致服务器死锁不能工作,使企业造成不必要的经济损失。这一点,必须引起高度重视,确定系统数据,数据文件及数据库界面时,系统开发中应标准化,规范化,代码是信息系统

的又一个重要因素,如我酒店的客人查询,则可按客人姓名全称查,也可以一、三字查。也可按房号查,也可按到达的期查,还可按国籍查等。目的在于便于维护使用方便,同时也应该本着集中统一的原则,各自分工,归口负责,各方协同,改进完善原则,根据国家标准单位统一标编制数据字典,但内容同一般文献所述数据字典的内容有很大差别。使之更加规范化标准化,同时还应包括编码对照表,数据项定义,即包括每一项数据的名称、类别、宽度和注解,而数据库文件或数据库定义。这里也应包括名称、字段、内容、个数、长度、适用范围等、上述这些内容都是程序设计的重要依据,也是用户和系统之界重要的界面。

(4) 系统数学模型及分析方法界面

在经济管理实践中,常常要与一种称为非确定性的或称为随机的现象打交道。例如在同一条件下生产的产品,会由于生产过程中存在的无法控制的偶然因素而发生质量指标的非确定性波动,管理人员必须科学地分析这种波动,才能进而实施产品的质量的控制,管理人员必须根据市场的结果和其他有关统计资料来作出较为合理的估计,例如要进行某项决策时必须各种非确定性因素的影响作出定量分析。建立数据模型,一旦模型建立数据准备步骤完成之后,我们就可以进入模型求解步骤,分析人员在这个阶段要设法确定决策变量值,以提供模型的最佳输出,一个模型尚能把问题的主要有关方面包括了,并提供一个可取的解,那末,它对决策肯定很有帮助。该界面主要是帮助用户采用数学模型和分析方法,落实具体方法,达到投资的预期的经济效益。因此,如果数学模型本身不先进不仅难以达到上述的目的,反而会给投资者本身带来一定经济损失。在建立数学模型中应特别注意模型先进性,充分使用数据资源和发挥计算机作用,促进计算机这先进技术和管理中起着更大的作用。目前我们宾馆所进行经济分析与预测基本都是使目前比较流行方法如概率等。现在这一预测基本达到原定设计的要求。

(5)、确定计算机系统

计算机硬件是保证系统正常运行必要条件。因此,一开始就应结合本单位具体情况进行调查,然后决定采用哪种方式,无论是分布式或是集中式都各有优点,不论采用哪种方式其设备质量应该是上乘,并应配备一支技术精良硬件维护队伍。这些都将直接影响信息整体模式,当初我们曾在选用哪一种微机管理系统的问提做过大量的调查工作。到访过本市几间中外合资酒店。我们发现他们共同特点都是采用集中式处理的小型计算机系统。没有中文处理功能,有的只用于前台业务管理,有的前后台各采用两套独立且型号各异并需要人工处理才能实现数据交换系统。无论从当时整体角度,还是从将

来扩充的需要都不适合我酒店楼层服务台分布式微机管理。因此,我们又根据本单位实际情况选用了当时较先进的“以太”微机网络,并配置了具有较强汉字系统,餐厅也使用了具有通讯能力较强的收款机及程控电话电脑计费系统硬件设备构成的整个微机管理酒店系统,因此整个系统具有较为先进功能,通讯能力强使餐饮,话费房租费用等一系列费用都能在楼层结帐时反映出来,这样大大提高服务质量,减少差错,提高了经济效益,并能及时准确反映出营业管理信息。更好调动各种积极因素、提高了工作效率为宾馆全面现代化管理打下了良好基础。

4、实施系统界面的步骤

系统设计的过程了也就是由逻辑模型得出物理模型的过程。直到系统研制人员手中的系统说明书。规定了将来研制人员系统应该哪些处理功能,按怎样顺序去完成,要存贮哪些数据等。完整地规定了新的信息系统应该做些什么。所以系统界的设计是在用户与系统研制人员充分调查协商的基础设计出来的,在没有编程之前,充分论述系统界面的正确性,实用性由系统研制人员模拟系统应能完成的各项功能,征得用户基本满意后,才转入具体的软件实施,根据任务要求技术条件,列举出对于每一项具体任务可用技术手段及处理方法。从中比较优劣,权衡利弊,从全局出发,选定最后合适的方法总体,就是整个项目的实施方案,最后,依据双方制定的界面来验收。因此,我们在设计各个界时都应有相应的各方面管理人员参加。由于经过双方共同的努力,所以该系统提供了一系列的中文操作提示目录,简单明了易于操作。除汉字的数据输入外,其他操作(查询报表等)只需稍加讲解,便可进行。这样就为管理人员充分利用电脑进行科学管理提供了方便。另外,系统能准确、及时地提供经营管理信息,在没有电脑以前难以达到,例如,开房部当天的客房出租情况,客人入住情况,客房营业统计报表等,以往只能隔天提供,用电脑后,则可以随时查询,而且查到是动态的信息。同样楼层管理人员可在客房部管理功能中查看掌握所管辖的楼层客房经营情况和客人入住情况,必要时便可根据各楼层的接待情况。适当地调配人力、使楼层服务工作做得更好,用了电脑后,支厅,楼层、商场的每班营业情况可统计查看,总经理在自己办公室就可以查询当天空房情况,客房是报表,客房营业报表,客房情况,这样,管理人员便及时有效掌握宾馆动态经营状况。合理调动各方面,及时组织客员,使宾馆以最小的投入达到最佳的经济效益,其外还有动力部报修系统,设在各处工作站可以根据各个部门动力设备运行情况,通过电脑可以反映所有设备使用情况,使各种设备得到及时维护,从而提高房间租房率,也即提高经济效益。

高级 UNIX 连网技术讲座

广州昂立自动化工程公司 冯家宁

第三讲 NFS 和 RPC

3.1 引言

信息被认为是一个组织中最有价值的财产。信息以多种形式分布在组织中的各个部门,工作小组及个人的文件里。在自动化世界的今天,这通常意味着这些信息被大量不同的计算机系统所处理和储存。局部网(LAN)被认为是把这些不同的系统连接在一起的一种方式。不幸的是,一个 LAN 的共享信息是通过与以纸张为基础的系统相同的方式——复制——来处理的。这导致了磁盘空间的浪费,数据完整性差和低信息量的内容。

解决问题的一条途径是分布式操作系统的建立。这要求所有系统在相同的操作系统上运行,在实践上这是不可行的。网络文件系统 Network File System NFS 是一个独立的网络服务系统,它被设计为操作系统的一部分并独立于机器的类型。

本讲讨论 NFS 和远程过程调用 Remote Procedure Call RPC,它们相关的协议及其 UNIX 的实现。我们假定大家对 UNIX 命令和 TCP/IP 连网是熟悉的。本讲的读者是设置、使用及维护计算机网络的用户。

3.2 历史背景

NFS 是由计算机厂商 Sun Microsystems 开发并于 1984 年底公布的。从那时起, NFS 及低一层的 RPC 被注册并实现运行在 110 个以上的计算机平台上。根据 SUN 的估计,有 100,000 台以上的计算机运行 NFS,使它成为应用中最普遍的连网机制之一。

NFS 使远程计算机系统的文件系统附加在用户自己的本地计算机上一样。文件系统是一组数据文件的组织,在 UNIX 操作系统之下,它呈现为目录结构的一部分。IBM PC 机的用户对一些稍微普及的,如 NOVELL 的 NETWARE 和 IBM 的 PC-NET 等网络产品是熟悉的。他们会问 NFS 究竟有什么特别。NFS 的设计在这些网络中有其独特之处。从一开始 NFS 就设计成独立于机器和操作系统,使一台给定的机器能为不同类型的机器提供文件。

NFS 首先在 SUN 的以 4.2BSD (Berkeley System Distribution) 为基础的操作系统 SunOS 上实现的。不久 NFS 就被移植到几个非 Sun 的环境下。第一个移植是运行在 4.2BSD 的 VAX 11/750 上。由于 Sun 操作系统也是以

4.2BSD 为基础的,整个移植用了约 2 人-星期。接着 NFS 很快又被移植到 11/750 上的 UNIX System V.2, IBM-PC 的 PC-DOS, Micro VAX II 的 DEC ULTRIX, Data General MV/4000 的 DG/UX。在 1986 年 2 月的一次会议上,16 个厂家的 5 个操作系统在以太网上演示了文件共享。

今天 NFS 运行在更多的操作系统和计算机上。由于其起源, NFS 始终保持与 UNIX 及未来操作系统的接近。这种接近是 NFS 在 UNIX 团体中如此普遍的原因。

3.3 概论

NFS 是在一个或多个网络上共享多个不同的计算机厂商的机器上的信息的一种机制。这种共享通过让用户存取一组称为文件系统的文件来完成。用户自己的文件系统可与一个或多个远程被 mount 的文件系统相组合,使在用户看来就象单独一组文件系统一样。对“安装” mount 的概念还要作些解释。mount 是一个使用本地磁盘的 UNIX 命令。NFS 扩展了 mount 命令使系统管理员及用户在他们的机器上能存取远程的文件与目录。

个人的计算机系统可以看作是一个客户或一个服务器或两者都是。在客户/服务器模式里,客户应用程序在本地机上运行,它向远程的服务器计算机提出请求。本地计算机叫做客户,远程的计算机叫做服务器。如果本地计算机配有硬盘使网上的用户能使用其资源,这样的系统就同时叫客户与服务器。

NFS 是独立于操作系统的。为达到这一设计目标, NFS 使用了远程过程调用机制,它叫做 RPC。RPC 在使用与性能上与本地的过程调用非常相象。利用 RPC,本地过程可以调用、合并远程服务、让服务器执行一些任务,读出其结果。RPC 使用另一组子程序,叫外部数据表示 eXternal Data Representation (XDR) 库,它与表示 CUP 结构的内部数据表示不同。XDR 是所有机器都能明白的“标准”数据表示。NFS 协议定义为一组远程过程调用,有相应的参量,结果及效果。NFS 网络服务可以被认为是一组能存取远程文件的远程过程库。

NFS 设计的另一个特点是无状态运行。NFS 的过程调用包含了完成调用所需的全部信息。服务器并不跟踪所有的请求。这使得意外的恢复变得非常容易,因为在事发

时无论是服务器还是客户都无需执行恢复过程。通过 RPC 机制, NFS 是独立于通信系统的。大多数 NFS 的实现都用 IP 上的 DARPA 数据报协议 UDP。UDP 是快速地交换称为数据报的小量信息的非常低消费的协议。UDP 是不可靠传输协议, 但 NFS 的无状态特性使 UDP 变得非常有效。

3.4 硬件与软件要求

NFS 和 RPC/XDR 可在广泛的软硬件平台上实现。运行要求与具体的实现有关。Sun Microsystems 的 UNIX 实现是在基于 UDP/IP 的 LAN 或连在广域网上的多个 LAN 上的。它直接支持 Ethernet/IEEE-802.3 和 MAP/IEEE-802.4。只要有适当的 UNIX 通信驱动程序, 任何网络都能支持。

几乎任何硬件平台都可配置成一个 NFS 客户。无盘工作站是典型的一种配置。NFS, RPC 和 XDR 只要求 UNIX 操作系统之上的非常小的一点内存。一个典型的 NFS 服务器是具有大容量磁盘的高速机器。根据客户数量的不同服务器可以从录入级工作站到大型机间的任何机器。

本讲座的重点是 UNIX 连网, 但 NFS 和 RPC 在很多操作系统上都有。至今, NFS 已经被移植到 4.2BSD, 4.3BSD, XENIX System V Release 2 和 3, ULTRIX, VAX/VMS, UNICOS (Cray), UTS(Amdahl), AIX(IBM), VM(IBM), MS-DOS 等其它系统上。几乎任何接受文件概念的操作系统都能接受 NFS。

3.5 协议、层次与错误检测

3.5.1 OSI 模型

国际标准化组织(ISO)是主管国际标准的组织。在通信领域, ISO 建立了一个标准的通信模式, 叫开放性系统互连或 OSI 模型。OSI 模型由七层组成。每一层可由一种类型的协议和寻址方式组成。模型的目的是规定通信产品开发的结构, 产品的组成。OSI 模型的每一层都为上一层服务。这七层及其功能见图 3-1。

OSI 协议层

应用层
表示层
会话层
传输层
网络层
数据链路层
物理层

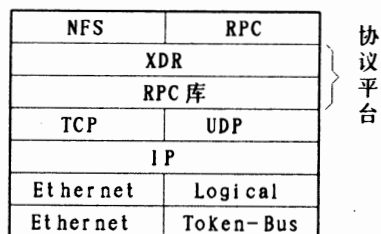


图 3-1 OSI 七层协议

在讨论 NFS 和 RPC 时, 我们集中在最顶两个应用层和表示层上。但先看看 NFS 和 RPC 在 UNIX 运行环境

下的各层结构也是有用的。特别的物理层有 Ethernet (IEEE-802.3) 和 Token-Bus (IEEE-802.4)。数据链路层协议有 EthernetLink 或 IEEE-802.2 Logical Link Control。网络协议是 Internet 协议(IP)。NFS 用的传输层协议是 UDP。NFS 不需要会话层。在表示层, NFS 用外部数据表示(XDR)协议编码。在应用层里 NFS 又分出两个子层 NFS 协议和 RPC 协议。

3.5.2 虚拟文件系统

为实现 NFS, 我们需要为现有的 UNIX 文件系统增加网络文件系统的支持。目的是使 UNIX 文件和网络文件对应用程序看来都一样。考虑到一些不一定支持的文件系统, 情况会变得更加复杂。在 Sun 4.2/4.3BSD 的 SunOS, 解决办法是把文件系统接口和文件系统本身分开。Sun 在实现 NFS 时定义的新的文件系统接口叫虚拟文件系统/虚拟文件结接口 Virtual File System/Virtual File Node (VFS/VNODE)。

VFS/VNODE 结构显示在图 3-2。从图 3-3 已可看到, 旧的 UNIX INODE 接口已被 VNODE 接口代替。INODE 仍旧使用, 但只在 UNIX 文件系统 VFS 场合下用。内核里的文件系统接口保持在较高的一级。

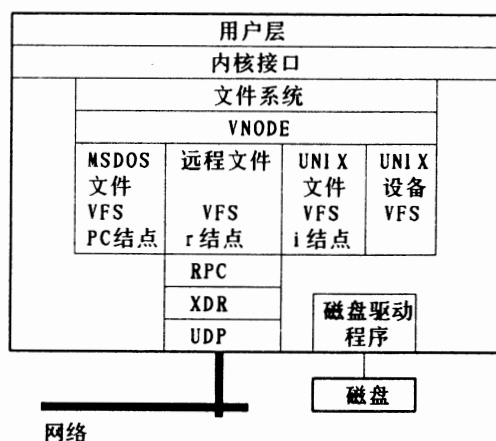


图 3-2. VFS/VNODE 结构

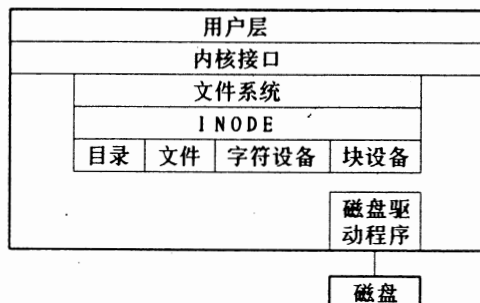


图 3-3. UNIX INODE 接口

从图 3-2 和 3-3 也可看出,正如有一个 UNIX 文件系统,也有一个 NFS 文件系统。与 INODE 不同, NFS 文件系统管理 RNODE,但概念是相似的(见图 3-4)。在下一节我们将描述 NFS 文件系统的细节。

属性
VNODE
文件句柄
标志
大小
长度验证
下一指针

图 3-4. RNODE

VFS/VNODE 接口使在 SunOS 内核里建立统一的文件系统。这种创新的风格使支持多文件系统类型的目的成为可能。更进一步,它使网文件与 UNIX 文件对应用程序来说都一样。

所有管理文件和文件系统的系统调用都修改成一个抽象统一的文件系统接口。这些系统调用现在管理 VFS 和 VNODE。VFS 数据结构在文件系统安装时生成。VFS 数据结构如图 3-5 所示。一个 VNODE 代表一个打开的文件或一个目录。VNODE 数据结构用图 3-6 来表示。每一与给定文件系统相联系的 VNODE 都包括在该文件系统的 VFS 连表里。

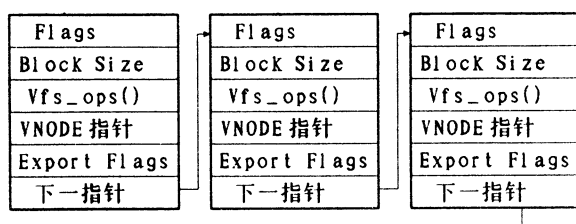


图 3-5. VFS 数据结构

标志
类型
参考计数
VFS ptr (VFS VNODE 所在)
VFS prt (VFS 安装所在)
VN-OPS() ptr
Private Data Ptr

图 3-6. VNODE 数据结构

对数据的实际操作,由文件系统里的适当过程来完成。VFS 和 VNODE 数据结构提供抽象文件系统与存取数据的实文件系统间的联系。在安装过程中,VFS 数据结构以及标识安装点的 VNODE 被建立。一个指向已安装的 VNODE 的 VFS 指针被放入 VFS 数据结构里。由于安装操作指定了文件系统的类型,内核把建立 VFS 结构和已安

装 VNODE 数据结构的工作交给适当的文件系统。在 VFS 结构和 VNODE 结构内有一指向文件系统里一组过程的指针,这些过程实现 VFS 或 VNODE 管理操作的标准集。当要进行对 VFS 或 VNODE 的操作时,内核把操作重定向到适当文件系统(UNIX, NFS, 等)来完成。

VNODE 与实文件系统适当数据结构间的连接是 VNODE 的一个指针。这指针叫私有数据指针,由适当的文件系统建立。在 Berkeley UNIX 文件系统里,指针均向一个 INODE 数据结构。在 NFS 里,指针指向 RNODE 数据结构。你现在可以看到抽象文件系统与实文件系统间是如何联系的。

其它操作系统包括了与 VFS/VNODE 相似的功能。一个有趣的例子是包括在 AT&T UNIX SYSTEM V 里的文件系统交换 File System Switch (FSS)。从功能的观点来看,FSS 和 Sun 的 VNODE 有所不同。但由于许多其它基于 UNIX 的操作系统已实现 VNODE 且 VNODE 出现时 NFS 是主流,Sun 和 AT&T 已同意把 VNODE 的接口包括在下一版的 System V 里,就是 System V Release 4。IBM AIX 里包括 VNODE 的接口是没有价值的。

3.5.3 旧的 UNIX 文件系统接口

旧的 4.2BDS 文件系统包括有让应用程序通过内核系统调用操纵文件系统目标的接口,该接口叫 INODE,其目标有目录,文件,字符设备和块设备。虽然有这些功能,但此接口在没有重新连接一个应用程序时,不许附加其它文件系统的目标。

VNODE 和 VFS

NFS 是这样工作的:通过让机器在网上共享信息,NFS 执行两项重要的工作——移出文件系统和安装文件系统。移出 Exporting 是一过程,它让本地机通知网上的其它机器要安装的文件系统。当服务器起动时,shell 文件 rc.local 运行 exports 命令。exports 命令会读 /etc/exports 文件并通知 UNIX 内核,服务器愿意提供哪个目录作共享。指定了什么使用权限。安装后台程序(rpc.mountd)和几个 NFS 后台程序(nfsd)也由同样的 rc.local 的 shell 文本起动。这些后台程序在起动后便挂起并等待请求。

当客户起动时, mount 程序读 /etc/fstab 文件。mount 程序依次请求适当的远程服务器或服务器提供客户在 /etc/fstab 中指定的目录以使用。服务器上的安装后台程序接受请求并决定客户所请求的文件系统是否有效。如果有效,安装后台程序就返回一个文件句柄给客户的系统内核。文件句柄是客户用于标识服务器文件系统的一块数据。文件句柄对客户来说是含糊的,使用它的客户并不需要解释它的内容。它只有服务器自己才明白(见图 3-7)。

一旦客户收到文件句柄,客户系统内核通过统一文件

系统接口传递 mount 请求。由于要安装的文件系统被看作是一个 NFS 文件, 客户上的文件系统建立一个 VFS 数据结构和已安装 VNODE 数据结构并提供一个指向 RNODE 的指针。图 3-4 解释了 RNODE 数据结构。VNODE 建立后, NFS 文件系统建立的 RNODE 会用到由安装操作提供的文件句柄。

服务器方:

文件系统#	文件#	文件事件#
-------	-----	-------

客户方:



图 3-7. 文件句柄

在服务器上的数据被客户使用的过程中, 服务器的 NFS 后台程序会查对文件句柄以确定数据实际储存在哪。在正常运行时, NFS 后台程序将继续检查客户对数据的使用权。以下各节将更详细地讨论 NFS 的基本结构。

NFS 虚拟文件系统: NFS 文件系统是管理远程文件和目录的一组操作。这些操作是在 RNODE 上执行的。RNODE 是包含有文件和目录状态的数据结构。通过 VFS 和 VNODE 介面, 创建文件系统的操作传递到下层, NFS 文件系统被调用执行这些操作。由于实际的文件和目录并不存在于本地主机上, NFS 文件系统必须提供必要的机制执行这些操作。这一机制就是 NFS 协议。

NFS 协议是一组用 RPC 机制联络远程主机上适当的 NFS 服务器的过程, 它能提供文件和目录的使用。由于操作系统内所含的实际文件和目录客户上的不同, 我们需要一种表示文件和目录的抽象方式。文件句柄就是这种表示。

对于客户, 文件句柄是一块含糊的数据。客户并不明白文件句柄的结构。客户只知道通过传递文件句柄给远程系统来管理文件和目录。远程计算机知道句柄的结构并依次用它执行请求的操作。

现在可以看出从抽象文件系统到具体文件系统的转变。也可以看出文件句柄是一个让客户保持文件系统结构统一性的强有力的结构。

3.5.4 安装协议

安装过程是把一操作系统指定路径名转换为文件句柄的方法。某些操作系统并不为 UNIX 目录式文件系统组织提供有用的变换。这样安装过程就要定义为与 NFS 本身分开的协议。大多数的做法用 UNIX 的安装协议, 它把

UNIX 路径名转变为 NFS 文件句柄。如果支持一个 NFS 服务器的操作系统能提供合理地转换 UNIX 目录路径名为某种本地文件系统组织的方法, 那么 UNIX 的安装协议就能支持。不然使用服务器就得用不同的 mount 命令。这并不影响 NFS 的透明性, 因为 NFS 只处理文件句柄而不是路径名。

UNIX 的 mount 命令已经扩展到允许安装远程文件系统。这一扩展了的 mount 命令让用户指定文件系统的类型、要安装的文件系统名和远程主机名。mount 命令通过调用 vfs_mount() 过程使用统一的文件系统介面。如果文件系统的类型是 NFS 类, 那么 vfs_mount() 将调用 nfs_mount() 过程。nfs_mount() 过程用一远过程调用 mount 命令所指定的远程主机上的安装后台程序。

安装后台程序为被安装的远程文件系统返回一个文件句柄, 它依次被 nfs_mount() 过程使用以得到文件系统的属性并为正在安装的文件系统建立 VFS、VNODE 和 RNODE 结构。VNODE 是文件的抽象视图并指向包含更具体文件视图的 RNODE。在下一节里有关于实际使用 VNODE 和 RNODE 的更多信息。

安装服务器后台程序是基于 RPC 的服务, 在起动时它注册于远程主机的 RPC 服务器并等待 mount 请求。一旦有标有 RPC 服务器的请求, RPC 服务器调用安装后台程序里的一个正确进程。后台程序通过 /etc/hosts 文件让 hostid 与 hostname 配对并检查 /etc/exports 文件以决定请求的主机是否有权安装所请求的文件系统。如果是, 安装后台程序做一系统调用 getfh() 以得到文件和目录的文件句柄并返回文件句柄给本地主机。

3.5.5 文件的使用

为安装顶层目录结构的文件句柄提供了 mount 操作的一部分。客户可用它通过服务器的目录结构, 管理目录结构并使用文件结构内的文件。

文件的管理和存取通过一组含有 NFS 协议的过程来实现。许多这些过程用 VNODE 过程一一对应, 但在某些情况下 VNODE 过程用几个 NFS 过程来实现。组成协议的过程组如下:

- 由所有基于 RPC 的服务提供用于支持调试。
- getattr() 为文件句柄所定的文件返回文件属性。
- setattr() 用于设置远程系统上的文件属性。
- lookup() 以文件名和目录作为参量, 返回它们的文件句柄和属性。
- readlink() 按照文件句柄所指的符号链读数据。
- read() 从远程文件读数据。
- write() 向远程文件写数据。
- create() 在远程主机上建立文件。
- remove() 在远程主机上删除文件。

- rename() 在远程主机上改文件名。这必须在不违反 NFS 协议的无状态性质方式下进行。
- link() 在远程主机上建立一个文件的连接。
- symlink() 在远程主机上建立一个文件的符号连接。
- mkdir() 在远程主机上建立目录。
- rmdir() 在远程主机上生成目录。
- readdir() 读远程主机上的目录内容。
- statfs() 以文件作参数(文件句柄), 给出所在文件系统的属性。

构成 NFS 协议的过程的实际规范可从 Sun Microsystems 公司或 USENET 公共网上得到。其中包括有用 RPC 语言定义的过程, 过程所用的数据结构用 XDR 格式定义。RPC 语言是用于输入到 Sun RPC 协议编译实用程序的类 C 型语言。通过实用程序可以输出实际的 C 语言, 其中包括实现协议的必要编码。XDR 独立于机器的数据表示方式, 它用于描述数据结构和给数据编码。

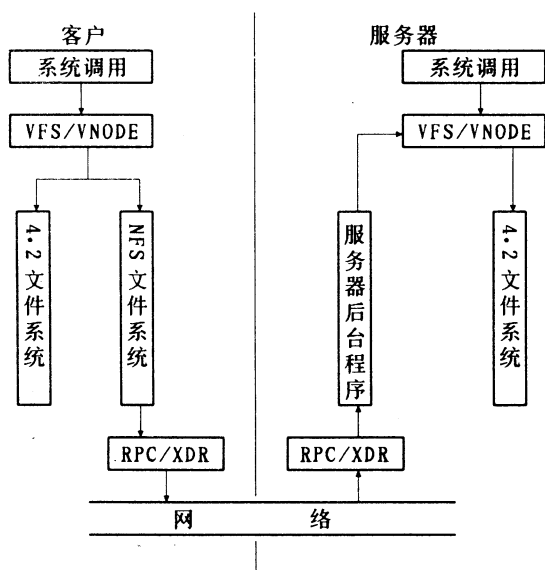


图 3-8. 在 4.2BSD 上使用文件

除了产生远过程调用的机制外, 现在应该明白获得远程文件使用的过程。一个在远程主机的 4.2BSD 文件系统上使用文件的例子见图 3-8。本地机或客户的应用程序的系统调用作用于统一文件系统层。为处理调用, UNIX 内核检查 VFS/VNODE 层以决定文件系统过程的位置。由于文件是远程的, VNODE 指针置为标识 NFS 文件系统。内核被定向到过程请求的 NFS 文件系统内的指定过程。

在此情况下, 过程实际上是调用位于服务器主机上的远程过程。通过 RPC 机制, 客户 NFS 文件系统请求服务器 NFS 文件系统执行所要的功能。这些请求被远程主机上的 NFS 后台程序接收。NFS 后台程序用 VFS/VNODE 接口处理请求以找到适当的本地文件系统来取得数据。在本例中, 适当的文件系统是服务器上的基于 4.2BSD 的文件系统。远程主机上的 4.2BSD 文件系统得到请求数据并回答 NFS 后台程序, 它又依次把结果通过 RPC 机制回送给本地主机。

从应用的角度看来, 使用本地磁盘的文件与远程机器上的文件并没有什么不同。NFS 协议保留了大部分使用 UNIX 文件系统的符号以保持高度的透明。使用 UNIX 文件符号上的例外 Sandberg(1986) 作过详细的描述, 我们在下一节作解释。以后的各节将讨论 RPC 及 NFS 是如何使用 RPC 的。

3.5.6 UNIX 文件使用的符号

在实现 NFS 时, 设计者们尽力保持 UNIX 文件使用的符号无需对服务器或协议作改动。UNIX 可以把打开的文件删除, 让程序隐藏临时文件。NFS 把删除打开文件的动作转换为对文件改名而把文件关闭后的删除工作留给客户内核。把当前用户有效地锁住, 在文件打开后改变其存储权限也是可能的。由于本地文件系统只检查打开权限。在 NFS 情况下, 由于安全原因每一次 NFS 调用的存取权限都要受到检查。于是在一次打开操作时, NFS 保存客户的证件并用它检查以后的客户存取。出于防护某些交互作用是不可能的, 因为无法协调两个不同的用户存取服务器上的同一个文件。例如, 当一个客户有一个文件打开, 第二个客户可以把它删除, 这使第一个客户的所有请求都将失败。

3.5.7 错误检测

不涉及 NFS 协议的无状态特性就无法讨论 NFS 的错误检测。说 NFS 是无状态的意味者每一操作都是彼此完全独立的。这里是一个具体的例子, 每一个 NFS 的写操作被看作分开的事件。当进行一项 NFS 操作时, 如果网上的服务器坏了, 客户就一直等待直到服务器恢复正常。一旦服务器恢复, 客户就从原来断开的地方开始继续运行。换句话说, 无需复杂的恢复机制。

为满足这种操作 (以及让 NFS 在更多的网上运行), NFS 只要求底层通信传输机制有数据报服务。传输层选用 UDP。但是 UDP 并不为块的序列化提供一个机制。这样 NFS 就得结合这些功能。所以虽然 NFS 不是传统意义上的一个传输系统, 但它结合了传输机制的许多错误检测特性以使它能在广泛的实际网络上运行。

(作者地址: 东山金城宾馆曙前楼 511 室
电话: 7754888-3511)

如何提高XENIX系统的安全性

浙江省台州地区人民银行 林荣庆

XENIX 系统已普遍应用, 它的安全已成为广大用户非常关心的问题, 随着版本的不断更新, XENIX 系统已形成一系列比较完整的完全保护机制, 如口令保护、文件权限和加密命令等。用户应该熟悉这些保护机制, 并在实际应用中不断加强和完善安全措施, 笔者就如何提高 XENIX 系统的安全性提出具体的实施方法

一、设置口令保护。

口令保护是 XENIX 系统最基本的保护机制, 在建立用户目录时把预先设置的口令以加密形式保存在 `/etc/passwd` 口令文件中, 除该用户自己外, 其他人无法知道其口令明文。用户注册时先打入有效的用户名, 再输入口令, 若输入正确, 则进入系统, 否则拒绝用户登录, 从而达到保护目的。因此用户设置的口令既要便于记忆, 又不能过于简单, 避免以用户自己的姓名的缩写或工程项目的代号作为口令, 以防被非法用户猜中, 普通用户要经常更换自己的口令, 防止无关人员偶然知道。对于早已形同虚设、失去保护作用的口令, 系统管理员有责任帮助用户重新设置新的口令, 系统管理员要设置口令使用的期限和长度, 促使用户定期更换口令。

二、文件权限的设置。

文件保护也是 XENIX 系统最基本的保护机制, 口令保护可以确保别用户不能直接进入自己的目录, 但这并不保险, 因为其他用户可以通过 `CD` 命令进入你的目录, 也可以通过文件全路径名访问你的目录和文件, 只有恰当地设置文件和目录的权限, 才能有效地保护用户的文件和目录的安全。

XENIX 系统把使用文件的人分成三个等级, 即文件的创建者—文件的主人, 同组的人和其他人。三种人的权利等级降低的, 文件的主人权力最大, 由它决定下面二级应用权力。每一个等级的人对文件都有三种不同的存取权, 即阅读权 `r`, 写入权 `w` 和执行权 `x`。

RWX	RWX	RWX
主人	同组人	其他人

文件的存取权都由文件的主人决定, 用户可以通过 `chmod` 命令正确设置自己的文件和目录的存取权, 可以达到对文件和目录的有效保护。例如要取消同组人和其他人的读、写和执行权, 则除文件目录的主人外, 其他用户既不能读目录, 也不能删除、修改和建立文件,

也不能执行其中的文件。

在 XENIX 系统中, 设备都当作文件看待, 称为特别文件, 它们存于 `/dev` 目录下, 主要用于系统中设备之间的数据传输, 这些特别文件是系统安全的一个重要方面, 另外系统配置文件, 例如 `/etc/rc`、`/etc/ttys` 等, 都要特别加以保护, 不能允许普通用户修改。

三、用户的设置

XENIX 系统是多用户操作系统, 合理地设置用户可以保护系统的安全, 每个要用 XENIX 系统机器的人, 都要建立自己的用户目录, 避免几个用户使用一个用户目录, 就是系统管理员进行非系统维护的一般性操作, 如学习研究和开发应用程序等, 也不能以超级用户身份进行, 应该有普通用户的目录, 以普通用户身份进行操作。超级用户也称特权用户, 其权力是无限大的, 它们可以任意修改和删除任何人的文件, 对系统的管理和维护负有特殊的责任, 因此知道超级用户口令的人越少, 系统就越安全, 特别对系统刚入门的人不应该成为超级用户, 而知道根目录口令的人, 不是在必须维护系统时也不应该在根目录下工作。

为防止普通用户随意进入其他用户目录或对系统文件造成伤害, 可指定普通用户使用受限制的 Shell, 强制用户在有限范围内使用系统。也可以在普通用户的 `.Profile` 文件中设置安全的命令目录和运行环境, 如路径 `PATH` 等设置。

一个更安全有效的方法是使业务操作人员只能利用应用程序系统处理业务, 可由系统管理员编制专用的 Shell 程序, 代替标准的 Shell, 用户进入系统后立即执行专用的 Shell 程序, 程序运行结束后自动退出系统, 回到登录状态。同时把用户一些常用的命令, 例如软盘格式化、拷贝、列目录等命令也做在用户的专用 shell 程序里, 减少业务操作人员同操作系统的接触, 并且去掉用户某些不需要的操作权限, 使其在程序的严格控制下操作。

例如用户 `lrq` 专用 Shell 程序为 `gz.sh` 内容如下:

```
foxplus gz
exit
在 /etc/passwd 口令文件中, 用户 lrq 一行原是:
lrq::201:50::usr:lrq:/bin/sh 改成
lrq::201:50::usr:lrq:/usr/lrq/gz.sh
```

则用户 `lrq` 登录后直接进入 `FOXBASE` 运行 `gz.prg`

程序, 执行完毕后自动退到 login: 状态。也可以在用户的 .profile 文件中增加 foxplus gz 和 exit 二行, 达到同样的效果。

四、用户终端的设置。

利用 XENIX 系统适用于多用户终端设备的特点, 可以给各应用系统的用户分配固定的终端, 这样非法用户即使知道其他用户的口令和密码, 也无法在自己的终端上运行其他用户的应用程序。具体做法是在用户的 .profile 文件中增加以下几行程序。

```
if test 'tty' = /dev/ttyla
then echo "你不能使用本终端"
exit
fi
foxplus gz
exit
```

同时可在用户使用终端前由系统管理员用 enable 命令激活各终端, 关机前再由系统管理员用 disable 命令锁住各终端。

五、给 XENIX 系统加锁。

把 XENIX 系统的机器只限于系统管理员开启主机, 防止其他普通用户, 特别是非法用户随意开启机器, 可进一步提高系统的安全性。可以在进入多用户前设置口令检测, 即加锁, 此口令只由系统管理员掌握, 原理和做法如下。

/etc/rc 含有 XENIX 系统一系列初始化命令, 系统开机时就先执行这些命令, 它们显示启动信息, 启动各种系统进程, 装载文件系统, 然后进入多用户登录选择, 相当于 DOS 操作系统中的自动批处理文件, 对所有用户都是有效的。今在 /etc/rc 文件中设置口令检测, 如输入口令正确则允许进入系统, 即允许普通用户登录, 否则自动关机, 有效地防止其他普通用户随意进入 XENIX 系统。在 /etc/rc 文件的末尾加入下面一段 Shell 程序。

```
echo "请输入进 XENIX 系统的口令"
if test 'inpasswd.exe' != "XENIX"
then exec /etc/haltsys
fi
```

这里 "XENIX" 就是系统管理员预先设置的口令, 其中 inpasswd.exe 是 C 语言编制的执行程序, 源程序如下:

```
inpasswd.c
#include <curses.h>
main()
{
char c[10]
initscr();
noecho();
getstr(c);
echo();
puts(c);
endwin();
}
```

六、执行正常的关机步骤。

普通用户完成业务操作后, 不能一走了事, 必须从终端退出, 即在普通用户提示符 \$ 下按 Ctrl+D 键, 回到多用户登录状态, 显示 login:, 否则非法用户照样可以进入系统操作。使用 XENIX 系统必须正确关闭主机, 否则会造成文件系统混乱, 下次开机时就会出现如下信息:

```
The system was not shut down properly,
and the root file system should be cleaned.
proceed with cleaning (y/n)?
```

含义是系统没有正确关闭, 根文件系统需要清理, 问是否继续清理?

这时就要清理, 系统检查和恢复文件系统, 但有时也会造成不能恢复, 不得不进行系统维护, 不但费时费力, 而且可能危及数据文件, 因此正确关机显得尤其重要。

XENIX 系统提供了二条关机命令, 一条是 haltsys, 一条是 Shutdown, 但都是在超级用户下执行, 普通用户为了关机而经常进入超级用户, 对系统的安全极为不利。为使普通用户既不进入超级用户又能执行正常关机, 可以设置一个用户, 专门作为关机用户, 方法如下。

1、在超级用户下建立关机用户, 假设名为 gjyh, 它也有口令, 应让其他普通用户都知道。

2、修改 /etc/passwd 文件, 把关机用户的用户标识号和用户组标识号都改成同 root 超级用户一样。

3、在关机用户的 .profile 文件的末尾增加下面一命令 exec /etc/haltsys。

4、普通用户按 Ctrl+D, 在多用户方式下以关机用户注册, 当回答口令正确后, 则立即执行 .profile, 屏幕显示正常关机后的信息, 此时可关闭主机电源。

七、建立超级用户的后备口令

超级用户也是特权用户, 是在 XENIX 系统中享有最高权利的人, 文件保护机制对它不起任何作用, 因此知道超级用户口令的人不宜多, 且超级用户也要经常更换自己的口令, 这对系统的安全是非常必要的。超级用户权利越大意味着责任越大, 但是一旦超级用户忘记了自己的口令而无法进行系统维护时, 问题就严重, 因此有必要建立超级用户的后备口令, 方法有多种, 下面介绍一种方法。

先制作启动软盘, 把超级用户口令置成无口令状态, 也就是后备口令, 把此时的口令文件 /etc/passwd 复制到启动软盘上, 再在硬盘系统中设置超级用户所需要的口令。当超级用户忘记口令而无法进入系统维护时, 就用启动软盘启动系统, 当单用户方式以 root 登录要求输入口令时, 就打回车键 (即无口

令设置), 从而进入系统, 显示并, 再用 passwd 命令重新设置超级用户口令。

八、软盘启动和维护

多用户 XENIX 操作系统已广泛应用, 日常维护问题比较突出。未正常关机往往引起系统混乱, 超级用户忘记口令而无法登录, 及其他操作故障时会发生, 造成硬盘文件系统不能正常运行, 重新安装系统不但费时费力, 而且会丢失大量的操作系统以外的用户文件, 给工作带来严重损失, 显然对系统的安全是十分不利的。因此要解决利用软盘启动和维护问题, 下面介绍维护软盘制作和使用。

1、制作维护软盘

维护软盘要在硬盘文件系统正常的机器上制作, 准备一张 1.2M 新软盘, 进入硬盘系统的超级用户。

(1) 格式化软盘, 把软盘插在 0 号驱动器。

```
#format / dev / rfd096ds
```

(2) 构造带有引导和根文件系统软盘

```
#mkdev fd
```

根据屏幕显示进行如下选择, 选择文件系统的软盘类型为双面、每磁道 9 个扇区, 选择软盘文件系统内容为根文件及有引导。此时软盘上已产生部分系统文件且可引导。

(3) 把启动过程中用到的文件和常用命令复制到软盘。

```
#mount / dev / fd0 / mnt
#cp / dev / console / mnt / dev
#cd / bin
#cp lcat fsck / mnt / bin
#cd / etc
#cp passwd ttys getty rc group init inir login mount umount
haltsys
cp / mnt / etc
```

(4) 产生 .profile 文件。

```
#vi / mnt / .profile
PATH = / : / bin / etc
HZ = 50
export PATH HZ
#umount / dev / fd0
```

2、使用方法

在硬盘系统启动中主要用到下列文件:

boot, xenix, init, inir, ttys, getty, passwd, rc, group, login, console。

如上述文件有损坏, 硬盘系统不能引导, 或不能进入多用户或系统设置不符合要求等, 在这种情况下, 可用维护软盘启动, 进入软盘文件系统, 再把硬盘安装到软盘文件系统中, 然后根据故障现象, 把维护软盘上的有关文件复制到硬盘上, 卸下硬盘, 这样就能恢复硬盘文件系统。

以上所谈的提高 XENIX 系统安全性的方法都是行之有效的, 已在实际运用中取得了良好的效果。

324

UNIX 系统使用终端时可能出现的问题

湖南省冷水滩市零陵地区保险公司 王连军

UNIX 操作系统是一种运行小型机和超级微型机上的多用户、多任务操作系统, 现在使用 UNIX 操作系统的用户日渐增多。在 UNIX 环境下, 多个用户通过终端同时使用一套微机系统, 一个任务在运行时, 可以启动另一个任务同时运行, 运行效率高、速度快, 系统结构、保密性更趋完美。

笔者在最近安装 Unisgs6035 微机时, 使用 Unisgs 公司的 UNIXSVR4 操作系统安装完成后, 以普通用户注册运行 Vi 编辑程序, 输入汉字时屏幕上不显示汉字却是“10331297...”这样的数字代码, 退出 Vi 编辑光标停在第 25 行上, 运行 ls、cat 等命令所有显示均在 25 行, 屏幕不滚动。笔者使用的是台湾产 CLC-2400K 中西文终端, 而在 GW230 终端上无此现象。以前使用 XENIX 系统时也无此现象。遇到这种问题可能是因为终端设置不当, 应修改注册用户的 .profile 文件对终端方式进行设置, 解决的办法如下:

1、退出 Vi 程序后光标停在第 25 行屏幕不滚动:

解决办法: 修改普通用户的 .profile 文件, 加入下面两行:

```
TERM = Vt100
```

```
export TERM
```

即: 设置终端方式为 Vt100 类型。

2、Vi 编辑程序汉字不能正常显示

解决办法: 修改 .profile 文件, 加入下面两行:

```
LANG = china
```

```
export LANG
```

即: 设置终端为中文汉字显示方式。

修改完后退出用户重新注册, 故障消失。以上在 Unisgs6035 微机 UNIXSVR4.2.5-2 操作系统下运行通过。用户在使用过程中如果出现上述两个问题可参考以上方法解决, 而且只要将 .profile 文件修改一次, 不必重复设置。

325

本地硬盘的备份与恢复

成都热电厂 宋捷

在 NetWare 网上, 对于双工的文件服务器而言, 网络硬盘的安全性较本地硬盘的安全性高, 故我们可将本地硬盘上的数据备份到 NetWare 文件服务器上, 以便在本地硬盘发生故障时恢复这些数据, 网络管理员有义务通知所有工作站上的操作员备份本地硬盘上的一些要求安全性的数据文件, 但网络管理员要管理所有工作站上本地硬盘的备份与恢复是相当麻烦的, 为了方便地管理本地硬盘上数据的备份与恢复, 网络管理员可以编制一个菜单文件 MANAGE.MNU, 以便于工作站上的操作员完成他们各自的本地硬盘的备份与恢复。

NetWare 提供了一个 MENU 命令用于生成用户自己的菜单, 它类似于 SYSCON、SESSION 和 FILER 这些实用程序提供的菜单, 用户能用任何文本编辑程序来建立自己的菜单文件, 这些菜单文件可用 ASCII 格式来保存, 其中可以包括一个主菜单和各级子菜单, 通过将若干 NetWare 实用程序组合成一个菜单作业, 可以加速对常用命令序列的访问和执行, 由于 NetWare 的 MENU 命令放在 SYS:PUBLIC 目录下, 故菜单文件 MANAGE.MNU 最好放在 SYS:PUBLIC 下, MENU 将自动在 PUBLIC 目录下寻找菜单文件。

在建立菜单文件 MANAGE.MNU 时应注意第一行的菜单标题 %MANAGE MENU, 5, 20, 0 总是与左边缘齐平键入, 其中, 百分号表示菜单头, MANAGE 为菜单名, 后面的三个数字为定制菜单规范号, 分别表示垂直位置、水平位置及色彩三者的规范值, 垂直位置的计算公式为: $C = A + B / 2$, 其中 C 为菜单垂直位置, A 为屏顶到菜单项间要求的行数, B / 2 为菜单行数的二分之一, 水平位置的计算公式为: $F = D + E / 2$, 其中 F 为菜单水平位置, D 为菜单左边的列数, E / 2 为菜单列宽的二分之一, 可以建立 NetWare 的全彩色菜单系统, NetWare 有 5 个规定的缺省调色板, 编号为 0 至 4, NetWare 提供的缺省菜单将这些调色板用于如下目的: 调色板 0 用于清单, 菜单和正常文本; 调色板 1 用于屏幕的主标题和背景色; 调色板 2 用于帮助信息; 调色板 3 用于出错信息; 调色板 4 用于退出和报警入口, 菜单标题后的选项 1 至 9 也应与左边缘齐平键入, 其中, 菜单选项 1、2、3、4 可分别将 1 号至 4 号工作站的本地硬盘上的数据文件备份到文件服务器 SERVER, 菜单选项 5、6、7、8 可分别将文件服务器的网络硬盘上的数据恢复到 1 号至 4 号工作站上, 在备份与恢复时主要使用的是 NetWare 的 NCOPY 命令, 盘符 M: 是文件服务器上的数据卷被映射到的驱动器, 每个选择项执行的菜单命令应放在各选择行的下面, 且需缩进放置, 在菜单正本运行时, MENU 将自动按字母顺序排列选项清单, 但对编了号的菜单选项, 则应按次序进行, 在进行

备份与恢复的选项 1 至 8 中, 最后的选项 9 中的 LOGOUT 命令与感叹号合用表示将系统注销, 菜单文件建立好后, 还应在网络硬盘上建立与所有本地硬盘上的应用目录对应的子目录作备份用, 在使用菜单文件 MANAGE.MNU 时, 只要在网络提示符下键入 MENU MANAGE, 便可将菜单调入并进入菜单中的各个选项了, 按 Esc 键可退出菜单。

%Manage Menu,5,20,0

1. WORKSTATION1 -> SERVER
 NCOPY C:\CW*. * M:\CC\CW
 NCOPY C:\CW1*. * PPPM:\CC\CW1
 NCOPY C:\CW2*. * BBBM:\CC\CW2
2. WORKSTATION2 -> SERVER
 NCOPY C:\CW*. * M:\CH\CW
 NCOPY C:\CW1*. * PPPM:\CH\CW1
 NCOPY C:\CW2*. * BBBM:\CH\CW2
 NCOPY D:\CBM*. * YY\CB
 NCOPY D:\HNM*. * YY\HN
3. WORKSTATION3 -> SERVER
 NCOPY C:\YF*. * M:\YY\YFC
 NCOPY D:\YF*. * M:\YY\YFD
 NCOPY D:\RL*. * M:\YY\RL
 NCOPY D:\GD*. * M:\YY\GD
 NCOPY D:\CL*. * M:\YY\CL
 NCOPY D:\92*. * M:\YY\92
4. WORKSTATION4 -> SERVER
 NCOPY C:\MASTR*. * M:\MASTER
 NCOPY C:\NOVELL*. * F:\NOVELL
5. SERVER -> WORKSTATION1
 NCOPY M:\CC\CW*. * C:\CW
 NCOPY M:\CC\CW1*. * PPPC:\CW1
 NCOPY M:\CC\CW2*. * BBB C:\CW2
6. SERVER -> WORKSTATION2
 NCOPY M:\CH\CW*. * C:\CW
 NCOPY M:\CH\CW1*. * PPPC:\CW1
 NCOPY M:\CH\CW2*. * BBB C:\CW2
 NCOPY M:\YY\CB*. * D:\CB
 NCOPY M:\YY\HN*. * D:\HN
7. SERVER -> WORKSTATION3
 NCOPY M:\YY\YFC*. * C:\YF
 NCOPY M:\YY\YFD*. * D:\YF
 NCOPY M:\YY\RL*. * D:\RL
 NCOPY M:\YY\GD*. * D:\GD
 NCOPY M:\YY\CL*. * D:\CL
 NCOPY M:\YY\92*. * C:\92
8. SERVER -> WORKSTATION4
 NCOPY M:\MASTR*. * C:\MASTER
 NCOPY F:\NOVELL*. * C:\NOVELL
9. Logout
 !Logout

注: 完成以上菜单文件在 NOVELL NETWARE V3.11 系统下通过。

全国电子报刊联合征订简明目录表

订阅代号	报刊名称	刊期	开本页码	单价	全年订价	通讯地址	邮码
1-145	电子商报	周二	对开 4 版	0.20	20.80	北京石景山路 23 号	100043
1-28	计算机世界报	周报	4 开 160 版	0.60	30.00	北京市 750 信箱	100039
1-48	北京电子报	周报	4 开 8 版	0.20	10.44	北京朝阳区东三环北路 36 号	100026
1-49	国际电子报	周报	4 开 32 版	0.25	12.00	北京市 750 信箱	100039
2-354	电视技术	月刊	16 开 100 页	2.60	31.20	北京 743 信箱	100015
2-355	电声技术	月刊	16 开 48 页	1.60	19.20	北京 743 信箱	100015
2-675	电信技术	月刊	16 开 60 页	2.00	24.00	北京东长安街 27 号	100740
2-75	无线电	月刊	16 开 64 页	2.40	28.80	北京东长安街 27 号	100740
2-888	电子与电脑	月刊	16 开 60 页	2.40	28.80	北京万寿路 173 信箱	100036
2-889	电子技术应用	月刊	16 开 80 页	2.00	24.00	北京 927 信箱	100083
2-892	电子世界	月刊	16 开 40 页	1.60	19.20	北京市 165 信箱	100036
4-141	电子技术	月刊	16 开 48 页	2.20	26.40	上海 085-253 信箱	200009
4-236	中学科技	月刊	16 开 48 页	1.90	22.80	上海冠生园路 393 号	200233
4-316	现代通信	月刊	16 开 32 页	1.50	18.00	上海 085-253 信箱	200009
4-386	无线电与电视	双月刊	16 开 56 页	2.50	15.00	上海市瑞金二路 450 号	200020
4-489	实用无线电	双月刊	16 开 56 页	2.30	13.80	上海市冠生园路 393 号	200233
46-115	电脑	月刊	大 16 开 96 页	3.00	36.00	广州石牌华南师范大学内	510631
61-74	软件报	周报	4 开 4 版	0.24	12.48	成都市金河街 75 号	610015
61-75	电子报	周报	4 开 8 版	0.30	15.60	成都市金河街 75 号	610015
61-8	西部电子信息报	周报	4 开 4 版	0.18	9.36	成都市桂王桥西街 66 号	610017
61-87	电子文摘报	周报	四开四版	0.18	9.36	成都 118 信箱	610015
62-175	实用电子文摘	双月刊	16 开 128 页	4.00	24.00	成都市金河街 75 号	610015
62-189	家庭电子	月刊	16 开 32 页	1.40	16.80	成都 118 信箱	610015
66-32	机械与电子	双月刊	16 开 48 页	2.00	12.00	贵州贵阳市延安西路 67 号	550003
77-19	电脑报	周报	4 开 8 版	0.28	14.64	重庆市双钢路三号	630013
81-10	中国电子报	周四	对开四版	0.25	51.50	北京 4357 信箱	100043
82-141	国外电子测量技术	季刊	16 开 48 页	3.00	12.00	北京海淀区学院路 5 号	100083
82-339	计算机世界月刊	月刊	16 开 120 页	2.00	24.00	北京 750 信箱计算机世界月刊部	100039
82-340	家电维修	月刊	16 开 32 版	1.50	18.00	北京东四十三条 32 号	100007
82-417	微型机与应用	月刊	16 开 52 页	1.80	21.60	北京 927 信箱	100083
82-454	信息与电脑	双月刊	大 16 开 64 页	2.50	15.00	北京西直门南大街 16 号	100035
82-512	电脑爱好者	月刊	16 开 56 页	2.20	26.4	北京海淀区中关村南二街五号 102 室	100080
82-518	今日电子	月刊	大 16 开 96 页	6.80	81.60	北京海淀区车道沟一号滨河大厦九层	100081
82-541	电子制作	双月刊	16 开 48 页	1.80	10.80	北京东四十三条 32 号	100007
自办发行	电子天府	双月刊	16 开 128 页	5.00	30.00	成都桂王桥西街 66 号	610017
自办发行	电讯技术	双月刊	16 开	4.00	24.00	成都市外西茶店子东街 48 号	610036
自办发行	电子质量	月刊	大 16 开 48 页	2.68	32.16	广州 1501 信箱 9 分箱	510610
自办发行	广东电子	月刊	大 16 开 48 页	2.50	30.00	广州 1501 信箱 9 分箱	510610
自办发行	电子产品可靠性与环境试验	双月刊	16 开 72 页	3.00	18.00	广州 1501 信箱 9 分箱	510610
自办发行	音响世界	月刊	大 16 开 132 页	8.50	102.00	广州石牌五山路科技街 108 号	510630

NDD、DISKEDIT 修复软盘一例

江油市物资回收利用公司 潘 敏

一张 360K 软盘，系统区域严重损坏，根目录区和文件分配表均不可读取，用 DIR 命令显示“FILE NOT FOUND”，用 PCTOOLS 以磁盘服务和文件服务两种方式检视该盘均报告“Bad CRC on diskette read”，用户的 79 页试题分 6 个文件存于盘上，已全部无法调出。

用 NORTON 工具包的 DISKEDIT 浏览此盘，因磁盘受损严重，显然无法以逻辑盘方式打开。幸运的是功能强大的 DISKEDIT 为我们提供了以物理方式编辑磁盘的功能。经过浏览，发现该盘 ROOT 和 FAT 区域已杂乱无章，毫无修复的可能。考虑存在盘上的试题都编了页号，试图用 DISKEDIT 查找页号定位、读取磁盘物理扇区的方法恢复文件，但该文件编辑时多次存盘，磁盘中存在大量重复数据，用该法工作量过大、效率太低。转而考虑用磁盘医生 NDD 进行修复。

首先用 COPYWRITE 将原盘复制一张工作盘，以后操作便在工作盘上进行。运行 NDD 扫描磁盘，检查 FAT 和目录结构时报告非法，因磁盘根目录区信息丢失，NDD 未报告有文件，而在进行丢失簇检查时(Lost Clusters)，将文件内容全部作为丢失簇处理，报告“*There are 340 lost clusters in 339 chains*”，按其提示对 FAT 和目录结构进行修复，丢失的簇转化为 96 个 *._DD 文件，每个文件长度均为 1024 个字节，对其进行存储时，报告磁盘无足够的空间。退出 NDD，将软盘上生成的 96 个 *._DD 文件拷贝到硬盘后将软盘上所有的文件删除，再次运行 NDD，报告目录非法，继续进行丢失簇检查，报告“*There are 243 lost clusters in 242 chains*”，存储时仍报告磁盘空间不够，再将文件拷至硬盘。反复几次进行上述操作，直到不再出现错误信息。需要注意的是，每次操作生成的文件名有一部份是相同的，向硬盘拷贝时需拷至不同子目录下。

现在，磁盘上所有的文件内容都已存在于 *._DD 文件中，剩下的工作便是整理这些内容了。考虑这些文件是用 WPS 编辑而成，因 WPS 文件有其独特的结构，其文件头若被破坏，用 WPS 编辑器调出来的内容便会面目全非。用 PCTOOLS 查找包含 WPS 文件头信息的文件，特征字符串可输“Invalid parameter for WPS”或其子字符串。将找出的文件全部移走，用 DOS 命令“COPY *._DD SAVE.WPS”将文件合并，再用 WPS 编辑器编辑文件 SAVE.WPS，发现文件页码顺序有错乱，利用 WPS 的块写文件功能，将每一页定义成块，写入硬盘，文件名取其页号。对其它几个目录下的 *._DD 文件如法炮制。此时，原文的每一页均以文件的形式存在于磁盘。最后用 WPS 的读文件功能，按页号顺序读入每一个块文件。原文 79 页内容除前面 7 页外全部恢复，并已按页号顺序合并成一个大文件。

再用 DISKEDIT 搜寻原盘，又挽救出 3 页正文和一张封面，剩下 4 页因数据被破坏已无法找出。至此，原盘上的内容基本恢复，且原文设置的各种控制码均未改变，修复工作大功告成。

需要注意的是，这种方法只适用于修复文本数据，对于程序文件是不适用的。抛砖能引玉，本文旨在为读者提供参考，在操作时，应根据修复内容灵活变通，不可生搬。

328

更 正

本刊 93 年第六期《散手四招》一文中有误。第 48 页右栏倒数第 6 行中“在 PCTOOLS 7.0 工具软件包中……”应为“在 NORTON 5.0 和 NORTON 6.0 工具软件包中……”，特此更正。

东海0520C

软驱维修一例

青海省工商银行西宁市支行 杨海宁

故障现象:

一台 DH0520C 型机软驱 1.2M (型号 DDF5-27C, 与日本 TEAC 公司的 FD-55B 结构和电路基本相同)。读软盘正常, 往软盘拷贝文件和格式化软盘时屏幕出现:

Write protect error writing drive A

Abort Retry Ignore?

FORMAT A: 时屏幕出现

Attempted write-protect violation

format failure

.....

根据以上出错信息, 分析认为是写保护检测电路工作不正常所致。

这有三种情况:

- 1、光电管被灰尘堵住或位置不到位。
- 2、写保护检测电路元器件损坏造成光电管电压不正常。
- 3、写保护发光管或接受管本身已坏。

顺着以上思路, 逐一检查。打开主机盖, 卸去两个软驱的卡盘螺丝, 拔掉软驱上的电源线和数据线, 将软驱拿出来, 先观察接受管是否被灰尘堵住或位置偏移, 仔细检查一切正常。怀疑是写保护检测电路有问题, 用小夹子将接受管两脚短路, 然后联机进行写文件测试一切正常, 排除了写保护检测电路的接受问题, 接着将软驱脱机检查接受管, 在线测量接受管正向阻值为 25K, 反向 45K 正常。为了进一步证实, 将接受管焊下后放到另一个驱动器上联机正常。这样就肯定了写保护检测电路的接受部分和接受管都正常。检查发光部分 (因发光管不好拆, 所以一般先检查接受部分) 联机测得发光管电压为 1.2V 的正常工作电压, 取掉电路板观察发光管是否有灰尘堵住, 没有, 将发光管从电路板上焊下换上好的试读写正常。

体会: 因发光管是红外线光, 通电检查时肉眼看不见光, 在维修时测量发光管的正反向电阻也都正常 (外电路也正常) 就要怀疑发光管的光可能变弱了, 这时就要换新的, 本例就是这种情况。

329

LQ1600K 打印

机纸尽检测器故障

成都飞机工业公司经理部 柯 纯

故障现象: LQ1600K 打印机加电后自检正确, 虽然打印纸没有装入但纸尽灯不亮, 一按“进纸/退纸”键, 打印机执行退纸操作。

故障分析及维修: 若机器没有打印纸, 则加电后所有灯亮, 然后除电源灯外其它灯都熄灭, 小车归位后, 纸尽灯亮同时蜂鸣器鸣响, 这时按“进纸/退纸”键就进纸。由于打印机能正确初始化, 输纸机构能工作, 所以打印机的控制电路, 驱动电路基本上是正常的, 故障出在纸尽检测器上。

LQ1600K 打印机的纸尽检测器是一个机械触点开关, 开关一端通过 CN9 接地, 另一端通过 R66 电阻接 +5V, 并通过 R65 接 CPU 的第 2 脚, CPU 的第 2 脚通过电容 C16 接地。当有纸时开关断开, +5V 通过 R65 送到 PA1, CPU 确认打印纸装好, 纸尽灯不亮, 一按“进纸/退纸”键, 打印纸退出。当打印纸退完后, 纸尽开关接通, PA1 电平变低, CPU 确认打印机缺纸, 点亮纸尽灯并使蜂鸣器鸣叫, 这时再按“进纸/退纸”键, 打印纸将自动进入到打印位置。

由故障现象可知是纸尽检测开关损坏。检查发现 PA1 总为高, 所以 CPU 总判定打印机装好, 造成上述故障现象。更换纸尽检测开关后故障排除。

330

编辑同志:

您好!

最近我获得一套游戏软件“侠影记”, 这是台湾出品的大型全中文角色扮演游戏, 支持所有显示方式, 图形精美, 音响出众, 语言幽默, 很值得一玩。但该游戏版图很大, 不易找到道路。我在进入一个魔教的山洞后学会了“残影金龙斩”功夫, 攻击力大大提高。但有一道栅栏门却不知怎样打开, 至使游戏不能进行下去, 特借《电脑》杂志之一角, 希望有高手予以指点, 不胜感激。(若有朋友需此软件, 请来信索取。地址: 江苏南京龙潭建设新村 564-11。邮编: 210034)

此致

礼!

读者: 胡伟

1993.6.16

331

微机开关电源的维修方法与实例

武汉汉理新技术公司

冯晓沁

目前最普遍使用的 IBM / AT 及其兼容机一般都采用功率为 200W 的双管半桥式开关电源。

这类电源的故障可分为三类：前级故障，控制驱动电路故障，输出级故障。前级故障往往表现为保险丝熔断，故障原因则常是整流桥被击穿或功率开关管损坏。这类故障比较容易找出故障原因。

输出电压各档均无电压这种现象，故障往往出在控制回路和驱动电路。此时由于输出电压没有建立，无反馈信号，使得控制电路的工作状态正常与否无法判断。因此维修时，可在不加 220V 交流电的情况下，在 +12V 输出端接一个可调直流电源，给控制集成块（以 TL494 为例）提供工作电压，这样就可以用示波器观察控制回路及驱动电路的波形了。在维修中，会发现有的型号电源，按上述方法观察不到 TL494 的驱动脉冲波形。这是因为在这种电路板 TL494 的死区控制端接有保护电路。维修中因为只有 +12V 是正常的，其它电压均为 0，保护电路会使死区控制端电压变为 +5V，使得 TL494 电路内的比较器 1 无脉冲输出从而也没有驱动脉冲输出。这时只须将死区控制端 4 脚与相联的保护电路断开就可以观察到控制电路的波形了。用这种人为设定工作状态的方法逐级观察可以判断控制集成块（如 TL494），驱动三极管，功率开关管及其附近的偏置电阻，开关二极管等元件正常与否。

输出级故障，常见的有 PG 信号不正常。286 电源中 PG 信号是由 LM339 产生的。LM339 是四电压比较器集成电路，每个比较器的输出和输入电压的关系是：

$$V_{\text{输出}} = V_{\text{饱和}} \times [(V_{\text{同相}} - V_{\text{反相}}) \text{ 的符号}]$$

V 饱和的数值是 5V，各直流输出电压检测信号反馈到 LM339 进行逻辑与，当各检测信号正常时，LM339 就会产生一个正常的 PG（电源好）信号，其值为 +5V。

下面举几个维修实例：

例 1，电源各档均无输出，保险丝完好。

检修过程，保险丝完好，一般不会是前级故障。静态测量整流桥，大功率管，果然没有问题。加电测量，整流桥输出为直流 350V 左右，这说明前级电路没有问

题。接下来需要检查控制与驱动电路。见图 1，拆除 D7，D8，使 TL494 的死区控制端第 4 脚无论在什么情况下都为低电平。将直流可调电源调至 12V，接至开关电源 12V 输出端，TL494 开始工作，用示波器观察其 8 脚和 11 脚，均输出正常的高频矩形脉冲，一路查至 T2 的初级，发现 T2 的 6 端与 8 端波形极不对称，把 T2 的次级与后面的电路断开，再观察初级波形，波形恢复正常。以此可以断定，T2 初级以前的电路和 T2 本身均没有问题。把原来断开的电路连好，进一步观察，发现 Q1 的 be 极之间为不对称波形，Q2 的 be 极间没有波形，进一步检查，发现电阻 R8 开路，更换该电阻，故障排除。

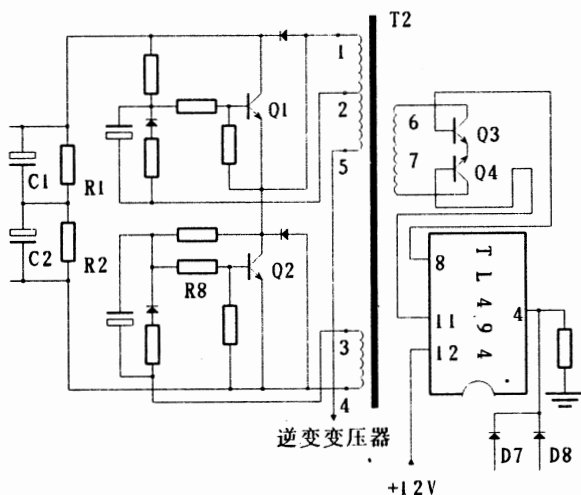


图 1

例 2，电源各档均无输出，保险丝完好。

检修过程，按上述方法，给 TL494 提供正常的工作电压，结果它的 8 脚和 11 脚均为恒定电平，无任何矩形脉冲输出，静态测量 TL494，发现 11 脚-12 脚之间电阻与 8 脚-12 脚之间电阻不等，更换 TL494，故障排除。

例 3，电源直流输出 +5V，+12V，-5V，-12V 均正常，但 PG 信号电压为 0.14V。

检修过程，该型号电源的 PG 信号由 LM339 的 2

脚产生, 根据 LM339 的内部电路图和电路连线, 2 脚的电压与其它引脚电压有如下关系:

$$\begin{aligned} V_2 &= 5 \times (V_5 - V_4) \text{ 的符号} \\ &= 5 \times (V_1 - V_4) \text{ 的符号} \\ &= 5 \times [5 \times (V_7 - V_6) \text{ 的符号} - V_4] \text{ 的符号} \end{aligned}$$

第 4 脚和第 6 脚是 +5V 的检测信号, 同时也作为比较器的基准电压, 正常时应为 2.5V。第 7 脚电压是 +12V 的检测信号。经测量发现第 7 脚电压很低, 而第 7 脚电压是从 +12V 整流输出处通过整流二极管 D12 再分压得到的。通过 D12 后的电压正常值为 24.5V, 而该电源此处只有 17V 左右。把 D12 拆下来测量, 正反向电阻都为 100 欧姆, 更换该整流二极管, PG 信号就正常了。

例 4. 电源直流输出 +5V, +12V, -5V, -12V 均正常, PG 信号最初为 +5V, 5~10 分钟之后变为 0V。

检修过程, 由此可以判定, 一定有一个电容存在漏电。则量 LM339 的各引脚, 发现 1 脚的电压由 4.6V 逐渐下降。当电压降到 2.5V 以下时, PG 信号就变成 0V 了。1 脚除与 5 脚相联之外还通过一电阻和一滤波电容接地。正是这个滤波电容漏电引起了 PG 的不正常。

例 5. 电源启动后读软盘时电源保护。

检修过程, 这种现象说明 +12V 带负载能力太差, 测量 +12V 输出端, 实际电压只有 9.8V。只有 12V 组不行, 说明问题出在逆变变压器次级 12V 输出绕组以后的电路中。于是, 更换 +12V 整流二极管 D16, +12V 电压输出仍是 9.8V 左右, 说明原 +12V 整流管性能是正常的。把与逆变变压器次级 +12V 绕组相联的瓷片电容 C22 拆下来测量, 发现它已经开路。更换该瓷片电容, +12V 输出的端电压就正常了。

例 6. 电源偶尔可以启动, 一经启动便能正常工作下去。

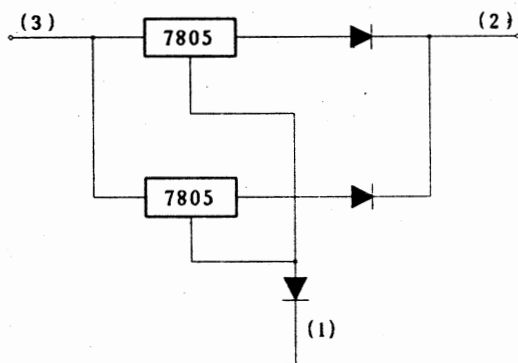
检修过程, 用前面所说的检查控制驱动电路的办法, 观察到各级波形均正常, 说明控制电路没有问题。检查整流桥, 功率开关管也没有发现问题。经过仔细观察, 发现只有 C1, C2 (200W 电解电容), 见图 1, 两端电压完全降为 0V 的时候。电源才能启动。由此推测, 可能是两个 200W 大功率电解电容与两个功率开关管构成的充放电回路性能变差, 充放电速度太慢, 导致电源不能启动。充放电时间常数 $\tau = RC$, C 值越小, 充放电越快。用两个 150 μ F (200W) 的电解电容更换原来的 470 μ F (200W) 的电容, 电源就可以启动了。

用 7805 稳压块巧修

四通 MS-2401 打字机

石家庄市计算机职教中心 陈丁君

笔者在维修四通 MS-2401 打字机时发现, 因 ST-3052V 三端稳压集成块被烧坏, 没有 +5V 电压输出, 导致机器无法工作, 而手头上没有 ST-3052V 这一集成块, 市场上又难以买到, 一时无法修复打字机。于是笔者试着用很方便买到的 7805 三端稳压集成块代之, 由于 7805 稳压块的额定输出电流比 ST-3052V 稳压块要小, 尽管加大了散热片, 使用不到一小时 7805 稳压块还是被烧坏。笔者想如果用两块 7805 稳压块并联使用应该可以达到目的。通过对 7805 稳压块内部电路的分析, 发现不能把两块 7805 稳压块机械地并联在一起使用, 因为任何两块 7805 稳压块其性能不可能完全一致, 如果把它们机械地并联使用, 在稳压输出过程中两输出端之间就会产生电位差, 引起其中一稳压块产生电流倒灌现象, 不但达不到稳压目的, 而且会使两块稳压块均被烧坏, 严重时还会烧坏主板中的其它集成块。笔者采用在两稳压块输出端各加一个二极管隔离的方法来避免电流倒灌现象的发生, 同时在它们的接地端再加上一个同样型号的二极管抬高接地端的电位以保证能得到稳定的 +5V 电压。通过上述连接方法 (下附电路图), 笔者很快就修复了这台 MS-2401 打字机。



巧用FOX函数简化程序提高效率

江苏省扬中县统计局 符卜成

FOX 拥有比较丰富的函数, 编程者如能巧妙地利用这些函数, 往往能收到事半功倍的效果。现把笔者在 FOX 编程中利用函数的一些收获介绍给大家。

一、利用 IIF 函数, 实现三重选择。见举例一: 判别一个增长百分值 A(二位小数), 如果 $A > 0$, 那么 $R = \text{增长 } A\%$; $A < 0$, $R = \text{下降 } A\%$; $A = 0$, $R = \text{平}$ 。可见有些三重选择, 利用 IIF 函数, 其一条命令, 就能代替用 DO CASE.....ENDC 或 IF.....ENDI 实现三重选择的八九条命令。

二、利用 LTRIM 函数, 生成无空格的数据型字符。见举例二: 任意大小的正整数值, 都能转换成无空格的数据型字符。可见利用 LTRIM 函数, 就无须用 IF.....ENDI 命令判断输入值的大小来确定字符串长度。

三、利用 AT 等函数, 实现多重选择。见举例三: 任何正整数值转换成中文大写, 如果用 IF.....ENDI 或 DO CASE.....ENDC 命令编制就显得十分繁杂。

四、利用 & 函数, 实现数组功能。见举例四: 假如有数据库 NUA, 共有 30 个数据型字段, 字段名分别为 A1, A2.....A30, 另有数据库 NUB, 共有 30 条记录一个数据型字段, 字段名为 N (长度为 NUA 库中的最大值), 利用 & 函数, 使 NUA 库中的各字段值依次成为 NUB 库中的数据记录。

五、利用 & 函数, 简便地进入子程序。目前流行的 25 行 FOX, 给广大用户带来更广阔的天地, 一个屏幕就可容纳几十个子程序菜单。见举例五: 这是一个简化的工资报表程序总菜单, 各子程序命令文件名统一为字母 GZ 开头加选择序号, 用 & 函数替代 DO CASE.....ENDC 命令, 程序显得简单明了。

```
set talk off
set safe off
close all
```

***举例一:

```
INPUT '录入需判别增长百分值' TO A
R=IIF(A=0,'平',IIF(A>0,'增长'+STR(A,6,2)+'%';
,'下降'+STR(-1*A,6,2)+'%')
? R
```

***举例二:

```
INPUT '录入需换成无空格的数据型字符值' TO X
X1=LTRIM(STR(X,16))
? X1
```

***举例三:

```
NU='0零1壹2贰3叁4肆5伍6陆7柒8捌9玖'
ACCE '录入需转换成大写的任何正整数值' TO V
N=1
DO WHILE N<=LEN(V)
N1=LTRIM(STR(N,3))
P&N1=SUBSTR(V,N,1)
NU&N1=SUBSTR(NU,AT(P&N1,NU)+1,2)
K=IIF(N=1,NU&N1,K+NU&N1)
N=N+1
ENDD
? K
```

***举例四:

```
BB=1
DO WHILE BB<=30
B=LTRIM(STR(BB,3))
USE NUA
P&B=A&B
USE NUB
GO BB
REPL N WITH P&B
BB=BB+1
ENDD
```

***举例五:

```
CLEA
XZ=0
@ 3,5 SAY '1生成文件5录入10审核'
@ 5,5 SAY '15修改 20汇总 25打印'
@ 1,0,7,30 BOX
@ 13,8 SAY '选择(0退出)' GET XZ
READ
XZ=LTRIM(STR(XZ,2))
IF XZ='0'
RETU
ENDI
DO GZ&XZ
```

谈PCTOOLS对深层子目录的加密

广东省曲江大宝山矿铁路管理处 储汉宾

一、提出问题

PCTOOLS 是一个强有力的、非常实用的工具软件包。它给广大的电脑用户提供了多方面的帮助。目前,介绍 PCTOOLS 使用技巧的文章很多,但介绍用 PCTOOLS 对深层子目录管理的文章却不多。现在,笔者就以对深层子目录加密为例,介绍一种用 PCTOOLS 对深层子目录进行管理的方法。

二、原理

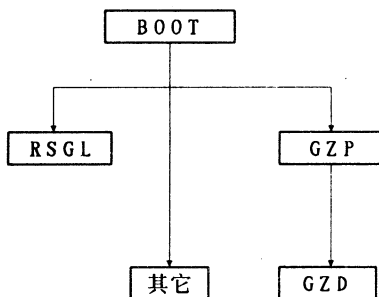
大家都知道,文件目录表是由 32 个字节的目录登记项组成,32 个字节目录登记项的分配结构如下:

位移	内容	长度
00	文件名	8
08	扩展名	3
0B	属性	1
0C	保留区	10
16	生成或最后修改的时间	2
18	生成或最后修改的日期	2
1A	起始簇	2
1C	文件长度(字节)	4

利用 PCTOOLS 的磁盘图形功能“Map”找出目标深层子目录的起始簇号,然后,根据所得的起始簇号数据,找出该目标子目录的目录登记项,并对这个目录登记项进行隐藏加密处理。

三、实例

假设硬盘 C 上存在着如下目录树:



其中一级子目录 GZP 存放工资管理的程序, GZD 存放着工资管理的数据。为了安全保密起见,对二级子目录进行加密。其步骤如下:

- 1、启动 PCTOOLS, 读入 C 盘的根目录文件;
- 2、按 F3 键进入磁盘管理状态, 按 M 键进入磁盘

图形管理;

3、按 F 键后, 再按 F10 与 G 键, 找出二级子目录 GZD 下第一个文件的磁盘存储起始簇号;

4、按 ESC 键返回磁盘管理状态, 按 E 键进入编辑状态, 按 F2 键后, 在所出现的菜单下, 按 C 键后输入第三步所得的起始簇号, 于是就找到二级子目录在磁盘中的位置。

5、按 PGDN 键找到二级目录的目录登记项, 将第十二字节 10 修改为 1A, 将第三十二字节 00 修改为 02, 或将该子目录名改为小写字母 gzd。

这样修改后二级子目录就在 DOS 命令和 PCTOOLS 下, 都不能显示出来。通过这种方法就能够达到保护数据的目的。

335

删除子目录的简捷方法

广东省机械技校省机床厂分校 杨 扬

一般删除子目录的方法是使用 DOS 的 RD 命令或 PCTOOLS 的磁盘特殊服务功能。但都必须先将子目录下的文件删除。如果遇到多级或多分支的大型子目录名, 将要逐级删除文件, 再逐级逐个删除目录名, 非常烦琐。

其实 DOS 提供了一个 DQ 命令, 可以极为方便快捷地一齐删除子目录及其下属各级全部文件和子目录。方法是在 DOS 提示符下键入 DQ (子目录) 即可, 如缺省子目录名删除当前子目录。回车后可见提示:

All files and directories in d: \子目录名 will be deleted! Are you sure (Y/N)?

提醒你注意, 确认后键入“Y”删除工作顷刻完成。注意, 该命令破坏性极强, 使用时一定要小心。

另外, 这是外部命令, DOS 系统文件中必须含有 dq.exe 文件方可执行。

336

DOS状态下的菜单制作

江苏盐城微机应用研究所 徐富军

重庆大学汽车工程系 89 级 谢志坚

制作 DOS 状态下的功能菜单，给日常应用程序做一个美丽的封装，是每位程序员们的向往。但 DOS 没有这类的菜单命令。虽然，以往报刊也提供了一些方法，但不够简洁、易更改、通用性不强。

下面介绍两个简单实用的 DOS 状态下菜单制作。

方法一：

先调用 DEBUG 调试程序，做以下操作：

```
C>DEBUG MENU
File not found
-A
2479: 0100 MOV DX, 0110 ; 菜单数据区首址
2479: 0103 MOV AH, 09
2479: 0105 INT 21
2479: 0107 AX, 0C01
2479: 010A INT 21
2479: 010C MOV AH, 4C
2479: 010E INT 21
2479: 0110
-RCX
CX 0000
:10
W
Writing 0010 bytes
-Q
```

再用文字编辑软件生成以下内容的文件（暂命名为 XUD）：

```
0---WPS
1---CCED
2---BASIC
3---FOXBASE
4---DOS
```

请选择：\$；\$

最后用 COPY 命令把菜单文件连接起来：

```
C>COPY MENU+XUD MENU.COM
```

这样 MENU 就做成了。

方法二：

此方法与方法一不同之处在于方法一把菜单内容都写到 MENU.COM 文件中，使文件不便于修改，而以下介绍的，则是把菜单内容写在一个批处理文件中。

先调用 DEBUG 调试程序，做以下工作：

```
C>DEBUG MENU.COM
File not found
-A
2479: 0100 MOV AX, 0C07
2479: 0103 INT 21
2479: 0105 MOV AH, 4C
2479: 0107 INT 21
2479: 0109
-RCX
CX 0000
:9
W
Writing 0009 bytes
-Q
```

再模仿以下批处理程序生成自己的菜单即可

```
@ECHO OFF
CLS
ECHO MENU MODEL
ECHO 1---CCDOS
ECHO 2---PROGRAMING
ECHO 3---WPS
.....
ECHO Press number 1-5 or E...
IF ERRORLEVEL 49 GOTO SUB1
IF ERRORLEVEL 50 GOTO SUB2
.....
:SUB1
CD\CCDOS
CCCC
.....
```

编后语：

本文的“方法一”是徐富军作者所写，“方法二”则是谢志坚的作品，由于两位作者的文章内容非常接近，所以经编辑修改后联名发表，稿费由两位作者平分。

利用递归方法绘制分形图案

中山大学物理系 徐志凌

自 1975 年 B.Mandelbrot 提出分形几何概念以来,分形几何在许多领域得到了广泛的应用,随着计算机作图技术的发展,分形已经成为一种日益流行的艺术形式。

所谓分形,简言之,就是指组成部分与整体以某种方式相似,局部放大后可以在某种程度上再现整体。举个简单的例子,如图 1 所示,一棵树是由一些分枝构成的,就其中某个分枝来看,它具有与整棵树相似的形状,是由更小的分枝以相同的规律构成的。于是我们就称这一棵树具有分形结构。

在计算机程序设计中,递归方法与前面所说的分形有很密切的联系。大家知道,递归就是一个函数或子程序不断调用自身的过程。每一次调用,实际上都是在不同的层次上对自身的重复,这与分形的法则是相对应的。

基于这种考虑,如果我们制定一些简单的规则,利用递归方法,就可以在计算机上绘制出各种各样美丽的分形图案。

本文中所附的图 1~图 3 都是作者利用递归方法绘制的,在此,作者提供用 TURBO C 语言编制的两段源程序供参考,其运行结果分别为图 1 图 2。

读者不妨充分发挥自己的想象力,制定一些巧妙的规则,绘出更美好的图案。另外,作者正在收集各种分形图案,大家若有兴趣,可与杂志社或与作者本人联系,以期互相交流。

下面就几幅图的绘制规则作一简单说明。

图 1 中每个分枝(包括主干)的绘制规则是,先按某一方向画一条直线段,然后在此直线段上找到一系列节点,在每一节点处向左、右偏转 60° 各画一条分枝。本文中各节点的位置和节点处所画分枝的长度按“黄金分割法”求得,即第一节点所截得主干的两部分长度之比为 0.618,其余类推。当分枝的长度小到规定值时就返回,不再画更小的分枝。

图 2 中,先画一个大三角形,连接三角形三条边的中点,得到四个较小的三角形,将外围的三个小三角形经过与大三角形相同的处理,得到一系列更小的三角形。以此类推,将三角形不断分割下去,直到最小的三角形的边长小于某个值时返回,不再细分。

图 3 的绘制遵循如下规则:先画一个大正方形,将其分成 16 个相等的小正方形,如图 4 所示,将编号为 2 的小正方形移至 2' 处,8 移至 8' 处,15 移至 15' 处,9 移至 9' 处,擦掉 2、8、15、9 四个小正方形,这样得到的 16 个小正方形,每一个再经过与前面相同的处理,不断重复,直到最小的正方形边长小于某一值时返回,最后可得到图 3 的结果。

附图及程序清单:

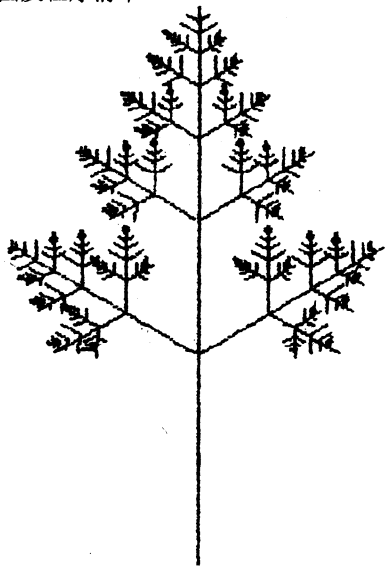


图 1

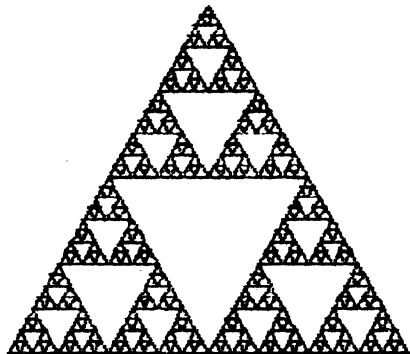


图 2

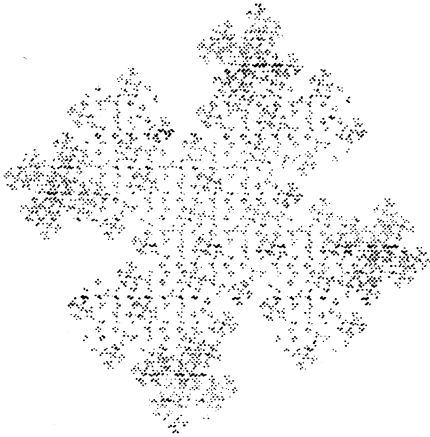


图 3

```

/* Program frac1.c */
#define g 0.618
#define PAI 3.14
#include <graphics.h>
#include <math.h>
float thita=60.0;
main()
{ int gm,gd;
  detectgraph(&gd,&gm);
  initgraph(&gd,&gm,"");
  grow(300,300,280.0,90.0);
  getch();
  closegraph();
}
grow(x,y,lenth,fai)
int x,y;
float lenth,fai;
{ int x1,y1;
  int nx,ny,count;
  float nlenth;
  x1=x+lenth*cos(fai*PAI/180.0);
  y1=y-lenth*sin(fai*PAI/180.0);
  line(x,y,x1,y1);
  if(lenth<10) return;
  nlenth=lenth;
  nx=x;
  ny=y;
  for(count=0;count<7;count++)
  { nx=nx+nlenth*(1-g)*cos(fai*PAI/180.0);
    ny=ny-nlenth*(1-g)*sin(fai*PAI/180.0);
    grow(nx,ny,nlenth*(1-g),fai+thita);
    grow(nx,ny,nlenth*(1-g),fai-thita);
    nlenth*=g;
  }
}

/* Program frac2.c */
#include <graphics.h>
main()
{ int gm,gd;
  int x1=360,y1=100,x2=260,y2=273,x3=460,y3=273;
  detectgraph(&gd,&gm);
  initgraph(&gd,&gm,"");
  line(x1,y1,x2,y2);

```

```

  line(x2,y2,x3,y3);
  line(x3,y3,x1,y1);
  draw(x1,y1,x2,y2,x3,y3);
  getch();
  closegraph();
}
draw(x1,y1,x2,y2,x3,y3)
int x1,y1,x2,y2,x3,y3;
{ int xml,ym1,xm2,ym2,xm3,ym3,fx,fy;
  xml=(x1+x2)/2;
  yml=(y1+y2)/2;
  xm2=(x2+x3)/2;
  ym2=(y2+y3)/2;
  xm3=(x3+x1)/2;
  ym3=(y3+y1)/2;
  line(xml,ym1,xm2,ym2);
  line(xm2,ym2,xm3,ym3);
  line(xm3,ym3,xml,ym1);
  fx=xml-xm2;
  fy=ym1-ym2;
  if((fx*fx+fy*fy)<150) return;
  draw(x1,y1,xml,ym1,xm3,ym3);
  draw(xml,ym1,x2,y2,xm2,ym2);
  draw(xm3,ym3,xm2,ym2,x3,y3);
}

/* Program frac3.c */
#include <graphics.h>
main()
{ int gm,gd;
  int x0=360,y0=170,side=128;
  detectgraph(&gd,&gm);
  initgraph(&gd,&gm,"");
  move(x0,y0,side);
  printf("\n");
  getch();
  closegraph();
}
move(x0,y0,side)
int x0,y0,side;
{ int m,i,j;
  m=side/4;
  setfillstyle(1,1);
  bar(x0,y0-3*m,x0+m,y0-2*m);
  bar(x0+2*m,y0,x0+3*m,y0+m);
  bar(x0-m,y0+2*m,x0,y0+3*m);
  bar(x0-3*m,y0-m,x0-2*m,y0);
  setfillstyle(0,0);
  bar(x0-m,y0-2*m,x0,y0-m);
  bar(x0+m,y0-m,x0+2*m,y0);
  bar(x0,y0+m,x0+m,y0+2*m);
  bar(x0-2*m,y0,x0-m,y0+m);
  if(m<8) return;
  for(i=0;i<4;i++)
  { for(j=0;j<4;j++)
    { if(! ((i==0&&j==2)||((i==1&&j==0)||
      (i==2&&j==3)||((i==3&&j==1))))
      move(x0-3*m/2+i*m,y0-3*m/2+j*m,m);
    }
  }
};
move(x0+m/2,y0-5*m/2,m);
move(x0+5*m/2,y0+m/2,m);
move(x0-m/2,y0+5*m/2,m);
move(x0-5*m/2,y0-m/2,m);
return;
}

```


中华学习机

A, B, T 类文件 de 相互转换

青岛市 37003 部队 783 办公室 翁元祥

我们知道中华学习机上经常用到的文件有三种:即 A 类文件, B 类文件和 T 类文件。A 类文件是专门用于保存 BASIC 程序的文件。T 类文件是用于保存数据信息的文件, 常称为文本文件, B 类文件是用于保存机器语言的文件(含二进制文件), 但有时要将上述三种类型的文件相互进行转换, 以满足不同的需要, 如将 A 类文件转换为 B 类文件存取就打破了常规方法, 为我们提供了一种文件保密方法, 再如要将 DOS 下的 A 类文件转到 CPM 下使用, 也需要将 A 类文件先转成 T 类文件, 然后才能转为 CPM 下的 BASIC 文件, 总之需要是多种多样的, 因此, 解决好这三种文件的相互转换是很有意义的, 笔者介绍给大家的是一个简单实用的通用程序, 程序的主要功能有三个: 一是将 A 型文件转为 B 型文件; 二是将 A 型文件转为 T 型文件; 三是将 B 型文件转为 A 型物件; 四是将 B 型文件转为 T 型文件, 程序采用模块化结构编成, 通俗易懂, 使用方便, 转换时只要根据提示即可完成上述各项操作, 程序在 CEC-I 中华学习机上通过, 附原程序如下:

```
5 D$ = CHR$(4)
10 PR#3: CALL 1002: HGR2: HOME
20 VTAB 2: PRINT " 中华机 A,B,T 类文件相互转换"
30 PRINT "*****"
40 VTAB 4: HTAB 4: PRINT "1,A 文件转为 B 文件": VTAB 5:
HTAB 4: PRINT "2,A 文件转为 T 文件"
50 VTAB 6: HTAB 4: PRINT "3,B 文件转为 A 文件": VTAB 7:
HTAB 4: PRINT "4,B 文件转为 T 文件"
70 VTAB 8: HTAB 4: PRINT "5,退出....."
80 VTAB 10: INPUT "请选择: "; E: IF E < 1 OR E > 5 THEN 80
85 ON E GOSUB 100,200,300,300,700
90 GOTO 10
100 HOME: PRINT "A 转 B": PRINT "请按以下步骤进行:
LOAD A 类文件名"
120 PRINT "首地址: PEEK(104) * 256+PEEK(103)"
130 PRINT "末地址: PEEK(106) * 256+PEEK(105)"
140 PRINT "再键入 BSAVE 文件名, A 首地址, L 末地址-首地址
+1 回车即可!"
145 GET W$: END
```

```
200 HOME: PRINT "A 转 T": PRINT "请按以下步骤进行
:LOAD A 类文件名"
210 PRINT "再键入以下语句行:"
215 PRINT "0 PRINT ^D OPEN Fn": PRINT ^D WRITE
Fn": POKE 33,33: LIST: PRINT ^D CLOSE": END"
218 PRINT "其中 Fn 为 T 类文件名"
220 PRINT: PRINT "然后 RUN, 最后 EXEC Fn 即可": GET F$
: END
300 HOME: IF E = 3 THEN PRINT "B 转 A": GOTO 310
305 PRINT "B 转 T:"
310 INPUT "B 类文件名: "; F$
315 PRINT D$; "BLOAD"; F$
320 L = PEEK(43616)+PEEK(43617) * 256
325 A = PEEK(43634)+PEEK(43635) * 256
326 IF E = 3 THEN 329
328 B = L+A: C = 100: D = 10: INPUT "T 类文件名: "; F$: GOTO
330
329 B = L+A: C = 100: D = 10: INPUT "A 类文件名: "; F$
330 C0 = C: C1 = C+D: C2 = C1+D: C = C2+D
340 D$ = CHR$(13)+CHR$(4)
345 PRINT D$; "OPEN"; F$: PRINT D$; "DELETE"; F$
350 PRINT D$; "OPEN"; F$: PRINT D$; "WRITE"; F$
355 PRINT C0; "FORI = "; A; "TO"; B
360 PRINT C1; "READ N: POKE I, N"
365 PRINT C2; "NEXT I"
370 FOR I = A TO B
375 LL = LL+1: IF LL = 40 THEN LL = 1
378 IF LL = 1 THEN PRINT: PRINT C; "DATA"; C = C+D
380 PRINT PEEK(I)
382 IF I = B OR LL = 39 THEN 385
284 PRINT " ";
385 NEXT I
390 PRINT: PRINT D$; "CLOSE"; F$: HOME
395 IF E = 4 THEN HOME: HTAB 15: VTAB 5: PRINT "转换成功!": END
396 PRINT "请输入 NEW, 再键入 EXEC 文件名, 最后键入 SAVE
文件名即可": END
700 HOME: HTAB 15: VTAB 6: PRINT "再见!"
710 END
```

1993 年广东省青少年奥林匹克信息学竞赛

试题分析及参考解答

邬家炜

1993 年广东省青少年奥林匹克信息学竞赛的决赛在广东佛山市举行。决赛共分为一、二试。下面就一试试题（已在 1993 年《电脑》杂志的第 8 期上登出）竞赛的情况作一个简单的分析，并给出有关的参考解答供大家参考。

第一题：这个题目难度不大，不少的同学都做对了，但也有部分同学对题目中比赛的规则和记分方法未能正确理解。因而会出现处理第 10 局时，不管第几球，反正击倒全部 10 个柱就奖励滚两球。而忘记了题目一开始就摆出的第 10 局最多可滚三球。又如在记分方法上未能注意处理好各问累计得分超过 100 分时，以后每击倒一柱，得分以 2 分计算。（即不再是击倒一柱得一分）下面给出一个参考解答。

```

DIM s(10),score(10),next(10)
LET plus=1
LET total=0
CLEAR
PRINT "ord strike score total"
FOR i=1 to 9
  LET next(i)=2
  DO
    LET pro
    ="di"str
    (i) "ju"str
    (3-next(i))"
    ASK cursor y,x
    DO
      SET cursor y,x
      INPUT prompt pro
    :n
    LOOP until n>-1 and n<11 and int(n)=n and n+s(i)<=10
    FOR k=1 to i-1
      IF next(k)>0 then
        LET score(k)=score(k)+n*plus
        LET next(k)=next(k)-1
      END IF
      IF next(k)=0 then
        SET cursor y,40
        LET total=total+score(k)
        IF total>100 then LET plus=2
        PRINT "di";k;"ju";repeat
        (" ",5-len(str
        (score(k))))score(k);" ";total
        LET y=y+1
        LET next(k)=-1
      END IF
    NEXT k
    LET score(i)=score(i)+n*plus
    LET s(i)=s(i)+n
    IF next(i)=2 and n=10 then EXIT DO
    IF next(i)=1 and s(i)=10 then EXIT DO
    LET next(i)=next(i)-1
    LOOP until next(i)=0
    IF next(i)=0 then
      SET cursor y,40
      LET total=total+score(i)
      IF total>100 then LET plus=2
      PRINT "di";i;"ju";repeat
      (" ",5-len(str
      (score(i))))score(i);" ";total
      LET next(i)=-1
    END IF
  END IF

```

```

NEXT i
LET next(10)=2
LET ord=1
DO
  LET pro
  ="di 10 ju"str
  (ord)"
  ASK cursor y,x
  DO
    SET cursor y,x
    INPUT prompt pro
  :n
  LOOP until n>-1 and n<11 and int(n)=n
  FOR k=1 to 9
    IF next(k)>0 then
      LET score(k)=score(k)+n*plus
      LET next(k)=next(k)-1
    END IF
    IF next(k)=0 then
      SET cursor y,40
      LET total=total+score(k)
      IF total>100 then LET plus=2
      PRINT "di";k;"ju";repeat
      (" ",5-len(str
      (score(k))))score(k);" ";total
      LET y=y+1
      LET next(k)=-1
    END IF
  NEXT k
  LET score(10)=score(10)+n*plus
  LET s(10)=s(10)+n
  IF ord=1 and n=10 then LET next(10)=3
  IF ord=2 and s(10)=10 then LET next(10)=2
  LET next(10)=next(10)-1
  LET ord=ord+1
  LOOP until next(10)=0
  IF next(10)=0 then
    SET cursor y,40
    LET total=total+score(10)
    IF total>100 then LET plus=2
    PRINT "di";10;"ju";repeat
    (" ",5-len(str
    (score(10))))score(10);" ";total
    LET next(10)=-1
  END IF
END

```

第二题：这个题的难度一般，其主要的算法是判别输入的 S 字符序列中是否有 P 字符序列（即单词表中的单词）有的则输出，但是题目中还附加了二个条件，而按附加的二个条件对字符串的划分，同学在做题时就不很理想。本来题目的第 4)，第 5)，已写得很明确，并且还举了三个例子。但就是分解的算法没有做好，因而对很简单的测试用例也未能通过。例如字符串 ACELL：应分解为：A，ACE，CELL；对条件一最多能划出 2 个单词 A，CELL（输入的每个字母只能用一次）；对条件二可划分为 A，CELL（输入的每个字母均要使用且仅使用一次）。再者，那些是非法的输入也没有考虑，例如，空串 THE END 等。下面给出个参考解答。

```

uses crt;
var
  i,j,k,l,n,m,t,v,o,p:integer;
  a,b:array [1..100] of string[10];
  d:array [1..100,1..100] of 0..1;

```

```

c:array [1..100] of integer;
st,ss:string;
z:array [1..100] of string;
u:array [1..100] of integer;
procedure find(e,f,g:integer);
var
  r:integer;
begin
  r:=0;
  if e<>0 then
  begin
    c[e]:=c[e]+1;
    if e>1 then
    if d[c[e-1],c[e]]=0 then r:=1;
  end;
  if (r=0) and (g=0) then begin
    k:=0;ss:="";
    for i:=1 to e do
    begin
      k:=k+length(b[c[i]]);
      ss:=ss+b[c[i]]+'';
    end;
    if k=n then begin
      l:=l+1;z[l]:=ss;end;
    end;
    if (r=0) and (e=v-1) and (g=1) then
    begin
      for i:=1 to v-1 do
        write(b[c[i]],' ');
        write('':10);
      end;
      if (r=0) and (e<m) then
      begin
        c[e+1]:=0;
        if e+1>v then v:=e+1;
        for f:=1 to o do
          find(e+1,f,g);
        end;
      end;
    end;
  end;
begin
  clrscr;
  A[1]:='A'; A[2]:='RAM'; A[3]:='AT'; A[4]:='HEN';
  A[5]:='THEN'; A[6]:='NEXT'; A[7]:='TEXT'; A[8]:='ATHEN';
  A[9]:='THERE'; A[10]:='HERE'; A[11]:='HEAT'; A[12]:='CAT';
  A[13]:='READ'; A[14]:='THE'; A[15]:='EAT'; A[16]:='SIDE';
  A[17]:='AS'; A[18]:='ACE'; A[19]:='ALL'; A[20]:='CELL';
  A[21]:='CALL'; A[22]:='SELL'; A[23]:='DISK'; A[24]:='END';
  write('st=');read(st);
  n:=length(st);j:=0;
  for i:=1 to n do
  if (st[i]<'A') or (st[i]>'Z') then j:=1;
  if j=1 then writeln('No..');
  m:=0;
  while j=0 do
  begin
    writeln('No. 1');
    for i:=1 to n-1 do
    for j:=i to n do
    begin
      ss:="";
      for k:=i to j do
        ss:=ss+st[k];
      for k:=1 to 24 do
        if a[k]=ss then begin
          m:=m+1;b[m]:=a[k];t:=0;
          u[m]:=i-1;
          for l:=1 to m-1 do

```

```

if b[l]=a[k] then t:=1;
if t=1 then m:=m-1;
if t=0 then write(a[k],':10);
k:=24;end;
end;
for i:=1 to m do
for j:=1 to m do
d[i,j]:=0;
for i:=1 to m-1 do
for j:=i+1 to m do
begin
  t:=0;
  for k:=1 to length(b[i]) do
  for l:=1 to length(b[j]) do
  if k+u[i]=l+u[j] then t:=1;
  if t=0 then d[i,j]:=1;
end;
c[1]:=1;v:=0;o:=m;:=0;m:=m+1;
find(0,0,0);
m:=v-1;writeln;
writeln('NO..2');
find(0,0,1);
writeln;writeln('NO..3');
for i:=1 to l do
  write(z[i],':10);
end;
end.

```

第三题：题目难度中等。从测试的结果来看，大部分的同学都能判断输入的一个密码是否属于 CS9301 这个帐号。但按题意生成所有的密码就没有做对，这主要反映在算法上对密码的词序，可用字符范围，特别是字母 C 可用二次等问题表达不清，而导致生成错误密码，甚至个别的生成的速度极慢。其实题目的第 1 问与第 2 问是相关联的。为什么有些同学的答案中第 1 问得分，而第 2 问却错呢？那是因为我们以测试数据（测试用例）只能检查所编写的程序有没有完成这些功能，而不能证明程序算法的正确性。何况我们只能用一组很有限的测试数据来测试，下面给出一个参考解答。

```

program tt;
uses crt,dos;
CONST DI:ARRAY[1..11]OF CHAR='LCWZYGC9301';
H:SET OF CHAR=['L','C','W','Z','Y','G','C','9','3','0','1'];
var N,M,i,j,k,jj,kk,l,p:integer;
tot:longint;
S:STRING;
f,B:ARRAY[1..10]OF SHORTINT;
bi,cg:boolean;
HH:ARRAY[0..9]OF SHORTINT;
bk,cb:char;
FF:ARRAY['A'..'Z']OF BYTE;
a:array[0..10]of char;

procedure print;
var l,i,j,k:integer;
s1,s:string;
begin
  f[b[4]]:=4;
  if (f[2]=0)and(f[7]<>0) then
  begin
    f[b[4]]:=0;exit;
  end;
  if (f[2]<>0)and(f[7]<>0) then
  if f[2]>f[7] then

```

```

begin
  f[b[4]]:=0;exit;
end;
s:='';
for i:=1 to 4 do
  begin a[i]:=d[b[i]];s:=s+a[i];end;
for i:=2 to 5 do
  for j:=i to 5 do
    for k:=j to 5 do
      for l:=k to 5 do
        begin s1:=s;
          insert('1',s1,i);
          insert('0',s1,k);
          insert('3',s1,j);
          insert('9',s1,i);
          inc(tot);clrscr;
          writeln(tot,' ',s1);
        end;
        f[b[4]]:=0;
      end;
    end;
  end;
end;
PROCEDURE find3;
var i,j,k,l,m,kk:integer;
begin
  fillchar(b,sizeof(b),0);
  fillchar(f,sizeof(f),0);
  i:=1;b[i]:=0;tot:=0;
  repeat if b[i]<7 then
    begin inc(b[i]); bi:=true;
      if f[b[i]]>0 then bi:=false;
      if bi then
        if i=4 then begin f[b[4]]:=i;print(f[b[4]]:=0;end
        else begin f[b[i]]:=i;inc(i);b[i]:=0;end;
        end
      else begin dec(i);
        if i>0 then f[b[i]]:=0;
        end;
      until i=0;
      * writeln('total=',tot);
    end;
  begin clrscr;
    hh['0']:=3;hh['9']:=1;hh['3']:=2;hh['1']:=4;
    repeat
      repeat {
        i:=write('choose a number(2..3)=');readln(n) {
        it}
      until iorresult=0;
      until (n=2)OR(N=3);
      CASE OF
        2:BEGIN BI:=TRUE;
          WRITE('S=');READLN(S);
          IF LENGTH(S)<>8 THEN BI:=FALSE;
          FOR J:=1 TO 8 DO
            IF S[J] IN H THEN
              ELSE BI:=FALSE;
              IF (S[1]>'Z')OR(S[1]<'A') then bi:=false;
              if NOT(bi) then writeln(' this is not I')
            else
              begin
                for ch:='0' to '9' do
                  if ch in b then
                    for bk:=chr(ord(ch)+1) to '9' do
                      if bk in b then
                        begin k:=pos(ch,s);
                          l:=pos(bk,s);
                          if (hh[ch]-hh[bk])*(k-l)<0 then
                            begin writeln(' THIS IS NOT I');exit;
                          end;

```

```

end;
FILLCHAR(FF,SIZEOF(FF),0);
FOR J:=1 TO 8 DO
  IF (S[J]>='A')AND(S[J]<='Z') THEN
    INC(FF[S[J]]);
  IF (FF['Y']>1)OR(FF['L']>1)OR(FF['Z']>1)OR(FF['G']>1)
  OR(FF['C']>2)OR(FF['W']>1) THEN BI:=FALSE;
  IF NOT(BI) THEN begin writeln(' not into I');exit;END;
  writeln('this is in I');
end;
END;
3:FINI3;
END;
end.

```

第四题：本题难度大（相对我们的考生来说，也许是他们很少接触类似这样的仿真程序）根据题意，把问题解一点倚定在给定时间，三个排队队列所包含病人人数都是明确的，排队队列发生的变化情况，一是诊室来了一个新病人，这样就会有一个队列增加一个病人，另一种情况是三个队列中有一个排队队列的第一个病人，看完了病，这时这个排队队列就会减少一个病人。使排队队列状态发生变化的活动有四个（一个病人走进诊室，加上一个病人离开的三种情况）。本来要描述这样的事实并不难，但可能是考生对仿真二字不理解，没有考虑如何使所编的程序能不断地反映现实世界（牙科诊室），使人看着你所编的程序的运行结果，就知道牙科诊室发生了什么。下面我们给出一个参考解答。

```

10 DIM A(3),Q(3,20),T(3)
20 INPUT "PLEASE INPUT THEN TIME LONG";L
30 T=1:PRINT "NOW TIME","ID","BEGIN TIME","END TIME","DOCTOR"
40 X=INT(RND*5)+5:T=T+X
60 IF T>L THEN GOTO 200 ELSE PRINT T,
70 FOR I:=1 TO 3
80 A(I)=A(I)-X:GOSUB 300
90 IF A(I)<0 THEN A(I)=0
100 NEXT I:Q=0:FOR I:=1 TO 3:Q=Q+T(I):NEXT I:IF Q>=20 THEN GOTO 190
110 M=T(I):P=1
120 FOR I:=2 TO 3
130 IF T(I)<M THEN M=T(I):P=I
140 NEXT I:W=W+A(P)
150 ON P GOSUB 610,620,630
160 ID=ID+1:PRINT ID,A(P)+T,A(P)+T+T2,P
170 A(P)=A(P)+T2
180 T(P)=T(P)+1:Q(P,T(P))=A(P)
190 GOTO 40
200 PRINT " THE DOOR IS CLOSE!"
210 PRINT "THEN WAIT TIME IS ";W/ID:END
300 FOR J:=1 TO T(I)
310 Q(I,J)=Q(I,J)-X
320 NEXT J:IF Q(I,1)<=0 THEN GOSUB 400
340 RETURN
400 FOR G:=1 TO T(I)-1
410 Q(I,G)=Q(I,G)+1
420 NEXT G:IF T(I)>0 THEN T(I)=T(I)-1
430 RETURN
610 T2=14+INT(RND*5):RETURN
620 T2=19+INT(RND*5):RETURN
630 T2=20+INT(RND*5):RETURN

```

谈谈MS-DOS 5.0新增命令DOSKEY

湖北省远安县财政局 唐银红

MS-DOS 5.0 较以前版本的 DOS 相比较有了较大的改进, 诸如在内存管理、文件操作和磁盘维护、数据安全、在线帮助等方面, 还有本文介绍的新增命令 DOSKEY.

DOSKEY 是 MS-DOS 5.0 的一个外部命令, 可用于网络服务器中, 它是一个驻留内存(TSR)程序, 使用缺省参数驻留内存后, 其自身大约占用 3KB 左右的存贮器, 加上 512 字节的命令缓冲区, 约需 4KB 左右的存贮空间.

实际上, DOSKEY 是键盘输入与 MS-DOS 5.0 键盘接收信息解释程序的一个高级接口, 它是在内存中开辟一定数量的存贮器区域作为缓冲区, 用于记录自 DOSKEY 驻留内存后用户所键入的所有命令(包括用户调用的各种应用软件), 用户可以用 DOSKEY 定义的编辑键将其中的任何一条调出来, 直接重发或进行编辑修改后重发. 此外在 DOSKEY 的支持下, 用户可以在一个命令行中一次输入多条 DOS 命令, 或者创建一个命令宏定义(简称为宏), 所谓宏, 有点类似于批程序, 就是一次定义多条命令到宏体中, 要执行这些命令, 键入宏名就可以了.

一、DOSKEY 命令的格式

命令格式:

```
DOSKEY [/REINSTALL] [/BUFSIZE =size] [
  /MACROS] [/HISTORY] [/INSERT|
  OVERSTRIKE] [ MACRONAME =
  [TEXT]]
```

(其中: 方括号中的内容表示是可选项, 管道符号表示两者中只能取一种).

参数和开关:

MACRONAME 为创建的宏指定名称.

TEXT 宏中记录的 DOS 命令.

/REINSTALL 安装 DOSKEY 的拷贝到内存中, 而不管内存中是否已经安装过 DOSKEY. 如果已经安装过 DOSKEY, 该开关将清除 DOSKEY 缓冲区中的命令.

/BUFSIZE =size 设置 DOSKEY 用于保存命令和宏

的缓冲区的大小, 以字节为单位, 缺省设置是 512 字节, 最小值是 256 字节.

/MACROS 显示缓冲区中所有的 DOSKEY 宏, 可缩写为 **/M**.

/HISTORY 显示缓冲区中记录的用户键入的所有 DOS 命令, 本开关可缩写为 **/H**.

/INSERT|OVERSTRIKE 指定修改 DOSKEY 命令时的编辑方式. **/INSERT** 为插入方式, 使用此开关时, 将编辑方式设置为插入方式, 在编辑时, 就象按下了 INS 键一样, 使键入的字符插入到原文的两个字符之间. **/OVERSTRIKE** 设置覆盖编辑方式, 键入的字符替换掉原文光标处的字符, 这也是 DOSKEY 的缺省设置.

以上两种编辑方式中, 小键盘上的 INS 键仍然是这两种编辑方式的转换开关, 按一次就从一种编辑方式转换到另一种编辑方式, 直到按下回车键转换到缺省的设置.

DOSKEY 中将两种编辑方式的光标形状设计的不同, 覆盖编辑方式时, 光标与 DOS 光标一样, 插入编辑方式时, 光标高度增加一倍.

在 DOSKEY 管理期间, 可能出现这种情况: 定义的缓冲区不是太大, 而用户使用了较多的 DOS 命令, 使得缓冲区容纳不下, 在这种情况下, DOSKEY 按照队列(先进先出)的管理方式, 将最先使用的 DOS 命令从缓冲区中清除.

二、DOSKEY 命令使用的编辑键

为了能再次调用或编辑 DOSKEY 缓冲区中保存的 DOS 命令, DOSKEY 程序定义了一些编辑键, 其中有些与 DOS 命令中的编辑键相类似, 有些是重新定义的. 以下是 DOSKEY 中使用的编辑键及其功能的主要描述:

UP ARROW(↑) 调出 DOSKEY 缓冲区中最近使用的一条命令, 每按一次, 就调出前一条命令.

DOWN ARROW(↓) 调出正在显示的命令的后一条命

令，每按一次，就向后推一条。

PAGE UP(PgUp) 调出缓冲区中最早一条命令。

PAGE DOWN(PgDn) 调出缓冲区中最后一条命令。

F7 显示所有存贮在DOSKEY缓冲区中的命令及其在缓冲区中的顺序号，每行显示一条，前面是顺序号，后面是命令名，顺序号和命令之间用冒号分隔。顺序号是根据命令的执行顺序(即进入DOSKEY缓冲区的顺序)而指定的，最先执行(进入缓冲区)的就是1号。

ALT+F7 清除保存在DOSKEY缓冲区中的所有命令。

F8 在DOSKEY缓冲区中搜索以指定字符或字符串开头的命令，要使用这个编辑键，首先在DOS提示符后输入要DOSKEY搜索的命令的第一个字符或前几个字符，不区分大小写，但在DOS提示符与指定字符之间不能有任何空格，然后按F8键，DOSKEY就显示最近使用的一条与之相匹配的命令，再按一次F8键，DOSKEY继续向前搜索与之相匹配的命令，直到搜索完整个缓冲区。

F9 根据顺序号调出命令。当用户按下F9键后，DOSKEY就要求输入命令在缓冲区中的顺序号(Line number)用户输入顺序号后，DOSKEY就调出缓冲区中相应的命令，显示在当前行上。要想知道各个顺序号所表示的命令名，可按F7键列出所有命令及其顺序号。

ALT+F10 清除DOSKEY缓冲区中已定义的宏。如果要删除某个已定义的宏，用带MACRONAME=参数的DOSKEY命令。

ARROW LEFT(←) 光标向后(左)移动一个字符。

ARROW RIGHT(→) 光标向前(右)移动一个字符。

CTRL+ARROW LEFT(←) 光标移到前一个字段的一个字符上。所谓字段，就是用空格分隔的字符串。

CTRL+ARROW RIGHT(→) 光标移到下一个字段的第一个字符上。

HOME 光标移到当前编辑的DOSKEY命令的第一个字符上(行首)。

END 光标移到当前编辑的DOSKEY命令的最后一个字符后(行尾)。

CTRL+HOME 删除光标左面的所有字符，但不包括光标处的那个字符。

CTRL+END 删除光标右面的所有字符，包括光标处的那个字符。

INS 插入编辑方式和覆盖编辑方式的转换开关。

DEL 删除光标处的字符。

BACKSPACE 删除光标前的一个字符。

ESC 清除显示的命令，并将光标移到行首(DOS提示符之后)。此键与DOS命令中的编辑键的定义稍有不同，它是清除当前行上显示的所有字符，并将光标移到行首，而在DOS中，它表示使当前行作废，另起一行以重新输入命令，作废的行依然显示在显示器上，只不过在行尾加上一个反斜杠表示该行作废。

三、重新调用DOSKEY缓冲区中的命令

用以下三种方法，可以调出、编辑和重发保存在DOSKEY缓冲区中的命令。

①如果缓冲区中的命令不是很多，可用UP ARROW(↑)和DOWNARROW(↓)依次向前或向后调出各条命令。

②紧接DOS提示符后输入命令的第一个字母或前几个字符，然后按F8搜索缓冲区中最近使用的一条以指定字符或字符串开头的命令，找到后就将其显示在当前行上，如果不是所需要的命令，可按F8键继续向前搜索。

③先按F7键，将DOSKEY缓冲区中的所有命令在屏幕上显示出来，观察所要调用的命令的顺序号，然后按F9键，此时，DOSKEY就要求输入命令的顺序号，键入顺序号，就可以调出相应的命令。

例如，某用户在DOSKEY驻留内存后，已经使用了以下四条命令(可按F7键观察)：

213 ;调用2.13汉字系统

CD\CRPG ;进入报表处理子目录

CRPG ;调用汉字报表处理软件

COPY *.DBF A: ;将修改后的数据文件备份

如果按↑键，最后一条命令COPY *.DBF A:就显示在当前行上，继续按↑，就显示第三条命令CRPG，如果此时按↓键，第四条命令又出现在显示器上。按PgUp，显示第一条命令213，按PgDn键显示最后一条命令。

如果要调用第三条命令，可以连按↑键两次，或键入C(可不分大小写)，然后按两次F8键(因第一次按F8键搜索的是最近使用的一条以C开头的命令COPY)，还可以按F9键，再键入CRPG命令在缓冲区中的编号3。

无论用哪种方式调出缓冲区中的命令，都可以用前面介绍的编辑键对命令进行任意编辑。编辑时，整个命令行都显示在显示器上，编辑非常方便。无论光标在什

么位置,只要按下了回车键,命令就被重发了。

④一行输入多条命令。在 DOSKEY 的支持下,可以在一行中输入多条命令,各个命令之间用 CTRL+T 分隔,每行的字符总数,包括空格在内,不能超过 127 个字符。

例如,象下面一样将上述四条命令放在一行之内:

```
213\TCD\CRPG\TCRPG\TCOPY *.DBF A:
```

在敲入回车后四条命令就被依次执行。

⑤利用 DOSKEY 创建批程序。在 DOSKEY 中使用 /H 开关和输出重定向,能方便地创建一个批程序。例如要在 C 盘根目录下建立一个名为 CRPG.BAT 的批程序,包含以上四条命令,可按以下步骤进行:首先按 ALT+F7 键清除 DOSKEY 缓冲区中的所有命令,然后依次调用以上四条命令,全部执行完后,可按 F7 键察看,确定缓冲区中的命令是否全部都是批程序中所要求的,最后键入以下命令:

```
DOSKEY /H > C:\CRPG.BAT
```

四、DOSKEY 宏

前面,我们已经多次提到过 DOSKEY 宏,下面就详细地介绍 DOSKEY 宏。

①DOSKEY 宏中的专用字符。

以下的字符,在 DOSKEY 宏定义中具有特殊的意义:

\$G 或 \$g 这个符号等价于 DOS 中的输出重定向符号 (>),是将输出送到文件或设备中。

\$G\$G 或 \$g\$g 这个双写的符号等价于 DOS 中的“添加”输出重定向符号(>>),是将输出添加到已经存在的文件的末尾。

\$L 或 \$l 该符号等价于 DOS 中的输入重定向符号 (<),用它指定从设备或文件中读取输入。

\$B 或 \$b 等价于命令行中的管道符号(|),用于将宏输出到一条命令之中。

\$T 或 \$t 如果创建 DOSKEY 宏时的命令不止一条,则各条命令之间用 \$T 或 \$t 分隔。

\$ \$ 输入美元符号(\$)。

\$1 到 \$9 定义宏时使用的可替换参数,调用时在宏名后依次给定相应的参数。这很象批程序中的 %1 到 %9,使宏能完成不同的任务。

\$ * 定义宏中使用的可替换参数,与上面的 \$1 到 \$9 不同,\$1 到 \$9 只能代表一个可替换参数,而 \$ * 是将宏名后的所有信息作为一个参数。

②宏的创建、运行和删除

例如,要创建一个名为 QFC 的宏,完成以下任务:对驱动器 A 中的盘进行无条件(/U 开关)、快速

(/Q 开关)格式化,然后用 SYS 命令将系统传送到 A 盘中,最后将 C 盘 DOS 目录下的所有扩展名为 .EXE 的文件拷贝到 A 盘中。可用以下命令来创建这个宏:

```
DOSKEY QFC=FORMAT A:/Q/U$ TSYS  
A:$TCOPY C:\DOS\*.EXE A:
```

每用一条带 MACRONAME=TEXT 参数的 DOSKEY 命令,就创建了一个宏。这些宏被依次存放在 DOSKEY 缓冲区中,直到被删除为止。

要调用已定义的宏,键入宏名即可。例如,在光标处打入:

```
QFC
```

则 QFC 宏被调用,完成对 A 盘的格式化、系统传送和文件拷贝任务。

因为设置的 DOSKEY 缓冲区数量有限,所以应及时删除无用的宏。删除一个宏,就是定义该宏时,仅有宏名,而不赋给宏任何内容,就象下面一样,将 QFC 宏从缓冲区中清除。

```
DOSKEY QFC=
```

这就是说,对已经存在的宏,可以随时重新进行定义。

如果要删除所有的已经定义的宏,按 ALT+F10 键。

③在宏中使用可替换参数

象前面定义的 QFC 宏中,有时感觉使用不方便,例如要对 B 盘做相同的工作,就不能用 QFC 宏,还有一种可能,如果 A 为 1.2MB 的驱动器,要格式化低密盘时,必须带 /4 开关,如果在定义宏时使用了可替换参数 \$1 到 \$9,就可以完成诸如上述的操作,使用时就感觉到非常方便。就象下面一样:

```
DOSKEY QF=FORMAT $1 /U/Q$ 2$  
TSYS $1$TCOPY C:\DOS\*.EXE $1
```

如果是对 B 盘进行上述操作,键入下面的命令:

```
QF B:
```

如果是格式化 A 驱(1.2MB)中的 360KB 的软盘,并进行相应的操作,使用以下命令格式:

```
QF A: /4
```

有时需要替换的参数不能确定时,用 \$ *。

例如,用列目录命令 DIR 观看 C 盘 DTP 目录下的 SAMPLES 子目录下的扩展名为 .DOC 的文件,分别用不带开关、带 /P/W 开关和 /P/W/OD/L 开关,则可用下面的命令定义名为 DDIR 的宏:

```
DOSKEY DDIR=DIR $ * C:\DTP\SAM-  
PLES\*.DOC
```

在调用时,参数跟在宏名之后:

DDIR 相当于 DIR C:\DTP\SAMPLES\
*.DOC

DDIR /P/W 相当于 DIR /P/W C:\DTP\
SAMPLES*.DOC

DDIR /P/W/OD/L 相当于 DIR /P/W/
OD/L C:\DTP\SAMPLES*.DOC

④宏名与DOS命令同名

定义宏时,允许将宏名取得与DOS命令名相同。以下规则使系统能够识别调用的是DOSKEY宏,还是DOS命令:

调用宏时,必须在DOS提示符(>)之后,立即输入宏名,在宏名与DOS提示符之间不能有任何空格。

调用DOS命令时,在DOS提示符后输入一个或多个空格,然后再输入DOS的命令名。

⑤利用批程序保存宏

由于宏是保存在DOSKEY缓冲区中,也就是说它位于存储器中,因此,也就具有易失性,一是重新引导系统时,二是用带/REINSTALL开关的DOSKEY命令时,三是按ALT+F10时,都将清除所有的宏。如果这些宏还有用,就必须重新定义,这就显得有些麻烦。此时可将缓冲区中的宏,输出到一个批程序中保存起来,必要时利用批程序来定义宏。例如在前面已经定义了QFC、QF和DDIR三个宏,利用/M开关和输出重定向,创建一个名为MACROS.BAT的批程序:

```
DOSKEY /M > MACROS.BAT
```

此时,MACROS.BAT文件中包括以下三条语句:

```
QFC=FORMAT A: /Q /U $TSYS A: $  
TCOPY C:\DOS\*.EXE
```

```
QF=FORMAT $1/ Q/ U $2 $TSYS $1 $  
TCOPY C:\DOS\*.EXE $1
```

```
DDIR=DIR $ * C:\DTP\SAMPLES\  
*.DOC
```

用各种文本编辑软件(象WS等)在每条语句前面加上DOSKEY,当编辑好后,每次调用MACROS.BAT批程序时,就定义了QFC、QF和DDIR三个宏。

注意:不能在一个宏中调用另一个宏,也不能在一个批程序中调用一个宏。

五、宏与批程序的区别

宏和批程序很相似,当输入其名称时,就调用了它们之中所包含的命令集合,但它们之间还是有很大的区别:

①宏保存在内存中,可以从任何盘、任何目录中调用,且速度相当快;而批程序保存在磁盘上,调用时必须

指明它存放的路径,否则,DOS就找不到它,并且批程序总是从磁盘中调到内存,因而执行的速度慢于宏。

②在定义宏时,将所有的命令放于一行之中,各命令之间用\$T分隔,而且字符总和不能超过127个;而在批程序中,一条命令单独占用一行,且对命令的个数没有限制。

③宏和批程序都对CTRL+C(CTRL+ BREAK)非常敏感。在宏中,按CTRL+C只能中断正在执行的那条命令,而且是不等用户确认,转而去执行下一条命令。如果一个宏中定义了三条命令,就得按三次CTRL+C才能完全中断这个宏,如果按CTRL+C的次数少于宏中定义的命令个数,那么,下个余留的命令将被立即执行;而在批程序中,只要用户按下了CTRL+C,批程序就会停下来,询问用户是否确实要退出批程序,如果用户确认,就停止整个批程序,而不管还余下多少个命令还没有被执行。

④虽然两者都能使用可替换参数和重定向符号,但二者的表示方式不同。宏中的可替换参数是\$1--\$9,并使用专用重定向符号\$G、\$G\$G和\$L;而在批程序中的可替换参数是%1--%9,并使用DOS中的重定向符号>、>>和<。

⑤宏中不能使用GOTO命令,也就是说,宏只能完成一些较为简单的任务,而要完成比较复杂的任务,还必须用批程序。

⑥可以在宏中定义环境变量,但不能在宏中使用环境变量。例如设定ABC为C:\DOS目录(SET ABC=C:\DOS),但不能在宏中用ABC代替C:\DOS,象DIR ABC在宏中是错误的;然而,在批程序中使用所有环境变量。

备注:

①在DOSKEY中,只能将用户从键盘中输入的信息保存到缓冲区中,如果用户调用一个批程序,DOS将顺序地执行批程序中的每条命令,而不将这些命令保存到缓冲区中。

②DOSKEY命令行与DOS命令行的区别:DOS的命令行只能显示光标以前的字符,而DOSKEY是将整条命令都显示在当前行上,这对编辑修改非常方便。

③本文所说的命令,是广义的DOS命令,意即所有扩展名为.COM、.EXE和.BAT的可执行文件。

④如果DOSKEY用缺省的参数驻留内存,则缓冲区为512字节,大约可以记录35条命令。

⑤当用户键入CTRL+T时,在屏幕上显示的是符号qT,本文中用^T表示。

★ 新一代驱动器——光磁软盘系统

福建省尤溪第一中学 陈明忠

硬盘(Fixed-Disk)存储容量大,传输和存储速率高,是计算机重要的存储部件之一,但它相对固定,软件交流不可能借助硬盘。软盘驱动器(Floppy-Disk)的存储介质便宜,软盘又可灵活装卸,便于携带和交换,所以它是计算机特别是工作站、微机配置中不可缺少的部件。但它的容量小存储速度慢,例如 Windows 3.1 需要 10 多张 1.2M 软盘, WPS 系统也需要 6 张 1.2M 软盘,这样大的数据量在安装或拷贝时,要占用较长的机时。刚出现的光盘系统有许多优点,但由于硬件的要求,使得光盘系统的记录头份量重,体积大。所以寻道时间长,存储速度低,这也是它光盘一时无法代替软硬盘系统的重要原因。随着计算机软硬件系统的飞速发展,现有的软硬盘系统已不能满足要求。这就出现了新一代的光磁软盘系统,它是在现有的软盘技术基础上扩大容量,提高存储速度,并采用光定位寻道等新技术,所以它不仅容量大而且存储速度快。

要提高容量就要提高位密度和磁道密度,这在磁介质上是容易做到的。但提高位密度后,由于磁头和介质之间读写时有大量的灰尘,更容易造成磨损和误码,光磁软盘系统采用了 ECC 校正技术,将可纠错突发码为 80 位,是硬盘的 8 倍,这样盘片上允许的可纠正突发码为 70 个,充分保证了读写可靠。

标准光磁软盘系统的道密度为 1245TPI,现在的软驱采用步进电机寻道定位,精度低,不能满足要求,光磁系统采用光伺服定位系统。在磁介质上每两条 17.27 微米宽的磁道间压上 4.57 微米宽的光伺服道,它的伺服信号不会被磁化或磨损而丢失。工作时,由红外发光二极管发射的光束通过读写磁头中间小孔,射到压制在介质上的光伺服道上,从光伺服道反射的信号又通过磁头中间的孔,经过透镜,光检测器等,将光信号转成电信号来控制伺服电机,实现精确定位。

为了提高数据传输率,光磁软盘系统和主机的接口采用了 SCSI(Small Computer System Interface)系统接口级,这一接口技术以前主要用在小型机,工作站,网络服务器上,被称为“智能卡”,采用该接口后,读写速度比普通软驱快 1 倍,格式化速度快 10 倍,明显地提高了读写速度。

由于光磁软盘系统卓越的性能,在美国 Micro 标准委员会的推荐下,由光磁软盘系统发明者 insite 公司和

磁介质生产者 3M, Maxell 以及 SCSI 主适配器生产厂等组成了光磁技术协会,并自己制定出工业标准—21MB 光磁软盘系统。该系统非格式化容量 24.7MB,格式化容量 21MB,每扇区容量为 1KB,柱面 753 个,读写磁头 2 个。磁介质尺寸为 3.50 英寸,选择钕铁为介质。软盘驱动器采用+5V 电源供电,最大功耗 6W,外型尺寸为 101X150X254MM³。由于 3M 和 Maxell 加入光磁技术协会,以他们超群的磁介质生产技术,生产规模及全世界的经销网以及他们支持光磁软盘技术承诺,为光磁技术的应用提供了可靠的保证,更重要的是参与制造光磁系统的厂家,如 MKE, IOMDGA 这些公司都具世界级的生产能力和卓越的技术实力,可充分满足全世界对光磁驱动器日益增长的需要。

光软盘相当于一个移动的硬盘,它在各方面都是有广泛的应用前景。一些大型软件一张盘就足以容纳,不必装在几张软盘上,不仅携带安装方便而且这些软件在没有硬盘支持时可直接在光软盘上运行。一张 21MB 的光磁软盘相当于 14 张 1.44MB 软盘,在一张光磁软盘上可存放大量软件,而不必都存放在硬盘上,这就免除了受破坏和病毒感染之苦,也不必担忧失窃和泄密。随着笔记本计算机的发展,大容量的光磁软驱完全可代替硬盘,使它只要携带一个光磁软盘就足够了。特别是在数据量大的方面,如图像处理,电子出版,动画制作,多媒体技术等方面更有应用价值。而且和其它存储介质相比光磁盘的成本更低,国内目一张光磁软盘约二百多元。

特别值的一提的是,光磁软盘系统在设计上顾及了向下兼容,它的读写磁头是有两个铁芯的双隙磁头,既能读写 21MB 光软盘,又能读写标准 720KB 双密和 1.44MB 高密软盘,这种特殊的双隙读写头,其中一隙用于光磁软盘的读写,另一隙用于标准容量 3.5 英寸软盘的读写,这对用户来说是至关重要的,不必像其它微机一样,在一台机器上配置两种型号的软驱,而且在采用光磁软驱后可照样读写原有 3.5 英寸盘不会造成资源的浪费。随着光磁软盘技术的发展,光磁软盘的容量将得到大大的提高,成本也会大幅度降低,相信不久光磁软盘将代替现有的软盘驱动器。

波斯王子二代 攻略秘诀

华南师范大学计算机系九〇级 梁宇种

相信大家都玩过波斯王子一代了吧。大家一定会被其优秀的画面、机关所吸引。然而第二代却更为强劲，堪称 PC GAME 经典之作！它不仅在音效、画面、机关上大进了一步，而且让玩家如若亲临其境，深深体会到“山穷水尽疑无路，柳暗花明又一村”的意境，其新鲜感层出不穷！

下面，我给大家介绍一下该游戏的攻略秘诀：

二、控制方法（小键盘为主）

1、移动

左右跑动：4（左箭）、6（右箭）

上跳：8（上箭）（要一直按住）

下蹲：2（下箭）

左右跳动：7、9

左右蹲行：1、3

小步、抓：del 或 shift（要一直按住）

2、战斗

出剑：0

格挡：8（上箭）

进/退格挡：7、9

前后移动：4（左箭）、6（右箭）

攻击：0 或 CTRL

收剑：2（下箭）或 5

3、控制功能

功能选择：Alt-O

龙虎榜：Alt-H

音乐开关：Alt-M

音效开关：Alt-S

跳关：Alt-N（前四关有效）

重玩游戏：Alt-A

新游戏：Alt-R

读取进度：Alt-L

记录进度：Alt-G

返回 DOS：Alt-Q 或 Ctrl-Q

二、过关攻略

第一关：到达地面一直向左跑，跑到码头尽头时跳，抓（shift），再按 8 上船。

第二关：经过流沙地时不要踏到有标记的地板。

第三关：跳三下使震掉的石块把门的开关打开。

第四关：找到出口后向右走到尽头，踏下开关，拼命往回跑。

第五关：坐飞毯。

第六~第十三关：主要考察你的跳跃水平。小心铡刀，过铡刀时用 1、3 爬过。

第九关：从顶层走，到尽头时往下跳骑上马背。

第十三关：骑飞马。

第十四关：关底。

三、使用绝招

1、元神出入

从第十关到第十四关都可用此法。元神出窍后可自由穿透闸门，从高处掉下、被铡刀砍都不扣生命点，敌人不能发现你的存在但你可用刀砍他，同时能在某些松动的地板上站稳，在第十四关“水晶宫”上方还能发射“气功”等。但使用这种绝招要耗费一些生命点。

元神出窍：把王子站在原地，然后不停按左右键使王子两边转动。

元神入窍：把王子的元神站在倒下的肉身上，按 1、3 键转动。

2、特殊模式

启动波斯王子时键入 PRINCE MAKINIT 即可进入该模式。在这种模式下具有如下功能：

K 无敌，杀死画面上所有敌人。

R 复活

+增加时间

-减少时间

F1 慢动作开关。这个开关是跳跃水平低的玩家之至宝！

F6 显示位置尺。能给你带来许多方便。

Alt-N 过关，不受限制。

shift-T 增加能量。

shift-K 减少能量。

3、修改游戏

①生命无限法

用 PCTOOLS 对 PRINCE.EXE 查找 88 0E 92 5C C6 46，将其改成 90 90 90 90 C6 46 即可。但王子落下深渊、掉入熔岩、被铡刀砍、被蛇咬到还是会死。

②天下无敌法

用 PCTOOLS 对 PRINCE.EXE 查找串 3B C8 7E 02 8B C8 88 4C 12，将其改为 31 C9 90 90 90 90 88 4C 12。遇到敌人，要么是已经死的，要么只用一刀。

③时间不变法

用 PCTOOLS 对 PRINCE.EXE 查找串 FF 0E 2A 5E 74 40，将其改为 90 90 90 90 74 40 即可。



游戏数据存放

福建省福州大学 陈 涛

三国志游戏软盘是由 KOEI.CO 设计开发的智力型游戏软盘，汉字命令指示明了清楚，存盘文件自己定义，字节为 30752 存于软盘或硬盘。

游戏中黄金数量很重要，用于征兵、购物、奖赏、开垦、作战等，笔者通过 comp 和 Debug 软件的多次运行观察，发现其数据存放规律。

三国志领地分为 41 郡，有若干诸侯，每郡有其固定地址，数据以十六进制数存放。

第一郡，黄金 (2E94, 2E95) 粮食 (2E96~2E98) 人民忠诚度 (2EA3) 地价 (2EA4) 治水度 (2EA5) 良驹 (2EA6) 米价 (2EA7)

第二郡，黄金 (2EB7, 2EB8) 粮食 (2EB9~2EBB) 人民忠诚度 (2EC6) 地价 (2EC7) 治水度 (2EC8) 良驹 (2EC9) 米价 (2ECA)

照此推算，每郡相差 35 个偏移量递增。

例如，第五郡地价数据存处：

$$2EC7_{(16)} + (5-2)_{(16)} \times 35_{(10)} \\ = 2EC7_{(16)} + 3 \times 23_{(16)} = 2F30_{(16)}$$

诸侯信任度位于诸侯所在郡，第五郡为 2E69，推算同上。

这样，在游戏中，用软件适当修改存盘文件中的一些数据，更加速三国统一进程，增添几分乐趣，但切记不要贪心，以免爆机。

344

《神州八剑》游戏攻略心得

无锡总参工程兵科研一所 曹宇杰

(1) 第三关 (转魂国) 有一件攻打“统天地府”的必要宝物——“魔法登山钩” (“统天地府”在第一关)。“魔法登山钩”埋藏在左面大海中的一个小岛上的树林中。因此，一定要先到第六关去取到能任意穿越树林的宝物“森林神像”再回到第三关来才能取到“魔法登山钩” (“森林神像”在第六关右下角的两小岛之间的大海)。

(2) 由于在每个月结束时你需要付给部队佣金，因此在组成一支大军后使用停时术是一个省钱的办法 (军队人少的时候，买停时术的钱可能会比佣金还要多)。停时术的使用方法是：在走动状态下，选择使用法术停时术，这时屏幕下方表示时间的红线变成绿线，其缩短的速度变为原来的三分之一 (缩完一次表示一个月结束)。

(3) 涉水术是获取“森林神像”的必备法术，而且第七关是由许多小岛组成。因此，在第五、第六应多储备一些涉水术。

本人曾把此游戏从头至尾玩通过三遍，总结出一些小窍门，一并献上。

(1) 在天涯海角取得的钱财要全部分发给部下以增加领导能力。

(2) 在第一关一定要先取得“魔法竹筒”再攻城，因为有了“魔法竹筒”以后，每攻克一座城池便能使魔法空位增加。

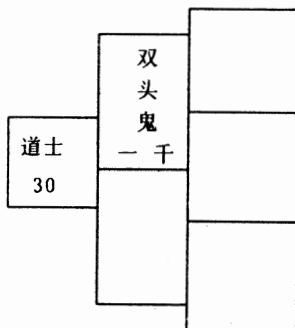
(3) 在战斗时应注意“各个击破法”，将敌人打到只剩

下几个时，虽然他们还活着，但是已经没有攻击力，这时就应转而攻击其他敌人。

(4) 一定要快速攻城过关，有些不必要的宝物可以以后再拿。

(5) 只有道士和双头鬼会使用法术，道士只有第七关才能招募，双头鬼在第三、第四关可以招募，并且都有一定的数量，道士约 70 人，双头鬼约 1300 人，所以一定要尽可能的保护他们。

(6) 最后一个秘诀：千万不要买“地震术”。因为两点，一、人数越多法术威力越大；二、“雷火术”可以对所有的敌人同时攻击。因此，采取如下队型可保持常胜。



首先，道士使用“凝滞术”使对手无还手之力，然后双头鬼使用“雷火术”。

345

神秘的“1991”病毒

正大电脑应用与数据修复有限公司 胡向东

最近,连续有四个单位的电脑受到被笔者称之为“1991”病毒的感染,用目前国内收集到的 SCAN 软件,根本无法检测出来,使人有一种神秘的感觉。

该病毒是一个系统型的病毒,因其以“1991”为病毒标记,故笔者取其名为“1991”病毒。

该病毒的神秘之处表现在如下几方面:

(1) 一般的硬盘主引导程序的前面几条语句是:(设首语句放在 XXXX: 1000 处)

```
XXXX: 1000 CLI
XXXX: 1001 XOR AX, AX
XXXX: 1003 MOV SS, AX
XXXX: 1005 MOV SP, 7C00
```

而“1991”病毒的前面四条语句与正常的硬盘主引导程序一模一样,因此,即使把带病毒的硬盘主引导区调入内存去观察,也极易被该病毒蒙骗过去。

(2) 一般的系统型病毒都占据内存高端位置,因而病毒程序都免不了要改动 0: 0413H 的值,使电脑的可用内存减少若干 K 字节(一般是 1K 至 2K。从而有经验的用户可以用 PCTOOLS 等工具检测到内存值比系统加电时自检到的内存值要少,从而容易暴露。

而“1991”病毒却驻留在内存的低端,驻留在段地址为 0000 的地方(病毒首语句驻留在 0000: 0540 处),因此该病毒一反常态,没有更改 0: 0413H 中的值,因此也难以用内存值来判断病毒的存在,使这种病毒更带有一种神秘的色彩。

(3) 一般的系统型病毒,病毒驻留内存时都改写了 int 13H 的入口地址,因此在中断向量表 0: 4CH 的 int13H 入口地址中,很容易看到系统型病的病毒磁盘中断程序。

而“1991”病毒驻留内存后,从中断向量表中看

不出磁盘中断 int13H 的入口地址有什么改变,因此即使用一个不带病毒的系统盘启动来观察中断向量表,通过对比比较也发现不出有什么破绽。更增加了这种病毒的神秘感。

磁盘中断 int13H 一般是由最底层的 ROMBIOS 和 IBMBIOS 这两部分组成,ROMBIOS 的内存入口段地址在 F000, IBMBIOS 的内存入口段地址在 0070。

在一般情况下,无毒的电脑内存 0070: 00B4 中,存放着 ROMBIOS 的 int13H 调用程序的入口地址: F000: A1D1,而入口地址在 0070: 1F51 的 int13H 中断服务程序在运行到

```
0070: 104A CS:
0070: 105A CALL FAR (00B4) 时
```

会自动转入到 ROMBIOS 的 F000: A1D1 程序去执行。

而“1991”病毒却巧妙地偷梁换柱,把 0070: 00B4 的跳转指针,指向“1991”病毒的磁盘中断入口 0000: 05D4,从而达到既起到传染作用,又不易暴露的目的。

(4) “1991”病毒全长只有 132H 字节,体积较少,特别是它和其它系统型病毒或混合型病毒,尤其是和一些未知名的新病毒交叉感染时,更使人感到神秘。

受到这种病毒感染的电脑,一般的症状是磁盘读写速度放慢,软盘放进带毒的机器中读写都会受感染,用带毒的软盘启动机器(不管放的是带毒的系统盘或是带毒的数据盘),机器中的硬盘都会被感染。

由于“1991”病毒的磁盘中断不更改 0: 0413H 的值,因此在实际运行中该病毒盘中断程序常常不受 DOS 保护而受到破坏,造成硬盘数据或软盘数

据被乱写。笔者接触过感染“1991”病毒的四个单位中，就有二个单位的硬盘的 FAT 表和目录表受“1991”病毒所破坏。

识别软盘引导扇区的“1991”病毒并不难，用 DEBUG，执行

```
L 0100 0 0 1 (把 A 盘零扇区调入内存 100H)
```

这时若在内存 100H 处看到

```
XXXX: 0100 CLI
XXXX: 0101 XOR AX, AX
XXXX: 0103 MOV SS, AX
XXXX: 0105 MOV SP, 7C00
XXXX: 010B MOV DS, AX
XXXX: 010A STI
```

就可以判断该软盘一定是带病毒的，因为常见的 DOS 版本格式化后的软盘，引导扇区的首语一定是 JMP XXXX（常见的有 JMP 013E, JMP 012A, JMP 012C, JMP 0136, JMP 0140 等等）。

再加上在 XXXX: 022CH 中有机器码 91H, 19H，即病毒标记 1991，则可肯定该软盘有“1991”病毒。

“1991”病毒把原软盘引导扇区，搬到 0 道 1 面 3 区中，

故消除软盘的“1991”病毒，通常有二种方法，方法一是用 DEBUG 把同类型无毒的软盘的引导区读入内存

```
L 0100 0 0 1
```

然后把有毒的软盘放在进 A 驱动器中，再执行 DEBUG 的写盘命令

```
W 0100 0 0 1
```

软盘解毒的方法二是用 DEBUG 写如下程序

```
XXXX: 0100 MOV AX, 0201
XXXX: 0103 MOV BX, 1000
XXXX: 0106 MOV CX, 0003
XXXX: 0109 MOV DX, 0100
XXXX: 010C Int 13
XXXX: 010E Int 3
```

然后把带毒软盘放入 A 驱动器，再执行 G 命令，（即键入 G）

此时观察一下调入内存 1000 处的程序，看是否为正确的引导程序，若是，则执行 DEBUG 的写盘命令：

```
W 1000 0 0 1
```

即可达到解除软盘中的“1991”病毒的目的。

识别硬盘中的“1991”的病毒，可用 DEBUG 执

行如下程序，把硬盘的主引导扇区读入内存 1000H 处

```
XXXX: 0100 MOV AX, 0201
XXXX: 0103 MOV BX, 1000
XXXX: 0106 MOV CX, 0001
XXXX: 0109 MOV DX, 0080
XXXX: 010C Int 13
XXXX: 010E Int 3
```

若此时在 XXXX: 112C 处发现病毒标记 1991，则可确认该硬盘已感染 1991 病毒。

清除硬盘中的“1991”病毒，类似于其它系统型病毒的消除方法。

可以用上述的程序把任一个不带病毒的硬盘主引导扇区读入内存 1000 处。

然后用一个文件存起来（不妨设为 AT2 文件）

接下来是用一个不带毒的系统盘启动硬盘有病毒的机器，再用上述程序把带毒硬盘的主引导扇区读入内存 1000H 处，

然后用 DEBUG 的读文件命令把文件 AT2，读入内存 2000H 处，

```
n AT2
```

```
L 2000
```

再就是把原硬盘的分区参数表（11BE 至 11FF）移至 21BE 至 21FF，

```
M 11BE 11FF 21BE
```

最后执行以下程序

```
XXXX: 0100 MOV AX, 0301
XXXX: 0103 MOV BX, 2000
XXXX: 0106 MOV CX, C001
XXXX: 0109 MOV DX, 0080
XXXX: 010C Int 13
XXXX: 010E Int 3
```

即可把不带毒的硬盘主引导程序和原来的硬盘分区参数表写回到硬盘主引导扇区中。

值得注意的是，一定要保留带毒硬盘的分区参数表，否则容易出乱子！

总之，魔高一尺，道高一丈，目前笔者已编写了检测并清除“1991”病毒的消毒软件。需要咨询及进行数据修复的可直接与本人联系，正大电脑应用与数据修复有限公司的邮码是 510630，地址是：广州天河龙口西路 1-38 号，电话：5515961-5931，BB 机 3322322-4364。

一种新型的计算病毒—V300E

广东花都市 颜岳军

V300E 病毒是一种文件型病毒，被感染后的文件，文件长度增加 300E 字节，故取名 V300E。用目前流行的 CPAV 消毒软件检测不出来。该病毒修改 DOS 中断 INT 21H，在文件的执行过程中感染该文件，并判断是否发作。该病毒发作时，用当前内存的内容覆盖硬盘的前 64 个扇区，从而导致引导失败，丢失的数据几乎不可能恢复。而且发作的条件容易满足。加上每感染一个文件要占用 12K 多的内存（实际可执行部分不足 1550 字节），所以该病毒的危害性比较大，是一种恶性病毒。

正如其它文件型病毒一样，V300E 病毒在执行过程中也修改了单步中断 INT1H 和 INT3H 的中断向量入口，用来反用户跟踪。除此之外病毒程序中还有一段指令代码是经过加密的，须在程序的动态执行过程中解密后才能执行。再加上一些堆栈的灵活使用，对该病毒的分析有一定的难度。

一、病毒的检测

对该病毒的检测，如果借助 PCTOOLS 和 DEBUG 等工具是不难的，如果有以下特征：

1. 用 DIR 命令发现可执行文件增长了 12K 字节。

2. 用 DEBUG 查看中断向量表 INT 21H 入口偏移值为 02CB。

3. 用 PCTOOLS 在执行文件中可找到 EB 38 90 31 32 33 34 35 36 37 32 等字节。

可以判断系统已经感染了 V300E 病毒。

另外如果出现系统死机，在屏幕当前光标位置出现“TOO MANY WRITE PROTECT!!!”字样并响铃警告，则是病毒的发作。

二、病毒的消除

和诸多文件型病毒的消除一样，消除该病毒可遵循如下步骤：

1. 恢复文件被修改的内容，也就是 EXE 文件的文件头，.COM 文件的起始几条指令。

2. 修改被感染文件，也就是删除病毒附加在原文件上的 12K 字节长的代码。对该病毒的消除，提供以下数据：

1) .COM 文件，原文件前面 20 个字节内容保存在病毒代码部分的 01A~03AH，只要找到病毒代码的起始部分把相对于起始值 01A~03AH 的内容写入文件

的 00~020H 字节，就恢复了 .COM 文件被修改的指令。然后将感染后的文件减去 300EH 字节，写入 .COM 文件就消除了该病毒。

2) .EXE 文件，.EXE 文件的文件头的恢复则比较麻烦。原文件头的四个数据：堆栈模块相对于加载时的段地址的段偏移量；当对装入模块给出控制时的 SP 值；当对装入模块给出控制时的 IP 值和装入模块中代码段的位移量。也就是原文件头的 0E~0FH；10~11H；14~15H 和 16~17H 八字节的内容。它们分别保存在病毒部分的第 04A~04BH；048~049H；044~045H 和 046~047H 八字节中。而对于有关 EXE 文件长度的信息，即文件头的 02~05H 四字节的值，可由以下公式计算得到：

文件长度 MOD 512 和 (文件长度-1) / 512

如果余数为 0 (文件长度 MOD 512=0)，则 02~03H 的值为 200H，04~05H 的值为：(文件长度-1) / 512。否则 02~03H 的值为：文件长度 MOD 512。04~05H 的值为 (文件长度-1) / 512+1。具体用手工搬除方法来消除该病毒就不作介绍，下面给出用 Turbo C 2.0 编写的消毒程序。该程序在 AST 286/386/486 以及 SZ 386 上调试通过，能消除当前目录下文件感染的病毒。顺便提一句的是：由于该病毒是在文件执行时感染的，故用修改文件后缀的方法来达到免疫的目的是不可能的。

```
/*kill.c*/
#include<conio.h>
#include<dos.h>
#include<dir.h>
#include<stdio.h>
#include<bios.h>
FILE *fopen(),*fp;
unsigned char *buf1;
main()
{
    unsigned int now=len,now=drive,s=drive,i,yy;
    buf1=(char*)malloc(1024*sizeof(char));
    killv300e();
}
killv300e()
{
    register int done;
    struct tblk f;
    done=findfirst("*.com",&f,0);
    while(! done)
    {
```

```

killcom(f.ff-name, f.ff-fsize);
done=findnext(&f);
}
done=findfirst("*.exe", &f, 0);
while(! done)
{
    killexe(f.ff-name, f.ff-fsize);
    done=findnext(&f);
}
nion REGS regs;
killcom(filename, filelen)
    long filelen;
    unsigned char filename[15];
{
    long int i;
    unsigned int ch, ch1, ch2;
    if(fp=fopen(filename, "rb+"))
    {
        i=filelen;
        i-=0x300e;
        filelen=i;
        fseek(fp, i, 0);
        ch=fgetc(fp);
        ch1=fgetc(fp);
        ch2=fgetc(fp);
        if((ch==0xeb) && (ch1==0x58) && (ch2==0x90))
        {
            fseek(fp, i+0x1a, 0);
            for(i=0; i<32; i++)
                buf1[i]=fgetc(fp);
            fseek(fp, 0, 0);
            for(i=0; i<32; i++)
                fputc(buf1[i], fp);
            fclose(fp);
            op=file(filename, filelen);
            printf("%c%c%c%SFOUNT v300e virus And killed", 7, 7, filelen,
me);
        } else fclose(fp);
        } else
        {
            printf("%c%c%c%SCNOT OPEN THE FILE\n", 7, 7);
        }
    }
    killexe(filename, filelen)
        unsigned char filename[15];
        long filelen;
    {
        long int i, j;
        unsigned int ch, ch1, ch2;
        if(fp=fopen(filename, "rb+"))
        {
            filelen-=0x300e;
            j=filelen;
            fseek(fp, j, 0);
            ch=fgetc(fp);
            ch1=fgetc(fp);
            ch2=fgetc(fp);
            if((ch==0xeb) && (ch1==0x58) && (ch2==0x90))
            {
                fseek(fp, i, 0);
                fseek(fp, 0x1a, 1);
                for(i=0; i<64; i++)
                    buf1[i]=fgetc(fp);
                buf1[22]=buf1[42];

```

```

                buf1[23]=buf1[43];
                buf1[20]=buf1[44];
                buf1[21]=buf1[45];
                buf1[14]=buf1[48];
                buf1[15]=buf1[49];
                buf1[16]=buf1[46];
                buf1[17]=buf1[47];
                i=j;
                j--;
                i=i%512;
                j=j/512;
                if(i==0) {
                    buf1[2]=00;
                    buf1[3]=02;
                    buf1[4]=j&0xff;
                    buf1[5]=j>>8;
                } else {j++;
                    buf1[2]=i&0xff;
                    buf1[3]=i>>8;
                    buf1[4]=j&0xff;
                    buf1[5]=j>>8;
                }
                fseek(fp, 0, 0);
                for(i=0; i<32; i++)
                    fputc(buf1[i], fp);
                fclose(fp);
                op=file(filename, filelen);
                printf("%c%c%c%SFOUNT 300E Virus And killed", 7, 7, filelen,
me);
            } else fclose(fp);
            } else
            {
                printf("%c%c%c%SCNOT OPEN THE FILE\n", 7, 7);
            }
        }
        op=file(fname, flen)
        unsigned char fname[30];
        long int fien;
    {
        int inhandle;
        char buf3[100];
        regs.x.dx=(int) &fname[0];
        regs.b.al=2;
        regs.b.ab=0x3d;
        intdos(&regs, &regs); /*打开文件*/
        inhandle=regs.x.ax;
        regs.x.ax=0x4200;
        regs.x.cx=0;
        regs.x.dx=0;
        regs.x.dx=inhandle;
        intdos(&regs, &regs);
        regs.x.ax=0x4201;
        regs.x.bx=inhandle;
        regs.x.cx=flen>>16;
        regs.x.dx=flen&0xffff;
        intdos(&regs, &regs); /*移动文件指针到文件尾*/
        regs.b.ab=0x40;
        regs.x.cx=0;
        regs.x.dx=(int) &buf3[0];
        regs.x.bx=inhandle;
        intdos(&regs, &regs); /*写文件*/
        regs.b.ab=0x3e;
        regs.x.bx=inhandle;
        intdos(&regs, &regs); /*关闭文件*/
    }

```


单片微机与模糊控制讲座

第五讲 自组织模糊控制方法

广东工学院 余永权

模糊控制中,控制机理是由人们对生产过程或对象的模糊信息归纳后产生的控制规则确定的。但是,人们在对生产过程或对象进行经验总结时,由于认识的偏差,总结的方法不合适等因素的影响,得到的经验是不十分准确或不完善的,从而导致模糊控制规则较为粗糙;最终引起控制的效果不随人意。另外,由于生产过程或对象是随着外界条件和本身的条件而有所变化的,纵使原来总结的经验十分完善,也不可能总是符合外界和过程在新条件新状态下的情况。这两种情况促使人们考虑一种可以在运行中能自动对本身的参数进行调整,使系统的性能不断改善,以适应不断变化的情况,保持控制达到所希望的效果。

能够对模糊控制规则进行自动调整的控制方法称为自组织模糊控制 SOFC。

在这一讲中介绍自组织模糊控制的原理和方法。

5.1 自组织模糊控制的基本思想

自组织模糊控制是使系统的性能不断改善,直到系统的输出达到预期的精度的自调整控制系统。自组织模糊控制的功能包括下列两种:

- 1、对系统的品质进行辨识;
- 2、对系统执行模糊控制。

一个自组织模糊控制器是在通常的模糊控制器上增加了几种功能部件,它们分别是性能测量,控制量校正和控制规则修正。下面分别对这些功能部件作简要介绍。

5.1.1 性能测量

性能测量是对系统的输出特性进行测量,并将其和预定的特性进行比较,以判别实际输出和预期输出的偏差。在模糊控制中,一般取“偏差”和“偏差变化率”两个参数来衡量实际输出与预期输出的偏离情况。

性能测量的结果给出对输出特性的校正量 P_0 。在模糊控制中,性能测量后的输出校正量是用校正表给出的。典型校正表如表 5-1 所示。

在表 5-1 中,当偏差 c 处于 -2 处,而偏差变化率 Δc 趋向给定值,则不加校正量;当偏差变化率 Δc 离

开给定值并为 1 时,则校正量 $p = -2$; 其余同理。表

5-1 输出特性的校正表

P	Δc	趋向给定值				离开给定值		
		-3	-2	-1	0	1	2	3
c	大于给定值	-3	-2	-2	-3	-4	-4	-4
		-2	0	0	0	-2	-2	-3
		-1	0	0	0	-1	-1	-2
		0	0	0	0	0	0	0
小于给定值	1	0	0	0	1	1	1	2
	2	0	0	0	1	1	1	3
	3	2	2	3	4	4	4	4

5.1.2 控制量校正

控制量校正就是根据性能测量时所得到的输出校正量 P , 再求出相应的控制量校正量 r 。

对于单输入单输出的系统,一般取:

$$r = p \quad (5-1)$$

5.1.3 控制规则修正

根据求得的控制校正量 R , 则可以得到实时的控制量 V :

$$V = u + r \quad (5-2)$$

其中: u 是未校正的控制量;

r 是控制校正量。

模糊控制的一般规则为:

IF $c = \underline{A}_i$ AND $\Delta c = \underline{B}_j$ THEN $u = \underline{C}_{ij}$

采用了控制校正量 r 之后的控制规则为:

IF $c = \underline{A}_i$ AND $\Delta c = \underline{B}_j$ THEN $v_0 = \underline{D}_{ij}$

从而有新的模糊关系:

$$\underline{R}'_{ij} = (\underline{A}_i \times \underline{B}_j) \times \underline{D}_{ij} \quad (5-3)$$

从而有总的新关系:

$$\underline{R}'_{ij} = U(\underline{A}_i \times \underline{B}_j) \times \underline{D}_{ij} \quad (5-4)$$

这样,则实现了对控制规则的修正。

自组织模糊控制系统的结构框如图 5-1 所示。在图中看出:模糊化,模糊算法,精确化,性能测量,控制量校正、控制规则修正等环节组成了自组织模糊控制器。

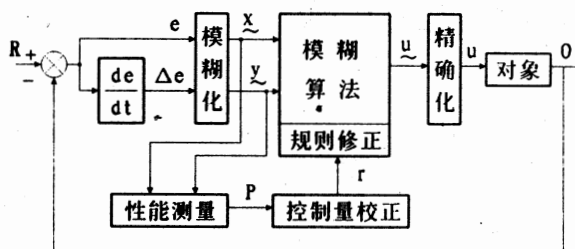


图 5-1 自组织模糊控制系统

上面介绍的是以修改控制规则来改变系统性能的自组织模糊控制系统。在应用中，改变系统性能有三种方法：

- 1、改变模糊量的隶属函数；
- 2、改变偏差 e ，偏差变化率 Δe 和控制量 μ 的比例系数 k_1, k_2, k_3 ；也即是改变系统的增益；
- 3、改变模糊控制规则。

以上任何一种方法，或它们的组合都可以形成自组织模糊控制系统。

5.2 控制规则的单因子自调整

在自组织模糊控制系统中，控制规则的修正需要通过关系矩阵的求取来实现，这种方法在实际控制系统中是不可取的。因为，它不但计算繁锁，而且时间花费较多，所以，速度响应存在问题。

在使用控制表的模糊控制系统中，控制表本身就是控制规则一种表示形式。在控制表中，所有的量都是模糊量。典型的控制表如表 5-2 所示。

表 5-2 典型的控制表

u \ Δc(y)	-3	-2	-1	0	1	2	3
c(x)							
-3	-3	-2	-2	-2	-1	-1	0
-2	-3	-2	-2	-1	-1	0	1
-1	-2	-2	-1	-1	0	1	1
0	-2	-1	-1	0	1	1	2
1	-1	-1	0	1	1	2	2
2	-1	0	1	1	2	2	3
3	0	1	1	2	2	3	3

在表 5-2 中， c 用 x 表示， Δc 用 y 表示，则有：

$$u = \left\langle \frac{x+y}{2} \right\rangle \quad (5-5)$$

并且，可以写成：

$$u = \langle 0.5x + 0.5y \rangle \quad (5-6)$$

在式 (5-6) 中， x 的系数用 α 表示，则很明显 y 的系数为 $(1-\alpha)$ ，故有

$$u = \langle \alpha x + (1-\alpha)y \rangle \quad (5-7)$$

其中： α 取 0~1 之间的值。

取 α 为不同的值时，则表示对偏差 c 和偏差变化率

Δc 取值的数量不同；显然，所得的控制规则也就不同。这就说明，调整系数 α 可以修改控制规则。

当 $\alpha=0.2$ 时，则有控制表如表 5-3 所示。

表 5-3 $\alpha=0.2$ 时的控制表

$u \backslash \Delta c(y)$	-3	-2	-1	0	1	2	3
$c(x)$							
-3	-3	-2	-1	-1	0	1	2
-2	-3	-2	-1	0	0	1	2
-1	-3	-2	-1	0	1	1	2
0	-2	-2	-1	0	1	2	2
1	-2	-1	-1	0	1	2	3
2	-2	-1	0	0	1	2	3
3	-2	-1	0	1	1	2	3

当 $\alpha=0.7$ 时，则有控制表如表 5-4 所示。

表 5-4 $\alpha=0.7$ 时的控制表

$u \Delta c(y)$	-3	-2	-1	0	1	2	3
$c(x)$							
-3	-3	-3	-2	-2	-2	-2	-1
-2	-2	-2	-2	-1	-1	-1	-1
-1	-2	-1	-1	-1	0	0	0
0	-1	-1	0	0	0	1	1
1	0	0	0	1	1	1	2
2	1	1	1	1	2	2	2
3	1	2	2	2	2	3	3

从上面 α 取值不同的表 5-4，表 5-3 可知： α 对控制表的调整有极重要的作用。用 α 修改控制规则有如下特点：

- 1、修改方便。只要改变 α 的值即可改变控制规则。
- 2、反映了人的智能活动， α 值的大小不同，说明对 c 和 Δc 的加权程度不同，这符合人在控制时的智能推理。例如，对象的阶次较高时，对 Δc 的加权应大于对 c 的加权，故 α 取值较小；反之亦反。
- 3、这种方法产生的控制规则也体现了人们推理中的连续性，单值性和正则性等特点。它避免了单凭经验来确定控制规则时产生的空档和断跳情况。

4、可以用数学解析式来表示控制规则，不但处理方便，并使自组织模糊控制有了直接的有效手段。

5.3 控制规则的多因子自调整

控制规则的单因子自调整可以很方便地进行控制规则修改；但这种方法存在一些不足。该方法中，控制规则依赖一个 α 参数；一旦确定 α ，则偏差 c 和偏差变化率 Δc 的权重就确定了；在实际中，控制系统处于不同工作状态时，要求 c 和 Δc 的权重不同；当 c 的绝对值较大时，控制系统的主要矛盾是消除 c ，故此时 c 在控制规则中应占较大的权重；当 c 绝对值较小时，使系统尽快减少超调达到稳态是主要矛盾，故此时要求 Δc 在控

制规则中占较大的权重。单因子调整控制规则无法实现系统状态变化的这种需要。

为了克服单因子自调整的不足,故提出了多因子自调整的方法。在多因子自调整中,当偏差 e 等级不同时,引入不同的修正因子 α_i 。

5.3.1 控制规则双因子自调整

双因子自调整是在 $e=0, \pm 1$ 时采用一个修正因子 α_1 ,在 $e=\pm 2, \pm 3$ 时采用另一个修正因子 α_2 ;所以,控制规则表达式(5-7)改成:

$$u = \begin{cases} <\alpha_1 x + (1-\alpha_1)y>, X=0, \pm 1 \\ <\alpha_2 x + (1-\alpha_2)y>, x=\pm 2, \pm 3 \end{cases} \quad (5-8)$$

初始的控制表如表 5-2 所示,在取 $\alpha_1=0.4, \alpha_2=0.6$ 时,则得到的修改的控制表如表 5-5 所示。

表 5-5 $\alpha_1=0.4, \alpha_2=0.6$ 时的控制表

$u \Delta e(y)$ $e(x)$	-3	-2	-1	0	1	2	3
-3	-3	-3	-2	-2	-2	-1	-1
-2	-3	-2	-2	-1	-1	-1	0
-1	-2	-2	-1	-1	0	1	2
0	-2	-1	-1	0	1	1	2
1	-2	-1	0	1	1	2	2
2	0	1	1	1	2	2	3
3	1	1	2	2	2	3	3

在取 $\alpha_1=0.5, \alpha_2=0.7$ 时,得到的修改后的控制表 5-6 所示。

从表 5-6,表 5-6 中可以看出,采用两个修改因子 α_1 和 α_2 之后,控制规则的修改更加灵活,有利于满足系统不同状态的控制需要。

表 5-6 $\alpha_1=0.5, \alpha_2=0.7$ 时的控制表

$u \Delta e(y)$ $e(x)$	-3	-2	-1	0	1	2	3
-3	-3	-3	-2	-2	-2	-2	-1
-2	-3	-2	-2	-2	-1	-1	-1
-1	-2	-2	-1	-1	0	1	1
0	-2	-1	-1	0	1	1	2
1	-1	-1	0	1	1	2	2
2	1	1	1	2	2	2	3
3	1	2	2	2	3	3	3

给出一个被控对象,它的数学模型为:

$$G(s) = \frac{1}{s(s+1)(s+2)}$$

则采用单因子和双因子进行仿真的结果如图 5-2 所示。其中曲线 2, 3 分别对应单因子 $\alpha=0.5$ 和 $\alpha=0.6$ 时的系统单位阶跃响应;曲线 1 对应于双因子 $\alpha_1=0.4, \alpha_2=0.6$ 时系统单位阶跃响应。从图中可以

看出,双因子修改的控制规则有较好的控制效果:超调小,响应时间短。

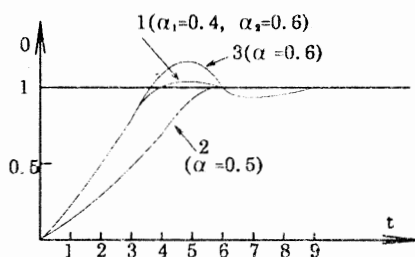


图 5-2 双因子和单因子控制仿真

5.3.2 控制规则的四因子自调整

双因子修正控制规则比单因子要好,但是 α_1, α_2 的调节范围有限,对不同状态下的修正因子的不同要求尚感不足。故提出了四因子修正控制规则的方法,对于表 5-2 所示的控制表,相当在每种偏差等级用一种修正因子;所以,控制规则表达式(5-8)就改变成下面的有四个因子 $\alpha_1, \alpha_2, \alpha_3, \alpha_4$ 的表达式:

$$u = \begin{cases} <\alpha_1 x + (1-\alpha_1)y>, x=0 \\ <\alpha_2 x + (1-\alpha_2)y>, x=\pm 1 \\ <\alpha_3 x + (1-\alpha_3)y>, x=\pm 2 \\ <\alpha_4 x + (1-\alpha_4)y>, x=\pm 3 \end{cases} \quad (5-9)$$

其中: $\alpha_1, \alpha_2, \alpha_3, \alpha_4$ 取 0~1 间的值。

在取 $\alpha_1=0.45, \alpha_2=0.55, \alpha_3=0.65, \alpha_4=0.75$ 时,则控制表 5-2 就改变成下面表 5-7 所示的控制表。

表 5-7 四因子修正的控制表

$u \Delta e(y)$ $e(x)$	-3	-2	-1	0	1	2	3
-3	-3	-3	-3	-2	-2	-2	-2
-2	-3	-2	-2	-2	-1	-1	-1
-1	-2	-2	-1	-1	0	1	1
0	-2	-1	-1	0	1	1	2
1	-1	-1	0	1	1	2	2
2	1	1	1	2	2	2	3
3	2	2	2	3	3	3	3

给出一个对象,它的数学模型为:

$$G(s) = \frac{1}{s(s+0.5)}$$

采用双因子($\alpha_1=0.5, \alpha_2=0.7$)修正控制规则,和采用四因子($\alpha_1=0.45, \alpha_2=0.55, \alpha_3=0.65, \alpha_4=0.75$)修正控制规则,分别得出对给定对象进行模糊控制的阶跃响应如图 5-3 所示的曲线;其中曲线 1 是四因子控制结果,曲线 2 是双因子的控制结果;可见,四因子控制效果比双因子好。

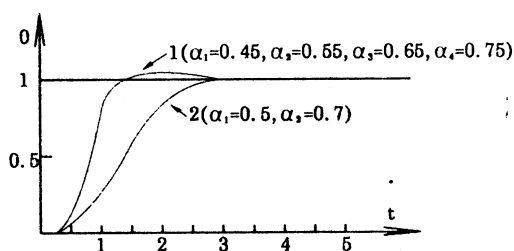


图 5-3 双因子与四因子控制情况

5.4 控制规则自寻优

多因子或双因子修正控制规则对控制系统品质的提高十分有效；但是，还需要寻找一种方法，以使到利用多因子或双因子能实现控制规则的自调整和自我完善。

为了实现控制规则自寻优，采用 ITAE，即时间乘以偏差绝对值积分最小的性能指标为目标函数，而用修正因子为寻优参数；按照目标函数值的大小而不断对因子进行修正，使目标函数值逐步减小，以达到提高系统品质的效果。

控制规则自寻优的控制系统结构如图 5-4 所示。

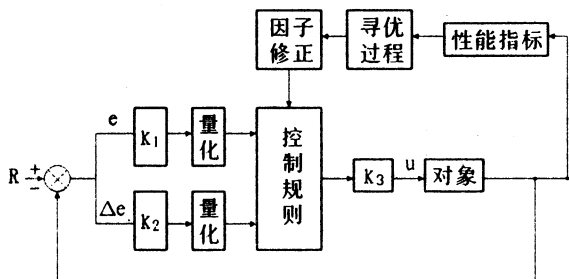


图 5-4 控制规则自寻优的控制系统

目标函数为 J (ITAE):

$$J (ITAE) = \int_0^{\infty} t |e| dt \quad (5-10)$$

由于计算机控制时，组成的是离散系统，故式 (5-10) 也采用离散算式:

$$\begin{aligned} J (ITAE) &= \sum_{i=1}^n iT |e(i)| T \\ &= \sum_{i=1}^n iT^2 |e(i)| \end{aligned} \quad (5-11)$$

其中: T 是采样周期, i 是采样序号。

如果把采样周期 T 看成单位时间时，则可取 $T=1$ ，式 (5-11) 可写为:

$$J (ITAE) = \sum_{i=1}^n i |e(i)| \quad (5-12)$$

控制规则的寻优过程可采用单纯形法，随机方向法或黄金分割法等。下面以单纯形法说明寻优的过程。

设有二个因子是寻优参数，则是二变量，也即是 2 维的寻优过程。这时在 2 维平面中取边长为 a 的正三角形，则有它三个顶点座 (x_1, y_1) (x_2, y_2) (x_3, y_3) 。然后，以三个顶点的值求出三个目标函数值。比较三个目标函数值，取目标函数值最大的顶点为最劣点。接着抛弃最劣顶点，并以该顶点的对应底边为轴翻转，产生一个新的三角形。再重复上述过程，不断抛弃最劣顶点，从而可找到最佳的顶点，使目标函数最小。最终得到的顶点座标就最优的因子。

二因子寻优过程如图 5-5 所示。

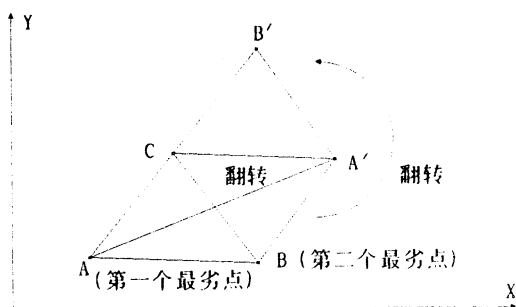


图 5-5 二因子寻优过程

对于四因子过程似类，只是应在 4 维空间中取一个有 $(4+1)$ 个顶点，每个面都是正三角形的多面体。

对于一个给定的对象，其传递函数为:

$$G(s) = \frac{1}{s(s+1)} \quad (5-13)$$

取初始修正因子为: $\alpha_1=0.3$, $\alpha_2=0.4$, $\alpha_3=0.5$, $\alpha_4=0.6$; 以单纯形法为寻优方法，目标函数为 J (ITAE)，则寻优的结果为: $\alpha_1=0.29$, $\alpha_2=0.55$, $\alpha_3=0.74$, $\alpha_4=0.89$ ，对单位阶跃响应的结果分别如图 5-6 中的曲线 1、2 所示，其中曲线 1 是初始条件响应曲线， J (ITAE) = 5.467；曲线 2 是寻优后的响应曲线， J (ITAE) = 1.732。

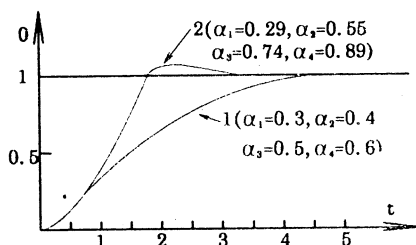


图 5-6 控制规则自寻优后的响应曲线

CPAV 口令密码的破译

江苏省南通市邮电局微机室 曹小忠 陈 健

【摘要】本文介绍了近来十分流行的病毒防护软件 CPAV 的口令字的组成、特点及破译方法。

CPAV 是美国著名的 Central Point 软件公司推出的 Anti-Virus 高级病毒防护软件。它以其友好的界面、强大的功能,受到用户的青睐。为加强系统管理与使用安全性,CPAV 提供了两种菜单方式:一种是仅含基本功能的简化菜单,适用于普通用户;一种是包括系统可选项配置等内容在内的全功能菜单,适用于系统管理员。CPAV 提供设置口令功能,用户可为 CPAV 设置密码位数 ≤ 16 的口令字及改变口令。口令一旦设置后,下列情况需要录入正确的口令字方能执行。

(1) 在 Express Menu 中选择 Full Menu, 即从简化菜单进入全功能菜单;

(2) 在 Options 菜单中配置 Detection Only 以及 Allow Network Access;

(3) 在 Configure 菜单中选择 Change Alert Message, Immunization Exceptions 以及 Changing the Password.

CPAV 包括口令字在内的所有任选项配置信息被放在 CPAV.INI 文件中,这是一个可编辑的文本文件,基中“0”表示该可选项设置为关 (OFF),“1”表示开 (ON)。口令密码 password 由 16 位字符组成,其内容已经过重新编码。用户如果不知其编码规则是很难破译的。

经过分析研究,我们发现 CPAV.INI 文件中口令 password 的组成具有以下规律:

(1) 16 位字符从右到左反方向放置;

(2) 每位字符均已经过转换,需要查找以下解码表才能破译。

找到了以上规律,就不难破译 CPAV 中已经设置的口令密码。由于上述解码表中的字符并非一一对应关系,所以 CPAV 的口令密码并不唯一。

如 password = TTTTTTTTTTg^b^Ec, 其正确的口令密码可为: monkey 或 mcekzq.

通过查看 CPAV.INI 文件,可以获知 CPAV 的口令密码内容。如果用户拥有对 CPAV.INI 文件的修改

权,也可直接修改可选项配置信息及口令字。如:令 password = TTTTTTTTTTTTTTTT (对应解码为空格),可取消口令字。

表 1 CPAV (V1.2 版) 口令密码解码表

序号	密码	解码
1	.	—
2	-)
3	0	^
4	1	
5	9	\
6	A	@%
7	C	?
8	E	co
9	L	d
10	M	b
11	R	a
12	S	f
13	T	空格
14	V	*
15	^	3enz; > / \$
16	a	l.
17	b	k<
18	c	m
19	e	gtu
20	f	h
21	g	qy
22	h	!
23	i	6rw (+")
24	n	psx: &
25	o	g
26	p	{
27	r	5lv. }
28	s	2.##
29	t	78i'
30	u	-
31	w	04j
32	y	=

(注: 解码中所有小写字母亦可用大写)

GW386上的初始配置信息文件

山西长治市人民银行 彭起顺

在 GW386 上的 XENIX 系统下, 有这样一个文件, 其名字为: \USR\ADM\MESSAGES. 我们知道, 在 GW386 每次启动后, 屏幕上均要把当前机器上的设备名及其地址, 中断等数据显示出来, 这些数据是 GW386 出厂时的初始配置. GW386 的配置不同, 这些显示的数据信息也不同. 为了把每次显示的这一现场保存下来, 以便于用户在以后某些时候回过头来查看这些数据, 可以明白在此以前某个时刻对机器的硬件配置进行过变动等情况, 所以, XENIX 提供了文件 /USR/ADM/MESSAGES 来专门存放机器每次启动时的现场 (并且在每次启动的现场之前加上当时启动的精确时间), 这种记录进入多用户状态为触发条件 (若进入单用户维护状态则不进行记录), 以追加的方式对文件 /USR/ADM/MESSAGES 进行更新. 这就使得这个文件增长得很快, 对于 2M 内存 40M 硬盘, 双软并配有一个 60M 的磁带机的 GW386 来说, 每次启动的现场就要在该文件中占 1314 字节, 天长日久, 这个文件就要占去相当多的硬盘空间, 这有时是很不合算的. 为此我们需要定期的对其进行清除, 这种清除, 依用户自己的具体情况, 可以采取如下几种方式之一.

第一种清除方式: 忽略该文件的信息数据, 系统每次启动后即把该文件删掉, 这可以由如下一段 SHELL 程序来实现:

```
#clean /usr / adm / messgaes
rm /usr / adm / messgaes
```

第二种清除方式: 在该文件达到一定长度时, 自动把其删掉, 实行此功能的 shell 程序可以这样来编制, 用户可以先将当前机器上的 /usr / adm / messages 文件删掉, 之后重新启动系统, 注册完毕, 可以用 | 命令查看此时该文件的长度, 这一长度就是该文件每次在系统启动后所增加的长度, 如前述的 GW386 的这一长度为 1314 字节, 这样, 我们可以取该数值的半数来做

/usr / adm / messgaes 文件的最大长度, 如取该数的 20 倍来做这一最大长度, 则此文件长度超过 26280 字节时 (即在上一次删除该文件之后进行了 20 次以上启动时, 即第 21 次启动时), 就自动把其删掉.

```
#clean /usr / adm / messgaes
sp= 1 /usr / adm / messages awk '{print %5}'
if [ $SP -gt 26280 ]
then
rm /usr / adm / messgaes
fi
```

第三清除方式: 在该文件达到一定长度 (如 26280 字节) 时, 不将其删掉, 而是报警给用户, 以提示用户该文件是该清理了, 清理与否? 由用户自己决定:

```
#clean /usr / adm / messgaes
sp= 1 /usr / adm / messages awk '{print %5}'
if [ $SP -gt 26280 ]
then
echo "\n\n\n\n\n^G 警告: /usr / adm / messgaes 文件该清除了!!! \n^G"
fi
```

用户可以根据自己的需要选择其中一种清除方式程序, 并把这一程序放在文件 /etc/rc 中, 由系统在每次启动时自动去做这项工作. /etc/rc 文件类似于 DOS 中的自动批处理文件 AUTOEXEC. BAT. 如何把以上清除 /usr / adm / messgaes 文件程序放在 /etc/rc 中也有多种方式, 最原始最直接的方式是给该程序一个名字, 将其直接放在 /etc/rc 本身之中, 但笔者认为最好的方式是把以上清除 /usr / adm / messgaes 文件的某一个程序段嵌入到文件 /etc/rc.d/4/cleanup 中, 因该文件是处在 /etc/rc 调用范围内的文件之一, 且具体任务就是清除一些不必要的磁盘占用空间.

另外, XENIX 系统中还有一个与此类似的文件 /etc/wtmp, 其每次增加的长度为 1728 字节 (对于前述配置的 GW386), 读者也可以采取类似的方法清除该文件.

鼠标激活内存驻留程序方法

合肥矿山机器厂 CAD 室 姜金友

使用键盘是向内存驻留程序发出激活信号的一种便利方法,过去许多流行的内存驻留程序都利用键盘激活。现在情况有所变化,微机通常都配置了鼠标器,尤其是用于进行 CAD 的机器,鼠标器的使用率非常高,相对键盘来说,用户更乐于使用它。因此设计内存常驻程序时有必要考虑用键盘和鼠标两种硬设备作为触发器。本文以 Microsoft 鼠标器为例说明激活方法。

微机上使用的鼠标器都安装在 COM1 或 COM2 口上,使用之前还必须运行驱动程序。Microsoft 的鼠标驱动程序通过重定向 33H 中断向量,提供了很多低级功能调用。其中 12、20、24 号功能用于设置鼠标器硬件中断的入口及相应条件。12 号功能调用入口参数意义如下:

AX=12, 功能号,

DX: 中断服务程序偏移地址,

ES: 中断服务程序段地址,

CX: 中断程序入口条件。定义哪些条件可以引起中断。其每一位都对一个特殊的条件,如下所示:

位	条件
0	鼠标位置移动
1	左键按下
2	左键放开
3	右键按下
4	右键放开
5	中键按下
6	中键放开

要使中断程序在某种条件下起作用,就把对应的位置 1,否则置 0。只要置 1 的某一位对应的事件发生就立即进入中断例程,进入中断例程时 BX 寄存器保留着按钮状态,服务程序可以通过它进一步限制条件。程序一通过 CX=10 定义只要鼠标左、右键之一按下即进入中断程序,中断程序首先比较是否左、右键同时按下,若不是中断返回,否则显示字符串“MOUSE LEFT & RIGHT BUTTON HAS BEEN PRESSED”后返回。

须注意的是,若中断程序调用了 0 号功能,中断入口及条件均将被清除,需要在中断返回时再次调用 12 号功能进行设置。另外结束中断服务程序指令只能用 RETF,而不能用 RET 或 IRET。低版本的汇编编译

程序不支持 RETF,此时可改用 RET 0 指令。

除了利用 INT 33H 的功能调用设置鼠标硬件入口外,还可以通过重定向 0BH(或 0CH)号中断入口来实现。对 286 以上机器,0BH 是 COM2 对应的硬中断号,0CH 是 COM1 对应的硬中断号,一旦鼠标事件发生均要产生 0BH(或 0CH)号中断调用,因此可以通过拦截该中断而激活驻留程序。程序二就是通过重定向 0BH 号中断检测鼠标,一旦左、右键同时按下即将屏幕转为 640x350 的图形模式并显示有关信息。(注:该程序的前提条件是鼠标安在 COM2 口上,若安在 COM1 口上则需重定向 0CH 中断向量)。

该法同前述方法相比存在一个不足,就是在中断服务程序里不能再使用鼠标器进行操作。程序三是一个常驻内存的计算器程序,可以用键盘和鼠标输入数据计算三角函数、对数及进行普通运算等。该程序既可用鼠标也可用键盘激活,鼠标激活方法同程序一。

程序一:

```
code segment
    assume cs:code,ds:code
    org 100h

start: jmp install
display proc near
J1:    mov ah,0eh
        mov cx,01h
        mov bl,04h
        mov al,cs:[si]
        cmp al,0
        jz J2
        int 10h    ;显示提示信息
        inc si
        jmp J1
J2:    ret
display endp
new-mouse-intr proc far
    cmp bl,03h
    jnz J3    ;比较鼠标左、右键是否同时按下。
    push cs
    pop ds
    lea si,string
    call display
J3:    retf    ;也可用ret 0指令。
new-mouse-intr endp
string db "MOUSE LEFT & RIGHT BUTTON HAS BEEN
        db "PRESSED.",0eh,0dh
```

```

        db "PRESS CR KEY TO CONTINUE.", 0ah, 0dh, 0
install: push cs
        pop ds
        push cs
        pop es
        mov ax, 12
        mov cx, 10
        lea dx, new-mouse-intr
        int 33h          ;设置鼠标中断入口。
        mov bx, cs: [2ch]
        mov es, bx
        mov ah, 49h
        int 21h          ;释放环境块。
        lea dx, install
        mov cl, 4
        shr dx, cl
        inc dx
        mov ax, 3100h
        int 21h          ;驻留内存。
code    ends
end start

```

程序二:

```

code    segment
        assume cs: code, ds: code
        org 100h
start:   jmp install
old-int0bh dw ?, ?
display proc near
        ;同程序一,略。
display endp
new-int0bh proc far

```

sti

```

        push ax
        push bx
        push cx
        push dx
        push si
        mov ax, 03h
        int 33h
        cmp bl, 03h
        jnz J3
        mov ax, 10h
        int 10h
        lea si, string
        call display
        mov ax, 33
        int 33h

```

;软件重置,清除鼠标事件,不能使用0号

功能调用。

```

J3:      pop si
        pop dx
        pop cx
        pop bx
        pop ax
        cli
        jmp dword ptr cs: old-int0bh
new-int0bh endp

```

```

string db "Now screen mode: 640x350, Press CR key to co
ntinue", 0ah, 0dh, 0
install: push cs
        pop ds
        push cs
        pop es
        mov ax, 350bh
        int 21h
        mov old-int0bh, bx
        mov old-int0bh+2, es
        mov ax, 250bh
        lea dx, new-int0bh
        int 21h
        mov bx, cs: [2ch]
        mov es, bx
        mov ah, 49h
        int 21h
        lea dx, install
        mov cl, 4
        shr dx, cl
        inc dx
        mov ax, 3100h
        int 21h
code    ends
end start

```

程序三:略,如读者需要可同作者联系。

351

广告索引

1. 白云山电源设备厂
2. 广州易通计算机应用工程开发公司
3. 广州市电子设备公司
4. 广州科教电脑设备厂
5. 赛宝星河
6. 金泽科技企业有限公司
7. 广州海谊电子仪器实业公司
8. 华力科技开发公司
9. 特强(广州)电子有限公司
10. 广州市昌华计算机有限公司
11. 宏辉科技有限公司
12. 珠海征达科技工贸有限公司广州经营部天河商行
13. 顺达电子有限公司
14. 广州市声乐电脑商行
15. 高域电脑科技有限公司
16. 拓展计算机技术应用研究所
17. 南方计算机网络有限公司
18. 香港现代电子出版社
19. 电子工业出版社广州科技公司
20. 广州袖珍计算机技术服务中心
21. 华粤电脑工程公司
22. 电脑杂志社科技开发经营部
23. 清华大学科学馆

352

Turbo C 中使用汉字

浙江大学

陈兆前 孟建伟

编制一个优秀的应用软件,从减小工作量缩短开发周期考虑,无疑要良好软件工具支持。当前流行的软件工具如 Microsoft C, Quick C, Turbo C 等,功能强大,致命的弱点是并不支持汉字。而为了获得良好的人机界面,汉字输出信息是必不可少的。并且在不少场合,汉字输入也是必要的。解决这个矛盾有两种方法。(1) 汉化软件工具,使与汉字兼容。这当然能够解决这个矛盾。国内也出现了一些汉化软件工具,如南京大学的汉化 Turbo C 等。可是这样做不仅需要大量精力,汉化后的软件工具也很难在功能上代替原来的软件工具。何况对集成环境的汉化对编程人员是多余的。矛盾的焦点仅在于汉字的输入输出处理。(2) 扩展软件工具的功能,使与汉字兼容,与上一种方法相比,这不失为一条捷径。本文拟在这方面做一些初步的探讨,以 Turbo C 为例,其原理也适用于类似的系统。

当前已经出现了支持字符汉字的计算机。这类计算机可以象对待字符一样对待汉字。汉字模由硬件产生,使用者只需要提供机内码即可,问题就十分简单。遗憾的是大多数计算机并无此功能,汉字只能在图形状态下由软件显示。正是鉴于这中现实,讨论以下两类实现汉字使用的方法。

一、自编实现汉字的输入及输出。

1、汉字的输出

原理:读入字库文件以取得点阵,往屏幕上写点阵。这相对较简单,只要熟悉字库结构,很容易实现。以下是一个在屏幕上显示“朋友你好”的简单例子。思路是在 XSDOS 下用中文编辑软件 WPS 给出汉字内码,根据内码计算机字模点阵在字库中的位置;读出字模点阵,按像素打印到屏幕上。所用字库是标准字库经过一些简单处理得到的。

例 1: Pirnthz.c

```
#include <stdio.h>    #include <fcntl.h>
#include <graphics.h>  #include <io.h>
int zkfd=0; /* 字库文件名 */
void initgr(void) /* 图形初始化 */
{
    int grdr=DETECT; int mode=0;
    initgraph(&grdr,mode,"tc");
}
void init(void) /* 字库初始化 */
{
    zkfd = _open("cclibj.dot",O_RDONLY);
    if (zkfd == -1) { /* 字库不存在 */
        printf("THE FILE CCLIBJ.DOT NOT EXIST!\n");
        exit(0);
    }
}
```

```
/* 完成从内码到区码的转换,取出字模点阵
incode: 内码; mat: 字模点阵 */
void get_hz(char incode[], char mat[])
{
    unsigned char high,low; unsigned long position;
    high = incode[0] - 0xa0;
    low = incode[1] - 0xa0;
    if(high == 0x29) high = 0x28;
    if(high >= 0x39) high -= 0x6; /* 换算成区位码 */
    position = (high-1) * 94 + (low-1);
    position *= 32; lseek(Zkfd,position, SEEK SET);
    __read(zkfd,mat,32);
}

/* 在(X0,Y0)打印一个汉字,code[]:内码; color:汉字颜色控制 */
void outchinese(int x0,int y0,char code[],int color)
{
    register int i,j,x,y,pos; char mat[32];
    unsigned char MASK[] = {0x80,0x40,0x20,0x10,
                           0x08,0x04,0x02,0x01};

    get_hz(code,mat); y = y0;
    for(i = 0; i < 16; ++i) { /* 打印汉字点阵 */
        x = x0; pos = 2 * i;
        for(j = 0; j < 16; ++j)
            if((mask[j % 8] & mat[pos + j / 8]) != NULL)
                putpixmap(x,y,color);
                ++x;
        ++y;
    }
}

void main()
{
    char *s = "朋友你好";
    int y = 100; int x = 20;
    init(); initgr();
    while(*s != NULL){
        while((x < maxlinenum) (*s != NULL)){
            outchinese(x,y,s,blue);
            x += 20; s += 2;
        }
        x = 20; y += 20;
    }
    getch(); closegraph();
}
```

字库太大时,内存的浪费会很严重。如果程序中只是用到某些汉字,可以摆脱原字库的控制,构造自己的字库。这只要把用到的汉字的字模从原字库取出,放到一个文件中或者直接赋予程序中的变量即可。把上面的例子稍加修改就可以实现此字库变换。这对在小内存计算机上实现汉字处理是很有利的。

2、汉字的读取。

原理:读外码,查外码字典,取得相应内码。读取汉字工作量较大,而且有一定难度。必须做好编码工作,构

造从外码字符串到内码的转换字典。具体的说,要做编码,外码输入,重码显示,重码选择等工作。要使输入方便实用,不能不考虑用词组输入,需要构造词组字典。如果这些字典大小受内存限制,还要考虑压缩以减少内存。使输入方便毕竟不易。

当要输入的汉字较少时,可以采用各种实用方法,例如,可以显示所有汉字,用菜单输入,正象一些游戏软件所做的。

应用以上方法,可以实现一个基本的汉字系统。这样的系统缺点是性能往往比不上通用的汉字系统显示速度。输入方法等都难以尽如人意,并且汉字的输入输出都要系统的调用自己的子程序,使用上不够方便。优点是汉字显示效果好而极易控制,在此基础上开发的软件具有良好的可移植性。

二、利用汉字系统的支持。

一般的汉字系统都修改了 INT 10H 显示中断和 INT 16H 键盘中断,并且尽量与 BIOS 保持一致。高级语言程序的显示和绘图也往往通过 BIOS,这就是可以利用汉字系统的原因。下面以 Turbo c 2.0 和 Liuph-ccdos 5.10 为例进行说明。

附加说明一点:有些机器,如果只需输出输入汉字信息而不需要同时输出图形,那么不必要初始化为图形状态。标准 C 函数 printf, scanf 可以用来直接打印读入汉字,如浪潮 386;有机器则必须先初始化为图形状态,如 ALR286。以下程序均在 ALR286 上运行通过。

初始化图形状态后,我们幸运的发现,标准 C 函数 printf, scanf 和 Turbo c 的图形库函数是兼容的。在一个程序中,很容易同时处理汉字信息和图形信息,下面的例子在屏幕上画圆,并输出输入汉字:

例 2: II cllow.c

```
#include <stdio.h> #include <graphics.h>
void initgr(void) /* 初始化图形状态,见例子 1 */
void main()
{
    char name[8]; char msg[20];
    initgr(); setcolor(BLUE);
    circle(200,200,100);
    printf("您的名字是")
    setcolor(YELLOW);CIRCLE(100,100,60);
    STRCPY(msg,"您好,");
    goto (5,20); /* 输入定位 */
    scanf("%s",name);
    strcat(msg,name);
    strcat(msg,"!");
    goto (10,30); /* 输出定位 */
    printf("%s",msg);
    getch(); /* 为观察屏幕 */
}
```

客观的说,这样已经基本做到了同时处理汉字和图形,汉字的写,读,定位简单实用。可是显示的效果比较粗糙,因为不能控制显示汉字的色彩。牺牲部分可移植性,可以调用显示中断 BIOS 的中断弥补这个缺陷。例 3 是按照这各思路实现的多背景色多前景色汉字显示与图形

显示。此环境下汉字有八底色八前景色以及亮暗两各属性。例子以七前景色以及亮暗两种属性在屏幕 10-70 行。1-24 列重复显示“朋友你好”四字。

例 3:COLORHZ.C

```
#include <string.h> #include <dos.h>
#include <conio.h> #include <graphics.h>
#include <stdio.h>
union REGS inregs,outregs;
void initgr(void) /* 初始化图形,见例 1 */
void initcur(void) /* 光标初始化 */
{
    inregs.h.bh=0;
    inregs.h.ah=0x02;
    inregs.h.dl=10;
    int86(0x10,inregs,outregs);
}
int row=1; int col=10;
void changecur(void) /* 变换光标位置 */
{
    if(col==71){
        col=10; row++;
    }
    else col++; /* 如果到最后一行,则重复打印最后一字 */
    inregs.h.bh=0;
    inregs.h.ah=0x02;
    inregs.h.dl=col;
    inregs.h.dh=row;
    int86(0x10,inregs,outregs);
}
#define C(a,b) b+(a<4) /* 颜色属性计算 */
void drawhz(char c,int ck,int bk)
{
    int color;
    color=C(bk,ck);
    inregs.h.ah=0x09;
    inregs.h.bh=0;
    inregs.h.al=0;
    inregs.x.cx=0x01;
    inregs.h.bl=color;
    int86(0x10,inregs,outregs);
    changecur();
}
void drawhz str(char * s,int bk,int ck)
{
    int i=0;
    while(s[i++]!=NULL)
        drawhz(s[i],bk,ck);
}
void main()
{
    char s[]="朋友您好";
    int i,j;
    initgr(); initcur(); setcolor(RED);
    for (i=1;i<16;i++){ /* i=0 时,前景色为黑,不打 */
        for (j=0;j<16;j++){
            circle(200,200,j*10);
            drawhz str(s,i,j);
        }
    }
}
```

以上程序中,因为在 Liuph-ccdos 下工作,没有必要关闭图形状态。可以看出,在汉字 DOS 下实现汉字,图形处理要比自编程容易的多,但是程序的可移植性明显降低了。

文本方式下的汉字显示

上海工业大学 刘树东 郎文鹏

【摘要】本文介绍了 EGA/VGA 上实现文本方式下汉字显示的方法，并给出了用 TC2.0 编写的完整程序清单。

一、基本知识

EGA/VGA 是通用图形显示适配器，有两种基本操作模式：文本模式和图形模式。图形模式下，DRAM 中的一个单元表示 CRT 上的一个单点；在文本模式中，一字节 ASCII 码输入显示内存形成屏幕上显示的一个完整的文本字符。文本方式占用内存少，CPU 开销也少。

一般的汉字显示都是在图形模式下画点完成的，为节省内存，有些用户系统采用小字库技术。但有些场合，如实时性要求较高时，或者要求文本模式显示，就需另辟新途。文本方式下的汉字显示便应运而生。

PC 为了将 ASCII 码文本转换成屏幕上的一个像素阵列，用到了一张转换表即字符发生器。在 MDA 和 CGA 中，该表装在 ROM 中；而 EGA/VGA 中，该表装在 DRAM 的页面之中。利用 BIOS 功能调用，可以装入用户自定义的常规字符集，可以同时活化一或两个字符集，从而可以显示用户定义的特殊字符，包括汉字。

二、ROM BIOS

标准的 PC/XT/AT BIOS 提供一系列驱动 MDA 和 CGA 的视频驱动服务，由 INT10H 调用。EGA/VGA 都包括扩充的板上 BIOS，扩展视频驱动功能并与原程序反向兼容。EGA/VGA 的 BIOS 卡 16Kb，在 C0000H-C3FFFH 之间。

INT10H BIOS 的功能相当丰富，与装入用户字符集有关的中断调用有：

①、装入常规字符发生器：功能 17，与功能 0

输入参数：AH=10H

AL=0

ES: BP=系统 RAM 中字符数据首址

CX=装入的字符数 (1-256)

CX=装入的起始字符 (0-256)

BL=选择的 EGA 字符表 (0-3)

BH=每个字符的高度 (1-32)

返回值：无

②、初始化 INT IFH 向量：功能 17，子功能 32

输入参数：AH=11H

AL=20H

ES: BP=系统 RAM 中字符数据的地址

返回值：无

③、置图形模式为显示常规字符集：功能 17，子功能 33

输入参数：AH=11H

AL=21H

ES: BP=系统 RAM 中字符数据的地址

CX=每个字符的字节数

BL=显示的行数

BL=1: 14 字符行

BL=2: 25 字符行

BL=3: 43 字符行

BL=4: DL 中包含字符行数

返回值：无

其中，①用于文本模式，②用于 CGA 兼容图形模式，③用于组合文本与图形。三者都将修改字模表，从而实现汉字的文本方式显示。

三、汉字显示

我们通常能获得的汉字点阵是 16×16 或 24×24，而 EGA/VGA 的字符集分别为 8×14 和 8×16。因此，16 点阵汉字需占两个西文字符，点阵数据也需做一些简单交换或压缩；24 点阵需占 6 个西文字符，分两行显示。当然，显示器不用 25 行显示，而用 14 行时，24 点阵汉字只需占 3 个字节。

实现汉字显示的步骤包括：

①获取汉字字模数据；

②设置显示模式，装入汉字字模；

③如果同时活化两个字符集，应重设调色板，关闭前景亮度功能，以免两个字符集显示亮度不同；

④按其属性显示汉字所占用的字符。

本文所附程序清单是文本方式下显示汉字的实例，用 TC 编写，在配置 VGA/TVGA 的 286 上通过。它将“电脑杂志”的 16 点阵字模从大字库中取出，装入字符集 1，并显示在计算机屏幕中央（参所附清单）。

在图形模式下, CPU 的文本处理功能实际上是把文本字符画入 DRAM。此时仍可采用修改字模表的方法显示汉字。

四、应用

修改字模显示汉字, 占用内存少, CPU 负荷小, 且在文本和图形模式都可使用。虽然同一时刻 EGA/VGA 至多只能显示 256 个汉字, 但结合动态装入字模的方法, 在许多场合都足用了。因此许多著名软件, 如 Norton 系列的 Ncc, NDD, Soarccr, CPAV 等采用修改字模的办法改善界面; 在工控软件中, 达到方法也用来显示中文提示菜单、版权信息、特殊字符等。

附程序清单

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

void str16 (char *hz);

unsigned char dot[4][32]; /*待装字模点阵*/
unsigned char col[17];
unsigned char far *p;
struct SREGS s;
union REGS r;

void main ()
{ int mode, driver, i;
  unsigned myES, myBP;

  str16("电脑杂志");
  for(i=0; i<17; i++) col[i]=i%8; /*调色板数据*/
  p=dot[0]; /*指向待装字模点阵的首址*/
  gotoxy(25, 2); /*设置光标位置*/
  textattr(MAGENTA*16:CYAN*16:0*8); /*选用字符集0*/
  cprintf("ABCDEFGH ABCDEFGH\n"); /*显示字符*/

  bdos(12, 0, 8); /*等待敲键*/
  myES=-ES; /*保护ES和BP*/
  myBP=-BP;
  -BP=FP-OFF(p); /*ES:BP=字模点阵的首址*/
  -ES=FP-SEG(p);
  -AX=0x1100; /*装入字模*/
  -BH=0x10; /*字符高度 VGA:16;EGA:14*/
  -BL=1; /*装入字符集1*/
  -CX=8; /*装入字符个数*/
  -DX='A'; /*起始字符*/
```

```
geninterrupt(0x10); /*调用视频中断BIOS*/
r.x.ax=0x1103; /*选择活动字符集*/
r.h.bl=4; /*活化字符集0和1*/
int86(0x10, &r, &r);
r.x.ax=0x1002; /*设置调色板*/
r.x.dx=FP-OFF(col); /*ES:DX=调色板数据首址*/
s.es=FP-SEG(col);
int86x(0x10, &r, &r, &s);
-BP=myBP; /*恢复ES和BP*/
-BP=myBP;
window(25, 10, 50, 15); /*设置光标位置*/
textattr(MAGENTA*16:CYAN*16:8); /*选用字符集1*/
cprintf("ABCDEFGH ABCDEFGH\n"); /*显示汉字*/
```

```
bdos(12, 0, 8); /*等待敲键*/
r.x.ax=3; /*重设文本方式*/
int86(0x10, &r, &r); /*恢复缺省设置*/
exit(0);
}

void str16 (char *hz) /*取所需字模点阵*/
{ FILE *fp;
  long offsetl;
  char *s=h;
  int i, j, k=0;

  if ((fp=fopen("c:\\cclibj.dot", "rb"))==0) {
    printf("Can not open the lib file! \n");
    exit(1);
  }
  while(*s) {
    offsetl=(long) ((*s+89)*94+*(s+1)+95)*32;
    fseek(fp, offsetl, SEEK-SET);
    for (i=0; i<16; i++)
      for (j=0; j<2; j++)
        dot[k][i+j*16]=fgetc(fp);
    s+=2;
    k++;
  }
  fclose(fp); /*关闭字库文件*/
}
```

参考文献:

[1]明智,《中高分辨率接口板 EGA/VGA 应用开发指南》北京科海培训中心, 1990.6

赋值运算符“=”在 ORACLE 数据库 的 PRO * C 程序接口中的妙用

岳阳巴陵公司第一计算机应用研究所 陈 实

在 ORACLE 数据库中 SQL 语言是一种非过程化语言，它的特点是，前后语句的执行一般不存在相互依赖的关系，在使用方式上与计算机操作系统中的命令语言类似，比较 C、FORTRAN、COBOL 等过程化的高级语言，SQL 语言缺少“循环”、“条件分支”等灵活的程序结构，因此，尽管 SQL 语言的功能很强，在使用上仍存在一些限制，为此，人们设计了可嵌套在过程化语言（如 C 语言）中的 SQL 结构，有了这种结构，程序员就可把 SQL 语言的特点和高级语言（也称主语言）的特点结合起来设计应用程序，这种应用程序比单独地使用 SQL 和主语言，功能更强，灵活性更大，由此而产生的程序结构，我们称之为 PRO * C 程序接口。

然而本人通过对它较长时间的使用，发现它还存在某些不足之处，即利用 PRO * C 结构编制的应用程序（在此称为 PC 文件），在预编译时，预编译程序（PCC）把 PC 文件中的某些特殊变量（如存放 ORACLE 数据库“表名”的变量，这里不仿设它为 A）当作常数字符串来进行处理，送到预编译程序（PCC）预编译时系统自动定义的特定数组变量（这里不妨设它为 B）中，而不是作为变量，把它的内容传递到这一特定数组变量中，在程序运行过程中，调用这一特定数组变量 B 时，系统把存放 ORACLE 数据库“表名”的变量 A 的名字作为变量 B 的“值”来处理，而没有把变量 A 的内容作为变量 B 的“值”来处理，因此，程序运行时就会面目全非。那么，我们可不可以不在 PC 文件预编译后，利用编辑软件打开预编译的产物（C 文件），找到特定数组变量 B，然后用变量 A 中的内容替代变量 A 的名字在特定数组变量 B 中的位置呢？回答是肯定的，因为，大家知道，在 C 语言中，利用赋值运算符“=”，实现这一功能，是非常容易的，下面为一实例程序：

/*

程序名：CS.PC

功能：程序运行时，利用键盘输入的“表名”，删除

ORACLE 数据库中某表的内容。

注意：

用 PRO * C 预编译 CS.PC，Microsoft ‘C’编译，连接 CS.C，然后运行 CS.EXE。

方法如下：

```
C>PCC INAME=CS.PC HOST=C
C>CL -AL -C CS.C
C>LINK CS,..., SQLMSC/SE: 512/STACK: 10000/MAP
C>CS
```

附程序清单：

```
*/
#include <stdio.h>
EXEC SQL BEGIN DECLARE SECTION;
VARCHAR udi[10]; /* 存放用户名的数组变量 */
VARCHAR pwd[10]; /* 存放用户口令的数组变量 */
EXEC SQL END DECLARE SECTION;
EXEC SQL INCLUDE SQLCA;
main()
{
    char YYY[3]; /* 存放用户要处理的表名 */
    scanf("请输入用户所要处理的表名%S", &YYY);
    /* -----
    以用户名为ack、口令为ack的身份进入ORACLE数据库。
    如果产生错误，程序去执行标记为errexist的一段程序。
    -----*/
    strcpy(uid.arr, "ack");
    uid.len=strlen(uid.arr);
    strcpy(pwd.arr, "ack");
    pwd.len=strlen(pwd.arr);
    EXEC SQL WHENEVER SQLERROR goto errexist;
    EXEC SQL CONNECT :uid IDENTIFIED BY :pwd;
    EXEC SQL DELETE FROM YYY; /* 此语句运行的结果是
    删除不存在的名为"YYY"表的内容，而不是处理数组变量"YYY"
    里面的"值"。*/
    EXEC SQL COMMIT WORK RELEASE; /* 提交数据给OR
    ACLE */
}
```

在 MSDOS5.0 下为 2.13H 和 SPDOS 合理地配置 CONFIG.SYS 文件

河南省南阳地区农业银行 陈君阳

目前 286 及 386 以上档次的微机已成为主流, 在这类微机上使用新推出的操作系统 MSDOS-5.0, 更是如虎添翼, 受到广大用户的喜爱。但是, 有些汉字操作系统在 DOS-5.0 下存在着难以安装和不能充分利用资源等问题, 如 2.13H 和金山 SPDOS 汉字操作系统。虽然有些文章介绍了这两个系统的安装方法, 但对这二者兼顾不够, 对常在这两个系统之间转换的用户来说, 每次转换仍然需要更新 CONFIG.SYS 文件后, 重新启动微机, 并且牺牲了 DOS-5.0 新增加的部分强大功能, 存在着硬件和软件资源的浪费。

2.13H 和 SPDOS 都有汉字系统的撤离功能, 能不

能在撤离一个汉字系统后既不用修改 CONFIG.SYS 配置文件也不用重新启动, 而只要执行相应的批处理命令就可以进入另一个汉字系统?

能不能在 2.13H 和 SPDOS 下都可以让汉字系统利用扩展内存或扩充内存存放字库, 而把宝贵的基本内存让给用户支配?

能不能更充分地利用 386 微机的富裕内存, 使 1.2M 甚至 1.44M 的软盘拷贝一次完成, 避免恼人的多次插换?

能不能让 DOS 驻留在高内存区 HMA, 让用户使

```
goto p1;
errexit;
EXEC SQL ROLLBACK WORK RELEASE; /*不提交数据
给ORACLE */
printf("error");
p1:
exit(0); /* 返回到DOS */
}
```

/* 程序名: CS.C
/* 此程序(CS.C)是预编译程序(PCC)预编译(CS.PC)时随机产生
的 */

```
static char sq003[16]="DELETE FROM YYY";
/* SQ003[16]数组变量由预编译程序(PCC)预编译(CS.PC)时随机
产生,利用编辑软件看预编译(CS.C)时,可发现这一数组变量。*/
```

```
main
{
char YYY[3]
scanf("请输入用户所要删除的表名%S", &YYY);
sq003[12]=YYY[0]; /*用真要处理的表名替代假表名"YYY"在
sq003数组变量中的位置,此语句由用户查看(CS.C)后编写,YYY
与SQ003数组变量之间"值"的传递由赋值符"="建立桥梁。*/
sq003[13]=YYY[1];
sq003[14]=YYY[2]
```

SQLOSQ(SQ003,...); /*SQLOSQ函数利用SQ003[16]数组变量
中的内容执行删除功能,SQLOSQ函数是由预编译程序(PCC)预编
译时自动产生的,它是预编译程序(PCC)的特定函数 */

}

用高端内存块 UMB, 给用户更大的存储空间?

能不能同时使上述功能共存而不发生冲突?

很多用户对此非常关心。回答也是可行的, 笔者经过对 DOS5.0 系统配置的分析, 针对广泛使用的 2.13H 和软字库版的金山 SPDOS-5.0 两个汉字操作系统, 设计了一个较为合理的配置文件 CONFIG.SYS, 然后对这两个汉字操作系统做一些修改, 较为圆满地解决了这些问题, 弥补了有些方法的不足。下面把系统的配置方法和汉字系统的修改方法介绍给大家共享。

系统配置文件 CONFIG.SYS 的清单:

```
C:\>type config.sys
DEVICE=C:\ANSI.SYS
DEVICE=C:\DOS\HIMEM.SYS /INT15=256
DEVICE=C:\DOS\RAMDRIVE.SYS 260 /E
DEVICE=C:\DOS\EMM386.EXE 1440 RAM
DOS=HIGH,UMB
SWITCHES=/K
FILES=30
BUFFERS=20
```

下面对配置文件作一些解释:

第 1 行, ANSI.SYS 必须为 2.13H 本身所带的驱动程序, 若使用 DOS-5.0 带的 ANSI.SYS 会造成死机。

第 2 行, HIMEM.SYS 所带的参数 /INT15=256 是为金山 SPDOS 装载二级字库预留 256KB 的扩充内存 (EXTENDED MEMORY) 而设。因为金山 SPDOS 直接使用中断 15H 接口来管理扩充内存 (EXTENDED MEMORY), 而不能使用由 HIMEM.SYS 管理的扩充内存 XMS。

第 3 行, 用 RAMDRIVE.SYS 在 HIMEM.SYS 管理的扩充内存 XMS 中为 2.13H 建立 260KB 的虚拟盘。

第 4 行, 在 386 以上档次微机中用 EMM386.EXE 仿真 1.44M 扩展内存 (EXPANDED MEMORY) 和管理高端内存块 UMB。其作用一是增加几十 KB 的高端内存; 二是用 PCTOOLS 复制高密度软盘时, 能够一次性读写完毕, 避免了多次插换软盘的烦恼。(当内存小于 3M 时不能一次完成复制); 三是为 MFOXPLUS 数据库提供 64KB 的扩展内存, 从而提高数据处理的速度。

第 5 行, 让 MD-DOS 在高内存区 HMA 中运行, 为用户节省基本内存。

第 6 行, 屏蔽键盘增强管理功能, 使用常规键盘功能, 以避免汉字输入与显示时的半个字滞后现象。

第 7-8 行, 可有可无, 视个人需要而定。

为使汉字系统正常运行, 仅这样配置是不够的, 还需要对一些文件进行如下的修改:

对 2.13H 读虚拟盘字库的执行文件 FILE3.COM 修改如下:

```
C:\213>DEBUG FILE3.COM
```

```
-A 278
```

```
2EC0:0278 ADC DX,+14; * * 该句原为: ADC DX,+10
```

```
2EC0:027B
```

```
-W
```

```
Writing 001A9 bytes
```

```
-Q
```

```
C:\213>
```

对金山 SPDOS 的读字库执行文件 SPLIB.COM 参照文献

[1]修改如下:

1.若用户硬盘小于 32M, 用 PCTOOLS 查找 SPLIB.EXE 文件中两个 16 进制字符串, 查找 80 7F 16 F8 改为 80 7F 17 F8, 查找 8B 47 10 改为 8B 47 11, 通过这样改造后, 在小于 32M 硬盘的微机上通过。

2.若硬盘大于 32M 时, 读者可参照文献[1]自行解决。或参照文献[3], 编写修改 DOS 绝对读盘中断的程序, 运行驻留内存后, 再运行 SPLIB.EXE。

注意: 配置文件中的 2-4 行必须按顺序来配置, 且所有的设备驱动程序均需指出其路径。第二行为 SPDOS 预留的内存必须为 256KB, 否则 2.13H 不能正常显示。

对 286 微机, 理想内存最小为 2M, 因微处理器所限, 不能提供高端内存 UMB, 也不能仿真扩展内存。所以应把配置中的第 4 行去掉, 并把第 6 行中的参数 UMB 去掉。

对 386 以上的微机, 理想内存最小为 3M, 当内存为 2M 及以下时, 用 PCTOOLS 不能一次完成高密度软盘拷贝。

无论那种微机, 当内存小于 1M 时, 都不能兼顾 2.13H 和 SPDOS, 应根据个人需要进行裁剪, 分别配置。

按照这个配置及修改方案, 运行 2.13H 时选择把字库装入虚拟盘, 运行金山 SPDOS 时把二级字库装入 256KB 的扩充内存, 最终可供用户支配的内存都在 500 KB 左右, 用户还可以把 DOSKEY、SETVER、APPEND、FASTOPEN、SMARTDRV 等强大的 DOS 命令或驱动程序放在高端内存 UMB 中运行, 以提高效率, 而不会加大内存开销。

笔者在 3712KB 内存的 MITAC-386DX 和 2048KB 的中环 286 微机上运行通过, 所用 PCTOOLS 为 5.0 版, 得到满意的效果。

参考文献

[1] 郭建华 在 MS-DOS 5.0 系统中安装金山汉字系统, 《计算机世界月刊》, 1993.1 P59

[2] 中国科学院沈阳计算所信息技术开发公司 祝军等译, 《Microsoft MS-DOS 5.0 磁盘操作系统》, 辽出临图字[1991]第 259 号。

[3] 郑长兵 高版本 DOS 绝对读写盘功能调用的分析及完善, 《计算机世界月刊》, 1992.12 P24。

自适应各种汉字系统的屏幕保存/恢复程序

湖南省双峰工商银行 罗 辉

【摘要】本文提供的中西文屏幕信息的保存和恢复 (SAVE/RESTORE) 程序,较好地解决了各种汉字系统下不同显示卡屏幕信息的保存和恢复,也是对 FOXBASE+ 的 SAVE/RESTORE SCREEN 命令的补充。

曾有许多刊物刊登了一些用于保存和恢复汉字系统屏幕信息的程序,以弥补 FOXBASE+ 中 SAVE SCREEN/RESTORE SCREEN 命令不能处理屏幕汉字信息的不足。但使用中我们发现,这些程序都只能在特定的环境下运行。针对不同的显示卡和不同的汉字系统,必须对程序进行相应的修改才能适应,否则将造成屏幕闪动或根本不能运行,对系统的通用性以很大的制约性。

我们知道,现在市面流行的汉字系统的屏幕信息显示大都是通过调用中断 10H 实现的,而 INT 10H 本身对不同显示卡具有自适应性,能自动调节内存屏幕缓存区的地址和大校因此我们可以通过调用 10H 中断来避开具体显示卡的判别,以提高程序的通用性。

下面在 TURBO PASCAL 5.0 环境下运行通过的 SAVE.PAS 和 RESTORE.PAS 程序,就是根据上面的思想编制的中西文屏幕信息的保存/恢复程序,编译成 EXE 文件后就可以在 DOS 环境下或各种汉字系统下运行,或者在 DBASE/FOXBASE+ 环境下通过 RUN 命令使用它们。它们的使用格式分别是:

SAVE diskfile coll row1 col2 row2

和 RESTORE diskfile

其中, diskfile 是屏幕信息存盘文件名, coll, row1, col2, row2 分别是要保存的屏幕范围:起始列、行,终止列、行的坐标值。通过调节 coll, row1, col2, row2 的值可以方便灵活地对各种中西文屏幕任意区域信息进行保存和恢复,弥补了 FOXBASE+ 中的 SAVE 和 RESTORE 命令的缺陷。

本程序在 DOS 3.3、UCDOS 2.1、CCDOS 4.0、西山 CCDOS 4.03、CCDOS 2.13H 等系统下均运行通过。

```
program save; { 屏幕信息保存程序 SAVE.PAS }
uses dos;
var
  reg: registers;
  fp: file of byte;
  col, row, coll, row1, col2, row2: byte;
  errcode: integer;
begin
  if paramcount <> 5 then { 命令行参数的合法性检查 }
  begin writeln('Paramter Error! ', #7); exit; end;
```

```
val(paramstr(2), coll, errcode); { 命令行中屏幕坐标转换 }
val(paramstr(3), row1, errcode);
val(paramstr(4), col2, errcode);
val(paramstr(5), row2, errcode);
if (coll < 0) or (row1 < 0) or (col2 > 79) or (row2 > 24) then
  { 坐标的合法性检查 }
begin writeln('Screen over! ', #7); exit; end;
{$I-}
assign(fp, paramstr(1)); rewrite(fp);
{$I+}
if IOResult <> 0 then exit;
write(fp, coll, row1, col2, row2); { 保存屏幕坐标范围值 }
with reg do
begin
  for row:=row1 to row2 do
  for col:=coll to col2 do
  begin
    ah:=2; dl:=col; dh:=row; bh:=0;
    intr($10, reg); { 循环置光标历经要保存的点 }
    ah:=8; bh:=0; intr($10, reg);
    write(fp, al, ah); { 保存屏幕点的 ASCII 码值和属性 }
  end;
  ah:=2; dl:=coll;
  dh:=row1; bh:=0; intr($10, reg); { 置光标回到起始点 }
  close(fp);
end;
end.
```

```
program restore; { 屏幕信息恢复程序 RESTORE.PAS }
uses dos;
var
  reg: registers;
  fp: file of byte;
  col, row, coll, row1, col2, row2: byte;
  errcode: integer;
begin
  if paramcount <> 1 then { 命令行参数的合法性检查 }
  begin writeln('Paramter Error! ', #7); exit; end;
  assign(fp, paramstr(1)); reset(fp);
  if IOResult <> 0 then exit;
  read(fp, coll, row1, col2, row2); { 读屏幕坐标范围值 }
  with reg do
  begin
    for row:=row1 to row2 do
    for col:=coll to col2 do
    begin
      ah:=2; dl:=col; dh:=row; bh:=0;
      intr($10, reg); { 循环置光标历经要保存的点 }
      ah:=8; bh:=0; read(fp, al, bl); cx:=1;
      intr($10, reg); { 写各屏幕点的 ASCII 码值和属性 }
    end;
    ah:=2; dl:=coll; dh:=row1; bh:=0;
    intr($10, reg); { 置光标回到起始点 }
    close(fp);
  end;
end.
```


一种简单有效的 C 程序加密方法

云南省化工设计院 尹敬东 杨德荣

在软件的安全性方面,有许多行之有效的加密方法,如激光打孔、磁道弱位技术等,但这类方法一般的程序开发人员往往不容易掌握。笔者利用常被人们忽略的磁盘卷标号做密钥,编写了一个通用函数 islabel(),供广大用 C 语言开发软件的同行参考。该函数在程序的任何地方调用均可,其返回值为 C:盘卷标号。由于您开发的软件存储于硬盘,非法拷贝者至多将整个硬盘的文件全部拷走,而往往忽略当前机器某一逻辑盘的卷标号是什么。因此,只要在程序的某处调用该函数,并判断函数返回的当前 C:盘卷标号是否正确,便能断定使用您的软件的用户是否合法。

程序例中,假设密钥为 C:盘卷标号 ALPHA-1,函数 main program()为假设的应用程序。该程序用 Borland C++2.0 编写,在 AST P11 386/33、MSDOS3.30 环境下运行通过。

源程序清单:

```
#include <stdio.h>
#include <string.h>
#include <dos.h>
#include <dir.h>
#include <stdlib.h>
char * islabel();
int main(void)
{
    char * s; int done;
    s = islabel(); done = strcmp(s, "ALPHA-1");
    if (done == 0) main program();
    else { printf("You are unlawful user!\n");
        getch(); abort();
    }
    return 0;
}
char * islabel()
{
    char * fl;
    struct fblk fblk;
    findfirst("c:\ *.*", &fblk, FA_LABEL);
    strcpy(fl, fblk.ff_name);
    return fl;
}
int main program()
{
    printf("ok\n");
    return 0;
}
```

358

一类链表丢失的文件修复方法

西安市空军工程学院 程江勇

在磁盘的使用当中,由于不慎或其它因素造成磁盘文件 FAT 表的断裂,使文件读错误,给用户带来很大的损失。笔者经过对 FAT 表的分析,找到了一种修复文件的方法。现通过一例说明之,以飧读者。

故障现象:文件 TD.DAT 原有 670 多 K 字节(存放于 1.2M 软盘之上)。列目录时文件长度显示正常,但只能读出很少内容,且混乱。用 PCTOOLS 显示其磁盘映象图,一切正常,但显示文件图时只有几扇内容,说明 FAT 表发生断裂。

修复方法:(1)先把原盘按 DISKCOPY 方式拷入一张空盘,删去其它几个文件。可使 FAT 表中只含有 TD.DAT 的链。

(2)首簇的修复。在 PCTOOLS 下,进入 VIEW/EDIT 磁盘服务功能,先进入目录区(ROOT),TD.DAT 目录项第 26,27 字节为文件首簇号,记下后进入 FAT 表;找到首簇所指单元,发觉它并不是文件开头,而是文件中部某一地方。按照 FAT 表的记录方法。(即本簇内容为文件下一簇的地址,每三个 8 位数据为一组,每组记录两个簇,读时遵守低位在前,高位在后的原则,每次读 16 位,且当簇号为偶数时取低 12 位为下簇地址,簇号为奇数时取高 12 位。文件末尾用 FF8~FFFH 中任一个标记,未用的簇填 00H,如 FAT 表开头为:

F9 FF FF 03 04 00 05 06 00 簇 002
中内容为 4003,由于簇 002 为偶数,取低 12 位,即得下一簇地址为 003,又可得 003 中内容为 0040,取高 12 位,得 004,以此类推。)逆推找到文件首簇为 002,故在目录表中,把 TD.DAT 项的第 26,27 字节改为 0200。

③簇链的修复。对于连续的磁盘空间,其簇号的存放也是连续的,如 021 022 023…。由此可查到簇号错乱的地方,把它改正即可。对于不连续的磁盘空间,由于 FAT 表会填以 00H,故可很容易找到断裂处,只要把 00 区前一个的簇内容填上 00 区后第一个簇号的地址即可。在文件的末簇填入 FFF 标记。文件即修复完毕。

修复后的文件可正常使用。在 PC-AT 机上, DOS3.3 下通过。

359

电脑杂志社科技开发经营部推荐如下产品给广大用户：

♪93 年《电脑》第 7、8 期介绍的高品质彩色低辐射 SUPER VGA14" 彩色显示器。

1、低辐射：采用低辐射设计，对使用彩色显示器的人无伤害。特别是对使用大量彩色显示器的单位。如期货交易市场、股票交易市场、大量彩色显示器的计算机房。采用该显示器能使辐射控制在国际规定的安全范围内。

2、超高压保护电路：该显示器设计的超高压保护电路。使高压绝不会超出正常范围而使用显示器产生辐射。

3、过电保护电路：该显示器设计的过电保护电路，特别适合我国国情。一旦电网发生强烈波动。它会自动切断电源，以确保彩显不被烧毁。

4、符合 CSA、TUV、UL、FCC、GS 等多个国际标准。

♪满足 UL、CSA、TUV 国际标准的 PS / 2 电源

♪双声道音效卡(带 DOS、WINDOWS 音效驱动软件,两个小音箱)

♪主机板系列：

①IBM CHARLOTTE 工厂生产的 486SLC2 / 50 带 387 协处理器，64KCACH. 可扩展至 128K. 带 2 个 VESA 接口主机板

②IBM CHARLOTTE 工厂生产的 486SLC2 / 66 带 387 协处理器，64KCACH. 可扩展至 128K. 带 2 个 VESA 接口主机板

♪便携计算机伴侣——CITIZEN PN48 世界最小的 48 点阵打印机。

♪超想软件产品：

CXDOS5.0 网络及单机字符型中文系统软件。该版本具有大部分西文软件不用汉化，可直接使用如西文 FOXPRO2.5、QPRO、CLIPER、QBASIC、PCTOOLS、WORDSTAR4.0 等等。该软件汉字的输入接口与 WPS 相同，因此，除提供自然码这一方便易用的汉字输入法外，可挂接许多的汉字输入方法。CXDOS5.0 本身含金山公司授权使用的 CXWPS 汉字排版编辑系统。

以上产品，欢迎垂询。

地址：广州市德政北路 393 号

电话：3361567、3362849

传真：3361566

邮码：510055