

電腦 5 1991

中国软件行业协会会刊



注册 商標

CWY系列交流參數 高抗干擾穩壓器

CWY FAMILY PARAMETRIC
AC POWER REGULATOR

全國範圍產品責任保險



三相系列



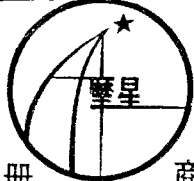
單相系列

高抗干擾
功能特殊
電腦必備
終身保修

廣州白雲山集團公司

廣州白雲山電源設備廠

廠址：廣州市沙河同和 郵政編碼：510515 電話：714403 電掛：0839



广州白云山电源设备厂

CWY 系列高抗干扰稳压电源

我厂是生产稳压器、变压器、变压器铁芯的省电子局定点厂，具有十多年的生产历史，是我国生产各类电源设备及其配件的骨干企业。所生产的 CWY 系列高抗干扰交流参数稳压电源是我厂 1984 年研制成功的国内首创稳压电源。通过国家技术部门鉴定，技术性能优越。比国内其他类型稳压电源有更明显的优越性。1986 年获国家科技进步奖。三相抗干扰稳压电源由中国科学院广州分院通过技术鉴定。1988 年获产品专利权。1989 年获中国科学院技术进步三等奖。1991 年获广州市优质产品奖。

本厂是生产 CWY 系列高抗干扰稳压电源的最早厂家，系列齐全。近年来经过工程技术人员研制攻关，对噪音、空载电流和漏磁干扰都取得了相当完善的解决，所以我厂产品与市场上同类型产品相比有更优越的性能，赢得了国内计算机用户、使用高精仪器设备等企、事业单位的信赖和广泛的应用，为我国电源系列发展作出卓越贡献。

购买时，请认明商标，提防有人剽盗本厂技术制造伪劣产品，使用户造成不必要的损失。本产品已向中国人民保险公司办理了全国范围产品责任保险，用户可放心使用。

单相抗干扰稳压器系列

一、型号及规格

型号	350	500	1k	2.2k	3.2k	5.2k	10k	15k	20k	30k	40k
容量	350VA	500VA	1kVA	2.2kVA	3.2kVA	5.2kVA	10kVA	15kVA	20kVA	30kVA	40kVA

二、主要技术参数

●输入电压单相交流 220V 50Hz

●电压稳定度

输入交流电压范围	输出交流电压稳定度
176~264V	$\Delta U_{出} < \pm 1\%$
160~264V	$\Delta U_{出} < \pm 2\%$
140~300V	$\Delta U_{出} +2/-7\%$

全國範圍產品責任保險



- 无过压危险，有自动短路保护特性
- 总谐波失真度 $< 4\%$ (开关式电源负载)
- 对电网振铃干扰或尖脉冲干扰抑制能力符合国际计算机电源要求
- 应变时间，输入电压跳变 $\pm 100Vp-p < 10 \sim 30ms$
- 有效功率可达到 80~90%
- 音频噪声低，位于 1.6 米距离处 50dB

三相抗干扰稳压器系列

一、规格：3kVA、6kVA、10kVA、15kVA、30kVA、40kVA

二、主要技术参数：●干扰脉冲抑制：输入脉冲 $< 400Vp-p$ ，输出不被发现；
输入脉冲 $> 1000Vp-p$ ，输出 $< 100Vp-p$ ；

●电压稳定度：输入 $-27\% \sim +36\%$ ，输出 $-2 \sim +1\%$ ；

●音频噪声低：位于 1.6 米距离处 52dB；

●效率高：89%，比一般电子交流稳压器效率高 20%，这对节能很有意义。

本单相、三相电源已与 VAX II / 750, VAX II / 785, PDP11 / 44 连机试验。运转正常，性能良好，广泛适用于计算机，自动控制设备，电子显微镜，X 光 CT，核磁共振断层扫描仪等高、精、尖设备等。与国外同类产品性能相等。

厂长：贝远娥

厂址：广州市沙河同和

邮政编码：510515

开户银行：广州农行白云营业所 帐号：80-431032 电挂：0839 电话：714403

注册商标

质量三包

欢迎来人来函订购

多谢惠顾

高抗干扰

功能特殊

電腦必備

終身保修

《自然码》广东总代理

——电脑杂志社

欲试用《自然码》请找电脑杂志社

为满足广大用户的求知欲，本社与北京超想电脑公司
免费赠送《自然码》普及版软件，欢迎来拷贝。

欢迎试用、欢迎惠顾、欢迎邮购

产品名称:	零售价	批发价	读者优惠价
一、4.1 版自然码 (编号: SM001) 加密软盘+自然码手册+键贴+塑料包装盒	350 元	250 元	300 元
二、超想汉卡 (编号: SM004) 汉卡+程序盘+自然码手册+键贴+塑料包装盒 (安装后共占内存 103K) (汉卡上配: 固化 16 点阵一二级字库、一万五千条 词组库、防病毒系统)	580 元	410 元	500 元
三、自然码词库卡 (编号: SM005) 词库卡+程序盘+自然码手册+键贴+塑料包装盒 (安装后共占内存 25K) (此卡上配: 固化一万五千条词组库、编码表及多字 词组、防病毒系统)	580 元	410 元	500 元
四、显示字库及防病毒卡 (编号: SM006) (安装后共占内存 46K) 显示卡+程序盘+塑料包装盒 (汉卡上配: 固化 16 点阵一二级字库、防病毒系统)	280 元	220 元	250 元
五、显示字库卡 (编号: SM007) (安装后共占内存 46K) 显示卡+程序盘+塑料包装盒 (汉卡上配: 固化 16 点阵一二级字库)	200 元	160 元	170 元
六、防病毒卡 (编号: SM008) 防病毒卡+最新解病毒软件盘+塑料包装盒	180 元	150 元	160 元
七、超级排版卡 (含防病毒系统) (编号: SM009) 超级排版卡+防病毒系统+WPS 用户大全+程序盘 +塑料包装盒	1700 元	1400 元	1600 元
八、自然码手册	12 元		

地址: 广州市天河路科技东街 49 号

电话: 514304 邮编: 510630

把此标记复印剪下贴在汇款单留
言处或信封背后可享受优惠价。



思索電腦公司



中國惠普授權代理



11月开始在本公司举行HP产品展示活动



地址：广州天河五山路106号

电话：516745 516746

传真：516770

邮编：510630

广州市科教电脑商场微机配置及价格一览表

机 型 配 置	CT80286-16				说明
CPU	80286-16	80286-16	80286-16	80286-16	可配协处理器
频率	10 / 22MHz	10 / 22MHz	10 / 22MHz	10 / 22MHz	塔 / 卧式机箱、液晶显示
内存	1MB	1MB	1MB	1MB	可扩展至 4MB / EMS
软驱	1.2M+360K	1.2M+360K	1.2M+360K	1.2M+1.44M	任选
硬盘	40MB	40MB	40MB	100MB	3.5" 高速硬盘
显示	MDA720×350	EGA640×350	TVGA1024×768	TVGA1024×768	进口原装 CASPER
串 / 并	2 / 1	2 / 1	2 / 1	2 / 1	可选四用户卡
键盘	101 键	101 键	101 键	101 键	XT / AT / 386
资料	DOS、随机手册	DOS、随机手册	DOS、随机手册	DOS、随机手册	可按用户要求
软件	CCDOS-2.13H	CCDOS-2.13H	CCDOS-2.13H	CCDOS-2.13H	DOS3.3 DBASE III
价格	5500 元	7500 元	8000 元	9500 元	赠送机械式鼠标器
机 型 配 置	CT386DX / 33-64K / 128K Cache RAM CT486DX / 33-128K / 256K Cache RAM				
CPU	80386DX-33	80386DX-33	80486DX-33	80486DX-33	可配协处理器
频率	33 / 58MHz	33 / 58MHz	33 / 150MHz	33 / 150MHz	塔式机箱、液晶显示
内存	4MB	4MB	4MB	4MB	可扩展至 32MB / 64MB
软驱	1.2M+1.44M	1.2M+1.44M	1.2M+1.44M	1.2M+1.44M	任选 (日本 TEAC)
硬盘	100MB	200MB	100MB	200MB	Conner 3104, 3204
显示	TVGA1024×768	TVGA1024×768	TVGA1024×768	TVGA1024×768	进口原装 CASPER
串 / 并	2 / 1	2 / 1	2 / 1	2 / 1	可配 4 / 8 用户卡
键盘	101 键	101 键	101 键	101 键	386 / AT / XT
资料	DOS、随机手册	DOS、随机手册	DOS、随机手册	DOS、随机手册	可按用户要求
软件	XENIX 2.32	XENIX 2.32	XENIX 2.32	XENIX 2.32	DOS3.3 DBASE III
价格	15000 元	16500 元	20000 元	21000 元	赠送光电式鼠标器
打印机	EPSON LQ-1600K	4400 元	BROTHER 1724	3300 元	日本原装
UPS	SANTAK 500W	1500 元	1000W (在线式)	6500 元	美国或日本
主板	80286-16 / 1MB	900 元	80386DX-33 / 4MB	4800 元	台湾产
硬盘	3.5" 高速 40MB 带卡	1550 元	3.5" 高速 100MB 带卡	2700 元	Conner 3044, 3104
软驱	TEAC 1.2MB	450 元	TEAC 1.44MB	400 元	TEAC-360KB 390 元
彩显	14" EGA (5154H)	2500 元	14" TVGA (5156H)	2800 元	进口原装 CASPER

全国各地诚征代理

地址：广州市天河东路商业街东座 25-27 号

通信：广州市 1753 号邮政信箱

邮编：510620

电话：511197、515564

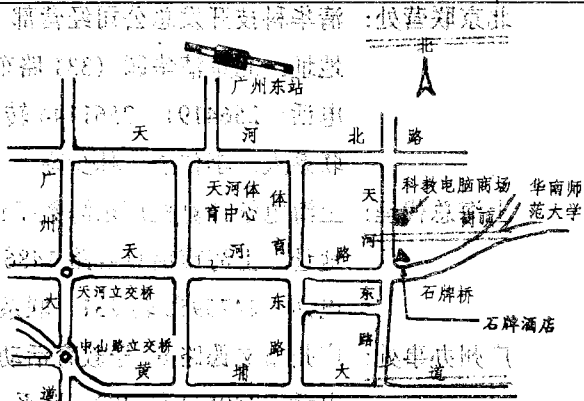
电挂：8360 FAX：511197

开户银行：广州中国银行天河支行

帐号：271-015110027

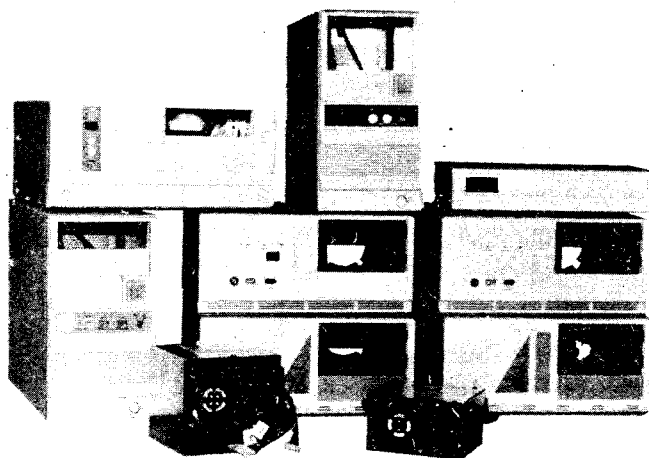
代办托运 免收运费 保用壹年

省外用户 来人购机 车票报销



中外合資奉獻機壳電源精品

攜手共進為你電腦錦上添花



※立式卧式等系列电脑主机箱

※适用于 PC-XT、286、386、486、终端机

※电脑开关电源 200W、220W、250W

※北京、广州备有大量现货

歡迎垂詢訂貨、請與下列單位聯絡

汕頭教學儀器設備公司 汕頭市教學儀器廠

总部: 汕头市中山路一中巷 电话: 251872 240737

图文传真: 551050 联系人: 林增鹏

北京办事处: 北京市海淀区城府路 45 号清华园宾馆 2207 房

电话: 2566611 转 254 联系人: 李仁杰

北京联营处: 清华科技开发总公司经营部

地址: 北京清华园 (331 路车清华园站斜对面)

电话: 2564191 2561144 转 3698、2091

联系人: 李仁杰 林少南

上海总代理: 上海电子计算机厂东海数字设备经营部

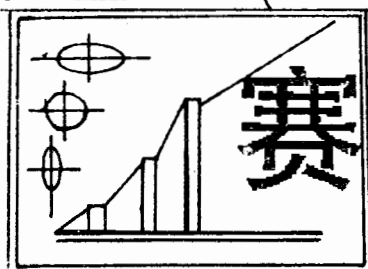
地址: 上海市南京西路 1486 号 图文传真: 2472298

电话: 2475950 转 231 联系人: 潘永柏 陈 民

广州办事处: 广州市文德路 69 号教育活动中心 302 室

电话: 340511 转 302 联系人: 陈国煌 梁德海

广州经济技术开发区



赛宝星河公司

[美国 TANDON 微机南方代理]

本公司位于广州天河科技街；由机电部第五研究所与佛山市无线电八厂联合经营。是一家专业从事办公自动化设备及技术咨询、维修以及软件开发的综合性计算机公司。公司拥有雄厚的经济实力、强劲的技术后盾以及完善的售后服务。公司还特约代理美国原装 TANDON 微机。

主要经营有

- (1) 承接计算机网络工程开发、安装、调试以及维护
- (2) 承接 CAD 系统以及排版系统
- (3) 特约代理美国原装 TANDON 微机
- (4) 销售美国原装 AST、COMPAQ、PS / 2 以及各种兼容 286、386、486 等计算机
- (5) 销售各种打印机、UPS 电源以及各种电源配件
- (6) 销售计算机资料及软件

地址：广州石牌科技街 63 号

电话：510293 510294

邮编：510630

机箱电源总汇

深圳航空電腦設備公司廣州辦事處

廣州南秀電腦技術開發服務部



电脑天地

== 广州天河电子器材经营部 ==

最优惠的价格

高品质的产品

完美的售后服务

主 营:

各类 486 386 286 PC 主板
3COM 系列 NOVELL 系列 (网卡、网络配件)
高档进口机箱
AST COMPAQ ARTLAN 系列整机

电脑配件

功能卡、硬盘、键盘、UPS 电源

办公设备

复印机、传真机、碎纸机、四通打字机、油印机、收款机

全国联系地点:

广州: 广州市天河科技东街 96~98 号, 电话: 514675 FAX: 514675

邮编: 510630

深圳: 深圳市兴华大厦八楼 电话: 359413 350483-442

FAX: 359413 邮编: 518031

北京: 北京市中关村白石桥路 27 号中央民族学院 6 号楼 4 层

电话: 8328877-2440

肇庆自动化（广州）公司简介

肇庆自动化（广州）公司是一家致力于计算机技术开发、网络工程、销售及维修服务的专业电脑公司，凭着雄厚的技术力量和综合实力以及公司同仁的通力合作及努力下，在用户中树立起服务良好、诚实可靠的信誉。我公司销售的计算机系统均经精心挑选，集国际最先进之科技精华，充分考虑了软、硬件的兼容性，每套系统均经严格、苛刻的测试，以确保机器性能的稳定，给予您最大信心。

本公司竭诚为阁下提供计算机系统全套服务如下：

* 计算机系统：

经营销售 TEC、COMPAQ、AST、IBM 全系列（包括 PC、286、386、486）微机，以及 MOTOROLA、BULL、DEC 等小型机系统。

TEC 系列微机特别介绍

TEC	8088-2	286-16	386SX-16/20	386/33	486/25
主频	12MHz	16/20MHz	16/20MHz	56MHz	115MHz
内存	640K	1M	2M	4M	4M
硬盘	/	40M	40M	100M	200M
高速缓存	0	0	0	64KB	64KB
串并口	一串一并	一串一并	二串一并	二串一并	二串一并
软驱	360KB×2	1.2MB+360KB	1.2MB+360KB	1.2MB+360KB	1.2MB+360KB
显示器	720×350 MDA 单显	1024×768 VGA 彩显	1024×768 VGA 彩显	1024×768 VGA 彩显	1024×768 VGA 彩显
机箱	小卧机箱	豪华型卧式机箱	豪华小立式机箱	豪华型立式机箱	立式带门机箱
键盘	101KB	101KB	101KB	101KB	101KB

以上各种配置（如硬盘、内存、软驱、显示器等）均可根据用户需要调整；

现有多功能 IDE 软硬卡批发价：120 元/块，KPT-102 键盘，批发价：195 元/只出售、各种接口卡长期现货供应，国内保修，并诚征各地代理。

* 网络工程

承接 Novell Netware，3+网络工程，经销网卡及网络配套系列产品，作为 JBC 之广州代理，诚意推荐集网卡和网卡于一身的 JBC 汉字网卡。

* 外围设备

提供各种打印机、绘图仪、数字化仪、大屏幕监视器、各种接口卡，特别推出 RICH 500/500XT 票据打印机，SKYFAX 电脑传真卡，MICROTEK 激光打印机等超值产品。经销 ROCHI 日本出产之“高密度”色带，其自身拉力比一般性尼龙色带为高，其驳口采用超声波焊接所以不易造成打印机头损耗及断针等问题，长期备有各种带芯和带架供应。

* 软件系统：

提供多种办公自动化中文排版系统（包括华光、MSOA、PUC 金山等），并承接各类软件工程的开发设计任务，令您的计算机如虎添翼。

* 维修服务：

凡属本公司的直接客户，均享受壹年免费保修，24 小时响应，48 小时到位的售后服务，本公司还为广大计算机用户提供维修服务，维修各类 PC、286、386、486 主板；维修各类打印机，特有 M1724 配件：（皮带轮，电源变压器，打印头电缆，驱动片，胶辊，步进电机等），并配有各种打印针，品种齐全。

欢迎各单位来函来电索取详细资料报价。

单位：肇庆自动化广州分公司

地址：广州东风东路 751 号（原振兴公司）

电话：751375 785037



9 针 打 印 机 总 汇

※江门计算机厂独家生产

TX-800

TX-850 (九针打印机的
价格, 24 针打印机的
效果)

质量可靠, 保修壹年
长期供应配件

※日本原装打印机

EPSON LX-800 (中国代理)

CP-135 九针打印机

CPF-80 九针打印机

LP-1516 九针打印机

LQ-1600K 24 针打印机

M-1724 24 针打印机

联系地址: 广州市天河科技东街 74 号 开户银行: 中国银行广州天河支行

联系电话: 517167

帐号: 271-018100175

邮政编码: 510630

TX-850: 九针打印机的价格, 24 针打印机的效果

TX-850: 九针打印机的价格, 24 针打印机的效果

TX-850: 九针打印机的价格, 24 针打印机的效果

TX-850: 九针打印机的价格, 24 针打印机的效果

TX-850: 九针打印机的价格, 24 针打印机的效果

TX-850: 九针打印机的价格, 24 针打印机的效果

TX-850: 九针打印机的价格, 24 针打印机的效果

TX-850: 九针打印机的价格, 24 针打印机的效果

TX-850: 九针打印机的价格, 24 针打印机的效果

TX-850: 九针打印机的价格, 24 针打印机的效果

TX-850: 九针打印机的价格, 24 针打印机的效果

TX-850: 九针打印机的价格, 24 针打印机的效果

TX-850: 九针打印机的价格, 24 针打印机的效果

TX-850: 九针打印机的价格, 24 针打印机的效果

中国科学院广州电子技术研究所

智通电子技术公司

(CCS IBM PS/2 中国南方维修中心)

本公司是一个多年从事研究、开发和应用计算机网络系统,工业自动化控制与监测系统,计算机信息处理系统、遥测遥控系统,电子工程(包括卫星接收天线、公共天线等)的高技术单位。以中国科学院为后盾,拥有雄厚的经济实力和众多的各类高技术人材。曾经承接并完成了卫星地面站自动检测系统,大型水库水文遥测系统,电站自动监控系统,由二十七台 PS/2 微机构成的 3⁺以太网计算机信息管理系统等。有多年计算机应用的经验,可在较短时间内为用户提供电子系统工程的各种研究、开发、应用和系统设计。本公司还是 CCS IBM PS/2 中国南方维修中心。代理 IBM PS/2 微机, RISC SYSTEM/6000 工作站, AS/400 小型机在国内的销售机维修业务。

公司竭诚为各界用户服务,并提供:

一、负责对购买 AS/400 小型机的用户进行出国培训前的初级培训。

二、人材培训,技术咨询,各种计算机资料和软件。

三、3⁺以太网, Novell 网, Token Ring 网, D-Link 网等计算机网络系统的软、硬件开发、安装、销售。

四、工业自动化系统,遥测遥控系统,电子工程等软、硬件开发、研究、设计、工程。

五、手写汉字识别系统,汉字文本识别处理系统,汉字排版及轻印刷系统。

六、中国科学院的各种新产品,各种进口元器件、数字仪表、通讯设备、示波器等。

七、销售美国 AST, COMPAQ, PS/2 及各种兼容 486, 386, 286, PC 微机, AS/400 小型机, IBM RISC SYSTEM/6000 工作站等。

八、高抗干扰电源, UPS 不间断电源,各种打印机、中英文打字机,各种电子电脑配件。

公司地址:广州市先烈中路 100 号大院内电子研究所一楼

公司电话: 753245-4131, 1107, 775600-610

邮政编码: 510070

图文传真: 753247

联系人: 李海华, 姚智鹏, 李建

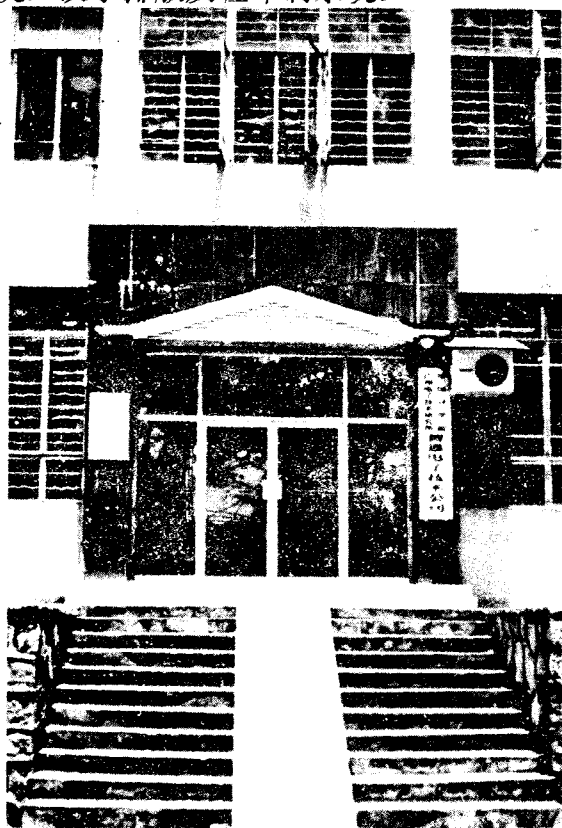
门市部地址: 广州市天河科技街第二座

171/176 号

门市部电话: 510450

邮政编码: 510630

联系人: 欧德



“Look and Feel”的争论及其影响

—计算机软件相似性引起的版权纠纷

华南师大微电子所 王桂海

摘要

软件开发过程中, 版权成立的条件是独创性, 即行为独立或形式独立。相似性是形式独立的否定。有一类相似是“Look & Feel”(外观与感受)引起的。“Look & Feel”只有缩小保护范围, 才有利于应用软件的改革与推广。

我们已对计算机的软件保护进行过多次讨论, 涉及到软件的版权和专利权的保护范围^[1], 版权与兼容软件的保护问题^{[2][3][4]}, 版权保护的法律模型^{[2][3]}等等。本文将讨论软件开发中在版权问题上引起争论的另一个焦点即“Look & Feel”(外观与感受)。

(一)

著作权(版权)保护那些由作家及其它有创作能力的人创作的原作品, 只要其作品是原作, 意思就是独立创作, 就可以取得著作权, 即令已有极为相似的作品存在。但版权保护的, 只是这些原作的表达形式, 而不延及作品的思想内容。中国《计算机软件保护条例》规定, “对软件的保护不能扩大到开发软件所用的思想、概念、发现、原理、算法、处理过程和运作方法。”^[5]

作品被承认是独创的, 则可以取得版权保护。形式上与已有的作品不同(我们称之为形式上独立, 记为 I_f), 或者形式上与已有作品相似甚至极为相似, 但可以证实创作过程是独立的(我们称之为行为独立, 记作 I_a)都可以被承认是原作, 都可以取得版权。这样, 当我们把取得版权记为事件 Cr , 被承认是独立创作记为事件 Ci , 则有:

$$Cr = Ci = I_f + I_a \quad (1)$$

I_f 和 I_a 都是取得版权的充分条件,

要否定一个作品具有版权(Cr , 指控其侵权), 就要指出它是“非独立创作”即 \overline{Ci} , 所以:

$$\overline{Cr} = \overline{Ci} = \overline{I_f + I_a} = \overline{I_f} \cdot \overline{I_a} \quad (2)$$

从(2)可以看到, 原告应从形式相似 $\overline{I_f}$, 和行为上不独立 $\overline{I_a}$ (被告与原告作品有过接触)同时举证。 $\overline{I_f} \cdot \overline{I_a}$ 是法院对侵权控诉受理的必要而充分的条件。

根据(1), 辩护者只须证实 I_f 或 I_a 中任一项为事实, 便可以驳倒对其侵权的控诉。设已有软件 A 的作者 Aa , 指控新出的软件 B 的作者 Ba : 1. 与软件 A 有过接触; 2. B 与 A 在形式上有足够的相似。那末, Aa 的指控是“完备”的。而 Ba 则可以从 I_a 或 I_f 中任一项或同时从两项中进行申辩和反驳。

对 I_a (行为独立)的证明, 现在流行的做法是证实本软件来自 Cleanroom(隔离室), 对此, 我们将另作专题

讨论。而从 I_f (形式独立)来申辩, 亦即因“相似性”而引起的争论, 则由于国内、国外都没有一个明确的“形式”与“思想”的分界线(The demarcation line between idea and form of expression), 使问题弄得十分复杂。我国《计算机软件保护条例》第三十一条规定: “因下列情况之一而引起的所开发的软件与已存在的软件相似, 不构成对已经存在的软件的著作权的侵权: (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) 由于可供选择的表现形式种类有限。”这里, 给出了三个避免受相似性指控的方面, 但不是具体的划分形式与思想的界线。我们曾经提出过一个关于软件相似问题的法律模型, 即 N_L - N_c 模型^[2], 在这个模型里编码层是属于表达形式, 一定受保护; 而思想层, 一定不受保护, 这是毫无疑问的。但这中间, 保护范围有多大, N_L (法律分界线)放在那里, 是一个“弹性”的问题, 有许多不同的观点。目前, 综合国外(主要是美国)和国内提出判别形式与思想的准则, 我们比较熟悉的有以下几种:

1. “S.S.O”准则。即把受版权保护的“形式”的范围, 扩大到结构(Structure)、步骤(Sequence)和组织(Organization)层次, 即在这些层次上相似, 就可以认为是形式不独立。这是美国法院1986年8月4日判决“Whelan对Jaslow”案提出的。但S.S.O准则在美国及国际上引起的争议很大, 主要认为这一准则把版权保护的圈定得过宽, 扩大到专利保护领域。^{[6], [7]}

2. “VML”准则。这是本文作者于1989年提出来的^[2], 即把两个软件的空间 V (总体结构), 子程序和模块集合 M (中间层次)和语言 L (包括数据成份, 运算成份, 控制成份和传输成份)作比较, 如果这三方面有足够的区别, 则可以说明两个程序是“不相似”的。这个方法已用于判别国产 ZIDOS 软件与 Apple II DOS3.3 的差异, ZIDOS 作为具有版权的软件已经通过鉴定^[8]。但这一方法尚未作为法律上的案例来应用。

3. “实质相似”(Substantial Similarity)和“功能相似”(Function Similarity)。提出“实质相似”准则的人, 主张从输入、输出、数据结构、逻辑系统方面来比较; 而主张“功能相似”则要求把功能相似而引起的程序体系上相似视作侵权。这更扩大了版权的保护范围。1988年7月15日美国

佛罗里达州南部地区联邦地方法院对“Pearl 系统公司对 Competition 电子公司”一案的判决,便是一例^[7]。

4. “look & Feel”相似。下面将专门讨论。

(二)

就广大计算机用户来说,总希望能很方便地使用程序。为了满足用户这一要求,销售商一般创造了所谓“用户友好”的程序。要制造出这种用户接口(user interfaces)是要有相当投资的,因而,制造者理所当然地要求这些制品受法律保护。但另一方面,各种竞争者都希望把这些接口移植到他们自己的软件上去,使自己的软件在“外观与感受”(Look & Feel)上与用户已经用惯了的软件没有什么不同,这当然是为了新的软件容易打开市场。但往往也正是因“Look & Feel”相似而引起诉讼。

这个问题的关键是用户接口或者说是人机界面,一般都有很大的共性,有点近乎“标准”。如果以此作为形式相似的指控,是否妨碍了各类以友好界面为基础的软件的真正革新?——这就是我们要讨论的问题。

我们先来分析美国著名的空前棘手的有关“Look & Feel”的案例: Lotus 对 paperback 一案。

Paperback 公司制造了一种外观上和 Lotus 著名的电子数据表格软件 1-2-3 非常相似的电子表格软件 VP-Planner (类似的软件 Broland 公司和 Snata Curz 公司也已制造,现在 Lotus 也正对他们起诉)。1989 年 Lotus 公司对 Paperback 公司提出起诉。案件的焦点是: Lotus 1-2-3,特别是它的“用户接口”是否应得到版权保护,被告的 VP-Planner 是复制了 1-2-3 受保护的形式还是只采用了不受保护的思想?要处理好这个问题并不容易。正如法院所注意到的:如果把版权保护范围划得太广,就给第一个写出这方面应用程序的人以过多的专利(垄断)权,这将妨碍了其它人去改进产品(即他人不敢去着手改进这类软件,因为容易造成侵权);如果保护范围划得太窄,程序工作者的成果很易被复制,又会影响了创作者的积极性,使人们不愿去进行富有成效的革新。^[9]

原告委托代理人 Hank Gutman 指出:双方都承认 VP-Planner 与 1-2-3 相似,但这两个程序相似到了这样的程度:“用户如果已会用 1-2-3,便无须再翻开 VP-Planner 的手册,就会使用 VP 这个程序”。^[10]所以,是典型的侵权。

这个案子不是程序内部结构之争而是用户接口、屏幕菜单之争。

被告的委托代理人 P. Gupta 辩解说:之所以认为 1-2-3 的用户接口不受版权保护,是因为纯粹结构方面的东西,是不能取得版权的,1-2-3 正是这样一种以数字来工作的结构。屏幕显示只有嵌入受版权保护的程序之后,才能取得版权,而 1-2-3 只是根据用户提供的公式再弄出一些数据。这种方法,人们在 1-2-3 问世前很久便知道了(即是属于公有知识范围)。^[10]

被告还提出:用户接口应当是不受保护的“标准”,

1-2-3 特别是它的菜单结构和宏指令程序,事实上已经是所有电子表格的工业标准。因此,他们除了复制 1-2-3 的表达要素外,别无其它选择,如果不复制这些要素(包括宏指令),已经受过 1-2-3 训练并精心编制出在 1-2-3 上运行的宏指令的用户,将不会愿意使用 VP-Planner 软件,而导致这个软件在商业上(注意,这里被告指的是商业上而不承认是技术上)的失败。^[9]被告的观点很清楚,他们是以“反垄断法”,保护竞争,提倡标准化为其立论基础的。

美国版权局官员 Phil Gill 在 1989 年 8 月 25 日给争议各方的一封信中指出:“版权不能对名字、标题以及类似于表格中行、列的题头,简单的检验单等提供保护,也不能对空白表格的排列方法、印刷格式提供保护。因而,一般来说,在一个特定的形式中,由词组或简单短语构成的屏幕菜单及类似的功能接口是不能作版权登记的。”^[10]这一观点明显地倾向被告一方。

美国联邦地方法院于 1990 年判原告胜诉,认为:不能因为“兼容性”和“标准化”,便剥夺了原作者对已结合到作品中去的表达形式所应拥有的“有限垄断权利”。被告不了解商业上成功的软件如 Excel,并非与 1-2-3 菜单格式或宏指令上相兼容。被告关于“标准化”的辩解也是站不住的,版权的目标之一,就是保护作品形式而鼓励改革,因而,对某种思想的表达形式越是革新,对这种表达形式的保护便越重要。如果认为 1-2-3 已经改进到在事实上已成为工业标准而可以自由去复制原告的形式,实际上是侵权。如果认为版权只保护那些一般性的改进而把显著的改革当作公有知识不加保护,那是对版权的曲解。最后,法院驳回被告关于标准化的引证,认为在法律条款上还没有这样的先例:把未经法律上承认的标准化宣布为为了公共利益最高利益所必须引用的。^[9]

从这个案例可以看到被告,原告、美国版权局(官方)和法院对这个案子的观点上的分歧。综合起来是:

被告方的意见:版权应保护改革,不能让一家垄断市场;接近于标准化的东西应不受保护;简单的格式以及调用程序,属于方法而不是程序形式本身。“Look & Feel”就是这样一个方法运用的反映,因而不能以“Look & Feel”相似与否来判断是否侵权。

原告(胜诉)方面的理由是:菜单及程序调用方法都应保护,这是为维护第一个开发者的利益所必需的。不能轻易承认标准化,不能把菜单一类结构形式列入公知范围。

(三)

这个案子给我们的启示:

1. 应认真研究在保护版权和反对垄断之间如何平衡才是合适的。版权是通过对智力作品的保护,促进创作繁荣,促进改革进步。它既防止剽窃,又鼓励竞争和反对技术垄断。产品(包括软件)进入市场,一般要经过发明(invention)、改革(或改进, innovation)、推广

(diffusion) 几个阶段。发明是发明者个人的贡献;但改进与推广,则往往不会停留在原发明者身上而是由其它人去完成。艺术性作品的衡量标准比较抽象,创作自由度很大,对某一作品“改造”之后产生的效果到底是好是坏,不易掌握,有时甚至会认为是损害了原作品。广大读者一般不欢迎对原著作出什么“改进”(当然,也有个别是相对成功的,如金圣叹改百二十回《水浒》为七十回;高鹗续八十回《红楼梦》为百二十回)。所以,文学作品对原著的保护是明确的。但技术性作品则不同,改进之后的效果,很容易甚至可以用量来衡量。一旦改进,社会效益便很显著,也是用户所欢迎的。改进者只是利用原著的某些因素,重新投资开发,应当是允许的。如果任何改进都要经原作者同意的话,改进与推广都是难于进行的。因为发明者在取得垄断地位之后,未必同意由他人来实现改进,但这却有悖于技术进步和产品的自由竞争。以 Lotus1-2-3 为例,1990 年,该软件的售价为 495 美元;而 Borland 公司的 Quattro Pro 电子数据表只是用 Lotus 的菜单命令字来代替原有的命令,便能干 1-2-3 可以干的一切,所占的内存更少,图象也更好,售价只须 99 美元。Lotus 当然不会同意这种改进,但用户却是欢迎的。所以,把版权扩大到 Look & Feel 这个层次,未必对技术发展和用户带来好处。

2.对“标准化”的引用,也存在着这样的问题。什么样的内容应列入标准或公知范围,同样应在用户与原作者利益之间选择合适的平衡点。用户希望实行标准化,但这一方面将削弱发明者的垄断地位,另一方面又会限制更进一步的改进(如果这种改进已起出标准化的规定的话)。所以,计算机软件开发中“怎样能够在标准化和受到保护的权益之间进行调解?怎样才能截然区分公共利益和私人利益。”^[11]“Look & Feel”是不是应列入标准,便是法律工作者要回答的问题。

3.“Look & Feel”是方法、工具还是用以表达程序内容、思想、概念的形式,要慎重考虑。“Look & Feel”只是用户与程序之间的接口,只是反映程序如何与用户进行“友好合作”的界面。主要牵涉的是屏幕菜单中的实际用词和形式结构以及发出命令的按键和功能键。这些“接口”毕竟不是程序编码层次,并不是直接表达思想、概念的形式,而是近乎方法。在这方面,应当鼓励兼容,形成相对的,事实上的(de facto)标准化。人机界面可选择的渠道很有限,够得上表达形式的分量很少。这一切都说明在这个层次上由独家来垄断是不宜的。

后记:本文作者与美国计算机法律专家 M.D.Scott 先生作过一次讨论,蒙他提供了部分尚未发表的资料(参考文献[9]),特此表示感谢。

参考文献

- [1]王桂海 计算机的法律保护 电脑 1991 第 3 期
- [2]王桂海 略论计算机软件版权保护与兼容软件 计算机世界 1990 (23) (24)

[3]王桂海 试论计算机软件保护与兼容性的关系 计算机世界 1991, (22) 97 (增版)

[4]王桂海 软件开发中的版权保护 电脑 1991 第 4 期

[5]《计算机软件保护条例》第七条

[6]P. Samulelson Is copyright Law Steering the Right Cousse? IEEE Software sept.1988

[7]陈励松 从美国判例看程序保护范围 计算机信息报 1989 年 12 月 26 日 23 版

[8]王桂海,龙庆华 ZJDOS 的版权保护分析(将发表)

[9]M.D.Scott “Look and Feel” and other Emerging Issues: The effect of standardization in the computer Industry on the Development of Computer Law (将发表)

[10]R.C.Johnson Spreadsheet copyrights may prove hard to get EET.Sept 4, 1989

[11]计算机世界国际动态版 90.10.24

编后话:本刊对计算机软件的法律保护专论文章到本期暂告一段落。广大读者对这个论题有什么看法、疑问、心得体会,或者希望还登那些论题,欢迎来信、来稿。

另一种消除分页符号方法

广西南宁三〇三医院信息科 黄大江

九一年第三期《电脑》杂志上介绍杨海军同志的“分页符号的消除二法”文章,两法皆可行,其实利用 WS 本身的查找替换功能 F5,就可方便地消除分页符号,即当用 D 命令编辑文件存盘回到主菜单后,可用 N 命令再次编辑该文件,找出被 WS 自动加上的分页符号(0D 8A),使其换成硬回车符(0D 0A)或软回车符(8D 0A)即可,方法如下:

在 WS 主菜单下,用 N 命令编辑被自动加上分页符号的文件,按 F5,当显示屏上显示“查?”后输入 P 然后按下 Alt 键不放右边小键盘上输入 138 数码及回车(找 0D 8A),当显示屏上显示“换为?”后再输入 P M P J 及回车(换成 0D 0A),或者当显示屏上显示“换为?”后按下 Alt 键不放右边键盘上输入 141 数码再输入 P J 及回车(换成 8D 0A),最后输入 NG 回车(自动更换所有的),然后按 F1 存盘退出,至此,该文件所有分页符号全部更换成硬回车或回车符。说明: P 即是按下 Ctrl 键不放再按 P 键, ^M 亦是用相似的方法输入。

另外,WS 本身已提供专为编写源程序的 N 命令,所以当编写源程序时应使用 N 命令编辑,就可以避免由于有分页符而使程序出错。

BASIC 语言动画技术应用实例

江西拖拉机发动机厂 黄焕如

内容提要: 本文从实用的角度讨论了 IBMPC-BASIC 语言动画技术的实现, 并由简到繁、从字体动画移动到图形动画变幻列举了五个实例, 这些实用程序简单易懂、动感性强、画面漂亮、可移植性好, 很适合学习 IBMPC-BASIC 语言动画技术的读者使用, 这些程序稍加修改也可应用于一些应用软件封面的制作。

计算机的图形学中的动画技术可以广泛应用于动画片、辅助教学、游戏、应用软件封面设计等方面。动画技术是指屏幕上显示出来的画面或其中一部分, 能按一定的规则及要求, 在屏幕上移动或变幻, 从而使计算机显示的图形能动态变化的技术。IBMPC-XT 微型机 BASIC 语言 (以下简称 BASIC) 是一种简单易学、内容丰富和使用方便的程序设计语言, 尤其是该语言具有较强的图形功能, 为许多程序员所喜爱。

彩色图形显示器有两种基本的工作模式, 即字母数字模式 (A/N) 和图形显示模式 (APA), 这两种工作模式可通过 SCREEN 语句来选择。字母数字模式中的图形是由特殊字符或图形字符构成的, 这种字符动画常见于一些游戏画面; 在图形显示模式中, 显示器屏幕上每个点 (或称“像素”) 均由程序控制其亮度和颜色, 因而能显示出质量较好的图形, 使用比较广泛, 本文讨论的就是在 IBMPC-XT 微型机上彩色图形显示模式中的动画处理。

在图形显示模式中, 按照分辨率的高低又可分为高分辨率、中分辨率和低分辨率三种模式, 其中高分辨率模式屏幕显示: 每帧 200 条线, 每线 640 个点, 每点只能取黑白两种颜色; 中分辨率模式屏幕显示: 每帧 200 条线, 每线 320 个点, 每点可以取 4 种不同的颜色; 低分辨率模式屏幕显示: 每帧 100 条线, 每线 160 个点, 每点可以取 16 种不同的颜色。目前 BIOS 中的驱动程序不支持低分辨率模式, 而高分辨率下几乎没有彩色, 因此一般均选择中分辨率模式。

在图形显示模式中, 一般有两种产生动画的方式。一种是利用 GET 和 PUT 语句来保存和取出图形, 称为“存取式”; 一种是利用 PSET、LINE、DRAW 或 CIRCLE 等语句来描绘点、线或圆的动画图形, 称为“轨迹式”。

一、“存取式”动画技术

为了方便使用, 下面介绍“存取式”中 GET 语句的格式:

GET (X1, Y1) - (X2, Y2), Z

其中 (X1, Y1) 和 (X2, Y2) 分别是待保存的一个矩形区域信息的左上角和右下角点的坐标, Z 是一个整形数组, 该语句的功能是把指定矩形区域内每个像素的彩色码存放于数组 Z 中。必须指出, 矩形区域的坐标应该容纳

待保存在屏幕上的全部信息, 数组 Z 的大小则必须足够容纳矩形中所有像素的彩色码。一般矩形区域坐标稍取大一些无妨, 而数组 Z 的最小值应为:

$$\text{MIN} = 4 + H * \text{INT}((W * B + 7) / 8)$$

其中: H 和 W 是矩形区域的高和宽, B 是每个像素对应的二进制数目, 中分辨率下 B 取 2, 高分辨率下 B 取 1。

与 GET 语句对应的是 PUT 语句, PUT 语句的格式是:

PUT (X, Y), Z[, FS]

其中 (X, Y) 表示矩形的左上角坐标, Z 是由 GET 语句保存信息的整形数组名, FS 为运算选择方式 (PSET、PRSET、XOR、OR、AND), 该语句的功能是把已保存的动画对象在屏幕上指定位置显示出来。

程序实例一就是这两条语句的简单应用, 该程序中第 30 行定义一整形数组, 用来存放需移动的汉字信息, 第 40 行和第 180 行均在 2.13D 以上版本汉字操作系统下 (16 行显示, 以下同) 调取 24 点阵汉字, 第 60~150 行将“设计:”和“黄焕如”字符串先后左右移动, 然后一起向上移动定位, 第 160~170 行将 24 点阵的“工资汇总和分配系统”汉字字符串整体向下移动定位, 第 190~300 行是美化用的画边框、画小园和填色等。

```
10 REM IBMPC-BASIC 语言动画技术应用实例之一
20 SCREEN 1:COLOR 1,0:CLS:KEY OFF:DEFINT A
30 DIM AC(1600)
40 PRINT CHR$(14) "[@50:10:0*2 工资汇总和分配系统]"
50 LOCATE 9,2:PRINT "设计:"
60 GET(0,90)-(66,110),AC%:LOCATE 9,2:PRINT ""
70 FOR X = 4 TO 120 STEP 2:PUT(X,90),AC%,PSET:NEXT
80 FOR X = 120 TO 60 STEP -1:PUT(X,90),AC%,PSET:NEXT
90 FOR X = 60 TO 100:PUT(X,90),AC%,PSET:NEXT
100 LOCATE 9,21:PRINT "黄焕如"
110 GET(150,90)-(250,110),AC%:LOCATE 9,21:PRINT ""
120 FOR X = 148 TO 180 STEP 1:PUT(X,90),AC%,PSET:NEXT
130 FOR X = 180 TO 148 STEP -1:PUT(X,90),AC%,PSET:NEXT
140 GET(80,90)-(250,110),AC%
150 FOR X = 0 TO 20:PUT(80,90-X),AC%,PSET:NEXT
160 GET(20,1)-(300,40),AC%
170 FOR X = 0 TO 30 STEP .5:PUT(20,0+X),AC%,PSET:NEXT
180 PRINT CHR$(14) "[@M_50:140*3 江拖发动机厂微机室]"
190 LINE(26,20)-(293,179),3,B
200 FOR Y = 5 TO 195 STEP 23
210 CIRCLE(4,Y),2,3:CIRCLE(298,Y),2,3
220 CIRCLE(12,Y+10),5,3:CIRCLE(305,Y+10),5,3
230 CIRCLE(18,Y+20),3,3:CIRCLE(315,Y+20),3,3
240 NEXT
250 FOR X = 18 TO 290 STEP 31
260 CIRCLE(X,5),2,3:CIRCLE(X,195),2,3
270 CIRCLE(X+14,10),5,3:CIRCLE(X+14,190),5,3
```

```
280 CIRCLE(X+26,15),3,3:CIRCLE(X+26,185),3,3
290 NEXT
300 PAINT(0,0),3,3
```

二、“轨迹式”动画技术

“轨迹式”动画技术是指通过描绘一系列的点、线或园，然后又擦去它们而产生的动态感。一般来说，该方法每次化在图形的建立、擦去、重画所需的时间将随着图形的复杂程度而增加，如果运动轨迹的计算比较复杂，则动画的速度将明显降低，因此该方法仅仅适用于动画对象比较简单的场合。

程序实例二就是“轨迹式”的简单应用，该程序中第 30~50 行画出两个椭圆并在中间部分涂色，第 140~220 行为一子程序，其功能是以一定的节拍使椭圆中间不断变换各种颜色、不同椭圆度的园框，同时把第 70 行定义的字符串中一个个汉字从屏幕外依次移入画面。顺便指出，汉字移入的动画感实际上也是利用“轨迹法”，显示、擦去、再在下一位置显示，如此循环就好像汉字在移动一样，当然这种简单的动画技术不需要使用 BASIC 语言的绘图语句。

```
10 REM IBMP-C BASIC 语言动画技术应用实例之二
20 CLS:SCREEN 1:COLOR 0,2:KEY OFF
30 CIRCLE(159,65),120,1,,,4:CIRCLE(159,65),75,1,,,52
40 CIRCLE(159,65),121,1,,,4:CIRCLE(159,65),76,1,,,52
50 PAINT(50,50),2,1
60 PRINT CHR$(14)“[@50:120*1*5 固定资产管理系统]”
70 YS$ = “江拖发动机厂微机室”
80 K1=1:K2=10:K3=0:K4=10:K5=1
90 K=36
100 LOCATE 14,K:PRINT MID$(YS$,K1,2):K=K-2
110 GOSUB 140
120 IF K<>K2 THEN LOCATE 14,K+2:PRINT “”:GOTO 100
130 K1=K1+2:K2=K2+2:IF K1<>25 THEN 90 ELSE END
140 K3=K3+1:K4=K4-1:K40=K4/15
150 IF K3>3 THEN K3=1
160 IF K4<2 THEN K4=10
170 CIRCLE(159,65),44,K3,,,K40
180 FOR I=1 TO 50:I=I+1:NEXT
190 CIRCLE(159,65),44,0,,,K40
210 K5=K5+1
220 RETURN
```

三、“存取式”和“轨迹式”结合的动画技术

比较复杂的动画技术往往还必须把“存取法”和“轨迹法”结合起来使用，把动画对象保存在某一数组中，然后使其按一定的轨迹移动；比较复杂轨迹的计算可以利用 BASIC 语言中的一些三角函数，动画的效果更好。程序实例三就是“存取法”和“轨迹法”结合的简单应用，其中第 60~280 行是在屏幕上描画一台微型计算机的图样，第 330~350 行画一个涂色的园，存入数组后擦去，第 360~460 行使这个园按正弦轨迹运动。值得注意的是，第 30 行定义的字符串在高分辨率下存入数组，然后在（第 290 行）中分辨率下取出，从而简单地实现了一屏同时显示两种字体的效果。

```
10 REM IBMP-C BASIC 语言动画技术应用实例之三
20 DEFINT A,B: DIM M(280): DIM A(160): DIM JS(100)
```

```
30 YS$ = “1991.1”
40 SCREEN 2:LOCATE 1,1:PRINT YS$:GET(0,0)-(110,16),JS
50 SCREEN 1,0:COLOR 0,2:CLS:KEY OFF
60 LINE(180,70)-(272,120),0,BF “画显示器”
70 LINE(180,70)-(272,120),1,B
80 LINE(189,75)-(240,115),1,B
90 LINE(252,108)-(262,110),3,BF
100 CIRCLE(257,83),2,,,1:CIRCLE(257,95),2,,,1
110 I=10:WHILE I<>0:LINE(272+I,70+I)-(272+I,120-I*1.4),1
115 I=I-1:WEND
120 LINE(160,125)-(292,148),0,BF “画主机”
130 LINE(160,125)-(292,148),1,B
140 LINE(210,128)-(280,145),1,B
150 LINE(245,128)-(245,145),1
160 LINE(210,136.5)-(245,136.5),1
170 LINE(250,140)-(254,141),2,BF
180 LINE(214,131)-(215,132),2,BF:LINE(214,140)-(215,141),2,BF
190 FOR I=1 TO 30 STEP 2.5
195 LINE(170+I,134)-(170+I,141),3:NEXT I
200 I=10:WHILE I<>0:LINE(292+I,125-I)-(292+I,148-I),1
205 I=I-1:WEND
210 LINE(278,115)-(300,115),1:LINE(160,125)-(180,114),1
215 PAINT(180,123),0,1
220 LINE(169,154)-(283,154),1: “画键盘”
230 LINE(169,154)-(145,180),1:
240 LINE(145,180)-(263,180),1:
250 LINE(283,154)-(263,180),1:PAINT(170,170),0,1
260 LINE(145,180)-(263,183),0,BF:LINE(145,180)-(263,183),1,B
270 I=0:WHILE I<=2:LINE(283+I,154+I)-(263+I,180+I),1
275 I=I+1:WEND
280 FOR N=0 TO 9 STEP 3:FOR I=1 TO 84 STEP 3
285 CIRCLE(175-N+I+2,163+N),1.5,,,1:NEXT I:NEXT N
290 PUT(80,88),JS
300 PRINT CHR$(14)“[@30:24*2*0 产供销和仓库管理系统]”
310 PRINT CHR$(14)“[@20:100*1 江拖发动机厂]”
320 LOCATE 12,9:PRINT “微机室”
330 CIRCLE(32,152),14,2:PAINT(32,152),2,2
340 GET(18,138)-(52,172),A
350 LINE(10,130)-(60,180),0,BF
360 “园动画部分”
370 FOR X=20 TO 280 STEP 4
380 B(X)=159-CINT(ABS(SIN(X*7.853981E-02)*X)\2)
390 NEXT
400 L=7:Y2=150
410 FOR D=0 TO 1:S=20+D*260:E=280-D*260
420 I=1:FOR X=S TO E STEP (4-8*D)*2
430 IF X=S THEN LINE(10,130)-(60,180),0,BF
440 Y=B(X):N=(170-Y2)\5:IF F THEN PUT(X2,Y2),A
450 PUT(X,Y),A:X2=X:Y2=Y:F=-1:NEXT
460 L=(L+7)MOD 64:NEXT
470 SS$=INKEY$“按任一键退出”
480 IF SS$="" THEN GOTO 400 ELSE END
```

四、利用随机函数的动画技术

在一些特殊的 BASIC 语言动画技术中，也可以即不使用“存取式”，也不使用“轨迹式”，而利用随机函数 RND 取一个 0~1 之间的随机数，经过某种运算后，再利用 BASIC 语言的作图语句产生各种大小不同的图形，也能产生很强的移动感。在这种方式下是否需要擦去图形的这一过程，就要根据实际情况决定，有时省略这一过程可能效果更好。程序实例四就是利用随机函数的动画技术的

简单应用, 其中第 30~60 行是画边框显示题头, 70~90 行是产生随机函数并运算后生成矩形的左上角和右下角坐标, 第 100~170 行是随着“嘟”声产生各种大小不同、颜色各异的矩形 (表示各种报表), 在这个程序中就没有采用擦去图行的做法, 而让这些矩形相互复盖, 就好像各种报表争先恐后地出现, 给人以极强的动画感。

```
10 REM IBMPC-BASIC 语言动画技术应用实例之四
20 SCREEN 1,0:COLOR 1,1:CLS:KEY OFF:I=1:DIM M(88)
30 LINE (50,16)-(270,43),1,BF
40 LINE (54,18)-(266,41),3,BF
50 LINE (62,19)-(257,40),2,B
60 LOCATE 3,11:PRINT "经济核算报表管理系统"
70 IX1=RND*250+35:IX2=RND*250+35
80 IX2=(IX1-IX2)/3+IX2
90 IY1=RND*110+55:IY2=170
100 LINE (IX1,IY1)-(IX2,IY2),RND*2+1,BF
110 LINE (IX1,IY1)-(IX2,IY2),0,B
130 LOCATE 13,11:PRINT "江拖发动机厂计算机室"
140 LINE (75,141)-(245,158),1,B
150 IA=ABS((IX1-IX2)*(IY1-IY2))
160 IS=(36400!-IA)/360+37:I=I+1
170 IF I/5=INT(I/5) THEN SOUND IS*10,3
180 SS$=INKEY$
190 IF SS$="" THEN GOTO 70
```

五、应用软件封面的简单制作

利用 BASIC 语言的动画技术设计一个应用软件的封面, 能给计算机的屏幕增加美感, 悦耳的乐曲令人精神振奋, 尤其能提高操作人员的使用兴趣, 根据以上介绍的动画技术, 任意选择其中的动画技术加以组合, 很容易设计出一个图文并茂的画面。以下提供一种简单制作方法:

1、首先利用 BASIC 语言丰富的绘图和音乐语句以及动画技术, 绘制一幅应用软件封面 (程序实例五 FM.BAS), 该程序的特点和有关说明如下: 第 30 行为歌曲《血染的风采》过门, 采用背景演奏方式, 使演奏和绘图同时进行, 实际曲谱在 380~460 行; 第 40 行将高分辨率下的字符串 YS2\$ 存入数组 JS, 在 270 行中分辨率下显示; 第 60~90 行绘制底图, 可调整步长改变图形和速度; 第 120~210 行绘制计算机图案; 第 280~350 行在计算机图案的小小屏幕内, 随着乐曲的节奏不断涌现各种各样、颜色各异的“财务报表”。

• 该程序在运行时可随意按任一键中断退出, 否则反复演奏乐曲和涌现各种报表。

2、使用汉字编译 BASIC 编译和连接成 FM.EXE 文件。

C> BASCOM / O FM; (使用 O 参数, 以便 EXE 文件可脱离 BASRUN)

C> LINK FM; (连接成 FM.EXE 文件)

3、编制批处理文件。

C> TYPE AUTOEXEC.BAT

ECHO OFF

FM

DBASE <文件名>

值得注意的是, 由于编译后的程序和解释 BASIC 下的程序执行时速度有一定的差异, 各种计算机的时钟频率不同, 故需根据实际情况调整程序, 以使得封面上图形变幻和乐曲的协调一致。

```
10 REM IBMPC-BASIC 语言动画技术应用实例之五
20 DIM JS(100):YS$="江拖发动机厂":YS2$="1991.1"
30 PLAY "MB T70 o4g8g16a16a8e16d16c4.o3a8o4g8g16a16a8f16e16d2o4d4"
40 SCREEN 2:LOCATE 1,1:PRINT YS2$:GET(0,0)-(110,16),JS
50 SCREEN 1,0:COLOR 0,2:CLS:KEY OFF
60 FOR X=0 TO 319 STEP 5:LINE(160,100)-(X,0),2:NEXT
70 FOR Y=0 TO 199 STEP 5:LINE(160,100)-(319,Y),2:NEXT
80 FOR X=319 TO 0 STEP -5:LINE(160,100)-(X,199),2:NEXT
90 FOR Y=199 TO 0 STEP -5:LINE(160,100)-(0,Y),2:NEXT
100 FOR I=1 TO 50:CIRCLE(150,90),60-I,0:NEXT I
110 PAINT (151,90),0,0
120 LINE(114,59)-(184,101),1,B
130 LINE(113.5,58.5)-(184.5,101.5),1,B
140 LINE(121,63)-(161,98),1,B
150 CIRCLE(172,70),2,1:CIRCLE(172,80),2,1
160 LINE(100,105)-(200,125),1,B
170 LINE(99.5,104.5)-(200.5,125.5),1,B
180 LINE(136,108)-(192,122),1,B
190 LINE(136,115)-(164,115),1
200 LINE(164,108)-(164,122),1
210 FOR I=1 TO 20 STEP 2:LINE(104+I,114)-(104+I,120),3:NEXT I
220 LOCATE 3,6
230 PRINT CHR$(14)"[ M_40:10%*1会计核算管理系统]"
240 LOCATE 14,9:PRINT "江":LOCATE 14,13:PRINT "拖"
250 LOCATE 14,17:PRINT "发":LOCATE 14,21:PRINT "动"
260 LOCATE 14,25:PRINT "机":LOCATE 14,29:PRINT "厂"
270 PUT(125,175),JS:GOTO 350
280 IX1=RND*250+35:IX2=RND*250+35
290 IX2=(IX1-IX2)/3+IX2
300 IY1=RND*110+55:IY2=170
310 LINE (IX1/7.5+120,IY1/6+65)-(IX2/7.5+120,IY2/6+65),RND*2+1,BF
320 LINE (IX1/7.5+120,IY1/6+65)-(IX2/7.5+120,IY2/6+65),0,B
330 IA=ABS((IX1-IX2)*(IY1-IY2))
340 IS=(36400!-IA)/360+37:I=I+1
350 READ DD$:IF DD$="G." THEN RESTORE 40
360 PLAY DD$:SS$=INKEY$
370 IF SS$="" THEN GOTO 280 ELSE SYSTEM
380 DATA d8,,e16,o3,b8,a16,g16,e8,g8,a8,,o4,
385 DATA e16,o3,b16,a16,g16,a16
390 DATA o4,c2,o4,e8,e16,e16,e8,o3,b16,a16,
395 DATA g4,,e8,a8,a16,o4,c16,o3,b16
400 DATA a16,g8,a2,o3,a4,o4,c8,o3,a8,o4,a8,
405 DATA e16,d16,c8,,o3,a16,o4,d8,d16
410 DATA e16,f8,e16,d16,d2,o4,e8,e16,e16,e8,
415 DATA o3,b16,a16,g4,,e8,a8,a16,o4
420 DATA e16,o3,b16,a16,g8,a2,o3,a4,o4,c8,o3,
425 DATA a8,o4,a8,e16,d16,c8,,o3,a16
430 DATA o4,d8,d16,f16,e8,,d16,c2,o4,g8,g16,
435 DATA a16,a8,e16,d16,c4,,o3,a8,o4
440 DATA g8,g16,a16,a8,f16,e16,d2,o4,d8,,c16,
445 DATA d8,d16,e16,g8,g16,a16,g8,e8
450 DATA d8,,e16,o3,b16,a16,g16,a16,o4,c8,c4,,
455 DATA o4,d4,d8,d8,a1,g1,G.
```

(程序文件名: ZC6.PRG)

适用于大型应用软件的软件保密硬件卡的研制

华南师大 郑潮庆 郑云婷

现在,我国对软件产权保护的立法工作已提出来了,但它只能起一种威慑的作用。破译加密应用软件的活动,由于它有较高的经济利益,实际上是不可能完全停止的,因此软件的研制者与生产者对自己研制的应用软件,特别是大型的应用软件的保护,必须完全依靠技术手段去实现,才能确保自己研制的应用软件的版权和经济利益。

软件保护系统通常由三部分组成:1.物理保护;2.信息加密;3.反跟踪。然而,不管是物理保护还是信息加密,都有它们的弱点,而且加密手段复杂,耗费人力和时间较大,条件不容易具备,也不容易掌握。为此,我们研制出一种与应用软件相结合的软件保密硬件卡,它的电路简单,但保密性能优越,特别适用于一些新研制成功,进行小规模生产的大型应用软件的保密工作。

一、设计思想

1.将应用软件的一部分重要数据与程序存贮在只读存贮器中,因此从软盘复制出来的应用软件内容是不完整和不齐全的,使复制的软件磁盘不能使用。由于存入只读存贮器的内容是无规律的,不可能用任何跟踪手段解密。

2.软件保密硬件应能够在主机内存贮空间的只读存贮器用户区内自由选择使用的存贮空间。

3.封闭整个保密硬件卡,并给予编号。

二、工作原理

为了将应用软件的一部分重要数据与程序存贮在只读存贮器中,可采用一个或多个只读存贮器。例如,采用一个 $8K \times 8$ 位的 2764 芯片作为只读存贮器,其连接电路如图1所示。

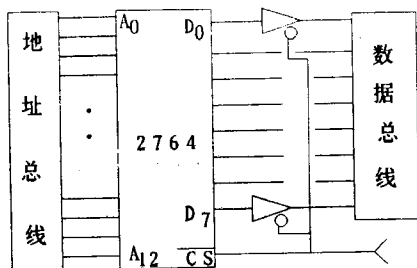


图1

2764 的地址线 $A_{12} \sim A_0$ 与主机的地址总线的 $A_{12} \sim A_0$ 相连。数据线 $D_7 \sim D_0$ 通过单向三态缓冲器与主机的数据总线 $D_7 \sim D_0$ 连接。当应用软件通过 CPU 发出相应的指令,使只读存贮器 2764 的片选引脚 $\overline{CS}=0$, ROM 被选中,同时使三态缓冲器导通,对地址总线给出相应的 $A_{12} \sim A_0$ 地址,对应 ROM 中的数据或指令就可以送到数据总线上,从而使应用软件获得完整和齐全的数据或程序

内容,应用软件就以正常地执行下去,否则,软件不能继续执行。

ROM 的片选信号 \overline{CS} 是通过译码电路得到的,可采用线译码或全译码电路形式。例如,对于 2764 采用全译码形式时,最简单的方法可用 8 端输入的与非门作译码器,将 $A_{19} \sim A_{13}$ 地址线按照只读存贮器占用的内存空间直接地或经过一个与非门接到与非门的输入端,并将 CPU 的存贮器读信号 \overline{MEMR} 也接到与非门的输入端,当地址总线输出的地址值对应于只读存贮器占用的内存空间地址时,8 端输入与非门的输入端全部为“1”,输出端则为“0”。此信号接到 ROM 的 \overline{CS} 端及三态缓冲器的控制端,ROM 被选通。这种联接方式虽然比较简单,但 $A_{19} \sim A_{13}$ 的地址值被硬件所固定,不能用程序加以变化,容易被解密。为此,我们设计一个能使译码地址随程序改变的译码电路。

图2所示是一个由多端输入与非门,双端输入异或门及 D 锁存器组成的程序译码电路。D 锁存器的作用是为译码地址锁存由程序给予的数据,异或门只有在两输入端分别是“0”和“1”时,输出为“1”。锁存器的输出端和地址总线的 $A_{19} \sim A_{13}$ 分别接到对应的异或门输入端,因此,当锁存器的数据由程序给定以后,相应的译码地址也

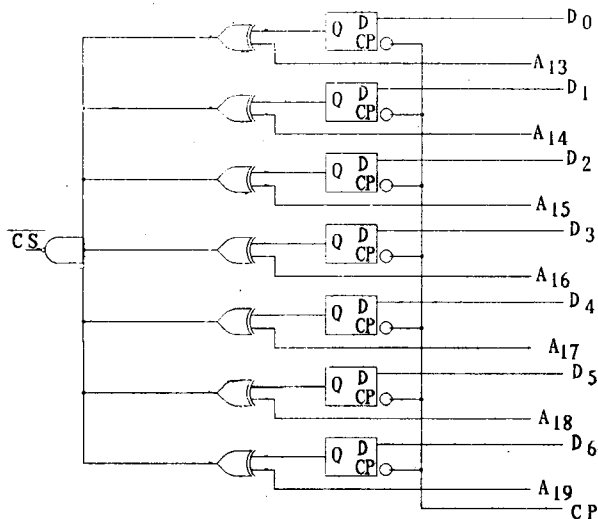


图2

就确定了。因为,当 D 锁存器的输出端为“1”时,对应地址线的值应该为“0”,那么异或门的输出是“1”。反之,D 锁存器的输出端为“0”时,对应的地址线的值应该是“1”。这样就可使全部的异或门输出全为“1”。由图2可见,异或门的全部输出端与非门的输入端相连,

MS-DOS 彻底剖析 (十二)

BIO 常驻模块总体结构

郭嵩山 陈政

一、概述

MS-DOS 的 IBMBIO 模块 (即磁盘 BIO) 是 DOS 内核和 ROM-BIOS 的接口, 它作为层次结构操作系统的次底层, 包括有标准设备驱动程序、初始化程序 SYSINIT I 和 II, IBMBIO 模块经初始化后, 常驻在 0070: 0000H 开始的内存。BIO 常驻模块的内容包括有三部分:

1) SYSINIT I 根据硬件配置对 IBMBIO.COM 中的标准设备驱动程序进行取舍, 留下来的常驻的标准设备驱动程序。例如, 当系统没有配置高容量软盘驱动器或硬盘驱动器时, SYSINIT I 就会根据硬件配置将原标准设备驱动程序的相应处理程序及为其配置的块设备控制块舍去。

2) 由 SYSINIT II 为创建操作系统运行环境而建立的数据结构, 这些数据结构包括有新增加的磁盘参数表 (DPT、磁盘路径表 (DPAT) 以及 FILES、FCBS、BUFFERS 和堆栈运行环境等一系列的表格。

3) 由 SYSINIT II 解释系统盘上的 CONFIG.SYS 文件, 加载由 "DEVICE=" 命令所指定的各个可安装的设备驱动程序。

初始化后常驻在内存的标准设备驱动程序作为 DOS 内核和 ROM-BIOS 的接口, 它为各串行通讯口、打印机、键盘、显示器、软硬磁盘驱动器及时钟等物理设备提供了相应的软件接口, 并将其改造成能供 DOS 内核调用、具有统一调用格式的接口逻辑设备。

因此程序译码器的输入地址就能够不是固定的, 因为多端输入与非门的输入信号是由 D 锁存器及 A19~A13 的地址值通过异或门给定的, A19~A13 的地址值要由 D 锁存器的信号给后才确定。由此可知 D 锁存器的输出数据是由硬件与程序给定, 所以 A19~A13 的值也可按程序给定改变。

为了将程序给定的数据写入 D 锁存器, 需把相应的数据送到数据总线上, 并使 D 锁存器的选通端 CP 有一个负脉冲。数据才能写入到 D 锁存器中。这个负脉冲可通过一个 "存贮器" 写指令产生。为此, 可把地址总线的 A19~A13 的地址线和存贮器写信号 \overline{MEMW} 通过某些非门或直接联到一个多端输入与非门的各个输入端上。这样就可使用存贮器写指令的形式输出一个数据给 D 锁存器。D 锁存器由程序给定后, A19~A13 的地址值也相应确定, 因而可用上述给定地址及 A12~A0 地址从适当的只读存贮器的某一内存空间中读出所需要的数据与程序内容。

BIO 标准设备驱动程序本身由设备策略例程 (strategy Routine)、设备中断例程 (Interrupt Routine) 和设备标题 (也称设备头 Device Header) 组成。其中设备中断例程由主程序及块设备、控制台、辅助设备、时钟设备和打印设备等驱动程序所组成。

整个 BIO 常驻模块的总体结构如图 1 所示。

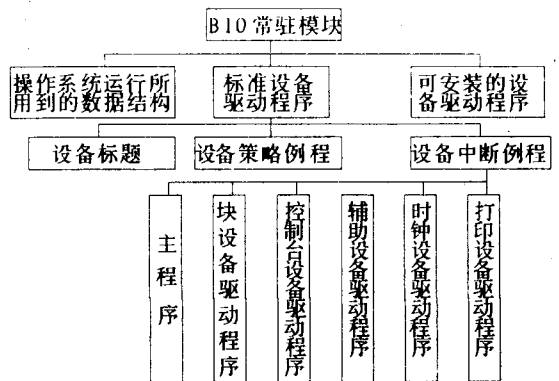


图 1 BIO 常驻模块的总体结构

MS-DOS 作为典型的层次结构的操作系统, 各层模块具有统一的入口和出口, DOS 内核作为 BIO 的上层模块, 是通过填写 I/O 请求标题 (也称 I/O 请求头) 来调

三、工作特点

本加密工具有下列的特点:

1. 方法简单, 容易实现。软件保密硬件电路简单, 元件不必特制。
2. 没有相同的重复产品, 检查复制品容易。硬件卡中的电路可做到每个不同, 没有重复并有编号, 为有重复必有复制品。
3. 软件破译困难, 破译耗费大。由于存贮器读指令在程序中使用众多, 读软件保密卡中的存贮器内容的指令无规则地隐藏于其中, 使解密者不可能找到相应的解密规律与解密程序进行解密, 而要从整个大型软件中找出这些指令是非常麻烦, 花费时间也要相当多, 使破译软件的费用与时间大大超过其所得到的利益, 复制后又容易被查出, 使复制者失去破译复制本软件的兴趣, 从而得保护软件研制者的版权和经济利益。

用 BIO 模块的，而 BIO 模块则通过设备标题（内放设备策略例程和设备中断例程的入口地址）供 DOS 内核调用的。DOS 内核与 BIO 模块间的信息传递就是在这两种数据结构上进行的操作。下面，详细介绍它们的内部结构。

二、设备标题

设备标题是 MS-DOS 所定义的五种标准设备（如图 1 所示）都使用的一种数据结构。每个字符型设备各使用一个设备标题，而所有的块设备都共用一个块设备标题，表 1 列出了设备标题的结构，在表 1 中，设备标题共长 18 个 Byte，它的前 10 个（0AH 个）字节对字符型设备和块设备都是相同的，分别用来记录其设备属性、设备策略例程和设备中断例程的入口地址。设备标题只有最后 8 个字节的意义对字符型和块型设备来说是不同的，字符型设备在这字段中存放设备名，而块设备则用来存块设备参数，对于不同版本的 MS-DOS，该字段中各字节的意义也会有所不同。表 2 以 2.1 和 3.3 两种版本为例，列出了该字段的内容。

MS-DOS (3.3 版) BIO 模块的设备标题在系统装入后，常驻 0070: 16E 至 247H 处，其具体内容请参看表 3。读者如结合表 1 和表 2 的内容，就不难读懂表 3 数据中各个设备标题各字段的值，根据表 3 每个设备标题头一个字段值，我们便可得出 BIO 的标准驱动程序设备链，为便于读者阅读，我们用图 2 示出了该链的结构。

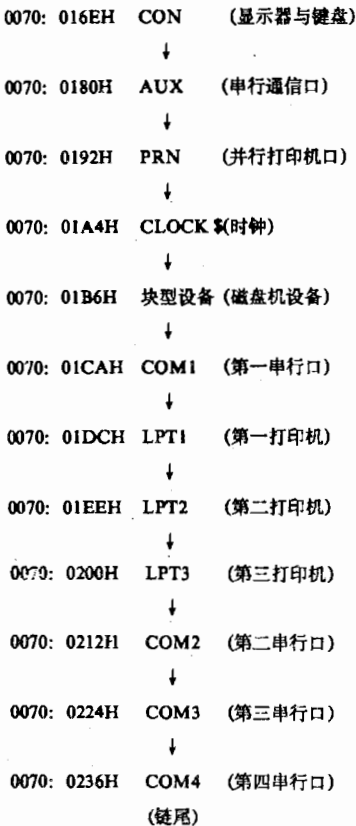


图 2 BIO 标准设备驱动程序设备链

表 1 设备标题的结构

偏移	字 段	类型
+00H	指向下一个设备标题的指针	DWORD
+04H	设备属性	WORD
+06H	设备策略例程入口地址	WORD
+08H	设备中断例程入口地址	WORD
+0AH	字符设备逻辑名 / 块设备参数	QByte

表 2 设备标题中的块设备参数

字节序号	3.3 版	2.1 版
0	块设备部件总数	
1	最后一次读写的驱动器号，其值为 80H，为第 1 台硬盘，其值为 81H，为第 2 台硬盘。其值为 0 及 1，分别为 A、B 软盘，其值为 FFH，表示 I/O 出错	
2	块设备共享驱动器标志，其值为 00H 时，表示当前使用的块设备与其他块设备共享驱动器	最后一次读 / 写的磁道号
3	支持 1.2MB 高容量软盘驱动器标志，其值为 01H 表示支持	上次读 / 写的磁道号
4	其值为 02H 时表示系统中有共享磁盘驱动器的情况	时间计数值低字
5	其值为 01H 时表示支持高级时钟功能，并设置 INT 6CH	
6	其值为 01H 时表示磁盘驱动器规格与块设备控制块记录一致	时间计数值高字
7	其值为 01H 时表示是当前块设备，不必作介质检查	

对于可安装的设备驱动程序，由 SYSINIT II 解释 CONFIG.SYS 文件并将其加载入内存后，由 SYSINIT II 将其各自的设备标题链接起来并链接到标准设备驱动程序设备链尾，使之处于 OS 管理之下。

根据表 1 的结构，我们不难读出表 3 的内容，表 4 列出了表 3 中各设备属性字段的内容。

表 4 各设备属性值

设备名	CON	AUX	PRN	CLOCK \$	块设备	COM1
属性值	8013	8000	A040	8008	0840	8000

设备名	LPT1	LPT2	LPT3	COM2	COM3	COM4
属性值	A040	A040	A040	8000	8000	8000

表3 DOS (3.3 版) 设备标题数据区

0070:016E	80 014...	
0070:0170 70 00 13 80 DC 05 E7 05-43 4F 4E 20 20 20 20		p.....CON	;CON 设备标题
0070:0180 92 01 70 00 00 80 DC 05-ED 05 41 55 58 20 20 20		..p.....AUX	
0070:0190 20 20			;AUX 设备标题
0070:0192 A4 01 70 00 40 A0-DC 05 08 06 50 52 4E 20		..p.@.....PRN	
0070:01A0 20 20 20 20			;PRN 设备标题
0070:01A4 B6 01 70 00-08 80 DC 05 2E 06 43 4C		..p.....CL	
0070:01B0 4F 43 4B 24 20 20		OCK \$;CLOCK \$ 设备标题
0070:01B6 CA 01-70 00 40 08 DC 05 34 06		..p.@...4.	
0070:01C0 03 00 00 01 00 00 00 00		;块设备标题
0070:01C8 -6B 6A			;IBMBIO 有效标志
0070:01CA DC 01 70 00 00 80	kj.p...	
0070:01D0 DC 05 ED 05 43 4F 4D 31-20 20 20 20	COM1	;COM1 设备标题
0070:01DC EE 01 70 00		..p.	
0070:01E0 40 A0 DC 05 0E 06 4C 50-54 31 20 20 20 20		@.....LPT1	;LPT1 设备标题
0070:01EE 00 02		..	
0070:01F0 70 00 40 A0 DC 05 16 06-4C 50 54 32 20 20 20 20		p.@.....LPT2	;LPT2 设备标题
0070:0200 12 02 70 00 40 A0 DC 05-1E 06 4C 50 54 33 20 20		..p.@.....LPT3	
0070:0210 20 20			;LPT3 设备标题
0070:0212 24 02 70 00 00 80-DC 05 F3 05 43 4F 4D 32		.p.....COM2	
0070:0220 20 20 20 20			;COM2 设备标题
0070:0224 36 02 70 00-00 80 DC 05 F9 05 43 4F		6.p.....CO	
0070:0230 4D 33 20 20 20 20		M3	;COM3 设备标题
0070:0236 FE FF-70 00 00 80 DC 05 FF 05		..p.....	
0070:0240 43 4F 4D 34 20 20 20 20		COM4	;COM4 设备标题

设备标题中属性字段共 2 个 Byte, 16bit, 其各个 bit 的意义如下:

- 位 15: 1/0 表示字符型/块型设备;
 - 位 14: 1/0 表示支持/不支持 IOCTL 功能;
 - 位 13: 1/0 表示非 IBM 格式/IBM 格式块设备;
或 1 表示输出直至忙 (字符型设备), 对 2.x 版本, 该值为 0;
 - 位 11: 1/0 表示支持/打开可更换介质, 对 2.x 版, 该值为 0;
 - 位 10~位 5: 其值为 0 (保留);
 - 位 4: 1 表示当前是 CON 设备;
 - 位 3: 1 表示当前是时钟设备;
 - 位 2: 1 表示当前是 NUL 设备;
 - 位 1: 1 表示当前是标准输出设备;
 - 位 0: 1 表示当前是标准输入设备。
- 根据上述的内容, 我们不难读懂表 4 中所列出各设备属性值所代表的意义。

三、I/O 请求标题

表5 I/O 请求标题的结构

字段位移	意义	类型
00H	I/O 请求标题长度	BYTE
01H	请求部件号	BYTE
02H	命令代码	BYTE
03H	设备驱动程序返回状态字	WORD
05H	DOS 保留区	QBYTE
0DH	介质标志字节	BYTE
0EH	传送的内存 (缓冲区) 地址	DWORD
12H	传送的字节数/扇区	WORD
14H	传送的起始扇区号	WORD

在 MS-DOS 中, 一个应用程序或系统程序 (如 DOS 的 COMMAND 模块) 对 I/O 设备提出 I/O 操作请求时, 总是通过 DOS 内核来完成。如前所述, DOS 内核是通过建立 I/O 请求标题来作为 DOS 内核模块调用 BIO 的统一出口。I/O 请求标题的功能是用来存放描述内核要求设备驱动程序完成何种操作的信息。它的前 13 个 Byte 是其静态部分, 对所有的 I/O 设备请求都一样, 后若干个 Byte 由根据所申请的设备及所完成的功能不同而不一样, 表 5 列了一个典型的 I/O 请求标题的结构。

在表 5 中, 设备驱动程序返回状态字段是用来存放 BIO 模块执行 DOS 内核请求的 I/O 操作后, 从 BIO 传送回 DOS 内核的执行结果信息 (忙、完成或出错信息)。

返回状态字段的结构如图 3 所示

15	14-10	9	8	7-0
指错位	保留位	忙位	完成位	指错代码 (在 15 位为 1 时)

图 3 返回状态字的结构

其中, 在第 15 位为 1 时, 低 8 位 (0-7 位) 的意义如下:

位 7 至位 0=00H 写保护
 =01H 未知单元
 =02H 磁盘机未准备
 =03H 非法命令代码
 =04H 数据错 (CRC)
 =05H I/O 请求标题长度错
 =06H 寻道错
 =07H 未知介质
 =08H 找不到扇区
 =09H 打印机无纸
 =0AH 写错
 =0BH 读错
 =0CH 一般错
 =0D-0EH 保留
 =0FH 更换磁盘机错

;*****

; 设备驱动程序使用的数据区 *

;*****

0070:0000 58 1B 2A

;以下块设备的“命令—入口地址”转换表

0070:0003 18 1B 1A C4 09-74 0A 71 06 80 0C 6D 06
 0070:0010 88 06 88 06 6E 0C 65 0C-88 06 88 06 71 06 FD 1B
 0070:0020 06 1C 2B 0C 88 06 88 06-88 06 DD 12 88 06 88 06
 0070:0030 88 06 05 17 29 17 00

;以下控制台设备的“命令—入口地址”转换表

0070:0037- 0A-88 06 88 06 88 06 71 06
 0070:0040 B2 06 E0 06 88 06 2B 07-43 07 43 07 88 06 00

;以下辅助设备的“命令—入口地址”转换表

0070:004F 0A
 0070:0050 88 06 88 06 88 06 71 06-58 07 7F 07 88 06 B9 07
 0070:0060 C2 07 C2 07 A2 07 00

;以下时钟设备的“命令—入口地址”转换表

0070:0067 09-88 06 88 06 88 06 71 06
 0070:0070 69 09 6D 06 88 06 88 06-C5 08 C5 08 00

;以下打印设备的“命令—入口地址”转换表

0070:007D 18 88 06
 0070:0080 88 06 88 06 71 06 7F 06-6D 06 88 06 88 06 E6 07
 0070:0090 E6 07 02 08 88 06 88 06-88 06 88 06 88 06 32 08
 0070:00A0 88 06 88 06 78 08 88 06-88 06 88 06 71 06 71 06

表 6 命令代码—入口地址转换表清单

四、标准设备驱动程序命令代码

在表 5 中, I/O 请求标题中偏移 02H 的字段存放 DOS 内核要求 BIO 模块完成何种操作的命令代码, 在 DOS3.3 (含 3.2 版) 最大的命令代码为 18H 个, 不同的设备驱动程序只支持其中的一些子功能, 只有极少数的命令代码 (如 08H/09H 即写及写且校验) 是既支持字符型设备, 也支持块型设备的。各种标准设备驱动程序所支持的命令代码— (相应的处理程序) 入口地址转换表位于内存 0070: 0003 至 00AFH (3.3 版) 或 0070: 0017 至 00BFH (3.2 版) 中, 其转换表结构如图 4 所示。

最大命令 代码 N	命令 0 入口地址	命令 1 入口地址	命令 n 入口地址	00
--------------	--------------	--------------	-------	--------------	----

1 个 Byte 2 个 Byte 2 个 Byte 2 个 Byte 1 个 Byte

图 4 命令代码—入口地址转换表结构

各个标准设备的命令代码—入口地址转换表在数据区按块设备—控制台设备—辅助设备—时钟设备—打印设备的顺序排列, 3.3 版本的每种设备转换表以 00 作为结束标志, 而 3.2 以下版本则不设置转换表结束标志单元。

表 6 列出了 IBMBIO 模块装入内存后各命令代码—入口地址转换表的内容。

为便于读者理解, 我们将表 6 的内容整理成表 7。

从表 7 中可以看出:

1) 各种标准设备命令代码的最大值: 块设备和列印设备为 18H、控制台设备和辅助设备 0AH, 时钟设备为 09H。

2) 各种设备对于本设备所不支持的命令, 仅设备相应的返回状态完成标志, 不作任何 I/O 操作, 具体的有如下四种不同的处理方式:

a. 表 7 中有 (1) 标记栏, 它调用了入口地址为 688H 的处理程序, 其功能是将 I/O 请求标题的返回状态字的“完成位”置 1。

b. 表 7 中有 (2) 标记栏, 它调用了入口地址为 671H 的处理程序, 其功能是将 I/O 请求标题的状态字设置为“未知命令”的错误代码, 并将“完成位”置 1。

c. 表 7 中有 (3) 标记栏, 它调用了入口地址为 66DH 的命令处理程序, 其功能是在请求标题中将“忙”位及“完成位”置 1。

d. 表 7 中有 (4) 标记栏, 它调用了入口地址为 67FH

命令 代码	功 能	相应处理程序入口				
		块型 设备	控制 台设备	辅助 设备	时钟 设备	打印 机设备
00H	初始化	1A1BE	(1)	(1)	(1)	(1)
01H	介质检验	09C4H	(1)	(1)	(1)	(1)
02H	建立 BPB	0A74H	(1)	(1)	(1)	(1)
03H	I/O 控制读	(2)	(2)	(2)	(2)	(2)
04H	读	0C80H	06B2H	0758H	0969H	(4)
05H	不等待非破坏 性读	(3)	06E0H	077FH	(3)	(3)
06H	输入状态	(1)	(1)	(1)	(1)	(1)
07H	刷新输入 缓冲区	(1)	072BH	07B9H	(1)	(1)
08H	写	0C6EH	0743H	07C2H	08C5H	07E6H
09H	写且校验	0C65H	0743H	07C2H	08C5H	07E6H
0AH	输出状态	(1)	(1)	07A2H	/	0802H
0BH	刷新输出 缓冲区	(1)	/	/	/	(1)
0CH	I/O 控制写	(2)	/	/	/	(1)
0DH	设备打开	1BFDH	/	/	/	(1)
0EH	设备关闭	1C06H	/	/	/	(1)
0FH	可移动介质	0C2BH	/	/	/	(1)
10H	输出直至忙	(1)	/	/	/	0832H
11H	保留	(1)	/	/	/	(1)
12H	保留	(1)	/	/	/	(1)
13H	扩充 IOCTL 功能(块设备) 或置/取循环 检测次数(打印 设备)	12DDH	/	/	/	0878H
14H	保留	(1)	/	/	/	(1)
15H	保留	(1)	/	/	/	(1)
16H	保留	(1)	/	/	/	(1)
17H	得到逻辑驱动 器映像	1705H	/	/	/	(2)
18H	置逻辑驱动器 映像	1729H	/	/	/	(2)

表7 命令代码及处理程序入口地址

命令处理程序,其功能是将 I/O 请求标题中字节数扇区单元清 0,“完成位”置 1。

3) 命令代码 13H、17H 是 3.2 和 3.3 版本所增加的功能。

4) 控制台设备、辅助设备、时钟设备和打印机设备的 08H 及 09H 功能是一样的(即设有“写且校验”的概念)。

五、设备驱动程序执行原理

如前所述,标准设备驱动程序包括设备策略例程和设备中断例程,设备标题是驱动程序所使用的数据结构,标准设备驱动程序运行时先执行设备策略例程,然后再执行设备中断例程,具体执行原理如下:

1. 设备策略例程

设备策略例程常驻在内存 0070: 5DC~5E6H (3.3 版),所有的设备驱动程序都使用该策略例程。策略例程所完成的工作十分简单,它只是将从 DOS 传来的(通过 ES: BX) I/O 请求标题的首址保存在 BIO 的数据区 0070: B8~BBH 的四个 Byte 中。

2. 设备中断例程

设备中断例程常驻在内存 0070: 5E7~69BH 处,不同的设备从不同的入口进入设备中断例程,这些入口分布在 0070: 5E7~637H。设备中断例程执行过程如下:

- 1) 对于包含有多台设备的同类设备,要先传送设备号;
- 2) 找出本类型设备的“命令代码-入口地址”转换表;
- 3) 根据 I/O 请求标题,读出其相应的命令代码及其他参数;
- 4) 计算相应命令入口地址;
- 5) 转执行相应的命令处理程序,完成 DOS 内核所提出的 I/O 操作;
- 6) 填写请求标题中返回状态字段,将返回信息远返回传送回 DOS 内核。

图 5 示出了设备中断例程执行的总流程图。为了便于那些希望深入了解的读者,我们在列表 8 中给出了 3.3 版的整个 BIO 常驻设备驱动程序主程序的详细注释清单。

;设备决策例程入口

;=====

0070:05DC 2E CS:

0070:05DD 89 1EB800 MOV [00B8],BX ;把请求标题首址保存在
0070:B8H~BBH 处

0070:05E1 2E CS:

0070:05E2 8C 06BA00 MOV [00BA],ES

0070:05E6 CB RETF

;CON 设备中断例程入口

;-----

0070:05E7 56 PUSH SI

0070:05E8 BE 3700 MOV SI,0037 ;“命令-入口地址”转换表
始址

0070:05EB EB 4B JMP 0638

;AUX,COM1 设备中断例程入口			0070:061E 56	PUSH SI	
;=====			0070:061F 50	PUSH AX	
0070:05ED 56	PUSH SI		0070:0620 B002	MOV AL,02	;物理设备号
0070:05EE 50	PUSH AX		0070:0622 B403	MOV AH,03	;逻辑设备号
0070:05EF 32C0	XOR AL,AL	;逻辑设备号	0070:0624 BE7D00	MOV SI,007D	;“命令-入口地址”转换表首址
0070:05F1 EB10	JMP 0603				
;COM2 设备中断例程入口			0070:0627 2E	CS:	
;=====			0070:0628 8826D105	MOV [05D1],AH	;保存逻辑设备号
0070:05F3 56	PUSH SI		0070:062C EB0B	JMP 0639	
0070:05F4 50	PUSH AX		;CLOCK 设备中断例程入口		
;=====			;=====		
0070:05F5 B001	MOV AL,01	;逻辑设备号	0070:062E 56	PUSH SI	
0070:05F7 EB0A	JMP 0603		0070:062F BE6700	MOV SI,0067	;“命令-入口地址”转换表首址
;COM3 设备中断例程入口					
;=====			0070:0632 EB04	JMP 0638	
0070:05F9 56	PUSH SI		;块设备中断例程入口		
0070:05FA 50	PUSH AX		;=====		
0070:05FB B002	MOV AL,02	;逻辑设备号	0070:0634 56	PUSH SI	
0070:05FD EB04	JMP 0603		0070:0635 BE0300	MOV SI,0003	;“命令-入口地址”转换表首址
;COM4 设备中断例程入口					
;=====			;标准设备中断例程公共程序		
0070:05FF 56	PUSH SI		;=====		
0070:0600 50	PUSH AX		0070:0638 50	PUSH AX	;保存寄存器
0070:0601 B003	MOV AL,03	;逻辑设备号	0070:0639 51	PUSH CX	
0070:0603 BE4F00	MOV SI,004F	;“命令-入口地址”转换表首址	0070:063A 52	PUSH DX	
0070:0606 EB31	JMP 0639		0070:063B 57	PUSH DI	
;PRN 设备中断例程入口			0070:063C 55	PUSH BP	
;=====			0070:063D 1E	PUSH DS	
0070:0608 56	PUSH SI		0070:063E 06	PUSH ES	
0070:0609 50	PUSH AX		0070:063F 53	PUSH BX	
0070:060A 33C0	XOR AX,AX	;逻辑设备及物理设备号	0070:0640 2E	CS:	
0070:060C EB16	JMP 0624		0070:0641 A26C01	MOV [016C],AL	;保存物理设备号
;LPT1 设备中断例程入口			0070:0644 2E	CS:	
;=====			0070:0645 C51EB800	LDS BX,[00B8]	;取出请求标题
0070:060E 56	PUSH SI		0070:0649 8A4701	MOV AL,[BX+01]	;取请求标题
0070:060F 50	PUSH AX		0070:064C 8A670D	MOV AH,[BX+0D]	中的参数
0070:0610 32C0	XOR AL,AL	;物理设备号	0070:064F 8B4F12	MOV CX,[BX+12]	
0070:0612 B401	MOV AH,01	;逻辑设备号	0070:0652 8B5714	MOV DX,[BX+14]	
0070:0614 EB0E	JMP 0624		0070:0655 97	XCHG AX,DI	
;LPT2 设备中断例程入口			0070:0656 8A4702	MOV AL,[BX+02]	;取命令代码
;=====			0070:0659 2E	CS:	
0070:0616 56	PUSH SI		0070:065A 3A04	CMP AL,[SI]	;若命令代码大于最大命令代码
0070:0617 50	PUSH AX				;转
0070:0618 B001	MOV AL,01	;物理设备号	0070:065C 7713	JA 0671	
0070:061A B402	MOV AH,02	;逻辑设备号	0070:065E 98	CBW	
0070:061C EB06	JMP 0624		0070:065F D1E0	SHL AX,1	
;LPT3 设备中断例程入口			0070:0661 03F0	ADD SI,AX	;求出“命令-入口地址”表中相应项
;=====			0070:0663 97	XCHG AX,DI	

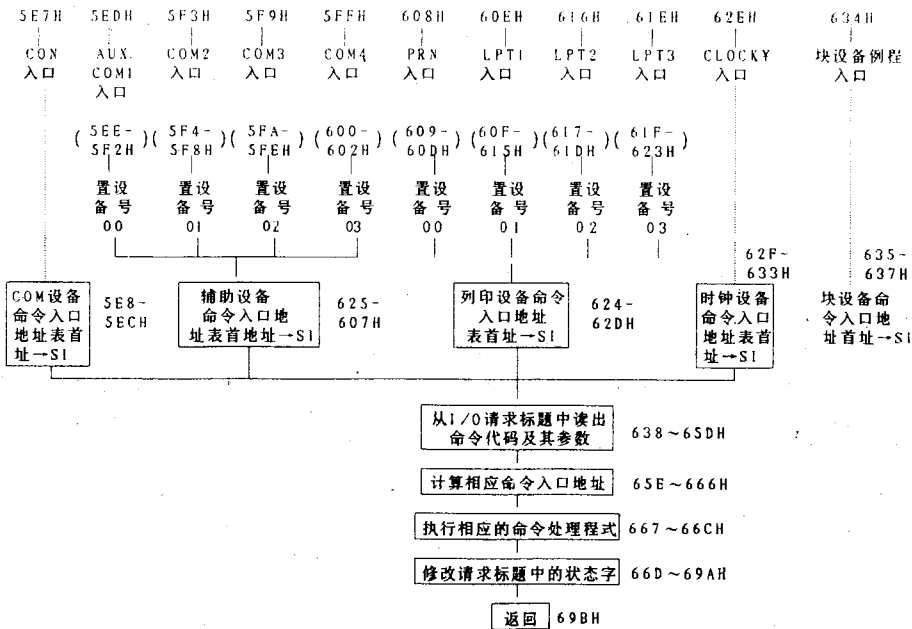


图5 设备中断例程总流程

```

0070:0664 C47F0E  LES DI,[BX+0E] ;取出[BX+0EH-11H]处的
                                指针
0070:0667 0E      PUSH CS
0070:0668 1F      POP DS
0070:0669 FC      CLD
0070:066A FF6401  JMP [SI+01] ;按求得地址跳转
0070:066D B403    MOV AH,03 ;置“忙”位及“完成”位
0070:066F EB19    JMP 068A
0070:0671 B003    MOV AL,03 ;设为“未知命令”错
0070:0673 2E      CS:
0070:0674 C51EB800 LDS BX,[00B8] ;取请求标题首址
0070:0678 294F12  SUB [BX+12],CX ;计数字置为实际I/O数
0070:067B B481    MOV AH,81 ;设置“出错”位
0070:067D EB0B    JMP 068A
0070:067F C51EB800 LDS BX,[00B8] ;取请求标题首址
0070:0683 33C0    XOR AX,AX
0070:0685 894712  MOV [BX+12],AX ;把[BX+12-13H]清零
0070:0688 B401    MOV AH,01 ;设“完成”位
0070:068A 2E      CS:
0070:068B C51EB800 LDS BX,[00B8] ;取请求标题首址
0070:068F 894703  MOV [BX+03],AX ;置“返回状态”字段
0070:0692 5B      POP BX ;恢复寄存器
0070:0693 07      POP ES
0070:0694 1F      POP DS
0070:0695 5D      POP BP
0070:0696 5F      POP DI
0070:0697 5A      POP DX

```

```

0070:0698 59      POP CX
0070:0699 58      POP AX
0070:069A 5E      POP SI
0070:069B CB      RETF ;设备中断例程结束

```

参考文献:

1. 郭嵩山, 陈政: *IBMBIO 启动模块彻底剖析* 香港《微电脑技术》90年第10、11期

欢迎订阅《科技发展与改革》杂志

《科技发展与改革》是国家科委领导下的全国第一家大型综合性科技月刊。主要宣传科技战略和发展趋势, 发布重要文件、政策、法规, 及时报道国内外最新科技成果和动态, 开展技术成果推广、专利及技术转让、需求信息发布等多种服务。是读者跟踪科技发展、拓展知识视野、进行信息交流的重要窗口。本刊时代感强、信息量大、文风清新、形式活泼, 欢迎各界朋友订阅。《科技发展与改革》杂志为16开、64页月刊, 国内外公开发行, 每期定价1.60元, 全年订费19.20元。全国各地邮局均可办理订阅手续, 订阅代号: 82-404。如读者在当地邮局未订到本刊, 也可直接向杂志社邮购; 订费采用邮汇、银行信汇均可。

邮汇地址: 北京2837信箱;

邮政编码: 100044。

联系人: 何京 电话: 842.2852 或 842.2851

银行信汇: 北京工商银行紫竹院分理处

账号: 892181-93

西文 DOS 下的汉字彩色显示

中国市政工程西南设计院 刘建

摘要: 本文给出的 QUICK BASIC 编写的汉字显示程序, 具有使用方便, 占用内存小, 兼容性好, 显示灵活, 适用于 EGA、VGA 多种显示器的特点。

应用软件的开发, 常常用到汉字屏幕显示, 如设计各种常见的屏幕中文菜单等等。中文操作系统当然可以方便地完成这种工作, 但一般中文操作系统占用内存很大, 一些较大的应用程序根本无法使用。还有一些编译程序本身也与中文操作系统不兼容, 通过这类软件开发的应用软件在中文操作系统下运行时, 常常导致死机和其它异常。因此国内不少学者和软件工作者提出了各种办法来解决上述问题, 但这些方法都偏重于依赖机器级的汇编语言, 其使用范围和应用程度受到了较多的限制。目前已有较为理想的方法即是: 利用 16×16 点阵汉字库直接用 QUICK BASIC 编成西文 DOS 下显示汉字的程序。但迄今为止, 这种方法还局限于 CGA 模式, 即 640×200 分辨率和 320×200 模式下, 使屏幕的着色和汉字的多行多列显示不可兼得。笔者在分析 EGA、VGA 显示器的显示特性的基础上, 成功地将上述方法推广到 EGA、VGA 显示模式下, 使西文 DOS 下的汉字显示也充分发挥了 EGA、VGA 的特性, 这样显示的汉字可着色、行列数与中文操作系统比较毫不逊色。还可以在同一个屏幕上显示出大小不同的汉字, 使西文 DOS 下的汉字显示更为灵活和实用, 给软件工作者提供了更为便捷的编程方法。

本程序采用 QUICK BASIC 语言编制, 程序直接读取中文 16×16 点阵字库中的中文字模, 并可在屏幕上任一位置上显示汉字, 显示模式可以按需要选择, 若系统配置为 EGA 则可选模式为 9 以下, 即 640×350 , 而有 VGA 显示器的系统, 则可选显示模式为 13 以内, 即 640×480 以内的全部模式。同时可以根据需要在同一种屏幕下显示大小不同的字形, 也能够使用调色板语句 "PALETTE", 对屏幕背景涂色或改变所显示汉字的色彩。

程序使用的字库为中文操作系统的 16×16 点阵字库, 字库字节数为 "132352", 由于汉字通常占用两个字节空间, 两字节分别由两个 ASCII 码构成, 设两字符对应的 ASCII 码分别为 A1 和 A2, 则该汉字的对应的区位码为: $QW = (A1 - 160) * 100 + A2 - 160$

其中区码即为: $Q = \text{INT}(QW / 100)$;

位码为: $W = QW - Q * 100$

笔者使用的汉字库中汉字对应的 16 个编码信息记录起始位置为: $WZ = (Q - 1) * 94 + W$

由于各种中文操作系统的 16×16 点阵字库不尽相同, 故汉字编码起始位置的计算可能会有差异, 但一般汉字库均以国家标准 GB2312-80 的一、二级字库为基础, 区位码的计算基本一致。只要找出各字库对应的编码起始位置

计算法, 就可以正确地读出各个汉字的编码信息, 得到所需显示的汉字图形映象数据, 再用 PUT 语句将其在屏幕上显示出来。

由于使用 PUT 语句显示图形映象数据时, 整形数组的第 1 个元素应是所要显示图形的列点阵数, 第 2 个元素应是所要显示图形的行点阵数, 故对 16×16 点阵汉字而言, 就需要一个长度为 18 的整形数组来储存单个汉字的图形映象数据, 而其第一个元素中应存放整数 16, 第二个元素中亦应存放整数 16, 其余 16 个元素均取自汉字库内对应的编码信息。这样就可以在 CGA (640×200) 方式下正确的显示出汉字。这样的显示模式下, 屏幕为黑白色, 显然未能发挥出 EGA、VGA 的特性。而上述方式直接引入 EGA、VGA 显示模式明显不行, 这是因为 EGA、VGA 模式 (如 SCREEN7-13) 下, 屏幕的幅数增多, 显示属性又不一致, 故呈现出五彩斑斓的显示效果, 字形也不全。针对这种情况, 笔者采取改变汉字图形映象数组的大小, 改变字符显示的点阵行数, 有规律的重新组织取自汉字库的汉字编码数据在映象数组中的分布, 对无编码数据的行点数据中填充 0 值。从而使全部汉字的图形映象数据都被放入同幅、同属性的行列内, 有效的避免了汉字显示出杂色和残缺, 由于改变了汉字图形映象的点阵行数, 因此使汉字的显示字高也得到了有规律的变化, 如在 SCREEN9 (640×350) 模式下, 就可得到三种大小, 且显示质量均较好的屏幕汉字显示。由于使用了 640×350 的显示方式, 因此加上调色板的使用就可以改变汉字颜色或改变背景的颜色, 使西文 DOS 下汉字显示的效果大为改观, 给制作各种屏幕汉字菜单提供了较好的选择。

由于中文操作系统的运行一般要求占用计算机几百 KB 的内存, 以便装入汉字库, 但对于大多数应用软件而言, 所需显的汉字一般仅为几十至几百个, 采用这种小内存、小字库和西文 DOS 支持下显示汉字是最经济和适用的方法, 特别是对于大多数工业控制计算机而言, 其内存一般都小于 512KB, 因此这种编程方法更为必要和有效。

使用这种方法设计应用软件的过程中, 还可以针对所需显示的有限汉字, 从 16×16 点阵字库中提取汉字的图形映象数据, 建组成应用软件所需的小字库, 或者直接将其置入 DATA 语句中, 嵌入 QUICK BASIC 的源程序以内, 使应用软件的管理和应用更为方便。

值得注意的是, 在 SCREEN 10 模式下, 屏幕有较多属性都呈闪烁状态, 具体说明详见 QUICK BASIC4.0 语言参考手册。

虚盘的动态建立和撤销

华中理工计算机系 88 级 殷军

我们知道 DOS3.0 及以上版本均提供了虚拟磁盘驱动程序 VDISK.SYS。虽然虚拟盘不具备实际盘长期保存信息的优点,系统一旦下电,虚拟盘上的内容将全部丢失,但是,由于在虚拟盘上存取文件的速度远较实际盘的速度快得多,因此,对于计算机用户特别是不拥有硬盘的用户,往往出于提高存取效率或者减少对驱动器及磁盘磨损的考虑,而建立一定容量的虚盘。然而,在实际应用中,如果虚拟盘建得太大,会出现内存不够而导致文件无法运行之危险;反之,若建得太小,又会出现收不到实际效果的现象。针对这个问题,本文所提出的解决方法是建立外部命令,实现虚盘的动态建立和撤销,显然此方法要比通常的解决方法即重新启动系统,生成合适容量的新虚盘来得方便,并也省时得多。以下以 DOS3.3 为例,阐述此外部命令的实现。

一、DOS3.3 的系统环境块 EMB

经过对 DOS3.3 的研究和分析,笔者发现系统是否承认一个驱动器,取决于以下几项内容的存在:

- 1、磁盘参数表 DPB
- 2、磁盘路径表
- 3、逻辑驱动器号
- 4、设备驱动程序

幸运的是, DOS3.3 在其系统内存低区建立了一个系统环境块 EMB,其结构如图 1 所示:

设备驱动程序链头地址	+25H
最末驱动器号	+22H
逻辑驱动器号	+21H
DOS 不能自动关闭的 FCB 数目	+20H
FCBS 运行环境链头地址	+1EH
磁盘路径表首址	+1AH
BUFFERS 运行环境链头地址	+16H
每个缓冲区写字节数	+12H
现行输入设备程序地址	+10H
现行输入时钟设备程序地址	+00H
FILES 运行环境链地址	+08H
DPB 链链头地址	+04H
	+00H

图 1 环境管理块 EMB 结构

因此,我们只要将 EMB 中与虚盘有关的相应部分做修改,如 DPB 入链、建立磁盘路径等,系统便会对此虚盘实施管理。

程序一:将虚盘的 DPB 链入 DPB 链

入口: DS: SI=DPB 链首址

ES: DI=虚盘 DPB 的入口址

出口: 无

```
DPB1: CMP WORD PTR (SI+18H), 0FFFFH
      IE DPB2
      LDS SI, (SI+18H)
      JMP DPB1
```

DPB2: mov (SI+18H), DI

mov (SI+1AH), ES

程序二:建立虚盘的磁盘路径表

入口: ES: SI=磁盘路径表首址

AL=当前最后的逻辑驱动器号 (0=A, 1=B) 等。

出口: 无

MOV CL, 51H

MUL CL

ADD SI, AX

ADD SI, 44H

MOV BYTE PTR ES: (DI), 40H

程序三:将虚盘的设备驱动程序链入设备驱动程序链。

入口: DS: SI=驱动程序链首址

ES: DI=虚盘的设备驱动程序入口址。

出口: 无

MOV AX, (SI)

MOV BX, (SI+2)

MOV (SI), DI

MOV (SI+2), ES

MOV ES: (DI), AX

MOV ES: (DI+2), BX

注: DPB 结构, 磁盘路径表等结构详见 1991 年第 3 期《电脑》杂志。

二、虚盘的安置。

如果我们仅仅象 DOS 那样将虚盘设置在内存低端区,显然仍然可能会出现内存不够的现象,同时对我们回收被虚盘所占的内存也带来麻烦。如图 2 所示:

一旦我们将虚盘撤销,从图中可以看到,将会出现空闲内存块 1。这时,如果一个程序其运行空间小于空闲内存块 1 与空闲内存块 2 空间之和而大于其中任何一个,则当它被加载执行时,依然会出现“”的错误而导致运行失败,显而易见。如果我们多次地动态建立和撤销虚盘,类似于空间内存块 1 的碎片将越来越多,最后极有可能导

致系统崩溃的危险。

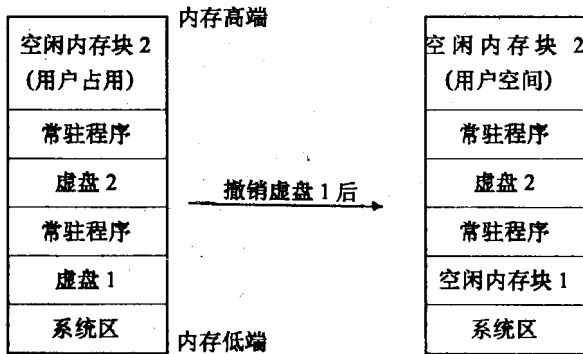


图 2 虚盘建立在低端

最好的解决方法是将虚盘建立在内存高端，如图 3 所示：

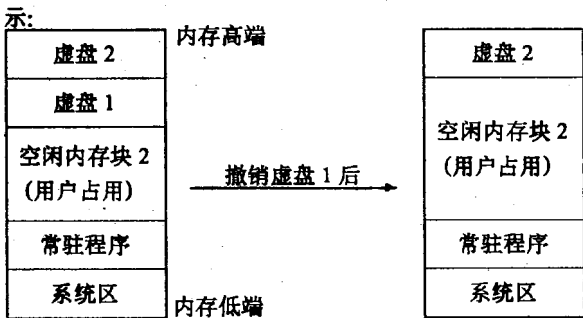


图 3 虚盘建立在内存高端

即每建立一个虚盘，就从最后的内存块中截取高端内存给虚盘，而当撤销一个虚盘时，则将最近建立的虚盘撤销，其容量“还”给最后一块内存块。显然，此种方法将再也不会出现上述的内存碎片问题，同时对常驻在内存低端的常驻程序也不会产生任何影响和破坏。

程序四：获取最后一块内存块地址

入口：ES=程序当前的 PSP 段址

出口：ES=最后一块内存块段址

MOV AX, ES

DEC AX

MOV ES, AX

LASTO: CMP BYTE PTR ES: (0000), 5AH

JZ LAST1

MOV AX, ES: (0001)

ADD AX, ES: (0003)

INC AX

MOV ES, AX

JMP LASTO

LAST1:

程序五：截取高端内存块段址

入口：ES=最后一块内存块段址

AX=截取内存的字节数（1节-16字节）

出口：无

SUB ES: (0003), AX

PUSH AX

MOV AX, ES

INC AX

MOV ES, AX

POP AX

SUB ES: (0002), AX

对于一个虚盘，它的建立所需内存字节数可用以下公式推算：

$MEMORY = (DPB \text{ 占用字节数} \times 2 + \text{虚盘驱动程序带驻字节数} \times 2EH) \times 10H + \text{虚盘的实际容量}$ 。

三 重要的两个 DOS 保留功能调用

由于我们要对系统环境块 EMB 进行一系列的修改，因此首先就需要得到它的地址，DOS 提供了这样的一个功能调用 AH=52H。

入口：AH=52H

出口：ES: BX=EMB 地址

另外，设备驱动程序初始化，仅提供 BPB（即 BIOS 参数块），而我们需要的是 DPB，虽然 BPB 与 DPB 之间可以通过我们自己编制的程序来转换，但是 DOS 提供了另外一个保留功能调用 AH=53H，它可帮助我们实现 BPB 与 DPB 之间的一系列转换。

入口：AH=53H

DS: BX=BPB 地址

ES: BP=DPB 地址

出口：BPB 被转换成 DPB。

四、虚盘动态建立和撤销的一般步骤。

至此，我们可以实现虚盘的动态建立和撤销了。建立虚盘的一般步骤如下：

- 1、获取系统环境块及最后内存块的地址。
- 2、按虚盘驻留时所需内存字节数计算虚盘在内存高端的起始地址。
- 3、读虚盘驱动程序到内存高端。
- 4、初始化虚盘驱动程序。
- 5、将 BPB 转化成 DPB。
- 6、修改系统环境块，DPB 等入链，并将逻辑驱动器号置 1。
- 7、截取高端内存，实现虚盘常驻。

与之相对应的撤销虚盘的一般步骤如下：

- 1、获取系统环境块地址及最后内存块的地址。
- 2、从而获取虚盘及虚盘相应的 DPB 地址。
- 3、修改系统环境块，删除虚盘的 DPB 等，并将逻辑驱动器号减 1。
- 4、释放内存，实现与最后内存块合并。

虽然可以实现动态建立虚盘的外部命令了，但是在编制程序的时候，还应考虑以下几个细节：

- 1、即使在内存允许条件下，虚盘也并不是可以无限多地动态建立。

由于磁盘路径表的个数由系统环境块中最后的驱动器号所决定，因此如果建立虚盘的个数超过最末驱动器号与

采用 82C11 芯片并行打印机接口的维修

江苏盐城微机应用研究所 曹迎春 沈 勇

目前,随着微机集成度的提高,体积的缩小,许多微机的并行打印机接口不再用十多块 74 系列芯片组成,而利用一块 82C11 专用芯片加上一块可编程逻辑芯片(如 PAL16L8、82S129)和一只收发器芯片(74LS245)等组成,占用面积小,可靠性好。82C11 芯片集部分译码器、数据缓冲器、数据寄存器、控制寄存器、控制驱动器和状态缓冲器于一体,完成读写数据口、读状态口和读写控制口的全部功能(见图 1)。

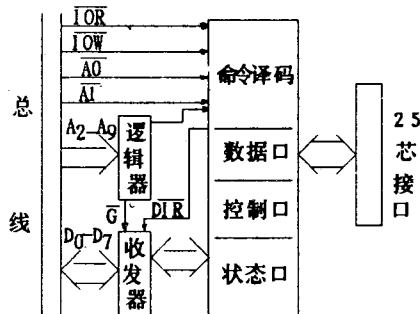


图 1

可编程逻辑芯片的作用是根据总线的地址线 A2~A9,译出口地址 0011011110XX (0010011110XX),总即为 378~37A (278~27A),输出低电平有效信号。这个低电平有效信号(暂称为“译码”)送给芯片 82C11,可编程逻辑驱动器号之差,可能会导致系统崩溃。

2、撤销虚盘时,当前缺省驱动器号绝对不能是待撤虚盘。

3、在初始化虚盘驱动程序之前,一定要致 CS: 040A 处的字内容为 0 (CS=驱动程序段址)。

原来此字内容=1000H=64K,它用来判断虚盘常驻后内存剩下空间是否还大于或等于 64K,若小于则将调整虚盘容量直至剩余内存空间至少为 64K。由于动态建立时是建立在内存高端,因此,若不将此字清零将可能在内存高端形成一个 64K 的空闲内存区,它造成了内存空间的浪费。

4、为了安全起见,在每建立一个虚盘时,应至少给用户留 64K 的内存。

此举是为了保证有足够的内存供文件能够正常运行。

五、结束语

本文从 DOS3.3 的系统环境块着手,讨论了动态建立和撤销虚盘应考虑的问题和一般步骤。虽然笔者已经将此思想付诸于实现,并收到了很好的应用效果,但由于水平有限,难免有考虑不周之处,还望读者指正。

逻辑芯片同时也控制收发器的使能端 (\bar{G})。

芯片 82C11 根据“译码”信号,地址线 A0, A1 和 \overline{IOR} 、 \overline{IOW} ,译出对 378 口, 379 口, 37A 口的读写命令,同时产生一个 DIR 信号控制收发器 (LS245) 的数据流向。

当 IOW 有效时,芯片 82C11 接受总线通过收发器送来的数据或控制信号,并送往 25 芯接口;当 \overline{IOR} 有效时,芯中 82C11 把 25 芯接口送来的打印机状态信息或本身保存的数据、控制信息送给收发器,并由收发器送往总线。

由于市场上 82C11 芯片很难购得,所以一旦 82C11 芯片故障,便造成并行打印口无法修复,从而使整个控制板(往往与 RS-232 口、游戏口、软驱控制器做在一起)报废。我们在维修中发现 82C11 芯片的故障,往往并不是整个芯片损坏,使全部功能丧失,而是某一功能失效,甚至是某一功能中的一部分失效。例如:盐城无线电总厂一台 IBM 286 机打印口故障现象为“打印机始终缺纸”,其它功能正常,即能正确地把数据或控制命令发往打印机或返回给总线,在读打印机状态时,能把除“纸尽”以外的状态信号送往收发器,输入纸尽信息正常,即 82C11 的 27 脚输入有高低电平变化,但经它送往收发器时始终为高电平(82C11 的 14 脚输出),即“打印纸缺纸”。又如:盐城供电局长城 286 多功能板的打印故障现象为“打印机始终错”,其它功能正常,即 378 口读写正常,37A 口读写正常,但读 379 口时 D3 位恒为 0 (打印机错),用逻辑笔测得 82C11 的 29 脚恒为低电平,割断该脚与 25 芯接口第 15 芯的连线,15 芯状态为高电平,而 82C11 的 29 脚为低电平,可见 82C11 芯片的 29 脚就不能正确接收打印机错误状态信息。

在购不到 82C11 芯片又要保留原卡板情况下,我们对原有的 82C11 芯片进行“修补”,即保留 82C11 芯片功能正常的部分,补充 82C11 失效的功能,从而修复打印口,保留原有卡板。在此以芯片 82C11 传递缺纸状态功能失效的修复为例。

设计要求:

切断芯片 82C11 第 14 脚与收发器 LS245 第 13 脚 (BD5) 的通路。当可编程逻辑芯片的输出“译码”信号有效,且 A0=1, A1=0 (地址译码为 379),并且 \overline{IOR} 有效时,将 25 芯接口第 12 芯的纸尽状态信息直接送给收发器的 BD5 端,其它打印机状态信息仍由芯片 82C11 送给收发器。两者一起由收发器送往总线。当读写 378 口、37A 口时,“恢复”芯片 82C11 第 14 脚与收发的 LS245 第 13 脚的通路,完成数据或控制命令的双向传递。

设计方法:

1、切断 25 芯接口第 12 芯与芯片 82C11 第 27 脚的连线;切断芯片 82C11 第 14 脚与收发器 LS245 第 13 脚的连线。

2、按照图 2 利用一块 74LS04 一块 74LS20 和一块 74LS125 做成一个小电路板。该板上引入原控制卡上的十根信号线:芯片 82C11 第 14 脚,收发器 LS245 第 13 脚,25 芯接口第 12 芯,地址线 A0 和 A1, \overline{IOR} 和 \overline{IRW} , 原可编程逻辑芯片的输出“译码”信号,以及 +5V 和 0V。

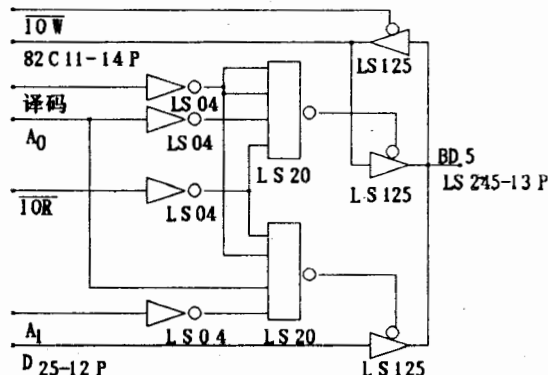


图 2

3、当 \overline{IOW} 有效时,打开 LS125 的第一个缓冲门,把收发器 LS245 第 13 脚的 RD5 信号送给芯片 82C11 的第 14 脚(收发器本身的控制正常)。这时 \overline{IOR} 无效,LS125 的第二三缓冲门隔断(因为 LS125 为三态输出),不影响第一个缓冲门通过的信号,从而完成写数据口、写控制口的工作。

当 \overline{IOW} 无效时,LS125 的第一个缓冲门隔断,不允许把收发器的信号 BD5 送给芯片 82C11。

4、当 \overline{IOR} 有效,可编程逻辑芯片的“译码”输出有效,并且 A0=0 (地址为 378 或 37A) 时,经过 LS20 的第一个四输入正与非门,输出一个低电平,使 LS125 的第二个缓冲门打开,把 82C11 第 14 脚的信号送给收发器 LS245 的第 13 脚,即完成读数据口或控制口的工作,这时由于 A0=0 则 LS20 的第二个正与非门输出高电平,使 LS125 的第三个缓冲门隔断,所以不会把打印机纸尽状态信息送入收发器 LS245。

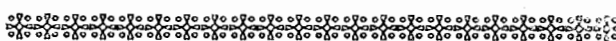
5、当 \overline{IOR} 有效,译码输出芯片输入有效,并且 A0=1, A1=0 (口地址为 379) 时,LS20 的第二个正与非门输出一个低电平,使 LS125 的第三个缓冲门打开,把 25 芯接口第 12 芯的纸尽状态信号送给收发器 LS245,即完成读状态口的工作。这时由于 A0=1,则 LS245 的第一个正与非门输出高电平,使 LS245 的第二个缓冲门隔断,所以不会把芯片 82C11 的错误纸尽信息送给收发器 LS245。

在做这块小电路板时,为了保持信息传递速度一致,

应尽量少的使用芯片,而且应和传输时间短的高速芯片。尤其是写数据口,写控制口的频率较高,速度较快,所以用 \overline{IOW} 信号直接控制 LS125 的第一个缓冲门,不必考虑地址译码,因为收发器本身原有控制正常, \overline{IOW} 有效,写其它 I/O 设备时,对此无影响,这样大大提高了写数据口、写控制口的速度。为了确保数据传递的正确性,必须选择具有三态输出功能的芯片(如 LS125)作为信息通路芯片。

我们用上面方法修复的 IBM 286 的 I/O 卡和长城 286 的多功能卡,已分别使用二年和一年,一直很正常。这种成本低,效果好的维修方法深为用户满意。

目前,专用大规模集成电路芯片日益增多,这种芯片一旦损坏,市场上又难以购得,采用这种保留原芯片的正确功能,而对失效功能进行“修补”的方法,无疑是一种较为可行的方法。借此文推荐给计算机维修同行,以供参考。



改 GWBASIC 使其适用于 CCBIOS 2.13H

山东潍坊质检所计算机室 李志刚

CCBIOS2.13H 自推出以来,深受广大用户的欢迎,尤其是它能在 EGA、VGA 卡的支持下显示 25 行以上的彩色汉字,给使用带来了很多方便。但美中不足的是以前使用较为普遍且为大家最熟悉的 GWBASIC 却不能正常运行。在 2.13H EGA (或 VGA 下同) 25 行汉字状态下,如启动 GWBASIC,你将会发现,仅能在屏幕的上部显示 5×40 个字符,无法使用,在其它 CCDOS 系统 EGA25 行汉字状态下也会出现这个问题。

通过对 GWBASIC.EXE 的分析发现,这是由于 GWBASIC 没有 EGA25 行汉字状态 (相应的屏幕状态字为 AL=10H) 的相应处理程序所致。又对 2.13H 分析得知:用 CE26.COM (或 CV26.COM) 启动后,只要屏幕状态字 AL 不小于 06H,2.13H 都会将屏幕置为 EGA25 行汉字状态。因此我们只要将 GWBASIC 程序启动时,取到的屏幕状态字 10H,改为 06H,利用 GWBASIC 原有对 06H 状态的处理程序,便可使 GWBASIC 在 EGA25 行汉字状态下正确运行。这样修改后的 GWBASIC 在 EGA25 行汉字状态下不仅可以恢复运行,而且可以显示多至 16 种颜色的彩色汉字,这样修改的 GWBASIC.EXE 同样适用于其它 CCDOS 系统的 EGA25 行状态。具体修改操作如下:

```
C>REN GWBASIC.EXE GWBASIC
C>DEBUG GWBASIC
-UIB46 1B53
4465: 1B46 B40F          MOV AH, 0F ; 原有取屏幕状态
                        部分
4465: 1B48 CD10          INT 10
4465: 1B4A 883E5706      MOV (0657), BH
4465: 1B4E A25B06      MOV (065B), AL
4465: 1B51 A25406      MOV (0654), AL
-A1B46
4465: 1B46 CALL 6055      ;修改部分
4465: 1B49 NOP
4465: 1B4A
-A6055
4465: 6055 MOV AH, 0F ;新添部分
4465: 6057 INT 10
4465: 6059 TEST AL, EF
4465: 605B JNZ 605F
4465: 605D MOV AL, 06
4465: 605F RET
4465: 6060
-W
Writing DE00 bytes
-Q ■
```

PC 机上试题库图形文件的压缩、还原和显示

西安交通大学 李忠勇

摘要: 本文通过对屏幕图形文件内部结构的分析,详细叙述了图形文件的机理和显示过程,提出了一种新颖、高效的屏幕图形压缩法,被压缩的图形文件仅为点阵图形储存方式的十分之一左右。非常适宜于计算机图形的远距离传输和图形文件的储存。这种压缩后的图形文件无需恢复文件本身便可高速地直接在计算机屏幕上显示图形和在打印机上拷贝图形。

关键词 试题库, 屏幕图形, 图形文件, 文件的压缩, 位图, 象素。

一、试题库与图形文件

在我国,试题库是伴随着教学改革的深入和 PC 机的广泛应用而涌现的一项新的软件包。它的诞生极大地方便了广大的教育工作者,提高了教学质量和教学水平,这项应用现代化教学手段的成果必将很快在高等院校的各个学科得到最广泛的应用。

一个好的试题库集成环境软件包应包括试题文字的建立,试题图形的建立,试题文字和相应图形的管理,查询,修改和自动组合,还应包括试卷的诊断,对考试结果的分析及部分试题、图形的自动形成。由于各专业,学科之间的极大差异,许多专业都需要拥有本学科特色的试题库集成环境软件包,然而对于大多数非计算机、控制类的

教学人员来说,试题库图形文件的建立和管理是一项非常棘手的难题,目前国内虽然流行着一些非常出色的绘图软件,如 AUTOCAD, WIMDOWS 等绘图软件,然而由于它们的封闭性和绘图软件本身的庞大,虽然它们有着卓越的特性,也未能在试题库管理软件包中得到实际地应用。

以交通大学为例,目前各专业虽在纷纷动员力量研制适合于本学科试题库管理软件包,然而在试题库管理软件包集成环境中具有绘图和图形管理功能的却寥寥无几,即使有此功能,也多是用 BASIC 和 C 语言绘出,然后以点阵图形储存方式储存,经过一段时间的应用,这种方式的缺陷就会明显的暴露,我们知道,考题的保密性要求很高,为了防止考题泻漏,试题及图形往往储存在软盘上加

以保存, 用位图方式储存图形每帧画面至少 16K, 若以每套考题加答案 15 幅画面记, 则一张 1.2M 的高密盘仅能储存五套考题的图形, 这对于试题的查询和管理是极为不利的。

图形文件的压缩不仅对试题库的管理, 对整个计算机软件界都是一项重大的课题。本文旨在针对试题库中图形文件的特点提出一种新颖、高效的压缩方法, 并介绍如何截取其它绘图软件的图形以被您编制的高级语言加以利用。

二、图形文件在 PC 机上的结构

我们知道, IBM-PC 机屏幕上显示的图形总能在内存中找到唯一对应的屏幕缓冲区, 如单色显示器的屏幕缓冲区首址为 B000: 0000, CGA 显示器的屏幕缓冲区首址在 B800: 0000, 而 EGA/VGA 显示器的屏幕缓冲区首址为 A000: 0000, 对 CGA 显示卡, 图形屏幕被划分为 200 行, 每行对应屏幕映射区中的 80 个字节, 高分辨率时, 具有黑色和白色, 每屏为 640×200 个像素点, 每个像素点对应于映射区中一个字节 1 位, 其中内容为 1 表示白色, 内容为 0 表示黑色, 每个字节存放 8 个像素点特性, 中分辨率时, 具有 4 种颜色, 每屏为 320×200 个像素点, 每个像素点对应于映射区中一个字节的 2 位, 即每个字节存放 4 个像素点特性, 屏幕缓冲区实际上被划分为两部分, 一半对应于偶数扫描行, 一半对应于奇数扫描行, 即总数为 100 行的偶数行存放在起始地址为 B800: 0000 的内存中, 而总数为 100 行的奇数行存放在起始地址为 BA00: 0000 的内存中, 采取这种排列是为了便于硬件线路向屏幕上绘图。

最简单常用的图形文件就是把屏幕缓冲区的内容以文件形式存入磁盘, 以 CGA 显示屏为例, 屏幕缓冲区的大小为 $640 \times 200 / 8 = 16000$ (bytes) 因此只要我们保存起始地址为 B800: 0000 的 16K 字节屏幕缓冲区的内容至硬盘或软盘中即可。

无论何种计算机语言要显示具有 16K 大小的图形文件的内容到屏幕上只需要把图形文件中的数据重新写入内存中起始地址为 B800: 0000 的屏幕缓冲区即可, 这种屏幕图形的保存和显示方式速度很快。如前所述, 这种图形文件所占的容量太大, 而且复原显示时首先显示偶数行的图形, 然后再补上奇数行的图形, 虽然无伤大雅, 但看上去总有不舒服感。特别对目前流行的 TVGA 强分辨率显示适配器而言, 每帧屏幕图形的保存需占用多达 768K 字节, 这种图形文件的容量常常是我们所不能容忍的。

三、图形文件的绘制和截取

目前国内较流行的优秀的绘图软件有 AUTOCAD 和 WINDOWS 中的绘图工具。然而这些绘图软件保存的图形都是自成一体封闭式的。为用户自编的高级语言难以调用, 更无法进行管理和修改, 对试题库集成环境的编制人员来说这些绘图软件还太过于庞大, 许多功能是极少使用的。因此只宜作为外部绘图软件通过高级语言中的外壳

指令加以调用。

在懂得了 PC 机上屏幕图形的结构之后, 我们可以通过保存屏幕缓冲区的内容得到普通高级语言都能调用的图形文件 (1)。这种保存屏幕缓冲区内容的程序必须首先运行以驻留内存, 然后在适当的时候接受用户指令, 完成形成图形文件的功能, 这样形成的图形文件就可被几乎所有高级语言和汇编语言加以调用。

根据上述原理, 我们编制了一段名为 "SAVESCR" 的图形截取程序, 程序修改了 DOS 系统中屏幕动态硬拷贝的中断向量 5H 的入口地址, 使用户按下 "Shift+PrtSc" 键时, 不是转入运行屏幕硬拷贝服务程序, 而是转入运行保存屏幕缓冲区内容的服务程序。这样在运行外部作图软件作图完毕返回我们编制的试题库管理集成环境后, 我们可方便的让试题文字和图形同时在屏幕上显示, 并且对录下的图形象对待试题库管理集成软件中内部作图软件作出的图形一样, 可进行图形的修改和图形文件的压缩处理。

在我们编制的名为 "SAVESCR" 的屏幕图形截取程序中, 图形文件的文件名是通过命令行指令把其放入程序段前缀中的。这就为图形和试题的对应及管理提供了极大的方便。

好的试题库集成环境管理软件都应带有具有一定功能的内部作图模块。对不同的学科这种作图模块将呈现不同的特点。如控制系统类的试题图主要是原理图, 重点在电气元件集成块, 而不注重线条的粗细。但某些特殊情况下, 也需要画些很漂亮, 带立体感的图形, 因此, 图形的截取对试题库管理软件来说也是非常重要的。

顺便指出, 用上述方法不仅能在使用 autoCAD 和 Windows 绘图软件绘图时截取所绘制的图形, 我们同样可以用它来截取游戏程序或其它应用程序中精美漂亮且逼真的各种图形对这些图形进行巧妙的加工利用, 就可以为我所用。

四、屏幕图形的压缩

在试题库管理软件中, 由于绘制图形是为了与试题配合打印后给学生考试用, 而不仅是为了在计算机屏幕上绘制精美漂亮的图形, 因此均采用黑白模式作图。对 CGA 显示器来说, 黑白模式作图的分辨率为 640×200 , 比用彩色模式的 320×200 分辨率高一倍。

对 EGA/VGA 显示适配器, 用黑白模式作图时可用图形位储存方式形成图形文件, 则图形文件所占的字节要大大节省。对现在流行的 1024×756 的彩色图形适配器而言, 储存一幅精美的彩色图形通常需占用 100~500k 的容量。

这里提出的屏幕图形压缩方法将重点针对黑白模式下的图形文件。

通常我们在计算机屏幕上作出的图并不充满整个屏幕, 特别是屏幕的顶端和屏幕的底部往往都是空的。这样当我们把整个屏幕的全部像素用位方式全部保留下来之后, 我们会发觉大量的位都充满了 "0", 另外在某些连续的

位上又充满了“1”。因此,我们利用这点进行第一次压缩处理。

设用位方式保存的图形文件的第一位值为0,(此条件通常均能被满足)从第一位开始往后连续的数值为0的位数直至遇到值为1的位为止,其值计为 $N_0(1)$, $N_0(1)$ 满足:

$$0 < N_0(1) < 2^{30} - 1 \quad (1)$$

然后从第一位值为1的位开始往后连续的数值为1的位数直至遇到值为0的位为止其值计为 $N_1(1)$, $N_1(1)$ 满足: $0 < N_1(1) < 65526$ 。

以后反复数0和1的位,记为 $N_0(1)$ 和 $N_1(1)$ 。

$$0 < N_0(1) < 65526 \quad i = 2, 3, 4, \dots \quad (2)$$

$$0 < N_1(1) < 65526 \quad i = 1, 2, 3, \dots \quad (3)$$

当数到图形文件的尾部时若为 $N_0(1)$,则 $N_0(1)$ 可能大于65525,因此我们应定义 $N_0(1)$ 为长整数

$$0 < N_0(1) < 2^{30} - 1$$

以上各不等式在CGA显示适配器上均可满足。若在EGA/VGA显示适配器上作图,可定义 $N_0(1)$ 、 $N_1(1)$ 为长整数。

如果我们把以上各数组的值以字符串的形式存入数据文件中的话,则数据文件将具有如图(1)或图(2)所示的排列:

$$0 \ N_0(1) \ N_1(1) \ N_0(2) \ N_1(2) \ N_0(3) \ N_1(3) \dots \quad (3) \dots$$

图(1)

$$1 \ N_1(1) \ N_0(1) \ N_1(2) \ N_0(2) \ N_1(3) \ N_0(3) \dots \quad (3) \dots$$

图(2)

数据文件中开头的0表示紧跟在后面的数据值是象素值为0的个数,开头的1表示紧跟在后面的数据值是象素值为1的个数。

注意,数据文件中的全部数据都是无符号的正实整数,因为按约定,象素值为0的个数与象素值为1的个数是交替排列的。

从图(1)和图(2)得到的数据文件也就是经压缩后的图形文件,经过第一次压缩处理后一般试题库中图形文件的大小约为5k左右,比原需的16k字节已大大减小。

我们知道,全部由数据构成的文件本身是具有很大可压缩性的,图(1)和图(2)表示的文件就是全部由数字构成的,而且全部是无符号的正实整数。

目前计算机内的每一字节长度均为8位,每位可表示数据0~256,如图(4)表示的数据文件中数值123+空格需占用4个字节,事实上我们可用一个字节来不示,即把已上数值123变成计算机内码储存。

若经第一次压缩后的图形文件满足方程(1)、(2)、(3)的条件,则图(1)和图(2)转换成计算机内码储存

的格式为:

字	字	字	字	字	字	字	字	字	字	字
节	节	节	节	节	节	节	节	节	节	节	
1	2	3	4	5	6	7	8	9	10		

$$0 \text{ 或 } 1 + N_0(1) \quad N_1(1) \quad N_0(2) \quad N_1(2) \dots$$

图(3)

前4个字节的最高位用来表示图(1)和图(2)中的0或1,之后约定为2字节不示1位数。

经压缩后的图形文件的最后一位数据通常是 $N_0(\max)$,即通常表示内容为0的象素的个数,若如此,其数据将可能出现:

$$N_0(\max) > 65535$$

但在绘制图形时,最后一位数据若为 $N_0(\max)$,计算机在绘图方面将不做任何工作,因此, $N_0(\max)$ 可忽略,使其不出现在图(3)中。

把图(1)和图(2)的图形文件变成图(3)所示的用计算机内码表示的图形文件后,则一个16K的位图形文件可压缩成4K字节左右。

仔细分析图(1)和图(2)的图形文件,可发现一半以上的数据并未被压缩。在图(2)中我们约定为对每一个数据用二字节表示,而图(1)和图(2)中一半以上的数据都小于9,大量的数据值为1,这些数据在图(1)、图(2)和图(4)中占有相等的字节。

进一步,我们把方程(2)和(3)表示的正整数分为两类 d_1 和 d_2

$$1 < d_1 < 127 \quad (4)$$

$$128 < d_2 < 65535 \quad (5)$$

满足条件(4)的数据用一个字节表示,而满足条件(5)的数据用二个字节表示。如图(4)所示。

字	字	字	字	字	字	字	字	字	字	字	字
节	节	节	节	节	节	节	节	节	节	节	节	
1	2	3	4	5	6	7	8	9	10	11		

$$\text{长度 } 0 \text{ 或 } 1 + N_0(1) \quad d_1 d_2 d_1 d_1 \dots$$

图(4)

图中的数据仍按 $N_0(1)$ 、 $N_1(1)$ 或 $N_1(1)$ 、 $N_0(1)$ 的顺序交替排列,对数据 d_2 ,其0-127的数据是不可能出现的,因此我们利用这份空档以测试图(4)中的各个数值是属于 d_1 还是属于 d_2 。

图(4)中我们以前两个字节表示图形文件所占的字节数。

经过以上第二次压缩处理后,图形文件的容量将大大减少。无可置疑,方程(1)对所有图形适配器都是满足的。方程(2)和(3)对CGA和Hercules图形适配器都能得到满足。对EGA/VGA图形适配器除图形过份简单的情况外,大多可得到满足,大量实际证明,对16K位图的图形进行以上压缩后,被压缩的图形文件仅有1.6K左

右, 其压缩效果是令人振奋的。

五、图形文件的还原和直接显示

首先我们会绍在不用还原被压缩的图形文件的情况下, 屏幕图形的快速显示。

为快速地在屏幕上绘出图形, 我们首先把屏幕缓冲区的内容全部清零, 而后只需根据图形文件提供的数据, 在屏幕缓冲区内的相应像素点上填上 1 的值即可。

根据图 (4) 所示的被压缩后的图形文件的格式, 我们首先从文件中取出前二字节以确定整个图形文件的长度, 然后我们取出图 (4) 中第三到第六个字节, 并取第三个字节的最高位以判断其后紧跟的数据是 $N_0(1)$ 还是 $N_1(1)$, 屏蔽掉第三字节的最高位, 第三字节的低七位与第四、第五、第六字节共同构成了 $N_0(1)$ 、或 $N_1(1)$ 的值。以后 $N_0(5)$ 和 $N_1(1)$ 将交替排列。

从被压缩后的图形文件中读得的 $N_0(1)$ 、 $N_1(1)$ 的值为计算机内码, 我们首先必须把它转换为正整数, 转换为正整数的过程是一个字节一个字节进行的, 以判断相应的值是一个字节表示的还是二字节表示的。

设图形文件中第三字节的最高位为 0, 即其后的数值以 $N_0(1)$ 、 $N_1(1)$ 的顺序排列。则我们可根据 $N_0(1)$ 算出像素值为 1 的起始点, 根据 $N_1(1)$ 算出像素值为 1 的结束点, 程序中只需用画线语句从起始点到结束点画一条长度为 $N_1(1)$ 的直线即可。起始点的具体算式为

$$X_s = X_0 + N_0(1) - X_{\max} \cdot \text{int} \{ (X_0 + N_0(1)) / X_{\max} \}$$

$$Y_s = Y_0 + \text{int} \{ (X_0 + N_0(1)) / X_{\max} \} \quad (6)$$

结束点的具体算式为

$$X_e = X_s + N_0(1) - X_{\max} \cdot \text{int} \{ (X_s + N_0(1)) / X_{\max} \}$$

$$Y_e = Y_s + \text{int} \{ [X_s + N_0(1)] / X_{\max} \} \quad (7)$$

式中 X_0 、 Y_0 为上一次画线时的结束点, X_{\max} 表示一条扫描线上像素点的个数, 对 CGA 适配器, X_{\max} 取 640。

我们这时采用的图形直接还原显示法, 由于省去了像素值为 0 的点不画, 而在大部分图形中像素值为 0 的点占大多数, 因此图形的绘制速度很快, 以 BASIC 语言为例, 仅比用 BLOAD 语句把图形文件调入内存的速度稍慢, 而经过二次压缩后的图形文件直接显示速度与只经过一次压缩后的图形文件直接显示速度比较, 其细微的差别是程序操作者难以察觉的。

在还原了屏幕图形后, 如果需还原以位图表示的图形文件, 只需把屏幕缓冲区中的内容以位图的形式存入图形文件即可。

六、结论

随着计算机显示屏呈现越来越高的分辨率, 计算机图形的绘制越趋精美、漂亮, 然而, 这些精美、漂亮的图形

的保存和传输却是使软件工作者深感头疼的问题。本文介绍的重点针对黑白模式下图形文件的压缩方法, 其压缩效果是显著的。

本文介绍的图形压缩方法分二步进行。第一次压缩后的图形数据若存入图形文件中也可用于直接绘图显示, 其压缩比为 4:1, 经第二次压缩后的图形文件效果更加显著, 与位图储存方式相比其压缩比为 10:1, 作者统计了 120 幅在 CGA 上作出的控制理论试题的图形文件, 以本文介绍的图形压缩方法得到的图形文件的平均长度为 1583 bytes, 而原用位图方式储丰的图形文件为 16392 bytes。

图 (5) 所示为一控制理论试题图形, 被压缩后的文件长度为 2060 bytes。

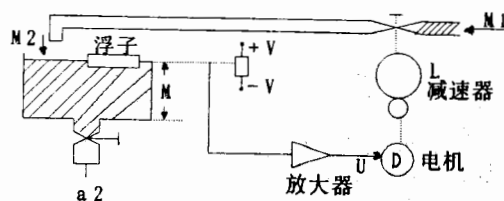


图 (5)

本文介绍的图形压缩方法作者已用于试题库集成环境管理软件中多年, 其使效果是非常明显的, 在一张容量为 360k 的普通低密盘上就可存下试题图形 200 余幅。

文中还介绍了从被压缩的图形文件直接作图的方法, 其图形还原的速度是很快。

七、参考文献

- (1) 沈美明等, IBM PC (0520) 汇编语言程序设计, 清华大学出版社, 1987
- (2) 张福炎等, IBM PC 的原理与应用 (续二), 南京大学出版社, 1991
- (3) 张昌平, 游戏图形图象的巧妙再利用, 《电脑》1991 年第 2 期

华大电脑技术服务部

我服务部以服务于教学、服务于用户为宗旨、优惠提供 APPLE II、中华学习机、IBM PC/XT、AT 系列计算机、主机售后长期保修、多种型号 24 针点阵中英文打印机、金山 II 型汉卡及办公桌面排版轻印刷系统、免费培训排版技术、承办软硬件培训班、提供技术咨询、硬件维修、继续赠送 APPLE II、CEC-I、IBM 类系统、工具、教学应用软件和软件使用说明书。

本服务部以最低之价格、最快之速度、最佳之信誉服务于广大用户, 请函索报价单。

联系地址: 长沙市劳动东路 128 号华大电脑技术服务部 范延华

或: 长沙重型机器厂长重中学计算机室 单振宇
(长沙市东塘广场乘 8 路车矿山站下)


```

10 HGR,HCOLOR=7
15 FOR U=770 TO 790: READ X: POKE U,X: NEXT
DATA 173,48,192,136,208,5,206,1,3,240,9,202,208,245,
174,0,3,76,2,3,96
20 HPLLOT 0,10 TO 40,10 TO 40,159 TO 279,159:HPLLOT
50,159 TO 50,10 TO 60,10 TO 60,159
25 HPLLOT 279,10 TO 279,159
30 FOR I=7676 TO 7701
40 READ X:POKE I,X:NEXT
60 POKE 232,252: POKE 233,29
65 HCOLOR=0:HPLLOT 50,0 TO 50,10:HPLLOT 60,0 TO
60,10: HCOLOR=7
70 DATA 1,0,4,0,54,62,36,60,54,7,32,44,36
80 DATA 49,46,36,53,54,54,46,32,36,44,50,54,0,
85 SCALE=1: ROT=1: DRAW 1 AT 6,6
87 SCALE=1: ROT=1: DRAW 1 AT 54,6
90 VTAB 21: INPUT "VO=? (0<VO<25):";VO:IF
VO<0 OR VO>25 THEN 90
100 T=SQR(28):GOSUB 310
110 FOR I=0 TO T STEP .5
120 XO=VO*I+50:Y=5*I*I+10
130 SCALE=1: ROT=1: DRAW 1 AT XO,Y
140 SCALE=1: ROT=1: DRAW 1 AT 45,Y
144 NEXT
150 DRAW 1 AT 45,154: DRAW 1 AT VO*(I+.2)+50,154
160 V1=4/5*VO:V2=9/5*VO
170 SCALE=1: ROT=1: DRAW 1 AT 54,6
180 GOSUB 310
190 POKE 768,192:POKE 769,70:CALL 770
195 RESTORE
200 FOR N=0 TO T STEP .5
210 X1=V1*N+50:X2=V2*N+50:Y=5*N*N+10
220 SCALE=1: ROT=1: DRAW 1 AT X1,Y
230 SCALE=1: ROT=1: DRAW 1 AT X2,Y
233 NEXT
240 DRAW 1 AT V1*(N+.2)+50,154
246 IF VO<22 THEN DRAW 1 AT
V2*(N+.2)+50,154:GOTO 250
248 DRAW 1 AT 276,154
250 PRINT "DO YOU WANT TO COPY BY PRINTER
?(Y/N)": GET A$
260 IF A$="N" THEN 280
270 POKE 1913,65:PR#1:PRINT CHR$(17); CHR$(
(24):PR#0
280 PRINT "DO YOU WANT TO GO ON?(Y/N)":
GET B$
290 IF B$="Y" THEN 90
300 TEXT:HOME:VTAB 11:HTAB 14:PRINT"THANK

```

```

YOU!!!: VTAB 12:HTAB 15:PRINT "BYE BYE!!!"VTAB
23:END
310 FOR J=3 TO 53 STEP 10: DRAW 1 AT J,6
320 NEXT: RETURN

```

电脑剪贴画

福建省南平师范 李 锋

不少软件封面的图形十分精美，遗憾的是总有一些有关设计者的字符，以致不能方便地为我所用。笔者从中央电视台《七巧板》栏目中得到启发：何不搞个高分辨图形的剪贴程序，对现有图形进行剪贴，经过尝试，编写出以下的两个程序（清单附后）。

该程序能从高分辨某一页的图案中剪下一个矩形画块，放在内存\$6000中，也可将放在该范围的画块贴回高分辨任一页的图案之中。

在监控状态下键入程序，并以JT.OBJ为名存盘，然后，在APPESoft状态下打入程序二，并运行；则屏幕将显示“J OR T”要你回答是剪还是贴。在剪的状态下需回答画块的在的页数，画块左上角的坐标x, y ($a \leq x < 280$, $0 \leq y < 192$)，画块横向长度HL，画块纵向长度VL。在贴的状态下只需回答所贴图案的页数和画块左上角的坐标。

本程序操作简便，笔者曾利用它将“中华超级汉字系统”的封面之长城图案，改装到自己的软件中去。

程序一

```

0300-A0 00 B1 B8 48 C8 20 98
0308-D9 68 C9 41 D0 03 4C 47
0310-03 C9 42 D0 03 4C 68 03
0318-4C C9 DE 20 B9 F6 85 02
0320-86 00 84 01 A5 02 85 05
0328-A5 03 85 1B A5 04 85 1C
0330-A5 06 85 08 A5 07 85 09
0338-60 A5 05 A6 00 A4 01 20
0340-11 F4 A5 04 85 1C 60 20
0348-1B 03 20 39 03 A0 00 B1
0350-08 E6 08 D0 02 E6 09 A4
0358-E5 91 26 E6 E5 C6 10 D0
0360-EC E6 05 C6 1B D0 E3 60
0368-20 1B 03 20 39 03 A4 E5
0370-B1 26 E6 E5 A0 00 91 08
0378-E6 08 D0 02 E6 09 C6 1C
0380-D0 EC E6 05 C6 1B D0 E3
0388-60

```

程序二

```

10 PRINT CHR$(4)"BLOADJT.OBJ": POKE 1013,76: POKE 1014,0:
POKE 1015,3: POKE 6,0: POKE 7,96
20 HOME
30 VTAB 4: HTAB 10: PRINT "J OR T": GET A$

```

CEC-I、APPLE 机磁盘螺旋轨的原理与制作

浙江省委党校 朱桂棋

CEC-I、APPLE 机使用的 DOS 系统，自 1978 年问世至今，尽管 DOS 版本几经更新，但磁盘轨道间距却毫无改变，然而，软件开发者却早已研究起轨道的间距问题，并制作出了许多特殊格式的轨道，用于软件的加密。就目前来说，一种经所谓“螺旋轨”（类似的还有“唱片轨”）的特殊轨道格式加密的磁盘，没有任何 COPY 软件能够自动复制，除非采用软件启动跟踪的方法，人工分析，才有可能解破。所以，熟悉、制作螺旋轨，对于软件加密，保护软件的合法权益，意义重大。

一、标准轨道与螺旋轨道的相异点

DOS 系统所设定的标准轨道格式如图 1 所示，若干个轨道组成若干个同心圆环，其中最外面的一个圆环为 0 道，由驱动器机械定位板确定，从外往里依次为 0、1、2……34 道。这些轨道之间的距离 (a) 为 0.527mm 左右，由于驱动器磁头存在着一定的磁隙，轨道间距不能做得太小，否则会引起道与道之间数据的覆盖。

螺旋轨或唱片轨，名称不同，轨道格式基本相同，如图 2 所示，每个轨道不是一个完整的圆环，而是一段圆弧。如果把这些圆弧头尾相接，则近似于一个平面螺旋，故名曰为螺旋轨或唱片轨。

在螺旋轨格式下，如果使某一轨（如图 2 的 X 轨）与其直径方向的相邻轨（如图 2 的 Y 轨）仍保持标准格式下的轨距（图 1 的 a），则该轨与其头尾相串轨（如 Z 轨）的轨距（图 2 的 b）可以大大减少，其轨距只要符合不等式 $b > a/k$ (k 为一个圆周上的螺旋轨数目，且 $k < 4$) 便可保证所有轨道的数据互不覆盖。

二、磁头进、退道的原理

磁头沿磁盘半径方向的移动，是由步进电机的转动实现的。步进电机的定子共有 4 个绕组，分别为 $\Phi 0$ 、 $\Phi 1$ 、 $\Phi 2$ 、 $\Phi 3$ 相，图 3 为经简化的示意图，图中的开关 K1—K5 在实际电路中是不存在的，仅仅是逻辑上对应于一些

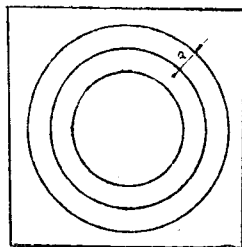


图 1

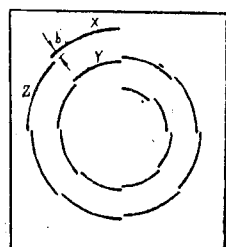


图 2

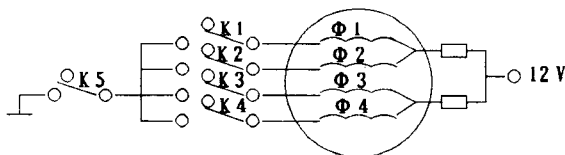


图 3

软开关，步进电机是否工作，以及 4 相的通断与否，可通过访问软开关实现。软开关的功能见表一。步进电机各相以 $\Phi 0$ 、 $\Phi 1$ 、 $\Phi 2$ 、 $\Phi 3$ 的顺序循环通电，磁头将从外园向内园走，道号增加；以 $\Phi 3$ 、 $\Phi 2$ 、 $\Phi 1$ 、 $\Phi 0$ 的顺序循环通电，磁头则从内往外走，道号减少。进行一次通断操作，磁头便移动 $1/2$ 标准轨距。如某一时刻是“ $\Phi 0$ 通 $\Phi 1$ 断 $\Phi 2$ 断 $\Phi 3$ 断”，当改为“ $\Phi 0$ 断 $\Phi 1$ 通 $\Phi 2$ 断 $\Phi 3$ 断”时，磁头便前进 $1/2$ 标准轨距。要走过一个标准轨道，必须再进行一次操作。

DOS 系统的走道程序设计得比较复杂（其目的是为了走道平稳、快速），但基本原理仍是如此。

三、螺旋轨的技术方案和实用程序

如果把图 2 中的 b 值定为与标准轨距 a 相等，那么，螺旋轨走道和程序的编写毫不困难，利用 DOS 走道程序便可。这种格式的螺旋轨，用 DISK NUMCHER PLUS 一类的 COPY 软件便可轻易复制，所以，只有 b 小于 a 才有价值。b 定为何值呢？应该按照步进电机能够达到的最小的步进量来确定。步进电机是按步旋转的，由前所知，反映到磁头上的能够达到的最小步进量是 $1/2$ 标准轨距。值得庆幸的是，利用特殊的方法，可以使磁头最小步进量达到 $1/4$ 标准轨距，其原理是：某相与其邻相同时通电，则步进电机在转动时，由于转子磁极在转动到二相之间时，二相对磁铁的吸引力相等，达到动态平衡，磁头便固定不动，实现了走 $1/4$ 标准轨距。具体地，如按以下顺序操作，便可实现 b 为 $1/4$ 标准轨距的进道：

“ $\Phi 0$ 通 $\Phi 1$ 断 $\Phi 2$ 断 $\Phi 3$ 断 $\rightarrow \Phi 0$ 通 $\Phi 1$ 通 $\Phi 2$ 断 $\Phi 3$ 断 $\rightarrow \Phi 0$ 断 $\Phi 1$ 通 $\Phi 2$ 断 $\Phi 3$ 断 $\rightarrow \Phi 0$ 断 $\Phi 1$ 通 $\Phi 2$ 通 $\Phi 3$ 断……”

上面的“ \rightarrow ”表示走了 $1/4$ 标准轨距，对于退道，循环

```
40 IF A$="J" THEN 70
50 IF A$="T" THEN 130
60 MUSIC 150,150:GOTO 30
70 INPUT "PAGE(1/2)";A:POKE 230,A*32
80 INPUT "SETX,Y";X,Y
90 INPUT "PIC HL";HL:POKE 4,INT(HL/7)
100 INPUT "PIC VL";VL:POKE 3,VL
110 & BX,Y
120 END
130 INPUT "PAGE(1/2)";A:POKE 230,A*32
140 INPUT "SETX,Y";X,Y
150 & AX,Y
160 END
```

顺序反过来便可。磁头走到二相都通电的轨道时，应保持各相状态不变，进行该轨的读写操作。功能强大的 EDD 之类的 COPY 软件，虽然也能走 1/4 标准轨距，却复制不出螺旋轨磁盘。原因很简单，这类软件只能对整个园周轨道进行复制，这样，写操作时把相邻 1/4 轨距的螺旋轨给覆盖掉了。

制作螺旋轨最后要涉及的就是扇区数问题了,显然,一个螺旋轨上的扇区数不可能仍是16个。由于b已定为1/4标准轨距,那么,一个圆周上安排4个螺旋轨是可能的,它满足 $b=a/k$ 式。这样,长度为1/4周长的螺旋轨,应该容纳 $16/4$,即4个扇区。实际制作时,考虑到走道需要时间,走道结束后还需延时一段时间以待磁头稳定,故1个螺旋轨上安排3个扇区是比较合理的。

最后, 给出螺旋磁磁盘的制作程序。限于篇幅, 仅给出最重要的格式化程序, 至于读写扇区程序, 读者可参考本格式化程序及 DOS 系统的读写程序自行编写。如有困难又有兴趣, 可与作者联系。

格式化程序以源程序的形式列出，目的在于便于理解，供编制读写程序时参考。下面对程序作些解释。

行号 11—18 的指令使磁头归 0 道；20—27 行指令检测磁盘是否写保护，如是，响铃、显示“ERR”字样后停止执行；29—37 行指令通过调用 DOS 系统的有关程序，完成 DOS 数据缓冲区清 0 等工作；38—62 行是程序的核心，尽管只有 60 几个字节，却完成 140 个螺旋轨道的进道，并在进 1 道后，调用写道子程序 WT；WT 子程序完成在某一螺旋轨道写上 3 个扇区，子程序要调用 DOS 系统的写位址栏子程序 \$BC56 及写 1 个扇区子程序 \$B82A。为防止螺旋轨扇区与标准 DOS 扇区发生混淆（格式化螺旋轨时并未把整个磁盘重写，原盘上的信息会留下一部分），经 WT 程序写入的地址栏标记为“D5、AA、9A”，并非标准 DOS 下的“D5、AA、96”，在编制读写扇区程序时应加以注意。至于螺旋轨的其它格式与标准 DOS 无异，为了加强加密效果，扇区格式可以做更多的修改。

软开关地址	功能
C080+S0	K1 断, $\Phi 0$ 相断电
C081+S0	K1 通, $\Phi 0$ 相通电
C082+S0	K2 断, $\Phi 1$ 相断电
C083+S0	K2 通, $\Phi 1$ 相通电
C084+S0	K3 断, $\Phi 2$ 相断电
C085+S0	K3 通, $\Phi 2$ 相通电
C086+S0	K4 断, $\Phi 3$ 相断电
C087+S0	K4 通, $\Phi 3$ 相通电
C088+S0	K5 断, 同步电机及步进电机断电
C089+S0	K5 通, 同步电机及步进电机通电

3C3E 注: SO 为槽号 $\times 16$ 值

SOURCE FILE: INITB

```

0000: 1 *****
0000: 2 *INIT.SPIRAL.DISKETTE...
0000: 3 *DESIGNED.BY.ZHU.GUI.QI.*
0000: 4 *****
0000: 5 *
002B: 6 SLOTX16 EQU $2B
0044: 7 TRACK EQU $44
005F: 8 SECTOR EQU $3F
-----
NEXT OBJECT FILE NAME IS INIT.OBJO
-----
2000: 9 ORG $2000
2000: 10 *-----
2000: A2 60 11 LDX ##60
2002: 8A 2B 12 STX SLOTX16
2004: BD 89 CO 13 LDA $C0B9,X
2007: BD 8A CO 14 LDA $C0BA,X
200A: A0 54 15 LDY ##54
200C: 8C 78 04 16 STY $478
200F: A9 00 17 LDA ##00
2011: 20 A0 B9 18 JSR $B9A0
2014: BD 8D CO 19 LDA $C0BD,X
2017: BD 8E CO 20 LDA $C0BE,X
201A: 10 09 21 BPL INIT
201C: 22 *-----
201C: 23 *WRITE.PROTECTED.ERR
201C: 20 2D FF 24 JSR $FF2D
201F: A6 2B 25 STOP LDX SLOTX16
2021: BD 89 CO 26 LDA $C0B8,X
2024: 60 27 RTS
2025: 28 *-----
2025: A9 FE 29 INIT LDA ##FE
2027: 85 41 30 STA $41
2029: A9 60 31 LDA ##60
202B: BD CB BE 32 STA $BECB
202E: 20 B5 BE 33 JSR $BEB5
2031: A9 A9 34 LDA ##A9
2033: BD CB BE 35 STA $BECB
2036: A9 9A 36 LDA ##9A
2038: BD 84 BC 37 STA $BC84
203B: A0 00 38 LDY ##00
203D: 84 44 39 STY TRACK
203F: 98 40 INIT0 TYA
2040: 29 03 41 AND ##03
2042: 0A 42 ASL A
2043: BD B1 20 43 STA MEMORY
2046: 05 2B 44 ORA SLOTX16
2048: AA 45 TAX
2049: BE B2 20 46 STX MEMORY1
204C: BD B1 CO 47 LDA $C0B1,X
204F: AD B1 20 48 LDA MEMORY
2052: C9 06 49 CMP ##06
2054: F0 0F 50 BEQ INIT1
2056: BD B3 CO 51 LDA $C0B3,X
2059: 20 B2 CO 52 JSR WT
205C: AE B2 CO 53 LDX MEMORY1
205F: AD B0 CO 54 LDA $C0B0,X
2062: 4C 70 20 55 JMP INIT2
2065: A6 2B 56 INIT1 LDX SLOTX16
2067: BD B1 CO 57 LDA $C0B1,X
206A: 20 B2 58 JSR WT
206D: BD B6 CO 59 LDA $C0B6,X
2070: 20 B2 20 60 INIT2 JSR WT
2073: CB 61 INY
2074: 4C 3F 20 62 JMP INIT0
2077: 63 *-----
2077: 64 INITEND PLA
2078: 68 65 PLA
2079: 68 66 PLA
207A: A9 96 67 LDA ##96
207C: BD 84 BC 68 STA $BC84
207F: 4C 1F 20 69 JMP STOP
2082: 70 *-----
2083: 98 71 WT TYA
2085: 4B 72 PHA
2087: A9 1A 73 LDA ##1A
208A: 20 AB FC 74 JSR $FCAB
208B: E6 14 75 INC TRACK
208B: A5 44 76 LDA TRACK
208D: C9 04 77 CMP ##04
208F: 90 10 78 BCC WTEND
2091: C9 BD 79 CMP ##8D
2093: B0 E2 80 RCS INITEND
2095: A9 FF 81 LDA $FFF
2097: B5 3F 82 STA SECTOR
2099: E6 3F 83 WT1 INC SECTOR
209B: A5 3F 84 LDA SECTOR
209D: C9 03 85 CMP ##03
209F: B0 03 86 BNE WT2
20A1: 60 87 WTEND PLA
20A2: A8 88 TAY
20A3: 60 89 RTS
20A4: A0 10 90 WT2 LDY ##10
20A6: A6 2B 91 LDX SLOTX16
20A8: 20 56 BC 92 JSR $BC56
20AB: 20 2A B3 93 JSR $B82A
20AE: 4C 59 20 94 JMP WT1
20B1: 95 *-----
20B1: 96 MEMORY DFB $00
20B2: 97 MEMORY1 DFB $00

```

*** SUCCESSFUL ASSEMBLY: NO ERRORS

中华机配用 24 针汉字打印机的方法

福建师大实验中心 吴庆祥

中华学习机一般都配用 9 针 80 列字符打印机, 将它用于文字处理时, 汉字不如 24 针打印机打印的汉字美观, 而且不能打印宽行文稿及油印蜡纸, 常常感到不便。本文介绍一种将 24 针带汉字库的打印机连接到中华机上使用的方法, 以克服上述的不足之处。

许多汉字打印机 (如 LQ1000K 等) 都具有 RS232 接口, 因此我们只要在中华机上插上一块 RS232 接口卡就可以将它们连接起来。但这样还无法打印出中华学习机的汉字, 因为中华学习机的汉字内码不是国标码, 要将中华机内的汉字送到汉字打印机打印出来, 就必须将中华机汉字内码转换成国标码, 因此我们用 6502 汇编语言编写了一个具有汉字内码转换功能的打印驱动程序 (程序清单见附录), 只要按清单中的机器代码打入中华机后, 取名为 PRINT 用 BSAVE 存到磁盘上, 要用时打入 BRUN PRINT, 就可以将它装入内存并设置 \$300 处的跳转向量表, 此后, 汉字打印的联机与脱机可用下列指令完成:

CALL 768 接通打印机相当于 PR#1

CALL 771 脱开打印机相当于 PR#0

在程序中的应用, 如下列程序所示:

10 CALL 768; REM 接通打印机

20 PRINT "汉字送到 24 针打印机上"

30 CALL 771: REM 取消发送到打印机的方式

40 END

中华机 RS232 接口与汉字打印机 RS232 接口的连接可采用三线接法, 即先分别将中华机 RS232 接口和打印机的 RS232 接口的 4、5、8 脚连在一起, 然后用一电缆线将中华机 RS232 第七脚信号地连接到打印机 RS232 的信号地第七脚, 中华机 RS232 的数据发送线第二脚接到打印机 RS232 的数据接收线第三脚, 打印机的准备就绪线第 2 脚接到中华机的 RS232 数据接收线第三脚上。如果所采用的是 LQ1000K 打印机, 它的 RS232 接口是一个梅花插座, 它与中华机 RS232 卡的连接方法如下: 从梅花插座的缺口处起, 顺时针方向数, 第二孔为准备接收数据线, 可接到中华机 RS232 卡的第三脚上; 第三孔是接收数据线可接到中华机 RS232 卡的第二脚上; 第五孔是信号地可接到中华机 RS232 卡的第七脚上, 中心孔为保护地可接到中华机 RS232 卡的第一脚上。用这种接法制作一条 25 芯转换成梅花插座的电缆则可将打印机与中华机连接起来。

此外, 利用 PRINT 程序还可将中华机上的文件资料传送到 IBM PC 及其兼容机上, 也可利用这个程序要将中华机文件送到华光激光照排系统上, 印刷出高质量的印刷品来。

* 9000 913D

9000-A9	00	85	73	A9	90	85	74
9008-A2	00	BD	16	90	9D	00	03
9010-E8	E0	0C	D0	F5	60	4C	22
9018-90	4C	A8	90	4C	C0	90	4C
9020-B3	90	A5	36	CD	38	91	F0
9028-08	8D	3C	91	AD	38	91	85
9030-36	A5	37	CD	39	91	F0	11
9038-85	37	AD	39	91	85	37	A9
9040-23	8D	90	C0	A9	11	8D	90
9048-C0	60	29	7F	C9	7F	F0	0B
9050-2C	40	91	D0	11	29	7F	20
9058-91	90	60	A9	FF	8D	40	91
9060-A9	00	8D	42	91	60	8D	43
9068-91	8A	48	98	48	AD	43	91
9070-20	92	C3	09	80	20	91	90
9078-EE	42	91	AD	42	91	C9	02
9080-B0	08	68	A8	68	AA	AD	43
9088-91	60	A9	00	8D	40	91	F0
9090-F1	48	AD	90	C0	29	03	F0
9098-F9	68	8D	91	C0	60	AD	90
90A0-C0	4A	90	FA	AD	91	C0	60
90A8-AD	3C	91	85	36	AD	3D	91
90B0-85	37	60	AD	3E	91	8D	55
90B8-AA	AD	3F	91	8D	56	AA	60
90C0-AD	55	AA	CD	3A	91	F0	09
90C8-8D	3E	91	AD	3A	91	8D	55
90D0-AA	AD	56	AA	CD	3B	91	F0
90D8-0C	8D	3F	91	AD	3B	91	8D
90E0-56	AA	20	3F	90	60	2C	44
90E8-91	D0	15	20	9E	90	8D	43
90F0-91	C9	A0	B0	14	A9	00	8D
90F8-41	91	AD	43	91	09	80	60
9100-A9	00	8D	44	91	AD	43	91
9108-60	AD	41	91	D0	0E	A9	FF
9110-8D	44	91	8D	41	91	20	21
9118-91	A9	7F	60	A9	00	8D	41
9120-91	98	48	8A	48	AD	43	91
9128-20	89	C3	29	7F	8D	43	91
9130-68	AA	68	A8	AD	43	91	60
9138-4A	90	E6	90	00	00		

幻方百字诀

上海市闵行区青少年科技指导站 阴 申

传说四千多年前大禹治水，洛水里浮出一只大乌龟，驮来了一幅“洛书”：龟背上刻着一些奇妙的数字，横三行、纵三列以及两对角线上的三个数字之和都等于15（见图1）。从此，“洛书”便成了吉祥的象征。随着“洛书”的不断发展，这些奇妙的纵横图被介绍到欧洲，取名“幻方”，而且一直引起人们强烈的兴趣。

8	1	6
3	5	7
4	9	2

图 1

这些数字上的“迷宫”，具有许多奇妙的性质，格内数字从1连贯，各横行纵列及两对角线上数字之和都相等。如何构造幻方，令许多人伤透脑筋，这里就介绍一种奇数阶实数幻方（如3×3，5×5等等）的制作方法。

请先记下一个“百字诀”：实数幻方奇，始于顶中间，规律要牢记，不能差毫厘，上下和左右，全都要相连；顶接下右底，右往左上延，相差只一格，不可有偏离；数码依次进，尽向右上偏，中途右上角，这是关键点；前面若受阻，下降一格继；数字禁短缺，重复要避免，往复循序转，胜利定可期。

下面我们就这一百字诀的应用举一实例，如图2。这是一个5阶实数幻方（纵横图），图中纵横均为5格，共5×5=25格，每格填一个数字（1-25），其纵横每一条线以及对角线之和都是65，下面我们对照图3分析：

17	24	1	8	15
23	5	7	14	16
4	6	13	20	22
10	12	19	21	3
11	18	25	2	9

图 2

按照百字诀：1应在上方中间（始于顶中间），2应在1的正下偏右一格（顶接下右底，相差只一格），3应在2的右上方（数码依次进，尽向右上偏），4应在3上面一行的最左边一格（右往左上延，相差只一格），然后填5，6应在5的下面一格（中途若受阻，下降一格继），然后再填7到15，16应在15的下面一格（中途右上角，这是关键点）……直至填好25。

根据百字诀我们可以写出程序一。实际上，“中途右上角，这是关键点；前面若受阻，下降一格继”可以并成“循序转一圈，下降一格继”，设N为所填幻方的阶数，则一圈即为N格。于是，只要遇到N的倍数，则下降一格，如图2中的6，16等。如此，程序一就可改写成程序二。

图4是图3的填写顺序，图中实线为第一循环，虚线为第二循环。

		1		15
	5			16
4	6			
				3
			2	

图 3

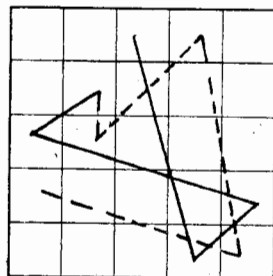


图 4

程序一

```

10 INPUT "N=";N
20 IF N/2=INT(N/2) THEN 10
30 DIM A(N,N)
40 I=1:J=(N-1)/2:A(I,J)=1
50 FOR K=2 TO N*N
60 I=I-1:J=J+1
65 IF I=0 AND J=N THEN I=2:J=N-1
70 IF A(I,J)>0 THEN I=I+2:J=J-1
80 IF I=0 THEN I=N
90 IF J=N THEN J=0
100 A(I,J)=K: NEXT
110 FOR I=1 TO N
120 FOR J=0 TO N-1
130 PRINT TAB(J*4+1)A(I,J);
140 NEXT: PRINT
150 NEXT
    
```

程序二

```

10 INPUT "N=";N
20 IF N/2=INT(N/2) THEN 10
30 DIM A(N,N)
40 I=1:J=(N-1)/2:A(I,J)=1
50 FOR K=1 TO N*N-1
60 IF K/N=INT(K/N) THEN I=I+1:GOTO 80
70 I=I-1:J=J+1
80 IF I=0 THEN I=N
90 IF J=N THEN J=0
100 A(I,J)=K+1: NEXT
110 FOR I=1 TO N
120 FOR J=0 TO N-1
130 PRINT TAB(J*4+1)A(I,J);
140 NEXT: PRINT
150 NEXT
    
```

利用逻辑零扇区的 I/O 参数表 来修复硬盘主引导扇区的分区参数

广州市计算机应用研究所 胡向东

硬盘主引导扇区受破坏, 是 IBM 系列微机硬盘的常见故障。

常见的外部症状是不承认硬盘, 或者虽然认硬盘, 但连 DIR 这样的简单操作都无法进行。

造成这种故障的原因大致有如下四种:

第一, 是计算机病毒破坏所致。

例: 某药厂的一台 IBM PC/AT 机, 因受广大一号病毒感染, 硬盘主引导扇区完全受破坏。而在一般情况下, 广大一号病毒并不破坏主引导扇区的分区参数 (详见 90 年第六期《电脑》杂志胡向东所写《大麻、广大一号、六四病毒及打印病毒》), 但当病毒程序改写主引导扇区时突然受到干扰, 造成写盘时的缓冲区指针指错, 使得硬盘中的主引导扇区中, 既不是原机的主引导程序和分区参数, 也不是病毒程序, 而是杂七杂八的一些机器码。因此该机不但不能用 C 盘启动, 而且用软盘启动后也不能转到 C 盘。

第二, 是不成熟的解毒软件、免疫软件所致。

计算机病毒固然是社会一大公害, 但一些不成熟的解毒软件、免疫软件在社会上流行, 有时也害人不少!

例: 笔者曾随广东省计算机安全监察处的同志到某外轮公司, 该公司的电脑感染上大麻病毒后, 到某电脑公司用几百元购得一张消毒盘来消毒, 结果把一台 IBM PC/XT 的主引导扇区全部破坏, 该机虽然感染大麻病毒, 但未消毒前仍可用硬盘启动, 但消毒后则连 C 盘也不承认, 原因是解毒程序在设计上出了毛病, 有些因素在程序设计中没有考虑进去。

例: 某工厂在 IBM PC/AT 上运行某电脑公司的“牛痘”软件, 本意是想预防计算机病毒, 结果是吓出一身冷汗, 原来贮存有大量数据的 30 兆硬盘一下子变得无法进入, 经笔者实地调查, 原来是这个“牛痘”软件设计不完善, 把这部 AT 机的主引导扇区彻底破坏了。(注: 该软件在一些 XT 机上确实起到预防系统型病毒的作用, 笔者也听到一些用户对该软件的赞扬)。

第三, 是电压不正常所致。

例: 某公司电脑室, 一条外接电源线碰断后重接, 由于接线马虎, 造成短路, 接线处直冒火花。当时一台计算机正在读写硬盘, 造成失控, 把硬盘主引导扇区破坏而不认 C 盘。

例: 省公安厅某处机房, 突然停电, 操作人员只关主机而不关 UPS, 当恢复供电后, 开总闸, 开主机, UPS 送电电压不正常, 造成主引导扇区分区参数被破坏, 使得一

份贮存在硬盘中的统计报表几乎不能在当天报出。(笔者在另一工厂, 也遇到过类似的例子, 可见遇到停电时, 操作也不要掉以轻心)。

第四, 是误操作所致。

例: 某单位一年青人, 按照电脑杂志介绍的一些程序来运行, 由于一些参数设置错误, 直接破坏硬盘的主引导扇区, 使硬盘不能正常读写。

笔者在 91 年第一期《电脑》杂志上所写《IBM 系列微机数据维护技术研究》中, 已谈到过硬盘主引导扇区的恢复。

该文在这个问题上的几个要点是:

(1) 主引导扇区是由主引导程序和分区参数表两个部分组成。

(2) 主引导程序可以从其它 IBM PC 系列微机的硬盘中提取。

(3) 硬盘分区参数表由 4 个区, 每区 16 字节参数组成, 分区参数表以 55AA 结束, 每一个分区参数都有特定含义, 文中介绍了几种常见的 10 兆、20 兆、30 兆硬盘的分区参数表。

(4) 读者可根据提取出来的硬盘主引导程序和仿制出来的分区参数表, 根据 90 年第一期《电脑》杂志胡向东所写《硬化病毒的分析及排除方法》恢复硬盘主引导扇区。

但是, 由于硬盘的型号不同, 因此即使是相同容量的硬盘, 也有不同的物理参数, 这就给仿制分区参数表带来困难。

本文在这里介绍的是一种较通用的仿制硬盘分区参数表的方法, 这就是利用逻辑零扇区的 I/O 参数表来修复硬盘主引导扇区的分区参数。

在硬盘逻辑零扇区 0 字节偏移 0BH 处开始, 有一个占 19 个字节的硬盘 I/O 参数表。

在硬盘主引导扇区的分区参数受破坏, 系统不承认 C 盘时, 我们可以用 A 盘启动, 运用 debug 作工具, 运行本文提供的程序便可读出 C 盘的逻辑零扇区。(注: 在 C 盘主引导分区参数未破坏时, 用 debug 的 L 1000 201←便可实现同样的效果)

读出第一类型硬盘逻辑零扇区的程序是:

```
mov AX, 0201 ; 读一个扇区
mov BX, 1000 ; 调入内存 1000 处
mov CX, 0001 ; 0 道 1 区
mov DX, 0180 ; C 盘 1 面
int 13 ; 磁盘中断
```

以毒攻毒

——boot 区型病毒的症治

武汉工业大学 孙国保

boot 区型病毒具有以下特点：将正常的引导记录从磁盘的 boot 区移至磁盘的空扇区或某些固定的扇区处，而病毒程序占据原 boot 区。系统启动时，boot 区的病毒程序被读至内存 0000:7C00 处执行，主要完成病毒程序的高端驻留、中断向量的截留盗用，其中 int13（磁盘操作中）往往被修改指向病毒传染部分。boot 区型病毒往往由两部分组成：一部分在 boot 区，而另一部分在磁盘数据区中，后者由 boot 区病毒部分读入，两部分病毒合为一个完整的病毒程序。然后病毒程序从磁盘中将正常的 boot 区引导记录读至 0000:7C00 处执行正常的引导。boot 区型病毒是一种常见的病毒而且传染力极大。小球、合法大麻、巴基期坦智囊病毒等均为 boot 区型病毒。

对这类病毒的症治关键在于对病毒程序占有磁盘空间的回收，其中包括：(1) 找到正常的 boot 区引导记录，写回 boot 区。(2) 将数据区的病毒程序占有的空间释放。一般病毒程序人为设置病毒的占有簇为坏簇，因此要将该簇置为可用簇。

症治的主要思想：编制一类病毒程序 P 占据 boot 区。为模拟病毒程序驻留内存的真正地址，在编写类病毒程序 P 时，考虑到若 P 驻留内存高端，则病毒驻留高端的实际地址将下移，又由于进行 boot 区引导时，没有访问内存 0000:1C00 开始的内存区域，固可使 P 驻留于此。并截留盗用 int13，使得凡进行磁盘操作的入口参数均被显示或记录于工作区中。将含有类病毒程序的盘放于 A 驱动器，含有 boot 区型病毒的盘放于 B 驱动器中，通过 P 引导 B 盘的病毒 boot 区。由于磁盘读、写操作本质上总是要调用 int13 完成，因此病毒程序进行了哪些磁盘操作，进行操作的驱动器号、磁道号、扇面号、磁头号以及缓冲区首址均被显示出来。用 PCTOOLS 工具搜索上述指出的扇区，将含有 55AA 标记的 boot 区引导记录写回 boot 区，而对另一些扇区则获取其病毒源程序进行分析或改动其对应 FAT 表位置为 000 表示将簇释放，成为可用簇。注意到进行真正的 boot 区引导需要将首扇 FDT 表读入，据此测试是否含有 IBMBIO.COM 和 IBMDOS.COM，以

int 3 ;断点中断

读出第二类硬盘逻辑零扇区的程序是：

mov AX, 0201 ;读一个扇区

mov BX, 1000 ;读入内存 1000 处

mov CX, 0002 ;读 0 道 2 区

mov DX, 0080 ;读 C 盘 0 面

int 13 ;磁盘中断

int 3 ;断点中断

例：一台 IBM PC/XT 的 10 兆硬盘的逻辑零扇区的 I/O 参数如下：

0002 每扇区字节数 512

08 每簇扇区数为 8

0100 保留扇区数为 0001

02 FAT 表数目为 2 个

0002 根目录登记项数为 512 个

F350 本分区总扇区数 50F3H

F8 硬盘标志为 F8H

0800 每个 FAT 的扇区数为 8

1100 每道扇区数为 17

0400 磁头号数为 4

1100 隐含扇区数为 17

由 C 盘是活动分区，可知分区参数表第一个参数为 80H。

由每道扇区数 0011H，隐含扇区数 0011H，可知分区参数表中指示本分区首扇区所在磁头号、扇区号、磁道号的第二、第三、第四个参数分别是 01 01 00。

由硬盘容量 10 兆和使用 DOS，FAT 表每一簇用 1.5 字节表示，故分区参数表第五个参数即操作系统标记为 01（若 DOS 中，FAT 表每一簇用 2 个字节表示，则取标记 04，但在大容量硬盘在，若本分区总扇区数不能用两个字表示，取标记为 06）。

由本分区总扇区数 50F3H，隐含扇区数 0011H，可知共扇区 5104H = 20740 扇区。由每道扇区数 0011H，和磁头号 4，可得末扇区磁道号为 130H，故 CH 为 30H，CL = 11H + (01H) · (40H) = 51H，DH = 03H，从而得知分区参数表中第六、第七、第八个参数分别为 03 51 30，表示本分区末扇区所在磁头号、扇区号、磁道号。

由隐含扇区数 0011H，可得分区参数表中表示本分区已使用扇区数的第九、第十、第十一、第十二参数为 11 00 00 00。

由本分区总扇区数 50F3H，可得到硬盘分区参数表第十三、十四、十五、十六参数分别是 F3 50 00 00。

从而可以由逻辑零扇区的 I/O 参数表推断出硬盘主引导扇区的分区参数。

笔者曾利用上述方法修复硬盘的主引导扇区，收到了很好的效果。 ■

及它们顺序是否正确, 否则提示:

Non-system disk or disk error

然后进入等待更换上系统盘。由于用户盘中一般不含上述两个文件, 巧妙地利用这一特点可区分出哪些是病毒程序读、写磁盘的显示; 若为系统盘则可将 IBMBIO.COM 文件的首字符变为其它字符即可, 稍后再改回来。利用上述方法判定是否感染 boot 区病毒方法很简单: 若 B 驱动器中的磁盘感染了 boot 区病毒, 那么至少多了一个将真正 boot 区读入内存 0000: 7C00 处的磁盘操作, 因此, 磁盘操作次数 > 3 (反映在程序结果中不少于 3 行磁盘操作参数显示) 均表明感染了 boot 区型病毒, 否则为正常 boot 区。类病毒程序按以下方法得到: A > MASM boot; A > LINK boot; A > DEBUG boot.EXE-W 7C00 0 0 1; 写入 A 驱动器的磁盘 boot 区-Q

最后附上程序清单和测试结果。

类病毒程序 P

```
0BF1:7C90          JMP 7C2B
                   磁盘 I/O 参数表
                   磁盘基数表
0BF1:7C2B FA      CLI
0BF1:7C2C 33C0    XOR AX,AX
0BF1:7C2E 8ED0    MOV SS,AX
0BF1:7C30 BC007C  MOV SP,7C00
0BF1:7C33 16      PUSH SS
0BF1:7C34 07      POP ES
0BF1:7C35 8ED8    MOV DS,AX
0BF1:7C37 A11304  MOV AX,[0413]
0BF1:7C3A 2D0200  SUB AX,0002
0BF1:7C3D A31304  MOV [0413],AX
0BF1:7C40 B106    MOV CL,06
0BF1:7C42 D3E0    SHL AX,CL
0BF1:7C44 2DC007  SUB AX,07C0
0BF1:7C47 8EC0    MOV ES,AX
0BF1:7C49 BE007C  MOV SI,7C00
0BF1:7C4C 89F7    MOV DI,SI
0BF1:7C4E B90001  MOV CX,0100
0BF1:7C51 FC      CLD
0BF1:7C52 F3      REPZ
0BF1:7C53 A5      MOVSW
0BF1:7C54 8EC8    MOV CS,AX
0BF1:7C56 BE0500  MOV SI,0005
0BF1:7C59 31C0    XOR AX,AX
0BF1:7C5B FB      STI
0BF1:7C5C CD13    INT 13
0BF1:7C5E BA0100  MOV DX,0001
0BF1:7C61 B90100  MOV CX,0001
0BF1:7C64 B80102  MOV AX,0201
0BF1:7C67 BB007E  MOV BX,7E00
```

```
0BF1:7C6A CD13    INT 13
0BF1:7C6C 7308    JNB 7C76
0BF1:7C6E 4E      DEC SI
0BF1:7C6F 75E8    JNZ 7C59
0BF1:7C71 E85100  CALL 7CC5
0BF1:7C74 EBFEE    JMP 7C74
0BF1:7C76 31C0    XOR AX,AX
0BF1:7C78 8EC0    MOV ES,AX
0BF1:7C7A 0E      PUSH CS
0BF1:7C7B 1F      POP DS
0BF1:7C7C BE007E  MOV SI,7E00
0BF1:7C7F BF007C  MOV DI,7C00
0BF1:7C82 B90001  MOV CX,0100
0BF1:7C85 FC      CLD
0BF1:7C86 F3      REPZ
0BF1:7C87 A5      MOVSW
0BF1:7C88 B80000  MOV AX,0000
0BF1:7C8B 8ED8    MOV DS,AX
0BF1:7C8D A14C00  MOV AX,[004C]
0BF1:7C90 8B1E4E00 MOV BX,[004E]
0BF1:7C94 2E      CS:
0BF1:7C95 A3507F  MOV [7F50],AX
0BF1:7C98 2E      CS:
0BF1:7C99 891E527F MOV [7F52],BX
0BF1:7C9D A14000  MOV AX,[0040]
0BF1:7CA0 8B1E4200 MOV BX,[0042]
0BF1:7CA4 2E      CS:
0BF1:7CA5 A3547F  MOV [7F54],AX
0BF1:7CA8 2E      CS:
0BF1:7CA9 891E567F MOV [7F56],BX
0BF1:7CAD 8C0E4E00 MOV [004E] CS
0BF1:7CB1 0F064C00107D MOV WORD PTR
                                [004C],7D10
0BF1:7CB7 EA007C0000 JMP 0000:7C00

0BF1:7CC5 51      PUSH CX
0BF1:7CC6 53      PUSH BX
0BF1:7CC7 52      PUSH DX
0BF1:7CC8 B504    MOV CH,04
0BF1:7CCA 89C2    MOV DX,AX
0BF1:7CCC 250F00  AND AX,000F
0BF1:7CCF 3C0A    CMP AL,0A
0BF1:7CD1 7802    JS 7CD5
0BF1:7CD3 0407    ADD AL,07
0BF1:7CD5 0430    ADD AL,30
0BF1:7CD7 B40E    MOV AH,0E
0BF1:7CD9 BB0700  MOV BX,0007
0BF1:7CDC 9C      PUSHF
0BF1:7CDD 2E      CS:
```

```

0BF1:7CDE FF1E547F      CALL FAR[7F54]
0BF1:7CE2 B104           MOV CL,04
0BF1:7CE4 89D0           MOV AX,DX
0BF1:7CE6 D3E8           SHR AX,CL
0BF1:7CE8 FECD           DEC CH
0BF1:7CEA 75DE           JNZ 7CCA
0BF1:7CEC B8200E         MOV AX,0E20
0BF1:7CEF BB0700         MOV BX,0007
0BF1:7CF2 9C            PUSHF
0BF1:7CF3 2E            CS:
0BF1:7CF4 FF1E547F      CALL FAR[7F54]
0BF1:7CF8 5A            POP DX
0BF1:7CF9 5B            POP BX
0BF1:7CFA 59            POP CX
0BF1:7CFB C3            RET
-
0BF1:7D10 50            PUSH AX
0BF1:7D11 E8B1FF         CALL 7CC5
0BF1:7D14 89C8           MOV AX,CX
0BF1:7D16 E8ACFF         CALL 7CC5
0BF1:7D19 B201           MOV DL,01
0BF1:7D1B 89D0           MOV AX,DX
0BF1:7D1D E8A5FF         CALL 7CC5
0BF1:7D20 8CC0           MOV AX,ES
0BF1:7D22 E8A0FF         CALL 7CC5
0BF1:7D25 89D8           MOV AX,BX
0BF1:7D27 E89BFF         CALL 7CC5
0BF1:7D2A 53            PUSH BX
0BF1:7D2B BB0700         MOV BX,0007
0BF1:7D2E B80D0E         MOV AX,0E0D
0BF1:7D31 9C            PUSHF
0BF1:7D32 2E            CS:
0BF1:7D33 FF1E547F      CALL FAR[7F54]
0BF1:7D37 BB0700         MOV BX,0007
0BF1:7D3A B80A0E         MOV AX,0E0A
0BF1:7D3D 9C            PUSHF
0BF1:7D3E 2E            CS:
0BF1:7D3F FF1E547F      CALL FAR[7F54]
0BF1:7D43 5B            POP BX
0BF1:7D44 58            POP AX
0BF1:7D45 2E            CS:
0BF1:7D46 FF2E507F      JMP FAR[7F50]

```

附录 I

测试结果:

一、正常 boot 区引导

(AX CX DX ES:

BX result memory-size)

(1) 0000 0000 0001 0000: 0078 succ

(2) 0201 0006 0000 0000: 0500 succ

(3) Non-system disk or disk error

Replace and strike any key when ready

分析: (1) 表明: 进行磁盘系统复位成功

(2) 表明: 将 FDT 表首扇读入内存 0000: 0500 处成功

(3) 表明: 非系统盘, 提示用户

二、STONE 病毒引导

(1) Warning: 13Hentrance change! 9F80: 0015

(2) 0000 0000 0001 9F80 F000 succ 0002

(3) 0201 0003 0100 0000 7C00 succ

(4) your PC is now stoned!

(5) 0201 0001 0080 9F80 0200 fail

(6) 0000 0001 0080 9F80 0200 succ

(7) 0201 0006 0000 0050 0000 succ

分析: (1) 表明: STONE 病毒修改了 INT 13H 中断入口为 9F80: 0015

(2) 表明: STONE 病毒驻留内存高端, 共占 2KB

(3) 表明: 将真正的 boot 区 (0 道 1 面 3 扇区, 即 FDT 表最后一扇) 读入内存 0000: 7C00 处。

(4) 表明: 系统感染了 STONE 病毒这是 STONE

病毒的特征。

(5) 表明: STONE 病毒试图访问 C: 盘 (对 C 盘进行感染) 但未成功。

(6) (7) 为真正的 boot 区引导结果。

附录 II

磁盘 I/O 中断 INT 13H 参数表 (部分)

AH	操作	输入寄存器内容	结果寄存器
0	磁盘系统复位	无	无
1	读磁盘状态	无	(AL)=磁盘状态
2	读扇面到存储器	(DL)=驱动器号 (0-1,80H) (DH)=磁头号(0-1) (CH)=磁道号(0-39) (CL)=扇面号(1-9) (AL)=扇面数 (ES: BX)缓冲区地址	(AL)=读出的扇面数 CF=0: 操作成功 (AH)=0 CF=1: 操作失败 (AH)=状态
3	从存储器写到扇面	同读操作	同读操作

(程序文件名: ZD8.PRg)

充分利用反病毒工具，有效防止病毒传播

——介绍 Turbo Anti-virus v6.80a

广西玉林地区建设银行 吴小以 刘昌斌

一、引言

随着电脑在各项业务中的广泛应用，电脑病毒传播也更为流行。只要使用受带病毒感染的磁盘，该机就有可能成为病毒的扩散者，进而感染与之发生读/写关系的磁盘，扩散到别的计算机，引起病毒的广泛传播。

然而计算机病毒并不是那么可怕的。只要利用反病毒软件，建立完善的反病毒体系，就有可能防止病毒的污染。目前就有一套极为出色的反病毒软件—Turbo Anti V6.8A (Turbo 反病毒软件包 6.80a 版本)，只是目前还有不少用户尚未充分发挥其作用而已。

二、Turbo Anti-Virus V6.80a 功能简介

Turbo Anti-Virus V6.80a 是美国著名的软件开发公司 Borland (波兰) 公司的产品，是一套功能极强的反病毒软件包。它由四个部分组成：

- 1、Install 装配程序
- 2、Tsafse 真正安全实用程序
- 3、Bootsafe 引导区安全实用程序
- 4、Tntvirus 反病毒主体程序

其中后三者构成一个强大的反病毒体系。下面逐一介绍它们的功能：

1、Install

用于安装本反病毒软件包，假如不再需要，也可由它进行拆卸。

2、Tsafse 实用程序

为系统安全提供保证，它又以设置逻辑开关对系统资源进行保护，当系统资源受到侵犯或发生改变时（包括受到病毒的侵害），TSafe 将发出警告，由用户确定是正常的还是非法的改变，如属非法，使之失败（FAIL），达到避免病毒侵蚀的目的。Tsafse 可以设置以下开关：

(1) 磁盘格式化 (Format Diskettes)

处于 On 状态时，当格式化磁盘时发出警告，如确实要格式化磁盘，可照进行，否则强返中止。这主要用于对付磁盘杀手 (The Disk Killer Virus) 之类的病毒，这类病毒在一定条件下格式化受感染磁盘。

(2) 硬盘低级格式化 (Hd Low Level Format) 对低级格式化硬盘提出警告，避免对硬盘的非法低级格式化。

(3) 驻留程序 (Restore)

由于大多数病毒都属内存驻留型程序，所以此逻辑开关意义重大。当病毒企图把自己装入内存时，Tsafse 发出

警告，可以使病毒的装入失败。

(4) 一般写保护 (General Write Protect) 使任何对文件进行改变的企图破产。

(5) 使被污染文件失败 (Abort Infected Files) 即当受污染文件执行时强迫其中止，防止病毒扩散。

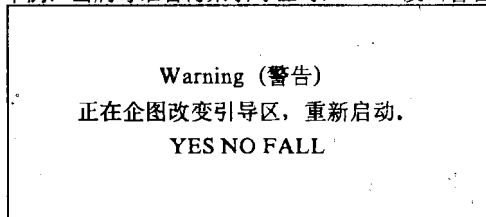
(6) 引导区污染 (Boot Sector Infection) 引导区已遭病毒污染时，发出警告，提供自动恢复功能。

(7) 保护引导扇区 (Protect Boot Sector) 当病毒企图污染引导扇区时发出警告。

(8) 保护可执行文件 (Protect Executable File)

文件类病毒主要污染可执行文件，此开关使当病毒企图进攻可执行文件时遭受失败。

举例：当病毒准备污染引导区时，Tsafse 发出警告



图：Tsafse 警告窗口示例

选择 Fail，使病毒误以为磁盘写保护而停止攻击，引导区未受污染。

3、Bootsafe 实用程序

它可以为指定的驱动器建立清洁的引导区和分区表的映象，既可以建立在硬盘，也可以建立在软盘上。一旦引导区遭到污染，Bootsafe 将发出警告并根据映象恢复引导区原先状态。

通过使用 Bootsafe，对引导区病毒的忧虑将成为过去。

4、Tntvirus 实用程序

是该软件包的主体，它的主要功能有：

- (1) 发现病毒 (“国产病毒”除外)
- (2) 清除病毒，此软件包一共可以发现、清除 154 种病毒，其中引导区型病毒 22 种，文件型病毒 132 种。
- (3) 对系统和所有文件提供免疫功能或者取消免疫功能；
- (4) 可以使引导区对某些病毒（如小球病毒）具有免疫功能；
- (5) 建立数据的完整性备份。

(6) 建立引导区、文件的映像;

(7) 恢复硬盘引导区 (对非病毒污染的损坏有效)。等等

综上所述, Turbo Anti Virus V6.80a 软件包具有防止系统受病毒侵害 (“国产病毒”有效) 发现并清除病毒, 提供免疫功能和恢复硬盘引导区等强大的反病毒功能, 利用它的 Tsafe Bootsafe 和 Tntvirus 实用程序可以构成一个有效的反病毒系统。此外, 它还有一个显著特点, 本身的所有可执行文件运行前将首先检查自身, 一旦病毒已污染了它们, 它们将诊断并消除之, 即具有自我保护功能。

三、“国产病毒”的克星——KILL

Turbo Anti-Virus V6.80a 有一个“缺点”, 它能防止国产病毒的污染, 但不能发现和清除已经侵犯“得逞”的“国产病毒”。

对付已经侵犯的国产病毒, 可以使用公安部研制并不断更新版本的病毒杀手 KILL 和病毒检测程序 SCAN。

四、充分利用现有工具, 防止病毒的污染

对银行各业务部门的电脑来说, 杜绝病毒的污染有特别的意义。那么如何利用 Turbo Anti-Virus V6.80a 有效地防止病毒的感染呢?

首先利用公安部的 KILL 对微机进行病毒清除, 以确保目前使用的机器的硬盘中无“国产病毒”。然后在各台微机上安装 Turbo Anti-Virus V6.80a 软件包。所要做的工

作有几点:

- (1) 利用 Install 实用程序在 C: 上安装 Turbo Anti-Virus V6.80a 软件包, 并把 Bootsafe 和 Tsafe 安装在 C: 的 Autoexec.bat 批处理文件中, 这样做的目的是: 每次启动机器都首先检查内存和引导区中是否有病毒, 如有马上清除, 保证进入的是一个清洁的环境; 启动时既装入 Tsafe 使系统资源得到有效保护。

- (2) 马上执行一次 Bootsafe, 以产生引导区和分区表的清洁映像, 供引导区受到污染时自动恢复。

- (3) 马上运行 Tntvirus, 置 Option 菜单中的 Check All fil 逻辑开关为 ON, 对硬盘所有文件进行 (包括数据文件等) 具有免疫功能。

以上工作, 可以保证计算机系统不受病毒的污染, 因为目前国内外发现的病毒只有两类型: 一种是引导区型病毒, 另一种是文件型病毒。Bootsafe 提供了引导区安全的保证, 而 Tntvirus 保证了所有文件不受污染, 在病毒企图侵犯计算机系统时, 由 Tsafe 发出警告, 于是警告窗口中的 Fail (失败) 选择使病毒的侵犯自动失败而起到防止病毒污染的作用。于是我们就有了一个安全的计算机系统, 见图 3。

此外还可利用 Tntvirus 或 Bootsafe 中提供的引导区恢复功能, 在软盘上建立引导区的影像, 以便当硬盘引导区由于其它的不可预见的原因损坏无法启动时作恢复。为此还建立了一张装有 turbo Anti-virus 的启动软盘。

五、计算机病毒并不可怕

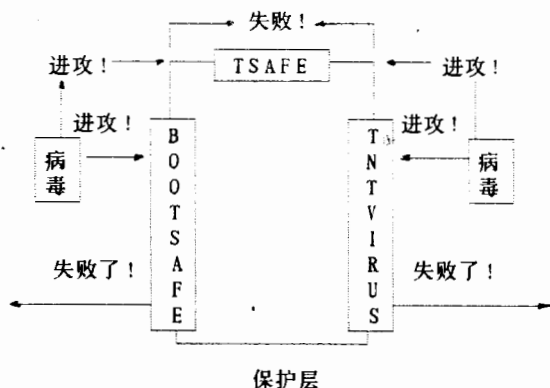


图3 安全计算机系统

计算机病毒多由那些精通计算机操作系统的人编制, 他们或为显示自己的编程技巧, 或者为了开恶意的玩笑等种种原因。一般说来, 计算机病毒都具有很巧妙的编程技巧, 因而具有易传播、隐蔽和能复制自身等特点。

然而, 归根到底, 计算机病毒也是一种计算机程序, 只不过是一种专门利用计算机系统本身的弱点攻击计算机资源的程序。因此病毒本身除了编制巧妙, 目的恶劣之外, 并没有什么可怕之处, 利用现有的反病毒软件工具, 是完全可能发现, 消除病毒的。

北京一读者来信:

Traveller 病毒是一个纯粹的良性病毒, 没有任何破坏功能, 其发作时, 仅打出一串字符而已, 这一串字符是: !!!!!!! ——>> Traveller (C) BUPT 1991.4 Don't panic I'm harmless <<——!!!!!!!

BUPT 的意思是北京邮电学院 (Beijing University of Posts and Telecommunications), 而非英国合从社 (British United Press)。

编者: 我们重申, 病毒的研究应在公安部门的领导下, 有组织地开展。分散的个别研究很容易产生新的病毒并使之传播, 给国家给个人带来巨大的损失。■

小 启

漏订明年《电脑》杂志的读者, 可直接汇款到本编辑部。

全年 6 期共 6.60 元 (含邮包费)。

编辑部地址: 广州石牌华南师范大学

邮政编码: 510631

用高级语言在 IBM-PC 机单显上作图

杭州应用声学研究所 刘廷范

一、前言

目前,带单色显示器及单色显适配器的 IBM-PC 系列机仍有一定市场这不仅是因为其价廉,而且还因一般的单色显示器有较高的分辨率,如 720×350 , 它和 640×200 的彩色显示器相比,显示出的字符和图形更清晰好看。在 CCDOS 下,在单显上能显示出 16×16 点阵的汉字和符号,但还不能在上面直接作图。《计算机应用研究》90 年第 4 期朱庆生同志的文章《不用图形卡在单显上作图的方法》,介绍了一种用汇编语言子程序作图的方法,将它驻留在内存中,作图时用软中断调用,或用 BASIC 语言调用机器语言的方法进行调用,这不失为一种好方法。但用 BASIC 语言或其它高级语言调机器语言时,实现起来不太简单。本文介绍一种用 C 语言和 BASIC 语言在单显上作图的方法。用 C 语言时,只调用用 C 语言写成的一个函数子程序,而这个函数子程序预先编成 obj 文件,或直接把它的加入库文件中,供 LINK 用。用 BASIC 语言时,只调用用 BASIC 语言写成的一个子程序,该子程序可预先编入一文件中(例如 dot.bas),在编写作图程序时,用 Merge 命令将子程序加进去。

二、显示缓存的对应关系

下面讨论一下分辨率为 720×350 的单显的情况。在 CCDOS 下,单显显示缓存的起始应为 B0000H,而彩显对应的起始则为 B8000H。但单显和彩显的显示缓存的区别还不止此。彩显在高分辨率(640×200)方式下,每幅画面占缓存 B8000H 到 BFFFFH,缓存每一字节的每位代表一个点,且各位的顺序与每行点的顺序一一对应,且有如下关系:

第一行: B800H: 0000H- B800H: 004FH,

第三行: B800H: 0050H- B800H: 009FH,

...

第二行: BA00H: 0000H- BA00H: 004FH,

第四行: BA00H: 0050H- BA00H: 009FH,

...

在单显的情况下,每幅画面占缓存 B0000H 到 B7FFFH,缓存每一字节的每位代表一个点,且各位的顺序与每行点的顺序一一对应。每行对应的缓存区为:

第一行: B000H: 0000H- B000H: 0059H,

第二行: B200H: 0000H- B200H: 0059H,

第三行: B400H: 0000H- B400H: 0059H,

第四行: B600H: 0000H- B600H: 0059H,

从第 5 行起,其中段址重复上述关系,偏移量增加 5AH,以后每隔 4 行再加 5AH。

三、画点子程序及作图例子

1、使用 C 语言

使用 C 语言的画点函数子程序的清单附录 1。使用格式为: dott (列, 行, 颜色), 其中: 列: 0~719, 行: 0~347, 颜色为 1 或 0。附录 2 示出了作一个矩形的程序清单。该程序已在 MSC4.0 版下调试通过。

2、使用 BASIC 语言

使用 BASIC 语言的画点子程序的清单见附录 3。在调用该子程序之前,应指出变量 yy (列) xx (行) 及 CLR (颜色) 的值,其中 yy: 0~719, xx: 0~347, CLR: 1 或 0。附录 4 示出了一个作圆的程序清单。注意: 应使用具有 Shell 命令的高版本 BASIC 解释程序或编译程序,其中 1000 行至 1030 行的画点子程序不用键盘输入,在编写作图程序前,用 merge "dot" 命令即可。

```
extern char far *scrn;
dott (yy, xx, clr)
int yy, xx, clr;
{
    int a;
    a = (int) (xx / 4) * 90 + (int)
        (yy / 8) + (xx % 4) * 0x2000;
    * (scrn + a) = (clr * 0x80)
        > > yy % 8 | (* (scrn + a));
}
```

附录 1

```
#include "stdio.h"
#include "conio.h"
char far *scrn = (char far *) 0xb0000000;
main()
{
    int i;
    system("cls");
    for (i = 200; i <= 250; i++) {
        dott (i, 200, 1);
        dott (i, 250, 1);
        dott (200, i, 1);
        dott (250, i, 1);
    }
    getch();
}
```

附录 2

```
1000 DEF SEG = &HB000
1005 A = (INT (XX / 4)) * 90 + INT (YY / 8) + (XX MOD 4)
    * &H2000
1020 POKE A, ((CLR * &H80) / 2 * (YY MOD 8)) OR PEEK
    (A)
1030 RETURN
```

附录 3

```
5 SHELL "cls": KEY OFF
```

(特 17 页)

一个用 Turbo C2.0 编写的文件删除实用程序

段心弓

大家知道, DOS 中的 DEL 命令只能删除指定路径下的文件, 对于该路径下的子目录也是无能为力的, 用户盘上往往建有多个子目录, 若要删除盘上的某一类文件 (如 BAK 文件), 需要逐个进入各子目录——进行删除, 这是很繁琐的工作, 特别是对于一些多级子目录。

这里笔者用 Turbo C2.0 编写了一个文件删除实用程序 XDEL.C, 该程序可搜索给定路径以下的所有子目录 (如有的话), 只要与指定条件匹配, 无论文件放在那级目录下, 都可统统列出并予以删除。该程序经编译产生可执行文件, 放在根目录下可作为 DOS 的一个外部命令使用, 是对 DEL 命令的一个有力补充。

用法示例:

XDEL C: *.*BAK

(删除 C: 盘中所有 BAK 文件, 包括其中各级子目录中的文件)

XDEL C:\ABC*.*

(删除 ABC 子目录下的所有文件, 包括其中各级子目录中的文件)

程序清单 (在长城系统微机上通过)

* 程序名: XDEL.C

* 功能: 删除给定路径以下所有与指定条件相匹配的文件

* 作者: 威海市四八〇九厂计算中心段心弓

```
#include "stdio.h"
```

```
#include "dos.h"
```

```
#include "dir.h"
```

```
unsigned char name[40];
```

```
long unsigned n = 0;
```

```
int done, d = 0;
```

```
struct fblk f[10];
```

```
main(int argc, char * argv[])
```

```
{
```

```
    unsigned char curdir[40];
```

```
    if (argc == 1){printf("\n 正 确 用 法 :XDEL\n\n");
```

```
    printf("\n 示 例 :XDEL *.*BAK(删除 BAK 文件\n\n");
```

```
    exit(0);
```

```
}
```

```
else subs(argv[1]);
```

```
printf("\n 共删除 %ld 个文件", n);
```

```
}
```

```
subs(filename)
```

```
char * filename;
```

```
{
```

```
    char subs1[50];
```

```
    int dir = 50;
```

```
    /* 删除当前目录下与指定条件匹配的文件 */
```

```
    done = findfirst(filename, &f[d], 0Xff);
```

```
    while(!done){
```

```
        if((f[d].ff_attrib & 0X10) == 0X10)
```

```
            { if(d == dir){dir = d;
```

```
            getcwd(subs1, 40);
```

```
            printf("\n\n%s\S\n", subs1);
```

```
            printf("该目录中的下列文件将被删除...\n");
```

```
        }
```

```
        n = n + 1
```

```
        printf("%s\n", f[d].ff_name);
```

```
        if(remove(f[d].ff_name) == 1)printf("REMOVE\n\n");
```

```
        ERR);
```

```
}
```

```
done = findnext(&f[d]);
```

```
}
```

```
/* 搜索当前目录以下的子目录 */
```

```
done = findfirst("*. ", &f[d], 0Xff);
```

```
while(!done){
```

```
    if((f[d].ff_attrib & 0X10) == 0X10 && f[d].ff_name[0] != '\0')
```

```
{
```

```
        getcwd(name, 40);
```

```
        if(name[strlen(name)-1] != '\\') strcat(name, "\\");
```

```
        strcat(name, f[d+1].ff_name);
```

```
        chdir(name);
```

```
        subs(filename);
```

```
        getcwd(name, 40);
```

```
        for(i = strlen(name)-1; i && name[i] != '\\'; i--);
```

```
        if(name[i-1] != ':') name[i] = 0X00;
```

```
        else name[i+1] = 0;
```

```
        chdir(name);
```

```
        -d;
```

```
}
```

```
done = findnext(&f[d]);
```

```
}
```

```
}
```

(程序文件名: ZF13.PRG)

EGA 卡图形方式下屏幕的滚动和横向移动

兰州军区司令部作战部 刘建平

一、屏幕滚动

EGA 卡的视频缓冲区有 256KB, 可以储存几个屏幕的数据。因此, 屏幕显示的内容可认为是缓冲区的一个“窗口”。改变“窗口”在缓冲区的位置, 既可改变屏幕的显示内容。

“窗口”位置的改变是通过改变视频缓冲区的显示“起始地址”实现的。视频缓冲区内起始地址处的字符显现在屏幕的左上角。起始地址增大 80, 整屏将上滚一线, 若减少 80 就会下滚一线。在 40 列模式下, 则应增、减 40。

在图形方式下 CRT 控制器 0CH 和 0DH 寄存器分别为“窗口”起始地址的高、低字节。下面这个程序把屏幕窗口定在一个指定的象素上, 在一个循环中反复调用这个程序就可实现“窗口”的移动。

```
NAME ScreenOrigin
ARGx EQU [bp+4]
__TEXT SEGMENT byte public 'CODE'
    ASSUME CS:__TEXT
    PUBLIC __ScreenOrigin
__ScreenOrigin PROC    near
    push    bp
    mov     bp,sp
    push    si
    push    di
    mov     bx,ARGx
    mov     dx,3d4h
    mov     al,12
    out     dx,al
    inc     dx
    mov     al,bh
    out     dx,al
    dec     dx
    mov     al,13
    out     dx,al
    inc     dx
    mov     al,bl
    out     dx,al
    pop     di
    pop     si
    mov     sp,bp
    pop     bp
    ret
__ScreenOrigin ENDP
__TEXT    ENDS
END
```

二、屏幕的横向移动

在正常情况下, 屏幕上显示的内容在视频缓冲区每一行的第一个字节紧接前一行的最后一个字节, 如果屏幕横向移动则会产生滚动现象。要避免这种现象, 可以把缓冲区每一行的显示宽度加宽, 使其每一行都比屏幕窗口宽。

CRT 控制器的 13H 寄存器控制了每一行显示的字

节, 加大该寄存器的值可使缓冲区内行的行加宽, 使每行仅有一部分显示出来, 如此再横向移动时, 就不会发生滚动现象。下面这个程序可以改变视频区显示行的大小。

```
NAME BufferDIMS
ARGx EQU word ptr [bp+4]
ARGy EQU word ptr [bp+6]
__TEXT SEGMENT byte public 'CODE'
    ASSUME CS:__TEXT
    PUBLIC __BufferDIMS
__BufferDIMS PROC    near
    push    bp
    mov     bp,sp
    mov     ax,ARGx
    mov     bh,al
    mov     ax,ARGy
    mov     ah,bh
    shr     ah,1
    mov     al,13h
    mov     dx,3d4h
    out     dx,ax
    pop     bp
    ret
__BufferDIMS ENDP
__TEXT    ENDS
END
```

三、举例

上面两个程序都是在图形方式下调用的, 在文本方式则要考虑每个字符的高度, 使滚动和横向移动过程中不会出现字符显示不全的现象。下面用 Microsoft C 调用。先将屏幕制为 16 色 640 * 350 方式, 让屏幕滚动后, 重新设制显示缓冲区行宽, 每行加宽一倍则每行象素数由 80 * 8 变为 160 * 8。实际硬件“窗口”则每行仍为 80 * 8 个象素, 这样就可横向移动“窗口”, 显示缓冲区另一部分内容。程序在东海 AT 机调试通过。

```
#define mode__code 16
#define CharColumn 160
#define CharRows 350
main()
{
    int i,j;
    mode(mode__code);
    for(i=0;i<CharColumn/10;i++)
        printf("0123456789");
    for(i=0;i<CharRows;i++){
        ScreenOrigin(i*CharColumn/2);
        for(j=0;j<10000;j++);
    }
    mode(mode__code);
    BufferDIMS(CharColumn,CharRows);
    for(i=0;i<CharColumn/10;i++)
        printf("0123456789");
}
```


解决在内存 640K 的 IBM-PC 机上 不能运行 COBOL 的两种方法

秦皇岛煤炭工业管理学校 邵榆麟

COBOL 语言具有极强的数据处理能力,已无可争辩地成为国际上应用最广泛的一种高级语言,但在 IBM-PC 机上却出现一种在内存容量 512K 上能运行的汉字 COBOL 源程序,反而不能在内存容量更大的 640K 上运行的反常现象。经过正确编译和连接后的源程序,运行时立即显示

*** COBOL: PROGRAM TOO BIG TO FIT IN MEMORY 的错误信息。其它高级语言如 BASIC, PASCAL, dBASEIII 等均无这种现象。我们曾试图对 COBOL 编译程序进行剖析,探其渊源,但要分析并进而修改从内存中翻出的机器代码,将是一件工作量相当大的长期任务,解决不了当前的急需。后来我们分别从硬件和软件两方面着手,开发成功了两种经过实践考证、简单可靠而又行之有效的办法,现向大家介绍如下:

(一) 硬技术

此法直接将 640K 内存改装为 512K,其具体做法是从内存扩展板的后部卸去容量为 128K 的一排片子,使内存容量变为 $640K - 128K = 512K$ 。所以要从扩展板的后部拆卸,是为了保持内存地址的连续性,而且不必重新设置系统板和扩展板上的 DIP 开关。若内存由系统板 128K 和两块 256K 的扩展板组成,则必须从第二块扩展板的后部拆卸。此法的优点是经过改装后的机器和原装 512K 的机器性能完全相同,可以直接运行 COBOL 源程序,其缺点是时而卸时而装,不利于机器保养,而且卸片子需要一定的技术,一旦操作失误,容易损坏电路和片子,所以采用这种方法一般比较慎重。

(二) 软技术

主要是通过开发系统软件功能,使之为解决实际工作中的问题服务。这里,我们利用 dBASEIII 系统能执行 DOS 命令、带扩展名 .BAT 批文件,以及其它任何 .EXE 或 .COM 文件的功能,先把源程序在 COBOL 系统下,完成编译和连接,产生 .EXE 文件,然后把它放在 dBASEIII 系统下去运行。因为,汉字 dBASEIII 要求最少要有 520K 以上内存空间,从而避开了汉字 COBOL 系统要求 512K 内存配置的障碍。运行程序前,源程序盘上还应同时有公共运行时间库 COBRUN.EXE 文件,因为运行用户程序时,IBM COBOL 运行时间系统,检查每一个目标语言指令,并执行所需的功能,这包括控制显示器、打印机及磁盘文件输入和输出所需要的所有操作。整个过程可以归结为下述步骤:

设已在临时磁盘上建立了一个 MYFILE.COB 的源程序(存于当前驱动器 B 中)。

(1) 将 COBOL 系统盘插入 A 驱动器中,键入编译 COBOL 源程序的命令:

A: COBOL

产生目标文件 MYFILE.OBJ。

(2) 将编译生成的目标程序,连接成可执行的目标程序用命令:

A: LINK

产生 MYFILE.EXE 文件。

用户可列出临时磁盘目录,以证实两个文件 MYFILE.OBJ 和 MYFILE.EXE 在里面。

(3) 将公共运行时间库 COBRUN.EXE 拷到临时磁盘上。

(4) 从 A 盘启动汉字 dBASEIII,并在圆点提示符下,键入命令:

.RUN B: MYFILE

运行可执行程序,得运行结果。

注: 1、对于带硬盘的 IBM-PC/XT 机,只需把上述步骤中的临时磁盘改为硬盘(当前驱动器改为 C),其它不变。

2、如将 COBRUN.EXE 拷到 dBASEIII 系统盘上,运行时要更方便些(请不妨一试)。为此,事前需腾出磁盘空间,使只保留 DBASE.EXE 和 DBASE.OVL 两个主要文件。 ■

```
for (i=0;i<CharColumn/2;i++){
    ScreenOrigin(i);
    for (j=0;j<10000;j++){
```

```
}
mode(mode-code)
int mode-code;
{
union REGS r;
r.h.al=mode-code;
r.h.ah=0;
int86(0x10,&r,&r);
}
```

(程序文件名: ZD22.PRG) ■

TP-180 单板机 256K EPROM 的扩展

湖南湘潭市电子研究所 周明望

一般的单板机可供用户使用的 ROM 往往只有几 K, 对于某些要求 ROM 很大的场合显然是不够的, 笔者在研制产品的过程中遇到了这一问题, 由于市场上没有合适的单板机可购, 因此, 根据这一特殊需要, 对新型单板机 TP-180 作了 256K EPROM 的扩展, 下面具体介绍扩展的方法。

一、地址空间的分配

TP-180 单板机 CPU 是采用日立公司的 HD64180, 与 Z80CPU 兼容, 另外增加了 12 条新的指令, 使其应用更为灵活。寻址能力可达 512K, 其内部有一个存储管理器 (MMU), 可把 64K 逻辑地址空间 (0000H~FFFFH) 转换成 512K 实际地址空间 (0000H~7FFFFH), 地址总线相应地由 16 根 ($A_0 \sim A_{15}$) 转换成 19 根 ($A_0 \sim A_{18}$)。图 1 为扩展前实际地址空间分配图, 可供扩展使用的地址空间只有 (20000H~40000H) 128K, 为了得到地址连续的 256K 地址空间, 将 RAM 区向上移 128K, 图 2 为扩展后的地址空间分配图。这样既得到了从 20000H~5FFFFH 256K 地址空间, 又不会影响 64K RAM 区的大小, 只是减少了 RAM 的重复区。在单板机上只要将 A_1 、 A_2 两三角处割开, A_1 串园处用焊锡短接, 并按图 3 所示加上一与门电路, 即可达到以上目的。

RAM(64K)	\$ 7FFFF
RAM(64K)	\$ 70000
RAM(64K)	\$ 60000
RAM(64K)	\$ 50000
RAM(64K)	\$ 40000
无占用	\$ 20000
	\$ 08000
EPROM1	\$ 06000
EPROM0	\$ 04000
EPROM1	\$ 02000
EPROM0	\$ 00000

图 1

RAM(64K)	\$ 7FFFF
RAM(64K)	\$ 70000
RAM(64K)	\$ 60000
EPROM(256K)	\$ 04000
	\$ 20000
	\$ 08000
EPROM1	\$ 06000
EPROM0	\$ 04000
EPROM1	\$ 02000
EPROM0	\$ 00000

图 2

容量为 16K, 共需 16 片 27128。数据与地址总线分割加上

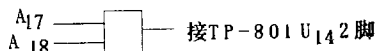


图 3

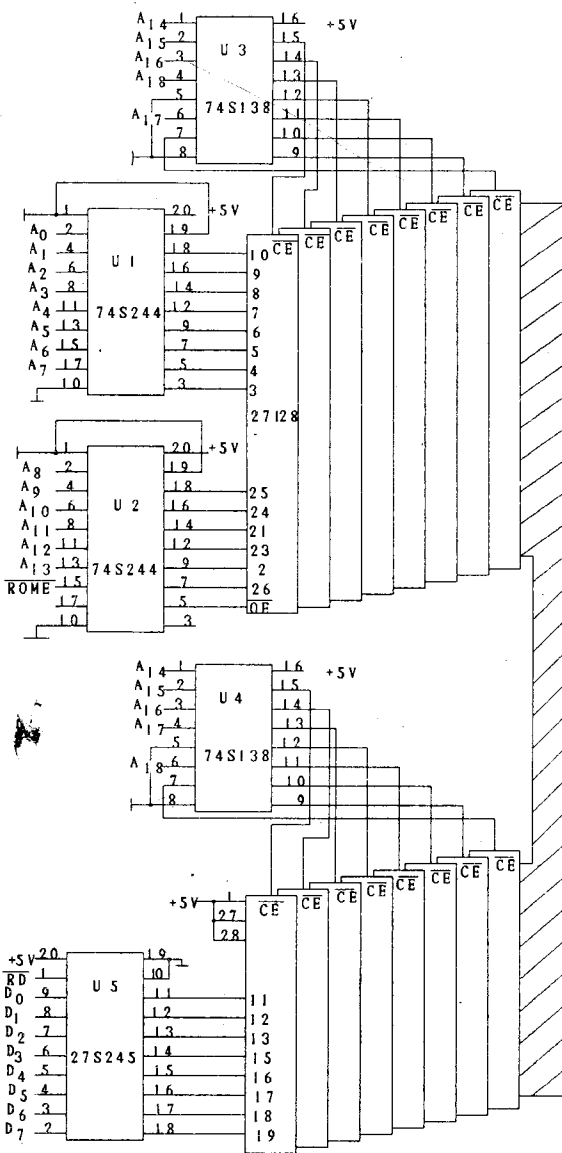


图 4

二、硬件原理

硬件线路如图 4 所示, 存储器芯片采用 27128, 每片

(转 22 页)

1000VA 在线式 UPS 性能比较表 1

型号	EXIDE POWERWARE 1000	山特 1KVA	电精 (DENSEI) 1KVA	山肯 (SANKEN) 1KVA	东芝 1KVA
类型	在线式 (微电脑控制)	在线式	在线式	在线式	在线式
输入电压	220V-18%+10%	220V±15%	220±15%	220V±10%	220V-15%+10%
输入频率	50Hz±5%	50Hz	50Hz±5%	50Hz/60Hz±5%	50Hz±5%
输出电压	220V±3%	220V±2%	220±2%	220V±2%	220V±3%
输出频率	50Hz±0.1%	50Hz±0.5%	50Hz±1%	50Hz/60Hz±0.1%	50Hz±1%
功率因数	0.65	0.8-1	0.7-0.8	0.7	0.85
失真度	<8% (最佳波形设计)	4%	5%	<5%	3%
工作方式	30KC 半桥式	20KC (MOSFET 方式)	1KC DARLINTON	14KC PWM MOS	10KC SDWM, DARLINTON
故障保护	(非常强)	有	有	有	有
旁路	自动旁路 <4ms	手动旁路	手动旁路	手动旁路	自动旁路
效率 (AC-AC)	90%	>75%	70%	有	>85%
报警方式	发光管, 声音, 微处理器	发光管, 声音	发光管, 声音	发光管, 声音	发光管, 声音, 微处理器
后备时间	7分钟	8分钟	12分钟	10分钟	10分钟
油机	可用	可用	可用	可用	可用
重量	17.2 千克	51 千克	85 千克	85 千克	85 千克
体积	34.2×13.3×31.4cm ³	44×21×50cm ³	80×100×35cm ³	80×100×35cm ³	80×100×35cm ³
通讯接口	可选 (RS232, LAN)	无	无	无	可选 (RS232)
平均无故障时间	5.5 万小时 (加自动旁路器 15 万小时)	无	无	无	无
噪音	45dB	55dB	60dB	60dB	50dB
注释	美国最先设计产品	日本产	日本产	日本产	日本产

1000VA 在线式 UPS 性能比较表 2

型号	SABNIFE 1KVA	KELONG 1KVA	DELTEC WATCHMAN II	SUPER 1KVA	UPSONIC 1KVA
类型	在线式	在线式	在线式	在线式	在线式
输入电压	220V-15%+10%	220-15%+10%	120V-25%+10%	220V±10%	220V±15%
输入频率	50Hz±5%	50Hz±5%	60Hz	50Hz±2%	50Hz±6%
输出电压	220V±2%	220V±3%	120V	220V±3%	220±2%
输出频率	50Hz±0.5%	50Hz	60Hz	50Hz±1%	50Hz±1%
功率因数	0.8	<3%	Crest factor > 3:1	0.8	0.6-1
失真度	<5%	有	高频, 三端口	3%	<5%
工作方式	20KC SPWMBJT	有	有	PWM	20KC PWM
故障保护	有	有	有	有	有
旁路	无	有	有	有	有
效率 (AC-AC)	>80%	有	有	有	有
报警方式	发光管, 声音	发光管, 声音	发光管, 声音	发光管, 声音	有
后备时间	20分钟	长时间	4分钟	6分钟	长时间
油机	可用	可用	可用	可用	可用
重量	43 千克	47×43×18.5cm ³	7×40×50cm ³	65 千克	56 千克
体积	26.5×65.5×30.0cm ³	无	可选 (RS232)	56.6×51.8×25.1cm ³	52.6×57×17cm ³
通讯接口	无	无	可选 (RS232)	无	无
平均无故障时间	5000 小时	无	可选 (RS232)	无	无
噪音	50dB	50dB	50dB	50dB	55dB
注释	瑞典产	上海产	超微型, 美国产	蛇口南方信息公司产	蛇口南方信息公司产

3kVA, 5kVA 在线式 UPS 性能比较表 1

型号	类型	输入电压	输入频率	输出电压	输出频率	功率因数	失真度	动态失真	工作方式	过载	故障保护	旁路	效率	报警方式	后备时间	油机	噪音	重量	体积	通讯接口	平均无故障时间	注释:
EXIDE PLUS5	3.3 / 5.5kVA 在线式	220V-20%+25%	45~65Hz	220V±2%	50 / 60Hz±0.1Hz	0.8 或 3:1 CREST FACTOR	<3% (对线性或非线性负载)	<5%rms	HIGH-F, LINK 20KC IGBT	300% 10CYCLE	非常强	有	80%	发光管, 声音, 微处理器	7分钟	可用	50dB	123 千克	39.4 × 81.3 × 86.4cm ³	可选 (RS232, LAN)	55,000 小时	3.3kVA UPS 可现场升级 5.5kVA)
电精 (DENSEI)	3kVA 在线式	220V±15%	50Hz±5%	220V±2%	50Hz±1%	0.7~0.8	5%	不好	IKC DARLINTON	110% 30 分钟	有	有	75%	发光管, 声音	15 分钟	可用	60dB	230 千克	60 × 45 × 70cm ³	无	日本	
山特 (SANTAK)	3 / 5kVA 在线式	220V±15%	50Hz (最大电流 22.5A)	220V±2%	50Hz±0.5%	1~0.8LAG	3% (只对线性负载)	4%	20KC PWM	110% 旁路	有	有	80%	发光管, 声音	8 分钟	可用	55dB	120 / 140 千克	68 × 26 × 63cm ³	无	台湾产	
UPSONIC3.5	3 / 5kVA 在线式	220V±15%	50Hz±3%	220V±2%	50Hz±1%	0.6~1	<5%		20KC PWM	125% 20 秒	有	有	90%	发光管, 声音	长时间		55dB	130 / 200 千克	55 × 64 × 34cm ³	无		
熊猫 BXY3-87	3kVA 在线式	220V-20%+13%	50Hz	220V±2%	50Hz±1%	0.8	3%	8% 100 毫秒	105% 10 分钟				90%	发光管, 声音	10 分钟			160 千克	48 × 71 × 97cm ³	无	南京 714 厂出产	
东芝 UTINB503	3kVA 在线式	220V-15%+10%	50Hz±5%	220V±3%	50Hz±1%	0.85	3%	10%	HIGH-F	150% 60 秒	有	有	>85%	发光管, 声音	10 分钟		50dB	119.75 千克	29.2 × 69.85 × 69.85cm ³	RS232	日本产	

3kVA, 5kVA 在线式 UPS 性能比较表 2

梅兰日兰 (MERLINGERIN)	山青 (SANKEN FBK-SS302)	DELETEC 2000	DELETEC7000
3kVA 在线式	3kVA 在线式	3kVA 在线式	5kVA 在线式
220V±10%	220V±10%	220~15%+10%	220V-20%+15%
50Hz±5%	50Hz±5%	50Hz/60Hz	50Hz/60Hz±3Hz
220V±2%	220V±2%	220V±2%	50Hz/60Hz±1.5Hz
50Hz	50Hz/60Hz	50Hz/60Hz	0.8
0.8-1LAG	0.8	3%	<10%
5%	5%	3%	5%
5% <50ms	三端口 HIGH-F	HIGH-F, PWM	HIGH-F, PWM
HIGH-F IGBT	150% 30 秒	700% 1CYCLE	700% 1CYCLE
120% 1 分钟	有	有	有
有	有	有	有
77%	85%	80%	80%
发光管, 声音	发光管, 声音	发光管, 声音, 微处理器	发光管, 声音, 微处理器
10~30 分钟	10 分钟	10~60 分钟	10~60 分钟
可用	可用	不可用	可用
52dB	<50dB	<55dB	<55dB
325 千克	130 千克	132 千克	225 千克
66×112×70cm ³	27×60×85cm ³	60.1×29.2×71.1cm ³	114.3×60.1×81.3cm ³
	无	可选 (RS232)	可选 (RS232)
法国产	日本产	美国产	美国产

型号	类型	输入电压	输入频率	输出电压	输出频率	功率因数	失真度	动态失真	工作方式	过载	故障保护	旁路	效率	报警方式	后备时间	油机	噪音	重量	体积	通讯接口	平均无故障时间	注释
梅兰日兰 (MERLINGERIN)	3kVA 在线式	220V±10%	50Hz / 60Hz±5%	220V±2%	50Hz / 60Hz±0.1%	0.8	3%	5%	15KCBJT	120% 10 分钟	有	有	77%	发光管, 声音	10~30 分钟	可用	52dB	325 千克	66 × 112 × 70cm ³		法国产	

第三代 4.1 版超想《自然码》

“自然码”是由北京超想电脑公司软件设计专家周志农研制的智能化全功能第三代汉字输入软件。由于“自然码”以 CCDOS 拼音为基础，面向普及，易学高效，特别适合于普通办公人员、作家、记者等等使用。在数百种输入法中“自然码”后来者居上，目前已成为普及型输入法中影响最大，用户最多的输入法之一。现在，全国各大机关部委、军委、报社以及东南亚等上百万用户均已普遍采用，并受到越来越多用户的称赞。

“自然码”突破了中文文书处理上文字录入困难的瓶颈，以词为主导、声形义结合、智能相关处理，融合多种输入方法为一体，让用户不必面对检索困难、茫然懊丧之苦，因此初学者容易入手，熟练之后可以不看屏幕盲打，使文书处理工作轻松写意，快速有效，充分体现了第三代全功能输入系统的特点。

最新 4.1 版超想自然码具备前所未见的先进智能处理库和全功能输入特色，它不像老式输入法那样，只能一个字一个字按编码输入，然后费时费劲地在屏幕提示行上翻页、查找、选字。“自然码”首先开创了多种输入功能浑然一体的“全功能”快速输入法，让用户能像使用英文打字机一样，对着键盘，挥洒自如、弹指如飞进行文稿输入工作。

“自然码”的全功能特点，可以概括为以下 7 点：

▲普及——它价格低廉，操作极其简单，从小学生到专业打字员适用。

▲易学——懂得汉语拼音的人，特别是会 CCDOS 拼音的用户，只要稍加指点立刻成为一级输入高手，练习一星期后，便可以达到每分钟输入 100 字的水平。充分利用自定义词组，每分钟输入可超过 200 个汉字。

▲易用——任何时候，只要有一台 PC 机，无论是否拥有硬盘，不论是彩显和单显，只要在加载了中文系统后，就可以立刻安装“自然码”。安装自码的过程十分容易，无需专人指导，只须按照安装说明书指示，即学即通。

▲快速——不必判码、不必拆解字根、不必死记编码，自然码充分发挥意到手随，心手合一、自然流畅的最高书写境界。单字输入加偏旁部首的发音声母不用翻页；词组输入遇有重码，在智能处理下，自动调整词序，自动记忆，用户仍然可以不看屏幕，快速盲打。二字词以声韵声韵四码输入，三字以上的词组只输入声母，方便快捷，超越联想。

▲友好——“自然码”提供基本词组一万五千条，而且允许用户大量加词。用户可在任何软件中随时加词，系统自动记忆，以后永远可用。因而在应用之后，词库日渐丰富，输入速度逐日倍增。“自然码”以词组为主要输入方式，不会输入错字，不必校对。作家直接用来写稿，不必思考字形，写作效率极高。

▲独特——“自然码”是目前所有中文输入系统中，唯一能够处理半角字符和立体词组编码的系统。在日常信函处理中，可以预先将用户名称、公司地址、电话及落款等都编为立体词组，即使多至数十行，使用时仅须一键输入。

▲全能——“自然码”除高频先见的拼音输入、智能相关处理、简码字、简码词以外，还提供南方音、画表、中文标点、中文数字、联想、词中选字、模糊输入、重输入、指定字、指定音节表、磁盘保密等等许多特殊功能，是用户足以信赖的工作伙伴。

新版自然码除保持原系统的“高效、易学、快速、自动加词、立体词组”等优秀功能以外，更增添了“磁盘文件保密、随时记忆字词搭配关系、自动调整字词优先次序、(北大金山、四通 2401 选择)”等新功能。

4.1 版附保证书，今后可凭保证书享受各种优惠服务。