

# 电脑

3  
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  
安全可靠  
功能特殊  
電腦必備

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84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。86年获国家科技进步奖。三相抗干扰稳压电源也由中国科学院广州分院通过技术鉴定。88年获产品专利权。89年获中国科学院技术进步三等奖。

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
容量	350VA	500VA	1KVA	2.2KVA	3.2KVA	5.2KVA	10KVA	15KVA

#### 二、主要技术参数

●输入电压单相交流220V50Hz

●电压稳定度

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

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扼制能力符合国际计算机电源要求

●应变时间（输入电压跳变 $\pm 100$ 伏） $< 10 \sim 30\text{ms}$

●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：3KV, 6KV, 10KV, 15KV, 30KV

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

●稳压度调整率：输入 $-40\% \sim +40\%$ ，输出 $-4\% \sim +1\%$

●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1/5。这对节能很有意义。

本单相、三相电源已与VAX11/750, VAX11/785, 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 副厂长、工程师：陈自如 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 账号：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

高抗干擾  
安全可靠  
功能特殊  
電腦必備

# 电脑杂志社

## 科技开发经营部

感谢各厂家及公司的支持与合作!

现代理如下产品:

1. Super 系列 286、386、486 微机、Super PUC II 型汉卡
2. 浪潮系列 LC0530 (286)、LC0540 (386) 微机
3. 双星 SS-0500 汉卡
4. 双星 SS-CEGA 汉卡
5. 双星 SS-CMGA 汉卡 (单显)
6. 双星电子留言钟
7. 双星 CAD、办公自动化系统
8. 陕西计算机厂中华学习机
9. 激光排版轻印刷系统
10. 小林通电子游戏机系列

欢迎来人来函联系订购、欢迎各厂家及公司提供产品。

地址: 广州市德政北路 393 号 邮码: 510055

联系人: 萨成熙 电话: 330644

## 华南师范大学微电子学研究所

### 华杰技术开发公司

经营: 计算机及其外围设备, 承接计算机网络工程, 经营的产品有: compaq 386 系列、AST 286 系列微机。以及 PC/XT、286、386 兼容微机及其配件。目前本公司产品有: 64 门/1624 门防火防盗微机管理系统, 电视高山发射台无人值守遥控机。另外, 公司下属三瑞电子厂承接电子产品加工业务, 目前加工的电子产品有: 电子笔记本、语言翻译机等。

地址: 广州天河五山路科技街 134 号

电话: 510234 邮码: 510530

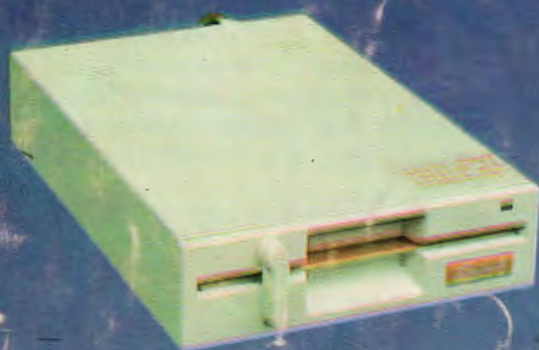
联系人: 麦哲明



# 广州超级电脑配件有限公司

GUANGZHOU SUPER COMPUTER PERIPHERAL CORPORATION

超级全系列133.4 mm (5 1/4 英寸) 半高软盘驱动器



中外合营广州超级电脑配件有限公司为广州市计算机公司属下生产企业。生产的超级牌全系列133.4mm (5 1/4 英寸) 半高软盘驱动器已推出市场, 该公司拥有现代化的生产设备和先进的测试设备, 所生产的系列产品机械结构紧凑合理, 实用可靠, 使用灵活; 控制电路全部采用专用IC, 集成度高因而器件量大大减少, 控制电路的可靠性大大提高, 产品的性能优越, 价格低廉, 产品除供出口外销, 还将供应国内市场。产品系列包括有SP-FD102, SP-FD128, SP-FD228三种规格, 其中FD102可应用于Apple II、II E、II C、Apple兼容微机及其中华教学微机系统中; FD128可应用于IBM PC-XT及其兼容微机系统中; FD228可应用于IBM PC-AT及其兼容微机系统中。该系列产品的MTBF均大于10000小时, 广州超级电脑配件有限公司执行质量第一, 用户至上的宗旨, 对售出的产品实行保修壹年。

## 东北三省总代理

龙江电子集团公司佳木斯计算机厂

群星电脑商场

地址: 佳木斯市保卫路445号

电话: 22183 联系人: 王福

## 北京总代理

煤炭科学研究总院北京建井所

地址: 北京市和平里 邮码: 100013

电话: 4215281 联系人: 侯胜溪

## 广州总代理

电脑杂志社

地址: 广州市德政北路393号

邮码: 510055

电话: 330644 联系人: 萨成熙

# 广州市科教电脑商场微机配置及价格一览表

配置	CT80286-16				说明
CPU	80286-16	80286-16	80286-16	80286-16	可插 80287
频率	10 / 22MHz	10 / 22MHz	10 / 22MHz	10 / 22MHz	按钮加速、液晶显示
内存	1MB	1MB	1MB	1MB	可扩至 4MB
软驱	1.2M+360K	1.2M+360K	1.2M+360K	1.2M+360K	可选 1.44M3.5"盘
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 DBASEIII
价格	7800 元	10300 元	11000 元	12500 元	赠送机械式鼠标器
配置	CT386DX / 33-64K / 128K Cache RAM				
CPU	80386DX-33	80386DX-33	80386DX-33	80386DX-33	可插 80387
频率	33 / 58MHz	33 / 58MHz	33 / 58MHz	33 / 58MHz	按钮加速、液晶显示
内存	2MB	2MB	4MB	4MB	可扩至 32MB
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	40MB	40MB	100MB	200MB	3.5"高速硬盘
显示	EGA640×350	TVGA1024×768	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	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	DOS3.31 DBASEIII
价格	17000 元	17800 元	21000 元	24000 元	赠送光电式鼠标器
打印机	LQ-1600K	4400 元	AR3240	5800 元	日本原装
UPS	500W	1500 元	1000W (在线式)	6800 元	美国或日本
主板	80286-16 (台湾)	1250 元	80386DX-33 (台湾)	7000 元	386 板为 4MB RAM
硬盘	3.5" 高速 40MB 带卡	1860 元	3.5" 高速 100 MB 带卡	3500 元	美国 Conner
软驱	1.2MB	450 元	360KB	390 元	1.44MB 430 元
彩显	14" EGA (5154H)	3000 元	14" TVGA (5156H)	3200 元	进口原装 CASPER

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

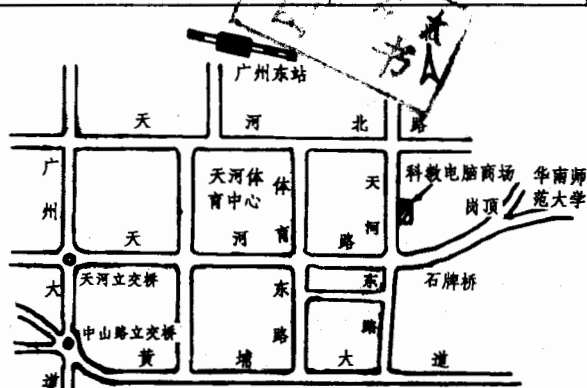
电话: 511197、515564

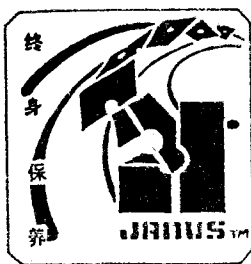
电挂: 8360 FAX: 511197

开户银行: 广州中国银行天河支行

帐号: 271-015110027

免收运费 保用壹年





# JANUS

“永远使顾客满意”  
“奉献给对工作严肃认真的人”

“JANUS”电脑软磁盘以其“优质片基、严格检测、设计曲雅、记录可靠，服务周到、恪守信誉、终身保养”多次获美国“Commercial Business”评比第一，并于1989年6月在中国市场进口软磁盘抽样评比中名列榜首，获得“不愧为名牌”的称号。

“JANUS”在软磁盘中全球首创“终身保养”，您所使用的“JANUS”，无论出现任何问题，都可无偿获得调换。今天，市场上“廉价货”必然会导致您将来重大的损失。“产品的可靠质量和优质服务”这是我们彼此成功的秘诀，请您在选择软磁盘时慎重而正确地做出决定。

“JANUS”电脑软磁盘，自1990年10月18日起，10片装的包装盒上，皆有激光立体反光标记，无此标记，皆为假货。

地址：广州市环市东路403号

电话：777905-150 765904

联系人：杨帆

“JANUS”远东总代理香港辉兄弟进出口有限公司

“JANUS”中国总代理广州杰纳斯电子有限公司

# 晨光智能仪器股份有限公司

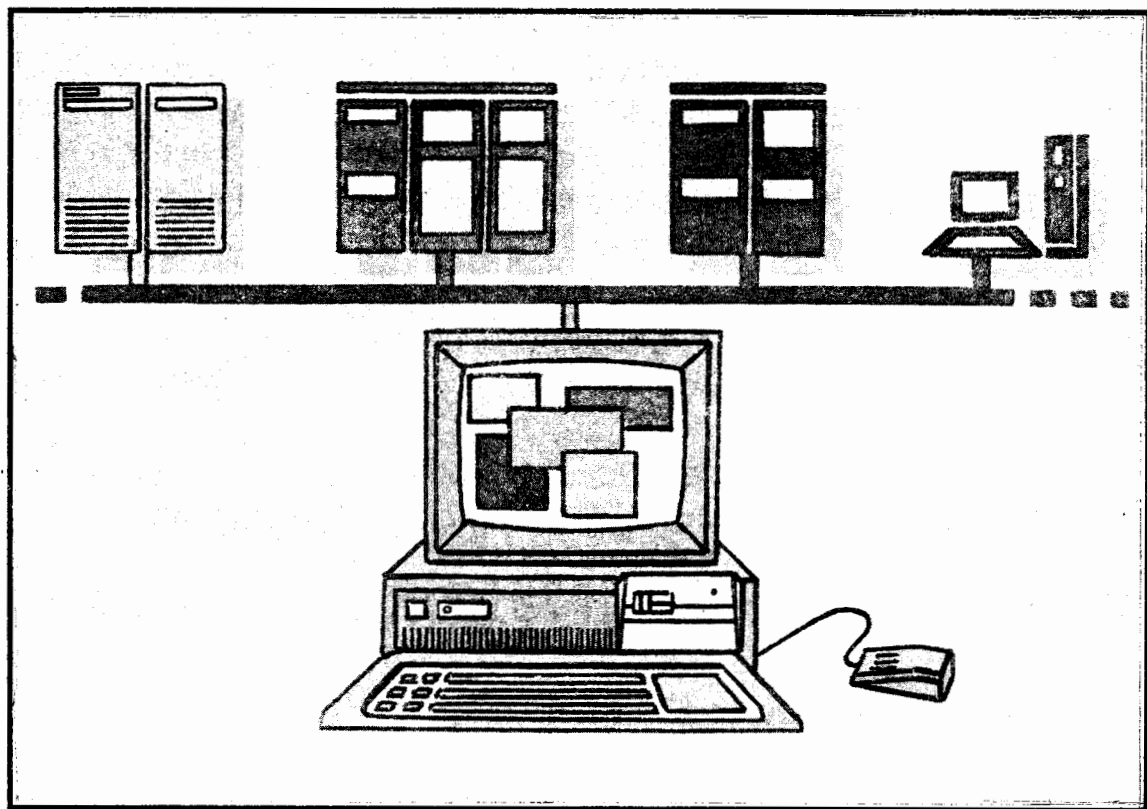


SUNLY

INTELLECTUAL

主要经营: AST、COMPAQ、CSL 等系列微机、兼容机及计算机配件

承接: Novell 网络工程、CAD 工程及自动化工程



联系地址:

广州: 广州市天河科技街 149、153 号

Tel: 510191、510299

传真: 510299

邮政编码: 510630

深圳: 深圳市蔡屋围新九坊 11 栋 3 楼

Tel: 271214、271132

传真: (0755) 271132

邮政编码: 518001



高能宗旨：诚恳、信誉、优质服务

广州经济技术开发区  
高能计算机公司

简介

总经理：廖仲琪 常年法律顾问：全朝晖律师

高能计算机公司是广州大型计算机专业公司之一，在国务院批准之广州高科技产业开发区注册并被评为国家级科技企业。公司拥有独立法人资格，经济上独立核算，法人代表：廖仲琪（高级工程师）。公司去年营业额近一千万元。

高能计算机公司是实业型企业，致力实业投资和为用户进行计算机管理工程、计算机与仪表控制工程、机房工程、机房配套工程及通讯工程之开发与实施。公司拥有较雄厚的经济实力，一支优秀的工程技术队伍和一支有经验的施工队伍。公司是中外合资高能抗静电地板厂，南海明珠化工厂的创办者和投资者并曾为广州羊城晚报社、广州电池厂、广州城建学院、上海冶金工业局、深圳财政局、佛山化纤总公司、江门城区物资公司、茂名石油公司等大中型企业完成各类大小工程 40 多项，得到上述单位的好评。

高能计算机公司拥有自己的生产基地和遍布全国各地的销售网络，销售世界名厂各类计算机主机，外围设备、机房设备、附配件、电子仪器、环保仪器和自己生产的产品。

高能计算机公司是美国山特公司授权之 SANTAK 牌不间断电源（UPS）系列产品之中国特约经销和特约总维修站，公司同时是名牌 GOLD LION 计算机系列配件之中国总经销和高能牌系列抗静电地板国内总经销。

高能计算机公司欢迎各种形式的技术合作，工程委托，贸易洽谈和技术咨询。

公司总部：广州花园酒店商业大厦 601、611 室

电话：355774、338999-601、611

门市部：广州五山路科技街 70 号

电话：510270

设计室：广州建设六马路青菜岗 1 号大院

生产基地：广东东莞市南城路供销大厦



## 广州达奇电脑科技有限公司 简 介

广州达奇电脑科技有限公司是广东省科委领导下的高技术企业，是实行科工贸一体化的新技术产业机构。公司以计算机及外部设备、网络与通信、自动控制、软件开发为主，进行以技术开发和技术服务为基础的研制、转让、销售等经营活动，为振兴国家信息技术产业，特别是广东省及邻近各省的电子计算机应用贡献自己的力量。

公司的创办者张学平总经理是海内外知名的数据库和网络专家。他的一项独到的科研成果——“dBASE 分析”从 1983 年起就为计算机界广大同行所称道，为在我国开展大型软件的解剖分析闯开了一条路子。他所分析与汉化的网络多用户 dBASE III PLUS FOXBASE 软件包，从 1986 年起通过多种渠道被全国各地的成千上万个单位所使用。我省一些典型的计算机网络系统，如万宝集团公司信息中心、深圳卷烟二厂、广州市供电局、交通银行广州分行、广州啤酒厂、广东省石油气具发展公司（即“万家乐”）、佛山市塑料一厂、广东省税务局、广州华侨房产开发公司等，就是他一手组织完成的。公司还拥有一支技术素质和服务素质均属上乘的设计、开发、维修队伍，并与广州地区各高等院校和研究所保持着由来已久的技术协作关系。

达奇公司是世界上最大的计算机公司——美国 IBM 公司在中国的指定代理商，以优惠价格提供 PC 机的升级产品——PS/2 系列微电脑。同时，公司还经销美国 COMPAQ 系列微电脑、ACER 系列微电脑和各种档次的兼容机。公司常备各种打印机，供用户选用。计算机网络工程是我公司的擅长技术，从系统分析、工程预算、布局设计、硬件选型、软件汉化到设备供货、安装调试以及应用软件开发，可提供一揽子技术服务并将自己的经验和实力落实到工程的每个环节。

公司在使用和比较国内外几个流行的电子 CAD 软件的基础上，自行研制了 ECAD 软件包。全套软件用 C 语言编程，计约 20000 条 C 语句。ECAD 在图素输入的方便性、布局和布线的合理性及布局和布线速度、布通率等方面，都优于国内外几个流行的电子 CAD 软件。ECAD 可适应 CGA、EGA、VGA 等多种彩色显示，而且用九针打印机都可打印出供照相制版的工艺图和电原理图。

公司下设“电脑迷俱乐部”（Hacker club），以期与对电脑痴心入迷者、对电脑软硬件研究独到者、对电脑发展想入非非者、对获取电脑知识兴趣盎然者建立友谊和保持联系。俱乐部陈列并销售香港陈鼎铭先生创办与发行的各期《现代电子》、《微型电脑》、《PC 电脑专刊》、《电子技术》、《工业自动化》、《现代电脑》等杂志及其它电脑书籍与资料，以供阅读、讨论和技术交流。

达奇电脑公司将充分发挥自己经营优势、人才优势和技术优势，为机关、工厂、学校、酒店、银行等企事业单位的计算机应用尽力。我们要与计算机界同行及广大用户一道，买电脑、用电脑、学电脑、玩电脑，携手到达奇妙无穷的电脑世界！

公司地址：广州花园酒店东楼 802、808 室

邮政编码：510064

电话：338999-802, 808

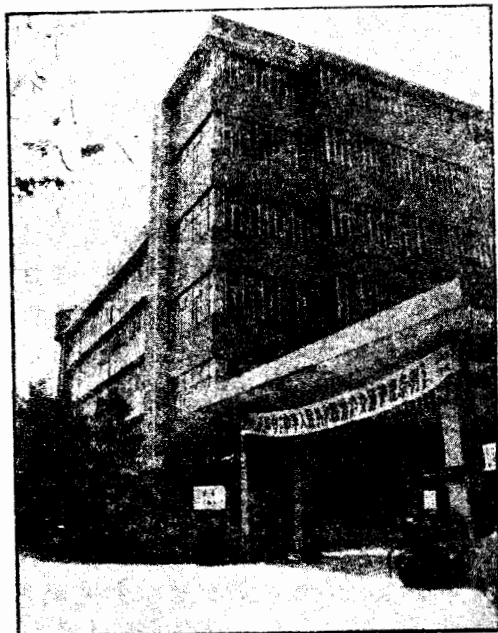
传真：752650



# 华南计算机公司小型机厂

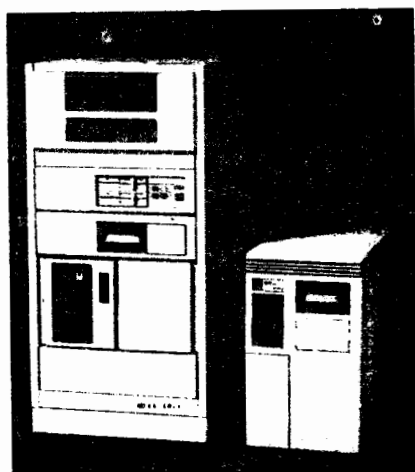
本厂是专门从事 2000 系列计算机 (PDP11、VAX 系列) 生产经营和科研开发的专业厂、拥有现场服务、产品维修、技术培训、开发及生产调试的专职技术队伍与 DEC 富士通 CDC 等国际著名计算机公司建立良好的合作关系, 享有相当高的代理级别。本厂已向国内广大用户提供 2000 系列各档机 600 多套, 遍布全国 28 个省、市、自治区。

本厂以“信誉第一、用户至上”为宗旨, 以“灵活的经营、有效的管理、精湛的技术、优质的服务”的方针、热忱为国内外用户服务, 赢得用户的一至好评。



## 主要经营产品

- HN-2220 • HN-2230(与 microVAX II VAX3500 全兼容)
- HN-2234 • HN-2238(与 microVAX3400 VAX3800 全兼容)
- 承接系统应用开发, 网络工程并协助引进 VAX6000、VAX4000、VAX3000 等
- VAX 机系统的升档, 内外存扩充
- 以优惠价格代理富士通、CDC 温盘、Emulex 各种接口等外部设备
- 承接 2000 系列各档机设备维修业务
- 凡我厂售出之产品均免费现场安装包修一年、服务终身。



HN2220 小型机

厂址: 广州市沙河沙太公路京溪桥

电话: 705696-266、279

邮编: 510510

联系人: 伍云成 李天元

销售服务部:

广州华南电脑大厦 4 楼

北京友谊宾馆苏园 60642 室

上海大陆饭店 720 房

成都珠峰宾馆 1219 房

西安电子科技大学 6 楼 101 房

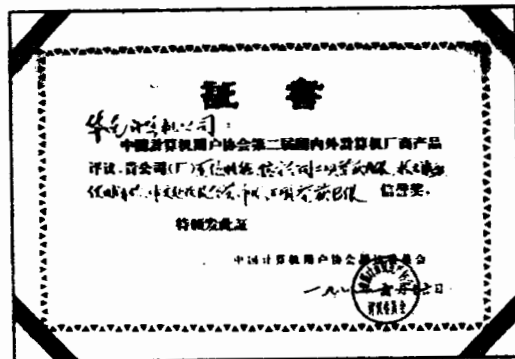
Tel: 351237

Tel: 8498288 8012121

Tel: 3270202-722

Tel: 662441-1219

Tel: 711020-2714、3110



评比奖状

# 广州市计算机公司天河技术服务部

广州市计算机公司天河技术服务部开业于一九八九年。是广州市计算机公司设在天河科技街的一个下属部门，侧重点在计算机及其配件的销售服务和技术服务方面。

广州市计算机公司天河技术服务部有坚实的后盾支持它的母体—广州市计算机公司是广州市电子工业系统的计算机专业公司。一九八五年七月经国家机械电子工业部计算机局批准为中国计算机软件与技术服务总公司广州公司。一九八八年经广东省加强企业管理领导小组批准为“省级先进企业”，公司有一支强大的

工程技术队伍，工程技术人员占职工总数 60% 以上。设计算机技术研究、质量、计量和动力设备等技術部门从事计算机系统、工业控制机、应用技术、计算机网络工程、3 英寸、5 英寸软驱动器和软件等开发工作。



公司引进了国外通用微机生产线的全套设备和技术，包括零部件 QC、元器件整形、插件、波峰焊、清洗、调试、老化、总装、系统调试等工序及 QC 的各种设备、仪器、仪表共 54 种 87 台套。月单班生产能力微机系统 1000 台，各种板 5000 件，已批量生产 8/16 位微机系统及各种功能板。一九八六年经机械电子部审查及颁发了电子计算机生产许可证。

## 主要产品：

国华-II 经国务院电子振兴办审定为全国计算机优选机型，荣获广东省优秀新产品一等奖，并被评为省优质产品，长城-0520CH；国华-286 高级中英文多用户微机系统，均受政府机关、科研、教育和工矿企业单位的欢迎。另外，公司与外商合资建立的中外合资企业：超级电脑配件有限公司，所生产的软磁盘驱动器大部分销往国际市场。

一九八九年开辦，我们作为 IBM 指定代理商 (PS/2 PC) 经销 IBM PS/2 系列微机。

广州市计算机公司天河技术服务部热诚欢迎国内外各界朋友前来洽谈贸易和进行各方面的经济、技术合作。

地址：广州市五山路科技街 62 号

电话：510476

联系人：高东辉 冯树岸

邮政编码：510630

# 广东外贸电脑公司天河产业区经营部

本公司从事电脑经营销售业务，软、硬件产品，办公室自动化用品及零配件，电脑易损耗品，并承接各种电脑、外设维修业务，本经营部一九八九年被评天河科技街最佳经营企业。

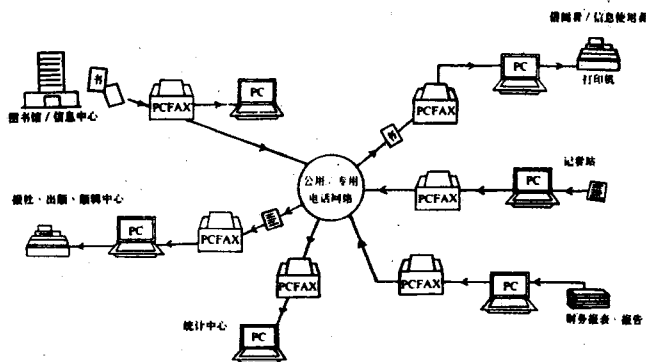
本公司受香港梅兰日兰电子有限公司委托接受法国梅兰日兰(Merlin Gerin) UPS 系列产品的咨询订货、购销业务，欲进一步了解有关 UPS 系列产品资料，有简章备索。

本公司是多功能电脑传真机 UA2110-PCFAX 产品的华南地区总经销。

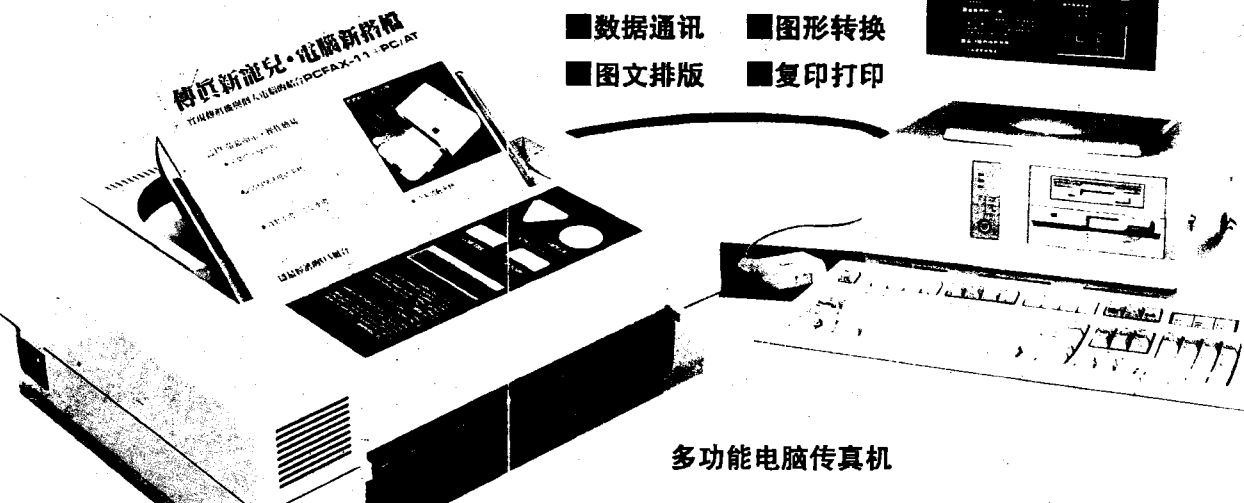
本公司是名牌打印纸一无锡“双龙”牌的广州科技街总经销。

本公司是 JANUS (杰纳斯) 磁盘、清洗盘及记录产品的一级代理经销商。

欢迎各地用户来人来函批发，零售订货，我们将以用户第一为宗旨，竭诚为用户服务。



- 智能传真
- 自动收发
- 图文扫描
- 图文编辑
- 数据通讯
- 图形转换
- 图文排版
- 复印打印



多功能电脑传真机

经营部地址：广州市天河五山路科技街 79 号、84 号、88 号

邮政编码：510630

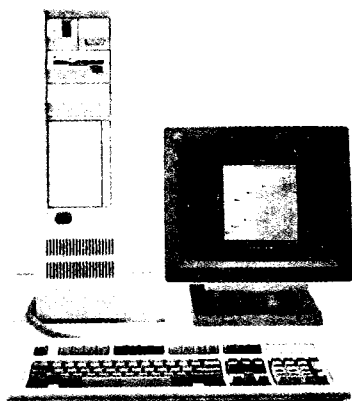
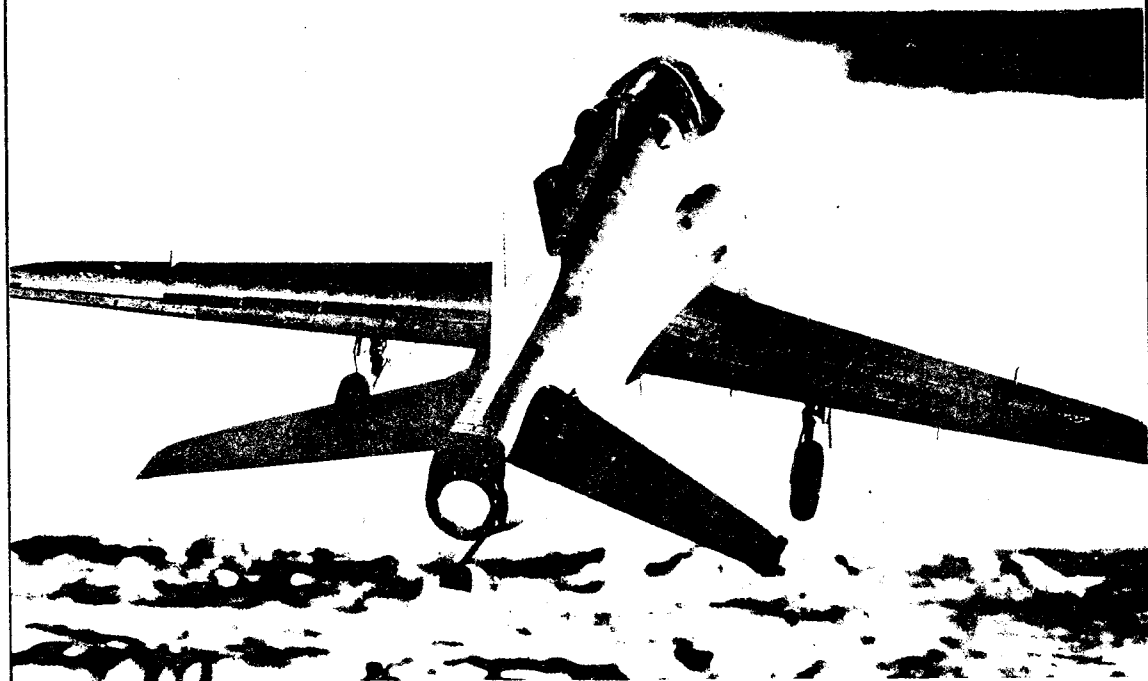
电 话：510219、510217

经 理：温国雄 胡彬武



# 新华电脑中心

## 引导您迈入信息时代



主营：微电脑、打印机、电源、盘片等。  
兼营：科教仪器

地址：广州市环市西路 202 号  
(民航售票处对面)

邮编：510010  
电话：667726、677690  
电挂：3700

# ZGX 系列全自动供电系统销售服务中心

- 一. 本系统适用于用电设备不能停电且需要连续长时间工作的场合, 如金融、统计、部队、通讯、邮电、医院、设计院、监测中心、调度中心、计算中心、外贸、公安、警卫机要、税务、海关、图书馆、地质及流动性大的系统和单位。现短时间停电可由 UPS 解决, 但长时间的停电或电感性负载 (空调等), UPS 就无能为力了, 而本系统就能很好地为各种用电设备, 如空调机、电风扇, 计算机等提供连续长时间电力。本产品已通过鉴定并列入火炬计划。现已在上述单位和系统投入运行。
- 二. 当电网停电(或电压过高、过低)时能自动启动发电机并自动转换到发电机供电; 当电网恢复供电(或电网电压恢复正常)时又能自动使发电机停机并自动转换到电网供电。本系统全自动运行, 不需人工干预, 远距离控制避免发电机噪音。
- 三. 本系统有自检功能, 对全自动供电系统的运行情况设有状态检测、故障监视和报警装置, 保证了供电系统的可靠性。
- 四. 技术参数:

供电系统输出电压:  $200 \pm 10\%V$

输出功率: 1000W 至 5000W, 超过 5000W 特殊定货

输出频率:  $50 \pm 2\text{HZ}$

控制柜对发电机控制距离:  $< 100\text{m}$ , 超过 100m 特殊定货

使用环境温度:  $-20^{\circ}\text{C} \sim +50^{\circ}\text{C}$

自动启动发电机时间: 3~60 秒可调

系统转换时间: I 型  $< 10\text{S}$ , II 型由 UPS 决定。

## 广州天河计算机公司

地址: 广州市中山四路 36 号

邮政编码: 510055

电话: 336875、351417

FAX: 351417

电挂: 3355

## 昆明电脑公司研究所

地址: 昆明市长春路 64 路

邮政编码: 650011

电话: 21983、34186

FAX: 37712

电挂: 5011

## 广州天河计算机公司专业经营

各种名牌优质电脑, 各种功能齐全的打印机, 各种规格型号的不间断电源, 各种电脑外围设备和消耗品、压感打印纸、双龙牌打印纸。

承接:

财务系统软件, 税务系统软件, 网络工程, 教育网络工程, 硬件开发, 工业过程控制, 办公室自动化工程及各种软硬件系统开发。

# 广州袖珍计算机技术服务中心

## 简介

广州袖珍计算机技术服务中心，位于广州东风东路 745 号，是一支专门从事袖珍计算机销售、技术开发、技术咨询、人员培训、翻译出版、维修综合服务的单位。多年来，在社会同仁的大力支持下，做了一些有益的工作。

### 一、抓住时机，为基层普及和提高计算机知识水平

开放的浪潮，给中华民族迎来了高科技电脑产品。1983 年袖珍计算机涌进中国市场后，中心率先在全国内办起了袖珍机 BASIC 语言应用培训班。随着培训班名声的逐渐扩大，全国二十多个省市派人来学习，并受外地的邀请，先后到了福建、海南、韶关等地办班，1986 年 CPPX 协会提出袖珍机二次开发的号召，中心开办起袖珍机器语言和接口技术班，为培训高一级袖珍机人材作了贡献。培训工作，从 1983 年至今，已举办了各种类型学习班 22 期，参加人数一千余人。

### 二、大力推广袖珍机应用成果

1984 年中心与中国袖珍机协会 (CPPX) 广州联络中心联合举办了全国首届软件交易会，来自全国二十多个省市自治区的几十个厂家和推销商，直接参加交易会的正式代表就有 300 多人，参观展销的人数不胜数。及时向用户介绍袖珍机应用成果，羊城 90 工程测量应用袖珍机软件成果交流大会 5000 元，为赞助单位之首，并不惜调用了大量人力物力，为搞好大会作出了应有的贡献。为了推广袖珍机成果，支持 CPPX 测量专业委员会的工作，投资 2 万元印刷发行工程测量程序集 (第一辑、第二辑)，使测量专业应用袖珍机走在其他行业的前头。为了促进袖珍机的发展，及时出版了多种袖珍机技术资料，早在 83 年就出版了 PC-1500 袖珍机丛书，此后，出版的有 HX-20 使用说明书，MZ-731 使用说明，HX-20 技术手册，近年出版的有 PC-1500 系统 ROM 分析，E-BASIC 应用，1500 驻机 ROM 分析与应用等，到目前止，这些技术资料总销量达 2 万余册，仅林浩顺、卢耀权合编的“PC-1500 袖珍机 BASIC 语言与应用”一书，就曾重印五次。

### 三、参与技术合作圈，一齐推动袖珍机的应用

中心与国内享有盛名的报刊杂志有着深厚的友谊，如与《计算机世界》《软件报》《电子报》《计算机信息报》《电脑》杂志等，一直保持着密切的联系，积极为这些刊物组织袖珍机应用新技术和介绍袖珍机新产品的文章，中心与国内有名望的袖珍机专家保持着密切的合作，为他们铺路架桥，联合开发出万字汉卡，开发出 E-BASIC 系统和电子记录模块，编写出版应用程序集等。这些实际的工作，把一大批技术力量发挥起来，这种力量所产生的效应，是不可估量的。

### 四、坚持为生产、为科研、为国防的需要开发产品

中心抓住用户最多的 PC-1500 机，重点开发此产品，使小机也能做大事。目前已开发的产品有：万字汉卡、多功能扩充卡、GE 系列卡、GLC 电子记录模块，超级汉卡、永久记录器、固化卡、515P 汉字卡、数据采集板等。现在中心正与其他单位合作，正在开发地籍测量数据采集电子手簿。中心的产品已销往新疆、黑龙江、四川、云南、山东、湖南、湖北、江西、广西、福建、北京、上海等全国二十多个省市地区。

### 五、坚持销售与技术咨询和维修相结合，切实保障用户利益

中心一贯坚持用户第一的原则，销售产品始终是建立在向用户介绍产品性能及应用基础上进行的，销售高科技产品，技术咨询是重要的一环。几年来，中心收到来自全国咨询信件上万封，上门来访者更不计其数了，中心工作人员都一一作耐心的答复。为进一步保障用户利益，从 86 年起实行了销售的产品三包一年，用户买了中心产品，包用户会使用，包保修，不合用包换。对中心研制生产的产品，实行保用三年，终身保修服务。目前，中心已经成立了全国最大的一家袖珍机维修单位，几年来经维修的袖珍机达几千台次。

### 六、呼吁生产袖珍机

进口的袖珍机为 PC-1500，HX-20，pb-700 面临绝种阶段，即国内库存已很少，而国外生产厂家又不肯供货了，但事实上生产，科研中，每年还需要相当数量的普及型袖珍机，处在袖珍机短缺的历史时期，怎么办？我们呼吁国内电脑生产厂家担负起时代的重任，走自己生产的道路！

## 美迪英汉字典计算机投放市场

KE-101 美迪电子英汉字典计算机可显示英语单词 26000 个及 512 个常用片语，更将《美国大学入学考试》中经常出现以及托福、GRE 考试常用单词收集在内。该机可同时显示英语单词、中文解释、K·K 音标、词性及该单词原始式、复数、过去式、过去分词、比较级或最高级。同时，该机计算功能可显示整个运算过程（四则运算），答案可达 12 位数。适合大、中学生、科研人员、外贸人员及有志学习英语之人士使用。

该机使用 1.5V AAA (4 号) 干电池 4 个，并有一年免费保修服务。每台售价 380 元，邮购每台加 15 元。

该机已获国家专利。由深圳宝伟电子有限公司生产。

竭诚欢迎来人来函洽谈批发业务。

广东省教育服务公司总经销。

地址：广州市广仁路四号之二

电话：352553 邮码：510035

电脑杂志社经营部：

地址：广州市德政北路 393 号 邮码：510055

电话：330644 联系人：萨成熙



## 供应陕西计算机厂中华学习机

陕西计算机厂中华学习机全国中华学习机测试评比第一，全国总销量占有率超过 50%，是学校家庭用计算机的最佳选择。

陕西计算机厂中华学习机

广州特约销售维修点（门市部）：

广州市广仁路四号之二

广东省教育服务公司

电话：352553

邮码：510036

电脑杂志社经营部：

地址：广州市德政北路 393 号

邮码：510055

电话：330644







# 保护知识产权的重大措施

## ——祝贺中国著作权法的施行

《中华人民共和国著作权法》于1990年9月7日公布，将于今年6月1日施行。这是我国保护知识产权又一重大措施。多年以来，尤其是实行开放改革政策以来，我国逐步以法律形式实行对知识产权的保护，著作权（版权）法的施行，将使这种法律保护更广泛，更完善。我们对此表示热烈祝贺。

中国著作权法的目的，是通过法律保护作品作者的权益，鼓励有益于社会主义精神文明，物质文明建设的作品的创作和传播，促进社会主义文化和科学事业的发展与繁荣。当然，著作权有其特殊性，即作为著作权对象的东西是创造性和创新性的体现，由此形成一国的文化。所以，尽管在实施中会有实际问题并对公众增加负担，著作权仍被公认为是为了公众的根本利益而值得加以保护的一种财产形式。最基本的精神是：独创是作品取得版权的最完备条件；抄袭是最典型的侵权。作者、出版者都应充分知晓和严格遵守这一法律。

中国的著作权法明确指出计算机软件属于受该法保护的作品范围，且将由国务院制定具体条例。这对广大计算机生产者，工作者，有着重大意义。

首先，软件工作者的智力劳动成果（不论其是否以商品形式出现）从法律上得到保护，这便有力地遏制了对软件成果不加尊重，随意复制出售的现象，这将直接提高软件生产和编写工作人员的积极性。

同时，也促使我国软件工作者更严肃地对待国内外同行的作品。在制作兼容性软件的时候，将更严格地使自己的作品符合明确的法律要求。

我国是将加入世界版权公约的国家，我们同时要密切注意本国作品在国外的版权。我国软件制作，不论是国内还是国外，将处于既可能获得著作权保护又可能会受到侵权指控的复杂处境。正因为这样，我们更应趁着版权法施行的大好时机，认真学习，熟悉这一法律武器，发扬一贯的创造精神，推进我国软件产业的发展。■

# 计算机的法律保护

## ——迎接中国版权法的施行

华南师范大学微电子学研究所 王桂海

【编者按】中国的《著作权法》于今年6月1日施行。计算机软件属于受该法保护的作品。为配合该法的实施，引起广大计算机软件工作者对版权的重视，我们将连续刊载有关计算机软件版权保护的文章，欢迎广大读者参加讨论。

1990年9月7日，中华人民共和国第七届全国人民代表大会常委会第十五次会议通过了《中华人民共和国著作权法》，并将于1991年6月1日起施行。这是我国继实施专利法之后实行知识产权保护的又一重大事件。本法把计算机软件列为受保护的作品（第三条、（八）），其保护办法由国务院另行规定（第五十三条）。

计算机软件被列为受著作权（版权）法保护的作品，是国内外都很关注的问题。我国广大的计算机工作者，尤其是软件工作者，将因专利法之外，更受到另一范畴的知识产权（发表权、署名权、修改权、保护作品完整权、使用权和获得报酬权）保护而发挥更大的积极性。在国际上，则是对长期埋怨中国不施行版权保护的国家，一个有力答复。1991年3月4日，美国洛杉矶时报报导：美国贸易方面的高级官员指出，如果中国不采取有效步骤制止计算机软件的抄袭行为，美国将在关税方面采取报复措施。

并认为，由于（美国）软件的随意被（中国）抄袭，美国在中国每年损失4亿美元。美国官方认为，计算机软问题是知识产权中的一个大问题。这个问题不解决，将严重影响中美贸易。中国外交部发言人在4月25日举行的记者招待会上则明确表示，中国奉行保护知识产权政策，美国对中国的指责没有道理。他说，10年来，中国在保护知识产权方面已做了许多工作，专利法已实施多年，著作权法将于今年6月1日施行。计算机软件保护问题在著作权法中已经规定，计算机软件保护法正在制定过程中。这些法律、法规的制定，对于推动科学技术进步和文化艺术的繁荣起了很好的作用。他还说，对于保护知识产权问题，我们的态度是积极的。如果美方仍然把中国列入所谓“重点国家”名单，那是“没有道理的，也不利于中美经贸关系的发展。”由此可见，版权问题，计算机软件的法律保护问题，是一个牵涉到国际贸易，国家利益的敏感问题。

但是,由于中国施行知识产权保护为尚短,版权法尚待实行;加以宣传教育还不普及,目前仍有一些软件工作者对著作权法不十分清楚,或者未能给以足够的重视。而弄清楚著作权法和专利法所保护的范同,弄清在什么样的情况下使自己的作品受到保护,怎么防止别人的侵权,对发展我国计算机软件,是有重大意义的。

为了便于讨论,我们分两步来论述,首先是专利与版权之间的保护范围的讨论,然后是版权范围内计算机软件和其它作品之间保护范围的比较,以及计算机软件的“表达形式”与“思想”界限的讨论。

### (一)

专利和版权,都是专有权,但两者都不得保护任何一种思想和理论。专利可以保护按照一定的思想或理论研究出的具体方法和具体技术,必要条件是保护对象具备新颖性(此前无人登记过),先进性(与现有技术水平比,有一定程度的发明)、实用性。因此,专利有强烈的排它性,保护第一个登记者而排除以后出现的类似技术和方法获得专利的可能。在保护层次上,它保护较高的(更抽象、更概括的)层次。版权则不同,它只保护作品独创的“表达形式”而不延及作品所反映的实质内容。由此可见,版权与专利相比,只是一种弱保护,只要具有不同的表达形式,即使两个作品运用同样的技术,采用同样的思想,起到同样的作用,也不存在侵权问题。就是说:“著作权法保护那些作家……及其它有创作力的个人创作的原作品,只要其作品是‘原作’,意思就是独立创作,就可以取得著作权,即使已经有极为相似的作品存在”。总的来说,版权要求被保护的是一种创作性的作品,但并不要求它是“首创”的。

我们先看美国一个有一百多年历史的著名案例(顺带说一下,美国法官对案子的判决,在法律条文的理解上有争议时,往往引用过去的案例作为判决的法律基础,即所谓 from cases to case,这一做法,我们可以作为参教)。上世纪一位名叫 Selden 的会计专家,创造了一种新的簿记技术。为此,他写了一本书,并举了一些分户帐的实例。这本书取得了版权。后来, Baker 写了一本阐述 Selden 的簿记技术的书,书中包括了 Selden 用过的例子,尽管不是一模一样,但却十分近似。于是, Selden 的继承人便控诉 Backer 侵犯版权。

1879 年,美国最高法院判决 Baker 没有对 Selden 构成侵权。这是因为法院指出了可以在书中描述的“实际技术”(practical art)本身和论述“实际技术”时所采用的表达方法之间的区别。

法官对原告说:“如果这种实用技术要求保护,那可以去申请专利(能不能获准,那是另一回事)。版权只保护实用技术是如何被描述的(而不是实用技术本身)”法官还认为,两本书如果因为对同样的实用技术进行描述而引起相似的话,是不能求助于版权法的。Baker 只是在内容而不是表达方法与 Selden 相同。

从这个例子可以看出,版权所保护的创造性属于很低

的层次,即具体的表达形式。而在这之上,它都认为是属于思想、理论一类,不是它的保护范围。带有创造性的技术、工艺、手段都是属于专利所管辖的。美国 1976 年《版权法》第 102 条(b)款规定:“在任何情况下,对作者原始作品的保护都不延及任何一种思想、处理过程、工艺、操作方法、概念、原理或发现,不论它们在该作品中是以何种形式被描述、阐明、图示或体现的。”

专利是强保护,不能自动获得,要经过申请和严格的审定。它的保护时间较短(一般不超过 20 年)。版权是弱保护,但这只是表明它与其它法律有不同的分工,决不是说它不重要。它牵涉的范围很广,而且可以在作品产生时(不论其是否发表)自动取得(有些国家规定在发表的作品上加上作者的某类声明或特定标志)。版权的保护期间很长,一般是作者有生之年及死后数十年。所以,认为版权是弱保护而忽略它的作用,将是一个不幸的误解。

### (二)

版权法是对包括计算机软件在内的作品进行保护的,在繁杂的作品类型中,不可避免地会出现判定上的差异(我国计算机软件保护办法由国务院另行规定,这种有明确针对性的法律条文,将有利于版权法在计算机软件范畴内施行)。1981 年 1 月 1 日,美国新的版权法第 101 条明确规定,计算机程序属于作品,受版权法保护,而该法所给出计算机程序的定义则是“指为了获得某一结果,直接或间接用于计算机的语句或指令的组合”。

如果从这个定义去看,“表达形式”定义在“语句或指令的组合”,是非常恰当的,也是十分有限的,和专利的保护有明显的“分工”。现在的问题是,由于作品本身的复杂性,“表达形式”与“思想”并不易于截然分开,从实行版权保护多年的国家(例如美国)的许多案例来看,在这两者之间,经常是有争议的。计算机软件的案子也是如此。

现在,我们看看另一个著名的美国案例:1986 年美国管辖新泽西、宾夕法尼亚、德拉维尔三州的法院判决 Rand Jaslow 对 Elaine Whelan 的计算机软件侵权。事情的经过是这样的:1978 年 R.Jaslow 雇用 Whelan 编制一个用于管理牙医实验室的软件, R.Jaslow 提供研制费用和给予 E.Whelelan 熟悉牙医管理业务的机会。双方都希望这个软件能打入市场。Jaslow 同意让 Whelan 保有版权,而自己则以取回收入的 10% 作为代价。一年后, Whelan 向 Jaslow 交出了用较难懂的 Event Driven 语言写成的 Dentalab 程序,此程序可以在 IBM 硬件上运行。这个软件投入了市场并获利。

1982 年,个人计算机大量兴起, Jaslow 认为,若以常用的语言例如 BASIC 来写一个类似于 Dentalab 的程序,而且用于 PC 机上,将会有个很好的市场。于是,他自己用 BASIC 语言重写了一个运行于 PC 机上的软件,称为 Dentcon,于 1983 年以较 Whelan 的软件便宜 10,000 美元的价格投放市场。于是, Whelan 向法院控诉 Jaslow 侵权。

Whelan 的主要依据是认定作品的结构 (Structure) 是其表达形式, 应受版权的保护, 她的律师在法庭上指出版权法把软件看作是文学作品, 如果承认小说与戏剧的结构受保护, 那末, 同样适用于计算机软件。

而 Jaslow 的主要依据则认为他只是引用了 whelan 程序中的属于“处理过程、工艺、系统和操作方法”。这种东西在版权法上明确规定是不受保护的。他只是研究过 Whelan 程序中的源码, 并没有照搬她的源码。

法院最后判决了 Jaslow 侵权, 其理由是 Whelan 所拥有的版权应包括程序操作、控制等方式; 计算机接收、汇编、检测和屏幕上产生有用信息等等方法。因此, 程序的结构 (Structure)、顺序 (Sequence) 和组织 (Organization) 都是版权所保护的。在这些层次上出现相似, 是认定侵权的主要依据。

这一判决, 提出了一个判别计算机软件的“表达形式”和“思想”界限的新标准, 即所谓 S.S.O 准则, 把表达形式扩大到结构、顺序和组织这个层次上。而“思想”则只包括“一般目的和功能”。除非某个程序只能用唯一的或级少的方法去构成, 才把结构层次考虑为思想的一部分, 除此之外, 结构, 顺序, 组织都属于保护范围。

法院这个判定, 在美国国内外引起颇多争议。一些制造计算机兼容软件的国家认为这是随意扩大了软件的保护范围, 不利于兼容软件的发展。在美国, 有人指出, 这个判决, 干脆把计算机软件看作是文学小说一类作品, 是没有考虑到计算机软件 (程序) 的复杂性和它工程化的特殊性, 把一些应当由专利去保护的内容搬到版权上来。匹兹堡大学法律教授 Pamela Samuelson 指出: 软件是第一个被列入版权保护的技术, 在这之前, 技术和实用性的著述都是由专利法去保护的。美国国会在 1980 年把软件补充入版权法的权限, 使之同时受专利和版权保护时, 并没有明确这两者的关系, 而 Whelan 案的判决, 恰好是抹煞了两者的区别这个关键问题。她说: 在 Baker 对 Selden 案中所认为版权不应保护而应让专利去考虑保护的“方法”, 在 Whelan 对 Jaslow 案中却列入了版权保护。而其实, 应当看到版权所保护的作品是分许多个层次的, 计算机软件从高层到低层, 版权只保护由编码组成的、按指令逐条排列的序列而不应延及其更高的层次。当然, 目前还不能找到层次划分的准确模式, 但却不能因此而把“表达形式”随意扩大。这样, 很不利于软件的工程化, 不利于软件的发展。

著作权法的根本目的, 是鼓励创作, 我国著作权法第一条便指出, 它“鼓励有益于社会主义精神文明、物质文明建设的作品的创作与传播, 促进社会主义文化和科学事业的发展与繁荣”。所以, 这首先要求作品是创作的。但是创作的东西能不能一定“与众不同”, 则取决于作品的类型。国外有人把作品分为三大类型: 艺术性作品 (Artistic works), 纪实性作品 (factual works) 和功能性 (或业务性) 作品 (functional works)。一般来说, 这三个类型中

艺术性作品享有最广阔的保护范围。即是说, 这类作品的自由度较大, 表达形式十分广泛, 相重性应当很低。以侦探小说为例, 第一万本侦探小说, 从版权的角度来看, 仍要求它较之以前的 9999 本能告诉读者某些新的东西。但是, 纪实性作品 (例如传记, 肖像画) 其表达形式便受到相当大的客观事实的限制了。因而, 版权给予它的保护范围便狭窄得多。两个画家对同一个人画一幅肖像画, 如果是采取同一角度 (例如正面), 尽管是独立创作, 在形式上 (形似), 我们不能指望他们的作品有更多的区别。这么一来, 版权所提供的保护范围, 只能是很低的表达方式的层次 (例如大小、色彩、线条、光线处理等), 在这之上的层次再要求有差异, 便不合理了。这种情况下, 两幅画尽管形式上十分相似, 相互之间也不能认为是侵权的。至于功能性作品, 例如工程图, 技术论著等, 其表达形式的范围更窄, 因为主题 (不受保护) 相同之下, 功能作品只能是大同小异, (这个“小异”, 反映了提供可保护的范围)。比如汽车的蓝图, 总难免有四个轮子。这类作品只有最终的工程图构本身, 是受保护的。在这之上, 结构、工艺、原理、方法, 统统不属保护的范畴, 不管你的工程设计运用了多奇妙的技巧, 多么先进的思想和理论, 版权都不能为这些东西提供保护。计算机软件属于哪类作品呢? 从它的作用和工程化处理的特点来看, 它应属于第三类作品, 即只享有最窄的版权保护范围, 也就是说, 它在最低的一个层次 (编码) 上, 享有版权, 编码便是它的“表达形式”。对于这三种类型的作品及其保护范围, 可以用图 1 来表示。

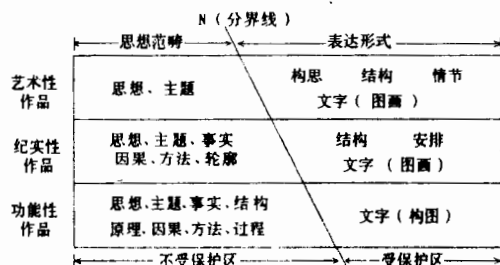


图1 三种不同类型作品受版权保护区域大小不同  
对于计算机软件, 我们可以得到如图2的表示

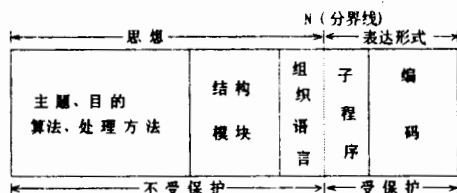


图2 计算机软件的版权保护区域示意图

当然, 图1、图2的N (分界线) 落在什么地方, 是根据不同的国家, 不同的法律, 甚至同一法律下不同法官的理解才可以确定的, 但这个界线一定存在。就计算机软



# 利用计算机打印条形码程序

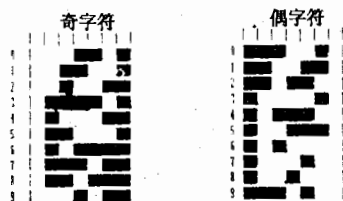
解放军后勤工程学院自控室 冯云

条形码技术是随着计算机技术的发展而产生和发展起来的一门数据输入技术,由于其高准确度、低成本及简便性,被越来越多的广泛应用于各个领域。

对于小范围应用条形码时,则条形码的印制就比较麻烦,特别是在变化较频繁的时候。采用个人计算机进行打印,则是一个比较方便的办法。本人用 pascal 语言编写了一个打印美国通用商品条形码;即 UPS-A 码的打印程序,只要输入相应的商品代码,即可以打印对应的条形码。

UPC-A 码由 12 位数字构成,第 12 位为校验位,其求法是:前 11 位中奇数之和乘以 3 再加上偶数位之和,这个和再加上一个数字成为 10 的倍数,这个数就是校验位数字。

UPC-A 码前 6 位为奇字符,后 6 位为偶字符,其编码规则如下:



相应的计算机程序附后。此程序大致思想是,先读入需打印的数据,计算出校验码,将数字对应的条形码存入一个数组中,然后将整个条形码数组用点阵方式打出,每个点打两遍,最后打印出条形码对应的数据,文末是几个打印出的几个例子。

件来说,它一定落在“编码”与“主题、目的”之间的某个位置上。由于 N 的不易确定,便构成许多争议,这是我们处理软件版权保护时应特别注意的。就是说,如果我们创作一个软件,我们能指望版权所保护的,只是接近或属于编码层次这样的表达形式。若我们确信自己的作品还有其它更高价值的技术性成就而要求保护,应当走专利保护的途径。

任何时候,独创是获得版权的最有力依据,只要作品能被有力地证明是独创的,则不论它与已有的作品是否相似,都可以获得版权。抄袭行为是违反版权法的,不管其多么巧妙,只要被证实是抄袭;或者,被指控抄袭而提不出独创的充分证据时,都会被认为侵权。至于兼客软件如何获得版,解决既与原作相似,又能被承认是独创的问题,将在另文再予讨论了。■

如需打印其它规则的条形码,可类似编写相应的程序。

```
program upca;
type codem=array (0..9, 1..7) of byte;
const left: codem=( (0, 0, 0, 1, 1, 0, 1),
(0, 0, 1, 1, 0, 0, 1), (0, 0, 1, 0, 0, 1, 1),
(0, 1, 1, 1, 1, 0, 1), (0, 1, 0, 0, 0, 1, 1),
(0, 1, 1, 0, 0, 0, 1), (0, 1, 0, 1, 1, 1, 1),
(0, 1, 1, 1, 0, 1, 1), (0, 1, 1, 0, 1, 1, 1),
(0, 0, 0, 1, 0, 1, 1));
right: codem=( (1, 1, 1, 0, 0, 1, 0),
(1, 1, 0, 0, 1, 1, 0), (1, 1, 0, 1, 1, 0, 0),
(1, 0, 0, 0, 0, 1, 0), (1, 0, 1, 1, 1, 0, 0),
(1, 0, 0, 1, 1, 1, 0), (1, 0, 1, 0, 0, 0, 0),
(1, 0, 0, 0, 1, 0, 0), (1, 0, 0, 1, 0, 0, 0),
(1, 1, 1, 0, 1, 0, 0));
var p: text; n, code, d, dum: integer;
darcode, xlm: string;
m: array (1..96) of integer;
procedure warnlar;
var i: integer;
begin
i:=4;
repeat
m(i):=0;
m(i+47):=0;
i:=i+1;
until i>45;
end;
procedure printout;
var i, j: integer;
begin
write(p,chr(27),' G ',chr(00),chr(192));
i:=1;
repeat
j:=255*m(i);
write(p, chr(j), chr(j), chr(j));
write(p, chr(j), chr(j), chr(j));
i:=i+1;
until i>96;
write(p, chr(10));
end;
procedure summer;
begin
repeat
val(darcode(n), d, code);
sum:=sum+d;
n:=n+2;
until n>13;
end;
procedure putmemo(oddeven: integer);
var i, j, k, d: integer;
begin
i:=1;
repeat
val(darcode(i+oddeven*6), d, code);
```

```

repeat
  case oddeven of
    1: m( (i-1) * 7+j+3): =left (d, j);
    0: m( (i-1) * 7+j+50): =right (d, j);
  end;
  j: j=j+1;
until j>7;
i: =i+1;
until i>6;
end;
begin
  assign (p, 'prn'); rewrite (p);
  writeln (p, chr (27), 'J', chr (09));
  writeln (' PLEASE INPUT DAR_CODE ');
  readln (darcode);
  n: =1;
  repeat
    m (n): =1;
    m (n+1): =0;
    n: =n+2;
  until n>95;
  putmemo (0);
  sum: =0;
  n: =1;
  summer;
  sum: =sum * 3;
  n: =2;
  summer;
  sum: =sum-10 * trunc (sum / 10);
  sum: =10-sum;
  str (sum, xlm);
  darcode: =darcode+xlm;
  putmemo (1);
  n: =1;
  repeat
    printont;
    u: =n+1;
  until n>10;
  warndar;
  printont;
  writeln (p, chr (10));
  writeln (p, darcode);
end

```

## 可限定时间的操作训练程序

沈阳市苏家屯铁路中学 陈志岭

在我国计算机事业飞速发展, 中华学习机以它独特的优点, 进入一些中小学校和一些家庭, 成为综合技术教育、职业技能培训的得力工具。但就使用者的识盘能力、操作指法如何, 不同的人应该是有差别的。通过运行本程序, 可清楚地测定出操作者的键盘操作水平的高低, 以便自我了解和进一步提高。

### 程序的功能特点

1. 可限定每次操作训练的时间, 省去人为记录时间, 计算出平均每分钟键入正确字符数的麻烦, 也使一机多人共用的教学时间分配得公平合理。

2. 可限定每次操作训练的字符总数, 同样可测试出平均每分钟键入正确字符数的速度。

3. 适应性强。由于操作者对键盘的熟悉程度不同: 优秀操作者键入速度应该是高的; 上机次数较少的操作员键入速度必然是低的, 为此它可对不同层次的使用者产生出其操作相对的科学评价。

4. 具有对比复查功能: 逐一核对标准文稿与键入字符的结果。

### 程序的原理与使用

为使程序简明、可读, 笔者用 BASIC 语言编写该程序的。程序以模块结构形成 (见程序中的注释语句)。运行中都由中文提示, 简要方框图所附, 当做限时测试时, 直接应用本程序即可。若做限量测试时, 允许操作时间选得长些, 同时需改动 20 语句中 J 变量的值, 使其为 30 的整数倍。

正确键入本程序并运行, 输入限定时间后, 微机便自动随机提取标准文稿的内容, 以每页 30 个字符数出示于屏上。操作者看着文稿键入相应的字符。键入速度越慢, 时钟走得相对稍快些。限定时间一到 (或达到操作字符总数), 即刻统计并出示测试结果。随后还可选择对比功能, 把所键内容与标准文稿 (以反白形式显示的) 对比检验。

经过两年多的教学实践, 不断补充和完善本程序, 深感离不开它了。有它不仅使课堂生动有趣味, 而且也为计算机操作竞赛、选拔培训优秀操作员等工作提供了准确科学的理论依据。现将它奉献给《电脑》杂志的爱好者们, 共同分享其快乐。

本程序适用于 CEC-I 及兼容机。

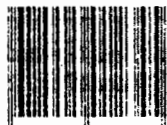
```

10 PRINT CHR$(4)*PR#3: PRINT: PRINT CHR$(12)
+CHR$(18): SPEED=255
12 VTAB(5): HTAB(5): PRINT"限定击键时间[1到9之":
PRINT: PRINT"间数字(分钟)]?";
14 GET I$: Y=VAL(I$): IF Y<1 OR Y>10 THEN
CALL-936: GOTO12: REM *****输入限时*****

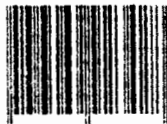
```



0987654321010

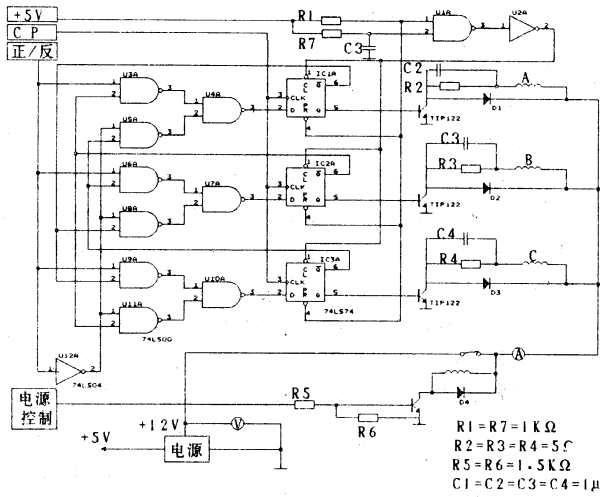


012345678905



923426736469





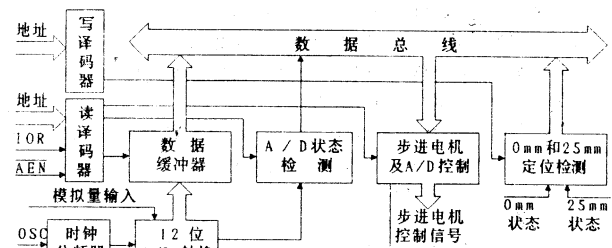
图一 步进电动机控制电路

继电器开关断开，切断 12V 电源。然而此时供给三相六拍分配器的 5V 电源正常供电，分配器输出端  $Q_1Q_2Q_3$  的脉冲分配状态不改变。当电源控制器再次通电时，步进电动机即能按原脉冲分配状态转动，既不会失步，又能省电。

从图一可见，由  $U_1 \sim U_{12}$  和  $IC_1 \sim IC_3$  组成一个能自动启动的三相六拍分配器。它的控制信号只有两个：控制脉冲输入“CP”，正、反控制信号“正/反”。因而编写工作程序及控制容易。 $U_1U_2$  及  $R_1R_2C_1$  组成的自启动电路能使分配器的三个 D 触发器  $IC_1, IC_2, IC_3$  不论是正向还是反向启动，都处于  $Q_1Q_2Q_3 = 100$  的步进电机初始脉冲分配状态。

当分配器电源接通瞬间，由于  $C_1$  的  $U_{C1} = 0$ ，与非门  $U_1$  的输入有一端为 0，使  $U_2$  输出一个 0 信号到  $IC_1, IC_2$  和  $IC_3$ 。从图一可见  $IC_1$  被置位，故  $Q_1 = 1$ ， $IC_2, IC_3$  被复位，故  $Q_2 = Q_3 = 0$ 。随后  $C_1$  被充电， $U_1$  的两个输入端均为高电平， $U_2$  输出为 1，各个 D 触发器处于触发翻转工作状态，自启动电路不起作用。由此可见，自启动电路能在分配器通电瞬间使分配器自动满足三相六拍步进电动机所需的脉冲分配要求。

输入输出通道



图二 输入输出通道方框图

本电路的方框图如图二所示。由图可见，本电路是由 256 分频电路、模拟量输入通道、A/D 状态检测和 0 毫米与 25 毫米定位检测数字输入通道，步进电动机控制及 A/D 控制数字输出通道组成。

本电路的主要作用是在微机工作程序的控制下，使步进电动机驱动机械调节系统去调节螺旋测微头转动选频激光器的反射棱镜，从而测出测微头 0~25 毫米内各个不同位置对应的激光频率，因此可由系统选定某一要求的激光工作输出频率。

1、模拟量输入通道 它能将激光雷达数据采集电路检测出来的模拟量转换为数字量，送微型计算机进行存贮，分析及处理。A/D 转换部件是 12 位的 ADC1210，以提高转换的分辨率与精度。

2、步进电动机控制及 A/D 控制数字输出通道 按选频激光器工作要求，输出数字控制信号给步进电动机，形成步进电动机的控制脉冲“CP”，“正/反”及“电源控制”等信号，同时输出启动 A/D 转换器工作的“启动”信号。

3、0 毫米及 25 毫米定位检测数字输入通道 为了防止测微头在步进电动机的驱动下，超过 0 毫米或 25 毫米工作位置，设定 0 毫米和 25 毫米定位开电路，当测微头有关装置触动这两个开关时，通道检测到高电平信号，令步进电动机反转一步后停止前进。

4、写译码器及读译码器 本控制系统采用 IBMPC 作控制用计算机，为了选道多个工作通道，用两片 74LS138 及一片 74LS20 组成读译码器，用一片 74LS138 组成写译码器，按输入地址及控制信号  $AEN, IOR$  及  $IOW$  选通不同的通道。

由上述分析可见，本系统电路设计简单、性能优良，能满足系统的控制工作要求。 ■

为科技人才树碑立传

为振兴中华贡献力量

《中国工程师辞典》华南编委会征集科技人员和企业资料

《中国工程师辞典》是第一部专门介绍各类专业科技人才及成果的大型工具书。聂荣臻、薄一波、王任重、周培源等老一辈革命家和科学家为该书题了词，赞扬该书出版“为国家建立健全了人才档案库”，“为社会和广大科技人员办了一件好事”。

凡具有高中级职称的各类专业科技人员（自然科学和社会科学）和骨干企业，均可申请入编。欲入编的个人和企业请贴附邮票回邮信封，寄中国工程师辞典华南编委会索取详细资料。地址：广州市小北路广州科技中心 1407 室，邮政编码：510050 ■

# 汉字注音的计算机实现

重庆市大足师范 宋运康

汉字拼音方案是我国文字改革的最重大成就之一。让计算机给汉字注音，在诸多方面都是很有意义的。

计算机汉字注音算法的依据是：ccdos的各种版本都有拼音输入方式，它和多种汉字编码输入方案一样，都是借助于一张编码表用查表法实现的。用拼音输入汉字有重码，但汉字反过来到拼音的对应是多对一，借用拼音编码扫描表，作必要的分析和转换，就能实现对任意汉字的注音。

这里需要简单了解一下扫描表。ccdos 2.0/2.1、3.0、4.0的拼音编码表都是与首尾、快速码表合而为一。从国标16区开始，87区结束，每区以94个汉字计（第55区也以94个字计），每一个汉字的编码占四个字节，对于2.0/2.1、3.0版，每四个字节的编码内容是：

第四字节	第三字节	第二字节	第一字节
76543210	76543210	76543210	76543210
XXXXXXX	P3	P2	P1
			S2
			S1

对于4.0版，这四个字节的编码内容是：

第四字节	第三字节	第二字节	第一字节
76543210	76543210	76543210	76543210
HHHHXX	S2	S1	X
			P3
			P2
			P1

S1, S2, P1, P2, P3 依次表示首尾第1码，首尾第2码，拼音第1码，拼音第2码，拼音第3码。x表示未用，H表示高频字信息。

上述三种版本的扫描表分别依次存放在 cccc.exe、ccdos.exe、cccc.ovr 中，用 debug 可以看到，相对于文件头的偏移量依次是 2DD6H、4DD1H、600H，当这三个文件调机内时，扫描表的起始地址相对于 cs 段分别偏移 2BD6、4BD1、400。

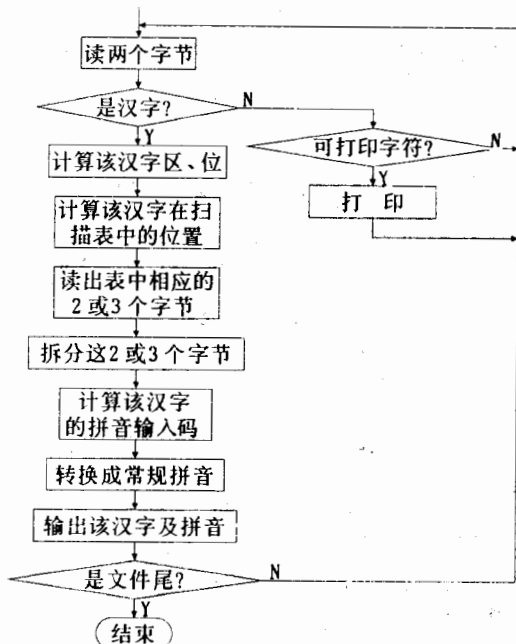
本文给出 turbo pascal 3.0 编写的文稿（文稿在 ws 纯中文状态输入）注音程序。主要流程图示如右：

程序采用读取某 ccdos 版本机内扫描码表信息实现（当然要在相应状态）。这在绝大多数情况都是正确的，但也有例外。比如 4.0 在高分单显状态下，由于系统采用 cccc.ov2 代替 cccc.ov1，而使码表首址就不是上面说的 cs: 400H（但 4.0 在高分单显装置模拟中分彩显状态的时，码表首址不变）。直接读取磁盘文件 cccc.ovr（对 4.0）、cccc.exe（对 2.0/2.1）、ccdos.exe（对 3.0）获得扫描表信息不会有例外，且在中、西文状态都行。只须把这些文件认定为 byte 型（或 char 型）的随机文件或顺序文件即可。只是要注意码表的起始位置是如前所述的相对于文件头的位置。低版本 ccdos 的拼音码表错误较多，应尽可能采用 4.0 版。

这里说明两点：

一是声调。可以在不占用任何额外内存、不影响系统的任何功能的前提下，给原扫描表补充声调信息。每个汉字的音调信息可存放于扫描表中该汉字未用的 bit（每个汉字占用三个即可），声调信息的写入可参考程序中的读码部分编写，同样把有关文件当作 byte 或 char 型随机文件。声调信息的获得可借助于某些汉字系统。人工获取一级汉字的声调也是很容易的，二级字大概需要借助一下字典。声调的显示和打印不会有什么问题，不必生造，在扩充 asc 码中几乎可以找到所需的全部加有声调符号的 a、e、o、i、u 等。

二是多音字问题。这个问题的解决要麻烦一点。笔者设想的解决方案是，先建立多音字音节文件。在国标汉字范围内按音序建立多音字音节记录，这个文件是很小的，因为汉字的音节总数只四百多个，即使加上声调也只一千多个，且能够成为多音字的音节的是极少数。这个文件的索引指针可存放于扫描表。多音字音节文件只在需要时调入机内。这类类似于 dbase 的备忘文件的处理。这样，在解决了声调和多音字问题之后，遵循方案规定的书写规则，得到的汉字注音文稿就是比较理想的了。最后要强调的是，汉字注音的计算机实现决不只用于给文稿注音，它的应用是多方面的，单就汉字信息的排序、检索而言，它的应用就是很重要的了。



附程序一:

```

B> type qq.bak
var n,i,version,codestart,segment,section,position:integer;
t:array[1..4] of byte;
c:array[1..3] of char;
c0:string[6];
b1,b2:byte;
textfile:string[14];
f:file of byte;
begin
hires;
write('namefile=');readln(textfile); assign(f,textfile); reset(f);
write('version[2 or 3 or 4]=');readln(version);
case version of
2: codestart:= $2bd6;
3: codestart:= $4bd1;
4: codestart:= $400;
end; (* 根据 dos 版本确定码表起始地址 *)
segment:= mem[0:127] * 256 + mem[0:126]; (* 查 cs 段地址 *)
while not(eof(f)) do
begin
repeat read(f,b1,b2);
if (b1 <= 175) and (b1 < 141) and (b1 < 138) then
write(chr(b1),chr(b2));
until b1 > 175; (* 读两个字符,直到是汉字为止 *)
c0:= ''; section:= b1-160; position:= b2-160; (* 确定该汉字区、位 *)
n:= codestart+94 * (section-16) + 4 * (position-1); (* 确定该汉字在码表中的位置 *)
for i:= 1 to 4 do t[i]:= mem[segment+n+i-1]; (* 读该汉字的码表信息 *)
if version = 4 then
begin
k[1]:= t[1] and $1f;
k[2]:= t[1] div 32 + (t[2] * 8) and $1f;
k[3]:= (t[2] div 4) and $1f;
end else
begin
k[1]:= t[2] shr 2 and $1f;
k[2]:= t[2] shr 7 + t[3] shl 1 and $1f;
k[3]:= t[4] and $1 shl 4 + t[3] shr 4;
end; (* 根据不同版本分析出该汉字的拼音输入码值 *)
for i:= 1 to 3 do c[i]:= chr(k[i]+96); (* 转换这些码值为三键拼音输入 *)
for i:= 1 to 3 do
case c[i] of
'a': if (i= 1) and (k[2] < 27) then c0:= c0+'zh' else c0:= c0+'a';
'y': if (i= 1) and (k[2] < 27) then c0:= c0+'ch' else c0:= c0+'y';
'u': if (i= 1) and (k[2] < 27) then c0:= c0+'sh' else c0:= c0+'u';
'o': if (i= 1) and (k[2] < 27) then c0:= c0+'s' else c0:= c0+'ong';
'e': if (i= 1) and (k[2] < 27) then c0:= c0+'f' else c0:= c0+'en';
'g': if (i= 1) and (k[2] < 27) then c0:= c0+'g' else c0:= c0+'eng';
'h': if (i= 1) and (k[2] < 27) then c0:= c0+'h' else c0:= c0+'ang';
'i': if (i= 1) and (k[2] < 27) then c0:= c0+'j' else c0:= c0+'an';
'x': if (i= 1) and (k[2] < 27) then c0:= c0+'x' else c0:= c0+'ao';
'v': if (i= 1) and (k[2] < 27) then c0:= c0+'v' else c0:= c0+'ai';
'y': if (i= 1) and (k[2] < 27) then c0:= c0+'y' else c0:= c0+'ing';
'v': c0:= c0+'u';
else if k[i]= 27 then c0:= c0+' ' else c0:= c0+c[i]; end;
(* 转换为常规拼音音节 *)
write(chr(160+section),chr(160+position),c0,'',6-length(c0));
(* 输出这个汉字及其拼音 *)
end;
close(f);
end.

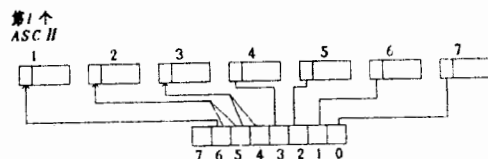
```

# BASIC 数据文件的压缩技术

福建省公安厅十一处 黄文

在《人口基本信息管理系统》中,由于人口信息的数据量庞大,按 100 字节/人计算,百万人口的城市就需约 100MB 的存贮空间,一般的微机都难以满足,因此十分必要对数据进行压缩存贮。在使用中我们实现了对 BASIC 数据文件的二级压缩。

人口信息对数据规范化要求较高,我们对数据进行了全编码存贮,这样在数据文件中实际仅存放着 0, 1, 2, ..., 9 阿拉伯数字。0~9 数字在微机内是以 ASCII 形式存放的,对数据文件以每两个数字为一组进行截取,得到结果为 00~99 之间的数字,再让每组数字对应一个 ASCII 字符,即 CHR(00)~CHR(99)。这时我们实现对原始数据的一级压缩,压缩量为 50%,数据存贮形式为 0~99 之间对应的 ASCII 字符。通过分析 ASCII 字符编码它们是以 8 位二进制数表示,能够表示 256 个字符。而我们前面只用到 100 个 ASCII 字符,小于 128。实际上只要用 7 位的二进制数就可以表示。换句话说 CHR(0)~CHR(99)的最高位均为 0,这就为我们提供了二次压缩的条件。以每 8 个 ASCII 为一组,第 8 个 ASCII 字符的低 7 位分别填到前 7 个 ASCII 字符的最高位。经过第二次压缩后数据又压缩了 1/8,两次合计压缩了 62.5%。上面介绍的数据压缩技术对其他语言编制的数据库文件同样实用。



## 书讯

### 《IBM-PC/XT 微型电子

### 计算机系统故障检修》

本书汇总了 200 个 IBM-PC/XT 微型计算机及其兼容机的常见故障与检修实例。这些故障实例基本上是作者多年来的教学和检修工作实践所积累的。故障例子真实,种类多,涉及范围广,是一份难得的资料。

全书共分十章及二个附录,分别讲述系统板故障、单色显示器及其适配器故障、彩色/图形显示器及其适配器故障、软磁盘驱动器及其适配器故障、硬磁盘驱动器及其适配器故障、打印机和适配器故障、电源故障和异步通讯适配器故障。本书附录给读者提供主要部位的波形图,IC 芯片清单等。每章开头叙述有关部件的电路原理。每个故障均叙述故障发生的现象、故障产生的条件以及详细分析故障产生的原因。全书以故障实例为主干,从故障分析出发,讲述了 IBM-PC/XT 的主要电路原理。

该书每册定价:12 元,邮购另加 1 元邮杂费(含挂号费)。附言请注明邮购内容及数量。

汇款地址:广州市 1753 号邮政信箱 萨成照收



## MS-DOS 彻底剖析 (十一)

## IBMBIO 模块数据结构详析 (下)

郭嵩山 冻政 朱国庆

## 五、为设备管理而设置的数据结构

## 1、块设备控制块 (BDCB: Block Device Control Block)

系统在 BIO 初始化时由 SYSINIT I 根据系统配置为每个块设备建立一个控制块, 用来记录该块设备的规格及当前状态的信息, 以便能对所有的块设备, 其中包括具有不同介质的软盘驱动器进行 I/O 操作, 以确保块设备操作前后的一致性。后述的驱动器参数块 (DPB) 和 BIOS 参数块 (BPB) 都是块设备控制块 BDCB 的一部分。这些块设备控制块按照其逻辑设备号顺序链接成一条链, 以便于系统的访问。所有的块设备控制块都建立在 BIO 的常驻段 (即 0070H 段), 其中 3.3 版的 BIO 为软盘驱动器保留了四个块设备控制块区域, 安排在内存 0070: 0482-05C8H, 每个 BDCB 之间用一个表格分隔符 (ASCII 码 90H) 分隔。BDCB 链的链首指针存放在 0070: 024CH 处。

BDCB 的结构如下:

偏移 (Byte)	长度 (Byte)	字段
+00H	04H	指向下一个 BDCB 的指针
+04H	01H	该块设备对应驱动器号
+05H	01H	该块设备的逻辑设备号
+06H	13H	块设备的 BPB
+19H	01H	当前介质容量描述字节
+1AH	02H	块设备的打开文件数计数
+1CH	0CH	当前介质的卷宗名
+28H	01H	驱动器规格字节
+29H	02H	块设备控制字
+2BH	02H	块设备支持的最大柱面数
+2DH	13H	块设备的 DPB
+40H	0CH	保留
+4CH	01H	块设备上一次 I/O 操作的起始柱面号
+4DH	02H	块设备为硬盘分区时, 其值为 0001H; 块设备为软驱时, 其值为最近一次占用驱动器时间计数低字
+4FH	02H	块设备为硬盘分区时, 为分区起始柱面号; 块设备为软驱时, 为最近一次用驱动器的时间计数高字

说明:

1) 最末块设备控制块的“指向下一个 BDCB 指针”字段内偏移值置为 0FFFFH, 作为链尾标志。

## 2) 介质容量描述字节各位意义如下:

位	标志设置	意义
7	1/0	介质最大逻辑扇区号 > 0FFFFH / 不大于 0FFFFH
6	1/0	介质总簇数 > FF6H (FAT 为 16 位形式) / 12 位形式
0~5		保留

## 3) 驱动器规格字节内容意义如下:

值	驱动器支持最大柱面数	驱动器支持每道最大扇区数
00H	28H	09H 或以下
01H	50H	0FH
02H	50H	09H
05H	本块设备为硬盘或硬盘分区	
07H	本块设备使用非标准格式驱动器	

4) 块设备控制字用来记录当前块设备的各种状态, 以作下一步操作的依据。其各位意义如下:

位	标志设置	意义
00H	1	本块设备支持不可移动介质
01H	1	本块设备支持大容量驱动器
02H	1	BDCB 中驱动器参数块 (DPB) 要更新
03H	1	当前访问道中各扇区格式一致
04H	1	共享标志
05H	1	占用驱动器标志
06H	1	介质卷宗名已更新
07H	1	BDCB 有效标志
08H	1/0	介质已变 / 介质未变
09~0FH		保留

## 2、磁盘参数表 (DPT: Disk Parameter Table)

DOS 对逻辑驱动器的 I/O 操作是通过 DPT 来实现的, DPT 主要包含了有关扇区、设备标题指针等有关信息, 以便于系统用来计算某个簇号所对应的物理扇区, 对该表参数的调用是由 DOS 内核来完成, 对用户则完全透明, 其 DPT 链首指针存于环境管理块 (EMB) 偏移 +00H 处, 每个 DPT 表长度为 20H 个 Byte, 其结构如下:

偏移 (Byte)	字段内容
+00H	相同设备的 BPB 序号
+01H	本 DPT 对应的逻辑驱动器号
+02H	每扇区字节数
+04H	每簇扇区数-1
+05H	每簇扇区数对 2 的对数
+06H	保留扇区数
+08H	FAT 个数
+09H	根目录项数

(续上表)

+0BH	文件区起始扇区号
+0DH	文件区所占扇区数
+0FH	每个 FAT 占用扇区数
+19H	目录区起始扇区号
+12H	相应设备驱动程序设备标题指针
+16H	介质描述字节
+17H	该 DPT 使用标志, 00 在使用, FFH 未使用
+18H	下一 DPT 指针
+1CH	当前目录的首簇号(0 为根目录, 1~FF7H 为目录文件首簇号, FF8~FFFH 未定)
+1EH	未用 (2 个 Byte)

说明:

- 1) 相同设备的 BPB 序号字段反映了块设备物理驱动器与逻辑驱动器的一对多关系。
- 2) 保留扇区数实际上是引导记录所占扇区数。
- 3) 文件区起始扇区号字段值实际上包含了引导记录、FAT 表和根目录所占扇区总数。
- 4) 介质描述字节各位意义如下:

位	标志设置	意 义
0	1/0	双面/单面磁盘
1	1/0	每道 8 个扇区/每道非 8 个扇区
2	1/0	可移动/不可移动介质
3~7		保留

介质描述字节标识了当前介质的类型, 但与实际介质类型如何对应则取决于设备驱动程序。

- 5) 最末一个 DPT 的“下一 DPT 指针”字段偏移值置为 FFFFH, 作链尾标志。

3、驱动器参数块 (DPB: Driver Parameter Block) 和 BIOS 参数块 (BPB: BIOS Parameter Block)

驱动器参数块包含了块设备所支持的标准介质的规格的各项参数, 块设备根据当前介质的不同规格而采用不同的驱动器参数块。DPB 的长度有 13H 个 Byte, 其结构如下:

偏移(Byte)	长度(Byte)	字 段
+00H	02H	每扇区字节数
+02H	01H	每簇扇区数
+03H	02H	保留扇区数
+05H	01H	FAT 个数
+06H	02H	根目录项数
+08H	02H	总扇区数
+0AH	01H	介质描述字节
+0BH	02H	每个 FAT 占用扇区数
+0DH	02H	每道扇区数
+0FH	02H	磁头数
+11H	02H	隐含扇区数

IBMBIO 使用了多个 DPB, 下面列出 3.3 版的 DPB 在内存位置。

名 称	装入内存地址
引导记录 DPB	0000: 7C0BH
规格标志 = 00H 的 DPB	0070: 2A58H
规格标志 = 01H 的 DPB	0070: 2A6CH
规格标志 = 02H 的 DPB	0070: 2A80H

BIOS 参数块包含了块设备中物理磁盘的特性, 可供

驱动程序根据给定的逻辑扇区号来计算相应的磁盘物理地址, 系统为每个块设备设置一个 BPB, 长度也为 13H 个 Byte, 其结构前 11H 个字节 (00~10H) 与 DPB 相同, 只是最后一个字段 (偏移+11H 处), 当块设备是硬盘分区时, 存放分区的起始逻辑扇区号。BIOS 内设备驱动程序根据物理磁盘的不同特性而在初始化时修改该块设备的 BPB, 以保证磁盘 I/O 操作能正常执行。为便于使用, BIOS 的保存有几种特定磁盘 BPB, 这些 BPB 内容是固定的, 不能由程序来修改, 这些特定的 BPB 在 IBMBIO 模块中位置如下表所列。

首 址	名 称
CS: 1E24H	5.25", 320/360KB 软驱的 BPB
CS: 1E39H	5.25", 1.2MB 软驱的 BPB
CS: 1E4EH	5.25", 720KB 软驱的 BPB
CS: 1E4EH	8", 单密软驱的 BPB
CS: 1E4EH	8", 双密软驱的 BPB
CS: 1E4EH	硬驱 BPB
CS: 1E4EH	磁带驱动器 BPB
CS: 1E4EH	其他设备的 BPB

驱动器参数块 (DPB) 和 BIOS 参数块 (BPB) 保存在相应块设备控制块 (BDCB) 中。通常将 DPB 前 0DH 个字节称为系统的 BPB。

#### 4、BPB 指针数组

BPB 指针数组是用来存放 BPB 首址偏移值的数组, 按照 BPB 所对应的逻辑块设备号, 每 2 个 Byte 存放 1 个块设备的 BPB, 依次存放。

在 SYSINIT I 运行中根据系统配置的块设备数目在内存 0070: 18A4H 处建立了一个 BPB 指针数组。此外 IBMBIO 模块内也为前述的 8 种特定驱动器的 BPB 建立了一个指针数组, 其首址为 CS: 1E61H, 长度 10HByte。

#### 5、磁盘 I/O 基数表 (DISK-BASE)

INT 1EH 的中断向量指向系统磁盘的 I/O 基数表首址。该表规定了软盘控制器控制软盘驱动器工作时所用的一些物理参数。在 IBMBIO 初始化过程中, 曾根据磁盘 I/O 操作的需要, 修改过磁盘 I/O 基数表。在 IBMBIO 运行的不同阶段, 并保持过两个磁盘 I/O 基数表, 一个是由 ROM BIOS 中读出到内存 0000: 0522H 处, 另一个是引导记录建立的 I/O 基数表, 其首址在 0000: 7C2BH, 磁盘 I/O 基数表长度 11 个字节, 其结构如下:

偏移	含 义	内容
+00H	高四位为步进速率, 低 4 位为磁头卸载时间	DFH
+01H	高七位为磁头加载时间, 低 1 位为非 DMA 方式	02H
+02H	马达等待时间	25H
+03H	扇区容量 (0: 128B, 1: 256B, 2: 512B, 3: 1024B)	02H
+04H	每道扇区数	09H
+05H	扇区间隔字节数	2AH

(续上表)

+06H	每扇区字节数('扇区容量'=0)	FFH
+07H	格式化时扇区间隔填充字节	50H
+08H	格式化时扇区数据区填充字节	F6H
+09H	寻道后磁头稳定时间	0FH
+0AH	命令等待时间(马达启动时间)	02H

## 6. 磁盘路径表 (DPAT: Disk Path Table)

DPAT 表是 MS-DOS 3.X 及其以上版本增加的一种数据结构, 是为对逻辑驱动器访问而设置的, 反映了相应逻辑驱动器的当前路径。在 BIO 的 SYSINIT II 工作时, 曾三次在 RAM 高端建立过临时的 DPAT, 但都在其工作过程结束时释放, SYSINIT II 最后根据系统的配置在 RAM 的低端建立起一组供 DOS 内核使用的 DPAT, 每个 DPAT 对应一个逻辑驱动器, 每组 DPAT 的个数可能大于磁盘参数表 DPT 的个数。每个 DPAT 长度为 51H 个 Byte, 由环境管理块 (EMB) 来管理, 其结构如下:

+00	+43H	+45H	+49H	+50H
目录路径区	标志	相应 DPT 指针	保留	结束标志

各字段意义如下:

目录路径区: 存放相应的逻辑驱动器当前路径, 长度为 44H 个 Byte;

标志: 该域值表明有无相应的 DPT, 其值为 4000H, 表示有相应的 DPT, 其值为 0000H, 表示无相应的 DPT, 其值为 8000H, 表示为远程设备。该域长度为 2 个 Byte;

DPT 指针: 长度 4 个 Byte;

结束标志: 其域值为 00H。

## 7. 地址场

地址场是块设备驱动程序实现命令代码 13H (扩充 IOCTL 功能) 进行读、写、校验与初始化四种子功能操作时所用的一种数据结构。命令 13H 支持对扇区格式不一致 (即每扇区包含的字节数不一致) 的磁盘进行 I/O 操作。地址场记录了某一指定磁道的各个扇区所包含字节大小的信息、地址场中的项数与磁盘每道扇区数相同, 每一项含 4 个字节, 顺次存放柱面号、磁头号、物理扇区号和每扇区字节数。地址中各项依次从低扇区号往高扇区号排列, 地址场各项之前的头两个字节存放当前介质的每道扇区数 (即该地址场的项数), 执行 13H 命令代码时, 先查出指定扇区的每扇区所包含的字节数, 然后将其值送至相应单元存放, 后再对相应扇区进行 I/O 操作。

IBMBIO 在内存 0070: 11D8H 处开辟了长为 OFEH 个字节用来建立当前的地址场, IBMBIO 最多可支持每道 47 个 (3FH 个) 扇区。

## 六、为 SYSINIT I 运行而设置的数据结构

## 1. IBMBIO.COM 文件参数表

在 IBMBIO 搬家子程序执行过程中, 将根据 FAT 重新将 IBMBIO.COM 文件读入内存, 为了完成这个操作, 系统在首址为 0000: 7BE2H (3.3 版) 的内存建立了一个 IBMBIO.COM 文件参数表, 该表在重装入过程结束后将释

入其所占的内存资源。该表包括系统盘及 IBMBIO.COM 文件的各种有关参数, 长度为 1BH 个字节, 其结构如下:

偏移(Byte)	字段内容
00~01H	磁头数
02~03H	每簇字节数
04~05H	I/O 操作的首扇区号
06~07H	I/O 操作的扇区数
08~09H	每个 FAT 所占扇区数
0A~0BH	隐含扇区数
0C~0DH	保留
0E~0FH	保留扇区数
10~11H	I/O 操作首簇号
12~13H	当前内存缓冲区首址 (偏移值)
14~15H	IBM.COM 文件起始逻辑扇区号
16~17H	每道扇区数
18H	驱动器号
19H	FAT 标志(01H 为 12 位 FAT, 04H 为 16 位 FAT)
1AH	介质描述字节
1BH	文件结束标志(FFH 结束, 00H 未结束)

## 2. 软盘空间分配格式表

在初始化 SYSINIT I 为系统个块设备生成块设备控制块时, 若软盘格式不是 IBM 标准格式, 则根据总扇区数查找相应的软盘空间分配格式表, 查出每簇扇区数、根目录项数及介质容量描述字节并将其填入块设备控制块 (BDCB) 中。在 SYSINIT I 工作期间, 由 SYSINIT I 在内存 0070: 2A22~2A41H 处 (共 20H 个 Byte) 建立 4 个磁盘空间分配格式表, 每个表长 8 个 Byte (最后一个字节存 00H, 作为结束标志), 其结构如下:

偏移(Byte)	字段内容
00~01H	逻辑卷中总扇区数
02H	每簇扇区数取以 2 为底的对数
03H	系统簇扇区数
04H~05H	根目录项数
06H	介质容量字节

## 3. 支持高容量软盘驱动器操作记录表

该表记录了 BIO 设备驱动模块中所有与高容量软盘驱动器相关指令存放的内存地址, 也即记录了从某一地址起有若干个字的指令与高容量驱动器有关。在 SYSINIT I 工作时, 先检查是否支持高容量软盘驱动器, 若不支持, 就把此表记录下来的所有与此有关指令改为 NOP (空操作) 指令, 该表建立在 0070: 2A9AH 处, 共 1EH 个字节。其结构如下:

+00H	+02H	+04H	+06H	+1CH
始址 1	有关指令占 用字节数 1	始址 2	有关指令占 用字节数 2	..... 结束标志 0000H

## 七、为系统配置而设置的数据结构

## 1. 设备参数区 (DPA: Device Parameter Area)

SYSINIT II 根据系统配置创建 OS 运行环境的过程中, 将读取或重置某些块设备参数, 为了暂存这些参数, IBMBIO 在内存首址为 CS: 1CF9H 处专门开辟了长度为

124H 个 Byte 的设备参数区其结构如下:

偏移(Byte)	长度(Byte)	字 段
+00H	01H	特殊功能标志
+01H	01H	设备类型
+02H	02H	设备属性字
+04H	02H	柱面数
+06H	01H	介质类型
+07H	0DH	系统 BPB
+14H	110H	磁道格式

其中各字段含义如下:

#### 1) 特殊功能标志字段

第 0 位: 为 1 时块设备当前 BPB 有效, 为 0 时系统 BPB 字段为新的有效的 BPB;

第 1 位: 为 1 时 DPB 中仅“磁道格式”字段有效, 为 0 时要求设备驱动程序在 DPA 中建立有关的设备信息;

第 2 位: 为 1 时同一道中所有扇区规格相同, 为 0 时可不同;

第 3~7 位: 值为 0, 保留。

#### 2) 设备类型字段以不同数值代表不同的块设备类型:

00H: 代表 5.25 英寸, 320/360KB 软驱

01H: 代表 5.25 英寸, 1.2MB 软驱

02H: 代表 5.25 英寸, 720KB 软驱

03H: 代表 8 英寸, 单密软驱

04H: 代表 8 英寸, 双密软驱

05H: 代表硬盘

06H: 代表磁带驱动器

07H: 代表其他类型块设备。

#### 3) 设备属性字段各位含义如下:

00H: 值为 1/0 表示可移动/不可移动介质;

01H: 值为 1/0 表示能/不能检测介质是否改变;

02~0FH: 值为 0, 保留

#### 4) 柱面数字段指出该块设备所支持的最大柱面数。

5) 介质类型字段指出所支持的多种规格介质的类型, 例如: 该值为 00H 表明介质是四密度, 1.2MB 软盘, 为 01H 表明介质是双密度。

6) 系统 BPB 字段根据特殊功能字段位 0 的值为 0, 表明这是新的有效的 BPB, 如该值为 1, 则块设备驱动程序将这个当前的 BPB 与引导记录盘的系统 BPB 比较, 以判别介质是否有改变。

7) 磁道格式字段是一张描述指定磁道规格、长度可变的表, 该字段最长可达 110HByte, 最多可支持 44H 个扇区/道, 其结构如下:

偏移+00H	每道扇区数 (n)
+02H	首扇区号
+04H	首扇区含字节数
+06H	次扇区号
+08H	次扇区含字节数
+0AH	...
(n-1)*4+2	末扇区号

$n * 4$  末扇区含字节数

#### 2. 关键字表 (KT: Key Table)

SYSINIT II 在解释 CONFIG 命令时, 为简化起见, 需用关键字标识符来取代正文中的配置命令, 故需设置关键字表, 当 BIO 装入内存后, 该表位于内存 CS: 1CA6H, 共 53H 个 Byte。该表包括十个表项, 每一表项结构如下:

偏移+00H	+01H	+n
关键字字符串长	关键字字符串	关键字标识符

该表最后以 00H (1 个 Byte) 为结束标志

#### 八、其他重要的数据结构

##### 1. 错误代码转换表

在块设备驱动程序调用 INT 13H 进行磁盘 I/O 操作出错时, AH 寄存器返回错误状态字节, 要将其转换成请求标题中的返回状态字段的错误代码, 需要设置一个错误代码转换表, 该表实际是由错误状态字表和错误代码表两部分组成, 长度各为 8 个 Byte。前者首址在 0070: 0271H, 存放中断 13H 返回的 7 种特定的错误状态, 后者首址在 0070: 0279H, 前 7 个字节存放与错误状态字节表相应的错误状态, 对于错误状态字节表中未列出的错误状态, 错误代码表一律视为“一般性代码” (错误代码 0CH), 存入在该表最后一个字节。错误代码转换表结构如下:

错误状态字节表

偏移+00H	+01H	+02H	+06H	+07H
错误状态	错误状态	.....	错误状态	当前错误状态
1	2		7	

错误状态代码表

+08H	+09H	+0AH	+0EH	+0FH
错误代码	错误代码	.....	错误代码	一般性错误代码
1	2		7	

##### 2. 月份天数转换表

IBMBIO 为进行日期转换, 需要用到每月的天数, 为此在 0070: 08B3H 开始的 0CH 个字节, 保存有 1~12 月份天数, 在进行天数转换时, 根据月份 i, 查出该月的天数, 其 i 月份天数地址公式为:

$$i \text{ 月份天数单元地址} = \text{转换表首址} + (i-1)$$

##### 3. 每月天数累加表

BIO 在进行由日期求总天数的转换中, 需设置每月天数累加表, 该表在内存 0070: 1B3CH, 长度为 18H 个字节, 每个表项占 2 个字节, 用以存放从当年 (非闰年) 1 月 1 日到当月 1 日 (不含该日) 的累计天数, 计算 i 月份 1 日之前的当年累计天数公式是:

$$\text{累计天数单元地址} = \text{累加表首址} + (i-1) * 2$$

以上介绍了 IBMBIO 所建立的各种表格型的数据结构, 文中所提及的地址均指 3.3 版, 对于表 3 中为设备驱动程序而设置的数据中的命令代码入口地址表和列表设备循环检测次数表, 将在后面文章中再作介绍。表 3 中为系

# 棒极了, DESK-LINK

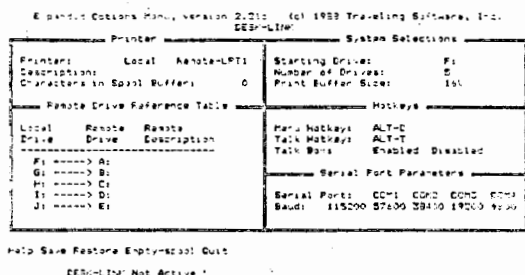
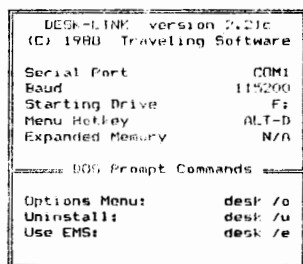
武汉大学计科系 雷军 张彤

DESK-LINK v2.21C 是由 Traveling Software 公司 1988 年出品的优秀双机通讯软件。

你和朋友们一起上机,如果你需要使用另一台机器硬盘中的某套软件,不幸的是,一个朋友正用那台机器,你不得不打断他的工作,用软盘把软件拷贝一份,再 COPY 到自己的硬盘中。如果你拥有 DESK-LINK,你可以在不打断朋友工作的同时,直接使用他硬盘中的软件,或者把软件拷贝到自己的硬盘中。

拥有 DESK-LINK,可以使很多问题变得简单。你不用购买打印机共享器,就可以和别人共享一台打印机。你不用装多用户软件或网络软件,就可以实时发邮件给另一台机器。DESK-LINK 具备一个小型点对点网络功能,而且操作简单,占用内存少(该程序约 46K)。

DESK-LINK 还能帮你解决很多困难。如 PS/2 上只有 3.5 寸"软驱,如果使用 DESK-LINK,把 PS/2 与 PC 机连起来,就可直接使用 5 寸"盘上的数据。如果你的软驱坏了,可以和别人共享软驱……



DESK-LINK 使用很简单。先把 RS-232 接口线插在两台机器的通讯口上 (COM1 或 COM2), 再在两台机器上运行 DESK。这样, 双机可在后台方式下共享软硬盘、打印机, 还可发送邮件。

DESK /o 显示状态菜单

DESK /u 撤除 DESK-LINK, 释放其所占内存

DESK /e 使用 EMS

热键 ALT-D, 按 ALT-D 后, 会弹出状态菜单, 按 'M' 键便会显示全部的设置、状态等。

如果通讯不能成功, 先检查线路, 无误的情况下再看看设置是否正确。Serial Port 项中有 COM1、COM2、COM3、COM4。如果你把线插在 COM1 口上, 选 COM1 项。双机须选用相同的 Baud 值。如果设置正确, 一般来说, DESK-LINK 就可正常工作。

## 共享磁盘

可把对方的软硬盘作为自己的逻辑盘。如把对方的 A 盘作为自己的 F 盘, 访问 F 盘时实际上是在访问对方的 A 盘。Starting Drive 用来设置对方驱动器作为本地逻辑盘的起始盘号; Number of Drives 显示可作为逻辑盘的远端驱动器数。

## 共享打印机

Printer 项中按左、右光标键去选择一个打印机。如果你想使用与自己机器连接的打印机, 选 LOCAL; 如果你想使用远程打印机, 就选择 Remote-LPT 三种中的一个。

Characters in Spool Buffer 可显示你送到远程打印机时放在后台缓冲区中的字符数。

Print Buffer Size 可设置缓冲区大小。

## 邮件 (Talk)

ALT-T 是 Talk 热键。按 ALT-T, 弹出一个窗口, 两人可在窗口中交谈。Talk Box 中有 Enable Disabled 两项, 选 Disabled 禁止对方中断自己的工作进入交谈方式。

热键可重新定义。

下面还有一排菜单。

Help Save Restore Empty-Spool Quit

按 H 得到 Help, 按 S 把设置存起来, 按 R 恢复设置为初始方式。E 清 Spool 后台打印缓冲区。

这么小的程序有这么多的功能, 不能不使人赞誉:

棒极了!

(软件库编号: IT123 50 元/片)

统配置而设置的 DRIVPARM 配置参数搜索表因所使用范围较小, 提示信息表因结构简单 (即为提示信息串), 本文限于篇幅就不作介绍了。

## 弹出式和下拉式菜单程序包的设计

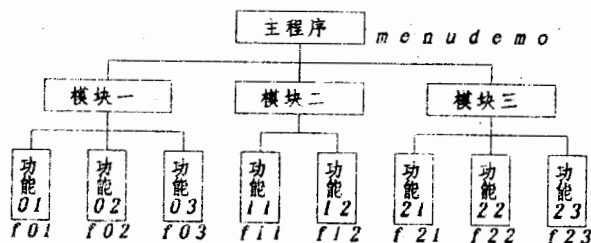
中山大学计算机系 董祖明

C语言因其强大的功能而日益受到广大程序开发人员的青睐。C语言表达能力强,可取代汇编语言处理低级运算,又有Pascal的良好模块化结构,能写出高质、高效的目标代码,而且具有较好的可移植性。

友好的用户接口是一个软件成功的必要条件之一。这也是软件开发中费力很大的一部分。而简明直观、使用方便的下拉式菜单和弹出式菜单,受到用户的普遍欢迎。本文为C程序员和C爱好者介绍一个菜单接口程序包的用法和实现技术(用Quick C实现)。

## 一.应用实例

设有一应用系统,其主要功能如下图所示。



图一 功能模块结构图

为简明起见,假设各功能模块函数均为void f(void)型的。其下拉式菜单实现程序如下。程序中类型为struct menuconfig的变量mainmenu和submenu定义了主菜单和子菜单的名称、左上角行列位置及菜单项的内容。函数menubar()的menu()功能是分别进入主菜单和子菜单,并返回选择的菜单项序号,或在按ESC键退出时返回-1。quitmenu()的作用是恢复菜单所覆盖的背景并释放保存背景的缓冲区。

```

/* menudemo.c menu usage demonstration */
#include "menu.h"
typedef void (* VFVTYPE)(void);
struct menuconfig mainmenu = {
    "Main Menu", 1,1,0, {"1.Module 0","2.Module 1","3.Module 2"}
};
struct menuconfig submenu[3] = {
    {"Module 0", 2, 1,0, {"1.function 1","2.function 2","3.function 3"}},
    {"Module 1", 2,12,0, {"1.function 1","2.function 2"}},
    {"Module 2", 2,23,0, {"1.function 1","2.function 2","3.function 3"}}
};

```

```

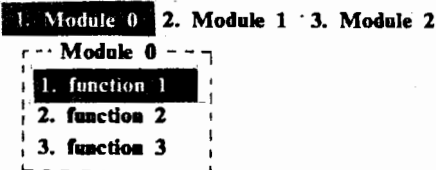
};
VFVTYPE func[3][3] = {
    {f01, f02, f03}, {f11, f12}, {f21, f22, f23}
};
void main(void)
{ int sub;
  for(;;){
    if(menubar(& mainmenu) == -1) { /* quit mainmenu */
      quitmenu(&mainmenu);
      return;
    }else{
      sub = menu(& submenu[mainmenu.selected]); /* enter submenu */
      quitmenu(&submenu[mainmenu.selected]);
      if(sub != -1){
        quitmenu(&mainmenu);
        func[mainmenu.selected][sub]();
      }
    }
  }
}
/* end of menudemo */

```

这段C程序比实现相同功能的FoxPlus+2.1程序还要短小精悍。用户可以用箭头和回车键或各菜单项的首字母“点菜”。菜单的外观类似Quick C和Turbo C的环境。读者可用以下命令编译并与菜单程序库连接:

QCL menudemo.c -link menus.lib

漂亮友好的菜单接口会让你的程序更有“专家水准”,更容易被用户接受。运行程序menudemo所显示的画面如下图所示:



图二 menudemo 示范程序

另外,这个菜单程序包还提供了一种程序与数据的独立性,也就是说,当你增加一个模块,又或者在某一模块中增加一个子功能时,只要修改menu和func数组即可,而主程序可原封不动。



## 二.实现方法

下面给出实现此菜单程序包的全部源程序,包括 menu.h 和 menu.c 两个文件。头文件 menu.h 定义了用户可见的宏、类型、系统全局变量和函数原型。类型 menuattr 定义菜单的菜单项、当前项和边框的前景背颜色(black, blue, green, cyan, red, magenta, yellow, white 及其高亮度值依次为 0 到 15), 以及组成边框的 8 个字符(从左上角起顺时针排列, SINGLE 和 DOUBLE 是预定义的单线和双线框)。菜单函数使用变量 \_\_menuattr 中的属性, 用户可通过对 \_\_menuattr 赋值来改变属性。类型 menuconfig 描述了一个菜单的名称、位置、当前项序号、各菜单项及背景保存区指针。

为了在退出一级菜单时能恢复被覆盖的背景, 利用 DOS 中断 INT 10 的子功能 8 和 9 分别读取当前光标位置的字符与其属性和在当前光标位置显示字符与属性, 见 m\_getch() 和 m\_putch()。函数 m\_getimage() 和 m\_putimage() 用于保存和恢复屏幕上一个方形区域。m\_putstr()、box()、titlebox() 用于显示菜单。menubar() 的功能是弹出一个横条形的菜单, 而 menu() 则是弹出一个方形的菜单。quitmenu() 与 menu() 和 menubar() 配对使用, 作用是恢复菜单后的背景, 若调用 menu() 或 menubar() 后不调用 quitmenu(), 则再进入相应菜单时不重画。源程序中有详细的注释, 不言自明。

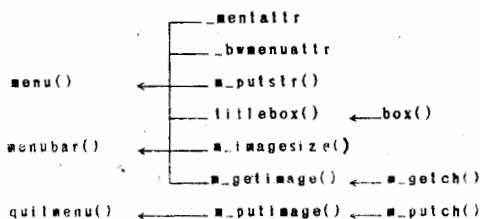
编译和连接方法如下 (以小模式为例):

QCL -c menu.c (编译为 menu.obj 文件)

LIB menus +menu (做成 menus.lib 库)

在应用程序中加入 #include "menu.h" 就可使用 menu() 和 menubar() 了。并且简单地组合它们可开发出代码简单但功能强大的菜单接口。

程序中各个过程(函数)之间的关系如下图所示:



图三 函数、全程变量之间的关系

全部源程序文件名为 ZA13S.PRg 拷入本期源程序盘 (编号为 P91-3) 中。邮购事宜参看第 28 页中小启

## 硬盘安全分区的实现

中国科技大学精密仪器系 87 级 梁文全

在硬盘上形成一个以上 DOS 分区, 通常的方法是使用硬盘管理程序, 如 ADM 等。但那些程序在实行分区时, 都将引起硬盘上数据的丢失, 因而对数据是不安全的。通过对 DOS 操作系统, 特别是 IBMBIO.COM 的分析, 我们可以用一简单的方法实现硬盘的再分区 (假定原来整个硬盘已格式化为 DOS 分区), 它不损害硬盘上的原有数据, 因而对数据是安全的。它只要求用户即将要建立分区的盘空间连续、空白, 这是容易做到的。

DOS 操作系统在确认硬盘时是以能否建立该盘的 BPB (基本输入输出参数块) 表为依据的, 在系统引导时 IBMBIO.COM 以 AH=8H 调用 BIOS 中断 13H 检测在内存 0000:0475H 的硬盘计数字节 HF\_NUM 以确定系统中安装的硬盘驱动器数。从经济角度出发, 大多数用户一般在微机中只安装一台硬盘驱动器, 即只有一个硬盘, 硬盘计数字节 HF\_NUM 值为 01H, 而实际中 DOS 操作系统是可以管理二个硬盘, 因而 IBMBIO.COM 将最多可为两台硬盘建立 BPB 表。IBMBIO.COM 在为硬盘建 BPB 表时若 HF\_NUM 为 00H, 则认为系统不存在硬盘, 跳过为硬盘建 BPB-B 表的程序; 若 HF\_NUM 值为 01H, 则以驱动器号 (BIOS 中断 13H 中的入口参数) 80H 建立 BPB 表; 若 HF\_NUM 为 02H, 则分别以驱动器号 80H 和 81H 建立 BPB 表, 以后 IBMDOS.COM 依 BPB 表为硬盘建立 DPB (磁盘参数块)。在 BPB 和 DPB 都正确建立以后, 操作系统也就承认了硬盘。

硬盘的安全分区就是从上述原理出发, 对机器中的唯一的硬盘实行分区, 以形成两个 DOS 分区。其方法就是在 IBMBIO.COM 在测试 HF\_NUM 字节前使其增 1, 同时修改 BIOS 中断 13H。在 IBMBIO.COM 使用 BIOS 中断 13H 以驱动器号 81H 读取硬盘分区信息表 (此时 IBMBIO.COM 设定入口参数为 AX=0201H, BX=0000H, CX=0001H, DH=00H, DL=80H 或 81H 时过滤到指定的存有第二分区的分区信息表的扇区中, 以便 IBMBIO.COM 能取得正确的分区信息, 从而能正确的建立 BPB 表。注意: 两个 DOS 分区的分区信息表不能放在同一扇区中, 否则只有第一个分区信息项发挥作用。IBMBIO.COM 在检测到有效的 DOS 分区 (分区信息项的第五个字节为 01H (对应 FAT 为 12 位) 或为 04H (对应 FAT 为 16 位)) 的分区项后将不再继续测试。

因必须在 IBMBIO.COM 测试 HF\_NUM 前完成 HF\_NUM 增 1 和修改 INT 13H 的任务, 因而必须修改主引导记录。简便的方法是将主引导记录移位, 把替换的程序放在引导扇区中, 则系统上电或复位时, 该程序首先取得控制, 完成上述任务, 程序的最后应是读入主引导记

录,并将控制权交给它。下面就以例子说明。

例: IBM PC/XT 机中 20MB 硬盘,在第二次分区前已格式化为单一 DOS 分区,剩余自由盘空间为 7.6MB,现在从中分出 3.5MB,形成第二 DOS 分区 D 盘, D 盘起始扇区为 0 头 512 柱面 1 扇区,结束扇区为 3 头 613 柱面 17 扇区。修改分区信息表如下:

第一 DOS 分区的分区信息项为: 80 01 01 00 04 03 51 FF 11 00 00 00 EF 87 00 00

第二 DOS 分区的分区信息项为: 00 00 81 00 01 03 91 65 00 88 00 00 18 1B 00 00

对第一 DOS 分区的分区信息项,它是供 IBMIO.COM 以 DL=80H 时测试用,可把它放在主引导扇区中,则此时 (DL=80H) INT 13H 的入口参数不用改动。

对第二 DOS 分区的分区信息项,它是供 IBMIO.COM 以 DL=81H 时测试用,由于实际中对硬盘号 81H 的硬盘不存在,因而必须修改以 DL=81H 发出的 INT 13H 的入口参数,使其指向硬盘 80H 的相关扇区。一般可使以入口参数 DX=0081H, CX=0001H 发出的 INT 13H 指向第二 DOS 分区的起始扇区为宜,即上述的第二 DOS 分区的分区信息项放在第二 DOS 分区的起始扇区上。

对应上例,替换程序 D\_\_INSTAL.ASM 如下:

```
D>type d__instal.asm
ABSO segment at 0
    org 413h
memory_size label word; 内存大小标志字
    org 475h
HF_NUM label byte; 硬盘计数字节
    org 13h*4
disk_vector label word; 中断 13H 的入口地址
    org 7c00h
boot_record label far; 自举入口标号
ABSO ends
;.....
```

```
CODE segment para public
    assume cs: CODE, ds: ABSO, es: CODE
    org 7c00h
start: mov ax, c5; 替换程序的开始
    mov ds, ax
    mov es, ax
    mov si, offset begin
    mov di, 7c00h
    mov cx, 100h
    rep movsw; 自身转移
    mov cx, 7c00h
    jmp cx
begin: xor ax, ax
    mov ds, ax
    mov ax, memory_size
    sub ax, 1; 内存减 1KB, 为驻留程序开劈空间
    mov memory_size, ax
    inc HF_NUM; 硬盘计数字节增 1
    mov cl, 6
    shl ax, cl
```

```
mov -cs, ax; 形成高位留段地址
mov si, offset remain
xor di, di
mov cx, 20h
rep movsb; 转移留部分
xor ax, ax
mov -ds, ax
mov al, 0EAh
stosb
mov ax, disk_vector
stoaw
mov ax, disk_vector+2
stosw; 设置 JMP *****: *****指令
mov disk_vector, 00h; 修改 INT 13H 的入口
mov disk_vector+2, es
xor ax, ax
mov es, ax
mov -cx, 3; 主引导记录在 0 头 0 柱面 3 扇区
mov -ax, 0201h
mov -bx, 7c00h
mov dx, 0080h
int 13h; 读取原硬盘主引导记录
jmp boot_record; 将控制权交给主引导记录
;.....
```

```
remain: cmp di, 81h; 留部分
    jne no_change
    mov di, 80h; 将 D 盘的 0 头 0 柱面 1 扇区过滤到 C 盘
    add ch, 00h; 的 0 头 512 柱面 1 扇区
    or cl, 80h
no_change: nop
    nop
    db 20 dup (90h); 以空操作填充
CODE ends
end start
```

将此程序汇编、连接而成 EXE 文件。利用 DEBUG.COM 将其读入,检测程序的结束,将 BX 清零, CX 置为结束地址的偏移量,以 D\_\_INSTAL.SYS 文件存盘,然后读入硬盘主引导记录,将其写到 0 头 0 柱面 3 扇区中(这是为适应上面的程序而做的,只要修改程序,则其扇区可灵活安排)。

读入 D\_\_INSTAL.SYS (DEBUG.COM 把它当作 COM 文件形式读入),在偏移地址 02BEH 处用 E 命令分别写入第一 DOS 分区项和第二 DOS 分区项。再将其相应地写到 0 头 0 柱面 1 扇区和 0 头 512 柱面 1 扇区 (DL 都为 80H) 中。对写入 0 头 512 柱面 1 扇区中的内容,应同时写到 0 头 0 柱面 2 扇区中作备份,因格式化 D 盘 (第二 DOS 分区) 时该扇区内容会丢失。现在,可以重新从硬盘起动系统,则系统已承认 D 盘。

现在 D 盘还不能使用,利用 PCTOOLS 工具软件,你会发现 D 盘没有空间, D 盘的文件分配表已被填满,根目录也一样,因而必须格式化 D 盘,可用 FORMAT.COM 软件完成。格式化完成后,再将 0 头 0 柱面 2 扇区中备份的第二 DOS 分区信息项拷贝回来,则以后只要硬盘起动, D 盘与 C 盘并存。

上述程序仅仅是为了说明实行分区和方法。利用上述

# 分页符号的消除二法

北京理工大学 71891 班 杨海军

汉化后的 Wordstar 有很强字处理功能, 因而深受广大计算机用户的喜爱, 它不仅被用于一般报表文章的编排, 也常被用于源程序的编辑。在 WS 环境中用 P 命令输出的表格也能很好地符合要求, 用 D 命令编辑过的一些源程序 (如 dBASE III) 也可以顺利地运行。

但是, 如果退出 WS 再向打印机输出这些用 D 命令编制的报表, 就往往不那么让人满意了。笔者曾用 WS 编制一份表格, 在 WS 中用 P 命令输出正常, 而后来在 CCDOS 环境下直接用 "TYPE  $\times \times \times > PRN$ " 命令向打印机输出则发生了混乱: 一些行被接到了上一行尾部, 有些行挤到了一块。另外, 用 WS 编辑过的源程序, 编译时也常出现一些莫名其妙的错误, 如 "语句太长", "不可识别的命令", 等等, 而重新查看有错的语句, 却发现一切正常。

这是怎么回事呢? 原来在 WS 中用 D 命令编辑过的文件, 每隔 55 行就被自动加上一个分页符号, 有时是隐藏的, 有时是可显示的 (分页符的隐显可通过键入 CTRL+OP 来控制), WS 系统凭借它来把整个文件分成许多

多页, 在用 P 命令输出时系统根据它把文件分页输出并把该符号作为内部命令滤掉。而一旦退出 WS, 分页符号往往被当作续行符或其它控制字符, 因此往往在第 NX55 行附近引起输出混乱。

消除分页符号的方法很多, 这里介绍两种简便可行的方法:

1、在进行字处理时, 每遇到分页符便插入一个空行, 即在第 NX55 行之后加入一个空行 (用 CTRL+N 命令或 Insert on 状态下的回车符均可)。这样, 尽管分页符仍然存在, 但分页符下面紧接的一行是空行, 即使系统把分页符作为续行符处理也不妨碍输出效果或程序运行。

2、在 WS 下, 用 P 命令, 当系统提问:

"输出到磁盘否?" 输入一个 Y, 随后系统提问:

"文件名:" 此时输入一个新的文件名。当系统提问: "自动换页否?" 打 N, "消除分页符?" 打 Y, 随后会发觉驱动器的灯开始亮, 等系统给出 "打印完毕" 信息时, 便可把老文件删除, 这样磁盘上已经有了一个新文件, 内容与被删掉的相同, 只是没有了分页符。这个新文件只要不再经 WS 中的 D 命令编辑, 可以在任何环境下输出令人满意的结果。

这两种方法相比, 第二种较好些。有兴趣的同志, 不妨一试。

## 再谈加快按键连发频率

哈尔滨工业大学 87 级 徐东晖

九〇年第六期《电脑》杂志上介绍了一个小程序 FASTKEY, 可以加快连发键速度。其实, 在 AT 机上, 可以用两条指令定量地控制其连发频率, 不但如此, 同时还可控制从按下键开始到连发的延迟时间。

这两条指令是 OUT96, 243: OUT96, n, 其中 n 为控制参数, 在 Dcbug 下发送这两条指令的命令为 060F3-060n-  
060n-  
n 与延迟时间和重复频率的关系较复杂, 笔者特编一简

单对照表如下备查: (注意: 表中参数 n 为 10 进制)

重复频率 参数 n 延迟时间	5 次/秒	10 次/秒	20 次/秒	30 次/秒
1/4 秒	20	12	4	0
1/2 秒	52	44	36	32
3/4 秒	84	75	68	64
1 秒	116	108	100	96

(注: 缺省 n = 44)

从中我们可以知道, 延迟时间是 1/4 秒的整数倍, 从

1/4 ~ 1 秒变化, 而重复频率最高则为 30 次/秒。任何一个参数加减 32 的整数倍仅影响延迟时间。大家可以试试别的 n 值, 但是请注意, n 的取值范围为 0~127, 超出范围将引起不可预料后果。



# 由电阻电容引起的微机故障

华东工学院 陈文高 展凌

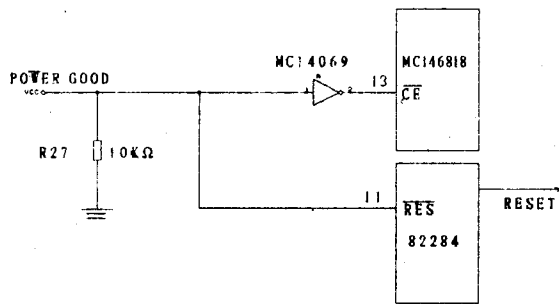
由电阻、电容的旁路作用或损坏引起的个人计算机的故障，往往不被人所注意，下面举几个故障实例。

## 一、由电阻旁路作用引起的故障实例：

故障现象：IBM XT/286 开机后，有时启动不了，出现死机现象。

故障分析与维修：

当 IBM XT/286 开机后，电源通过一段稳定时间，电源控制电路产生一个负脉冲  $\overline{\text{POWER GOOD}}$  信号，如图〈一〉所示，该脉冲接到时钟发生器的 82284 的  $\overline{\text{RES}}$  端，经过 82284 芯片的内部延迟和同步后，产生一个  $\overline{\text{RESET}}$  信号，把 CPU 的启动地址指向 FFFF0 单元，从而进入只读存储器，执行 BIOS 程序，开始对系统进行自检和初始化。同时， $\overline{\text{POWER GOOD}}$  信号又经过 MC14069 反相器接到日历时钟电路 CMOS RAM 芯片 MC146818 的芯片允许端  $\overline{\text{CE}}$ ，当 MC146818 存取总线周期时， $\overline{\text{CE}}$  必须是低电平有效。 $\overline{\text{POWER GOOD}}$  信号平时总是处于高电平状态，经过 MC14069 反相器后，满足  $\overline{\text{CE}}$  必须是低电平的条件。又同时在  $\overline{\text{POWER GOOD}}$  信号的输入端又并联了一个  $10\text{K}\Omega$  的电阻 R27。由此看来，IBM XT/286 的  $\overline{\text{POWER GOOD}}$  信号要提供的负载能力比起 IBM PC/XT 要大得多。因此，要求 IBM XT/286 的  $\overline{\text{POWER GOOD}}$  信号负载能力要大。因此，一旦由于器件老化使得  $\overline{\text{POWER GOOD}}$  信号负载能力不够，启动不了机器，就会出现死机现象。如果这时将并联电阻 R27 去掉，就可以减轻  $\overline{\text{POWER GOOD}}$  信号的负载，就有可能排除死机故障，我们有这种方法修复过机器。



图〈一〉

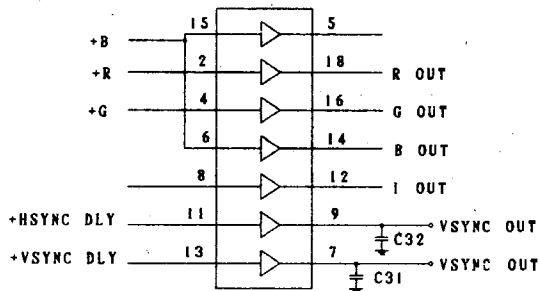
## 二、由电容的旁路作用引起的故障实例：

故障现象：640×200 彩色/图形显示卡，显示字符不能同步。

故障分析与维修：

CGA640×200 彩色/图形显示卡，在 PC 机中用得最多。图〈二〉中，HSYNC OVT 是送到彩色监视器作为水

平同步信号用，VSYNC OVT 是送到彩色监视器作为垂直同步信号用。用示波器检测这两个信号都有，说明由于长期工作后，器件老化，使得 HSYNC OVT 信号的幅值不足，不能使显示器进行正常的同步显示，就出现显示字符不同步。排除故障的方法之一是将 HSYNC OVT 旁路电容 C32 去掉，减少信号旁路，提高 HSYNC OVT 信号的幅值，就有可能排除故障。



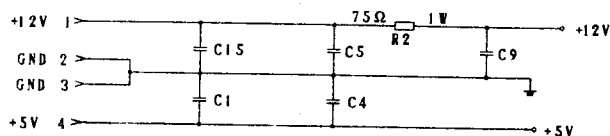
图〈二〉

## 三、由电阻损坏引起的故障实例：

故障现象：IBM PC/XT 硬盘驱动器不工作，并显示故障码 1701。

故障分析与维修：

硬盘驱动器需由主机提供给一个 +12V 的直流电源和一个 +5V 的直流电源才能工作，图〈三〉为硬盘驱动器外接电源的电路图。由于长期使用， $75\Omega$  的限流电阻 R2 失效，其阻值变为无穷大，等于开路，结果 +12V 的电源全部加在电阻 R2 上，相当于硬盘驱动器缺少 +12V 的电源，无法正常工作，而出现上述的故障现象。排除故障的方法是更换电阻 R2。

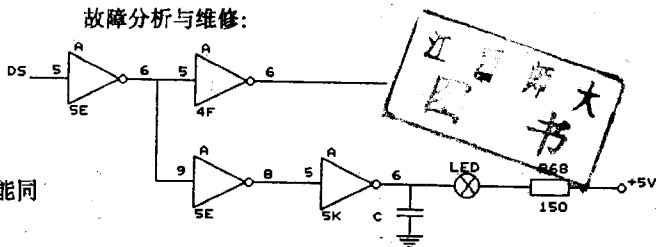


图〈三〉

## 四、由电容损坏引起的故障实例：

故障现象：IBM PC/XT 硬盘驱动器仍能正常工作，但硬盘指示灯总亮不灭。

故障分析与维修：



图〈四〉

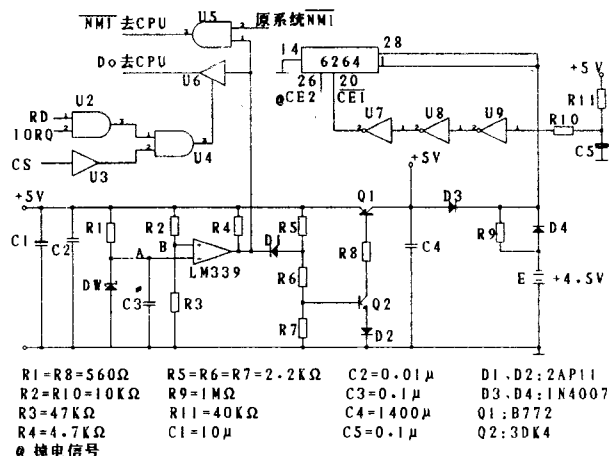
# 掉电保护及自动恢复实用电路和程序

重庆 肖明德

由于电力供应不足及其他一些原因,经常发生停电,这对于电加工来说,是特别头痛的事。目前大多数线切割机,是采用 Z80 系统的单板控制,在加工过程中,当突然发生停电时, RAM 之中的加工程序就消失了。再来电后,操作者又得重新输入加工程序,并且退回到原点,从头开始加工。不但多用了时间,而且对已加工过的轨迹来说,其尺寸精度必然受到影响。笔者设计了一个掉电保护电路,当停电时, RAM 中的加工程序可以完成无损的保护起来,当再来电时,楞以自动的从掉电时的加式点开始继续往下加工。现将该实用电路和相应的程序功能结构简述如下:

## 一、硬件结构:

硬件结构如图一所示。



图一

图中,由 R1, DW, C3, U1, D1 构成上电延时和掉电信号产生电路,为系统提供掉电中断和掉电测试信号。

R5, R6, R7, Q1, Q2, D2, R8 构成开关电路, Q1 是系统电源的开关管, Q2 是 Q1 的推动管, 单板机系统

当硬盘驱动器被选中时,反相器 5K 的输出端第 6 引脚为低电平,这时硬盘驱动器工作的指示灯 LED 亮,表示硬盘驱动器正在工作;当反相器 5K 的输出端第 6 引脚为高电平时,这时硬盘驱动器不工作,指示灯 LED 不亮。上述的故障现象硬盘驱动器能正常工作,但指示灯总亮不灭,这是故障出在指示灯电路上,如图〈四〉所示。经检查发现旁路电容 C 其绝缘电阻只有 28Ω,说明电容 C 已击穿,当去掉旁路电容 C 之后,故障排除,显示正常。

(除 6264 RAM 芯片外) 的电源由 D 点提供。U1 的电源不接 D 点,和 Q1 的输入端相联。

D3, D4 是主电源和辅助电源 E 的自动切换电路,当主电源供电时, D3 导通,向 6264 提供 +15V 电源,并通过 R9 向辅助电源 E 供电;当主电源消失, D4 导通, 6264 由辅助电源供电,维持 6264 中存贮的信息,起到保护数据的作用。这时 D3 截止, Q1 也截止,保证辅助电源 E 的电流不流向电源和系统其他电路元件。

U7, U8, U9, R10, R11, C5 构成 6264 的保护电路, U7 的输出端接到 6264 的片选信号脚 20 (CE1, 低有效), 当主电源供电时,通过 R11, R10 使 U7 输出低电平, 6264 的片选信号就由 26 脚 CE2 确定,当接系统的存储器地址译码时,就可对 6264 进行读写操作。U7, U8, U9 是一片 CMOS 的 4069, 它的电源是和 6264 共用的,当辅助电源供电时, U7 输出为高电平,封销 6264,起到保护作用。

U2, U3, U4, U5, U6 构成中断信号产生和掉电信号测试电路。掉电时, U1 输出低电平,通过 U5 输出一个负的中断请求信号,去 CPU 申请非屏蔽中断 (即  $\overline{NMI}$ ), 在中断服务程序中测试 U6 的输出,为“0”,则说明是掉电中断,若为“1”,则说明是原系统的非屏蔽中断。

将原系统的 RAM 芯片 (即 2114) 全部去掉,对原存储器地址译码电路稍加改造,就可用 8K 容量的 6264 芯片代替,使系统 RAM 容量扩大一倍。

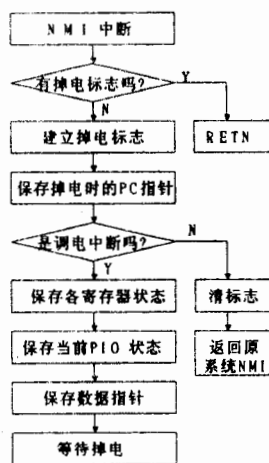
上电后, R1 与 DW 组成的欠压电路,输出 3.2V 的参考电压,经 R1 和 C3 组成的 RC 电路延时,使 U1 的两个输入端 A 和 B 有一定的相位差,保证 U1 的输出端为低电平, D1 处于导通状态, C 点钳位在低电平, Q1 是截止的。这保证不会发生系统在电源电压上升阶段,因电压不稳,总线信号不确定所造成的数据误写入。当电源上升, B 点电压超过 3.2V 时, U1 翻转,输出高电平, D1 截止, C 点为高电平, Q2 导通,推动 Q1 导通,经 D 点向系统提供 +5V 电源。 D3 导通, D4 截止,主电源向 6264, 4069 (即 U7, U8, U9) 供电,同时 U7 输出低电平,保证 6264 读写操作,主电源通过 R9 向辅助电源 E 充电。

当主电源掉电,电压低于 3.2V 时, U1 翻转,输出低电平, D1 导通, C 点钳位在低电平, Q2 截止, Q1 也随着截止,主机靠 C4 的电压完成保存数据的工作。在 U1 翻转时,产生一个负跳脉冲信号,经 U5 输出,向 CPU 提出非屏蔽中断请求, CPU 在中断服务程序中测试 U6 的输出,若是掉电中断,则将有有用数据转移,完成保护现场的工作。

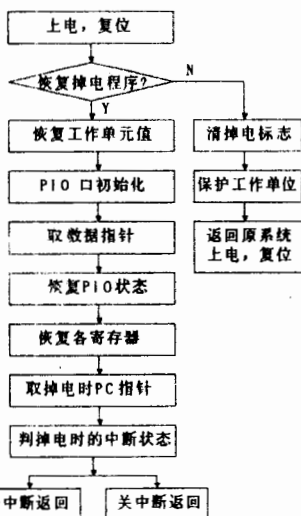
## 二、软件框图说明:



## 1. 掉电保护程序框图, 如图二所示。



图二



图三

掉电保护程序采用了 Z80CPU 的非屏蔽中断方式。这时中断不能被用户程序屏蔽, 也不受 CPU 是否在开中断状态的影响, 一旦有非屏蔽中断请求, CPU 立即响应。所以, 这种中断方式用于掉电这样紧急情况处理, 非常合适。Z80CPU 的非屏蔽中断入口为 0066H, 在该单元放一条无条件转移指令 (即 JPNMI), 转向非屏蔽中断服务程序。进入服务程序后, 首先检查掉电标志位, 这是防止多次请求, 在执行服务程序过程中, 又有请求信号, 如果发生, 无条件返回, 以防返回地址丢失。当第一次进入掉电中断服务程序时, 则建立一个标志位, 然后保存掉电时的 PC 内容, 即当时 SP 寄存器的内容。再测试是否为掉电中断, 若不是, 则为原系统的非屏蔽中断请求, 立即返回原系统的 NMI 中断; 是掉电中断, 则保存各寄存器状态, 保存当前 PIO 状态 (若有其他 IO 接口的状态需要保存的话, 也应保存), 然后保存数据指针, 即在栈完后的 SP 内容。最后进入等待状态, 等待主电源全部下降到自动切换为辅助电源供电, 以维持 RAM 之中的数据信息。

## 2. 上电自动恢复程序框图, 如图三所示。

上电, 或按复位键, CPU 必须从 0000H 单元开始执行程序。因此, 上电恢复掉电时的程序应从 0000H 单元转来 (也可用其他方法实现), 在系统程序的 0000H 单元放一条无条件转移指令 (即 JP REST), 转向自动恢复程序。

首先判别恢复掉电标志位, 若不恢复, 则清除在掉电中断服务程序中建立的掉电标志位, 以便掉电时进入中断服务程序。然后将有用的工作单位数据转移, 在恢复时使用, 再进入原系统监控程序。若要想恢复掉电时的程序, 则在原系统监控状态下, 将恢复掉电程序标志位置位, 按复位键, 则系统又从 0000H 单元开始执行, 再次转

入自动恢复程序, 再判恢复掉电程序标志位时, 则为“真”, 即要恢复掉电程序。程序往下走, 恢复工作单元, PIO 口初始化 (若有其他 I/O 口需要初始化, 也要完成初始化工作); 取数据指针, 就是取中断服务程序中保存数据时, 最后压栈的数据指针, 放入 SP 寄存器, 数据 Z80-CPU 压栈后进先出的原则, 恢复 PIO 状态, 恢复各寄存器状态, 最后恢复掉电时的 PC 指针, 即掉电时 SP 的内容, 然后判 CPU 掉电时可屏蔽中断的状态, 若当时为开中断状态, 则开中断返回, 否则不关中断返回, 即返回到掉电时的程序状态, 继续往下执行。

## 三. 程序清单及注释:

## 1. 非屏蔽中断服务程序清单:

```

FSNMI: PUSH HL;
LD HL, 3FD5H; 掉电标志单元号送 HL
BIT 0, (HL); 判零位
JR Z, NMI; 无标志转 NMI
POP HL; 有标志返回
RETN;

NMI: SET 0, (HL); 标志单元零位置 1
POP HL;
LD (3FF3H), SP; 保存掉电时 PC 指针
LD SP, 3FF0H; 数据单元首址送 SP
PUSH AF;
IN A, (40H); 取掉电信号
BIT 0, A; 是掉电中断吗?
JR NE, NM3; 不是, 转 NM3
PUSH BC; 是, 则保存寄存器内容
PUSH DE;
PUSH HL;
LD A, I; 保存 I 内容及中断状态
PUSH AF;
EX AF, AF';
EXX;
PUSH AF; 保存辅助寄存器内容
PUSH BC;
PUSH DE;
PUSH HL;
PUSH IX;
PUSH IY;
IN A, (80H); 保存 PIO A 口状态
PUSH AF;
IN A, (81H); 保存 PIO B 口状态
PUSH AF;
LD (3FFCH), SP; 保存数据指针
NM2: HALT; 等待掉电
NOP;
NOP;
JR NM2;
NM3: XOR A; 不是掉电中断
LD (3FD5H), A; 恢复寄存器原内容
POP AF;
LD SP, (3FF3H);
PUSH HL;
LD HL, 2ED2H; 此条为原系统 0066H 单元指令
JP 006AH;
  
```

## 2. 自动恢复程序清单:

```

REST: NOP
LD A, (3FD4H); 取恢复程序标志
CP 0FFH; 要恢复吗?
  
```

# 计算机辅助教学课件设计

华南师范大学微电所 龙庆华 金惠生  
任平 司徒锡康

## 引言

计算机辅助教学 (computer-aided Instruction) 简称 CAI。它是一种以计算机为中心的自动化教学技术。用计算机展示教学计划内容给学习者, 并通过学习者与计算机之间的相互作用完成教学功能。有时也称之为计算机辅助学习 (computer-Assisted Learning) CAL。它改变了传统的以课堂教学为主, 学生被动接受的局面。对多个学生同时进行个别教学。国内外实践证明, 通过计算机辅助教学系统对学生实行因人施教, 可以收到事半功倍的效果。因此计算机辅助教学系统的出现被称之为“第四次教育革命”, 因为它打破了传统的以“传授知识”为指导思想的“三中心论”(教师、课堂、书本) 而利用现代化教育技术实现以学生为中心的开放式个别化教学。特别是计算机技术及信息技术迅猛发展的今天, 计算机已进入到社会的各个领域, 并进入到家庭。这给计算机教育的发展提供了非常有利的条件。

计算机辅助教学系统主要依赖于计算机的硬件环境, 支持软件及教学软件。硬件方面, 目前国内中小学大多以中华学习机及苹果机系列为主, 而大专院校则以 IBM-PC

系列或兼容机较普遍。多数微机系统没有语音输出功能, 音响输出只利用原系统的发音装置—喇叭。教学软件方面, 也以完成个别教学环节(一个定律的演示)较多。整门课程的教学软件还不多见。我们在上述两个方面都做了新的尝试。我们在 IBM PC 系列微机上建立了一套文语转换系统, 在“支撑软件”的支持下可以完成汉字、日、英、俄文字母及若干科学符号的普通话或广东话的发音。并能自由控制计算机的显示及发声。在此基础上, 我们将“微型计算机原理及应用”教材, 改编成计算机辅助教学课件, 收到较好的效果。

## 计算机辅助教学课件设计

计算机教学软件 (Instructional Software) 或课件 (courseware) 是把教学计划内容在计算机上表现出来, 它不是简单的课程教材的复制, 而是要以计算机程序的形式, 充分利用计算机的表现功能把教学内容重新组织, 以简短的形式表达出深刻的内涵。它利用计算机的特点完成课堂教学所不能很好完成的功能。通过人机对话给学习者以自由选择的机会, 做到个别化教学, 因人施教。一般教学课件包括授课、模拟(表演)、练习、问答、测验及游戏

```
JR    NZ, NGT;      不恢复, 转 NGT
XOR   A;
LD     (3FD4H), A;   清恢复标志
LD     HL, 3DD0H;    恢复工作单元
LD     DE, 2E00H;
LD     BC, 01FFH;
LDIR;
CALL   0573H;        PIO 初始化
CALL   058CH;
LD     SP, (3FF1H),  取数据单元指针
POP    AF;
OUT    (81H), A;      恢复 PIO B 口状态
POP    AF;
OUT    (80H), A;      恢复 PIO A 口状态
POP    IY;
POP    IX;
POP    HL;            恢复辅助寄存器
POP    DE;
POP    BC;
POP    AF;
EXX;
EX     AF, AF';
POP    AF;            恢复 I 寄存器及中断状态
LD     I, A;
POP    HL;            恢复主寄存器
POP    DE;
POP    BC;
LD     A, (3FE6H);    判掉电前中状态
```

```
AND   04H;
OR    A
JR     NE, BN1;       掉电前是开中断状态, 转 BN1
POP    AF;
LD     SP, (3FF3H);    恢复掉电时 PC 指针
DI;
      关中断返回
RET;
BN1:   POP    AF;
LD     SP, (3FF3H);    恢复掉电时 PC 指针
EI;
      开中断返回
RET;
NGT:   XOR    A;
LD     (3FD5H), A;     清掉电标志
LD     HL, 2E00H;      转移工作单元数据
LD     DE, 3DD0H;
LD     BC, 01FFH;
LDIR;
LD     SP, 2E58H;      此条为原系统 0000H 单指令
JP     0003H;          返回原系统监控程序
```

## 四、结尾:

该掉电保护电路结构简单, 不改变原系统的操作方式, 与不影响原系统功能, 可以完成掉电保护, 上电也可以自动恢复, 在我厂几台线切割数控机床上应用, 操作者非常满意。辅助电源采用三节五号电池串联使用, 实际测得维持电流为 60 $\mu$ A, 按 0.1 安/时计算, 维持 6264 中的数据可达 1600 多个小时。 ■

等, 并有各种不同的表现形式及方法。“微型计算机原理及应用”是大专院校有关电子技术, 自动化或计算机方面不可缺少的课程之一, 此课实用性很强, 并且涉及面较广。初始接触似乎不难, 但深入掌握又不太容易。主要是顺序的概念, 时间的概念掌握较为困难, 课堂讲授图形、程序、时间的对照繁锁并难以表现, 而计算机就能较容易的实现并能生动地表达出时间变化的关系, 因此我们完成了此门课程的数个章节的程序编制, 并进行试教, 其反应是较好的。

### 1. 硬件环境及软件支撑:

我们建立的 CAI 系统框图如图 1。在 IBM PC (内存大于 512K) 系列微机的 PC 总线扩展槽内, 加入了 K<sub>x</sub>-1 型共振峰语音合成器板, 并连接声音输出装置。

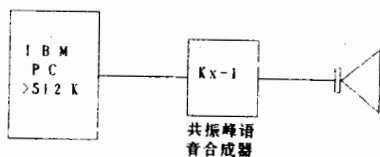


图 1 CAI 系统框图

其“支撑软件”是原 PC 操作系统。在 DOS 下启动汉字操作系统 CCDOS (版本不限) 并启动语音管理软件, 以控制文语转换。由于语音管理软件支持各种高级语言, 课件可用不同的语言编写, 我们采用编译 BASIC 编写。课件也可以在没有语音输出的 IBM PC 系列微机上运行, 只是没有语音输出。

### 2. 设计原则:

- 丰富的内容: 课件除应包含课本要求的讲授内容之外, 还应有供学习者自我检测的问答、测验、练习、总结等, 并有丰富的表现手法以取得良好的收效。

- 引人入胜: 课件要能适合不同“口味”或“水平”的学习者并能引起他们的兴趣, 课件的编写就要引人入胜, 有精彩的“包装”, 生动的讲解及图文并茂的表现形式。

- 自由选择: 课件面向不同要求的学习者, 千篇一律从头至尾的讲授是不适宜的, 应能发挥每一个学习者自身的主动性, 以取得最好的教学效果, 就需要课件有适当的提示, 内容的自由选择, 页面的自由前后翻动, 即友好的人机界面。

### 3. 实现方法:

#### • 模块化结构与链接技术

我们把课件内容分成若干章节, 每一章节又分解成若干独立的模块。根据整个教学软件的需要用链接的办法组织起来, 这样有利于编写也便于修改。在编译 BASIC 中 Chain 语句是不同于 PC BASIC 的, 它不能带有参数, 为此我们在程序中设置了一些软开关 (以控制程序的转向) 用参数传递的方法由一个模块传给另一个模块再进行链接。例如模块“Firstchl-1”与“Firstchl-2”连接。条件是参数 M\$ = “RET”则从“Firstchl-2”模块的第 100 句开始执行, 否则“Firstchl-2”从头开始执行。我们在“Firstchl-1”模块的开始处用 Common M\$ 语句, 并在链接语句前预

置 M\$ 的值, 然后链接“Firstchl-2”。在“Firstchl-2”模块开始处用 Common M\$ 语句接受参数 M\$, 并紧接着用判断语句 IF M\$ = “RET” Then goto 100, 实现对程序转向的控制。

在不需参数传递的情况下也可用下面语句实现模块的简单链接。

RUN “Firstchl-2”或 Chain “Firstchl-2”

#### • 窗口及动画技术:

为了实现“原理”的生动描述, 我们采用了窗口及动画技术相结合的表现形式。例如计算机工作过程部分, 我们用高分变率计算机图形具体说明数据在计算机各个部分的传送过程 (程序记数据、地址传送、存贮单元的寻找、数据向寄存器的传送、指令译码、运算器的工作及结果的回送等) 数据的每一次变动都在图形的相应位置上显示 (闪动标示) 并同时在屏幕上设置窗口做具体的文字说明, 并有一定的等待时间以供学习者观察思考。这种演示及讲解结合在一起的方式是传统课堂教学难以达到的。

在图形变换中有几个问题要加以说明:

图形清除: 在高分变图形方式工作, 简单用 CLS 语句不能把图形清除干净, 留下的残迹将影响屏幕的美观, 一般可采用两种方式清除。

(1) 用 Line 语句, 将底色充满所要清除的屏区。

LINE (X<sub>1</sub>, Y<sub>1</sub>) - (X<sub>2</sub>, Y<sub>2</sub>), 0, BF

(2) 用工作模式切换方法, 即 SCREEN1 及 SCREEN2 切换一次再返回到原来的工作模式, 可实现整屏的清除。

重复图形的移动: 如在讲解 Z80 时序的图形中, 插入等待周期, 实际上是将无等待周期的图形后半部向后移动再插入等待周期的工作图形。我们采用动画技术用 GET (X<sub>1</sub>, Y<sub>1</sub>) - (X<sub>2</sub>, Y<sub>2</sub>), ARR 语句, 将要移动的图形参数记入数组 ARR。再用 PUT (X, Y), ARR, PSET 语句将图形复现在要显示的位上, 这样避免了重画的烦琐操作。

#### • 友好的人机界面

##### 进度控制:

(1) 课件在每一章节前都有明确的菜单, 可供学习者自由抉择。

(2) 学习者可选不同的学习方式:

顺序式: 按规定程序进行

跳跃式: 先进行测试, 测验通过者可以跳过学习者已掌握的部分, 这样就可以就学习者掌握情况有针对性的讲授。

##### 自由翻页:

针对学习者回看以加深理解或快速浏览的需要, 课件设计了自由翻页功能即在“一幅”教材结束后在屏幕的右上或右下角给出“> <”标示, 学习者可选择“>”或“<”键自由前翻或后翻。

##### 自我检测:

课件各章节结束都有总结和测验以使学习都有自我测

# 中华学习机图文显示及打印技巧

四川省南充市一中 陈庆祥

## 一、如何在中华学习机上实现图文并茂。

设计 CAI 软件时，往往需要在图形画面上同时显示汉字或其他字符。在《电脑》杂志 89 年 6 期上曾刊登过《中华学习机上汉字与图形共存的方法》一文，该文给出以下两程序来实现图文并茂的目的：

```
5 REM PROGRAM 1
10 HOME
20 POKE 14216+6, 64
30 POKE 14716+4, 127
40 PRINT "正方形"
50 HPLLOT 50,50 TO 100,50 TO 100,100 TO 50,100 TO 50,50
60 END
```

```
5 REM PROGRAM 2
10 HGR 1
20 HCOLOR=3
30 PRINT "正方形"
40 HPLLOT 50,50 TO 100,50 TO 100,100 TO 50,100 TO 50,50
50 END
```

```
RUN
?syntax error in 10
```

可是，令人十分遗憾，经笔者反复试验的结果表明，上述程序根本无法达到其目的。程序 1 运行后，不仅不能绘出任何线段，而且程序本身也不见了。更为糟糕的是，



试掌握程度的机会。测验结果有相应的评分，对错也给出有声提示和讲评或给以精神鼓励（一段乐曲等）以提高学习者的兴趣。

短暂休息：为调节学习者在课程进行到适当阶段安排休息，我们设计了“音乐欣赏”等程序即用 BASIC 语言中音乐设计语句将数十段短小精彩的乐曲编成以菜单形式供学习者欣赏，从而增加课件的趣味性。

### · 语音控制：

我们开发的有声辅助教学系统中增加了语音合成器，使教学过程更为生动活泼。由于目前计算机语音输出的音质和自然度还有待改进，因此过多的计算机语音输出在教学中是不必要的，必须对计算机显示及发音人为的加以控制。在我们开发的系统中用打印 ASCII 码的 1~6 可以实现有声，无声；有显示，无显示（实现背景发声）；汉字发声与字母皆可发声的自由控制。如：

PRINT CHR\$ (3) 为有声

PRINT CHR\$ (4) 为有声

但这里有一个问题，即程序加入上述语句，屏幕上会出现不必要的显示而破坏了画面，为此我们采用了如下办法加以消除。

所有 BASIC 命令全部失效，以致于不得不重新冷启动来恢复整个系统。而程序 2 一运行就打印出第 10 行语法错误的信息。究其原因，是由于中文状态虽以高分辨率第二页图形区作为屏幕显示区，但零页中设定作图方式的专用单元 \$E6 尚未获得对应值 \$40，此时强行使用 HPLLOT 语句便会破坏若干重要的零页系统指针，产生上述后果。该文作者企图在 \$378D 和 \$3980 两单元中分别置入 \$40 与 \$7F 两值，来达到设定作图方式与颜色的目的，显然是毫无用处的，因为上述两单元是用户区（高分辨率图形区第一页）内的普通单元，与 \$E6、\$E4 单元并没有任何直接联系。固然，如果程序 1 在运行前已执行过一次 HGR2 命令，该程序尚可运行并取得相应结果，但决非第 20、30 语句的作用，因为删除它们结果也一样（估计这是该文作者失误的原因所在）。程序 2 的错误有两方面：一是误认为可以使用高分辨率图形区第一页来显示汉字（采用 HGR1），二是使用了 HGR1 这条 APPLESOFT 中根本不存在的命令（规定用 HGR 与 HGR2 分别进入高分辨率图形区第一页与第二页）。

在中华机上实现图文并茂的正确方法是：通过用 POKE 命令将 \$40 直接置入 \$E6 单元，或者直接使用 HGR2 命令（在此之前，系统应在中文状态下）进入高分辨率图形区第二页，即可在绘图时打印汉字。两种方法的

```
900 AUDIO$ = CHR$ (3)
```

```
910 U = CSRLIN: V = POS (0); 取光标
```

```
920 PRINT Audil$
```

```
930 Locate u, v: print " "; 消除显示
```

· 音图并茂的“包装”

课件的“封面”是很重要的，一开始就抓住学习者的兴趣对后面课程的深入很有好处。因此我们借鉴电影电视的处理手法开始就用多彩的图形显示课程的名称及涉及的内容，并伴有背景音乐（原系统喇叭）及背景汉语介绍（语音合成器输出）这种计算机“全方位”输出的方式取得很好的效果。

## 结束语

一个成功的计算机辅助教学软件的编制如同小说被改编成电影或电视剧本一样，需要有经验的教师利用计算机的表现手法将教学内容进行“再创作”。我们利用了计算机大部分表现手法计算机图形、动画、音乐、语音输出。课件编写取得了一些经验。随着计算机辅助教学的发展和普及，它将和电视教学一样在电化教学的领域占有一席之地。

区别在于: POKE 230, 64 仅进入高显第二页, 不清屏, 而 HGR2 不仅进入高显第二页, 同时有清屏作用。如在前面一种情况下需要清屏可使用 PRINT CHR\$(12) 命令。顺便提一句, 如仅用白色或采用的是单色显示器, 不必再使用 HCOLOR 命令或修改 \$E4 单元的值, 下面是给出的图文并茂程序实例:

```
5 REM PROGRAM 3
10 POKE 230, 64
20 PRINT CHR$(12)
30 PRINT "CEC-I 图文并显"
40 PRINT "及硬拷贝的制作"
50 HPLLOT 8,14 TO 130,14 TO 130,52 TO 8,52 TO 8,14
60 HPLLOT 6, 12 TO 132, 12 TO 132, 54 TO 6, 54 TO 6, 12
70 PRINT: PRINT "1991.1.10"
80 PRINT CHR$(18)
90 PRINT CHR$(4) "BSAVE YKB, A$4000, L$2000"
100 END
```

RUN  
CEC-I 图文并显  
及硬拷贝的制作  
1991.1.10

说明: 命令 PRINT CHR\$(18) 是清除第 11 行提示内容。第 80 行是将屏幕显示内容以二进制文件方式存入磁盘。其好处是可长期保存和随时调用, 亦可用于特殊的屏幕显示方式, 如双页交互式动画 (见《电脑》89 年第 2 期苹果机上的交互式动画一文, 或“黑幕”显示技术 (见下文三)。另外, 亦可并入其他的西文软件, 作为软件题头的画面, 这对于不具备汉字系统的 APPLEII 具有特殊的弥补作用。

## 二、图文硬拷贝的制作

有时, 需要将图形 (包括含有汉字的图形) 打印在纸上 (即制成硬拷贝)。我们在运行程序 3 时, 其中的第 20 行接通了打印机, 但是结果只打印出了汉字, 没有打印出图形。这是因为在中文状态下, 系统并没有提供打印图形的功能。不过, 我们可以将系统及时转入西文状态 (这时屏幕上的内容全部消失), 再通过特殊命令将屏幕上原有图形文字打印在纸上。下面的程序 4 是先在屏幕上显示出图形与汉字, 按一下任意键, 就将屏幕上的图文打印在纸上。如需将图形比例放大一倍, 将 POKE 1913, 2 改为 POKE 1913, 66; 如需打印反相图形, 则改用: POKE 1913, 34 (注意: PR#1 后的 PRINT 决不可少)。

```
5 REM PROGRAM 4
10 PRINT CHR$(4); "PR#3": PRINT
15 PRINT CHR$(12): POKE 1659, 0
20 PRINT CHR$(18)
30 PRINT "CEC-I 图文并显"
40 PRINT "及硬拷贝的制作"
50 HPLLOT 8,32 TO 130,32 TO 130,70 TO 8,70 TO 8,32
60 HPLLOT 6,30 TO 132,30 TO 132,72 TO 6,72 TO 6,30
70 PRINT: PRINT "1991.1.10"
80 GETH$: TEXT
90 PR#1: PRINT
100 POKE 1913, 2
110 PRINT CHR$(17)
```

120 PR#0: END

CEC-I 图文并显  
及硬拷贝的制作

1991.1.10

## 三、“黑幕”显示技术

“黑幕”技术在舞台表演中被广泛采用。表演开始之前, 工作人员先在幕布后布置好各种道具与灯光, 演员作好预定造型。演出开始才将帷幕打开, 将舞台上的情景展示给观众。在此之前, 观众不会看到上述的准备过程。

在中华学习机上可以方便地实现类似的“黑幕”显示技术, 显示的预定内容可以是图形、文字或图文并茂的画面。由于这样能避免了现场作图与显示字符的打点过程, 因此在设计某些辅助教学软件或游戏软件时很有用处。

在中华机上的“黑幕”显示技术分四步:

(1) 预先制作“造型”: 先将图形、文字显示在某一页高分辨率图形区内 (第一页或第二页), 然后以二进制文件的方式存入磁盘。

(2) 制作“黑幕”: 进入文本方式清屏, 或进入与“造型”不同的另一页高分辨率图形区 (用 HGR 或 HGR2, 二者都具有清屏作用)。这时屏幕上全黑, 什么也看不见。

(3) 调入“造型”: 在程序中用 BLOAD 命令调入磁盘中的图形文件。

(4) 拉开“帷幕”: 利用屏幕显示软开关进入预定的图形区, 将“造型”一下子显示出来 (屏幕显示软开关是几个专用单元, 访问它们可进入相应的显示方式, 但是不会清除屏幕)。以下的程序 5 便是调用程序 3 存入的“造型”, 使用“黑幕”显示技术的例子。其中 40、50 语句行使用了四个软开关, 使屏幕显示进入高分辨率图形第二页。30 语句的 CALL 50030 是调用一个机器语言子程序, 以保证图形清晰稳定。

```
5 REM PROGRAM 5
10 TEXT: HOME
20 PRINT CHR$(4); "BLOAD YKB"
30 GET X$: CALL 50030
40 POKE 49232, 0: POKE 49234, 0
50 POKE 49237, 0: POKE 49239, 0
60 END
```

## 小 启

本刊作者来信反映, 部分读者直接与作者联系购买软件, 给他们增添了许多麻烦, 影响了正常工作。今后请读者直接到杂志社软件库购买软件。多谢合作。





# 新编 CHAIN 程序介绍

郑州市技工学校 邵爱柏

APPLE DOS 3.3 系统主盘文件《CHAIN》，可用链接两个 BASIC 程序，使先运行的程序中的变量保存下来，传递给后运行的程序使用。但该程序在中文状态下无效，且西文下使用时，变量值也只能使用一次，因为再运行第二个程序时，被传递的变量均被初始化（等于 0 或空字符串）。有人用保存变量指针方法（存盘和调用）解决此类问题亦不十分理想。因为这样会增加读写操作，有时会因有写保护而出错。

新编 CHAIN 程序可以解决这些问题。它的设计思想也是保存变量指针，使之存入安全区在运行第二个程序时恢复这些指针。

示范程序如下：

<程序 A>

```
10 LOMEM: 24576: POKE 111, 255: POKE 112, 149
20 D$=CHR$(4): IF PEEK(777)=160 THEN 40
30 PRINT " ": PRINT D$; "BLOAD CHAIN"
40 REM 用户程序内容
50 A=100: A$="CEC-DOS": A$(1)="中华学习机"
60 PRINT A, A$: PRINT A$(1)
70 REM 将字符串变量，数组变量分别加上空串，空数组以便传递
80 A$=A$+K$: A$(1)=A$(1)+K$(1)
90 CALL 777: REM 保存变量指针
100 PRINT " ": PTINT D$; "RUN B"
```

<程序 B>

```
10 CALL 808: POKE 2053, 178: REM 恢复变量指针，使 10 语句成为 REM 语句，避免重复执行
20 B=A: B$=A$: B$(1)=A$(1): REM 赋值后以便重复运行
30 PTINT A, A$: PRINT A$(1): REM 变量已被传递
40 C=111: C$="CHAIN": C$(1)="郑州市技工学校"
50 PRINT C, C$: PRINT C$(1)
60 INPUT "还运行吗 (Y/N)?: " Y$
70 IF Y$="Y" THEN 20: REM 转移语句不清变量值，RUN 命令将清除变量值
80 PRINT "运行结束!": END
```

新编《CHAIN》另一功能是用来保护（移走）和恢复（迁回）BASIC 程序，以便将程序转存到不同 DOS 的磁盘上。比如将未加密的磁盘上的程序转存到加密磁盘上（加密盘能够中断且知道存盘调用文件命令）。

使用方法：

1. 将 CHAIN 文件调入内存
2. 调入未加密盘 BASIC 程序
3. 键入 CALL 838，将程序及指针保护好
4. 启动加密盘 DOS（欢迎程序可中断）
5. 键入 CALL 889，将程序迁回并恢复零页指针
6. 将程序存入加密盘

\* 309.3AB

```
0309- A0 00 A9 69 85 3C A9
0310- 70 85 3E A9 00 85 3D 85
0318- 3F 85 42 A9 03 85 43 20
0320- 2C FE 60 00 00 00 00
0328- A0 00 A9 00 85 3C 85 43
0330- A9 03 85 3D 85 3F A9 08
0338- 85 3E A9 69 85 42 20 2C
0340- FE 60 00 00 00 00 A0 00
0348- A9 60 85 3C 85 42 A9 00
0350- 85 3D 85 3F A9 FF 85 3E
0358- A9 90 85 43 20 2C FE A0
0360- 00 A9 00 85 3C 85 3E 85
0368- 42 A9 08 85 3D A9 48 85
0370- 3F A9 50 85 43 20 2C FE
0378- 60 A0 00 A9 60 85 3C 85
0380- 42 A9 90 85 3D 85 3F A9
0388- FF 85 3E A9 00 85 43 20
0390- 2C FE A0 00 A9 00 85 3C
0398- 85 3E 85 42 A9 50 85 3D
03A0- A9 90 85 3F A9 08 85 43
03A8- 20 2C FE 60
```

## 改进 BLOAD 命令

天津市 高炳悦

我们在使用一个 B 型文件时，经常希望知道它在内存中的首地址和长度。而通常使用的办法打入下列命令：

PRINT PEEK(43634)+PEEK(43635)\*256 显示首地址

PRINT PEEK(43616)+PEEK(43617)\*256 显示长度

因为是立即执行的方法，所以每次使用都得输入以上命令，既浪费时间又极不方便；如把它写入 BASIC 程序，则失去了通用性。

最好是在调入文件的同时就能知道它的首地址和长度，那就方便多了。要达到这个目的只要对 BLOAD 命令的处理程序作一些补充即可。这项工作可以通过文末的 BASIC 程序来完成。它适用于 CEC-I 和 APPLE II 微机。

运行本程序后，再使用 BLOAD 命令时就能看到由两个四位十六进制数表示的当前调入的 B 型文件的首地址和长度。本程序运行后，如果初始化一新盘，那么此盘的 DOS 也具有这样功能的 BLOAD。

程序清单：

```
10 REM IMPROVE BLOAD
200 POKE 40272, 181: POKE 40273, 82
210 FOR X=46774, TO 46825
220 READ A: POKE X, A
230 NEXT
240 DATA 32, 93, 163, 169, 114, 133, 249, 169, 170, 133,
250, 169, 193, 32, 214, 182, 32, 72, 249, 169, 96, 133, 249, 1
69, 204, 32, 214, 182, 32, 98, 252, 96, 32, 240, 253, 169,
164, 32, 240, 253, 160, 0, 177, 249, 170, 200, 177, 249, 32,
65, 249, 96
```

# 递归算法

邹家炜

递归的威力在于可以把一些复杂的算法表达得非常紧凑。对于递归定义的基本数据结构,如二叉树,更是如此。事实上许多问题都具有递归特性,所以在进行程序设计时恰当地使用递归算法往往使算法的描述比使用非递归方法易读,易于证明算法的正确性,和分析算法的复杂度。况且对于一些具有递归特性的问题,要直接表达或非递归形式可能是相当困难的。所以对递归算法的学习和运用应当给以足够的重视。下面我们就来谈谈递归算法。

## 一、递归

如果某一对象部分地用自身来定义,或包含了自身,我们就称它是递归的。

例如: 1.自然数是按递归规则来定义的:

a.1 是自然数。

b.自然数的后继是自然数。

2.二叉树是按如下规则递归定义的:

a.空树是二叉树。

b.非空二叉树由一个结点(根)和两个二叉树(左子树和右子树)组成。

递归这一术语也可以用来描述算法。而且由递归定义可以自然地导出递归算法。

## 二、递归算法的种种

### 1. 二叉树的前序遍历

我们知道,前序遍历就是先访问二叉树的根结点,然后遍历其左子树,接着遍历其右子树。当访问了树中的一个结点后,总是先向左移动。当无法以继续时,则向右移动,或者向后移动,直至又可向右移动,并继续前进为止。可见二叉树的前序遍历的递归定义为:若二叉树为空,则空操作;否则(1)访问根结点;(2)前序遍历左子树;(3)前序遍历右子树。描述这种遍历的确切方法是把这个递归定义自然地写成一个递归算法。

```
PROCEDURE preorder (bt); { 前序遍历根特点指针为
bt 的二叉树 }
```

```
BEGIN
```

```
  IF bt≠NIL
```

```
  THEN (write (bt↑.data); { 访问根结点 }
```

```
    CALL preorder (bt↑.Lchild); { 遍历左子树 }
```

```
    CALL preorder (bt↑.Rchild); { 遍历右子树 }
```

```
END;
```

### 2. 二叉树的后序遍历

我们知道二叉树的后序遍历的递归定义是:若二叉树为空,则空操作;否则(1)后序遍历左子树;(2)后序遍历右子树;(3)访问根结点。显然我们容易地把这个递归定义导出其递归算法。事实上只要将上面前序遍历算法中

的访问特点移到遍历左、右子树之后进行,便可得到后序遍历的递归算法。(程序略)

### 3. 插入排序

不妨设已知  $N$  个整数无序地存在于数组  $A$  中,插入排序的基本思想是为了排序前面  $j$  个元素,首先排序前面  $j-1$  个元素,当前面  $j-1$  个元素已排序后,则从第  $j-1$  个元素开始往前寻找第  $j$  个元素的正确位置,然后将第  $j$  个元素插入即可。

```
PROCEDURE sort (j: integer);
```

```
VAR
```

```
  i: integer;
```

```
  item: integer;
```

```
  found: boolean; { 找到插入位置标志 }
```

```
BEGW
```

```
  IF j>1 THEN
```

```
  BEGIN
```

```
    sort (j-1);
```

```
    i := j-1;
```

```
    item := a (j);
```

```
    found := false;
```

```
    WHILE NOT found AND (j>=1) DO
```

```
      IF item>=a (i) THEN
```

```
        found := true
```

```
      ELSE
```

```
    BEGIN
```

```
      a (i+1) := a (i);
```

```
      i := i-1
```

```
    END
```

```
      a (i+1) := item
```

```
    END
```

```
  END;
```

为排序  $a$  的前  $n$  个元素,可用下面的过程语句实现:  
sort (n)

如果扩充数组  $a$ ,引入辅助成分  $a(0)$ ,并预置  $a(0) = -\text{maxint}$ ,则过程 sort 的重复寻找插入点的条件 NOT found AND ( $i \geq 1$ ) 可简化为 NOT found.

### 4. 图案设计

利用递归过程作图,这就大大丰富了 LOGO 语言的功能。在程序设计中递归算法,对于编制模块化的高级程序是十分有用的。事实上,在 LOGO 语言中,一个极其简单的递归过程,仅由于每次赋给的实参不同,就可以设计出各种风格不同的图案。

例如: 程序一 (过程一)

```

TO SHAPE
FD 40
RT 90
FD 40
RT 90
FD 20
RT 90
FD 20
RT 90
FD 40
RT 90
FD 10
RT 90
FD 10
RT 90
FD 20

```

END

过程一画出了如下的图案。



程序二 (过程二)

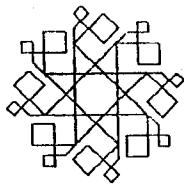
```

TO CRYSTAL
SHAPE
LT 45
FD 70
CRYSTAL

```

END

这个过程是一个画 CRYSTAL 的过程。开始时首先是画出了一个 SHAPE 的图案 (如上图), 然后海龟向左移动 45 度, 前进 70 步。由于我们在 CRYSTAL 过程体内又调用了 CRYSTAL 过程, 所以又画出了一个 SHAPE 的图案……这样无休止地画。于是得到的如下图就是 CHYSTAL (晶体) 的图案。



### 三. 把递归算法转换成非递归算法

我们知道某一算法递归地表达出来是相当简单, 但要非递归地开发却十分困难。但是, 事物总是一分为二的。递归算法也有它的缺点, 那就是在编译过程中要付出较大的额外的时间和空间的开锁, 很多程序设计语言并不允许递归作为一个正规特性。因此人们有时在开始设计算法时, 只能先设计递归算法, 然后将它转换成等价的非递归

算法。尽管这种转换并不容易, 但总的工作量常常小于直接开发非递归程序。

#### 1. 把递归算法转换成递推算法

我们可以把递推算法表示为: (1) 求初值  $V = V_0$ ; (2) 当条件  $P(V)$  成立, 则重复执行  $V = f(V)$ 。另外, 由递归关系得到的值的序列  $V_0, V_1, V_2, \dots, V_n$  具有如下性质: (1)  $V_0$  为初值, 单独确定。 (2)  $v_i = f(V_{i-1})$  或  $V_i = f(V_{i-1}, V_{i-2}, \dots)$  (3)  $P(V_i)$  总成立 ( $i < n$ ),  $P(V_n)$  不成立。因此, 一个递归算法能否转化为递推关系的算法, 就要看能否表示成  $V_i = f(V_{i-1})$  或  $V_i = f(V_{i-1}, V_{i-2}, \dots)$  的形式。其次对于  $P$  个自变量的递推公式要有  $P$  个确定的初值  $V_0, V_1, \dots, V_P$ ; 而且当  $V = n$  时是否有判断终止的条件。下面我们用一个例子来说明 BASIC 语言中利用递推算法来代替了递归算法。例如, 我们知道阶乘的递归公式是:

$$\begin{cases} f(n) = n * f(n-1) \\ f(0) = 1 \end{cases}$$

显然如果令  $n=0, 1, 2, \dots, n$ , 则有:  $f(0) = 1, f(1) = 1 * f(0) = 1, f(2) = 2 * f(1) = 2, \dots, f(n) = n * f(n-1)$  递推关系。这样求阶乘的递归算法就转化为: FACT1 (N, F)

(1)  $F = 1$

(2) 对  $I=1$  到  $N$ , 令  $F = F * I$

END

的递推算法, 并可得下面的程序。

```

100 F=1
110 INPUT N
120 FOR I=1 TO N
130 F=F*I
140 NEXT I
150 PRINT F
160 END

```

#### 2. 用堆栈技术把递归算法转换为非递归算法

为什么用堆栈技术可把递归算法转换为非递归算法呢? 要回答这个问题, 我们先来看看为了写出计算机算法, 我们先讨论遍历一棵树时要做些什么操作。向下处理分层结构, 只需沿着左链而下即可。每一从属表是由右链引出的线性链表。算法的要点在于, 当某一从属表列举完时, 应该做些什么。这时我们必须在分层结构中退回一级, 并接到我们从最后访问的级上离开的地方。换句话说, 当我们沿着分层结构向下移动时, 我们必须记住刚走过的地方的有关信息。我们需要的最小记录是分层结构每一级上最后被访问的结点。对于这些信息, 较合适的结构是堆栈。

如何利用堆栈技术把递归算法转换为非递归算法。下面我们先给出排序子程序中的递归算法的例子。

SUBROUTIN SORT (A, MIN, MAX)

1. IF MIN  $\neq$  MAX

# 将乐曲输给电脑

西安市 来跃深

我们已见到过许多让微电脑奏乐的程序。其结构一般是先输入一个小的机器码子程序，然后用 BASIC 程序调用它，这种方法子程序简单，但音高、音长的输入要从表中去找，而且表示范围较小，如本刊 1987 年 7 月的《微电脑演奏乐曲》一文中，音阶最低为 5，音长最长为  $1\frac{1}{2}$  拍。若音高、音长分别用双字节表示，输入的工作量明显增加。

对于中小学生，重要的是培养他们的思维方法。若要增加程序的通用性，使它可奏不同的乐曲，就需要将不同的乐曲输入给电脑，是需要多次输入的，而机器码子程序只需输入一次即可。为使乐曲输入迅速、方便，音高、音长的表示简单、直观，子程序稍复杂是必要的，应该的。因此我提出下面的方法，这种方法表示的音高、音长均为一个字节，其

对应关系如下：

音长对应：

节拍(拍)	$\frac{1}{16}$	$\frac{1}{8}$	$\frac{1}{4}$	$\frac{1}{2}$	$\frac{3}{4}$	$1$	$1\frac{1}{4}$	$1\frac{1}{2}$	$1\frac{3}{4}$	$2$	$2\frac{1}{2}$	$3$	$4$	$5\cdots$
数值	1F	18	14	12	13	11	54	32	74	21	52	31	41	51...

音高对应：

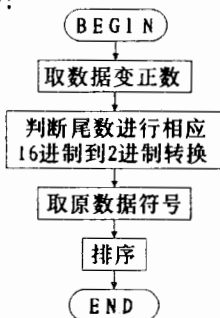
音高	0	1	2	3	4	5	6	7
	51	52	53	54	55	56	57	
	61	62	63	64	65	66	67	
	71	72	73	74	75	76	77	
80	81	82	83	84	85	86	87	
91	92	93	94	95	96	97		
A1	A2	A3	A4	A5	A6	A7		
B1	B2	B3	B4	B5	B6	B7		

表中省略了许多，读者已可以看出规律性了。由表中可见，用这种方法表示音长最短  $1/16$  拍，最长 16 拍，音高最低 1，最高 7，这已远超出平时的使用范围，足够我们使用了。

两张表虽然很大，只要掌握了规律，其实是不用记忆的。若对十六进制熟悉，可以拿着曲子直接在键盘上敲入，有兴趣的读者不妨试试。

(接 40 页)

2. 程序框图如下：



3. 转换结果举例：

SEL 机 数值	尾数 (绝对值)	小型机 表示码	转换为 微机码	通讯后 PS/2 数值
-0.07	$1/16 < F < 1/8$	BFEE147A	305C8FBD	-7.000000E-002
19.19	$1/8 < F < 1/4$	421330A3	18859941	+1.919000E+001
30.34	$1/4 < F < 1/2$	4212570A	50B8F241	+3.034000E+001
-0.6	$1/2 < F < 1$	BF666666	9A9919BF	-6.000000E-001

通过实验结果可以看到数据转换是正确和成功的。

五、小结

通过核电站安全监督系统中涉及到的 SEL32/7716 进制到 PS/2 二进制的格式转换处理可以看到，不同类型计算机的通讯中数据兼容性是很重要的，它将直接影响系统处理的真实性和可靠性。

A. 用 PART 来划分 A

B. CALL SORT (A, MIN, PART-1)

C. CALL SORT (A, PART+1, MAX)

2. RETURN

END OF SUBROUTINE SORT

在这个算法里有一递归调用后面紧接着 END OF SUBROUTINE，这样不用堆栈就可消除递归。例如可把 1.IF MIN≠MAX 改为 1.WHILE MIN≠MAX 但由于剩下的调用后面没有紧接着 RETURN，所以局部数据必须是栈式的。要保存和还原的数据是 PART 和 MAX。用两个数组 SPART, SMAX 和一个标量 TOP 作为堆栈。TOP=1 表示空栈。于是我们可得到下面的这个排序子程序的非递归算法。

SUBROUTINE SORT (A, MIN, MAX)

1.TOP=1

SRLAB: 2.WHILE MIN≠MAX

A. 用 PART 划分 A

B.SMAX (TOP) = MAX

C.SPART (TOP) = PART

D.TOP=TOP+1

E.MAX=PART-1

3.IF TOP≠-1

A.TOP=TOP-1

B.MAX=SMAX (TOP)

C.PART=SPART (TOP)

D.MIN=PART+1

E.GOTO SRLAB

4.RETURN

END OF SUBROUTINE SORT

# 226 种计算机病毒介绍

广州市计算机应用研究所 胡向东  
广州麦芽有限公司 鹿松

关键词: 病毒, 计算机病毒, 引导扇区, 文件  
Keywords: Virus, Computer Virus, Boat Sector, File

摘要: 文章列出了最新发现的IBM PC及其兼容机的病毒的主要特征及类型。

Summary: The following article outlines the major characteristic of the newly known IBM PC and Compatible virus. And it gives a brief introduction to the variants of the virus.

根据美国 McAfee 协会 1991 年 2 月 15 号公布的世界各地发现的病毒 [1], 及综合参考了其它资料 [2, 3], 此文介绍并整理列出了 226 种已发现的计算机病毒。

这 226 种病毒能感染的区域包括: 硬盘分区表, 硬盘或软盘的 DOS 引导扇区, 操作系统程序, 系统设备驱动程序, 扩展名为 COM 的文件, 扩展名为 EXE 的文件, 覆盖文件及其它能调入执行的文件。

病毒传染部份及编号:

- 1、使用了加密技术的病毒
- 2、使用了自身密码的病毒
- 3、驻留内存的病毒

- 4、感染 COMMAND.COM 的病毒
- 5、感染 COM 文件的病毒
- 6、感染 EXE 文件的病毒
- 7、感染覆盖文件
- 8、感染软盘系统引导区
- 9、感染硬盘系统引导扇
- A、感染硬盘分配表

226 种病毒名称及特征表:

序 号	病 毒 名 称		病 毒 传 染 部 位											增 加 字 节 长 度	损 害 后 果
	中 英 文 对 照		1	2	3	4	5	6	7	8	9	A			
1	Skism	斯奇森			是		是	是	是				1815	O, P	
2	Holocaust	大屠杀	是		是	是	是						3784	O, P, L, D	
3	Stone-90	石头-90				是	是						961	O, P	
4	903				是	是	是						903	O, P	
5	Dir-Vir	目录病毒	是		是	是	是						691	O, P, D	
6	Hybrid	混血儿				是	是						1306	O, P, L	
7	IKV528					是	是						528	O, P	
8	Iraqi Warrior	伊拉克勇士				是	是						777	O, P, L, D	
9	Little Pieces	小碎片			是		是	是					1374	O, P	
10	Saddam	萨达姆			是	是	是						919	O, P, D, L	
11	Monxla-B	蒙西拉-B				是	是						535	O, P, L	
12	Plague	瘟疫					是	是					改写	程序	
13	Sentinel	哨兵			是	是	是	是	是				4625	L, O, P, D	
14	Swiss 143	瑞士 143				是	是						143	O, P, D	
15	Voronezh			是	是	是	是	是	是				1600	O, P, D	
16	V-961	961 病毒			是	是	是						961	O, P,	
17	USSR-948	苏联-948		是			是	是	是				948	O, P, D	
18	USSR-711	苏联-711		是			是						711	P, D	

(续上表)

19	USSR-707 苏联-707		是		是	是						707	P, D
20	USSR-696 苏联-696		是			是						696	P, D
21	USSR-600 苏联-600		是		是	是						600	P, D
22	USSR-394 苏联-394		是		是	是						394	P, D
23	USSR-257 苏联-257		是		是	是						257	P, D
24	USSR-256 苏联-256		是		是	是						256	P, D
25	USSR-492 苏联-492					是						492	O, P
26	USSR-1049 苏联-1049			是		是	是					1049	O, P, L
27	USSR-311 苏联-311					是						321	O, P
28	USSR-830 苏联-830			是	是	是						830	O, P
29	USSR-529 苏联-529			是	是	是						529	O, P
30	USSR-516 苏联-516			是	是	是						516	O, P
31	USSR-2144 苏联-2144		是	是	是	是	是	是				2144	L, O, P, D
32	USSR-1049 苏联-1049			是	是	是						1049	O, P
33	USSR (3) 苏联 (3)		是			是						575	O, P
34	Tiny-133 小不点-133				是	是						133	O, P
35	Sverdlov 斯维特罗夫			是	是	是	是	是				1962	O, P
36	Label 标签			是	是	是						改写	程序
37	Kuka / Turbo 库卡 / 汽轮机			是	是	是						改写	程序
38	Justice 公正的赏罚			是	是	是						1242	O, P
39	Hymn 赞美诗			是	是	是	是	是				642	O, P, D
40	Happy New Year 新年快乐			是	是	是	是	是				1865	O, P
41	Destructor 破坏者			是	是	是	是	是				1150	O, P
42	Leapfrog Virus 跳蛙病毒			是		是						516	O, P, D
43	MGTU Virus (2) MGTU 病毒 (2)				是	是						273	O, P, D
44	Nina 尼娜			是	是	是						256	O, P, D
45	Lozinsky 罗申斯基				是	是						1023	O, P, D
46	BeBe 婴儿				是	是						1004	O, P, D
47	Best Wish 最美好的愿望				是	是	是	是				1024	O, P, D
48	Beeper (2) 遥控者 (2)			是		是						482	O, P, D
49	Parity 平等				是	是						441	O, P, D
50	F-word 父亲的话			是	是	是						417	O, P, D
51	Christmas Violator 圣诞扰乱者					是							O, P, D
52	Off Stealth 泄密	是		是	是	是	是	是				1689	O, P, D
53	Jeff 杰弗				是	是						828	O, P, D, F
54	Bloody! 血腥		是	是					是	是		N/A	B, O
55	ZeroHunt 无法找	是	是	是		是						N/A	O, P, D
56	Music Bug (3) 音乐臭虫 (3)			是					是	是		N/A	B, O
57	Dot Killer 小不点杀手			是	是	是						944	O, P
58	Father Christmas 圣诞老人				是	是						1881	O, P

(续上表)

59	3445	是	是	是		是	是					3445	O, P, D, L
60	Mirror (2) 镜子 (2)			是			是					928	O, P
61	Polish-2 警察-2			是	是	是						512	O, P, D
62	Polish-217 警察-217				是	是						217	O, P, D
63	Happy Day 快乐节				是	是						453	O, P
64	Monxla 蒙西拉				是	是						939	O, P
65	USSR 苏联		是				是					575	O, P
66	Polimer 聚合物				是	是						512	O, P, D
67	Datalock 数据锁			是			是					920	O, P
68	Carioca (2) 里约热内卢 (2)			是		是						951	O, P
69	529			是	是	是						529	O, P, D
70	Spyer 间谍			是		是	是	是				1181	O, P
71	Taiwan4 台湾4号			是	是	是	是	是				2576	O, P, D
72	Keypress (3) 按键 (3)			是	是	是	是					1232	O, P, D
73	Casper 菟子		是		是	是						1200	L, O, P, D
74	1605			是	是	是	是					1605	L, O, P, D
75	Violator (6) 扰乱者 (6)				是	是						1055	O, P, D
76	Blood-2 血-2					是						427	O, P, D
77	Wisconsin 威斯康星州		是		是	是						825	O, P, D
78	Christmas-J 圣诞-公平			是	是	是	是					600	O, P
79	Austria 奥地利				是	是	是					改写	程序
80	Leprosy-B 麻风病-B				是	是	是					改写	程序
81	Whale (3) 鲸鱼 (3)	是	是	是	是	是	是	是				9216	L, O, P, D
82	Invader (4) 侵略者 (4)		是	是		是	是	是	是	是		4096	B, L, O, P, D
83	Scott's Valley 苏格兰山谷		是	是		是	是	是				2133	L, O, P, D
84	Anarkia 阿纳奇亚			是		是	是	是				1813	O, P, D
85	Black Monday 黑色星期一			是	是	是	是	是				1055	L, O, P, D
86	Nomenclature 术语			是	是	是	是	是				1024	O, P, D
87	Anthrax-Boot 炭疽病-引导型			是							是	N/A	O, P, D
88	Anthrax-File 炭疽病-文件型			是	是	是	是					1206	O, P, D
89	651			是		是						651	O, P, D
90	Paris 巴黎				是	是	是	是				4909	O, P, D, L
91	Leprosy (5) 麻风病 (5)			是	是	是	是	是				改写	程序
92	Mardi Bros (3) 失败的星期二 (3)			是					是	是		N/A	B, O
93	1253-Boot 1253-引导型			是					是	是	是	N/A	O, P, D, L
94	1253-COM 1253-COM 型			是	是	是						1253	O, P, D, L
95	AirCop (3) 空中警察 (3)			是					是			N/A	B, O
96	400 (5)			是		是						Vary	O, P, D
97	P1 (6)		是	是		是						Vary	O, P, D, L
98	Ontario 安大略省		是	是	是	是	是					Vary	O, P, D



(续上表)

99	1226 (3)		是	是	是	是	是	是			1226	O, P, D
100	V2100 2100 病毒			是		是	是				2100	O, P, D, L
101	Plastique 塑料炸弹			是	是	是	是	是			3012	O, P, D
102	Wolfman 狼人			是	是	是	是				2064	O, P
103	Doom2 毁灭 2 号			是		是	是				2504	O, P, D, L
104	Flip (4) 热甜啤酒 (4)		是	是	是	是	是	是			2343	O, P, D, L
105	Fellowship (3) 友谊 (3)			是			是				1022	O, P, D, L
106	Flash 闪烁			是	是	是	是				688	O, P, D, L
107	1008		是	是	是	是					1008	O, P, D, L
108	Taiwan 3 台湾 3 号			是	是	是	是	是			2905	O, P, D, L
109	Armagedon 阿麻格顿			是	是	是					1079	O, P
110	1381						是	是			1381	O, P
111	Tiny (13) 小不点 (13)				是	是					163	O, P
112	Subliminal (3) 异常 (3)			是	是	是					1496	O, P
113	Sorry 对不起			是	是	是					731	O, P
114	RedX (2) 红色的 X (2)				是	是					796	O, P
115	1024 (2)			是	是	是					1024	O, P
116	Joshi (4) 乔士 (4)	是		是				是	是	是	N/A	B, O, D
117	Microbes 细菌			是				是	是		N/A	B, O, D
118	Print Screen (2) 打印屏幕 (2)			是				是	是		N/A	B, O, D
119	Form (2) 形式 (2)			是				是	是		N/A	B, O, D
120	July 13th 7 月 13 号	是					是				1201	O, P, D, L
121	5120 (3)				是	是	是	是			5120	O, P, D, L
122	Victor (2) 胜利者 (2)			是	是	是	是	是			2458	P, D, L
123	JoJo (3) 乔乔 (3)			是		是					1701	O, P
124	W-13 (4)					是					532	O, P
125	Slow (5) 慢性子 (5)		是	是		是	是	是			1721	O, P, L
126	Frere Jacques 兄弟装糊涂			是		是	是	是			1811	O, P
127	Liberty (2) 自由 (2)			是	是	是	是	是			2862	O, P
128	Fish-6 (2) 鱼-6 (2)	是	是	是	是	是	是	是			3584	O, P, L
129	Shake 握手			是		是					476	O, P
130	Murphy 隐壁床			是	是	是	是	是			1277	O, P
131	V800 (3) 800 病毒 (3)	是	是	是		是					none	O, P, L
132	Kennedy (3) 肯尼迪 (3)			是		是					308	O, P
133	8 Tunes/1971 (2) 8 音调/1971 (2)			是		是	是	是			1971	O, P
134	Yankee-2 杨基 2 号					是	是				1961	O, P
135	June 16th 6 月 16 号				是	是					1726	F, O, P, L
136	XA1		是			是					1539	F, O, P, L
137	1392			是	是	是	是				1392	O, P, L
138	1210			是		是					1210	O, P, L

(续上表)

139	1720 (3)			是	是	是	是			1720	F, O, P, L
140	Saturday 14th (3) 14号星期六 (3)			是	是	是	是			685	F, O, P, L
141	Korea (4) 南朝鲜 (4)							是	是	N/A	B, O
142	Vcomm (5) 维康 (5)					是				1074	O, P, L
143	ItaVir 意大利病毒					是				3880	O, P, L, B
144	Solano (3) 东风 (3)			是	是					2000	O, P, L
145	V2000 (3) 2000病毒 (3)			是	是	是	是	是		2000	O, P, L
146	1559			是	是	是	是			1559	O, P, L
147	512 (5)	是		是	是	是				None	O, P, L
148	EDV (2) 编辑病毒 (2)	是		是				是	是	N/A	B, O
149	Joker 开玩笑的人			是	是	是					O, P
150	Icelandic- (3) 冰岛 (3)			是			是			853	O, P
151	Virus-101 101病毒		是	是	是	是	是	是		2560	P
152	1260 (3)		是			是				1260	P
153	Perfume (2) 香水 (2)					是				765	P
154	Taiwan (3) 台湾 (3)					是				708	P
155	Chaos 混乱			是				是	是	N/A	B, O, D, F
156	Virus-90 90病毒			是		是				857	P
157	Oropax (5) 山中女神 (5)			是		是				2773	P, O
158	4096 (4)	是		是	是	是	是			4096	D, O, P, L
159	Devil's Dance (2) 魔鬼跳舞 (2)			是		是				941	D, O, P, L
160	Amstrad (5) 阿姆斯特特 (5)					是				847	P
161	Payday (2) 发薪日 (2)			是		是	是	是		1808	P
162	Datacrime II-B 数据犯罪二 (B)	是		是		是	是			1917	P, F
163	Sylvia/Holland 斯丽维亚/荷兰女孩					是				1332	P
164	Do-Nothing 无事可做			是		是				608	P
165	Sunday (4) 星期日 (4)			是		是	是	是		1636	O, P
166	Lisbon (2) 里斯本 (2)					是				648	P
167	Typo/Fumble 排印工/出错			是		是				867	O, P
168	Dbase 数据库			是		是				1864	D, O, P
169	Ghost Boot 幽灵 (引导型)			是				是	是	N/A	B, O
170	Ghost COM 幽灵 (COM)					是				2351	B, P
171	Alabama (3) 阿拉巴马 (3)			是			是			1560	O, P, L
172	Yank Doodle (6) 扬基多得 (6)			是		是	是			2885	O, P
173	2930			是		是	是			2930	P
174	Ashar 阿沙尔			是				是		N/A	B
175	AIDS (4) 爱滋病 (4)					是				改写	程序
176	Disk Killer (4) 磁盘杀手 (4)			是				是	是	N/A	B, O, P, D, F
177	1536/Zero Bug 1536/零号病菌			是		是				1536	O, P
178	Dark Avenger (4) 黑夜复仇者 (4)			是	是	是	是	是		1800	O, P, L

(续上表)

179	3551 / syslock	3551 / 系统锁		是			是	是				3551	P, D
180	VACSINA	维克森纳			是		是	是	是			1206	O, P
181	Ohio	俄亥俄			是					是		N / A	B
182	Typo Boot	排印工 (引导型)			是					是	是	N / A	O, B
183	Swap Boot	交换 (引导型)			是					是		N / A	B
184	Datacrime-2 (2)	数据犯罪 2 (2)		是			是	是				1514	P, F
185	Icelandic II (2)	冰岛二号 (2)			是			是				661	O, P
186	Pentagon	五角大楼								是		N / A	B
187	Traceback (3)	反跟踪 (3)			是		是	是				3066	P
188	Icelandic (4)	冰岛 (4)			是			是				642	O, P
189	Fu Manchu (4)	佛玛树 (4)			是		是	是	是			2086	O, P
190	Datacrime (2)	数据犯罪 (2)		是			是					1280	P, F
191	1701 / Cascade (12)	1701 / 瀑布 (12)		是	是		是					1701	O, P
192	Stoned (5)	大麻 (5)			是					是	是	N / A	O, B, L
193	Den Zuk (3)	丹柱克 (3)			是					是		N / A	O, B
194	Ping Pong (7)	乒乓 (小球) (7)			是					是		N / A	O, B
195	Lehigh	勒海			是	是						改写程序	P, F
196	Vienna / 648	(23) 维也纳 / 648 (23)					是					648	P
197	Alameda (2)	林荫道散步 (2)			是					是		N / A	B
198	Friday 13th COM	13 号星期五					是					512	P
199	Jerusalem (41)	耶路撒冷 (41)			是		是	是	是			1808	O, P
200	Brain (3)	巴基斯坦智囊 (3)			是					是		N / A	B
201	Surviv-1	舒尔富-1			是		是					897	O, P
202	Surviv-2	舒尔富-2			是			是				1488	O, P
203	Surviv-3	舒尔富-3			是		是	是	是				O, P
204	Azusa / 2708	阿珠沙 / 2708 (打印病毒)			是					是	是	N / A	O, B
205	2708-B	(打印病毒变种)			是					是	是	N / A	O, B
206	Guangzhou University -1	广大一号			是					是	是	N / A	O, B, L
207	1591 / 1575	(毛毛虫病毒)			是		是	是	是			1575	O, P
208	AIDS information trojan	爱滋信息勇士	是				是					改写	程序
209	New Jerusalem	新耶路撒冷			是		是	是	是			1808	O, P
210	Mix1	混合 1			是			是				1618	O, P
211	Saratoga	沙如塔克			是			是				632	O, P
212	1168 / Datacrime-B	1168 / 数据犯罪-B	是				是					1168	P, F
213	405						是					改写	程序
214	1704 Format	格式化	是		是		是					1704	O, P, F
215	1704 / Cascade	1704 / 瀑布	是		是		是					1704	O, P
216	1704 / Cascade-B	1704 / 瀑布-B	是		是		是					1704	O, P
217	Ping Pong-B	乒乓-B / 小球-B			是					是	是	N / A	O, B
218	Vienna-B	维也纳-B					是					648	P

# 小型机与微机数据通讯中的数据格式转换

清华大学核电站模拟培训中心 方向

数据格式转换在不同机型计算机通讯中有着重要的意义。本文通过核电站安全监督系统中涉及到的 SEL32/77 小型机和 PS/2 微机数据通讯中数据格式转换问题,介绍了非兼容计算机通讯中进行数据格式转换的方法,说明了数据格式转换的重要性及可行性。本文的处理方法对其他机型的数据格式转换有一定参考价值。

## 一、背景

核电站安全监督系统是实时反映核电站状况的安全监督系统,它主要由安全参数显示系统 SPDS (Safety Parameter Display System)、故障诊断系统 FDS (Failure Diagnosis System) 和功能恢复系统 FRS (Function Recover System) 组成。本系统是在清华大学核电站模拟培训中心的模拟机上实现的,这台模拟机是实时、全范围模拟核电站状况的模拟机,全部软件由 SEL32/77 小型机支持。核电站安全监督系统的动态数据是由 SEL32/77 计算机的数据库 (DATAPOOL) 提供的,SPDS、FDS 和 FRS 是采用 PS/2 微机实现的,由于本模拟机与核电站安全监督系统都是在线的,这就要求 PS/2 从主机 (SEL32/77) 上采集的数据是实时反映系统状况的,这就牵涉到 SEL32/77 和 PS/2 之间实时数据通讯中的数据兼容性问题。

(续上表)

219	Jerusalem-B	耶路撒冷-B			是		是	是	是			1808	O, P
220	China Bomb	中国炸弹			是	是						1492	P, F
221	AIDS / Taunt	嘲笑艾滋病											
222	Hallochen	陈生, 你好											
223	April First	愚人节											
224	Golden gate-C	金门病毒-C					是						
225	Virus-B	B-病毒					是						
226	2730											2730	O, P

说明:

增加的空间:

N/A: 病毒不粘附到文件上

NONE: 病毒不改变文件长度 (而是粘附在尾部)

改写程序: 病毒改写文件开头, 不改变文件长度。

损害范围:

参考文章:

B——破坏或改写启动扇区

D——破坏数据文件

F——格式化或改写所有或部分磁盘

L——直接或间接破坏文件链接

O——影响系统运行速度

P——破坏程序或覆盖文件

[1]: VIRUS CHARACTERISTICS LIST V74 Copyright 1989, 1990, 1991 McAfee Associates, (408) 988-3832

[2]: 检测 78 种, 消除 14 种病毒软件使用指南 V16.0, 公安部计算机安全监察处

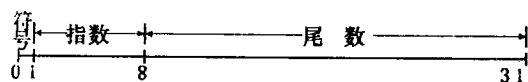
[3]: 计算机病毒大观, 《计算机应用研究》

## 二、小型机和微机的数据格式

如果小型机和微机中的数据采用同一标准格式, 则在数据通讯中数据是兼容的, 不需要转换, 但在 SEL32/77 和 PS/2 内部采用的数据标准格式是不同的。

### 1. 小型机 (SEL32/77) 内部数据格式。

SEL32/77 是 32 位小型机, 以浮点数表示为例, SEL32/77 的浮点数由三部分组成: 符号、尾数和指数。如图示:



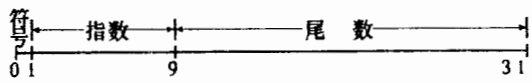
0 位是符号位, 用来表示值的正负; 尾数从第 8 位开始, 左边是高位; 1~7 位表示以 16 进制为底的指数。

浮点数值是以尾数乘 16 的指数次幂得到的, 其中指数是 1~7 位二进制数减 16 进制 40 得到的。即浮点数可由  $F \times 16^m$  表示,  $F$  为尾数 ( $1/16 < F < 1$ ,  $-1 < F < -1/16$ )。

### 2. 微机 (PS/2) 内部数据格式。

PS/2 使用 IEEE (Institute of Electrical & Electronics Engineering, Inc.) 格式。以浮点数为例, 浮点数值类型有 4 个字节 (共 32bit), 包括: 0 位符号位、1~8 位 2 进

制指数位和 23 位的尾数。如下图所示：



其中指数等于 8 位二进制数减去二进制 1111111 (十进制 127) 的差。

尾数表示 1.0~2.0 之间的一个数，其中 1.0 作为隐含部分，加上 9~31 位表示的部分就得到尾数。浮点数一般表示为  $F \times 2^m$ ，F 为尾数 ( $1.0 < F < 2.0$ ， $-2.0 < F < -1.0$ )。

三、数据转换方法

1. 小型机和微机数据格式的比较与转换

a. 符号位：小型机 (SEL32/77) 和微型机 (PS/2) 相同，可直接传送。

b. 指数位：SEL 32/77 是 1~7 位 16 进制表示，指数是由二进制值减去  $40_{16}$  得到；PS/2 是 1~8 位二进制表示，指数是由二进制值减去  $127_2$  得到。这部分的处理是将 1~7 位的 16 进制数转换为 1~8 位的二进制数。

c. 尾数位：SEL32/77 的 8~31 位是尾数位，表示范围是  $1/16 \sim 1.0$  和  $-1.0 \sim -1/16$ ；PS/2 的 9~31 位是尾数位，表示范围是  $1.0 \sim 2.0$  和  $-2.0 \sim -1.0$ 。

这部分的处理是将 8~31 位表示的数转换为 9~31 位表示的数。

2. 实现小型机 (SEL32/77) 16 进制到微型机 (PS/2) 2 进制的数据转换

下面以浮点数为例来说明数据格式转换的方法。

16 进制表示数为  $(1/16 \sim 1) \times 16^m$  或  $(-1 \sim -1/16) \times 16^m$ ，即  $(1/16 \sim 1) \times 2^{4m}$  和  $(-1 \sim -1/16) \times 2^{4m}$ ，二进制表示数为  $\pm (1 \sim 2) \times 2^k$ 。

设 16 进制表示数为 F。

当  $1/16 < F < 1/8$  时， $\pm F \times 16^m = \pm 2^4 F \times 2^{4m-4}$ ； $1.0 < 2^4 F < 2.0$

当  $1/8 < F < 1/4$  时， $\pm F \times 16^m = \pm 2^3 F \times 2^{4m-3}$ ； $1.0 < 2^3 F < 2.0$

当  $1/4 < F < 1/2$  时， $\pm F \times 16^m = \pm 2^2 F \times 2^{4m-2}$ ； $1.0 < 2^2 F < 2.0$

当  $1/2 < F < 1$  时， $\pm F \times 16^m = \pm 2^1 F \times 2^{4m-1}$ ； $1.0 < 2^1 F < 2.0$

通过以上四种变换可达到二进制表示机器码要求格式。

3. 处理方法及步骤

A. 如果变量的绝对值大于  $3.4 \times 10^{38}$ ，即超出微机表示的范围，则各位均置 1。

B. 如果变量的绝对值小于  $3.4 \times 10^{-38}$ ，即超出微机表示的范围，则各位均置 0。

C. 如果变量值等于 0，微机中 0 的表示为各位均置 0，则各位均置 0。

D. 其它值如下处理：

(1) 首先处理尾数

a. 将变量与 X'00FFFFFF' 相与，结果送寄存器 R，得到 8~31 位尾数；

b. 将寄存器 R 的第一位置 1；

c. 判断寄存器 R 的值并进行处理

I  $1/16 < R < 1/8$  时，逻辑左移 3 位 (实际 4 位，相当于尾数乘 16，因二进制尾数从第 9 位开始)，同时第 8 位置 0。

II  $1/8 < R < 1/4$  时，逻辑左移 2 位，同时第 8 位置 0。

III  $1/4 < R < 1/2$  时，逻辑左移 1 位，同时第 0 位与第 8 位置 0。

IV  $1/2 < R < 1.0$  时，将第 1 位与第 8 位置 0。

(2) 对指数进行处理

将变量与 X'7F000000' 相与，得到指数。将  $R_1$  减去 X'400

00000'，然后将  $R_1$  逻辑左移 1 位 (实际 2 位，相当于指数乘 4，化为二进制码)，结果送寄存器  $R_1$ 。

I  $1/16 < F < 1/8$ ，减去 X'02000000' (指数减去  $2^4$ )，加上 X'3F800000' (即  $127_2$ )。

II  $1/8 < F < 1/4$ ，减去 X'01800000' (指数减去  $2^3$ )，加上 X'3F800000'。

III  $1/4 < F < 1/2$ ，减去 X'01000000' (指数减去  $2^2$ )，加上 X'3F800000'。

IV  $1/2 < F < 1.0$ ，减去 X'00800000' (指数减去  $2^1$ )，加上 X'3F800000'。

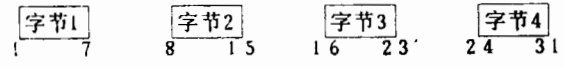
(3) 对符号位进行处理

将原第 0 位的值直接传送过去。

4. 对转换后的二进制码进行字节排序处理

由于 SEL32/77 和 PS/2 内部除数据格式不同外，字节存放顺序也不同，所以要进行字节排序处理。

(1) SEL32/77 的浮点数 (32bit) 表示成二进制字节如下图所示。



(2) PS/2 内部存储表示如下图所示。



即字节存放顺序正好相反。

(3) 转换方法：重新排序。

a. 将字节 1 取出放到相应位置；

b. 将字节 2 取出放到相应位置；

c. 将字节 3 取出放到相应位置；

d. 将字节 4 取出放到相应位置；

e. 将相应位置的 4 个字节重新合并，成为 PS/2 内部机器码的字节表示。

四、实验结果

1. 对于 SEL32/77 十六进制到 PS/2 二进制转换工作是在 SEL32/77 上用汇编语言 (ASSEMBLE) 实现的。

(转 32 页)

## 一种有效的硬盘加密方法

## ——口令码加密

信阳陆军学院实验中心 李日林

在微机的日常使用过程中,有一个令许多用户头痛的问题——自己的硬盘经常被其他人,甚至是操作很不熟练的人“侵占”使用,这样难免会出现许多故障。诸如:硬盘不能启动;硬盘读操作错误;系统文件被破坏等等,更有甚者,自己辛辛苦苦编制的应用程序被破坏掉了。有时候就不得不格式化硬盘,重新装入操作系统。这就使笔者想到:能不能对自己经常使用的微机硬盘进行加密,不让“非法”用户使用,保护硬盘和盘上的重要文件。通过对硬盘(PC/XT)的引导记录、BIOS系统调用及一些软中断分析后,笔者用汇编语言编写了加密保护硬盘的程序(LOCKC·ASM)和设置口令码的解密程序(KEY·ASM)。使用至今,效果比较理想,操作和使用都比较简单,现供有兴趣的读者参考、使用。

## 1. 设计思想:

众所周知,硬盘的引导记录在0柱面0道1扇区。调用系统中断,读取该引导记录,可以看到引导记录的最后两个字节分别为AAH、55H。分析该引导记录可以知道,系统自举时正是通过判断这两个字节的值是否为AAH、55H来确定此引导记录是否有效。如果是有效的引导记录,则启动操作系统进行自举;若是非有效的引导记录,则硬盘的引导记录将启动ROM BASIC。本文的程序设计就是基于这一点。

## 2. 程序功能:

程序一(加密程序):读取硬盘的引导记录,修改引导记录的最后两个字节的值,将修改后的引导记录写回硬盘。这样,“非法”用户想启动硬盘时,就会发现计算机直接进入了ROM BASIC;当他想用软盘启动后再进入硬盘时,屏幕提示“Invalid drive specification”。(错误的驱动器标志字母)

程序二(设置口令码的解密程序),把引导记录的最后两个字节的值改为正常值AAH、55H,然后写回硬盘。在此之前,为了进一步加强保密系数,还设置了一段口令码,这样,“非专家用户”很难进入你的硬盘。

## 3. 程序说明:

程序一较简单,注解也很清楚,不再详述。程序二中的过程INPUT1是一个通过键盘读入一个字符的过程;主程序的开始部分是口令码设置段,具体口令为“1234567”。在运行本程序时,只有键入“1234567”后,程序才确定你是合法的硬盘用户,否则程序不向下执行。这就是说,除该程序设计者以外的其他人不能进入该硬盘。当然,如果你对程序部分进行适当的修改,就可以设置只有你自己知道

的口令码了。该程序段后调用了13H软中断,读取引导记录,恢复最后两个字节为正常值AAH、55H,再写回硬盘。

对程序一、二进行汇编、连接后,就形成可执行文件LOCKC·EXE和KEY·EXE,然后执行LOCKC·EXE对你的硬盘加密,执行KEY·EXE解密。本文程序在PC/XT机上调试通过。

## 4. 程序实例:

## 程序一(LOCKC·ASM):

```

; THIS PROGRAM IS USED TO LOCK THE HARD
-DISK C:
; LOCKC·ASM 1.1 07/10/90 L.R.L
STACK SEGMENT PARA STACK
    DB 128 DUP(?)
STACK ENDS
CODE SEGMENT PARA PUBLIC
    ASSUME CS: CODE, DS: CODE, ES: CODE,
    SS: STACK
BUFFER DB 512 DUP(?) ; 1个扇区为512个字节
    ORG 100H
START: CLI; 清中断
    MOV DX, 0080H
    SUB AX, AX
    INT 13H
    MOV AX, 0201H
    MOV BX, OFFSET BUFFER
    MOV DX, 0080H
    MOV DX, 1
    INT 13H; 读取硬盘引导记录
    MOV AX, 0000H
    MOV WORD PTR BUFFER+1FEH, AX;
    将引导记录的最后两个字节置为00H, 00H
    MOV AX, 0301H
    MOV BX, OFFSET BUFFER
    MOV DX, 0080H
    MOV CX, 1
    INT 13H; 将修改后的引导记录写回硬盘
    INT 18H; 进入ROM BASIC
CODE ENDS
    END START

```

## 程序二(KEY·ASM):

```

; THIS PROGRAM IS KEY TO THE LOCKED HARD
DISK C:
; KEY·ASM 1.1 07/20/90 L·R·L
STACK SEGMENT STACK
    DB 256 DUP(?)
STACK ENDS

```





键, CX 的值减 1; 按 > 键, CX 值加 1), 这一段第 59~83 行。接下来有两种情况需要分别处理, 即绘实线和消隐线。判断这两种情况是由第 85 行实现的。从上述我们知道, 若 d1 中的值为 1, 则绘实线, 此时 d2 标志单元的内容不起作用, 而直接调用 10H 类中断的 12 号功能绘一个小绿点, 并留下该点移动的规迹。这段见第 87~90 行。若 d1 的值为 0, 则绘消隐线, 即不保留点移动的规迹, 此时 d2 标志单元的内容就起作用了。绘消隐线的作法是, 把点当前的位置着绿色, 而其前一个点的规迹着黑色 (实际上只保留一个点)。这样, 首先应该保存当前点的位置, 然后判断 d2 的值为何, 计算出前一个点的位置。此时, 从点的当前位置到其前一个点位置着以黑色, 即把点当前位置的前一个位置上的点消隐掉。此时, 再恢复点的当前位置, 并用绿色着这, 这段见第 87~111 行。这样, 点在移动时只是看见移动的点而看不见其规迹, 其好处是可以清晰地观察到点移动到了何处, 以便决定是否绘实线。

另外，由于消隐线实际上是一条黑色线，故可以把点移动到某一不需要的规迹线一端，使点向另一端移动，即可消隐掉该条规迹线。这同时也为我们的绘图提供了极大的方便。

HT.ASM

```

1 ; 键盘绘图程序
2 stac segment stack
3 db 256 dup(?)
4 stac ends
5 data segment
6 d1 db?
7 d2 db?
8 d3 db F1—实线 F2—消隐线 ESC—退出 * * * '
9 db? ' →右描, ↑上描, ←左描, ↓下描 $ '
10 d4 db ' 键盘绘图程序 Version 1.00 (c) Copyrig'
11 db ' ht Rcx corp 1988 山西长治人行彭起顺 $ '
12 data ends
13 pro segment
14 main proc far
15 assume cs: pro, ds: data, ss: stac
16 start:
17 push ds 37 mov ah, 2
18 sub ax, ax 38 int 10h
19 push ax 39 mov dx, offset d
20 mov ax, data 40 mov ah, 09h
21 mov ds, ax 41 in 21h
22 mov ah, 6 42 mov dx, 100d
23 mov al, 0 43 mov cx, 160d
24 mov cx, 0 44 get:
25 mov dl, 79 45 mov ah, 0
26 mov dh, 24 46 int 16h
27 int 10h 47 cmp al, 1bh
28 mov dl, 1 48 jnz p
29 mov dh, 0 49 ret
30 mov ah, 2 50 p: cmp ah, 3bh
31 int 10h 51 jnz pl
32 mov dx, offset d4 52 mov bl, 1
33 mov ah, 09h 53 mov dl, bl
34 int 21h 54 jmp plot—it
35 mov dl, 1 55 pl: cmp ah, 3ch
36 mov dh, 25 56 jnz plot—it

```

57	mov bl, 0	86	jnz noco
58	mov dl, bl	87	x: mov al, 2
59	piot-it:	88	mov ah, 12d
60	mov al, ah	89	int 10h
61	cmp al, 48h	90	jmp get
62	jnz not-up	91	noco:
63	mov bl, 1	92	push dx
64	mov d2, bl	93	push cx
65	dec dx	94	cmp d2, 1
66	not-up:	95	jnz x1
67	cmp al, 50h	96	inc dx
68	jnz not-down	97	x1: cmp d2, 2
69	mov bl, 2	98	jnz x2
70	mov d2, bl	99	dec dx
71	inc dx	100	x2: cmp d2, 3
72	not-down	101	jnz x3
73	cmp al, 4dh	102	dec cx
74	jnz not-r	103	x3: cmp d2, 4
75	mov bl, 3	104	jnz x4
76	mov d2, bl	105	inc cx
77	inc cx	106	x4: mov al, 0
78	not-r:	107	mov ah, 12d
79	cmp al, 4bh	108	int 10h
80	jnz lite-it	109	pop cx
81	mov bl, 4	110	pop dx
82	mov d2, bl	111	jmp x
83	dec cx	112	main endp
84	lite-it:	113	pro ends
85	cmp dl, 1	114	end start

[illegible]

## 简讯 电脑新花——

## 岭南电脑节

【本刊讯】由中山大学计算机协会、中山大学岭南学院计算机科学系学生会发起的第二届岭南电脑节，已于5月20日在中山大学隆重开幕。

本届电脑节为期2个多星期，内容丰富多彩，包括：

\* 广州高校杯汉字输入比赛: 电脑节期间将举行专门预赛, 选拔同学参加由中山大学、华南理工大学、华南师范大学、暨南大学、广东工学院等学校组织的大赛。

\* 计算机普及知识大赛：采用开卷形式，向全校同学宣传电脑知识，考察同学们的计算机知识，拓展视野，提高兴趣。

\* 软件汇展: 邀请有关单位到校进行计算机及软件演示及介绍, 内容有图形、通讯、排版等等。

\* 计算机普及讲座: 邀请广州地区计算机及应用界的著名学者和知名人士介绍计算机发展、应用状况。

• 新软件推广：开设专门学术讲座和采用上机演示辅导等多种形式，向各位同学介绍国内外新软件的使用知识，进行经验交流和软件交流。

\* 社会参观：组织计算机协会部分成员到一些公司、单位参观，使大家对计算机的应用有个感性认识。

该电脑节还得到社会众多单位的支持和赞助，办得成功。

## IBM PC BASIC 语言矢量字库及其应用

西北轻工业学院计算中心 蔚京生

本文给出了一个由 BASIC 语言生成的矢量字库。它包括了 IBM PC 机上所主要的 ASCII 码字符显示图形。这个矢量库可以通过文中给出的子程序调用,在图形方式下在屏幕的任意行标处画出这个字符集中的任意字符。并可使字符按比例放大,按任意角度旋转,从而解决了用 IBM PC BASIC 语言绘图时图形标注尺寸的困难。

## 一、BASIC 语言图形定义语句。

IBM PC BASICA2.1 是一种扩展的 BASIC 语言,它有一个功能很强的图形定义语句 DRAW。DRAW 语句可以用直线组成各种图形。这取决于它后面跟着的字符串,各字符串的含义如下:

Un: 向上画线                      En: 向右上方画斜线  
Dn: 向下画线                      Fn: 向右下方画斜线  
Ln: 向左画线                      Gn: 向左下方画斜线  
Rn: 向右画线                      Hn: 向左上方画斜线  
B: 仅移动不画线                  N: 移动后回原位

M: x, y 移动到相对座标: 在 x 之前加+或-。

移动到绝对座标: 在 x 之前不加+或-。

An: 设定角度 n, n=0 为 0° 1 为 90° 2 为 180° 3 为 270°

TAn: 设定旋转角度-360~+360; +逆时针, -顺时针

Cn: 点色码省略为前景色

Sn: 信率, 信率因子为 n/4, 省略几时为 4

X: 变量, 在字符串中使用变量时设定

## 二、矢量字库及共应用举例

用 DRAW 语句中的各种字符串组合可以产生一个顺序文件 PCPG.FT1。

\*1, #P PCPG.FT1 矢量字库清单

```
1: "////////////////////"
2: "////////////////////"
3: "CO R8U1L8U1R8U1L8U1R8 D6R5 cl"
4: "BR2UBU2U3BF6", "BM+1, -4U2BR2D2BM+5, 4", "BU2R4BU2L4"
  BM+1, 4U6BR2D6BR5"
5: "BER2EHL2HER2BHD6BR6", "M+4, -6BL3LDRUBM+3, 6ULDR"
  BR4", "BR4H4UEFDG2DFREUBM+5, 2"
6: "BM+1, -4EUBF6", "BR2HU4EBM+6, 6", "BREU4HBM+7, 6"
7: "BE1M+2, -4BL2M+2, 4BM+1, -2L4BM+8, 3", "BU2R4BH2D4BR6"
  , "BREUBM+6, 2"
8: "BU2R4BM+4, 2", "BRRULDBR7", "M+4, -6BM+4, 6", "BUU4ER2"
  FD4GL2HE4BM+4, 5"
9: "BM+1, -5ED6BR6", "BU5ER2FDG4R4BR4", "BUFR2EUHLBRE"
  UHL2GBM+8, 5"
10: "BR3UG3DR4BM+4, 2", "BUFR2EUHL3U3R3BM+5, 6", "BU2E"
  R2FDGL2HU4ER2BM+5, 6"
11: "BU6R4D2G2D2BR6", "BRR2EUHL2HUER2FDGL2GDFBR7",
  "BRR2EU4HL2GDFR3BM+4, 3"
```

```
12: "BERULDBU3RULDBM+7, 4", "BREUBU2ULDRBM+6, 4", "BM"
  +3, -6G3F3BR5", "BUR4BU2L4BM+8, 3"
13: "E3H3BM+8, 6", "BU5ER2FDGLBD3LURDBR6", "BR3L2HU4E"
  R2FD3LHERBF4"
14: "M+2, -6M+2, 6BM-3, -2R2BM+5, 2", "U6R3FDGL3BR3FDGL3"
  BR8", "BM+4, -1GL2HU4ER2FBM+4, 5"
15: "U6R3FD4GL3BR8", "U6R4BM-4, 3R3BG3R4BR4", "U6R4BM-"
  4, 3R3BM+5, 3"
16: "BM+3, -2RDGL2HU4ER2FBM+4, 5", "U6BD3R4BU3D6BR4", "B"
  RRR2RBLU6BLR2BM+5, 6"
17: "BEFEU5BM+5, 6", "U6BR4G4BEF3BR4", "BU6D6R4BR4", "U6F"
  2E2D6BR4", "U6BDF4BU5D6BR4"
18: "BRR2EU4HL2GD4FBR7", "U6R3FDGL3BM+8, 3", "BUU4ER2"
  FD4GL2HBR3FBR4"
19: "U6R3FDGL3BRF3BR4", "BUFR2EUHL2HUER2FBM+4, 5", "B"
  U6RE4BL2D6BR6", "BU6D5FR2EU5BM+4, 6"
20: "BU6D4F2E2U4BM+4, 6", "BU6D5FEUBDFEU5BM+4, 6", "UE4"
  UDL4DF4DBR4"
21: "BU6D2F2E2U2BM-2, 4D2BR6", "BU6R4DG4DR4BR4", "BR2L2"
  U6R2BF6", "BU6M+4, 6BR4"
22: "BU6R2D6L2BR8", "BU4E2F2BF4", "R5BR4", "BM+1, -6FBM+6,"
  5", "BRHU2ER3D4BUGL2BR7"
23: "U6BD2R3FD2GL3BR8", "BRHU2ER2FBD2GL2BR7", "BRHU2"
  ER3BU2D6L3BR7"
24: "BU2R4UHL2GD2FR3BR4", "BU3R3BM+1, -2HLGD5BR7", "B"
  D2FR2EU6L3GD2FR3BR4"
25: "U6BD2R3FD3BR4", "R2BLU4LBM+1, -2UOBN+7, 6", "BD2FR2"
  EU6BU2UOBN+4, 6"
26: "U6BD4M+4, -2BM-2, 1M+2, 3BR4", "R2BLU6LBM+8, 6", "U4BD"
  M+2, -1D4BU3M+2, -1D4BR4"
27: "U4BDM+3, -1FD3BR4", "BRR2EU2HL2GD2FBR7", "R3EU2H"
  L3D7EM+8, -3"
28: "BE4L3GD2FR3BD3U7BF4", "U4BDR2FBM+4, 3", "BUFR2EM"
  -4, -2ER2FBM+4, 3"
29: "BU4R3BH2D5FREDM+4, 1", "BU4D3FR2EU3BF4", "BU4M+2, 4"
  M+2, -4BF4"
30: "BU4M+1, 4M+1, -4M+1, 4M+1, -4BF4", "E4BL4F4BR4", "BD3M"
  +2, -3M+2, -4BL4M+2, 4BR6"
31: "E4L4BD4R4BR4", "BR3HU2LBRU2EBM+5, 6", "BR2U2BU2U2"
  BF6", "BREU2RBLU2HBM+7, 6"
32: "BU5EM+2, 1EBM+4, 6", "////////////////////"
  "////////////////////"
  "////////////////////"
```

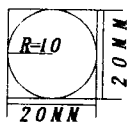
它是一个 ASCII 码字符的矢量字库, 每一个记录为一个字符:

```
!"#$%&'()*+,-./0123456789:;<=>?
ABCDEFGHIJKLMNOPQRSTUVWXYZ(\)
^_`abcdefghijklmnopqrstuvwxyz{|}~
```

使用时可以按顺序文件打开, 读入一数组 A\$ 中, A\$ 数组下标与该字符的 ASCII 码为 A(I) = A(ASC(A\$)) 关系, 根据此表达式可以写一在屏幕任意座标处画字

符的子程序，下面就以一个程序例子来说明其使用方法。

程序 1 首先把 PCPG.FT1 矢量字库打开, 读入一 A\$ (I) 数组中, 然后定义一个视口, 在视口中画一个圆, 和一个正方形见下图。



为圆和正方形标注尺寸时调用了名字为“DRAWCHAR”的子程序，入口参数：XX, YY为显示字符左下角座标，SC是字符放大倍数，TG为字符旋转角度，A\$中为要显示的字符串（数字可用STR\$函数转换）字符数不限，但超出屏幕则不显示，程序在IBM PC机上调试通过。

### 程序一

```

10 DIM A$(29)
20 ' FILENAME=GRAPH0,6.BAS
30 ' INTN FNT
40 OPEN "PCPG.FT1" FOR INPUT AS #2
50 FOR I=1 TO 128
60 INPUT #2,A$(I)
70 NEXT I
80 '
90 SCREEN 2:KEY OFF:CLS:N=5
100 VIEW (1,1)-(500,198),,1
110 X=200:Y=100:R=100:PI=3.141596
120 CIRCLE (X,Y),R,,,5/12
130 LINE -(X,Y)
140 LINE (X-R,Y+R*5/12)-(X+R,Y-R*5/12),,B
150 XP=X+COS(-PI/4)*R
160 YP=Y+SIN(-PI/4)*5/12*R
170 LINE (X,Y)-(XP,YP)
180 XX=XP-50:YY=YP+20:SC=1:TG=48:A$="R=10:GOSUB
310
190 LINE (X-R,Y+R*5/12)-(X-P,Y+R*5/12+10)
200 LINE (X-R+R*2,Y+R*5/12)-(X-R+R*2,Y+R*5/12+10)
210 LINE (X+R,Y-R*5/12)-(X+R+20,Y-R*5/12)
220 LINE (X+R,Y+R*5/12)-(X+R+20,(Y+R*5/12))
230 '
240 LINE (X-R,Y+R*5/12+5)-(X-R+R*2,Y+R*5/12+5)
250 A$="20 MM":XX=X-R+R-20:YY=Y+R*5/12+20:SC=1
:TG=0:GOSUG 310
260 '
270 LINE (X+R+10,Y-R*5/12)-(X+R+10,Y+R*5/12)
280 A$="20 MM":XX=X+R+30:YY=Y+R*5/12*1/4:SC=1
:TG=90:GOSUB 310
285 LOCATE 20,10,0
300 END
310 ' DRAW CHAR
:20 DRAW "BM=XX,,=YY:
330 FOR NN=1 TO LEN(A$)
340 S=SC*4
350 DRAW "S=S;+TA=TG;+A$(ASC(MID$(A$,NN,1)))

```

360 NEXT NN

370 DRAW "BM = XX;,, = YY;.

\*\*\*\*\*

## 用中文 ws 进行高级打印

柳州市统计局计算站 曾庆涛

在打印科技论文时常常需要一些上下标，如数学中的平方、立方或化学中的水、硫酸等等，但是如果您没有排版系统软件而又需要进行这样的高级打印，怎么办？只要您巧妙地使用中文 *ws* 的打印功能就能实现。

下面给出几个用中文 **ws** 进行高级打印的例子, 供没有排版系统软件的用户参考, 同时对广大的计算机爱好者亦是个趣味性的启发!

图一是在屏幕上看见的,图二是在 TH3070 打印机上打印出来的。所用微机为 IBM PC/XT, 操作系统为 CCDOS 4.0, 打印驱动程序为 TH3070E.COM。ΛY 的输入方法为: 按住 <CTRL> 键, 然后连续按 <P> 键和 <Y> 键, 于是在屏幕上可看到 ΛY 标志, 余此类推。

**.OP**

△Y0012

$\Delta D$  高温

$$\text{Fe} + 3\text{H}_2\text{O} \xrightarrow{\text{高温}} \text{Fe}_3\text{O}_4 + 4\text{H}_2 \uparrow$$
$$\Delta D + 2+$$
$$\Delta F \quad \text{Fe} + 2\text{H} \rightleftharpoons \text{Fe} + \text{H}_2 \quad \Delta F \uparrow$$

**△Y0003**

$$\Delta Y0000 \quad U = \frac{1}{4\pi\epsilon} \frac{Q}{\Delta D0\Delta FR}$$

△Y0101

AD 人口普查办公室

△F 广西柳州市△D统计局计算站△F制

图一 输入形式

$$3\text{Fe} + 4\text{H}_2\text{O}(\text{气}) \xrightarrow{\text{高温}} \text{Fe}_3\text{O}_4 + 4\text{H}_2 \uparrow$$
$$\text{Fe} + 2\text{H}^+ = \text{Fe}^{2+} + \text{H}_2 \uparrow$$
$$U = \frac{1}{4\pi\epsilon_0} \frac{Q}{R}$$

广西柳州市人口普查办公室制

图二 打印结果

# 用 TYPE 命令直接分页打印 WORDSTAR 文本文件

航空航天部南方公司 熊锡义

汉字 WORDSTAR 是一个优良的文字处理软件，可用来编辑中西文文件和程序，但其打印输出功能却不尽人意，特别是打印较长汉字文件时，行宽和换页功能难以控制，有些刊物介绍用 BASIC 程序实现分页打印，由于一般采用：

LINE INPUT#文件号，字符串变量名的格式读取 WORDSTAR 中的字符，当文本中一个自然段的字符数超过 255 时，就会出现麻烦。

本文介绍一种可以用 TYPE 命令直接分页打印汉字文本文件和源程序的方法。我们知道，WORDSTAR 存贮的文件，是 ASCII 码的顺序文件，每一个汉字占两个字节，它的 ASCII 码大于 160；数字、字母和标点符号占一个字节，它们的 ASCII 码在 32 至 127 之间，回车和换行分别是 13 和 10，除此之外，WORDSTAR 在编辑文本时，还添加了一些控制字符，正是这些控制字符，使之无法用 TYPE 命令来正常显示和打印 WORDSTAR 文本。

我编了一段 BASIC 程序，先将其文本文件清理，剔除多余的控制字符，通过计数适当添加了一些控制行宽和换页的字符，同时又增加了页头和页尾标记。经过这样对原来的 WORDSTAR 文本文件的处理，生成了一个新的文本文件，就可以直接用 TYPE 命令打印汉字文本和源程序了，使用时十分方便。

程序在 IBM PC 和 286 机上运行通过。

源程序如下：

```
5 REM filename prwl 1990 / 11 / 5
10 N=0: M$=CHR$(10): WS=CHR$(13): RS=""
20 COL=80: PL=25: PG=0
30 TITLE$="航空航天部南方公司工学院微机室"+SPACE$(10)
+DATE$
35 TL=LEN(TITLE$): R1$=STRING$(35,"-")
40 FOR I=1 TO 12: R$=R$+W$+M$: NEXT I
50 INPUT "plear enter filename: "; N1$
60 N2$=N1$+".pr"
70 OPEN N1$ FOR INPUT AS #1
80 OPEN N2$ FOR OUTPUT AS #2
85 GOSUB 200
90 IF EOF(1) THEN PRINT #2, RN$: CLOSE: END
100 A$=INPUT$(1, #1)
105 IF ASC(LEFT$(A$, 1))>160 THEN A$=A$+INPUT$(1, #1): K=K+1
110 IF A$<=CHR$(10) THEN 90
120 IF A$>CHR$(13) AND A$<CHR$(23) THEN 90
130 IF A$>CHR$(127) AND A$<=CHR$(160) THEN 90
140 K=K+1: IF K>COL THEN K=0: N=N+1: PRINT #2, A$; W$; M$;: GOTO 90
150 IF A$<>CHR$(13) THEN PRINT #2, A$;: GOTO 170
160 B$=INPUT$(1, #1): IF B$=CHR$(10) THEN K=0: N=N+1: PRINT #2, A$; B$;
```

```
170 IF N<PL THEN 90
180 PG=PG+1: GOSUB 300: N=0: K=0
190 IF EOF(1)=0 THEN GOSUB 200
195 GOTO 90
200 C$=SPACE((COL-TL)/2): PP$=W$+M$+W$+M$+W$+M$
210 PRINT #2, R$: C$: TITLE$: W$M$; R1$: PP$;
220 RETURN
300 R3$="第"+STR$(PG)+"页": RL=LEN(R3$): LB=(COL-RL)/2+10: RB=(COL-RL)/2-10
305 R2$=STRING$(LB,"-")+ "第"+STR$(PG)+"页 "+STRING$(RB,"-")
310 PRINT #2, PP$; R2$;
320 RETURN
```

下面对程序说明。

20 行是置每行字符数 COL 和每页行数 PL 的初值，30 行是页头标题，DATE\$ 是系统日期，开机系统自然生成；50 行要求输入 WORDSTAR 源文本文件，60 行为新生成的文本文件，它是在源文件之后加后缀 pr 形成；子程序 200 是打印页头标题，子程序 300 是打印页尾标记及换页；其他各句用于清理、添加。 ■

## 《五笔字型》系统下汉字制表符的快速输入

江苏泗阳县委机要室：陈复军

在编制一些表格文件时，汉字 9 区的制表符越来越被用户所喜爱，因其打印的表格比较美观实用。但 9 区制表符一般都只能用区位码方式进行输入，速度慢、效率低，针对这一缺点，不少报刊、杂志上都介绍了在 CCDOS 系统下的改进方法。本文也介绍一种在《五笔字型》系统下，实现部分 9 区制表符的快速输入方法。在使用《五笔字型》系统时，笔者发现，该系统除了能用区位码输入制表符外，还可利用右边的数字键盘直接绘制各种表格。具体方法是：先进入纯中文方式，再进入区位码方式，此时每按动一个数字键就能得到一个制表符。不仅使用方便，而且大大提高了效率。

数字键盘上部分键与制表符的对应关系如下：

0	1	2	3	4	5	6	7	8	9	-
	L	└	┐	┌	+	┐	┌	┐	┐	-

# 内存控制块与 COMMAND 下移的实现

华中理工大计算机系 88 级 殷军

我们知, DOS 系统是用一条内存控制块链掌握当前内存的分配和使用状况的, 对于每个内存控制块, 它的大小为 16 字节 (一节), 其结构如图断所示:

字节: 0	1	2	3	4	5...OF	H
标 记	分配块段号	分配块节数	保 留			

对图所示的各域其含义如下:

## (1) 标记

该域的字节值表明, 本控制块是整个控制块链中的一块还是最后地块, 即:

4DH = 非最后一块

5AH = 最后一块

## (2) 分配块段号

该域的字值说明, 本控制块相应的分配块是已被分配的, 还是空闲的, 即:

0 = 空闲的 (或可用的)

XXXX = 分配块段号 (即程序的 PSP 段址)

## (3) 分配块节数

该域的字值说明, 本控制块相应的分配块的容量 (节长度)。

由此可见, 一旦 DOS 在内存按加载程序的应用要求建立了若干个内存分配块, 在每个分配块前相应设置若干个内存控制块, 由此就形成了一条内存控制块链: 以各块位移, 标记为 4DH 始, 直至最后一块标记为 5AH。

在 COMMAND 程序加载内存之后, 总有最后一块的内存控制块出现, 它控制着当前用户可用的全部内存空间。这样, 当系统加载一文件时, DOS 就开始为用户分配和管理内存。

事实上, 当今大多外壳型的计算机病毒当它们实现病毒常驻内存时, 都进行了内存控制块的再处。采用的手法就是人为地制造内存控制块链的“碎片”, 从而将病毒程序搬到此“碎片”实现常驻。

现在, 我们以最经典的外壳型病毒 V2000 为例, 来看它是怎样修改内存控制块的。

9F01: 017B MOV CX, DS; CX = 当前 PSP 段址

9F01: 017D DEC CX; CX = 当前内存控制块段址

9F01: 017E MOV ES, CX; ES = CX

9F01: 0180 MOV DI, 0002

9F01: 0183 MOV DX, 00FF; DX = V2000 申请内存节数

9F01: 0186 MOV VX, (DI); CX = 当前机器可用内存

9F01: 0188 SUB CX, DX; CX = V2000 常驻后当前机器可用内存

9F01: 018A CMP CX, AX

9F01: 018C JB 01F1; 若病毒常驻将破坏源程序, 则放弃常驻

9F01: 018E POP AX

9F01: 018F ES:

9F01: 0190 SUB (DI+01), DX 修改内存块可用节数

9F01: 0193 MOV (DI), CX; 修改 PSP 段前缀中的当前机器可用内存; 以下实现病毒常驻

由上可看到, 病毒对控制块的操作一般就是修改位移 3 处的那个字, 即分配块节数, 常常减去若干节, 供病毒使用。

V2000 对内存控制块的“再处理”与一般病毒的处理还是有所不同的, 它所直接产生的后果就是使 COMMAND 程序在内存高端的常驻部分下移了!

其实原很简单: 由于内存控制块被修改, 导致系统对内存的控制减少, 从而 COMMAND 文件的常驻程序段址也得到相应修改, 这样当加载程序退出加载时, 原有高端的常驻程序相当于被破坏, 负责重新加载 COMMAND 常驻程序的常驻部分自然将 COMMAND 常驻部分重新定位, 即 COMMAND 被下移了。

我们提出这个方法将对那些常驻内存的程序极为有用。以前, 我们将程序常驻内存的控制权是由 DOS 控制的, 它的缺憾就在于当我们希望原常驻程序不再常驻内存时怎样收回被程序所占有的内存呢? 如果我们采用将 COMMAND 下移的方法实现程序常驻, 就可以做到随心所欲, 不用再担心内存控制块链是否会出现“碎片”的问题。当然不可否认, 采用这一方法将对编程者提出较高的要求。

参考书目: 《DOS 高级技术分析》 张载鸿编

③该程序是针对 11 行中文显示系统设计的, 除最后一行为提示信息行, 可显示行数为 10 行, 程序在顶端一行固定显示查询的数据项名称, 故实际滚动显示的行数为 9 行。稍作修改可适应不同显示行的中文系统。

# 备份文件结构信息的查询

江西拖拉机发动机厂 黄煊如

**内容提要:** 在 DOS 下没有直接查询经 BACKUP 命令的备份文件结构信息的命令, 这给硬盘数据的恢复工作带来了不便, 本文详细地分析了 DOS3.3 版本及其 DOS3.3 以下版本两种不同的结构, 并给出了一个通用的备份文件结构信息的查询程序, 供用户参考使用。

硬盘是一种存储容量大、读写速度快、使用十分方便的外存设备。因此用户都喜欢在硬盘上工作, 大量的数据和程序几乎都存放在硬盘上。为了将程序和数据妥善地保护好, 常用的方法是直接使用 DOS 盘中的 BACKUP 命令将信息备份在软盘上保存起来。DOS3.3 以下版本备份文件的格式基本相同, DOS3.3 版本采用了更有效的备份文件的方式, 其备份文件的格式不同于先前的版本, 这是两种不同的备份文件。

备份文件的结构信息对用户来说是不透明的, 这使查询备份文件中诸如路径、日期等信息倍感不便。尤其是 DOS3.3 版本下的备份文件, 在通常的情况下连备份的文件数量和目录都难以查阅, 这给用户的备份恢复工作带来了困难。本文就此问题对两种备份文件的结构进行分析, 并给出一个用 C 语言编制的通用备份文件的结构查询程序, 供读者参考。

## 一、两种备份文件的结构分析:

1. DOS3.3 以下版本经过执行 BACKUP 命令后, 在每一备份盘上增加了一个名为 BACKUPID.@@@, 文件长度为 80h (均用 16 进制数表示, 下同) 个字节的文件, 用来存放备份盘的序号、日期等信息, 而在每一个文件的前面另加 80h 个字节用来存放备份盘的序号、目录路径等信息。(其中结束标志为 00 表示该盘不是最后一张盘, 否则为 FF, 以下同)

BACKUPID.@@@文件结构

结束标志	盘号	日期等参数	空字节
1	2	3	7 8 80h

附加在每一文件头的结构

结束标志	盘号	空字节	目录路径
1	2	3	5 6 8h

2. DOS3.3 版本经过执行 BACKUP 命令后, 备份盘上建立了两个文件, 一个是包含了所有的文件的 BACKUP.XXX, 一个是保存了目录路径、文件名和其他控制信息的 CONTROL.XXX (其中 XXX 为磁盘顺序号), 这两个文件均是只读文件。

BACKUP.XXX 文件结构

文件 1 内容	文件 2 内容	.....	文件 N 内容
---------	---------	-------	---------

各文件内容所占字节由各个文件的实际长度决定

CONTROL.XXX 文件结构

文件头	文件 1 控制信息	.....	文件 N 控制信息
-----	-----------	-------	-----------

文件头占 D1h 个字节, 以下每个文件的控制信息依次占 22h 个字节

文件头中第 1~9 个字节为特定标志 (8B, "BACKUP", 20, 20), 0A~0B 字节为盘号, 8Bh 字节为结束标志, 8Dh 字节开始为目录路径名 (如为空字节表示为根目录), CCh~CDh 字节为本盘文件数。

备份文件的控制信息中第 1 个字节为特定标志 (22h), 2~0Dh 字节为文件名, 19h~1Ch 字节为文件长度, 1Dh 字节为文件属性, 1Fh~22h 字节为时间和日期。

## 二、通用备份文件结构信息的查询

笔者利用 C 语言编制了 BAUP.C 程序, 该程序在 4.0 版本下编译和链接成 EXE 文件, 可在 DOS 下直接使用。例如:

C> BAUP

该程序提示在 A 驱动器中插入经过 BACKUP 命令备份的软盘, 然后自动寻找 A 盘上的 BACKUPID.@@@ 文件, 如找到显示 DOS3.3 以下版本的备份文件的日期、结束标志等信息, 再提示输入任一文件名后显示路径; 如该盘为 DOS3.3 版本下的备份盘, 则首先提示输入盘号 (用 DIR 命令查文件的扩展名), 然后依次显示备份文件的结束标志、路径、本盘文件数以及全部文件名、字节数和文件属性。

以下为查询两种备份文件结构信息后屏幕上输出的结果: (示例)

### 通用备份文件的结构查询

将备份盘插入 A 驱动器, 按任一键继续!

该盘为 DOS3.30 以下版本经 BACKUP 备份盘, 盘号为: 1

备份日期: 1990 年 11 月 26 日

【查询备份路径】输入任一备份文件名: (含盘号和扩展名) A:HZK24S\213\HZK24S

### 通用备份文件的结构查询

将备份盘插入 A 驱动器, 按任一键继续!

输入备份控制信息文件扩展名: (如 001) 001

该盘为 DOS3.30 版本经 BACKUP 备份的最后一张盘, 盘号为: 1

备份路径为: DOS\DOS1\DOS2

本盘文件数: 3

文件名                  长 度                  属 性

FILEK.COM 721

隐形+系统

FILEKK.COM 1207

只读+隐形+系统

FILEKKK.COM 2417

读写文件

源程序如下:

```

/* BAUP.C 通用备份文件的结构查询 */
/* 适用于 DOS2.X 和 DOS3.X 版本 */
/* 源程序 C 语言 Ver 4.0 */
#include "stdio.h"
main()
{
    FILE *fp; int i=0, j=0, b, f[210];
    char p1[14], p2[14], *q;
    long k1, k[35];
    puts("通用备份文件的结构查询");
    puts("将备份盘插入 A 驱动器, 按任一继续!");
    gets(p2);
    /* ----- DOS3.3 以下版本 ----- */
    if((fp = fopen("a:backupid.@@@@", "rb")) != 0)
    {
        while(i <= 7) { f[i] = fgetc(fp); i++; }
        if(f[0] == 0x1f) {
            printf("该盘为 DOS3.30 以下版本经 BACKUP 备份的最后一张盘, 盘号为: ");
            printf("%d\n", f[1]);
        }
        else
        { puts("该盘为 DOS3.30 以下版本经 BACKUP 备份盘, 盘号为: "); printf("%d\n", f[1]); }
        printf("备份日期: %d 年 %2d 月 %2d 日\n", f[4] * 256 + f[3], f[6], f[5]); fclose(fp);
        printf("【查询备份路径】输入任一备份文件名: (含盘号和扩展名)");
        scanf("%s", p1);
        if((fp = fopen(p1, "rb")) == 0)
        { printf("%s", p1); puts("文件不存在!"); }
        exit(1);
        fseek(fp, 5L, 0); i = 1;
        while(i <= 100) { f[i] = fgetc(fp); i++; }
        i = 1; while(f[i] != 0x00) { printf("%c", f[i]); i++; }
        fclose(fp); exit(1);
    }
    /* ----- DOS3.3 版本 ----- */
    printf("输入备份控制信息文件扩展名: (如 001)");
    q = "a:control.";
    gets(p2);
    strcpy(p1, q); strcat(p1, p2);
    if((fp = fopen(p1, "rb")) == 0)
    { printf("%s", p1); puts("文件不存在!"); }

```

```

    exit(1); }
    i = 0;
    while(i <= 209) { f[i] = fgetc(fp); i++; }
    if(f[0] == 0x8b && f[138] == 0x1f) {
        printf("该盘为 DOS3.30 版本经 BACKUP 备份的最后一张盘, 盘号为: ");
        printf("%d\n", f[9]);
    }
    else
    { if(f[0] == 0x8b && f[138] == 0x00) {
        printf("该盘为 DOS3.30 版本经 BACKUP 备份盘, 盘号为: ");
        printf("%d\n", f[9]);
    }
    else
    { puts("非 DOS 备份盘!"); exit(1); }
    if(f[140] == 0x00) puts("备份路径为根目录.");
    else
    { printf("备份路径为: ");
    i = 140;
    while(f[i] != 0x00) { printf("%c", f[i]); i++; }
    printf("\n");
    }
    b = f[203] + f[202] * 256;
    printf("本盘文件数: %d\n", b);
    fseek(fp, 209, 0);
    printf("文件名      长度      属性\n");
    while(a < b) {
        i = 1;
        while(i <= 34) { k[i] = fgetc(fp); i++; }
        j = 1; while(j <= 12) { printf("%c", k[j]); j++; }
        k1 = k[27] * 65536 + 256 + k[26] * 65536 + k[25] * 256 + k[24];
        printf("%-10ld", k1);
        i = (int)k[28];
        switch(i) {
            case 0x21: q = "只读文件"; break;
            case 0x22: q = "隐形文件"; break;
            case 0x23: q = "只读+隐形"; break;
            case 0x24: q = "系统文件"; break;
            case 0x25: q = "系统+只读"; break;
            case 0x26: q = "隐形+系统"; break;
            case 0x27: q = "只读+隐形+系统"; break;
            default: q = "读写文件";
        }
        printf("%16s\n", q);
        a = a + 1;
    }
    fclose(fp);
}

```



# 计算机爱好者 软件库

计算机爱好者软件库本着质量第一，用户第一的精神在软件交流工作中将开展优质服务。欢迎读者多提意见和建议，只有读者的热心支持，软件库才能办好。欢迎个人、单位来软件库代销各自开发的软件。欢迎个人、单位交换软件。

一、I类软件APPLE类（为保证质量，仅单面拷贝）每片收10元，IBM类软件每片收16元。II类软件按软件后标价计收。个人购买9折收费，邮包费每次收5元。

二、订购软件请注明使用机型、主选目录和备选目录，避免兼容性问题发生给您带来损失。

三、订购软件清单的每个软件后，请注明片数及单价。

四、收到款后一周内寄出。软件寄出后一个月内（凭包裹单邮戳）不能运行的软件可免费退换。超过一个月，如软件损坏，可退回重新复制收费IBM每片10元，APPLE每片7元，另加邮费5元。

五、来信询问软件问题时请注明用户编号及软件编号，并附上贴好邮票的标准回邮信封，请在信封上写清地址和邮政编码。

六、汇款：广州市石牌华南师范大学内电脑杂志社收。

注：方括号内数字为片数，圆圈内J指游戏杆，K指键盘。

## IBM机软件 杂志源程序

P91-3 (1)

### 工具类

IT117 实用软件工具包 (COMPUTER USER'S HANDBAG)

(1) 本磁盘程序精巧实用，功能强劲，能解决用户许多实际问题。这个软件包括有能通过截取 BIO 17H 类中断（打印机中断），把原输出到打印机的数据重定向到磁盘文件的 LPTX 程序；有能使 shift 和任一字母键同时按下，把键盘从 capslock 状态中释放出来的 CAPSUNLK 程序；有协助用户组织和管理多个磁盘文件的索引程序 DISKIDX；有能获取所操作磁盘的面、道、区及其它返回码信息以及 BIO 13 类中断在程序中被引用的地址的驻留内存程序 INT 13H，利用该程序可以帮助用户识别拷贝保护时使用的中断伪装，从而帮助用户进行解密研究；有能捕捉读写盘错误的驻留内存程序 ERRMON；有用于后台通讯程序 DIALER95；有简单实用的正文列表程序 TFL；有通用文件目录管理程序 SDIR50 (V5.0) 及用于通讯软件 TMODEM32 等。

IT118 文件管理实用工具FILE COMMANDER 2.0 (1) 本软件旨在帮助用户更好地管理磁盘上的文件和目录，其主要功能有提示目录显示的功能；快速查找在文件中位置的功能；有能打印或显示文件 HEX 和 TEXT 格式全体或一部分的功能；有强化的文件复制、删除、隐藏的功能以及更好的目录管理的功能。

IT119 死机救星LASTRESO 1.0 (1) (30元/片) 当你用C语言或 BASIC 语言编写程序时或进行文书处理时，可能会因某种原因而发生死机，尤其是在你使用虚拟盘工作，死机所造成的数据损失更大。唯有关机才能获得解脱，但关机则意味着你必须浪费你死机前的一段工作。使用 LASTRESO 能够帮助你不用关机就能重新启动，使你得以保存这批数据。LASTRESO 是常驻内存工作的，它所完成的任务是一般排错程序（诸如 DEBUG 等）所不能做到的，读者如有兴趣，不妨一试。

IT120 能开视窗的编辑软件BLACKBEARD (1) 30元/片。

这是一个适合于程序员使用的编辑器，它能提高程序编写的效率。这个编辑器能打开多至13个的视窗（10个用于正常编辑），能设置前景和背景颜色，设置菜单。总之，这是一个性能卓越的编辑器。

IT121 强化功能的批处理管理程序BATMAN1.1 (1) 这是一个具有较强处理能力的批文件操作管理软件，它包括有下述的功能：利用数字和时间使扬声器发声，产生任何音调、音调的持续和两音之间的间隔；能演奏编好乐曲并有良好的音响效果；能转换显示器模式，可支持 EGA 系统 43 行或 VGA 系统 50 行，能在批处理时对用户回答作出响应；能设置文本的前景和背景色；能在任何情况下在屏幕上写文本以及在文本中设置边线画框等功能。

IT122 一种驻留内存扩充DOS功能的实用程序ZZIP 1.0 (1)

这是一个驻留内存的实用程序，它提供了屏幕暂停、数字钟、屏幕空白、键盘锁定和加速，999 个可编宏指令等功能。

IT123 DESK-LINK (1) 50元/片 阅本期《棒极了》，DESK-LINK》一文介绍。

### 工具类

IT124 SMARTWORK V1.4 印刷电路辅助设计 (1) (30元)

IT125 ORCAD / VST V3.10 逻辑模拟电路设计 (7)

IT126 ORCAD / PCB V3.21 电路板设计 (10)

IT127 TANGO 电路制作 (11) 350元一套

IT128 汉化 AUTO CAD 9.03 制图软件 (要 80287 处理器) (10)

IT129 MAPEDIT V1.0 (划地图用) (2)

IT130 TURBO LIGHTING 英语单词拼写检查 (2)

IT131 TOEFL 英语 (10)

IT132 GRE 研究生英语考试 (4)

IT133 WINDOWS V3.00 AT机, 386机用, 有多种管理功能 (19)

IT134 LQ1600K 配 AUTOCAD 9.03 或 2.6 以上作图的打印驱动程序 (1)

### 文书类

IW18 功能完善的字处理软件NYWORD (2.1) (1) 在上两期软件库中，我们介绍了两种各有特色、功能强劲的字处理器。在本期中，我们再介绍一种功能完善的字处理软件，该软件甚至拥有一些价格昂贵的字处理器不存在的性能，如高质量的多窗口编辑器、连字符 (-)、自动连接单词、栏目的离、合、索引和目录产生器、脚注、多行标题等功能。读者如有兴趣，不妨一试。

### 应用类

IA11 GWOBG 高级通用办公室图形生成软件 (长城机) (4)

IA12 SIDE FICK PLUS 办公室自动化 (4)

IA13 IS 汉化情报索系统 (3)

IA15 SKPLUS V1.00A 文字处理 (4)

IA14 WORKS 三合一制表软件 (6)

IA15 中文程序封面设计 (1)

IA16 中西文蜡纸对折打印 (1)