

电脑 6 1991

中国软件行业协会会刊



3M数据记录产品中国特约经销

中国南海石油联合服务总公司气象服务公司

地址：广州市福今路6号 邮编：510080
电话：7775711 7776918-402 电挂：4466

3M

电脑杂志社科技开发经营部

感谢各厂家及公司的支持和合作

代理计算机产品:

- 1、SUPER 系列 286、386、486 微机
- 2、浪潮系列 LC0530(286)、LC0540(386)微机
- 3、陕西计算机厂 CEC-E 增强型中华学习机
- 4、陕西计算机厂长安系列 286、386 微机
- 5、北京四达 252 汉字语音识别系统、语音对帐系统
- 6、Novell 网络系统及所需计算机配套件

为了方便广大读者维修和自己动手组装扩展微机系统,本部特设如下配件邮购业务:(为了把配件快捷、稳妥地邮寄到您的手中我们采用邮政特快专递,下表示单价含专递费)

品 名	单价(元)	品 名	单价(元)
1.44M 3.5"软驱	530	1.2M 5"软驱	560
360K 软驱	455	720K 3.5"软驱	500
ST-157A 硬盘(40M)带卡	1700	ST-351 硬盘(40M)带卡	1800
超级多功能卡(AT)	260	ST-100M 硬盘带卡	3100
多功能卡(XT)	190	超级多功能卡(XT)	280
单频显示卡	170	双频显示卡	180
调制解调器(300~2400 波特率)	2400	286-16 主机板(带 1MRAM)	940
3M DC600A 60M 磁带	235	60M 磁带机	3850
SUPER PUCII 型汉卡	1900	双星 SS-CEGA 汉卡	1900
TVGA 显示卡	850	SUPER-CEGA 汉卡	1800
光电式鼠标器	580	机械式鼠标器	480
Novell 8 位星形汉字网卡	1200	Novell 8 位星形网卡	750
Novell 16 位星形汉字网卡	1950	Novell 16 位星形网卡	1050
Novell 8 位总线汉字网卡	1300	Novell 8 位总线网卡	820
Novell 16 位总线汉字网卡	2154	Novell 16 位总线网卡	1184
486-33 主机板带 4M 内存, 256KB Cache	8500	386-33 主机板带 4M 内存, 64KB Cache	4800
IBM PS/2 外接 5"驱动器(360K)	2650	80387-33 协处理器	2350
80387-25 协处理器	2250	80387-20 协处理器	2150
80287-10 协处理器	1000	80287-8 协处理器	1000

地址: 广州市德政北路 393 号

电话: 5516911-3273 邮码: 510055

《自然码》广东总代理

——电脑杂志社

欲试用《自然码》请找电脑杂志社

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

欢迎试用、欢迎惠顾、欢迎邮购

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

地址: 广州市天河路科技东街 49 号

电话: 514304 邮编: 510630

将此标志复印剪下寄
来可享受读者优惠价。



长安

CEC-E 中华学习机

陕西就有希望 中华学习机



陕西省计算机公司

地址：西安市友谊东路15号
电话：335008 335201
传真：333943 邮编：710054

国内统一刊号：CN44-1188



广州白云山电源设备厂

CWY 系列高抗干扰稳压电源

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

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

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

单相抗干扰稳压器系列

一、型号及规格

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

二、主要技术参数

- 输入电压单相交流 220V 50Hz
- 电压稳定度

全國範圍產品責任保險

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

- 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 $\pm 100Vp-p < 10 \sim 30ms$
- 有效功率可达到 80~90%
- 音频噪声低，位于 1.6 米距离处 50dB



三相抗干扰稳压器系列

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

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

- 电压稳定度：输入 $-27\% \sim +36\%$ ，输出 $-2 \sim +1\%$ ；
- 音频噪声低：位于 1.6 米距离处 52dB；
- 效率高：89%，比一般电子交流稳压器效率高 20%，这对节能很有意义。

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

厂长：贝远娥

厂址：广州市沙河同和

邮政编码：510515

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

注册商标

质量三包

欢迎来人来函订购

多谢惠顾

高抗干擾
功能特殊
電腦必備
終身保修



机箱电源总汇

深圳航空電腦設備公司廣州辦事處

廣州南秀電腦技術開發服務部

深圳航空电脑设备公司是航空工业部属下的企业，从一九八一年起在全国国内率先专业从事电脑进口、生产和经营。本公司具有雄厚的经济实力和技术力量，在北京、西安已相继成立了经销部，业务范围遍布全国各省，她在广州的分支机构—深圳航空电脑设备公司广州办事处、广州南秀电脑技术开发服务部，根据现时市场需要自行生产 PS/2 及 L 型开关电源，该单位（企业）还与兄弟单位合作、或生产、或进口各款美观、流行的卧式、立式电脑机箱，另配套优质 101 键盘，及 1024×768 高分彩色显示器。所有产品都实行良好的售后服务，以最优惠的价格投放市场，欢迎各届人士选购，批发零售，无任欢迎。

联系地址：先烈南路 1~4 号（烈士陵园后门斜对面）

联系电话：762196

邮政编码：510060

BB 机：672080 CALL 5432

181 CALL 606767

博森企業有限公司

向各用戶、單位致以新年問候

CASPER

显示器系列

19 英寸 1984E

14 英寸 5158

14 英寸单色显示器

UDC

14 英寸单色显示器

主营：PC / XT、286、386、486 兼容机系列及配件

价格适宜、信誉第一

广州市天河区五山路科技街 196 号

电话：5510157 邮编：510630

广州市科教电脑商场进口微机配置及价格一览表

机 型 配 置	CT80286-16				说明
CPU	80286-16	80286-16	80286-16	80286-16	可配协处理器
频率	10 / 22MHz	10 / 22MHz	10 / 22MHz	10 / 22MHz	塔 / 卧式机箱、液晶显示
内存	1MB	1MB	1MB	1MB	可扩展至 4MB / EMS
软驱	1.2M+360K	1.2M+360K	1.2M+360K	1.2M+1.44M	任选
硬盘	40MB	40MB	40MB	100MB	3.5" 高速硬盘
显示	MDA720×350	EGA640×350	TVGA1024×768	TVGA1024×768	进口原装 TOPCON
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 FoxBase+
价格	5500 元	7500 元	8000 元	9500 元	赠送机械式鼠标器
机 型 配 置	CT386DX / 33-64K / 128K Cache RAM CT486DX / 33-128K / 256K Cache RAM				
CPU	80386DX-33	80386DX-33	80486DX-33	80486DX-33	可配协处理器
频率	33 / 58MHz	33 / 58MHz	33 / 150MHz	33 / 150MHz	塔式机箱、液晶显示
内存	4MB	4MB	4MB	4MB	可扩展至 32MB / 64MB
软驱	1.2M+1.44M	1.2M+1.44M	1.2M+1.44M	1.2M+1.44M	任选 (日本 TEAC)
硬盘	100MB	200MB	100MB	200MB	Conner 3104、3204
显示	TVGA1024×768	TVGA1024×768	TVGA1024×768	TVGA1024×768	进口原装 TOPCON
串 / 并	2 / 1	2 / 1	2 / 1	2 / 1	可配 4 / 8 用户卡
键盘	101 键	101 键	101 键	101 键	386 / AT / XT
资料	DOS、随机手册	DOS、随机手册	DOS、随机手册	DOS、随机手册	可按用户要求
软件	XENIX 2.32	XENIX 2.32	XENIX 2.32	XENIX 2.32	DOS4.0 FoxBase+
价格	15000 元	16500 元	20000 元	21000 元	赠送光电式鼠标器
打印机	EPSON LQ-1600K	4400 元	STAR CR 3240(彩打)	5800 元	日本原装
UPS	SANTAK 500W	1500 元	1000W (在线式)	6500 元	美国或日本
主板	80286-16 / 1MB	900 元	80386DX-33 / 4MB	4800 元	台湾产
硬盘	3.5" 高速 40MB 带卡	1550 元	3.5" 高速 100MB 带卡	2700 元	Conner CP3000、3104
软驱	TEAC 1.2MB	450 元	TEAC 1.44MB	400 元	TEAC 360KB 390 元
彩显	14" EGA (5154H)	2300 元	14" TOPCON(1024×768)	2600 元	进口原装 TOPCON

全国各地诚征代理

地址：广州市天河东路商业街东座 25-27 号

通信：广州市 1753 号邮政信箱

邮编：510620

电话：5511197、5515564

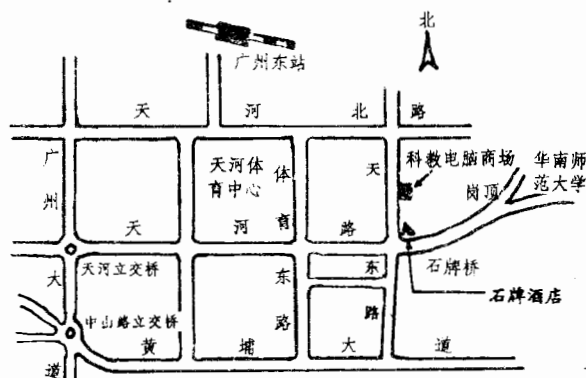
电挂：8360 FAX：5511197

开户银行：广州中国银行天河支行

帐号：271-015110027

代办托运 免收运费 保用壹年

省外用户 来人购机 车票报销



专论

国库业务电脑化的进程与思考 张诚等 (2)

网络通讯

Novell 网络共享汉字的调用 李向军等 (4)

APPLE 与 IBM PC 的单向串行通讯 ... 尹天盖 (5)

软件纵横

山外青山楼外楼——比 dBASE IV 更新的

Foxpro 黎伟强等 (7)

MS-DOS 彻底剖析 (十三)

MS-DOS 层间调用算法的实现 郭嵩山等 (10)

大学生之页

热键激活式 TSR 编程的通用格式 唐 维 (14)

使用与维修

袖珍机新秀—SHARP PC-E500 ... 刘士宏等 (16)

硬盘的软维护 马国忻 (18)

带电话拨打印机电缆造成机器故障的修复

..... 陈永青 (22)

CITIZEN CKP-5240 打印机维修一例

..... 陈祥伍 (23)

计算机辅助教学

以新观点来改进高校计算机教育 杨 成 (24)

中华学习机

CEC-I 中文状态下 CATALOG 的改进

..... 李 锋 (25)

保持在中文状态下操作 陈志岭等 (26)

中学天地

APPLE 图形压缩系统 梁宇翔 (26)

也谈快速随机读取 READ 语句 蔡安军 (27)

一个小发现—6502 的奇怪转移指令 ... 朱 希 (28)

竞赛辅导

炸弹与迷宫 黄东斌 (29)

游戏乐园

牌洗几次最好? 符云清 (30)

单片机演奏乐曲 王 兵 (31)

病毒防治

V2000 病毒的工作原理与清除 殷 军等 (33)

“侵略者”病毒分析及防治 苏武荣 (35)

一种精巧的病毒—世纪病毒 朱 军 (37)

广大一号及其两个变种病毒 胡向东 (37)

电脑用户

四通 2401—微机数据库文书管理系统

..... 林胜藩 (39)

M1724 打印机中 DIP 开关的设置对打印机

的影响 曹 许 (39)

一种通用的外部命令口设置法 易远泽 (41)

谈谈 DOS 与 XENIX 系统共享硬盘 ... 朱晓敏 (43)

提高昌泡排序效率的有效方法 罗国富 (45)

将 CCBios 2.13H 的显示扩展模块联接于其他

汉字系统 李志刚 (47)

单板机与单片机

一种用于 51 单片机防止程序“跑飞”的软件

抗干扰办法 刘志明 (48)

1991 年总目录

..... (49)

广告索引

3M 数据记录产品中国特约经销 封面

长安中华学习机 CEC-E 封底

广州白云山电源设备厂 CWY 系列高抗干稳压

电源 I

机箱电源总汇 II

博森企业有限公司向各用户单位致以

新年问候 III

广州科教电脑商场进口微机配置及价格一览表

..... IV

广州市天河思索电脑公司简介 (28)

袖珍机用户的喜讯 (48)

电脑杂志社科技开发经营部 封二

自然码广东总代理——电脑杂志社 封三

编 辑 《电脑》编辑部

出 版 电脑杂志社

(地址:广州市石牌华南师范大学内)

邮政编码:510631 电话:5516911-3273)

印 刷 韶关二九〇研究所地图彩印厂

总发行处 韶关市邮电局

订 阅 处 全国各地邮电局、所

定 价 1.00 元

出版日期 12 月 25 日

刊 号 ISSN 1002-9613

CN44-1188TP

国库业务电脑化的进程与思考

中国人民银行鄂西自治州分行 张 诚 向圣雄 张 楠

随着经济体制改革的不断深化,国库工作在国民经济中的地位和作用越来越重要,而电子计算机在国库业务中的广泛应用,使国家预算资金的收纳、分成、报解在质量、速度等方面有了明显的提高。实践证明,国库业务电脑化,是业务工作的一项重大改革。但是,对国库业务电脑化现状和存在的问题,有必要进行分析,找出这项业务工作中的矛盾,以便加速国库业务电脑化进程。笔者就此提出一些设想和对策与同行们共同探讨。

国库业务的工作效率是国家机构执行国家职能,维护国家政权机关的存在以及整个国家财政收支执行的决定性因素之一。国库业务核算容量不足、效率不高,是国家预算和财政收支的重大障碍。当前国民经济发展迅速、国库工作要求越来越高,使国库业务成倍增长,传统的手工操作已远远不适应新时期的国库工作要求,在这种情况下,国库业务处理电脑化就成为现实选择。

近几年来,国库电脑化进展迅速,核算质量、报解速度明显提高,全国一、二级电脑通讯网络也初具规模,据初步统计,全国省级分库与总库联网单位占整个省级分库单位的95%以上,中心支库与省级分库联网单位约占全国中心支库单位的60%,而且部分中心支库已经开发或正在开发、筹建以中心支库为中心的三级国库通讯网络,以湖北省鄂西中心支库为例,该支库在电脑开发,运用起步较晚,但电脑开发、利用效益却十分明显。1989年鄂西中心支库开始使用微机处理国库业务并与省分库联网,1990年其下属9个县(市)支库相继开始使用微机处理业务。到今年9月份,鄂西中心支库将与下属9个支库联网,形成以中心支库为中心的三级国库电脑网络结构,笔者在对该中心支库及下属支库手工核算和电脑核算抽样调查的基础上,加权平均得出国库核算工序用机器处理和手工处理所需时间及工作效益的数据对照情况如表一。

表一: 国库核算基本业务工序处理时间对比情况

工序类别	电脑处理	手工处理	提高工效
税票清分、并表	150秒	40分	16倍
查询、检查	3秒	2分10秒	43倍
总分核对	2秒	1分10秒	35倍
记帐	3秒	15分	300倍
综合工作效率	212%	100%	1.12倍

注: 1.表中税票以100张计算,所列均为单人单项操作能手业务工序时间。

2.综合工作效率是综合国库核算所有业务处理的近似值。

据表对比情况可以预料,在国库业务实现电脑化的进程中,国库从日常手工业务、帐务记帐、事后监督的处理,到综合管理和历史资料的存储以及综合分析等工作,都将使用电脑处理,国库核算体系也将发生重大变化,国库部门的条例、制度、规定要求以及工作方式和习惯做法都会有一个较大的变化和较长的适应过程。

国库业务处理电脑化,对我们原来的国库管理程序、业务处理方式、记帐方式以及传统的手工作法是一场新的革命,因此她的发展必然会遇到一些新的问题和困难。

1.国库核算程序研制不尽完善,横向库款通讯网络尚未形成。

尽管使用电脑的工作效率是手工工作效率的数倍,数十倍乃至百倍,但就其整体效率而言,电脑处理只比手工处理提高效率1.12倍(见表一注2)。究其原因,除了会计核算制度和国库规程的制约因素外,从技术上讲,就是至今仍缺乏一种理想而又全面实用的国库业务处理程序系统,在一定程度上限制了电脑功能的发挥。加上有许多业务工序(如编制记帐凭证)在现有的会计核算制度规程方式下是电脑处理非能的缘故。而且,我国目前国库业务微机程序的开发与应用仅局限于上下级国库部门之间的分成、报解(即纵向通讯网络)和国库库款记帐、并表。而税务部门与国库之间、专业银行与国库之间、财政部门与国库之间的收纳和支拨即财、税、行、库四位一体的横向库款通讯网络尚未形成。

从发达国家国库电子化的发展趋势看,这些高技术、多功能、智能化的电脑已作为独立系统支撑着当今国库的核算领域。例如:记帐凭证的编制,税款入库情况以及综合分析,除了调查研究外,无一不借助具有特定功能的现代电子系统。当然,在国外,新型的国库程序开发已经提供了多种实用的处理手段。特别需要指出的是,随着国库电子化的发展,一部分现代化的国库管理方式已经作为特殊的终端设备或子系统直接接入国库的计算机网络。如国库与经收处,财政库款处理设备,使其成为国库电脑处理系统的一个组成部分。

2.传统的思想观念制约国库电脑化的进程

在业务量增长很快的情况下,我们还仍然习惯于原始的作法,即靠增加人和机构去解决问题。近几年来国库业务机构不断扩大,人员不断增长便是一个很好的证明。究其原因,主要是国库干部对电子计算机不甚了解,顾虑

大,习惯于传统的操作方法,认为那样省事而不担风险,再加上计算机设备投资较多,各方面困难较大。从各种附属设备的购置到机房的装修,通讯线路的沟通等,都不容易一下子解决,因而考虑眼前的利益多,想等别人走出成功的路子。有了成熟的经验再上,这种传统观念和懒惰思想制约了国库电脑化的进程。

3、电子计算机知识的匮乏为国库电脑化增加了难度,管理不善使计算机的作用不能得到充分发挥。

有些支库干部虽然热情很高,但由于没有计算机基础知识,误认为计算机买来就能使用,对它的期望很大,希望它能解决一切业务问题,还有人认为微机操作简便,只要输入税票数据,计算机就象变戏法似的可对其自动处理,但实际上由于电子计算机除了需要通讯线路和电力保障外,还要进行二次性工作。因此,实现业务处理电脑化并不如想象中的那般容易。而一旦购买的计算机不能主动投入使用,其热情会很快降到零度以下。认为计算机太神秘,不好掌握,只好将其搁置一旁,加之有些国库干部素质低,过得硬的干部调动又频繁,造成了很大的浪费。

在当前国库业务迅速增长的情况下,使用计算机可以节省大量人员,缓解人员紧张的状况,但是从目前整个国库部门使用电脑后的情况来看,由于人机并行处理业务,人员不但没减少,反而还增加了相应的计算机系统管理员。另外,人们所固有的惰性使得应用电脑后不愿对传统的手工作法进行改革,要求电脑无条件的适应原始操作方法,致使电脑使用后无法真正达到省人节约时间的目的。还有部分国库干部习惯于手工操作的人工方式,担心应用了电脑会使自己无用武之地,这就给国库系统推广、应用电脑带来了一定的阻力。

三

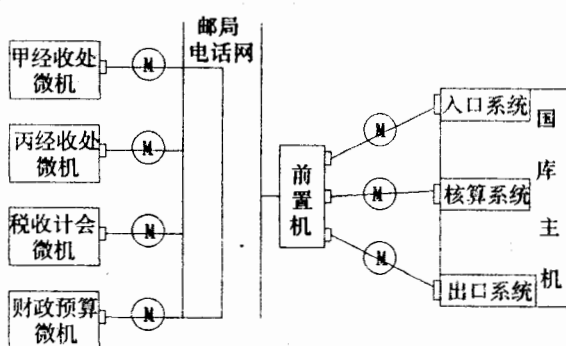
建立财、税、行、库四位一体的库款通讯系统,实现国库电脑横向网络,推动发展国库电脑使用范围,是未来国库电脑化发展的趋势。

笔者认为,在开发、应用国库电脑纵向网络的同时,研制、开发专业行经收处、税务、财政终端系统,不仅可以加速库款收纳、排解,提高核算质量,而且财、税、行、库四位一体充分及时的利用相互间税收、财政资金等信息,更好地为财政资金核算运用决策提供有利的帮助。

其专业行经收处、税务、财政终端系统总体结构设想模式如图所示。

整个系统涉及到主机、前置机和经收处、税务财政预算单位终端机三类计算机。运行会计核算系统、出口系统、入口系统都是本系统主机。前置机是国库主机与终端机部门之间的一个隔离层,使得经收处、财政部门不能直接进入主机造成国库主机的干扰,同时又对数据进行分类和加密,并负责向财政机关发送数据等。

当然,开发一项新型终端系统并非易事,特别是经收处终端系统涉及到的票据交换,凭证传递及银行会计核算制度等,需另作专题研究。



在此,笔者仅对加快国库业务处理电脑化进程提出以下建议:

1、认真抓好国库微机应用规划

国库系统应建立自己的业务处理流程技术规范 and 标准,财、税、行、库密切配合,在国库微机纵向网络的基础上,研制开发国库微机横向网络。在软件开发应用方面,要着重解决各自为政,重复开发的问题。作为一个系统,各国库部门之间的业务往来必须有统一的标准接口(如城市号、帐号、数据长度等格式)。统一的标准和要求,便于各支库之间,财、税、行、库之间数据传递,各部门共享一个数据库。

2、建立有效的故障恢复和事故应急系统

国库在国民经济中占有重要地位,业务上机后不允许因机器发生故障停止营业。因此要求电脑系统长期稳定,可靠运行。在科学技术飞速发展的今天,计算机的可行性和稳定性完全可以满足国库业务处理电脑化的要求。但意外因素(如停电、通讯线路中断等)和人为事故(如操作失误、非法操作等)会使电脑系统受到影响以致陷于瘫痪状态,若造成运行中断,既会打乱国库款数据,又会影响报解速度和质量,故障恢复和应急处理系统是电脑化不可缺少的组成部分。为此,建议国库部门尽快制定故障恢复和应急处理业务的规定和实际办法,以防事故突发时束手无策,造成损失。

3、抓好现有设备的使用,充分发挥其应用的效率,建立“国库计算机更新、应用开发基金”。

计算机投资较大,更新发展又很快,机器闲置不用是很大的浪费。现在有些地市库一方面要钱要指标买机器;另一方面又有许多设备闲置不用,这个问题值得我们国库干部重视,用好现有设备,使其发挥作用。

国库系统使用计算机具有应用面广,使用频率高,连续运转时间长等特点,尤其是并网系统的开通,对设备的更新换代提出了更高的要求,所以,各类计算机及其配套设备的折旧年限就更需要缩短以备急需。因此,财、税、行、库四家应尽快设立“国库微机更新、应用开发基金”,专门用于计算机更新、应用开发,以进一步推动电脑化的进程。

4、培养复合型人才,抓好国库人员的培训

Novell 网络共享汉字的调用

华南师范大学 李向军 郭 涌

近几年来,我国微型计算机局域网信息管理系统在机关、企业、金融、邮电通讯、医院领域已开始得到广泛的应用,根据我国的国情,这些网络系统在中大部分都要使用汉字。汉字操作系统的调用有很多种方法及途径。下文就是介绍如何在 Novell 网络中安装及调用 2.13F/H 共享汉字操作系统。

一、问题的由来

早几年,3com 公司的 3+ 网及 Novell 公司的 Novell 网开始流行时,网络中调用汉字绝大多数是使用汉字显示卡,因为汉卡速度快、占用内存少,因而在当时以 PC 机为主的工作站上使用它,是有一定的实际意义的。但随着科技的迅速发展,市场上价格的变化,汉卡的使用遇到了一些困难,一方面,汉卡的功能始终不如软字库的汉字操作系统强;另一方面,市场中 PC 机及 286PC 机(没有硬驱的 286 机器)的价格一跌再跌,使汉卡在工作站的造价中所占的比例大得难以接受;而且如果用 286 机器,软字库的运行速度不见得比汉卡慢多少。因此,使用软字库在网络中实现共享汉字的呼声越来越高。实际上,现在很多网络工作站已采用 286PC 机,其主板运行速度快,内存为 1MB,如果将汉字库放在设置成虚拟盘的 384KB 扩展内存,其使用则可与汉卡比美,且软字库的成本很低,因此使用共享软字库比使用汉卡更受客户的欢迎。

二、存在的问题

在微机上,很多人都懂得将 2.13F/H 汉字操作系统的字库驻留在虚拟盘上的操作,但如果要把 2.13F/H 汉字操作系统类似在微机上的操作安装到服务器,并在工作站上的虚拟盘调用汉字时,则存在一些问题,这些问题无论在网络手册还是在 2.13F/H 汉字操作系统使用手册,

都找不到解决的办法,这些问题表现为:一方面,如果采用在微机上安装的批文件 ZK.BAT 进行安装,则无法安装到 Novell 网络服务器上,即使修改了 ZK.BAT 文件中有关安装路径的语句,仍然不能成功安装;另一方面,如果先安装在工作站上,然后拷贝到服务器中去,这样虽可以安装到服务器上,但在调用时,则会出 command 丢失的提示后死机。

经过我们对问题的不断探索,终于找到了问题所在,并在新近安装的一个网络工程中得以实施。下面将我们的办法与大家交流。

三、在 Novell 网上安装 2.13F/H 及调用

1. 安装环境:

一台具有软驱及硬盘的 286 机作管理员,用于安装及调试。

其它工作站可以是 286PC 机或 286 无盘工作站以及更高档次的微机。

显示器采用高分单色显示器。

2. 软件实施

(1) 在具有硬盘的工作站上安装 2.13F/H 汉字操作系统。

注:如果受条件限制而没有硬盘,可用 ZKB.BAT 文件在软盘上安装。

(2) 在服务器 public 目录下建立一个名为 213 子目录。

(3) 将工作站硬盘中\213 目录下的所有文件拷贝到服务器中 SYS:\PUBLIC\213 目录中。

(4) 将 2.13F/H 系统里的 #0 号安装盘中 AUTOV.BAT 文件拷贝到服务器中 SYS:\PUBLIC 目录内。

(5) 修改服务器 SYS:\PUBLIC 中的 AUTOV.BAT 文件,源文件的内容如下:

```
1: ECHO OFF
2: CD\213
3: IF NOT EXIST D:\HZK16 COPY HZK16 D:
4: FILE3 D2
5: CCCC
6: CC11
7: INT10F
8: YX1
9: PRTA
10: FILE16B
11: FILE24A 1SFHK
12: FILE40A 1SFHK
```

培养和造就一批懂国库业务又懂计算机应用的复合人才,是国库部门一项长期的战略性工作。国库工作需要计算机专门人才,但更需要一大批复合性人才。从国库现有业务人员中,培养一批计算机应用人才是一条解决和提高国库计算机应用水平的重要途径。从培养复合型人才的方法看,无非有两种:一种是将计算机专业人员培养成懂国库业务知识的人才;另一种是从现有国库业务骨干中培养一批掌握计算机知识的复合人才。而后者潜力要比前者大得多。

在培养造就复合型人才的同时,抓好国库人员培训工作,以适国库业务电脑化的需要。近年来,国库部门举办不同层次,不同类型的中、短期培训班,参加人数达万人次,使之成为国库人员掌握新技术,获取电脑知识的主要途径。

APPLE 与 IBM PC 的单向串行通讯

云南省保山师专物理系 尹天益

由于历史的原因, APPLE 及其兼容机在我国拥有大量的用户, 许多用户单位对该机进行了多年的开发与应用, 设计了一些很好的应用软件并在实际工作中发挥了重要的作用。随着计算机技术的发展和进步以及用户对计算机应用的逐步深入, IBM PC 及其兼容机以其优良的性能价格比和丰富的软件已迅速进入到了众多用户的手中。由于该机显著的优点, 我们迫切希望把在 APPLE 机上运行的一些优秀软件及大量的各类档案数据移植到 IBM PC 及其兼容机上来运行。而要完成这一转换, 实现 APPLE 机与 IBM PC 及其兼容机的通讯是非常必要的, 否则将给用户造成很大的浪费。本文我们将对 APPLE 机与 IBM PC 及其兼容机的通讯作一探讨。

要解决这一问, 最简单的方法莫过于通过两机的 RS-232C 接口卡来实现 APPLE 机与 IBM PC 及其兼容机数据和程序的双向串行通讯。但采用此方法用户必须购置相应的硬件设备, 增加设备投资。而这种数据通讯用户往往只需要单向的 (APPLE→IBM PC) 一次性通讯。在 APPLE 机中有一个独特的 GAME I/O 接口, 它是一个双列 16 位插座, 其中包含有四个一位数字量输出端、三个一位数字量输入端、四个模拟量输入端以及数据选通端、+5V 电源端和接地端, 我们只需利用 GAME I/O 接口中的数字量输出端口, 就可为我们实现 APPLE 机到 IBM PC 的单向串行通讯找到一种廉价的通讯手段。APPLE 机的 GAME I/O 接口中有四个数字量输出端口 (AN0→AN3), 我们只需利用这四个端口中的一个即

可实现 APPLE 机 (利用 GAME I/O) 对 IBM PC 及其兼容机 (利用 RS-232C) 的单向串行通讯。要实现这一设想必需解决的关键问题有两个: 一是使 APPLE 机的输出送往 GAME I/O 数字端口中的某一个; 二是实现两机信号电平的匹配, 第一个问题我们可以用软件的方法来实现 (详见程序), 为了便于传送随机文件和 BASIC 程序, 特用汇编语言编写一子程序, 已在 APPLE 机中模拟类似 IBM PC BASICA 中的 LINEINPUT 语句的功能, 实现对文本文件的整行输入。第二个问题的解决必需借助于硬件采取相应的措施, APPLE 机 GAME I/O 接口数字量端口的开关电平是 0 伏和 3.6 伏, 而 IBM PC 机上 RS-232 卡上数字量开关电平是 -12V 和 +12V, 显然, 要实现这两个电平的转换匹配是非常简单的, 我们可根据自己手头的材料情况采取不同的方法来完成这一转换匹配工作。该通讯系统的联接电路如图 1 所示。

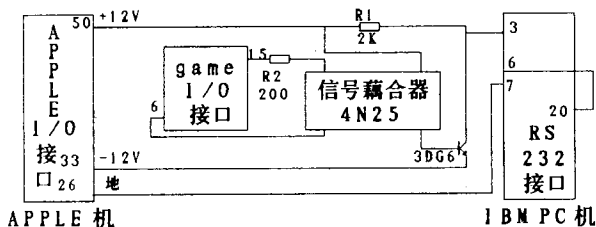


图 1

13: ZF243

14: PATH C: \; C: \213

15: KEY

16: CD\

修改后的 AUTOV.BAT 文件改名为 213.BAT, 该文件的内容为:

1: ECHO OFF

2: CD 213

3: IF NOT EXIST C: \HZK16 COPY HZK16 C:

4: FILE3 C2

5: CCCC

6: CH25; 高分单显显示方式

7: INT10F

8: YX1

9: PRTA

10: FILE16B

11: PATH F: \PuBLIC; F: \PuBLIC\213

12: KEY

13: CD\.

(6) 用有虚拟配置文件的系统盘启动无盘工作站, 入网后在 \PUBLIC 目录下执行 213.BAT 文件, 则在工作站上可调出该系统。

从上面我们看出, 要实现共享 2.13F/H 汉字操作系统软字库的调用, 关键是删除批文件 AUTOV.BAT 中的 11、12、13 语句, 即删去 24 及 40 点阵字库索引, 保留 16 点阵字库。原因是在网络中调用汉字时, 如果将 16 点阵、24 点阵、40 点阵字库同时打开, 则占服务器的内存太大, 以致覆盖内存中 command.com 文件, 从而造成死机。如果只打开 16 点阵字库, 由于其占用内存较少, 不致于覆盖 command.com, 便能将汉字按要求调用。

通过对问题的探讨, 我们了解到在网络中调用汉字的诀窍, 它不仅可以在 Novell 网中实现, 还可以在其他网络中实现, 从而结束了网络中汉字依赖汉卡的历史, 给无盘工作站带来了美好的前景。

该系统可实现 APPLE 机→IBM PC 及其兼容机 BASIC 程序、顺序文件及随机文件的传送。且对被传送文件的结构可在通讯程序运行的过程中加以定义。对 BASIC 程序的传送,必须先对被传送的程序文件进行初始化,即在被传送程序的最前面加入 0 号语句:

```
0 HOME: D$ = CHR$(4): INPUT "PLEASE
INPUT FILE NAME: "; N$: PRINT D$
"OPEN"; N$: PRINT D$; "WRITE"; N$;
POKE 33, 30: LIST1,: PRINT D$; "CLOSE"
```

然后运行该程序以生成一个以 T 类文件形式存盘的待传送程序,为通讯程序传送 BASIC 程序提供数据。

通讯系统的 APPLE 程序如图 2 所示, IBM PC 程序如图 3 所示。使用时可先运行 IBMJS.BAT, 再运行 APPLEFS。若待传送程序或数据文件在 IBM PC 机上将以 AB 为文件名存盘,则运行 IBMJS.BAT 程序时必须键入 IBMJS AB。APPLEFS 程序在运行的各个阶段均有提示。用户可按提示操作。

(图 2) APPLEFS 程序

```
1 FOR I=768 TO 836:READ A:POKE I,A:NEXT
2 FOR I=850 TO 894:READ X:POKE I,X:NEXT I
5 DATA 141,89,192,72,24,32,30,3,142,49,2,162,7, 74,32,30,3,202,208,
249,174,49,3,56,32,30,3,104,208,20,72,144,6,141,89,192,76,42
6 DATA 3,141,88,192,169,10,32,168,252,104,96, 0,201, 141,208,7,
32, 240, 253,169,138,208,195,201,138,208,1,96,76,240,253
7 DATA 32,190,222,32,227,223,162,0,32,117,253,134,253,32,57,213,165
8 DATA 253,32,82,228,162,0,160,2,32,226,229,160,0,165,253,145
9 DATA 131,200,165,111,145,131,200,165,112,145,131,96
10 HOME:VTAB 3:D$ = CHR$(4)
14 CALL -384
16 HTAB 4:PRINT "
17 HTAB 4:PRINT "APPLE——>IBM PC COMMUNICATION"
18 HTAB 4:PRINT "
19 CALL -380:PRINT "PRINT"
20 HTAB 3:PRINT "@@@@@@@@@@@@@@@@@@"
25 HTAB 3:PRINT "@ @ "
30 HTAB 3:PRINT "@ 1.SEND DATA FILE @ "
35 HTAB 3:PRINT "@ @ "
40 HTAB 3:PRINT "@ 2.SEND PROGRAM FILES @ "
45 HTAB 3:PRINT "@ @ "
50 HTAB 3:PRINT "@ 3.QUIT @ "
55 HTAB 3:PRINT "@ @ "
60 HTAB 3:PRINT "@@@@@@@@@@@@@@@@@@:PRINT
70 HTAB 4:PRINT " PLEASE SELECT:;
80 GEY A$:N=VAL(A$)
90 ON N GOTO 200,800,100
95 GOTO 10
100 HOME:NEW:END
200 HOME:VTAB 6:X=FRE(0):REM TEXT FILES COMMUNI-
CATION
202 HTAB 4:PRINT "////////////////////"
204 HTAB 4:PRINT "APPLE——(TEXT)——>IBM PC \/"
205 HTAB 4:PRINT "////////////////////"
206 HTAB 4:PRINT "\/"
```

```
207 HTAB 4:PRINT "\ 1.SEND SEQUENTIAL FILES \/"
208 HTAB 4:PRINT "\ 2.SEND RANDDOM FILES \/"
209 HTAB 4:PRINT "\ 3.QUIT \/"
210 HTAB 4:PRINT "\ \/"
212 HTAB 4:PRINT "////////////////////"
214 HTAB 4:PRINT "PLEASE SELECT:;GET A$:H=VAL(A$)
230 ON H GOSUB 300,500,10
250 GOTO 200
300 REM SEND SEQUENTIAL FILES:X=FRE(0)
305 HOME:PRINT "APPLE——>IBM PC SEQUENTIAL FILES"
310 VTAB 6:INPUT "PLEASE INPUT FILES:"N$
312 PRINT:INPUT "THE FILEDS NUMBER(N)=";N
313 DIM A$(N)
315 PRINT D$;"OPEN";N$:PRINT D$;"READ";N$
316 I=0:ONERR GOTO 325
320 CALL 850,A$(I):I=I+1:GOTO 320
325 POKE 216,0:ER=PEEK(222)
326 IF ER=5 THEN 335
327 IF ER=107 THEN PRINT D$;"CLOSE":HOME:PRINT "":GET
A$:GOTO 300
328 IF ER<>5 AND ER<>107 THEN HOME:PRINT "ER-
ROR";ER:STOP
335 PRINT D$;"CLOSE"
340 POKE -16295,0:PRINT "STRIKE A KEY WHEN READY.....":
GET A$:POKE 54,0:POKE 55,3
350 FOR J=0 TO I-1
360 PRINT A$(J)
370 NEXT J
390 PRINT CHR$(26):POKE -16296,0:POKE 54,189:POKE
55,158:PRINT CHR$(7)
400 RETURN
500 REM SEND RANDDOM FILES:X=FRE(0)
505 HOME:PRINT "APPLE——>IBM PC RANDDOM FILES"
510 VTAB 6:INPUT "PLEASE INPUT FILES:"N$
520 PRINT:INPUT "THE LIGHT FOR FILEDS(L)=";L
540 PRINT:INPUT "THE FILEDS NUMBER(N)=";N
560 DIM A$(N)
570 PRINT D$;"OPEN";N$;"L" I
580 I=0:ONERR GOTO 600
590 PRINT D$;"READ";N$;"R" I
595 CALL 850,A$(I):I=I+1:GOTO 595
600 POKE 216,0:ER=PEEK(222)
610 IF ER=5 THEN 635
615 IF ER=107 THEN PRINT D$;"CLOSE":HOME:PRINT "":
GET A$:GOTO 500
620 IF ER<>5 AND ER<>107 THEN HOME:PRINT "ER-
ROR";ER:STOP
635 PRINT D$;"CLOSE"
640 POKE -16295,0:PRINT "STRIKE A KEY WHEN READY.....":
GET A$:POKE 54,0:POKE 55,3
650 FOR J=0 TO I-1
660 PRINT A$(J)
670 NEXT J
690 PRINT CHR$(26):POKE -16296,0:POKE 54,189:POKE
55,158:PRINT CHR$(7)
700 RETURN
800 REM SEND PROGRAM:X=FRE(0)
```

山外青山楼外楼

—比 dBASE IV 更新的 Foxpro

黎伟强 徐海

一、Foxpro 的崛起

当今计算机世界中,硬件技术的飞速发展是大家有目共睹的,而软件产业亦正步入一个竞争激烈的时代。单从数据库管理系统(DBMS)软件方面即可窥一斑。

被称为计算机信息科学中发展得最快的领域之一的数据库技术,从60年代推出的第一个支持管理系统软件包开始,在数据模型方面迅速地经历了层次模型(如IBM的IMS系统),网状模型(HP的IMAGE)而发展到称为80年代数据库基础的关系模型,就我国情况而言,近年来微机方面应用非常广泛,DBMS软件中使用最多的应推dBASE系统和FoxBASE+系统。dBASE III, FoxBASE+, INFOXIMIX, UNIFY, INGRES 和 ORACLE 等软件已被广泛地应用于教学,科研,财经,企业生产,管理等各个方面。其中FoxBASE+是众多dBASE兼容系统中的佼佼者,主要特点是运行速度快(88年的2.1版声称比dBASE III快8倍),兼容性好(dBASE III程序无须任何修改即可在FoxBASE+下运行),功能更强等。

而现在80年代末90年代初,DBMS软件更进了一步,新推出的几种多用户关系数据库系统,又将DBMS软件推向界面友好,易于管理,提供强功能语言和开发工具的新阶层。它们是Ashton-Tate dBASE IV, FOX公司的Foxpro, Borland公司的Paradox, Revelation的Advance Revelation, Microrim的R: base以及Datacase公司的

Datacase 等等。

其中,88年Ashton-Tate公司公布的dBASE IV具备了支持多用户,包含SQL等特点,而FOX Software公司亦不甘示弱,紧跟在89年推出了它的新颖关系数据库管理系统Foxpro,现在已完善到1.02版。

Foxpro一改dBASE及其兼容系统的传统面貌,率先以全新的集成环境面向广大用户。它是FoxBASE+的版本升级,更吸收了dBASE IV的新特点,并且从各个方面增强了系统功能,据评测和比较,Foxpro在众多的新型DBMS软件中,以性能高,开发工具和使用方便上领先,特别适合于应用开发。

Foxpro的提高之处在于:

- 集成环境式的用户界面,用户无须编程即可以进行DOS服务和数据库的操纵。
- 用户可以极方便地开发出窗口,菜单界面的应用程序。

- 坚持“Nothing Runs Like The Fox”(无比的速度),Foxpro比FoxBASE+快3倍,比dBASE IV快8倍,比dBASE III快16倍,是微机中最快的关系数据库系统。

- 比FoxBASE+增强的近200条命令函数,比dBASE IV增强了近150条命令及函数。特别是大大增强了输入输出语句的功能,用户在检查输入合法性及输出的屏幕设计等编程方面的负担大大减轻。

```
810 HOME: HTAB 6: PRINT "*****"
820 HTAB 6: PRINT "APPLT-->IBM PC PROGRAM"
830 HTAB 6: PRINT "*****"
850 VTAB 6: INPUT "PLEASE INPUT PROGRAM NAME: "; N$
855 INPUT "PLEASE INPUT PROGRAM LINE NUMBER: "; N
858 DIM A$(N)
860 PRINT D$;"OPEN"; N$: PRINT D$;"READ"; N$
870 I=0: ONERR GOTO 890
880 CALL 850, A$(I): I=I+1: GOTO 880
890 POKE 216,0: ER=PEEK(222)
900 IF ER=5 THEN 930
910 IF ER=107 THEN PRINT D$;"CLOSE": HOME: PRINT "":
GET A$: GOTO 800
920 IF ER < > 5 AND ER < > 107 THEN HOME: PRINT "ER-
ROR"; ER: STOP
930 PRINT D$;"CLOSE"
940 POKE - 16295,0: PRINT "STRIKE A KEY WHEN
READY!.....": GET A$: POKE 54,0: POKE 55,3
950 FOR J=0 TO I-1
```

```
960 PRINT A$(J)
970 NEXT J
980 PRINT CHR$(26): POKE - 16296,0: POKE 54,189: POKE
55,158: CHR$(7)
1000 GOTO 10
```

图3 IBMJS.BAT 程序

```
C>TYPE IBMJS.BAT
MODE COM1:24,N,7,1
COPY COM1:%1
```

(本文程序文件名: ZC24.PRG)

参考文献: 1. 《IBM PC 联网原理及应用》 应吉康等编著

2. 《APPLE II 的简易通讯接口》 软件报 90 年

- * 可以不加修改地运行 FoxBASE+, dBASEIII PLUS 程序和绝大部分的 dBASE IV 程序
- * 支持鼠标, 利用 EMS 等并提供编程语句。
- * 支持数学计算, 时间约定等功能。
- * 支持用户程序的窗口式单步跟踪调试, 设置断点。
- * 报表自动生成器。

二、Foxpro 的安装和内容

Foxpro 的 1.0 版本以压缩的形式封装在 5 张 360 KB 的软盘上, 它运行的最低配置是:

- . 内存 512 K 以上
- . DOS 版本在 2.0 以上
- . IBM PC/XT, AT 等或兼容机(需至少 4M 的硬盘空间)

在安装时, 只要打入简单的安装命令即可, 如装向 C 盘时:

A> INSTALL C:

安装完毕后, 你的 C 盘就会有以下的内容:

1. Foxpro 的可执行二进制文件:

- .Foxpro Foxpro 的主模块, 集成环境
- .FOXSWAP 用于加载运行任意大的程序, 如 RUN FOXSWAP PCTOOLS
- .FOXDOC 程序辅助文献(流程图, 交叉访问表等)生成器
- .FOXVIEW 屏幕设计和程序自动生成器
- .FOXCODE 模板语言编译器
- .FOXGEN 与 FOXCODE 一起用于将模板语言生成 Foxpro 语言
- .FOXBIND 过程文件建立辅助器

2. Foxpro 提供的一些说明帮助及安装例行程序文件

- 3. Foxpro 使用的子目录
- DEMO 内含使用 Foxpro 的示范文件
- TEMPLCOD FOXCODE 使用的已编译的中间模板文件
- TEMPLGEN 供 FOXCODE 和 FOXVIEW 使用的模板子模块(include files)
- TUTORIAL 使用 Foxpro 的 TUTOR(示教)
- GOODIES 使用 Foxpro 编得的一些鼠标驱动等例子程序, 以及模拟 FoxBASE+ 环境的工具。
- SAMPLE 使用 Foxpro 的各种命令语句及所含功能的示例程序

其中, 由于 FoxCode, FoxGen 等均与 FoxBASE+2.1 相差不多, 故本文将着重介绍 Foxpro 的新颖之处。

在 Foxpro 中, 以下的后缀为系统默认和使用, 供用户参考:

- PRG Foxpro 源代码文件
- FXP 编译过的可执行文件

- ERR 编译后产生的错误信息文件
- DBF 数据库文件
- FPT 与 DBF 文件 MEMO 字段配合使用的文件
- RSC 系统运行时需要的有关信息数据文件
- LBX 标签文件
- IDX 索引文件
- FKY 出键宏记录文件
- FXD Fox 数据文件
- INC 模板子模块文件
- GEN 源模板文件
- MEM 内存文件
- COD 编译过的模板文件
- FV FoxView 屏幕表格文件

三、Foxpro 的集成环境

进入 Foxpro 后, 屏幕顶端是系统主菜单, 共有七个选项, 第一项呈高亮度显示, 此时光标出现在 COMMAND 窗口中, 相当于传统上的圆点状态。用在此窗口中, 可以键命令来完成要求的动作, 也可以使用菜单驱动(有鼠标的话就更方便了)。

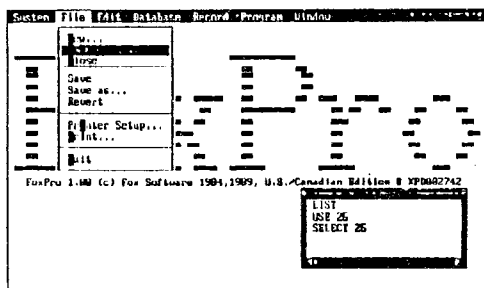


图 1

1. SYSTEM

其下拉菜单中又有十个选项:

(1). About Foxpro

显示 Foxpro 的版本号, 生产厂家等

(2). Help

Foxpro 提供的联机帮助信息, 在一窗口内显示, 提供了所有 Foxpro 的命令及二十多个内存变量的帮助信息, 也可通过在命令窗(Command)内键入 HELP 直接进入此窗, 或 HELP COMMAND(相应的 Foxpro 指令)直接得到此 COMMAND 的信息。

(3). Macros

定义 Foxpro 的键盘宏功能。可对多种功能键及数字字母键的组合(如 ALT-SHIFT, ALT-P)进行重定义, 使其完成指定的一个操作或一系列连续的操作, 并可指定一系列操作的各操作之间的时间间隔, 键盘的宏定义可通过 SAVE MACRO TO filename 命令存入文件(其默认后缀名为 .FKY), 并可用 RESTORE MACRO FROM filename 放入内存而重定义键的宏功能。这在制作自动演示程序的时候十分有用。它是通过记录用户的键击过程而实现的。

(4). Filer

提供各种文件管理操作,如文件的删除,存盘以及对树形目录的操作等。

(5). Calculator

简易的计算器,可进行十进制数的四则运算。

(6). Calendar / Daily

日历,并可用作记事,凡有记事的日子下面都加了一下划线。当选择此项时,系统主菜单将增加一新的下拉菜单 DAILY—提供所有对日历的操作命令。

(7). Special Character

各种特殊的字符,方便用户输入(如各种制表符)。

(8). Ascii Chart

ASCII 表,也可来输入。

(9). Capture

裁剪屏幕文本内容放入 Clipboard,可用 Edit 菜单中的 Paste(见后)放入编辑的文件中。

(10). Puzzle

一个小的游戏

2. File

提供文件的存储,打开,关闭,建立和打印等功能,此菜单各项都与其它流行软件的 File 菜单功能雷同,在此不详述。

3. Edit

主要提供了全屏幕编辑时的一些编辑命令,cut 是将标定的块放入裁剪板中,为以后复制(COPY TO),移动或删除(DELETE)操作所使用。Paste 是将裁剪板(Clipboard)中的内容放入编辑的文件中去。

全屏幕编辑时的键盘操作命令(包括光标移动及块操作命令)详见 TUTORIAL 子目录中的文件 TUTORIAL.TXT。其中, Foxpro 的块操作与流行的 WordStar 不同,见下图:

Shift-箭头(←,→)	选择一个字符
Shift-箭头(↑,↓)	选择一行
Shift-Ctrl-箭头(←,→)	选择从光标开始到单词的头或尾
Shift-Ctrl-End	选择从光标开始到文件头
Shift-Ctrl-Home	选择从光标开始到文件结束
Ctrl-A	选择整个文件

4. Database

此项是对一个数据库文件的各种操作:

建立, Browse(全屏幕编辑), 各种统计操作(如 TOTAL, AVERAGE, SUM, CALCULATE), 并可产生报表(report)文件或标签(label)文件。

Foxpro 的 browse 有其独到之处,它可同时打开 25 个库文件进行编辑,且采用的是表格方式显示,每个数据项分栏表示。每当 browse 一个库文件时,系统主菜单上又增加一个名为 BROWSE 的下拉菜单,通过它,各数据项的域宽可随意改变,看起来十分清晰明了。

5. Record

提供对库文件记录的操作:

增加,删除,恢复,查询定位(如 LOCATE, SEEK, 等)。

6. Program

提供与应用程序有关的各种操作:

DO(D): 运行当前编辑窗内的应用程序, Foxpro 在运行其程序前,若此程序未经 Foxpro 编译过或做过最新的更改, Foxpro 将自动对其运行编译。编译过程中,屏幕中央弹出一长条形窗口,其内显示正在编译的程序名或其子过程名,并以一渐增的亮条表示其编译过程的进行。

CANCEL: 取消当前应用程序的运行,效果与在 COMMAND 窗内执行 CANCEL 语句一样。

RESUME: 恢复执行被中断的应用程序。

ECHO: 相当于在 COMMAND 窗内执行 SET ECHO ON / OFF 一样。若 ECHO 为 ON,则 ECHO 项前有一菱形标志。

STEP: 单步执行,相当于在 COMMAND 窗内执行 SET STEP ON / OFF 一样。若 STEP 为 ON,则 STEP 项前有一菱形标志。

TALK: 设置交互式对话,相当于在 COMMAND 窗内执行 SET TALK ON / OFF 一样。若 TALK 为 ON,则 TALK 项前有一菱形标志。

COMPILE 对应用程序运行编译,生成 Foxpro 的伪编译文件及错误文件。这样在运行应用程序时,无须再进行编译,可大大加快运行速度,真正体现 Foxpro 快捷的特点。

FoxView, FoxDoc, FoxGraph 是 Foxpro 另外配备的软件工具,若它们放在 Foxpro 子目录下,其相应项呈较亮显示,并能在 Foxpro 内加载运行。

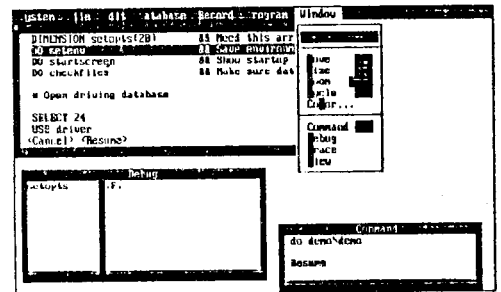


图 2

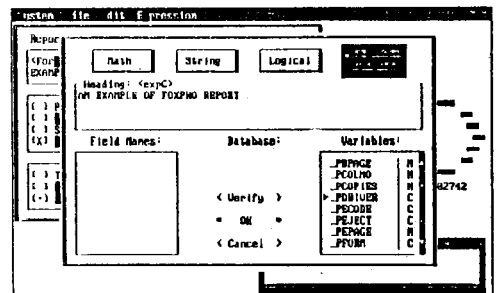


图 3

MS-DOS 彻底剖析 (十三)

MS-DOS 层间调用算法的实现

郭嵩山 李华坚

一、概述

如前所述, MS-DOS 作为一种典型的层次结构的操作系统, 每一层模块是相互独立的。各层模块间具有单向依赖关系, 上层模块对下层模块的调用, 具有统一的入口和出口。DOS 的 COMMAND 模块或用户程序, 在需要执行系统功能调用, 调用 DOS 内核时, 需要先将入口参数(如参加操作有关对象的地址等)送相应的寄存器, 并将功能调用号送 AH 寄存器, 如有子功能号, 则送 AL 寄存器, 然后以 INT 21H 的统一形式, 调用 DOS 内核。在进入 DOS 内核主程序后, 根据所调用的功能号, 查内核中系统功能调用入口地址向量表, 然后去执行相应的功能调用子程序。完成后, 填写返回参数, 将返回信息通过指定的寄存器送回给上层模块的调用程序, 从而实现了 COMMAND 模块或用户程序对 DOS 内核的调用。

而 DOS 内核在运行过程中, 凡需要进行设备操作的, 也都采用统一的调用格式, 通过填写 I/O 请求标题, 执行位于 DOS 内核中的 DOS-BIO 子程序, 实现 DOS 内

7. WINDOW

提供了对 Foxpro 内系统窗口, 用户窗口的各种操作: 隐藏, 移动, 更改尺寸, 扩展屏幕, 更改颜色, 设置阴影等。并在下拉菜单中显示出当前打开的所有用户窗口及系统窗口的堆栈次序, 用户可随意进入其中任一窗。

Foxpro 在调试程序方面比其它 DBMS 先进之处在于: 它增加了 DEBUG, TRACE 和 VIEW 窗口:

DEBUG 窗口可随时显示出你所想知道的程序变量的当前值(变量选择由用户输入)。

TRACE 窗口跟踪执行程序的每条语句, 每执行完一条语句, 只要按一下 "R" (RESUME) 键

即可继续执行下一条语句。

VIEW 窗口是 Foxpro 的重要工具之一, 它提供了打开数据库文件, 建立库文件之间的关系以及设定和更改许多 Foxpro 选项的简便方法。若要在 COMMAND 窗内激活 VIEW 窗口, 可键入 SET 或 SET VIEW ON, 或在 Foxpro 的应用程序中执行 SET VIEW ON 语句。

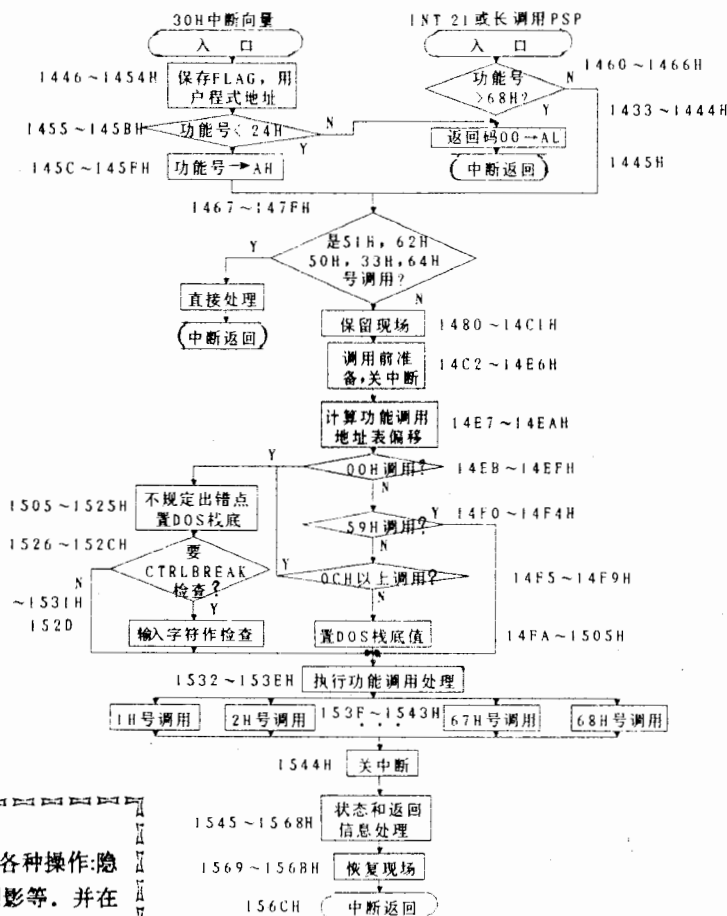


图1 DOS 内核主程序执行流程 (3.3 版)

核对 BIO 模块的调用。在进入 BIO 的设备决策例程和设备中断例程 (BIO 设备驱动主程序), 根据 DOS 内核提出的命令代码, 查位于 BIO 模块的 "命令代码——入口地址表", 并转执行相应的设备驱动程序。执行完后, 填写 I/O 请求标题中的返回状态字, 作为返回信息供 DOS 内核处理。

二、DOS 内核主程序结构

DOS 内核本身由主程序和 21H 类中断的上百个系统功能调用 (3.3 版有 69H 个) 及 00H、20H、25~27H、30H 等几个中断处理程序所组成, 从操作系统上层模块或用户程序进入内核的方式有直接执行 INT 21H 指令方式、长调用 PSP、长调用 30H 中断向量三种方式 (详细请参阅笔者所著《MS-DOS 命令级彻底剖析》一书 P21, 电脑杂

志社 90 年 10 月), 后两种方式都是为了兼容以前版本的, 目前已较为少用。

DOS 内核主控程序是作为内核模块的统一入口, 供上层模块进入。而在 DOS 内核主控程序中, 将所有的系统功能调用处理程序入口地址汇集在系统功能调用入口地址向量表中。下面, 列出流行的三种版本的有关数据, 以便于读者查找。

2.1 版: 主控程序在 B07-BF3H, 入口地址表在 C20-CCFH (00~57H 号调用);

3.2 版: 主控程序在 136C-1499H, 入口地址表在 DEF-EB8H (00~64H 号调用);

3.3 版: 主控程序在 1460-156CH, 入口地址表在 E00-ED1H (00~68H 号调用)。

系统功能调用地址向量表是由 DOS 内核主控程序调用, DOS 内核主控程序根据 COMMAND 模块或用户程序提出的功能调用号查表算出相应系统功能调用入口地址, 然后转执行内核中该功能调用的处理程式, 其计算公式为:

(地址向量表首址+功能号×2) = 某个功能调用入口地址。

下面, 以 3.3 版本为例, 列出该表的内容 (表 1)。

表 1 DOS (3.3 版) 内核系统调用地址向量表

-d26a:e00 edi

```
026A:0E00 99 63 7A 22 83 22 3D 23-5A 23 60 23 C5 21 F1 21
026A:0E10 56 1F BC 1F C8 1F 74 23-7E 23 51 1C 4D 1B 89 2A
026A:0E20 26 24 88 2B 1C 2C C3 23-74 2B 79 2B 5A 2B 98 24
026A:0E30 87 16 48 1B 3D 1B DA 1B-DC 1B 87 16 87 16 20 1C
026A:0E40 87 16 7E 2B 83 2B D7 23-AB 23 6F 1B 02 1D 6F 2B
026A:0E50 6A 2B CF 1B 92 17 AF 17-CF 17 E0 17 8B 19 29 1B
026A:0E60 6D 19 3D 63 22 1C 13 14-08 1C 5F 1B FE 1A 87 1B
026A:0E70 92 19 6E 2E B6 2D F2 2D-BF 6F 00 6F 9A 6C AC 6D
026A:0E80 00 6E 11 70 05 6E CC 6F-92 2E BC 6E DC 6E 4E 2D
026A:0E90 1F 64 3C 65 18 65 FB 5E-75 63 F1 5E 8F 2C E7 2C
026A:0EA0 07 14 0D 14 14 1C 8B 1C-86 19 F4 1C 37 70 6D 6E
026A:0EB0 5C 65 AC 1B 9A 70 7A 70-1F 71 7B 65 E3 67 6C 67
026A:0EC0 3A 6C 87 16 0D 14 87 16-3A 14 19 1A A9 1A E7 6C
026A:0ED0 CD 6C
```

(上述的段址为参考值)。

DOS 内核主控程序主要由内核入口、调用前准备, 执行系统功能调用及调用后的处理四部分组成, 下面分作介绍:

1. 内核入口(1443-1466H)

如前所述, 内核功能调用入口有 3 个, 其中 INT21 或长调用 PSP 的入口地址为 1460H, 30H 中断向量入口地址为 1446H。

2. 调用前的准备(1467-14E6H, 14EB-1531)在执行系统功能调用前, 首先对几种简单的功能调用(如 50, 51, 33, 62, 64H)进行判别, 使其不必进入主控程序, 而直接

进行处理; 然后保留调用前 ES, DS, BP, DI, SI, DX, CX, BX 和 AX 等 9 个寄存器的值; 接着, 要保存原来程序 PSP 段址及栈指针, 并将当前所用的栈转换成 DOS 内部栈, 并撤除中断屏蔽, 以允许中断; 同时对非字符型功能调用作 Ctrl-Break 检查, 根据不同调用设定不同的 DOS 栈底(字符型调用栈底定在 0A06H, 非字符型调用定在 836H)。

3. 执行功能调用

这部分包括根据给定的功能调用号计算其在地址表的偏移(14E7-14EAH)以及取得调用入口地址并执行之(1532-1543H)两部分。

4. 调用后的处理(1544-156CH)

这部分主要执行关中断, 恢复原程序工作栈指针及其存放的单元值, 并保存返回码, 恢复调用前 9 个寄存器的值, 在状态和结果处理完后, 执行中断返回。

为便于读者能深入理解 DOS 内核的工作原理, 下面给出了 3.3 版(其他版本原理类似)的内核主控程序执行流程图(图 1)和程序详细注释清单(见程序 1)。

程序 1

```
-u226:1460 156c
0226:1460 2E          CS:
0226:1461 3A26FF0D    CMP AH,[0DFF]      ;dfth 存放该版本
                                最大调用号
0226:1465 77DC          JA 1443
                                ;系统调用号超
                                出范围,转结束
0226:1467 80FC51      CMP AH,51
                                ;若为 51h,62h,50h,
                                33h,64h 号调用,
                                则不进入主控程
                                序而直接进入
                                各自的入口

0226:146A 74A1          JZ 140D
0226:146C 80FC62      CMP AH,62
0226:146F 749C          JZ 140D
0226:1471 80FC50      CMP AH,50
0226:1474 7491          JZ 1407
0226:1476 80FC33      CMP AH,33
0226:1479 7498          JZ 1413
0226:147B 80FC64      CMP AH,64
0226:147E 74BA          JZ 143A
0226:1480 E8FD00      CALL 1580
                                ;保存 9 个寄存器值
0226:1483 2E          CS:
                                ;DS:BX=CS:
                                [057c]:[057c]

0226:1484 8C1E7E05    MOV [057E],DS
0226:1488 2E          CS:
0226:1489 891E7C05    MOV [057C],BX
0226:148D 8CCB        MOV BX,CS
                                ;CS->DS
0226:148F 8EDB        MOV DS,BX
0226:1491 FE06CF02    INC BYTE PTR [02CF];置 dos 堆栈标志
0226:1495 33C0        XOR AX,AX
                                ;PSP 偏移
0226:1497 A3EA02      MOV [02EA],AX
```

0226:149A A1DE02	MOV AX,[02DE]	;保存当前运行 程序 psp 段址	0226:14FF 7531	JNZ 1532	
0226:149D A3E802	MOV [02E8],AX		0226:1501 BC060A	MOV SP,0A06	;改变 DOS 内 部堆栈指针
0226:14A0 A11E05	MOV AX,[051E]	;保存原程序堆 栈指针	0226:1504 EB2C	JMP 1532	
0226:14A3 A38405	MOV [0584],AX		0226:1506 A3E602	MOV [02E6],AX	
0226:14A6 A12005	MOV AX,[0520]		0226:1509 C606D10201	MOV BYTE PTR [02D1],01	;不规范出错点.
0226:14A9 A38205	MOV [0582],AX		0226:150E C606CE0200	MOV BYTE PTR [02CE],00	;出错处理 标志清 0
0226:14AC 58	POP AX	;AX = 调用号	0226:1513 C606D002FF	MOV BYTE PTR [02D0],FF	;置初始化写 保护盘标记
0226:14AD 50	PUSH AX		0226:1518 50	PUSH AX	
0226:14AE 89261E05	MOV [051E],SP	;保存程序堆栈 指针	0226:1519 B482	MOV AH,82	;执行 2A 类中断
0226:14B2 8C162005	MOV [0520],SS		0226:151B CD2A	INT 2A	
0226:14B6 8E1EDE02	MOV DS,[02DE]	;程序堆栈指针 送 psp 相应 byte	0226:151D 58	POP AX	
0226:14BA 89262E00	MOV [002E],SP		0226:151E C606040300	MOV BYTE PTR [0304],00	;撤消输出重 定向
0226:14BE 8C163000	MOV [0030],SS		0226:1523 BC8608	MOV SP,0886	;修改 DOS 内部 栈指针
0226:14C2 2E	CS:		0226:1526 F606E502FF	TEST BYTE PTR [02E5],FF	;ctrl-break 检 查标志设置否?
0226:14C3 C6060D0500	MOV BYTE PTR [050D],00	;临时盘标志 清零	0226:152B 7405	JZ 1532	
0226:14C8 8CCB	MOV BX,CS	;切换当前堆栈 为 DOS 内部栈	0226:152D 50	PUSH AX	
0226:14CA 8ED3	MOV SS,BX		0226:152E E82D40	CALL 555E	;是作 ctrl- break 检测
0226:14CC BC0607	MOV SP,0706	;置 DOS 栈指针 为 0706H	0226:1531 58	POP AX	;保存返回结果
0226:14CF FB	STI	;开中断	0226:1532 2E	CS:	
0226:14D0 8CCB	MOV BX,CS	;CS → DS	0226:1533 8B9F000E	MOV BX,[BX+0E00]	;BX = 某个功能 调用入口地址
0226:14D2 8EDB	MOV DS,BX		0226:1537 871E7C05	XCHG BX,[057C]	;恢复入口时的 DS:BX
0226:14D4 32FF	XOR BH,BH		0226:153B 8E1E7E05	MOV DS,[057E]	
0226:14D6 883E0303	MOV [0303],BH	;撤消重定向标志	0226:153F 36	SS:	;执行该功能调用
0226:14DA C606040301	MOV BYTE PTR [0304],01		0226:1540 FF167C05	CALL [057C]	
0226:14DF 883EF802	MOV [02F8],BH	;INT 24H 失败 标志清 0	0226:1544 FA	CLI	;关中断屏蔽
0226:14E3 883EF602	MOV [02F6],BH	;非空目录数单 元清 0	0226:1545 2E	CS:	
0226:14E7 8ADC	MOV BL,AH	;地址表偏移量 = 功能号 × 2	0226:1546 FE0ECF02	DEC BYTE PTR [02CF];	置程序栈底标志
0226:14E9 D1E3	SHL BX,1		0226:154A 2E	CS:	
0226:14EB FC	CLD		0226:154B 8E162005	MOV SS,[0520]	;恢复程序工作栈
0226:14EC 0AE4	OR AH,AH	;如为 00 号.59 号调用跳转	0226:154F 2E	CS:	
0226:14EE 7416	JZ 1506		0226:1550 8B261E05	MOV SP,[051E]	
0226:14F0 80FC59	CMP AH,59		0226:1554 8BEC	MOV BP,SP	
0226:14F3 743D	JZ 1532		0226:1556 884600	MOV [BP+00],AL	;保存返回码(将 扩展出错码送入寄 存器堆栈保护区)
0226:14F5 80FC0C	CMP AH,0C	;非普通字符 I/O 转 1506H	0226:1559 2E	CS:	
0226:14F8 770C	JA 1506		0226:155A A18405	MOV AX,[0584]	;恢复调用前的 栈指针单元值
0226:14FA 803ECE0200	CMP BYTE PTR [02CE],00	;非出错处 理转	0226:155D 2E	CS:	
			0226:155E A31E05	MOV [051E],AX	


```

0226:1561 2E      CS:
0226:1562 A18205   MOV AX,[0582]
0226:1565 2E      CS:
0226:1566 A32005   MOV [0520],AX      ;恢复现场
0226:1569 E80100   CALL 156D
0226:156C CF      IRET      ;中断返回

```

三、DOS 内核调用 BIO 算法的实现

在 MS-DOS 操作系统中, DOS 内核模块对 BIO 的调用, 采用了统一的调用格式, 也即以标准接口的方式进行调用。这个标准接口, 就是 I/O 请求标题 (其结构请参看上期)。不同功能的调用, 只是请求标题中不同字段填写不同的内容罢了。下面, 以读设备为例来说明调用 BIO 设备驱动程序的过程。

1、设备调用请求的提出

根据 DOS 的约定, 文件包括设备文件和磁盘文件两种。在 DOS 的系统功能调用中, 读设备的请求是在读文件记录时出现的。其执行过程如下:

1) 根据系统功能调用号, 进入相应的处理程序入口, 经初步的入口处理后进入首址为 28E9H 的记录处理总模块。

2) 在记录处理总模块中经过初步处理 (如建立 FCB 读 (写) 操作参数表, 并将待读 (写) 操作的参数填入表中), 转入首址为 3D14H 的读操作模块。

3) 读操作模块首先判断要读的是设备文件还是磁盘文件, 若是设备文件, 则提出读设备调用的请求。

4) 执行首址为 4E27H 的填写读设备请求标题子程序。

2、I/O 请求标题的填写

1) 在 DOS 内核工作区 (内核偏移 306H 或 324H 处) 开辟一个 I/O 请求标题区, 用以存放所建立的 I/O 请求标题 (本例在 306H)。

2) 在请求标题区内填写 I/O 请求标题的各项, 其中包括请求标题长度、请求部件号、命令代码 (本例读操作为 04H)、介质标志字节、传送数据所在的内存缓冲区地址、字节数/扇区、传送的起始扇区号等, 并将返回状态字段先行初始化。

3) 执行首址为 4E00H 的 DOS-BIO 子程序, 实现内核对 BIO 模块的调用。

在本文最后, 列出了首址为 4E27~4E59H 的填写读设备标题子程序 (见程序 2), 供读者分析、学习之用。

3、调用 DOS-BIO 子程序 (4E00~4E26H)

1) 取相应设备驱动程序的设备标题。

2) 根据设备标题中的偏移地址, 执行 BIO 设备决策例程。

3) 根据设备标题中的偏移地址, 执行 BIO 设备中断例程。

4) 返回状态信息存放在 I/O 请求标题的返回状态字段, 返回到原调用程序供处理。

首址为 4E00H 的子程序是执行各系统功能调用时都要用到的子程序 (请参看程序 2)。

程序 2

```

-u22b:4c00 4c59
022B:4E00 C57407   LDS SI,[SI+07]      ;取设备标题首址
022B:4E03 E840D1   CALL 1F46            ;1F46H 是一条 RET 指令
022B:4E06 8B4406   MOV AX,[SI+06]      ;取设备决策例程地址
022B:4E09 36      SS:
022B:4E0A A32003   MOV [0320],AX      ;存设备决策例程地址
022B:4E0D 36      SS:
022B:4E0E 8C1E2203 MOV [0322],DS
022B:4E12 36      SS:
022B:4E13 FF1E2003 CALL FAR [0320]      ;执行设备决策例程
022B:4E17 8B4408   MOV AX,[SI+08]      ;取设备中断例程地址
022B:4E1A 36      SS:
022B:4E1B A32003   MOV [0320],AX      ;存设备中断例程
022B:4E1E 36      SS:
022B:4E1F FF1E2003 CALL FAR [0320]      ;执行设备中断例程
022B:4E23 E828D1   CALL 1F4E            ;1F4EH 是一条 RET 指令
022B:4E26 C3      RET
022B:4E27 57      PUSH DI
022B:4E28 51      PUSH CX
022B:4E29 50      PUSH AX
022B:4E2A B104     MOV CL,04            ;置命令码为 04H,即读操作
022B:4E2C B016     MOV AL,16            ;I/O 请求标题长为 16H byte
022B:4E2E 16      PUSH SS
022B:4E2F 07      POP ES
022B:4E30 BF0603   MOV DI,0306          ;从内核 306H 建立请求标题
022B:4E33 AA      STOSB      ;填请求标题长度
022B:4E34 58      POP AX      ;填部件号
022B:4E35 AA      STOSB
022B:4E36 50      PUSH AX
022B:4E37 8AC1     MOV AL,CL            ;填命令码
022B:4E39 AA      STOSB
022B:4E3A 33C0     XOR AX,AX            ;状态字初值填 0
022B:4E3C AB      STOSW
022B:4E3D 83C708   ADD DI,+08          ;跳过 DOS 保留区
022B:4E40 58      POP AX
022B:4E41 86E0     XCHG AH,AL          ;填介质描述符
022B:4E43 AA      STOSB
022B:4E44 86C4     XCHG AL,AH
022B:4E46 50      PUSH AX
022B:4E47 8BC3     MOV AX,BX            ;填传送地址偏移
022B:4E49 AB      STOSW
022B:4E4A 8CD8     MOV AX,DS            ;填传送地址段址
022B:4E4C AB      STOSW
022B:4E4D 59      POP CX

```

热键激活式 TSR 编程的通用格式

成都科大水利系办公室工民建 89 级 唐 维

驻留内存程序 (Terminate Stay Resident, 简称 TSR) 属于 MS-DOS 的高级编程范畴。TSR 是这样一种程序, 它一旦驻留内存后就被挂起, 直到触发条件被满足后, 立即中断前台任务, 完成特定的功能再恢复前台任务。

正是 TSR 这种快速切换前台任务的特点, 使本质上是单用户单任务的操作系统 MS-DOS 带上了多进程的色彩。

而触发 TSR 的一种简洁手段就是设置“热键”, 由操作者在需要的时候按下热键, 激活后台任务。所以有许多 TSR 都采用这种热键激活方式, 包括著名的 Turbo

Borland 公司的 SideKick 软件; 更有许多计算机爱好者想涉足这块领域。为此, 笔者将自己的一点实践心得体会献给大家, 以供参考。

设置热键, 相应就要扩展键盘中断, 使之能处理热键, 并包含后台任务。这样在大多数情况下, 只要系统允许中断, 不管当时机器运行的是什么程序, 程序中是否在读键盘信息, 按下热键就能起到某种作用。

但是, 正如上段所述, 在大多数情况下是如此, 而在有些情况下, 都必须避免激活 TSR。这正是 TSR 编程的关键所在。所以笔者先给出“热键”启动方式的流程图, 然后对照流程图, 将一些关键点作些解释。

一、进入具有热键处理能力的 INT 09H 入口, 原入口地址应被保存, 用于被新例程调用链接。

二、当判断热键被按下后, 接着连续判断另外三个激活 TSR 应满足的条件, 顺序如流程图。

1、后台程序是否已被激活, 如果正在进行服务, 就不能再调用它, 重复调用会造成系统混乱。具体实现就是设置一个后台任务状态标志 BUSY 被挂起时 BUSY=0, 正在运行时, BUSY=1。

2、前台程序是否正在请求 BIOS 磁盘服务。因为系统中磁盘驱动器资源是不可共享的, 一个时刻只能由一个进程使用它, 否则会造成 BIOS 重入, 它与 DOS 重入一样会导致系统崩溃。

要判断当前磁盘是否正在活动, 就要修改磁盘中断 INT13H, 增加磁盘活动状态标志。

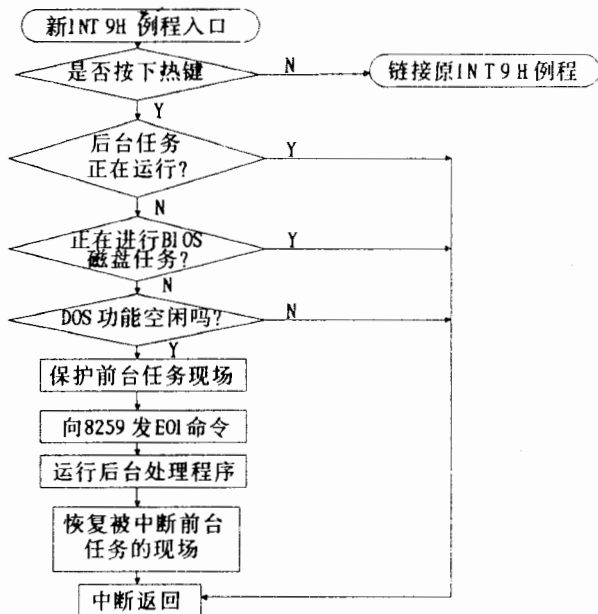
3、紧接着就是对避免 DOS 重入的实现。DOS 重入是 TSR 编程中最令程序员头痛的问题。DOS 代码不可重入是指系统核心不可在任意点处中断后使其代码为另一个进程重新使用。

所以避免 DOS 重入的关键之处就是如何判断当前 DOS 是否空闲。具体实现就是在安装 TSR 程序时用 DOS 中未载入文档的功能 34H, 获得 DOS 临界状态单元指针, 保存在 TSR 的内存单元 INDOS-PRT 中; 那么判断 DOS 是否空闲就是查看 INDOS 所指向的临界状态字节单元的值是否为零。

上述几个主要问题的编程逻辑, 请读者参考笔者在最后附上的热键激活式 TSR 编程的通用格式体会。

下面再谈谈扩展键盘中断这一类 TSR 所涉及的键盘原理:

IBM PC 的键盘是通过中断控制器 8259 和 8255 通用 I/O 芯片的 PA、PB 口与主机接口的。在一次按键输入时, 产生一个“通码”, 置于 8255 的 PA 口; 放开按键时送“断码”, 其值为“通码”加 80H; 按键或放开键时都同时要



```

022B:4E4E 58      POP  AX
022B:4E4F AB      STOSW      ;填字节数/扇区
022B:4E50 92      XCHGDX,AX
022B:4E51 AB      STOSW      ;填起始扇区数
022B:4E52 91      XCHGDX,AX
022B:4E53 87D1    XCHGDX,CX
022B:4E55 5F      POP  DI
022B:4E56 B80603  MOV  BX,0306 ;BX 指向请求标题首址
022B:4E59 C3      RET
  
```

(本文程序文件名: ZF33.PRG)

参考资料:

1、郭嵩山, 陈文彬: DOS 内核探秘 香港《微电脑技术》1989 年 10~11 月号

向 8259 发键盘硬中断, 8259 收到键盘硬中断后, 如果当前允许中断, 它就向 8086/8088 发出 INT9H 中断请求信号, INT9H 的中断处理程序从 PA 口得到通码或断码, 经转换后送到键盘缓冲区。

所以用户应在这里扩展热键处理功能。首先判断是否为通码, 如果是通码, 再判断是否按下了热键, 以决定是否转入用户的热键处理程序还是链接原键盘中断例程。

最后在新例程中还应注意, 因为 INT9H 是硬件中断, 而对任何硬件中断都应在每次中断后重新允许 8259 中断控制器工作, 即向 8259 发结束中断 (EOI) 命令, 否则 8259 芯片就不会清当前服务寄存器, 这样会导致中断在较低优先级上重复调用。

所以根据上述原理, 笔者设计了一个扩展 INT9H 的通用格式。

附: 热键激活式 TSR 的通用编程结构 (ZF35.PRG)

CODE SEGMENT

ASSUME CS:CODE,DS:CODE

ORG 100H

BEGIN:JMP INSTALL

INDOS-ptr DD ? ;DOS 临界状态单元地址
指针

INBIOS DB 0 ;BIOS 磁盘服务状态标志

BUSY DB 0 ;TSR 是否已被激活标志

INT090 DD ? ;INT09H 的原中断向量

INT130 DD ? ;INT13H 的原中断向量

INT09N PROC NEAR

STI

PUSH AX

PUSH ES

XOR AX,AX

MOV ES,AX

IN AL,60H ;取当前 8255 的 PA 口
的值

TEST AL,80H ;是“通码”码?

JNE ORGINT9 ;不是

MOV AL,BYTE PTR ES:[417H];

AND AL,CS:HOTKEY;

CMP AL,CS:HOTKEY ;按下热键?

JNE ORGINT9 ;没有链接原例程

CMP CS:BUSY,0

JNE IRET1

CMP CS:INBIOS,0

JNE IRET1

PUSH BX

LES BX,CS:INDOS

CMP BYTE PTR ES:[BX],0

POP BX

JNE IRET1

PUSH BX ;保护现场

PUSH CX

PUSH DX

PUSH BP

PUSH SI

PUSH DI

PUSH DS

IN AL,61H ;取当前 61H 口的值

PUSH AX

OR AL,80H ;向键盘发

OUT 61H,AL ;应答信号

POP AX

OUT 61H,AL

MOV AL,20H ;向 8259 发

OUT 20H,AL ;EOI 命令

CALL PROCESS ;调用中断处理程序

POP DS

POP DI

POP SI

POP BP

POP DX

POP CX

POP BX

IRET1 POP ES

POP AX

IRET ;中断返回

ORGINT9 POP ES

POP AX

JMP CS:INT090 ;链接原 INT9H 程例

INT09H ENDP

INT13N PROC NEAR

MOV CS:INBIOS ;设置 BIOS 磁盘活动
标志

PUSHF

CALL CS:INT130

MOV CS:INBIOS,0 ;取消 BIOS 磁盘活
动标志

INT13N ENDP

PROCESS PROC NEAR

RET

PROCESS ENDP

INSTALL:PUSH CS

POPDS

MOV AH,34H

INT 21H

MOV WORD PTR INDOS,BX

MOV WORD PTR INDOS+2,ES

CLI

袖珍机新秀—SHARP PC-E500

广州袖珍计算机技术服务中心 刘士宏 卢耀权

袖珍计算机在我国生产实践中,应用广泛,广大袖珍机用户编制了大量的行之有效的应用软件。在袖珍机中最受欢迎的是 SHARP PC-1500 机,但两年前厂家已停止生产此机种了,而国内仓库积压的产品,最近一年已接近全部销完,其他型号的袖珍机;如 HX-20, pb-700 亦接近尾声。在这种情况下,寻找一种可替代 PC-1500 机的产品,成了生产实际的迫切需要。广州袖珍机技术服务中心,最近二、三年来,探索了多种袖珍机,包括便携机在内,结合我国国情,对机子进行全面的性能价格比较后,现特向国内袖珍机用户推荐 SHARP PC-E500 袖珍计算机。我们认为该机种是适合原 PC-1500 机用户选择的一种现实机种。

一、PC-E500 简介

CPU : 8 位

菜单:有五种选择,即 BASIC 语言、计算、矩阵、统计、工程计算(统计、编辑、数学计算、科学计算、工程计算)。

LCD : 四行显示,每行40字符,点阵: 240×32

键: 89个,其中预存键10个

程序语言: BASIC

内存: ROM 256K, 主机 RAM 32K, 其中系统占 3.8K, 固定变量 312 字节, 用户内存 28K, 最大内存可扩至 96K, 有外插各种

```
MOV AX,3509H
INT 21H
MOV WORD PTR INT090,BX
MOV WORD PTR INT090+2,ES
MOV AX,3513H
INT 21H
MOV WORD PTR INT130,BX
MOV WORD PTR INT130+2,ES
MOV AX,2509H
LEA DX,INT09N
INT 21H
MOV AX,2513H
LEA DX,INT13N
INT 21H
MOV DX,OFFSET INSTALL
INT 27H
```

CODE ENDS

END BEGIN

注: 其中的 PROCESS 过程为通用的子程序, 只有一条返回指令。用户应根据具体需要进行编写。 ■

RAM 卡供选择, CE-212M 8K,
CE-2H16M 16K, CE-2H32M 32K,
CE-2H64M 64K。

输出: 10 尾数+2 指数(单精度)

20 尾数+2 指数(双精度)

接口: 11 芯, 可插接磁带、磁盘、打印机的接口等。

电源: 4×AA (6V)

功耗: 0.07W

工作温度: 0℃~40℃

尺寸: 200 (长)×100 (宽)×14 (高) mm

重量: 250 克 (含电池)

内装函数: 1101 个, 其中 124 个常数、744 个运算与数据、233 个工程软件。

BASIC 部分:

命令: 53 条

函数: 60 个

运算符: 16 个

机器语句: 3 条

可选用: (当接打印机、磁盘或磁带机时增加):

打印机(绘图机) 24 条型号 CE-515P

磁带机 22 条型号 CE-126P

磁盘 9 条型号 CE-140F

函数部分:

函数: 32 条

矩阵: 20 条(逆阵、矩阵传送、平方阵、等值)

其他: 12 条

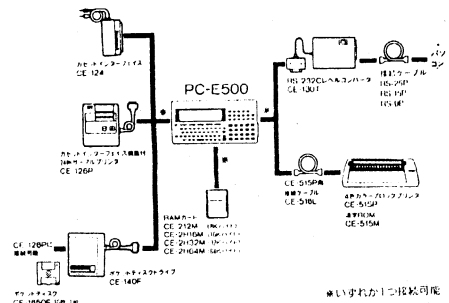
统计: 18 条

二、PC-E500 的几个特点(与 PC-1500 比较)

(1) 积木式结构。

PC-E500 象 PC-1500 机一样, 仍是积木式结构。核心是 PC-E500 主机, 主机外围可接打印机、磁盘机、绘图机、磁带机, 见下图。

■PC-E500 システム構成図



※いすれか1つ接続可能

图 1

(2) PC-E500 内存较 PC-1500 机内存增大了, 众所周知 PC-1500 内存最大为 28K, 而 PC-E500 内存最大可扩至 96K, PC-1500 地址从 0000~FFFF, 而 PCE500 地

址从 00000~FFFFF, 即寻址范围到 1 兆 (1500 机是 64K), 1 兆寻址范围将为开发 E500 更加展示了广阔的前景。

(3) PC-E500 菜单较 PC-1500 机丰富, 大家知道 1500 机具有 PRO, RUN 和 RESEVER 三个模式, 而 E500 则具有六个操作模式, 分述如下:

(A) BASIC 模式, 内包含 PRO 程序和 RUN 运行模式。

(B) CAL 计算模式, 主要作为一般计算器使用。

(C) MATRIX 矩阵模式, 对矩阵可进行加、减、乘、转置和求逆等运算。

(D) STAT 统计模式, 对样本进行数值个数, 总和, 平均值, 平方和, 标准偏差和离散度以及回归分析计算。

(E) ENG 工程模式, 包含了数学、科学、工程、统计和编辑五大部分的软件, 软件一揽见左表:

(F) AER 代数表达式保存模式, 可用来保存各种数学公式, 亦可进行计算, 也可用来保留通讯人姓名、地址、电话号码等。

(4) 增加了文件管理功能, 1500 机文件管理功能是很差的, 除了用磁带机进行存取及连接文件外, 似乎还谈不上对文件进行管理了。但 E500 则不同, 它增加了 INIT, FILES 等命令, 可将内存划出一个文件管理区, 用于存放程序和数据。

例: INIT "E: 10K" ↓ 将用户区划出 10K 作为保存文件区使用。

SAVE "E: ABC" ↓ 将 ABC 文件存入文件区中。

FILES "X: ABC" ↓ 从磁盘中查 ABC 文件。

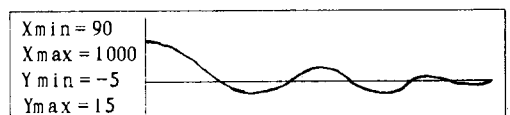
LOAD "F: ABC" ↓ 从扩展模块中将 ABC 文件调入用户区。

有了这个文件管理功能, 将为野外工作者采集数据与可靠保存数据信息提供了极大方便。

(5) 主机 E500 本身带有一个串行口, 这个串行口, 将为 E500 直接与红外测距仪, 电子经纬仪, 以及具有 RS-232C 串口的微机、仪器仪表和绘图机相连接, 进行数据流交换, 带来便利。这样一来, 实质上 E500 就起到电子记事本的作用。

串行口波特有: 300, 600, 1200, 2400, 4800, 9600 六种, 输出 CMOS 信号电平 4~6V。

(6) 增强了 LCD 液晶显示屏的功能, 1500 机是 LCD 单行 7×156 点阵显示, 可同时显示 7×5 的字符 26 个。而 E500 则是 4 行 240×32 点阵显示, 每行显示 40 个字符。由于屏幕加大到 4 行, 因此方便汉字提示显示, 以及 LCD 屏幕图案演示。下面为 LCD 演示的一种图案。



(Graph is displayed.)

名称	程序名		
一 PF1 MATH 数学	PF1[INT] 整数	PF1PRIME	质因子
	PF2[EQU] 方程	PF2GCMLCM	最大公约数和最小公倍数
		PF1 CVBIC	三次方程
		PF2NEWTON	解行列式(牛顿法)
	PF3DI&INT 微积分	PF3BISECT	解行列式(平分法)
		PF1INTGRL	龙贝格积分
		PF2DIFF	龙格-库塔微分
	PF4 FORM 公式	PF3INTPLT	
		PF1 FACT	因子分解公式(26个)
		PF2 TRIG	三角函数公式(49个)
		PF3 INTFRM	积分公式(42个)
	PF5 GRAPH 图形	PF4 GREEK	希腊字母读音表
		PF1 FUNC	函数图形
		PF2 DATA	数值图形
	PF1 FONC	PF3 FIGVRE	图形面积
		伽马函数	
二 PF2 SCI 科学		PF1 PHYS 物理	PF1 CONST
	PF2 CHEM 化学	PF2 M-CONN	度量衡制转换
		PF3 MOTION	方程式转换 21
		PF1 PERIOD	化学元素表
	PF3 EARTH 地球科学	PF2 ELECTR	外层电子
		PF3 ISOTCP	同位素表
	PF4 BIC 生物学	PF1 METEO	气象学
		PF2 PLANET	太阳 / 行星表
三工程 PF3 ENG	PF1 ELEC 电子工程	PF1 AMINO	核酸公式
		PF1 COM PLX	合成数
		PF2 EEFFRM	电子学公式 14 个
	PF2 MECH 机械工程	PF3 ELEMACT	电磁资料
		PF4 LAPLAC	转换
			机械等公式 12 个
四 PF4 STAT 统计	PF1 分布和概率率	PF1 NORMAL	标准分布
		PF2 T	T 分布
		DF3 CHI	CHI-正方形分布
		PF4 F	F 分布
五 PF5 EDIT 编辑			PF 键表编辑

图 2

硬盘的软维护

中国长城计算机集团公司湖南分公司 马国忻

硬盘在整个计算机硬件结构中是一个至关重要的部件,当计算机其它部分发生故障后,总可将其修复到原来的状态。但硬盘发生故障后,虽也可对其修复,但原有的数据则不一定能保存下来,一次 format 将会使几个月辛勤得出的数据或程序付诸东流。

硬盘的故障有很多种,但相当大一部分是由于关键数据丢失引起的(特别是遭病毒破坏时)。本文不讨论硬件电路及磁介质损伤等物理的修理方法,仅介绍当硬盘中的关键数据(主要指主引导区和 DOS 引导区)被坏后的非物理修复方法,即软件修复。目的是避免对硬盘格式化而保护好硬盘中原有的数据和文件。

常见的硬盘故障有如下几种:

1、硬盘无法自举:主要原因是主引导区或 DOS 引导区或 IBMBIO.COM、IBMDOS.COM、COMMAND.COM 文件被破坏。

2、列硬盘目录时文件名混乱或消失,主要原因是分区表或 BPB 表被破坏。

3、由 A 盘启动系统后不认硬盘,原因同上。

如何在不对硬盘格式化的情况下恢复硬盘的正常使用并保护原有数据呢?下面就几种情况分别讨论之。前提是文件分配表 FAT、根目录登记项 FDT 和数据区 DATA 完好。否则要做相应的工作才能恢复原数据(当文件群号连续,仅 FAT 遭破坏时)或只能恢复部分数据或根本不能恢复(其它情况)。

修复时要做的第一件工作是确定硬盘的头数、柱面数和每道扇数。这可从微机的设置中获得,或从 INT 41H (C 盘)、INT 4EH (D 盘)地址所指的硬盘数据区获得,偏移 0~1 处为柱面数,2 处为头数,0EH 处为每道扇数。

怎样判断被破坏的区域:

三、PC-E500 应用前景

SHARP PC-E500 机将为原 PC-1500 机用户提供一种新型的替代机种, E500 机的 BASIC 命令与原 1500 机命令大同小异,原 1500 机 BASIC 程序很容易改编成 E500 机程序。

E500 将为野外采集数据处理数据与保留数据带来方便,预期对野外使用的仪器仪表智能化创造了一个良好条件。

E500 有一个 AER 模式将为中学教师,大学生,研究生,工程师,以及商人,带来一个大量保存数学公式,保留通讯姓名地址等方便。

E500 预期可与 IBM 微机构成局部网络系统,将有可能进入乡镇企业,中小型企业进行企业管理中应用。

1. A>debug

-11000 201 (在内存 ds: 1000 处装入 DOS 引导区)

若失败,则为 DOS 引导区被破坏或主引导区的分区表被破坏或无硬盘有效标志 55AA,若成功,接做下步:

-d1000 11FF (查看 DOS 引导区内容)

其中除了 BPB 数据段外,其余部分应与 FORMAT.COM

文件中所含的 DOS 引导区数据一样(详见后述)否则为 DOS 引导区损坏或感染了病毒。

2. A>debug

-100a

XXXX: 100 MOV AX, 0201

MOV BX, 1000

MOV CX, 1

MOV DX, 80

INT 13

INT 3

-g=100 (在 ds: 1000 处装入主引导区数据)

-d1000 11FF (查看主引导区数据)

其中除分区表数据外,其余应与 FDISK.COM 文件中所含主引导区数据一样,否则主引导区遭破坏。(详见后述)

一、当 DOS 引导区完好时,修复主引导区步骤。

1、查该分区 BPB 信息, BPB 表位于 DOS 引导区偏移 0BH~21H 处,在下面的调用中位于 100BH~1021H 处。

A>debug

-11000 201 (在 ds: 1000 处装入 DOS 引导区)

若 1 命令失败,用下段程序调出 DOS 引导区

-A100

XXXX: 100 MOV AX, 0201

MOV BX, 1000

MOV CX, 1

MOV DX, 0180

INT 13

INT 3

-g=100

-d100B 1021 (查看 BPB 信息)

BPB 表各项含义如下:

100BH~100CH 每扇字节数

100DH 每群扇数

100EH~100FH 保留扇区 (DOS 引导区占用,为 1)

1010H FAT 个数,为 2

1011H~1012H FDT 项数,大多数为 200H

1013H~1014H 该分区总扇数

1015H 磁盘标记 硬盘为 F8

1016H~1017H 每个 FAT 扇数
 1018H~1019H 每道扇数
 101AH~101BH 磁头数
 101CH~101DH 隐含扇数, 即该分区前扇数
 101EH~1021H 当分区总扇数大于 FFFFH 时, 总扇数填入此处, 且使 1013H~1014H 处为 0. (DOS4.0 以上版本且分区大于 32M 时)

2. 从同 DOS 版的 FDISK.COM 文件中找出主引导区信息.

A>debug fdisk.com

-S100 F000 FA 33 C0 8E (找主引导区头)

-S100 F000 55 AA (找主引导区尾)

从返回的地址处分析这些信息, 找出 FA 与 AA 之间正好是一个扇区 (200H 字节) 且其间含有 'Invalid partition table...' 等信息的一段, 这就是主引导区信息.

3. 计算分区表的各项值. 在主引导区中可含有 4 个分区表, 每个表 10H 字节, 各代表一个分区, 它们位于主引导区的偏移 1BEH~1CDH, 1CEH~1DDH, 1DEH~1EDH, 1EEH~1FDH 处. 任一个分区表中各项含义如下 (以 1EEH~1FDH 为例):

1EEH 80H 表示是活动分区

1EFH 该分区始磁头号

1F0H (高 2 位) +1F1H 该分区始柱面号

1F0H (低 6 位) 该分区始扇号

1F2H 01H 为 12 位 FAT 表, 04H 为 16 位 FAT 表

1F3H 该分区末磁头号

1F4H (高 2 位) +1F5H 该分区末柱面号

1F4H (低 6 位) 该分区末扇号

1F6H~1F9H 该分区前扇数

1FAH~1FDH 该分区总扇数

当每道扇数等于隐含扇 (绝大部分情况) 时, 分区始磁头号、扇号、柱面号分别为 1, 1, 0, 分区末磁头号等于磁头号减 1, 末扇号等于每道扇数, 末柱面号由下式得出:

$\{[(\text{总扇数} + \text{隐含扇数}) + \text{磁头号}] + \text{每道扇数}\} - 1$

分区前扇数取相应 DOS 引导区中的隐含扇数. 总扇数与 BPB 表中一致. 当群数小于 1000H 时, 用 12 位 FAT 表, 否则用 16 位 FAT 表 (DOS3.X 版以上). 群数 = 总扇数 / 每群扇数.

4. 由上求出分区表信息后填入任一分区表位置处.

5. 将内存中含正确分区表信息的主引导区写入硬盘 0 头 0 柱 1 扇处:

例: 长城 386/33C 微机, 104M 硬盘, 8 磁头, 776 个柱面, 每道 33 扇, 含 DOS 分区 (DOS3.3) 和 XENIX 分区. 硬盘不能自举, 查明主引导区被破坏, 采用下述步骤修复.

1. 查 DOS 引导区 BPB 表, 得出隐含扇数为 33 扇, 每群扇数为 8 扇, 总扇数为 6807H.

2. 求 FAT 表位数.

因 $6807H + 8H < 1000H$, 取 12 位 FAT 表.

3. 因每道扇数等于隐含扇数, 取始磁头号、扇号、柱面号分别为 1, 1, 0.

4. 末磁头号为 7, 末扇号为 33, 末柱面号为 $\{[(6807H + 21H) + 8H] + 21H\} - 1 = 64H$.

5. 从 FDISK.COM 中找出主引导区信息.

A>debug fdisk.com

-s100 F000 FA 33 C0 8E

-s100 F000 55 AA

用 d 命令查看由上指出地址中的内容, 查得 B622H~B821H 处为主引导区信息.

-MB622 B821 1000 (将主引导区信息移至 1000~11FF 处)

用 F 命令将计算得出的 DOS 分区表信息写入内存 11BEH~11CDH 处, 内容为: 80 01 01 00 01 07 21 64 21 00 00 00 07 68 00 00.

当要保留 XENIX 分区时, 可由硬盘总扇数, 总柱面以及 DOS 分区已占用数得出, 写入另一分区表中 (例如 11EEH~11FDH 处). 注: 11BEH 和 11EEH 处字节不能同时为 80H.

6. 将主引导区信息写入硬盘, 接上, 不可从 debug 环境下退出.

-A100

XXXX: 100 MOV AX, 0301

MOV BX, 1000

MOV CX, 1

MOV DX, 80

INT 13

INT 13

-g=100

至此, 主引导区修复完毕.

二、当主引导区完好时, 修复 DOS 引导区步骤.

1. 求 BPB 信息, 对于 DOS 3.2 版以上的系统, 可由 DOS 的 440DH 系统功能调用获得.

A>debug

-A100

XXXX: 100 MOV AX, 440D

MOV BX, 3

MOV CX, 860

MOV DX, F000

INT 21

INT 3

-g=100

此时 F007H~F01DH 处即为 BPB 信息.

对于 DOS 3.2 版以下的系统, 因无此功能, 只能采用其它办法.

A>debug

-A100

XXXX: 100 MOV AH, 1C

XXXX: 102 MOV DX, 3

XXXX: 105 INT 21

XXXX: 107 INT 3

XXXX: 108

-g=100 107

此时 CX 中为每扇字节数, AL 中为每群扇数。再用 1 命令、或 PCTOOLS、或 INT 13 来查找 FAT 表所含扇数和 FDT 表所含项数。先寻找 FAT 表头, 它紧接在 DOS 引导区扇后, 头几个字节一般为 F8 FF FF (12 位 FAT 表) 或 F8 FF FF FF (16 位 FAT 表)。再寻找 FDT 表头, 其第一个登记项文件一般为 IBMBIO.COM 或 GWBIO.COM。并找出 FDT 表尾 (其后紧接着 DATA 数据区)。由 FDT 表大小可求出 FDT 登记项数 (每项占 20H 字节)。FAT 表头至 FDT 表头之间含两个 FAT 表, 由此可得出每个 FAT 表扇数。

2. 求主引导区中分区表信息。

A>debug

-A100

XXXX: 100 MOV AX, 0201

MOV BX, 1000

MOV CX, 1

MOV DX, 80

INT 13

INT 3

-g=100

从 11BE~11FD 中找出相应的分区表信息, 由上述步骤可得出 BPB 表中各项的值。

3. 从相同 DOS 版本的 FORMAT.COM 中找出 DOS 引导区信息。

A>debug format.com

-S100 F000 'IBM 3.3'

-S100 F000 55 AA

注意: '3.3' 为 DOS 版本号, 其它版本 DOS 系统应相应修改。若找不到时, 将 'IBM' 换为 'MSDOS' 或 'GW' 则一定能找到, 'IBM' 与 '55 AA' 之间正好含 1FDH 个字节且含有 'Non-system disk or disk error...' 等信息的为 DOS 引导区数据。从 'IBM' 往前移 3 字节即是该区之头 (共长一个扇区)。

4. 在该区距头部起 0BH~21H 处填入 BPB 信息。

5. 将该区内容写入硬盘中 DOS 引导区位置上, 假设该区内容在内存 ds: 296FH~2B6EH 处。

-W296F 201 当失败时用下段程序

-A100

XXXX: 100 MOV AX, 0301

MOV BX, 296F

MOV CX, 1

MOV DX, 0180

INT 13

INT 3

-g=100

DH 中应含该原始磁头号, CH 中应含始柱面号, 应与分区表中的数据一致。

例: 长城 0520 EM 机, 32M 硬盘, 4 磁头, 614 个柱面, 每道 26 扇, 一个 DOS (3.2 版) 分区, 被病毒感染后, 无法由硬盘启动, 查明主引导区完好, DOS 引导区遭破坏。

1. 查分区表信息

A>debug

-A100

XXXX: 100 MOV AX, 0201

MOV BX, 1000

MOV CX, 1

MOV DX, 80

INT 13

INT 3

-g=100

-d11BE 11FF

分区表仅一个内容如下: 80 01 01 00 04 03 9A 65 1A 00 00 00 56 F9 00 00 得始磁头、扇号、柱面号为 1、1、0, FAT 表项为 16 位, 末磁头号、扇号、柱面号为 03、1AH、265H, 该分区前扇数为 1AH, 该分区总扇数为 F956H。

2. 求 BPB 表有关的数据

A>debug

-A100

XXXX: 100 MOV AH, 1C

102 MOV DX, 3

105 INT 21

107 INT 3

108

-g=100 107

得 CX=200H (每扇字节数), AL=4 (每群扇数)。

-10 21 80 ↓ (调出硬盘中 1~80H 扇内容)

-d 0 FFFF

-10 21 80 (调出硬盘中 81H~100H 扇内容)

-d 0 FFFF

由于查得每个 FAT 表含 3FH 扇, FDT 含 200H 个登记项。由上可得出 BPB 表数据为: 00 02 04 01 00 02 00 02 56 F9 F8 3F 00 1A 00 04 00 1A 00 00 00。

3. 从相同 DOS 版本的 FORMAT.COM 文件中找 DOS 引导区信息。

A>debug format.com

-S100 F000 'IBM3.2'

XXXX: 2793

查证 2790H~298FH 为 DOS 引导区信息。

-M2790 298F 1000 (移至 1000H~11FFH 处) 用 F 命令将 BPB 数据写入 100BH~1021H 处, 再将此段引导区写入硬盘中。

-W1000 2 0 1 失败时用下段程序写入

-A100

```
XXXX: 100 MOV AX, 0301
        MOV BX, 1000
        MOV CX, 1
        MOV DX, 0180
        INT 13
        INT 3
```

-g=100

注意: 上述步骤适于 DOS2.1 版以上的系统。对于 DOS3.2 版以上的系统, BPB 表可用 440DH 系统功能调用获得。

三、当主引导区和 DOS 引导区都被破坏时, 修复步骤。

可以分别根据一、二节的方法进行修复。但在 DOS3.2 版以下的某些 DOS 系统中也可按下述方法修复, 但不保证所有的 DOS 系统均正确。

1、先初步恢复主引导区, 用 A 盘启动微机后对硬盘进行 fdisk。

A>fdisk

选 1 产生 DOS 分区, 把整个硬盘都分配给该区, 选 2 激活该分区。这步产生的分区表信息需由下列步骤验证或修改。

2、查找原 DOS 分区的始磁头号、扇号、柱面号和该分区原前扇数。

A>debug

-A100

```
XXXX: 100 MOV AX, 0211
        MOV BX, 1000
        MOV CX, 1
        MOV DX, 0180
        INT 13
        INT 3
```

-g=100

-d1000 1300

查看 1200H 开始处是否为 FAT 表头, 若不是则依次将上段程序中的

```
MOV CX, 1
MOV DX, 0180
```

改为

```
MOV CX, 1
MOV DX, 0280
```

```
...
```

```
MOV CX, 1
```

```
MOV DX, 0n80 (n 为末磁头号)
```

改为

```
MOV CX, 0101
MOV DX, 0080
;
MOV CX, 0101
MOV DX, 0n80
```

并依次运行查找, 直至找出 FAT 表头, 此时的 DH、CH 之值分别为该分区的始磁头号 and 柱面号 (始扇号总是为 1)。该分区前扇数由下式得出: (柱面号 × 磁头号 + 磁头号) × 每道扇数。

3、查找该分区的末磁头号、扇号、柱面号和总扇数。

采用上述方法寻找 FDT 表头、DATA 区头, 并求出 FAT 扇数、FDT 项数、每群扇数、每扇字节数、FAT 位数等。选一个扇数以满足下列两式 (在 (1) 式中不可出现余数):

$$\{[(\text{扇数} + \text{该分区前扇数}) + \text{磁头号}] + \text{每道扇数}\} - 1 \dots (1)$$

$$\{[(1 \text{ 个 FAT 扇数} - 1) \times 200H \times \text{每群扇数} + 2] < \text{扇数} \leq [(1 \text{ 个 FAT 扇数}) \times 200H \times \text{每群扇数} + 2] \dots (2)$$

注: (2) 式适于 16 位 FAT 表, 若是 12 位 FAT 表, 应将式中的 +2 改为 +1.5。

取该扇数为总扇数, 并由此求出末柱面号, 如果计算值与此时分区表的值很接近, 应取此时分区表的值。它们可能会与硬盘中原来的值有一点出入, 但不影响正常使用, 也不影响硬盘中原有的数据 (除非总扇数比原值小很多, 且正好将原来有数据的扇区排除在外, 但这种情况很难发生)。

4、根据需要修改分区表。

5、将修改后的主引导区写入硬盘中。

6、依照前述对 DOS 引导区进行修复。

做了这些工作后, 若还不能启动, 应查看硬盘中的文件, 当列目录正常且调用文件正常时, 可对硬盘进行 SYS。

A>SYSC:

再将 COMMAND.COM 拷入硬盘。这样做需首先保证分区表和 BPB 表与原来的一样 (其中末磁头号、末柱面号和总扇数可除外)。否则将会破坏 FAT、FDT 或 DATA 区, 使其前功尽弃, 切记要慎用 SYS 命令。此外在一、中恢复主引导区时, 也可先对硬盘 fdisk, 再修改分区表, 但不保证所有的 DOS 版本均可行。

若 FAT 表被破坏, 只要群号是连续的, 总是可以恢复原文件 (数据)。建议在第一次对硬盘格式化后, 先将重要的、不改动、不删除的文件拷入硬盘, 再拷入其它文件。并定期查看这些文件, 确保它们的群号是连续的。当重要的文件群号不连续时, 应及时压缩硬盘 (compress), 使群号连续, 以防不测。

关于引导区、分区表、BPB 及群的知识请参阅《DOS 高级技术分析》一书。

带电插拔打印机电缆造成机器故障的修复

河南洛阳矿山机器厂信息中心 陈永青

故障现象：用户在使用 IBM PC/XT 中，常因带电插拔打印机电缆，而线路板未固定紧，造成主机故障死机。

分析与排除：主机一开始就处于死机状态，无报警声，首先从 CPU 8088 着手，看其是否具备正常工作的基本条件，即查看三个信号时钟（CLK）、复位（RESET）、准备好（READY）是否正常。CLK 信号是 8088 的 19 脚，为 4.77MHz 时钟信号，用示波器观察正常；RESET 信号是 8088 的 21 脚，开机时，应有一正跳变，用逻辑笔测得正常；READY 信号是 8088 的 21 脚，正常工作时应为高电平或脉冲，在死机时应为高电平，经测其一直为恒定的低电平，故此信号不正常造成 8088 不工作，主机死机。READY 信号由定时器 8284 的 5 脚供，这一部份的电路如图 1 所示：

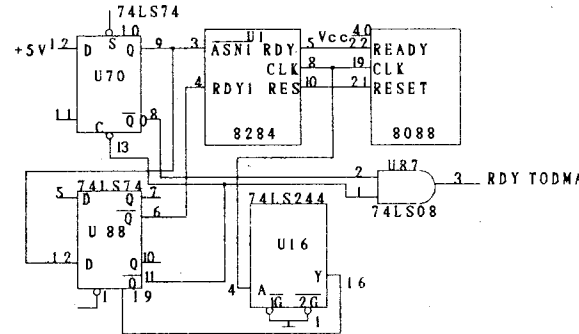


图1

8284 的 3 脚（AEN1）信号和 4 脚（RDY1 信号）决定着 5 脚（RDY 信号）的输出。开机时，RDY1 信号应为高电平，经测正常，AEN1 信号应为低电平，经测恒为高，不正常。AEN1 信号由 U70（74LS74）的第 9 脚输出，即此脚应为低电平，而 U70 工作正常，根据其真值表可知，要使 9 脚 Q 端输出为低电平，必须是其 10 脚 S 端为高电平（经测正常），13 脚 C 端为低电平，而此时测得 13 脚为高电平，不正常。U70 的 13 脚又由 U88（74LS175）的 11 脚提供，11 脚的输入端 12 脚为高电平，正常时 11 脚应输出低电平，经检查 U88 的时钟端 9 脚无时钟信号，造成触发器不能翻转，U88 的时钟信号为 8284 的 CLK 信号经 U16（74LS244）由 16 脚输出，测 U16 损坏（输入端为脉冲或高低电平，输出端为浮空状态），用一新的 74LS244 更换后主机启动正常。

主机正常后，显示器仍无显示，用交换法确认显示卡仍有故障，且无一长二短报警声。此显示卡采用 6845 彩卡控制器（图 2），首先检查控制器是否具备正常工作条件。

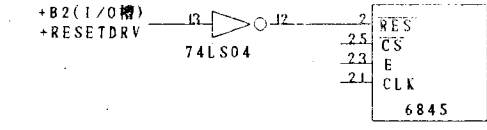


图2

用示波器观察时钟信号 CLK（21 脚）正常，但初始复位



附表给出 FDISK.COM 和 FORMAT.COM 文件中各引导区的位移值。

附表

DOS 版本	主引导区位移 FDISK.COM 中	头几个字节内容	DOS 引导区位移 FORMAT.COM 中	头几个字节内容
2.1	15E1	FA 33 C0 8E	19F8	EB 2C 90'GW 2.1'
3.0	1C4C	FA 33 C0 8E	21B8	EB 2A 90'IBM 3.0'
3.2	1CAD	FA 33 C0 8E	2790	EB 34 90'IBM 3.2'
3.3	B622	FA 33 C0 8E	296F	EB 34 90'IBM 3.3'
3.30A	B878	FA 33 C0 8E	29EA	EB 34 90'MSDOS 3.3'
3.31	CE02	FA 33 C0 8E	3121	EB 28 90'IBM 3.3'
4.0			151E	EB 3C 90'MSDOS 4.0'
5.0			163B	EB 3C 90'MSDOS 5.0'

CITIZEN CKP-5240 打印机维修一例

广州 1180 信箱 陈祥伍

故障现象:

脱机打印正常,联机打印乱码,且联不同主机故障相同。如在 DOS 提示符下转向列目录,

A: \>dir>prn

Vonwog in fivig A iw WWPGV / GGFOW

正确的应是: Volume in drive A is SUPER-CCDOS

故障分析:

乱码故障一般出在接口的选通 (strobe)、数据 (db0 ~ db7) 信号线上, 通常是某位固定为 1 或 0, 从而使从主机传来的 ASCII 码都加或减某一值 (2 的幂) 只要找出此值, 就可确定是哪一个数据位。但是此例比较特别, 因为仅从上述现象还很难确定哪一位异常。如上列行中字符 S 变成 W (对应的 ASCII 码 01010011 变成 01010111), 说明 bit2 恒为 1, 而字符 U 变成 W (对应的 ASCII 码 01010101 变成 01010111), 则说明 bit1 恒为 1, bit2 不恒为 1, 再有, 字符 p 不变 (ASCII 码 01110000), 又说明 bit1、bit2 都不恒为 1。为了进一步摸清规律, 笔者编了一段汇编程序 (附后), 运行该程序可打印出如下所示的数字字母字符, 它们的 ASCII 码为 30 ~ 7ah。

0123456789; < = > ? @ ABCDEFGHIJKLMNOPQRSTU
VWXYZ[\] ^ _ ' abcdefghijklmn opqrstuvwxy z

0167676789>?>?>?@_AFGFGFGHINONONOPQVWV
WVWXY^ ^^ ^ _afgfgfghinononopqvwvwvwxv~

004422000044220000442200004422000044220000442200004
422000044220000442200004

上列第一行是期望的输出, 第二行是实际打印的结果, 第三行是对应列字符 ASCII 码的增值。输出字符的 ASCII 码有规律地加 4 或 2, 可见故障是由 bit1 和 bit2 共同引起的。进一步分析发现: 当 bit1、bit2 都为 1 或 0 时, 输出字符不变, 当 bit1、bit2 任一位为 1 时, 则另一

信号 RES (2 脚) 为浮空状态 (正常时, 开机应有一跳变, 低有效), 此信号由主机板 I/O 槽 +B2 槽 (+RESTDRV 信号) 经 74LS04 非门反相得到, 测 +RESTDRV 恒为低, 无跳变, 怀疑是否主机板 +RESETDRV 输出信号有错, 当拔下显示卡再测主机板 +RESTDRV 信号时, 开机有正常跳变, 故问题出在显示卡上。经测非门 74LS04 输入脚 (13 脚) 被击穿, 对地电阻仅为 90 欧姆 (正常应为几百千欧以上), 造成 13 脚始终为低电平, 74LS04 的其它门均完好, 仍可使用, 用刀割断 +RESTDRV 信号与 74LS04 的 13 脚间印刷线路, 用外接飞线连至显示卡上另一未用的 74LS04 空门的输入脚上, 把输出脚与 6845 的 2 脚连通, 显示卡即恢复正常。

位也为 1, 输出字符加 4 (bit1=1) 或加 2 (bit2=1)。由此可断定打印机内接口线 db1 与 db2 粘结。

故障排除:

拆开打印机,发现 db0-db7 各自经过提升、限流电阻连到 PLCC 封装表面安装的 IC7 (CITIZEN p2503-300)。用酒精清洁 db1、db2 的通路,着重刮 IC7 的引脚 pin29、pin30 (与 db1、db2 相连),再连机打印,故障消失。

讨 论:

1.SMT (表面安装技术) 的应用, 使电路板更精密, 制造成本更低, 但也带来了一些弊病。特别是肮脏的环境会造成其细密的引脚粘结或短路, 这是使用和维修中应特别注意的。

2.硬件维修离不开软件诊断,下面的程序可向 LP0 输出 30H~7AH 之间的 ASCII 字符,仅供参考。

```

codeseg segment para 'code'
test     proc far
          assume cs:codeseg
          push ds
          xor ax,ax
          push ax
          mov dx,0000h
          mov ah,01h
          int 17h
          xor bx,bx
cont:     mov al,30h
loop:     mov ah,00h
          int 17h
          inc al
          cmp al,07bh
          jnz loop
          inc bx
          cmp bx,10h
          jnz cont
          ret
test     endp
codeseg ends
          end test

```

启 示

为了方便读者调试程序,也为了节省篇幅,较长的程序将不刊出。每期杂志设有一编号为 Pxy-n (其中 xy 为年号, n 为期别) 的软盘,将该期所有程序录入,程序文件名在文末标注。每片软盘邮购需 15 元 (含邮费)。

以新观点来改进高校计算机教育

河北大学电子与信息工程系 杨 成

培养使用计算机的能力不能单靠一门语言课

笔者对已经走上工作岗位的非计算机专业的毕业生,小范围地作了一个调查,发现多数毕业生几乎不会使用计算机,但是他们还是说自己已经在大学学过计算机—BASIC 语言。

外文,数学及政治类课早就是理工科的公共课。近十年来,由于计算机逐步应用到了国民经济的各个部门,所以计算机课也成为了公共课。然而目前多数学校所开设的计算机课,也仅是一门“BASIC 语言程序设计”课,再加上“机时”极有限的上机实践课。

对于一个理工科毕业生,使用计算机的能力,主要是指,对本专业应用软件的使用,开发和维护的能力。而对仅学过一门程序设计语言的学生,要读懂本专业的售品应用软件已是十分困难的,甚至是不可能的,更谈不上对本专业应用软件的二次开发和维护。道理很简单,用高级语言编写程序所需要的工作量,只占软件开发全部工作量的百分之十到百分之二十。

所以毕业生在校一门语言课的学习中,顶多只学会了开发软件所必需的知识,技能中的很少一部分。

因此建议,对非计算机专业的理工科学生所开设的计算机课程,还必须增加《结构化程序设计原理》,《数据结构》,《软件工程》等课程。

《结构化程序设计原理》也就是《算法设计》,它是独立于任何计算机语言而专门研究“解题方法”的软件基础学科。

“数据”是软件的基本元素。在程序设计中,采用不同的数据结构来组织数据时其程序运行效率(主要指时空指标)相差极大。故开设《数据结构》课程十分必要。瑞士学者 WIRTH 指出“算法+数据结构=程序”。

另外,学生还必须掌握软件开发与维护的普遍原理和技能,故开设《软件工程》课程也是完全必要的。

这三门课对一般专业,各开 30 学时,对计算机使用能力要求较高的专业(如管理信息专业)上述每门课至少开 40 学时以上。

非计算机专业的计算机语言课程应开设 PASCAL 语言

也许是由于“先入为主”等历史原因,多数院校的非计算机专业几乎清一色地开设《BASIC 语言程序设计》课,而使用的教材不少院校仍然是老式的标准解释 BASIC。

众所周知,BASIC 虽然有简单,易学,应用较广的优点,但老式 BASIC,它是一种非结构化程序设计语言。这对于树立学生“软件工程”的概念,培养学生严谨的编程作风以及让学生掌握“自顶向下逐步求精”的结构化程序设计

方法都是极为不利的。事实表明,不用软件工程概念和结构化程序设计方法开发出来的软件,其使用,扩充和维护都十分困难。

目前能够较好地克服上述弊端的语言首推 PASCAL 语言。PASCAL 语言方不仅是一种结构程序设计语言,而且有比其他任何语言都丰富的数据结构类型,这给各种应用带来极大的方便。尤其是它可以用简炼的动态数据结构,直接实现线性表,二叉树等复杂的数据结构,这样就使得在处理复杂的数据结构时,程序清晰,精练。

目前在国内微机上流行的 TURBO PASCAL V5.0,它在原标准 PASCAL 基础上又做了很大发展,又具有许多新特点,如编译代码优化,使得它成为编译最快,生成目标代码最短,执行速度最快的系统之一等。

建议,非计算机专业的语言课程开设《PASCAL 语言程序设计》课,讲授标准 PASCAL,上机则用 TURBO PASCAL。若尚未有条件立即从 BASIC 改为 PASCAL 的单位,可以采用新式的,符合结构化理论的 TRUE BASIC 来作为过渡。

应该向学生灌输“比较软件学”的观点

当今计算机语言数以千计,还在不断推陈出新,应用软件也层出不穷,各有千秋,新观念,新方法应接不暇。使学计算机,用计算机的人们感到,要跟上计算机飞速发展的步伐,真是谈何容易。难怪计算机专业的学生和教师很少有会跳舞的。

再看一下,有些院校计算机专业的课程设置中,仅计算机语言就有:一年级开 BASIC 和 FORTRAN,二年级开 PASCAL 和汇编语言,四年级开 PROLOG 或 COBOL。于是有人将计算机专业讥为“语言专业”。

所谓比较软件学,其基本思想就是研究程序语言之间,软件之间的相互联系和影响,使用的异同,表达方式的亲疏,从而为在已有的语言,软件的经验基础上,尽快掌握另一种语言,软件。

本来,PASCAL, FORTRAN, DBASE 等都有相象的程序控制结构和程序思维方式。LOGO 和 FORTH 则都是以扩充命令为中心的另一种思维方式。还有微机上三种典型的操作系统:CP/M, MS-DOS, XENIX 它们之间有着很深的内在联系和继承性。更不用说象 DBASE 的人事管理,工资管理,仓库管理等应用系统之间的共性了。

建议,计算机应用专业只开《PASCAL 语言程序设计》和《汇编语言程序设计》两门语言课。至于其它诸如 BASIC, FORTRAN 等可让学生以比较软件学的观点去看。同样,《操作系统原理》课程的实例讲述,只讲 UNIX 而令学生自学 MS-DOS 或 XENIX。逐步培养学生以

CEC-I 中文状态下 CATALOG 的改进

福建南平师范 李 锋

CEC-I 在汉字状态下显示磁盘目录时很不方便。所显示的磁盘目录间距大，上卷速度又很快，以至于不少目录尚未看清，就已卷过去，每次所显示的 23 条目录，留在屏幕的仅有最后四条，给我们的磁盘操作带来了困难。

原来，CEC-I 在中文状态下的磁盘操作系统仍是西文 DOS3.3。在西文状态下，文本的屏幕为 24 行 40 列，这样，DOS3.3 系统中规定在列目录时，显示的行数被置为 23 行，且规定，最长的文件名长度为 30 个，不足 30 个的系统直接以空格相补。这样当执行 CATALOG 命令时，每一行由文件加锁状态、文件类型、文件占有的扇区数及文件名这四部分组成，这四部分所占位置加上各部分间的空格，恰好凑足 40 个字符。显然 DOS3.3 对西文状态下的列目录安排是合理的。然而在 CEC-I 中文状态下，用户可使用的屏幕仅为 10 行 34 列，这样原来的西文状态下恰好凑成一行的 40 个字符就变成需占用二行，而且在实际应用中，文件名的长度一般不超过 25 个字符（其余机器自动以空格填满），这样就造成了在实际显示中第二行往往是空行的感觉，使得一屏最多只能显示 5 条目录。而原设计的一次显示二十多条目录，自然就造成了目录信息不断上卷，到最后只显示最后四条的情况了。

通过对 DOS 的剖析，我发现只要将 \$AE17，\$ADA4，\$AE30 这三个单元的值 \$10，\$16，\$15 改为 \$19，\$09，\$09 即可。

* 300LLL

```
0300-A9 4C      LDA # $4C
0302-8D F5 03    STA $03F5
```

```
0305-A9 10      LDA # $10
0307-8D F6 03    STA $03F6
030A-A9 03      LDA # $03
030C-8D F7 03    STA $03F7
030F-60          RTS
0310-20 7B DD    JSR $DD7B
0313-20 0C E1    JSR $E10C
0316-A5 A1      LDA $A1
0318-C9 02      CMP # $02
031A-D0 10      BNE $032C
031C-A9 1D      LDA # $1D
031E-8D 17 AE    STA $AE17
0321-A9 16      LDA # $16
0323-8D A4 AD    STA $ADA4
0326-A9 15      LDA # $15
0328-8D 3D AE    STA $AE3D
032B-60          RTS
032C-A9 19      LDA # $19
032E-8D 17 AE    STA $AE17
0331-A9 09      LDA # $09
0333-8D A4 AD    STA $ADA4
0336-A9 09      LDA # $09
0338-8D 3D AE    STA $AE3D
033B-60          RTS
033C-00          BRK
```

比较软件学为手段，触类旁通，举一反三地学习，创新，以适应日新月异的计算机时代。

模糊数学势必成为新一代计算机教育的基础核心课。人工智能 (AI) 的迅猛发展导致新一代计算机即将产生，作为 AI 的基础之一的模糊数学 (FM) 越发显示了它的重要性。

经典数学（即经典集合论）中的一个对象“非此即彼”，过去的计算机就是建立在这样的两值逻辑基础上的。自然和社会现象中，差异往往呈中介过渡，自然语的特点是模糊性。FM 正反映人脑及自然界这种“亦此亦彼”的差异过渡，把经典数学从两值逻辑基础扩展到连续值上来。

模糊数学与新一代计算机息息相关。新一代计算机对模糊数学的教学提出了紧迫要求。离散数学是计算机专业的必修基础课，不久，模糊数学及其应用范畴将会超出目前离散数学。所以模糊数学将成为重要的计算机科学的基礎课。

当前国内院校的计算机专业有的已经开设了模糊数学

类课程。建议还没有开设该课程的单位应创造条件积极筹备开设。

要让学生掌握计算机最重要的途径就是让其经常使用计算机。

计算机的课程，特别是语言课是实践性很强的学科。要收到好的教学效果，除教师课堂讲授外，最重要的环节就是靠学生上机实践来加深认识和理解，并逐步培养起学生自己动手的能力。尤其是学生软件调试能力，只能通过上机实习来培养，用其他任何形式都无法代替。学生上机时数至少应和讲授时数成 1:1 的关系。

然而，多数院校由于资金及人为的原因，学生上机时数严重不足，这也是造成学生使用计算机的能力差的主要原因之一。

目前 IBM-PC 及其兼容机的价格仅为 2000~3000 元/台（不包括硬盘和打印机）。

严重影响学生上机时数的机器台数奇少的局面，该是彻底改变的时候了。

保持在中文状态下操作

沈阳市苏家屯铁路中学 宿苏英 陈志岭

CEC-I 中华学习机有两种显示形式(中文状态和西文状态)可供直接转换选择。中文状态字符大、有汉字。分辨率也高。适于保护操作者的视力,符合教学卫生。因此对学校教学来讲,更应选用中文状态。但在上机操作中,往往由于不小心,错误地压下了西文转换键,使当前的中文画面遭到了不应有的破坏。为此有必要用“软”办法来保护中文状态,对西文转换键进行软禁。

键入下列机器码程序,检查无误后,运行 340G 就能进入中文状态,并且保护在中文状态下操作了。如果这时按下西文转换键,机器会发出“笛拉”声音,提示误按了西文键。从而达到了保护中文状态的目的。本功能也可用 BASIC 语言程序 2 来运行。作用与程序 1 相同。

在计算机做其它工作中,使用 PR#3 或使用 Ctrl-RESET 时,会使本功能失效。只要键入下列命令: CALL832↓就能恢复本功能。

程序 1

* 0300.038F

```
0300-20 AB C3 20 A3 F7 C9 91 F0 03 4C B9 C3 20 B9 C3
0310-A9 87 4C F0 FD EA EA EAEA EAEA EA EA EAEA EA
0320-29 7F C9 11 F0 09 20 ABC3 20 CE EF 4C B9 C3 A9
0330-10 8D B0 C0 20 B9 C3 A9 87 4C F0 FDEA EAEA EA
0340-A9 00 8D 7B 06 A9 03 AA4D 7B 04 EA EA 8A 8D 7B
0350-04 20 AB C3 20 54 EC AD50 C0 AD52 C0 AD55 C0
0360-AD57 C0 AD10 C0 A9 20 8D 53 AA A9 03 8D 54 AA
0370-8D 56 AAA9 00 8D 55 AA85 73 A9 92 85 74 EA EA
0380-EA 20 94 EC20 16 EC 20 D1 9E 4C B9 C3 00 00 60
```

程序 2

```
10 FOR I=1 TO 144:READ A
20 POKE 767+I,A:NEXT I:CALL 832
30 DATA 32,171,195,32,163,247,201,145,240,3,76,185,195,32,185,195
31 DATA 169,135,76,240,253,234,234,234,234,234,234,234,234,
234,234
33 DATA 41,127,201,17,240,9,32,171,195,32,206,239,76,185,195,169
34 DATA 16,141,176,192,32,185,195,169,135,76,240,253,234,234,
234,234
35 DATA 169,0,141,123,6,169,3,170,77,123,4,234,234,138,141,123
36 DATA 4,32,171,195,32,84,236,173,80,192,173,82,192,173,85,192
37 DATA 173,87,192,173,16,192,169,32,141,83,170,169,3,141,84,170
38 DATA 141,86,170,169,0,141,85,170,133,115,169,146,133,116,234,234
39 DATA 234,32,148,236,32,16,236,32,209,158,76,185,195,0,0,96
```

APPLE 图形压缩系统

广州华南师大计算机 90 级 梁宇翀

APPLE 机上,将一页高分辨率图形存盘会占用 34 个扇区,非常浪费磁盘空间。但如果用本图形压缩系统,会节省大量的存贮空间。例如,用本系统将下图进行压缩后存盘,仅占用 8 个扇区。

本系统使用方法如下:

\$E6 单元存放图形的页首地址高字节。

\$FA 和 \$FB 单元存放图形压缩资料的起始。

\$2FF 是压缩工作入口。

\$346 是还原工作入口。

还原或压缩工作完成后,计算机还会将图形压缩资料的尾地址放于 \$FC 和 \$FD 单元。

02FF-20

```
0300-7D 03 48 20 8A 03 85 08
0308-A2 01 68 20 B6 03 F0 0C
0310-48 20 8A 03 C5 08 D0 04
0318-E8 D0 EF CA 84 47 A0 00
0320-E0 04 B0 0A A5 08 20 C7
0328-03 CA D0 F8 F0 0D 98 20
0330-C7 03 8A 20 C7 03 A5 08
0338-20 C7 03 A4 47 A5 27 F0
0340-52 20 8D 03 D0 C0 20 7D
0348-03 AA 48 20 94 03 A1 FC
0350-D0 1E 20 C9 03 A1 FC 85
0358-08 20 C9 03 A1 FC 91 26
0360-68 20 B6 03 F0 69 C6 08
0368-F0 0E 48 20 94 03 D0 EC
0370-91 26 68 20 B6 03 F0 57
0378-20 C9 03 D0 CD A0 02 B9
0380-F9 00 99 FB 00 88 D0 F7
0388-98 60 20 94 03 B1 26 D0
0390-02 09 80 60 48 29 C0 85
0398-26 4A 4A 05 26 85 26 68
03A0-85 27 0A 0A 0A 26 27 0A
03A8-26 27 0A 66 26 A5 27 29
03B0-1F 05 E6 85 27 60 18 69
03B8-01 C9 C0 D0 09 A9 00 C8
03C0-C0 28 D0 02 85 27 60 91
03C8-FC E6 FC D0 02 E6 FD 60
```



ACE 语言(课件设计环境)



《编辑盘》

谜底: 1、溢出 2、中断 3、匹配 4、通讯(汛)

也谈快速随机读取 READ 语句

辽宁省朝阳市高级中学 蔡安军

《电脑》1990年第6期《快速随机读取 READ 语句》一文中介绍了快速随机不重复显示英文单词的方法。基本上做到了计算机出题时,使用者不需要时间进行等候。这种方法的实现分三步:第一步,按照单词存放顺序,把单词依次读入到变量 A\$(I), B\$(I) 中(见程序1 60~80 语句)。第二步,在同一变量 A\$(X) 和 A\$(Y)、B\$(X) 和 B\$(Y) 之间进行单词交换,从而形成随机不重复单词(见程序1 130~180 语句)。第三步,建立循环,把交换后的单词逐个显示出来。

这种方法虽然第三步显示单词时不需等待,但第二步进行单词交换时,如果单词总数很多,仍需较长时间进行等候,这里介绍一种更好的方法,可以真正做到快速随机不重复。这种方法的设计思路是:第一步与前面的方法一样(见程序2 60~80 语句)。第二步,在没被出过的单词中随机选一个显示出来(见程序2 150、160 语句)。第三步把这个单词放在剩下的没有出过的单词的最前面(见程序2 165 语句),然后重复第二、三步,直至出完全部单词。这种方法,出题与单词交换在同一个循环内完成,出一个单词,交换一个单词,因此完全不用时间进行等候,而且显示的单词随机、不重复。

现用这种方法改写了《快速随机读取 READ 语句》一文中的程序,功能与原来完全相同,但速度更快(程序2)。

程序 1

```
60 FOR I=1 TO Q1
70 READ A$(I),B$(I)
80 NEXT I
...
130 FOR I=1 TO Q1
140 X=INT(RND(I)*Q1)+1
150 Y=INT(RND(I)*Q1)+1
160 N$=A$(X):A$(X)=A$(Y):A$(Y)=N$
170 N$=B$(X):B$(X)=B$(Y):B$(Y)=N$
180 NEXT I
```

程序 2

```
10 PRINT CHR$(4):"PR#3":PRINT:HGR2
20 X=FRE(0)
30 Q1=18
40 HOME:VTAB 4:HTAB 8:PRINT "请稍等,正在读单词!"
50 DIM A$(Q1),B$(Q1)
60 FOR I=1 TO Q1
70 READ A$(I),B$(I)
80 NEXT I
90 HOME:VTAB 3:HTAB 8:PRINT "共有,Q1:个单词"
100 VTAB 5:HTAB 8:INPUT "练习多少个单词?":Q
110 IF Q<1 OR Q>Q1 THEN 100
```

```
120 W1=(100/Q)/3:W1=INT(W1*1000+0.5)/1000
130 HOME:Q2=Q:N=0:W=0
140 FOR I=1 TO Q
150 J=INT(RND(I)*Q2)+1
160 GOSUB 230
165 H$=A$(I):A$(I)=A$(J):H$=H$:H$=B$(I):B$(I)=
    B$(J):B$(J)=H$
170 NEXT I
180 VTAB 9:HTAB 5:INPUT "1.重复 2.重选 3.结束":K$
190 IF K$<"1" OR K$>"3" THEN 370
200 IF K$="1" THEN 370
210 IF K$="2" THEN 90
220 HOME:VTAB 4:HTAB 12:PRINT "下次见!":END
230 K=0
240 VTAB 1:PRINT Q2:"个没做":TAB(19):"共有":Q:"个单词"
250 VTAB 3:PRINT A$(J)
260 VTAB 5:INPUT "":K$
270 IF K$=B$(J) THEN VTAB 7:HTAB 24:PRINT "(正确)":
    GOTO 410
280 K=K+1:N=N+1
290 IF K<3 THEN VTAB 7:HTAB 24:PRINT "错误!":GOTO 420
300 VTAB 6:PRINT "答案是:"B$(J)
310 W=100-W1*N:W=INT(W*100+0.5)/100
320 VTAB 8:PRINT SPC(40)
330 VTAB 8:PRINT "拼错一次,扣,W1:分":TAB(18):"现总分为,W:
    分"
340 VTAB 9:HTAB 10:PRINT "练习下一个单词!":GET K$
350 VTAB 3:PRINT SPC(200)
360 VTAB 9:PRINT SPC(40):Q2=Q2-I:RETURN
370 HOME:Q2=Q:N=0:W=0
380 FOR J=1 TO Q
390 GOSUB 230
400 NEXT J:GOTO 180
410 FOR H=200 TO 0 STEP -10:MUSIC H,10:NEXT H:GOTO 310
420 VTAB 5:PRINT SPC(160):GOTO 260
430 DATA 碗,bowl,盘子,plate,鱼,fish,面包,bread,另一个人,other,肉
    ,meat
440 DATA 一,one,米饭,rice,水,water,他们,them,传递,pass,一玻璃杯,
    a glass of
450 DATA 给你,here you are,茶,tea,给,give,他,him,一(茶)杯,a cup of,奶
    ,milk
```

一个小发现—6502 的奇怪转移指令

朱 希

大家知道，在 6502 汇编语言中，PHA 是将 A 的内容压入堆栈的操作，其机器码是 48。可是，你尝试过这样来使用它吗？在中华机中进入监控后，键入：

* 300: A9 FC 48 A9 57 48 60

以 \$ 300 为首址反汇编，程序为：

300L

300-A9FC LDA# \$ FC

302-48 PHA

303-A957 LDA# \$ 57

305-48 PHA

306-60 RTS

.....

这看不出什么特别之处，可是运行它（300G），却实现了清屏功能，同 FC58G 的效果一样。入栈指令成了转移指令。

同样，若将 \$ 301 的内容换成 FE，将 \$ 304 的内容换成 7F，运行它时程序将转移到 \$ FE80 入口处执行，将文字显示方式设定为反相方式。若将 \$ 304 内容再换成 83，运行它时程序将转移到 \$ FE84 入口处执行，将文字显示方式设定为正常方式。若将 \$ 301 内容换成 FF，\$ 304 内容换成 39，运行它时程序将转移到 \$ FF3A 入口执行，扬声器响一声“啵”。

以上例子表明，一个 6502 汇编语言程序中，若有两次入栈指令 PHA 而无出栈指令 PLA，程序将转移到一个新的地址中执行，这个地址的高八位是第一次入栈指令中 A 的内容，低八位是第二次入栈指令中 A 的内容加一。

利用这种奇特的转移指令，可以使我们编制的软件有某种保密作用。事实上，台湾产的 MPF-II 小教授微机，其监控程序中就使用了这种方法。我在分析 MPF-II 的监控程序时，发现“R”命令进入 \$ FF0C 执行，而“W”命令也进入 \$ FF0C 执行。要知道，R 命令是读磁带，而 W 命令是写磁带，怎么能用同一段程序处理？让我们先看看“R”程序段：

@FF0AL

FF0A-A203 LDX#3

FF0C-BDE5EE LDA \$ EEE5, X; 实际 DE -> A

FF0F-48 PHA

FF10-CA DEX

FF11-BDE5EE LDA \$ EEE5, X; 实际 AD -> A

FF14-48 PHA; 此时程序转去 \$ EDAA 入口

执行“R”。

“W”程序段如下：

广州市天河思索电脑公司简介

广州市天河思索电脑公司是一间推广高新技术产品、提供优质服务的全新型的全民所有制企业。

公司主要从事计算机工程、系统应用软件、自动控制、CAD、网络工程、信息管理以及现代办公自动化管理等多方面的研究开发和推广应用工作，并逐步从商业性销售转变为技术性销售。公司在市内用于开发研究和经营的场地达一千平方米。公司正将业务拓展到北京、江苏、四川、山西等地。

“思索电脑”组建了一支训练有素、精明强干、具有全新意识和开拓进取精神的技术队伍。公司的职员全是受过高等教育和专业技术培训的多方面人才，其中具有高级技术职称的工程技术人员占总人数的 25%。此外，公司还同十多家高等院校、研究所建立了长期的全作关系，并聘请了数十名各方面的专家，组成了一个智囊团，为公司建立了坚强的后盾。

公司本着对客户负责，对社会负责的精神，坚持以一流的质量、一流的服务为广大客户提供应用软件开发的技术支持；提供保证系统软件的支持；并成为第一家提供设备三年免费保修服务以及报修 24 小时内作出妥善处理的新型电脑公司。

广东是改革开放的前沿，经济要快速发展必须有现代化的信息管理，计算机是必不可少的得力助手。为加快计算机这一高新技术在广东经济建设中的应用推广，实现现代化的科学管理，以推动国内计算机的发展，公司将努力加强企业内部管理，降低成本，实行薄利多销，降低计算机设备的价格，集中高科技人员开发系统及应用软件，把高新技术成果迅速转化推广应用；举办高新产品研讨推广会，举办计算机学习班，尽快提高广大客户的计算机应用水平，并开发研制一套全新的中文操作系统软件献给广大用户，免去西文软件汉化的繁琐工作。

公司全体员工上下一致，团结一心，以求实、进取、探索、创新为办事方针，以自身的经济活动为国家服务。参与改革，开拓市场，迎接高新技术革命的浪潮。

中国惠普授权代理 HP 产品有：

PC 全系列主机、绘图仪系列激光打印机系列、喷墨打印机系列、扫描仪系列及各种电子医疗仪器

地址：广州天河五山路 106 号 邮编：510630

电话：5516745 5516746 FAX: 5516770

FEDA-A201 LDX#1

FE0C-D02E BNE \$ FF0C; 转去 \$ FF0C 处理，处理后转移到 \$ EC9A 执行。

EEE6-EC 99 ED AD

小小一段程序，能使许多不知它功能的人感到莫名其妙，不知道程序发生了转移。 ■

炸弹与迷宫

黄东斌

炸弹与迷宫是1991年广东省青少年计算机程序设计竞赛的复赛题。(详见本刊1991年第4期)该题的背景对各位考生都很熟悉,但其难度比普通迷宫困难得多,主要在于算法效率上。若采用穷举方法,则其耗费时间将随迷宫规模及炸弹数成指数增长,对大的参数,需几小时完成。本文介绍一种快速算法,在286机上对多个测试数据,算法都在100ms左右完成。

迷宫中的方格分实点与空点(意义请看试题)。定义点V颜色数是始点到达V最小要使用的炸弹数。颜色数相同的实点组成的链,称为实点链。

本算法的基本思想是找出颜色数等于1的实点链,接着对此链中每一点找出连通块。那么很明显,与连通块相邻的未访问实点的颜色数等于2。这样,找出颜色数等于2的实点链。接着找颜色数等于3的实点链……直到终点出现在实点链或连通块中,即被访问为止。

在解题程序中迷宫中每一点都是一个PZ结构,分量visist为访问标志,quantity是实点与空点标志,father用来存贮路径,这是本算法的一个技巧,它避免了重新找路径的时间耗费,deblock为该点的颜色数,block-link用于连接实点链。

函数next-block入口参数是一条颜色数等于k的实点链,功能是找到并返回颜色数等于(k+1)的实点链。

函数find-block入口参数是颜色数等于k的点V,功能是访问所有与V相连通的点,并返回与此连通块相邻的实点链,其颜色数等于(k+1)。

另外,init-map与out-map过程分别处理迷宫的初始化及输出结果迷宫。

然而,算法中的初始化方法是有趣的,读者不妨深入下面给出本题解源程序主要部分(BOMB.PRT)。如读者感兴趣,可参考本期磁盘内的BOMB.C, BOMB.EXE文件。

程序清单如下:

Puzzle Bomb

```
#define TRUE 1
#define FALSE 0
#define BLOCK 1
#define EMT 0
#define NMAX 20
#define QUIT 0X1B
#define CHG ''
#define UP 0X48
#define DOWN 0X50
#define RIGHT 0X4D
#define LEFT 0X4B
```

```
#define C BLOCK 'B'
#define C EMT ''
#define ANTI BLOCK 'b'
#define ANTI EMT ''
```

```
#define A(i,j) a[i][j].quantity = BLOCK;
```

```
typedef struct { int x,y; } POINT;
typedef struct { int visist,quantity,father,deblock;
                POINT block link; } PZ;
PZ a[NMAX][NMAX];
int n,bomb;
```

```
int next_block(POINT *,POINT *);
void in(POINT);
void init_map(void);
void out_map(void);
```

```
main( )
```

```
{
    int flag;
    POINT head,rear,fp={0,0};
    init_map();
    a[0][0].visist = TRUE; a[0][0].deblock = (a[0][0].
    quantity == BLOCK);
    a[0][0].block link.x = a[0][0].block link.y = -1;
    a[0][0].father = 3;
    flag = 0; head.x = head.y = 0; rear = head;
    while (!flag)
        flag = next_block(head,rear);
    out_map();
}

/* ----- */
int next_block(POINT * phead, POINT * prear)
{
    POINT back head,back rear,t head,t rear;
    int flag = 0;
    int find_block(POINT,POINT *,POINT *);
    back head.x = back head.y = -1;
    back rear = back head;
    while (phead->x >= 0 !flag)
    {
        t head.x = t head.y = -1; t rear = t head;
        flag = find_block(*phead,t head,t rear);
        if (t head.x >= 0)
            if (back head.x >= 0)
            {
                a[back rear.x][back rear.y].block link = t head
                back rear = t rear;
            }
        else
        {
            back head = t head;
            back rear = t rear;
        }
        *phead = a[phead->x][phead->y].block link;
    }
}
```

```

    }
    * phead = back head; * prear = back rear;
    return(flag);
}

int find_block(POINT fp, POINT * phead, POINT * prear)
{
    int i = 0, flag = 0;
    POINT p;
    POINT f(POINT, int);
    if (fp.x == n-1 & fp.y == n-1) return(1);
    while (++i <= 4 !flag)
    {
        p = f(fp, i);
        if (p.x >= 0 & p.x < n & p.y >= 0 & p.y < n
            & a[p.x][p.y].visist == FALSE)
        {
            a[p.x][p.y].visist = TRUE;
            a[p.x][p.y].father = 5-i;
            a[p.x][p.y].deblock = a[fp.x][fp.y].deblock
                + (a[p.x][p.y].quanlity == BLOCK);
            if (a[p.x][p.y].quanlity == BLOCK)
            {
                a[p.x][p.y].block link.x = a[p.x][p.y].
                    block link.y = -1;
                if (phead->x >= 0)
                {
                    a[prear->x][prear->y].block link = p;
                    * prear = p;
                }
            }
            else
                * phead = * prear = p;
        }
        else
            flag = find_block(p, phead, prear);
    }
    return(flag);
}

POINT f(POINT src, int direction)
{
    POINT dst;
    switch (direction)
    {
        case 1: dst.x = src.x; dst.y = src.y+1; break;
        case 2: dst.x = src.x+1; dst.y = src.y; break;
        case 3: dst.x = src.x-1; dst.y = src.y; break;
        case 4: dst.x = src.x; dst.y = src.y-1; break;
    }
    return(dst);
}

```

牌洗几次最好?

重庆大学计算机系 符云清

一副扑克牌洗几次才能洗匀? 这个问题曾难倒不少数学家, 因为 52 张牌有 52! 种排列, 它是 NP 类难题。

美国哈佛大学数学家戴柯尼斯和哥伦比亚大学数学家贝尔, 采取与从不同的策略解决了这个问题, 并得到答案为: 七次。他们采用方法是: 把 52 张牌按 1 到 52 编号并按升序排列。洗牌时, 把牌分成两叠, 一叠从 1 到 26, 另一叠为 27 到 52, 均匀洗一次会出现 1, 27, 2, 28……之后再继续洗, 至于洗几次才能洗得均匀呢? 答案是七次, 若超过七次, 效果反而差。我在 PC 机上用 BASIC 程序模拟洗牌过程。从程序运行结果可以看到: 洗七次最匀, 洗八次则恢复了原样, 真是前功尽弃。这种模拟只是一种特殊情形, 但它却能得出正确答案: 七次!

程序清单如下:

```

LIST
5 DIM A(52), B(52)
10 FOR K = 1 TO 52: A[K] = K: NEXT K
11 LPRINT "THE ORIGINAL STATE IS:"; LPRINT
12 FOR J = 1 TO 26: LPRINT A[J];: NEXT J: LPRINT
13 FOR J = 27 TO 52: LPRINT A[J];: NEXT
J: LPRINT
15 T = 1
20 LPRINT "TIMES = "; T: LPRINT
25 I = 1
30 FOR J = 1 TO 26
35 B[I] = A[J]: I = I + 1
40 B[I] = A[J+26]: I = I + 1
45 NEXT J
50 FOR K = 1 TO 52: A[K] = B[K]: NEXT K
55 FOR K = 1 TO 26: LPRINT A[K];: NEXT
K: LPRINT
60 FOR K = 27 TO 52: LPRINT A[K];: NEXT
K: LPRINT
65 PRINT "RETURN TO CONTINUE": PRINT
67 INPUT C
70 T = T + 1
75 GOTO 20
80 END
OK
RUN

```


单片机演奏乐曲

王兵

单片机以其体积小、功能强、价格低廉而得到广泛应用。本文介绍 8031 单片机演奏音乐的小游戏，仅需要一套 8031 单片机的最小系统（一片 8031 单片机，一片 373，一片 EPROM 和一只 6M 晶体）和一只扬声器及电阻电容，即可连续演奏二个多小时你所喜欢的动听的乐曲。音域为三个八度，可以分辨半音。

一、设计原理

乐曲中的每个音包括二个要素，即音长和音高。8031 单片机具有二个定时器，定时器 0 和定时器 1。在引脚 P1.0 上，如果用定时器 1 控制输出脉冲的周期（即音高），用定时器 0 控制输出时间（即音长），P1.0 上接的扬声器就可以发出一个音符的音响。

1、音长 一张乐谱中有许多不同音长的音，如 16 分音符、8 分音符、4 分音符、2 分音符和全音符等。在 8031 中用定时器 0 产生 200ms 的 16 分音符的音长的中断信号，其它音长（包括附点音符）均可在此基础上累计。如 8 分音符的累计值为 02H，全音符的累计值为 10H。经计算定时器 0 产生 200ms 中断时，其定时器初值为：TH0=3CH，TL0=0B0H。

2、音高 每个音符的音高是固定的，从“1”到“7”36 个半音的频率可用下式计算：

$$f = 261.63 \times (\sqrt[12]{2})^n \text{ Hz}, n = 0, 1, \dots, 35$$

三个八度内的所有半音的音名、频率、定时器 1 定时寄存器初值和单片机中的音名如表 1。感兴趣的读者可以编一段 BASIC 程序算一算。如音“1”，频率为 523.2596Hz，产生该频率定时器 1 寄存器初值 TH1=0FEH，TL1=22H，在单片机中音名为 0DH。

3、音乐表 乐谱是以音乐表的形式事先写入 EPROM 的。地址从 0800H 开始，一个音占二个字节，低位字节存放音名，高位字节存放音长。如二分音符“1-”音名（即音高）为 01H，音长为 08H。杜鹃圆舞曲：3|i-3|553|3|2- 2|4|7- 2|2|5- 2|2|2|2|i-| 的音乐表为表 2。读者可以按规则自建曲目。

二、单片机演奏的实现

硬件电原理图如图 1，实际上是最小 8031 单片机系统，上电自复位。图中 EPROM 以 2764 为例，P1.0 输出端可加入放大环节。

EPROM 地址分配为：0000H 到 06FFH 存放程序，

0700H 到 07FFH 存放音高表，0800H 到 1FFFH 存放音乐表。

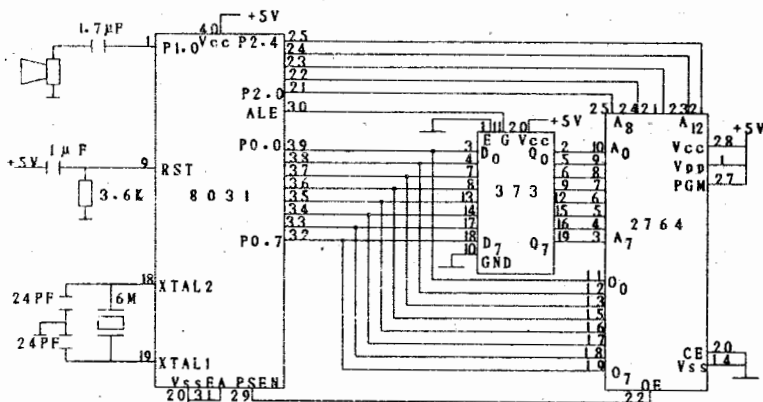


图1 硬件电原理图

8031 内部 RAM 地址分配为：50H 和 51H 存放音乐表指针，40H 和 41H 存放当前演奏的音的音名和音长，42H 和 43H 存放当前音的定时器 1 寄存器初值。

主程序和定时器 1 中断服务程序比较简单。主程序的主要任务是系统初始化和给音乐表指针 50H、51H 赋音乐表首地址 0800H。MOV TMOD, #00010001B 的含义是定时器 0、定时器 1 置为定时 16 位计数方式，MOV IE, #10001010B 的含义是允许定时器 0、定时器 1 中断 MOV IP, #00000010B 的含义是设定定时器 0 中断优先，SETB TR0 的含义是打开定时器 0。请参阅所附程序盘。定时器 1 的中断服务程序的任务是在 P1.0 引脚上输出脉冲。

定时器 0 的中断服务程序框图如图 2。

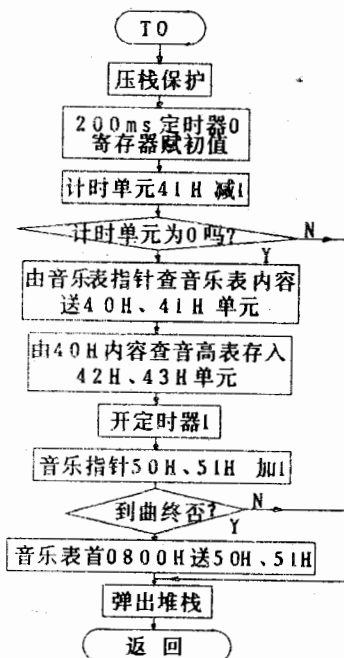


图2 定时器0的中断服务程序框图

演奏杜鹃圆舞曲的程序:(参阅程序盘 ZF23.PRG)

表 1 三个八度内所有半音音阶的频率、定时器 1 定时寄存器初值和单片机中音名 (休止符 TH1=00H, TL1=00H, 音名为 00H)

序号	音名	频率(Hz)	TH1	TL1	单片机中音名
1	1	261.63	FCH	44H	01H
2		277.1873	FCH	7AH	02H
3	2	293.6697	FCH	ADH	03H
4		311.1322	FCH	DCH	04H
5	3	329.6331	FDH	0AH	05H
6	4	349.234	FDH	34H	06H
7		370.0006	FDH	5CH	07H
8	5	392.0019	FDH	82H	08H
9		415.3115	FDH	A6H	09H
10	6	440.0072	FDH	C8H	0AH
11		466.1714	FDH	E8H	0BH
12	7	493.8913	FEH	06H	0CH
13	1	523.2596	FEH	22H	0DH
14		554.3741	FEH	3DH	0EH
15	2	587.3389	FEH	56H	0FH
16		622.2638	FEH	6EH	10H
17	3	659.2655	FEH	85H	11H
18	4	698.4675	FEH	9AH	12H
19		740.0004	FEH	AEH	13H
20	5	784.0031	FEH	C1H	14H
21		830.6223	FEH	D3H	15H
22	6	880.0135	FEH	E4H	16H
23		932.3418	FEH	F4H	17H
24	7	987.7816	FFH	03H	18H
25	1	1046.518	FFH	11H	19H
26		1108.747	FFH	1FH	1AH
27	2	1174.677	FFH	2BH	1BH
28		1244.527	FFH	37H	1CH
29	3	1318.53	FFH	42H	1DH
30	4	1396.934	FFH	4DH	1EH
31		1480	FFH	57H	1FH
32	5	1568.005	FFH	61H	20H
33		1661.243	FFH	6AH	21H
34	6	1760.026	FFH	72H	22H
35		1864.682	FFH	7AH	23H
36	7	1975.562	FFH	81H	24H

三、小结

上面简单介绍了单片机演奏音乐系统的原理和软、硬件, 笔者经过试验, 效果良好。为简单起见存放一个音符占用二字节, 实用上可以用一个字节存放一个音符。比如前 5bit 存放音名, 后 3bit 存放音长。假设每个音符演奏时间平均为 0.5 秒, 使用现行生产的最小容量的 16K 的 EPROM, 其中 1K 用于存放程序, 15K 用于存放音乐表, 则单片机可以演奏二个多小时不重复的乐曲:

$$15 \times 2^{10} \times 0.5 \text{ 秒} = 7680 \text{ 秒} \approx 2.13 \text{ 小时}$$

实际上 32K、64K、128K 大容量 EPROM 逐步得到广泛应用, 有机会使单片机演奏系统能演奏 16 小时的不同乐曲。通过简单的编程, 可以实现乐曲的不同调性、循环演奏、自选曲、快放、慢放等功能。如果采用功耗较小的 80C31 和 CMOS 器件, 随着集成电路造价的降低, 体积的缩小, 单片演奏机很有希望成为人人喜欢的可以随身携带的音乐盒。(本文程序文件名: ZF23.PRG)

表 2 杜鹃圆舞曲音乐表

乐谱	音乐表(从 0800H 开始存放)
3	1DH, 04H
1-3	19H, 08H, 1DH, 04H
1-3	19H, 08H, 1DH, 04H
5 3 3 1 3	20H, 04H, 20H, 02H, 1DH, 02H, 19H, 02H, 1DH, 02H
2- 3 4	1BH, 08H, 1DH, 01H, 1EH, 03H
7- 3 2	18H, 08H, 1AH, 01H, 1BH, 03H
5-2	14H, 08H, 1BH, 04H
4 2 4 2 2 2	1EH, 02H, 1BH, 02H, 1EH, 02H, 1BH, 02H, 18H, 02H, 1BH, 02H
1—	19H, 0CH ■

书讯 《FOXBASE+分析与应用》

FOXBASE+数据库管理系统自 1987 年推出以来, 已先后有 5 个版本问世 (1.00, 1.10, 1.12, 2.00 和 2.10)。这 5 个版本各有什么特点与不同? 本书详细分析了 5 种不同版本的差别、特点、功能以及 FOXBASE+与 dBASEIII 的关系。该书分析全面、透彻, 所谈内容对 dBASEIII 用户改用 FOXBASE+有很大帮助。

从应用角度出发, 书中还介绍了 FOXBASE+编程中的一些技巧, 每种技巧和功能都给出了许多完成的程序实例, 供读者选取。

全书 30 万字, 1991 年 8 月浙江大学出版社出版, 杭州时代书店办理邮购, 每册: 5.20 元 (包括邮挂费)。时代书店地址: 杭州市涌金路 76 号, 邮编: 310002, 开户银行: 建设银行第一办事处, 帐号: 10326119771, ■

V2000 病毒的工作原理与清除

华中理工大学计算机系 88 级 殷军 陶建中

V2000 病毒由 Vesselin Bontchev 于 1989 年制造, 是一种典型的外壳型病毒。它感染的目标是文件长度超过 1958 个字节的 .com 类文件和 .exe 类文件。从其造成的危害性来看, 它属于恶性病毒, V2000 将破坏磁盘上文件的完整性。

一、V2000 病毒的初始化

V2000 病毒的初始化简单流程见图 1。

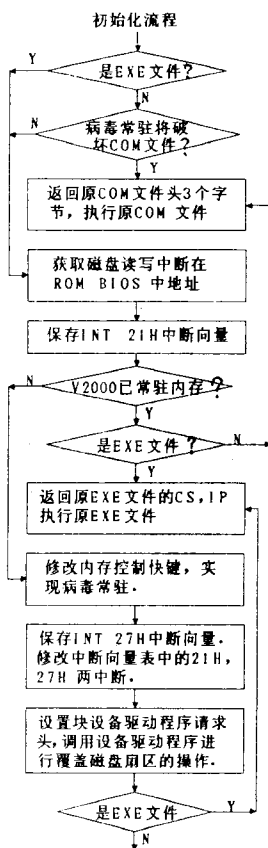


图 1

它主要完成以下三个工作: 获取磁盘读写中断在 ROM BIOS 中的地址, 完成覆盖磁盘扇区的工作以及实现病毒常驻。

V2000 利用磁盘上引导记录扇区的空闲字节设置了一个单字节的计数器和一个双字节的指针。此指针指向上次被覆盖的扇区号。

每当 V2000 病毒被初始化一次, 磁盘上的计数器便计值一次。一旦计数值成为 16 的倍数时, V2000 便开始对磁盘进行覆盖扇区的破坏性操作:

指针加上 64, 指向本次待覆盖的扇区。如果指针所指

扇区号有效 (小于磁盘上总扇区数), 则将 Boot 区内容搬到此扇区进行覆盖。否则取此指针的低 6 位值与文件区起始扇区号之和作为新的指针值, 再进行覆盖扇区的操作。

由于 V2000 病毒常驻在机器内存高端, 因此当病毒载体——当前加载的可执行文件退出加载时, 系统将在缺省驱动器上寻找 command 命令文件以期将常驻程序部分重新装入内存, 此时如果找不到 command 文件, 系统将会提示以下信息:

Insert disk with \command.com in drive A and strike any key when ready

这儿, 假设 A 为当前缺省驱动器。

这一点尤其希望引起注意, 因为一般的小程序是不会破坏掉常驻在内存高端的命令处理程序即 command 的常驻程序的。

二、V2000 病毒的传染

V2000 病毒通过接管系统功能调用 INT 21H 而实现其传染的目的。另外为了防止常驻程序修改 INT 21H, V2000 还截取了 27H 中断。在对文件进行感染的过程中, 它还修改了 13H 中断和 24H 中断。INT 13H 中断新的入口址为 CS: 079CH (注: CS 为 V2000 在内存高端的地址, 以下同)。前面已经提到在病毒初始化的过程中, 它获取了磁盘读写中断在 ROM BIOS 中的地址。事实上, 在对文件进行染毒操作时, 对磁盘的写操作就是通过此地址实现的。

修改 INT 24H 主要是为了防止因对写保护的盘强行进行写操作而出现的严重错误警告。新中断的入口址为 CS: 02EFH, 仅设置了一条指令: MOV AL, 03 便中断返回了。

为了获得对 INT 21H 的首响应权, V2000 是竭力阻止常驻程序再次修改中断向量表。如果发现 21H 中断在中断向量表中的入口址被修改, V2000 便会重新对中断向量进行串链: 将 V2000 设置的 21H 中断重置为链首, 常驻程序的 21H 中断则移到链中, 链尾为原来的 21H 中断。新的 INT 27H 入口址为 CS: 0331H。

修改后的 INT 21H 在病毒程序中的入口址为 CS: 0359H。V2000 对 21H 中断所做的功能号调用修改很多, 以下分别一一介绍之。

AH = 11H 与 AH = 12H。这两个功能调用主要完成匹配目录项的搜索。DOS 内部命令 DIR 就调用了此两个功能。由于 V2000 对这两功能进行了扩充, 因此, V2000 较一般的外壳型病毒来得更为隐蔽: 在遭到 V2000 侵袭的机器内用 DIR 命令查看文件, 会发觉染过毒的文件其文件长度并没有增长!

AX = 2521H 与 AX = 3521H, AX = 2527 与 AX = 3527H。这几个功能调用分别完成对 21H 中断的中断向量的获取与设置以及对 27H 中断向量的获取与设置。V2000 给 21H 中断与 27H 中断的中断向量设置了新的数据区, 分别为 CS: 0818H 与 CS: 0814H。之所以这样做, 是因为可以使 21H 中断链的链首始终为 V2000 设置的 21H 中断。

AH = 31H, 结束过程并驻留。对这一功能调用的截取也是为了防止常驻程序篡改中断向量表中的 21H 中断。

AH = 3CH 与 AH = 58H。这两个功能调用实现文件的建立。在 V2000 中, 此功能调用与 AH = 3EH 即关闭文件功能调用相配合。每当建立一个扩展名为 .COM 文件或者 .EXE 文件, V2000 便在 CS: 07C0H 处保存最新建立文件的句柄号。当关闭文件时, V2000 便判断当前关闭文件是否为最新建立的可执行文件。若是, 则调用感染程序对此文件进行染毒, 否则, 放弃传播工作。

AH = 3DH, 打开文件功能调用。AH = 43H, 改变文件模式功能调用。AH = 56H, 文件换名功能调用。在 V2000 中, 给这几个功能调用扩充的功能为: 若被操作文件为可执行文件, 则调用感染程序子过程对此文件染毒, 否则, 放弃染毒操作。感染程序子过程简单流程见图 2。

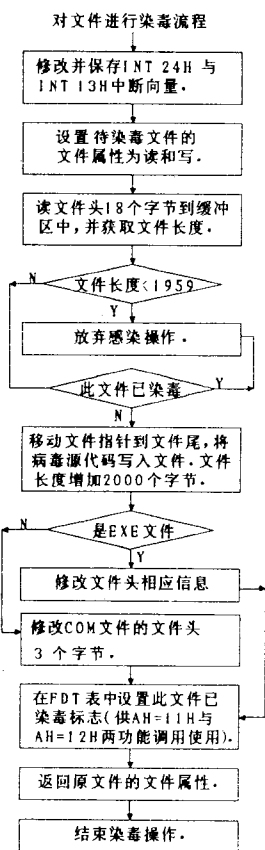


图 2

AX = 4B00H, 装入并执行程序。AX = 4B01H, 获得执行程序的信息。V2000 病毒对此功能调用操作的程序均将进行染毒操作。值得注意的是 AX = 4B00H 功能调用, 由于 DOS 对所有文件的执行都是通过此功能实现的, 因此, 在遭到 V2000 侵袭的系统内加载各程序时, 由于 V2000 做了一些无用功而将降低系统的工作效率。

通过感染程序流程(见图 2)我们知道, 凡是未染上 V2000 病毒的可执行文件均将被感染。事实上, 由于 V2000 常驻内存时破坏命令处理文件 command 的常驻程序部分, 因此第一个最有可能被感染的就是 command 文件。

遭到感染后的 .com 类文件, 其文件头 3 个字节将被修改为 E9n1n2, 其中 n2n1 = 原 com 文件长度 + 005D。而 .exe 文件, 其文件头被病毒所修改。其中 CS = (原文件长度 - 文件头后的节数) + n ([] 表示取整, n 为 0 或 1), IP = 0060H。这样, 当被感染的文件被加载执行是时, 无论是 .com 文件还是 .exe 文件均将先执行 V2000 病毒。

三、V2000 病毒的诊断与清除

如果一个并不是很大的程序(如 10K 左右, 突然破坏了 command 文件在内存高端的常驻程序, 这时我们就应该怀疑系统或许已遭到了 V2000 病毒的侵袭。

进入到 debug 状态下, 获取 INT 21H 中断地址。用 U 命令反汇编 21H 中断, 若头几条指令为:

```
CS: 0359 PUSH BP
CS: 035A MOV BP, SP
CS: 035C PUSH [BP+06]
CS: 035F POPF
```

说明系统已经遭到 V2000 的攻击。

对于文件的诊断可以用 PCTOOLS 或 debug 命令查看文件是否含有以下字符串:

Zopy me - I want to travel

若有, 则说明文件也已遭到感染。

作者编制了一个对单个文件进行检测与消毒的程序, 有兴趣的读者可与作者联系。

地址: 武汉市华中理工大学西六舍 5 号

邮编: 430074

书 讯

- IBM-PC/XT 微型电子计算机系统故障检修 200 例 12 元
- IBM-PC/AT 微机系统原理及维修技术 上册 29 元
下册 20 元
- 微型电子计算机及外围设备电路图集(二) 20 元
- MS-DOS 操作系统结构分析系列教材命令级结构彻底剖析 15 元
- 邮购加收 10% 的邮挂费。
- 地址: 广州华南师范大学电脑杂志社收
- 邮编: 510631 电话: 5516911-3273

“侵略者”病毒分析及防治

福建省政府办公厅技术处 苏武荣

最近, 厦门某合资企业使用台湾版“解毒”程序 KILL13.EXE 时, 发现了一种新奇的 PC 机病毒, 用公安部下发的 224 种病毒扫描程序 SCAN 3.0 能检测到该病毒, 名为“Invader (侵略者)”, 但用 35 种消毒程序 KILL35 无法消除该病毒。经分析, 该病毒是一种混合型病毒, 既能感染可执行文件 (.COM 和 .EXE 文件), 又能感染系统引导记录。该病毒代码量长达 4K, 表现部分使用可变钥加密, 变化无常, 病发时将毫无保留地摧毁硬盘 C: 和 D: 以及驱动器 A: 和 B: 中软盘的所有数据, 破坏力极强。下面分析该病毒, 并给出防治方法。

一、引导过程

该病毒既可由软、硬盘启动时将病毒引入系统, 也可由执行带毒文件时将病毒导入系统。

1、软盘或硬盘带毒启动时, 病毒引导程序将病毒主体程序置于内存高端, 占用 5K 内存, 并初始化有关参数, 把 INT 8H、INT 9H 和 INT 13H 的中断向量分别置为 CS: 0F75、CS: 02B8 和 CS: 0435。然后读取真正的引导记录程序, 并将控制权转之执行。

在此过程中, 新 INT 8H 中断执行的任务很特别, 也很巧妙, 它使 DOS 系统启动后, 自动修改 INT 21H 中断, 从而使该病毒既能感染系统, 又能感染可执行的 .COM 和 .EXE 文件。

2、执行带毒的 .COM 或 .EXE 文件时, 若系统内未驻留该病毒, 则初始化参数, 并修改 INT 8H、INT 9H、INT 13H 和 INT 21H 的中断向量, 然后执行原程序。

其中, 新 INT 8H 执行时间计数, 病发时将连续不断地演奏一首乐曲。新 INT 9H 将截获键盘中断, 按 Ctrl+Alt+Del 时病毒激活病发。新 INT 13H 将完成病毒传染给软硬盘引导记录扇区。新 INT 21H 将完成病毒传染给可执行文件。

二、传染过程

1、通过 INT 13H 感染系统引导记录 (Boot Record), 当程序读取软盘或硬盘系统区时, 读取次数每达三次, 就有一次传染机会。首先读取引导记录扇区, 如相对位置 003E~0041 的内容为“CB 3C FE FF” (即为免疫标志), 则认为已染上该病毒而不再感染, 否则传染之。对硬盘, 将病毒主体程序写入 0 面 0 道第 7 个扇区之后的 8 个扇区, 接着写入真正的引导记录程序, 病毒引导程序写入硬盘引导扇区。对软盘, 先重新格式化第 40 个磁道 (高密盘为第 80 道) 而后把病毒主体程序连同真正的引导程序写入刚格式化的磁道, 病毒引导程序写入软盘引导扇区。

2、通过 INT 21H 感染可执行文件 (.COM 和 .EXE 文件), 当运行文件 (AH=4B) 或打开文件 (AH=3D) 时, 病毒就试图传染。首先计算该文件所在盘剩余空间, 至少要有 4K, 然后读取、保存文件属性以及日期、时间, 并把文件属性强行设置为“可读可写”, 以相对位置 0012~0013 的内容是否为“90 19” (免疫标志), 来判定是否传染, 对 .EXE 文件传染时病毒主体程序附在原文件之后, 对 .COM 文件病毒主体程序加在原文件的前面, 这一点与“黑色星期五”、“星期天”、“落花 1701”以及“蠕虫 1575”等病毒都不同。传染后, 恢复原文件属性和日期、时间。

“侵略者”病毒不传染“COMMAND.COM.”文件和“ACAD.EXE”文件。

三、激发过程

该病毒激发时间很奇特, 带有随机性, 它跟机器运行时间和病毒传染个数有关, 换句话说, 是跟病毒传染速度有关。当某一闪按 Ctrl+Alt+Del 组合键, 试图热启动时, 刚好激发条件满足, 则病发。这时, 屏幕设置为中分辨率方式 (即 320×200), 并测试驱动器 A: 和 B: 中的软盘以及硬盘 C: 和 D: 的参数, 如果参数正确, 就将病毒程序逐一写入软、硬盘的每一面每一个磁道中, 几乎覆盖软、硬盘中的所有内容, 使软、硬盘成为“未格式化”的磁盘, 因此软、硬盘中的所有数据都将报废, 无法恢复。令人啼笑皆非的是, 病毒在完成这个过程中还连续不断地奏一首乐曲, 似乎在奏病毒“攻击”进行曲。

“侵略者”病毒的这种恶作剧极为恐怖, 遭此厄运真是微机用户的最大不幸。由于病发的随机性和特殊性, 因此奉劝微机用户发现“侵略者”病毒时, 不要惊慌失措, 急于用干净的 DOS 盘“热启动”引导系统, 因为以往许多人 (包括一些专家) 曾告诫人们, 发现病毒时应立即用干净的 DOS 盘引导系统。

四、诊断方法

1、检查内存容量: 用 PCTOOLS 工具软件的“System Information”功能查看 DOS 总内存和 PCT00 LS 发现的总内存是否一致, 若刚好少 5K, 可能就有病毒。

2、检查中断向量: 可以用 DEBUG 程序查看 INT 8H、INT 9H、INT 13H 和 INT 21H 的中断向量是否被修改而指向病毒程序, 如: 查看 INT 9H 的中断向量, 执行“D 0: 24 25”, 若有该病毒, 其值应为“B8 02”。

3、检查引导扇区: 可以运行 DEBUG 程序, 把软盘或硬盘的引导扇区调入内存查看。如: 调硬盘 C: 引导扇区, 先执行“L100 201”, 再执行“D100”若有病毒, 其内容就象下面列出的:

```

:0100      E9 9C 00 49 34 12 20 20-33 2E 33 00 02 02 01 00 ...I4.3.3....
:0110      02 70 00 D0 02 FD 02 00-09 00 02 00 00 00 00 00 .P.....
:0120      00 00 00 00 00 00 00 00-49 4F 20 20 20 20 20 20 .....IO
:0130      53 59 53 4D 53 44 4F 53-20 20 20 53 59 53 CB 3C SYSMSDOS SYS.<
:0140      FE FF FE FF 01 28 00 00-4E 6F 6E 2D 73 79 73 74 .....(Non-syst
:0150      65 6D 20 64 69 73 6B 20-6F 72 20 64 69 73 6B 20 em disk or disk
:0160      65 72 72 6F 72 2E 0A 0D-52 65 70 6C 61 63 65 20 error...Replace
:0170      61 6E 64 20 73 74 72 69-6B 65 20 61 6E 79 20 6B and strike any k

```

4. 检查可执行文件: 用 DEBUG 程序把该文件调入内存, 查看程序段内容。先单步跟踪, 后反汇编, 即执行“T”, 再执行“U”。若有病毒, 对于 .COM 文件, 反汇编的内容应为:

```

:0B76 E88AFF      CALL 0B03
:0B79 B3FF        MOV BL,FF
:0B7B B84342      MOV AX,4243
:0B7E CD21        INT 21
:0B80 3D7856      CMP AX,5678
:0B83 7513        JNZ 0B98
:0B85 B84442      MOV AX,4244
:0B88 BF0001      MOV DI,0100
:0B8B 2E          CS:
:0B8C 8B8D1400    MOV CX,[DI+0014]
:0B90 BE0010      MOV SI,1000
:0B93 03F7        ADD SI,DI
:0B95 FC          CLD

```

对于 .EXE 文件, 若有病毒, 反汇编的内容应为:

```

:0AA9 06          PUSH ES
:0AAA E856FF      CALL 0A03
:0AAD 2E          CS:
:0AAE 8C061600    MOV [0016],ES
:0AB2 2E          CS:
:0AB3 8C067400    MOV [0074],ES
:0AB7 2E          CS:
:0AB8 8C067800    MOV [0078],ES
:0ABC 2E          CS:
:0ABD 8C067C00    MOV [007C],ES
:0AC1 8CC3        MOV BX,ES
:0AC3 83C310      ADD BX,+10
:0AC6 2E          CS:
:0AC7 011EED0AADD [0AED],BX

```

五、消毒方法:

一旦发现“侵略者”病毒, 首先应该做的是立即关机, 然后重新开机, 使用正确的 DOS 系统盘从驱动器 A: 启动引导系统。

1. 硬盘消毒: 将 0 面 0 道第 15 个扇区的真正的引导记录程序写回引导记录扇区, 即可消毒。对于 DOS 3.0 以上版本格式化的硬盘, 由于病毒主体程序占用的空间是空闲的, 不会破坏原有数据, 而对于 DOS 2.0/2.1 格式化的硬盘, 病毒主体程序占用的空间恰好是部分 FAT 表及

FAT 副本, 解毒时应把 FAT 表和 FAT 副本的未损坏的内容合二为一, 这样可以救回一部分数据, 如果感染严重, 就无法恢复, 可能会出现硬盘无法启动等现象。

2. 软盘消毒: 由于病毒传染软盘时自身格式化一条空闲的磁道来存储病毒程序, 因此传染时不会破坏软盘的原有数据。消毒时, 只要把 0 面第 40 道第 9 个扇区 (高密盘为第 80 道) 的原引导程序写回软盘引导扇区, 即可消毒。

3. COM 文件消毒: 用 DEBUG 程序把带毒的 COM 文件装入内存, 执行“R CX”, 记下 CX 的值 (设为 X1), 执行“R CX”, 将 CX 的值改为 X1-1000H, 计算 X1+100H 的值 (设为 X2), 执行“M 1100 X2 100”, 把原文件内容移回原来 CS: 100H 处, 再执行“W”和“Q”, 写盘并退出, 即消毒。

4. EXE 文件消毒: EXE 文件较为复杂, 消毒时可按以下步骤:

(1) 用 DEBUG 程序将带毒的 EXE 文件调入, 即

A>DEBUG C: * * *.EXE

(2) 执行“D CS: 040 043”和“D CS: AEB AEE”, 记下该 EXE 文件未带毒前的重要参数, 其中 CS: 040~041 为原 SP 值, CS: 042~043 为原 SS 值, CS: AEB~AEC 为原 IP 值, CS: AED~AEE 为原 CS 值。

(3) 退出 DEBUG, 返回 DOS。

(4) 把带毒的 EXE 文件改名, 如:

A>REN C: * * *.EXE ABC

(5) 将文件 ABC 调入内存, 即

A>DEBUG C: ABC

(6) 执行“R BX”和“R CX”, 把显示出来的 EXE 文件长度 CX 减去 1000H, 若不够减, 则把 BX 的值减去 1, 即将 BX-1 赋值给 BX, 然后用 10000H+CX-1000H, 计算新的 CX, 修改 ABC 文件长度。

(7) 执行“D CS: 102 105”, 获得文件 ABC 所占实际扇区数, 用该数减去 8, 得到消毒后的文件所占实际扇区数。

(8) 用“E”命令修改该 EXE 文件的 5 个重要参数, 其中:

CS: 102~105H 改为消毒后的文件所占实际扇区数

CS: 10E~10FH 改为未染毒前的 SS 值

CS: 110~111H 改为未染毒前的 SP 值

CS: 114~115H 改为未染毒前的 IP 值

CS: 116~117H 改为未染毒前的 CS 值

(9) 执行“W”和“Q”, 存盘并退出, 返回 DOS。

(10) 将 ABC 改回原文件名, 即

A>REN C: ABC * * *.EXE

至此, 消毒完毕, 该文件已不带“侵略者”病毒。

从上述对感染“侵略者”病毒的软硬盘、COM 文件和 EXE 文件的消毒过程, 我们可以领略到排除引导型病毒和文件型病毒的许多带共同性的要领和步骤。根据病毒机理和消毒步骤, 就易于编制相应的检测、消毒软件。

六、免疫方法

1、在每一个软盘和硬盘引导扇区, 以及每一个可执行文件中增加前面所标的“免疫标志”, 这样做代价大, 况且也只能防治“侵略者”病毒, 并不能防治其它种的引导型病毒或文件型病毒。

2、通过更新 21 类中断的 4B 和 3D 等功能调用, 检查跟踪这些功能调用执行前的非法操作, 以诊断是否染上病毒, 阻止带毒程序的执行, 这种方法是“治本”的方法, 但应考虑争夺关键中断控制权等诸多问题。笔者研制的“缉毒警犬”采用了这种方法。 ■

一种精巧的病毒——世纪病毒

吉林工业大学管理学院 朱 军

在众多的病毒之中, “世纪病毒”以其巧妙的设计、严谨的构思被认为是最精巧、灵活的病毒之一。下面笔者就把这种处处透着巧妙的病毒给大家作一简介。“世纪病毒”又称“4096 病毒”、“Frode 病毒”或“一百年病毒”, 是外壳型病毒的一种。它是在 1989 年 10 月发现于以色列的一种感染范围很广的病毒, 它不仅感染 COM、EXE 和 OVL 类型的文件, 而且感染数据文件和 COMMAND.COM。

这种病毒的引入方式类似于小球病毒, 它在内存高端申请 6KB 的存储区域为其专用, 而不是使用 DOS INT21 的 31 号功能驻留在内存的。如果用 CHKDSK 或 PCTOOLS 检查内存容量将比实际少 6KB。

当病毒引入内存并取得控制权后, 它在 COMSPEC PATH 中寻找 COMMAND 然后感染它, 并把存贮在文件目录项中的时间字节增加 100 年, 而 DOS dir 命令执行时只显示文件目录项时间字节中存贮的年份的后两位。正是利用这种难以察觉的更改来作为病毒标识, 从而“知道”哪一个文件已感染, 哪一个文件未感染, 对于已染毒的文件, 在显示其文件长度时病毒把其减少 4KB, 即仍显示染毒前文件的大小。这种巧妙的“隐藏”使得所有凭靠文件长度变化与否来判断有无病毒的反病毒程序束手无策。

“世纪病毒”还运用了巧妙的加密方法和反调试技术。为避免被内存驻留反病毒报警程序察觉, 它使用了巧妙的中断盗用技术, 通过检查内存低端的系统调用的中断向量, 一旦发现某个中断被其它程序所截取, 它就恢复之以

便自身置于任何其它内存驻留程序之前。这样, 任何内存映象实用程序如 SMAP、SNOOP、PCTOOLS 等都无法找出端倪, 尤其对于内存驻留的通过检索中断向量来报警非法病毒存在的反病毒软件将对其毫无知觉。

尽管如此, 还是有办法消除它的, 最简单方便的解毒方法可使用 CARMEL 系列的 Turbo Anti Virus 反病毒软件, 如 TNTVIRUS.EXE 等。

我们研究病毒, 最终是为了达到控制和消灭它们的目的, 但同时我们若能吸取其中的精华, 比如“世纪病毒”中的几个技巧确实有一定借鉴价值, 通过这种途径, 那么软件人员的程序设计水平将逐步向高水平发展, 从而推动软件工程得以迅速发展。 ■

广大一号及其两个变种病毒

广州计算机应用研究所 胡向东

最近, 接连有两个广大一号的变种出笼, 即本文所称的广大一号 A 和广大一号 B。人们用手头上的消毒软件去消毒, 往往不能收效或者根本无法把它检测出来。

广大一号及其两个变种病毒的老祖宗是大麻病毒。因此它们与大麻病毒一样, 是操作系统型的病毒, 攻击的对象是硬盘的主引导扇区和软盘的引导扇区。病毒的长度均为 01BEH, 病毒 int13H 入口均为 XXXX: 0015, 病毒的定时计数器同为 0000: 046C。

大麻病毒显示“Your PC is now stoned!”而广大一号及其两个变种在病毒发作时显示“病毒一号—广州大学 XXX 制作”。

大麻病毒在感染软盘时, 把原引导程序放在软盘的 0 道 1 面 3 区。其写盘语句为:

```
XXXX: 0179 mov AX, 0301 ; 写一个扇区
XXXX: 017C mov BX, 0200 ; 缓冲区首址 200H
XXXX: 017F mov CL, 03 ; 扇区号 3
XXXX: 0181 mov DH, 01 ; 1 面
XXXX: 0183 push F ; 保存标志位
XXXX: 0184 CS:
XXXX: 0185 CALL FAR (0009); 执行原 int 13H
```

而广大一号泡制者为了逃过当时的解毒软件, 采取了程序移位和扇区更改的手法, 在感染软盘时, 把原引导程序放在软盘 0 道 0 面 5 区, 其写盘语句为:

```
XXXX: 0176 mov AX, 0301 ; 写一个扇区
XXXX: 0179 mov BX, 0200 ; 缓冲区首址 200H
XXXX: 017C mov CX, 0005 ; 0 道 5 扇区
XXXX: 017F mov DH, 00 ; 0 面
XXXX: 0181 push F ; 保存标志位
XXXX: 0182 CALL FAR (0009); 执行原 int 13H
```

而广大一号 A 和广大一号 B, 在写软盘语句中, 与广大一号大致相近, 所不同的是, 在感染软盘时, 把原引导程序放在软盘 27H 道 1 面 9 区中, 其语句是:

```
XXXX: 0176 mov AX, 0301 ;写一个扇区
XXXX: 0179 mov BX, 0200 ;缓冲区首址在 200H
XXXX: 017C mov CX, 2709 ;27H 道 9 区
XXXX: 017F mov DH, 01 ;1 面
XXXX: 0181 push F ;保存标志位
XXXX: 0182 CALL FAR (0009); 执行原 int 13H
```

大麻病毒在感染硬盘时, 把原主引导程序放在 C 盘 0 道 0 面 7 区中, 其写盘语句为:

```
XXXX: 025F mov AX, 0301 ;写一个扇区
XXXX: 0262 mov BX, 0200 ;缓冲区首址在 200H
XXXX: 0265 mov CX, 0007 ;0 道 7 区
XXXX: 0268 mov DX, 0080 ;C 盘 0 面
XXXX: 016B int 13 ;调用 int 13H
```

而广大一号在感染硬盘时, 把原主引导程序放在 C 盘 0 道 0 头 8 区中, 其写盘语句为:

```
XXXX: 0239 mov AX, 0301 ;写一个扇区
XXXX: 023C mov BX, 0200 ;缓冲区首址 200H
XXXX: 023F mov CX, 0008 ;0 道 8 区
XXXX: 0242 mov DX, 0080 ;C 盘 0 面
XXXX: 0245 int 13 ;调用 int 13H
```

广大一号 A 和广大一号 B 在感染硬盘时, 只是在原广大一号的基础上更改了扇区位置, 在病毒感染时把原主引导程序放在 C 盘 1 道 1 面 0FH 区中, 其写盘语句是:

```
XXXX: 0239 mov AX, 0301 ;写一个扇区
XXXX: 023C mov BX, 0200 ;缓冲区首址在 200H
XXXX: 023F mov CX, 010F ;1 道 0FH 区
XXXX: 0242 mov DX, 0180 ;C 盘 1 面
XXXX: 0245 int 13 ;调用磁盘中断 int 13H
```

广大一号 B 是在广大一号 A 的基础上发展起来的, 尽管广大一号 B 在感染软盘和感染硬盘时是把广大一号 A 中的有关语句原封不动地搬过来, 因而两种病毒的消毒方法完全相同, 但广大一号 B 在病毒的首语句中的确下了很大的功夫。大麻病毒、广大一号、广大一号 A 这三种病毒的首语句均为 JMP 07C0: 0005。

而广大一号 B 的首语句则和 MSDOS3.0 引导程序中的首语句相同, 即是 JMP 0136。因此许多检测、消毒软件不能检测出广大一号 B 更不能对它正确消毒, 即使是用 debug 把病毒调入内存, 许多人也会被广大一号 B 的首语句迷惑。

对广大一号病毒的检测及消毒方法, 可参考 1990 年第 6 期《电脑》杂志胡向东所写的《大麻、广大一号、六·四病毒及打印病毒》一文。

对操作系统型病毒如何使用 debug 作工具把软盘引导扇区或硬盘主引导扇区调入内存, 该文已作了介绍。

当把引导或主引导扇区的内容调入内存 CS: 100H 处时,

判别广大一号 A 的依据是:

- (1) 首语句为 JMP 07C0: 0005
- (2) XXXX: 023F 处的语句为 mov CX, 010F
mov DX, 0180
int 13H

判别广大一号 B 的依据是:

- (1) 首语句为 JMP 0136
- (2) XXXX: 023F 处的语句为 mov CX, 010F
mov DX, 0180
int 13H

用 debug 清除 360K 软盘上的广大一号 A 和广大一号 B, 可以按如下二个步骤做:

- (1) L 0100 0 2CF 1↓ (假定是 A 盘)
- (2) W 0100 0 0 1↓

用 debug 清除 1.2M 软盘上的广大一号 A 和广大一号 B, 可以按如下二个步骤做:

- (1) L 0100 0 4A9 1↓ (假定是 A 盘)
- (2) W 0100 0 0 1↓

用 debug 清除硬盘上的广大一号及二个变种, 所使用的方法同清除大麻 (又名硬化、固化、石头) 病毒一样, 详情可参考 90 年第 1 期《电脑》杂志上胡向东所写的《硬化病毒的分析及排除方法》。

广大一号的炮制者是谁, 已明明白白地写在广大一号的病毒提示中, 省市公安部门早已备了案。然而广大一号 A 及广大一号 B 的炮制者又是谁, 至今仍然是一个谜。

但从短短的时间里广州接连出现广大一号 A 和广大一号 B 以及本文所提供的资料来看, 不管广大一号 A 和广大一号 B 的炮制者如何自鸣得意, 他的狐狸尾巴实际上是已经暴露出来了!

广大一号及其二个变种已给许多计算机用户造成人力物力的损失, 造成不少计算机系统数据丢失甚至瘫痪。因此, 一定不能让这些病毒的炮制者逍遥法外!

一定要他们尝到“玩火者必自焚”的滋味!

这, 就是广大计算机用户的心声。因为计算机病毒及其炮制者, 实在太可恶了! ■

小 启

漏订明年《电脑》杂志的读者, 可直接汇款到编辑部订阅。

全年 6 期共 6.60 元 (含邮费)。

地址: 广州市石牌华南师范大学电脑杂志社

邮编: 510631 电话: 5516911-3273

四通 2401—微机数据库文书管理系统

珠海丽珠医药 (集团) 公司电脑室 林胜藩

四通 2401 电脑打字机使用已相当普遍, 文件编辑、存放、修改都很方便, 但四通 2401 打字机不具备文书管理功能, 当文件多时, 文件的检索、统计只能靠人工进行。我们可以利用 dBASE III (FoxBASE) 在微机上实现文书管理, 问题的关键是怎样把四通 2401 上使用的 3.5 寸 720K 软盘上的文件信息直接读入微机上的 dBASE III (FoxBASE) 数据库, 而不必采用通讯转换或整体文件拷具的方法在微机上复制四通文件。

只要微机上配备有 3.5 寸软盘驱动器和运行 DOS3.0 以上的版本, 可以在微机上直接读出四通盘上的文件目录。由于四通 2401 文件不是以 ASCII 码形式存放的, 不能直接在微机上用 TYPE 命令显示或用 WORDSTAR 等常用的文本编辑工具进行编辑, 但其目录形式和文件装裁是与 DOS 兼容的。对于一般的文书管理而言只要读出文件目录信息就足够了。

下面是在微机上读出一个四通 2401 盘目录的例子:

Volume in drive A has no label

Directory of A: \

说明书	1795	6-25-90	8: 28a
可供表 FMT	273	11-22-90	10: 46a
可供表	1988	11-22-90	10: 46a
LZR241 BAK	8882	11-14-90	3: 49p
LZR241	8882	11-14-90	3: 51p

5 File (s) 706560 bytes free

表一

为了录入四通文件信息, 采取如下步骤: 第一步, 建立目录数据库 ST.DBF 存放一个四通磁盘的文件目录, 库

的结构如下:

Structure for database: C: \ST.DBF

Number of data records: 0

Date of last update: 06 / 29 / 90

Field	Field Name	Type	Width	Dec
1	文件名	Character	9	
2	扩展名	Character	3	
3	长度	Numeric	9	
4	日期	Character	10	
5	时间	Character	8	
6	盘号	Character	6	
* * Total * *			46	

表二

数据库中的日期字段采用日期型, 为了使与 DIR 命令产生的文件目录中的日期格式相一致, 在录入数据时, 程序加插了 SET DATE AMER 语句。

在 DOS 下运行 DIR A: >ST.TXT 把插在 A 盘中的四通磁盘上的目录送到文本文件 ST.TXT 中, 文件格式正好同表一, 其中表头 (驱动器号、标号、路径和表头空行)、表尾 (文件总数、可用盘空间字节数) 对我们是无用的, 在转录入数据库 ST 后去除。

下一步, 在 dBASE III (FoxBASE) 下, 通过下述命令:

USE ST

ZAP

APPE FROM ST SDF

把 ST.TXT 目录文件的内容全部录入到 ST.DBF 中,

M1724 打印机中 DIP 开关的设置对打印的影响

江苏省溧水县统计局 曹 详

我们在使用 M1724 打印机的过程中, 发现在不同的系统不同的驱动程序下, DIP 开关第七位的设置直接影响字符的打印效果, 因为当 SW7 设在 OFF 档时, 打印数据和打印头 24 针的对应关系是 DATA1-8 对 PIN1-8, DATA1-8 对 PIN9-16, DATA1-8 对 PIN17-24; 当 SW7 设在 ON 档时, 数据位则是从 DATA8-1 依次对应

各针位, 所以如果 SW7 设置不当, 那打印出字符就会混乱不堪, 有的就是完全颠倒, 一般在 CCDOS4.0 下 SW7 设在 OFF 状态, 在 2.13 系统中则要设在 ON 状态, 在科印排版系统下也要设在此状态。要注意的是每次新的设置都要关机再开机后才有效。 ■

然后去除表头部分所占的 4 个记录和表尾部分所占的最后一个记录 (通过 GO BOTT 得到)。此外, 还可以根据需要对四通盘上的文件进行过滤, 删去对我们无用的文件, 如. \$\$\$, .FMT, .BAK 等文件, 这部分的命令如下:

```
GO BOTT
STOR RECNO() TO BT
DELE FOR RECNO() < 5 OR RECNO() >= BT. OR.
扩展名 $; 'COMEXEBAKSYSPRGIDXNDXBASDBFMT
$$$'. OR. 长度=0
```

```
PACK
REPL ALL 日期 WITH SUBS (日期, 3), 盘号 WITH
NO
```

这样, ST.DBF 中就录入了一片四通磁盘中有用的文件信息。每录入一片四通盘, 只要输入盘号就行了。盘号是字符型的, 可根据需要自定, 录入操作由计算机自动完成, 录入程序见后面清单。

下一步, 要把 ST.DBF 的记录加到主数据库 STWJ.DBF 中, 录入前要做几件工作:

其一, 防止重复录入, 主库 STWJ.DBF 中已记录的文件不重复录入; 其二, 主库中虽有但已更新的文件 (如盘号、文件名相同, 文件日期、文件长度已改变) 要更新, 即以新代旧; 其三, 根据每个文件的长度和预定的打印单价, 顺便算出各个文件的打印费。为此, 处理方法是先把 ST.DBF 库的内容拷入中间数据库 ST1.DBF 中, 经加工后再添加入主库 STWJ.DBF 中, ST1.DBF 的结构与主库完全相同。

STWJ.DBF 和 ST1.DBF 结构如下:

Structure for database: C:\ST1.DBF

Number of data records: 0

Date of last update 09/04/90

Field	Field Name	Type	Width	Dec
1	文件名	Character	8	
2	扩展名	Character	3	
3	长度	Numeric	9	
4	日期	Date	8	
5	时间	Character	8	
6	盘号	Character	6	
7	内容	Character	40	
8	单位	Character	12	
9	单价	Numeric	4	2
10	打印费	Numeric	6	2
11	备注	Character	8	
**	Total **		113	

经这样处理后, 录入四通磁盘的顺序可以是任意的, 每片盘的录入次数也不受限制, 既可以随时录入四通磁盘信息, 也可以定期分批录入。计算机数据库 STWJ.DBF 中只添加新的或已更新的文件。便于打字员或电脑操作员

进行操作。STWJ.DBF 库中每个记录中的“单位”、“内容”字段用于记录文件主和文件内容摘要, 在文书管理中用于模糊检索和统计, 这两个字段的内容由另外的“输入文件内容摘要”或“修改数据”模块完成。在录入四通盘文件目录信息的过程中, 文件基本打印费也已自动算出。至此, 可以利用 dBASEIII、FoxBASE 的所有编程技巧实现对四通文件的管理和其他应用。

笔者编写的“四通文书管理系统”程序主菜单如下:

```
~~~~~
§      文书管理程序提供如下服务      §
§      1.输入软盘文件数据  2.建立索引    §
§      3.输入文件内容摘要  4.检索        §
§      5.修改数据          6.删除        §
§      7.恢复暂时被删文件  8.整理文件    §
§      9.统计文件数量字数  0.退出        §
~~~~~
请做出您的选择.....
```

程序包括了主程序 STM. PRG 和过程文件 STPRO. PRG, 程序成功地对两台 2401 打字机的近两千个文件进行管理, 运行两年多来, 文件不断增加和更新, 文件的查询、统计计费、文件重打印重修改相当方便, 有力地促进生产、经营和办公自动化。程序中关键部分“录入软盘文件数据”如下:

PROC ST

* 输入软盘目录 ST. PRG

SET TALK OFF

SET SAFE OFF

DO WHIL .T.

SELE 1

SET DATE ANSI

USE STWJ INDE STWJ, STDW, STWJN ALIA DE
COPY STRU TO ST1

CLEAR

SET COLO TO G+

STOR SPAC (6) TO NO

WAIT“请把四通磁盘插入 A 驱动器, 按任一键继续.....”

@10, 10 SAY “请输入盘号: ” GET NO

READ

NO=UPPE (NO)

SET COLO TO GB

@12, 16 SAY “正在进行处理, 请稍等.....”

RUN DIR A: >ST.TXT

SELE 2

SET DATE AMER

USE ST

ZAP

APPE FROM ST SDF

GO BOTT

STOR RECNO() TO BT

一种通用的外部命令口令设置法

四川攀钢轨梁厂 易远泽

在日常工作中,因为操作员的好奇或一时的疏忽,误用类似 FORMAT 这样的外部命令,造成无法挽回的数据丢失的现象时有发生,有些应用程序(应用程序本身也是外部命令)若不按规定运行也会更改或删除某些有用的数据。能否给类似的外部命令设置一个口令来防止这样的误动作呢?回答是肯定的。给外部命令(其中包括应用程序和一些系统软件)设置口令的方法很多,一种方法就修改外部命令本身,但是,这种方法既繁琐而又不灵活,并且必须对要设置口令的所有外部命令进行一一修改,这很不方便。现介绍一种方法,可以同时任意多的任何外部命令设置口令,既简单灵活,而又不需修改外部命令文件本身。

我们知道,任何外部命令(包括应用程序和一些系统软件)都是通过操作系统提供的一个被称为 COMMAND.COM 的命令解释程序加载执行的,我们又知道,操作系统的 21H 号中断的 4BH 号功能调用,它允许一个应用程序运行另一个应用程序,COMMAND.COM 正是利用这一功能加载执行外部命令

```
DELE FOR RECNO() < 5.OR.RECNO() > =BT.OR.
```

扩展名\$, 'COMEXEBAKSYSPRGIDXNDXBASDBFMT

```
¥¥¥'.OR.长度=0
```

```
PACK
```

```
REPL ALL 日期 WITH SUBS (日期, 3), 盘号 WITH NO
```

```
USE ST1
```

```
APPE FROM ST
```

```
SET DATE ANSI
```

```
SET RELA TO DTOC (日期) + 文件名 + TRIM (盘号)
```

```
INTO DE
```

```
GO TOP
```

```
SET COLO TO 3
```

```
STOR 0 TO K
```

```
CLEA
```

```
DO WHIL .NOT.EOF()
```

```
IF 长度 = DE * 长度
```

```
REPL 备注 WITH "N"
```

```
ELSE
```

```
REPL 单价 WITH 2.0, 打印费 WITH 长度 * 2.0 / 200
```

```
K = K + 1
```

```
@10, 5 SAY STR (K, 5) + ":" + 文件名 + 扩展名 + " " + DTOC (日期)
```

```
ENDI
```

的。

基于以上事实,我们可以设想,能否编写这样一个中断程序:

1.此中断程序是 DOS 原 21H 号中断的一个外套,用于截获 DOS 的 21H 号中断的 4BH 号功能调用,检查其是否要加载程序执行,若是,则进一步检查其合法性,否则转原 21H 号中断处执行。

2.在我们的中断程序中,建两张表:一张用于存放需要设置口令的所有外部命令的文件名(不需要扩展名);一张用于存放与其相应的口令。

3.把从 4BH 号功能调用中获得的外部命令文件名(DS: DX) 与我们的外部命令表中的文件名进行比较,若找到相同的文件名,则要求给出正确的口令以后方可继续执行,否则返回一错误码退出,若没有找到相同的文件名,则说明并没有要求给该文件设置口令,转原 21H 号中断处执行。

下面给出按上述设想编写的一个中断程序,经调试、编译、连接,然后转化成 COM 文件,在 IBM-PC/XT

```
SKIP
```

```
ENDD
```

```
USE
```

```
SELE 1
```

```
USE STWJ INDE STWJ, STWJN, STDW
```

```
APPE FROM ST1 FOR 备注 < > "N"
```

```
USE
```

```
STOR " " TO HD
```

```
@18, 20 SAY "还继续吗?" GET HD PICT "!"
```

```
READ
```

```
IF HD = "N"
```

```
EXIT
```

```
ENDI
```

```
ENDD
```

```
CLOSE DATA
```

```
DELE FILE ST.TXT
```

```
CLEA
```

```
SET SAFE ON
```

```
RETU
```

注:主库 STWJ.DBF 的三个索引文件是:

主索引 STWJ.IDX: INDE ON DTOC (日期) + 文件名

按单位索引: INDE ON 单位 + DTOC (日期)

按盘号索引: INDE ON STR (VAL (盘号, 3)) + 文件名 ■

及兼容机上运行通过。具体步骤如下:

```
C> MASM INT4B; <CR>
C> LINK INT4B; <CR>
C> EXE2BIN INT4B.EXE INT4B.COM <CR>
C> DEL INT4B.EXE <CR>
C> INT4B <CR>
```

如果用户需更新口令或外部命令表,一种方法是使用调试程序 DEBUG.COM 修改 INT4B.COM 文件的外部命令表和口令表,另一种方法是编写一个小程序来实现对 INT4B.COM 中两个表的修改。

若把 INT4B.COM 放在 AUTOEXEC.BAT 中,则每次启动机器后,INT4B.COM 便自动驻留内存而起作用。由于程序短而精,仅占用大约 0.5K 字节空间,所以基本不会影响程序加载速度。

```
;
; =====
; Copyright (C) 1991 Yi Yuan Ze
; This program name is:INT4B.ASM
; To compile, link, and convert this program into a COM file
;
; =====
```

code segment

assume cs:code,ds:code

:org 0100h

start: jmp begin

main proc far

cmp ax,4b00h

jz intn

jmp cs:doptr

intn: sti

push ds

push es

push ax

push bx

push cx

push dx

push di

push si

push bp

push ds

pop es

push cs

pop ds

cld

mov di,dx

mov cx,50h

mov al,2eh

repne scasb

jnz int5

dec di

mov bx,di

std

mov cx,50h

mov al,5ch

repne scasb

jnz int5

inc di

inc di

cld

sub bx,di

mov bp,di

mov ax,offset fitab

mov dh,10

int0: mov cx,bx

mov di,bp

mov si,ax

repz cmpsb

jz int1

add ax,8

dec dh

jnz int0

jmp int4

int1: mov dl,10

sub dl,dh

xor dh,dh

shl dx,1

shl dx,1

mov si,offset patab

add si,dx

mov ah,9

mov dx,offset msgpas

int 21h

dec si

mov dx,5

int2: inc si

dec dl

jz int3

mov ah,7

int 21h

cmp byte ptr [si],al

jz int2

inc dh

jmp int2

int3: or dh,dh

jnz int5

int4: pop bp

pop si

pop di

pop dx

pop cx

pop bx

pop ax

pop es

pop ds

jmp cs:doptr

nt5: pop bp

pop si

pop di

pop dx

pop cx

pop bx

pop ax

pop es

pop ds

mov ax,1

谈谈 DOS 与 XENIX 系统共享硬盘

建设银行中山市分行计算机应用管理科 朱晓敏

前言

随着计算机应用的发展,现在越来越多的用户在购买微机时,一般以选择 286 或者 386 机为目标,不仅因为其价格下降,而且因为其性能比 PC/XT 提高三倍以上。但是一台 286 或 386 机在 DOS 下运行时只能起到一台高速的 PC/XT 机作用,其优点远远没有发挥出来,286 或 386 机只有在多用户多任务操作系统下,其性能才能得到完整体现。以 286 机为例,首先,在内存方面,DOS 系统下的 80286CPU 是工作于实模式,在这种模式下,CPU 的 24 根地址线中只有低 20 位有效,其存储器寻址能力最大为 1MB,用户即使配备了多于 1MB 的内存,其多余部分也不能使用;而多用户操作系统下的 286 机其 CPU 工作于保护模式,在这种模式下,CPU 的 24 根地址线全部有效,因此其内存直接寻址能力多达 16MB。其次,在硬盘管理方面,DOS 系统管理硬盘空间能力有限,用户常用的 DOS3.3 版本,其管理的硬盘空间为 33MB,286 机配备的硬盘通常大于 33MB,多出的部分空间只能划分出一个逻辑硬盘来管理,从而人为地将一个硬盘分成了两个文件系统;而多用户下的 286 机,由于发挥了 CPU 虚拟存

储功能,其逻辑寻址能力达 1GB,所以多用户系统下的 286 机可以管理足够大的硬盘空间。再次,DOS 系统下的 286 机工作于单用户单任务方式,系统资源一人独占;而多用户多任务系统下的 286 机,如果配备多用户卡及中西文终端,则组成为一个局部网络,多个用户可同时共享系统的软硬件资源,从而打破了一人垄断一台机的局面。尽管配备多用户操作系统的 286 或 386 机有其优点,但由于目前大部分的应用软件是在 DOS 系统下开发的,必须在 DOS 系统下才能运行,对于想在微机上安装多用户操作系统的用户势必要考虑原有 DOS 软件运行问题,为了解决这一矛盾达到一机多用的目的,因此就提出了两种操作系统共享硬盘的设想。目前微机上较为流行的多用户多任务操作系统是 XENIX 系统,本文就如何在同一硬盘上安装启动及使用这两个操作系统谈些实践体会。

一、安装前的准备工作

在一台微机的硬盘上安装 XENIX 及 DOS 系统,首先要做的工作是给硬盘分区,分区可用 DOS 及 XENIX 的 FDISK 程序。如果原先已将硬盘分成了 DOS 分区及逻辑分区或全部划给了 XENIX 系统,则应先用 DOS 或 XENIX 的 FDISK 程序将分区删除;对于刚出厂的硬盘,一般已将它分成了一个 DOS 分区及一个非 DOS 分区,这时可用随机测试盘的低级格式化程序将硬盘做一次低级格式化,然后再分区。具体分区步骤请参考 DOS 及 XENIX 的 FDISK 程序的用法。要注意的是,DOS 的 FDISK 是由用户在 DOS 系统下调用的,而 XENIX 的 FDISK 是在 XENIX 安装过程中,由安装程序自动调用的,已安装 XENIX 的用户也可在 XENIX 系统中调用 FDISK 程序来分区。XENIX 和 DOS 哪个占第一分区都可以,但用户在建立这两个系统分区时会看到 DOS 与 XENIX 的 FDISK 所显示的分区表中,两个分区的位置不一致,当 DOS 为第一分区时,在 DOS 的 FDISK 所显示的分区表中,DOS 分区放于第一栏,XENIX 分区放于第二栏,但在 XENIX 的 FDISK 所显示的分区表中则刚好相反,其实这并不表明 XENIX 分区是第一分区,而主要是看分区所在的起始柱面或磁道的位置,处在硬盘分区开始柱面的或磁道的那个分区应为第一分区。另外,两个系统的 FDISK 所显示的分区起始终止及分区大小信息亦不一样,这是因为 DOS 的 FDISK 所显示的分区信息是用柱面来标识的,而 XENIX 是以磁道来标识的,两者之间的关系是:磁道数=柱面数×硬盘的磁道数。DOS 和 XENIX 分区各占多少要看硬盘容量及用户的使用情况,当用户的主工作区为 XENIX 分区时,可将该区分大些,否则 DOS 分区大些,但最好保证 XENIX 分区 10MB 以上,以便于系统运行自

```

            ired
doptr dd      0
msgpasdb     0dh,0ah,'# # # Please input PASSWORD
# # #',0dh,0ah,'$'
fitab db      'FORMAT'
db           'CHKDSK'
db           'FDISK'
db           'INT4B'
db           6*8 dup(' ')
patab db      'yyz0'
db           'YYZI'
db           'disk'
db           'intB'
db           6*4 dup(' ')
main endp
begin: mov     ax,0
mov     es,ax
mov     bx,0084h
mov     ax,es:[bx]
mov     cx,es:[bx+2]
mov     word ptr doptr,ax
mov     word ptr doptr[2],cx
mov     es:[bx],offset main
mov     es:[bx+2],cs
mov     dx,offset begin
int     27h
code ends
end      start (本文程序文件名: ZD15.PRG)

```


如。

二、DOS 及 XENIX 系统的安装

DOS 和 XENIX 的安装顺序可以任意选择，一般来讲先安装 DOS 系统方便些，DOS 分区建立之后即可按下面步骤安装：

(1) A 盘启动

(2) A>FORMAT C: /S

(3) A>COPY *.* C:

完成之后重新启动系统，插入 XENIX 安装盘即可进行 XENIX 系统安装。在安装 XENIX 过程中，安装程序会调用 FDISK 程序提示用户建立 XENIX 分区，如果原先已建立该分区并已激活则可跳过这一步，否则要仔细输入分区起始磁道及分区大小等信息并将分区激活，注意不要与 DOS 分区的磁道重叠，然后用完整扫描方式将 XENIX 分区格式化。在扫描过程中，如果发现分区起始位置有坏道，应回到 FDISK 重新分区，跳过坏道，否则 XENIX 将无法引导。由于 XENIX 系统的安装比较复杂，详细过程请参考 XENIX 用户手册，这里不详叙。

三、DOS 与 XENIX 的启动过程

在谈两个系统启动过程之前，先来粗略分析一下两个系统在硬盘上的分区情况。

DOS 和 XENIX 的 FDISK 程序执行之后，硬盘的 0 头 0 柱面一扇区便建立了一张硬盘分区表及主引导记录，前者描述了硬盘分区状况，其中包括了分区个数每个分区的起始磁道分区大小及自举标志等信息。后者供硬盘自举时执行，以识别哪一个分区能自举并有效。真正的 DOS 及 XENIX 的引导记录是放于 0 头 0 柱面 1 扇区之后各分区之首。它们是在安装系统时装入的。

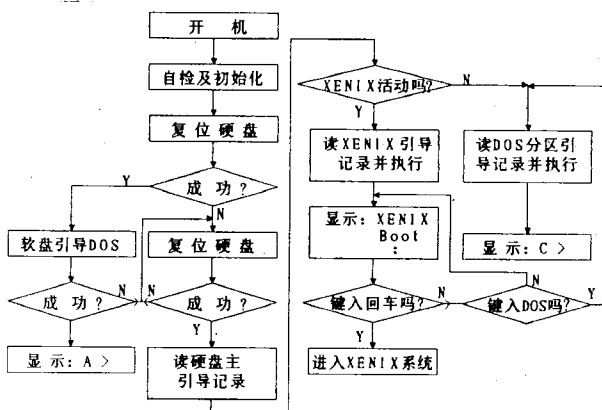
DOS 与 XENIX 的启动过程如框图所示，开机自检及初始化系统之后，先复位软盘系统，如果 A 驱动器中已准备好 DOS 系统盘即启动 DOS，否则复位硬盘系统，成功之后读入硬盘 0 头 0 柱面一扇区的主引导记录并执行，当确认了哪个系统分区为可自举时，即从该分区的读入引导记录并执行，从而启动该系统。不过 DOS 与 XENIX 的引导过程不一样，DOS 的引导程序直接将 DOS 系统的三个文件 (IBMBIO.COM IBMDOS.COM COMMAND.COM) 读入内存并把系统控制权交给 DOS；而 XENIX 的引导程序还留有一个出口，即在引导的过程中，让用户选择要引导哪个操作系统，在引导的过程中会出现如下信息：

```
xenix
boot
```

并等待，这时用户如果键入回车或等待超时，引导程序便装入 XENIX 核心，并将系统控制权交给 XENIX，如果用户键入 DOS 三字符，则表明用户要引导 DOS 系统，这时程序转向装入 DOS 引导程序并执行即启动 DOS。

从以上分析来看，将 XENIX 指定为活动分区在启动

系统时有选择余地，如果将 DOS 分区指定为活动分区，当要启动 XENIX 时，就得在 DOS 下将 XENIX 分区激活并重新启动系统，才能实现 XENIX 的引导。DOS 为非活动分区的情况下，用 DOS 系统软盘启动后，用户仍然可以进入硬盘的 DOS 分区使用硬盘要注意的是，XENIX 不可以用软盘来启动。



四、DOS 与 XENIX 之间的联系

XENIX 系统本身提供了一组实用程序帮助用户在 DOS 与 XENIX 之间建立联系，这组程序的程序名都冠以 dos 字样，如 dosls doscp doscat 等，使用户在 XENIX 系统下能存取位于硬盘 DOS 分区的文件及目录，格式化 DOS 软盘等。XENIX 系统下的 /etc/default/msdos 文件中安排了 DOS 文件系统提示符 (如 A: B: C: 等) 对应于 XENIX 系统中 /dev 目录下的 XENIX 设备文件。虽然 XENIX 系统下不能运行 DOS 软件，但可以利用 XENIX-DOS 交叉开发系统在 XENIX 下建立起能够在 DOS 系统下运行的程序。另外，DOS 系统下的 DBASE INFORMIX UNIFY 数据库文件稍作转换即可成为 XENIX 下的 INFORMIX UNIFY 等数据库文件，反之亦然。XENIX 与 DOS 之间的这些联系，无疑给那些在硬盘上安装有两个操作系统的用户带来方便。

参考文献：

《SCO-XENIX V 系统技术丛书》中科院希望高级电脑技术公司

《DOS 高级技术分析》张载鸿编国防工业出版社出版

《IBM-PC/AT 微机系统原理及维修技术》(上册) 华南师范大学计算机系

小 启

本人愿与各地计算机爱好者交个朋友，一起交流经验与心得。另：为感谢大家的支持，本人特备数百种最新之 IBM 软件赠送给各位朋友及《电脑》读者，欢迎交换软件，详情可来函联系。

联系地址：广州市东兴中路东兴一街 4 号 801

邮 编：510600

联系人：杨 军

提高冒泡排序效率的有效方法

华南师大电教系 罗国富

一、引言

排序是一种十分重要的运算。这种运算既可以简化数据的检索,又可以提高计算机存取数据的效率。因此,排序在各种计算机管理系统的软件中得到了广泛的应用。排序方法虽有很多,但由于冒泡排序是一种简单而著名的排序方法,因而在 BASIC 程序设计中最为采用。目前的 BASIC 语言教科书中,虽然几乎无一例外地都对冒泡排序方法作了介绍,有的还对原来的排序程序作了改进并给出了改进后的程序。但对它们在实际应用中存在的致命的低效率问题并没有进行讨论。为寻求更高的排序效率,必须对现有的冒泡排序程序作再进一步的改进。本文将对现有的冒泡排序程序存在的问题作一深入的分析,并在这一基础上提出一种有效的改进方法。

二、目前的冒泡排序程序排序效率低的原因

为对冒泡排序程序提出切实可行的改进意见,实现更高的排序效率,首先需对冒泡排序的排序过程进行深入的分析,找出低效率的原因所在。冒泡排序的排序过程是:设表长为 n ,首先对表中 n 个数据进行顺序扫描,每次扫描比较相邻两个数据值的大小,若为逆序,则两两交换数据位置。这样,经过 $(n-1)$ 次扫描(每次扫描都进行 $(n-1)$ 次的比较)后,表中全部数据便依序排列好了。实现这一排序过程的 BASIC 程序如下:

程序 1

```
100 FOR I=1 TO N-1
110 FOR J=1 TO N-1
120 IF A (J) >= A (J+1) THEN 140
130 T=A (J): A (J) = A (J+1): A (J+1) = T
140 NEXT J
150 NEXT I
```

这是一个大家熟悉的冒泡排序程序。程序中通过外重循环控制表的扫描次数,通过内重循环控制每次扫描时数据的比较次数并完成逆序时数据的交换。利用程序 1 可实现任意表的排序。显然,该程序的排序过程是十分清晰的,用它来帮助初学者理解冒泡排序的基本思想无疑是一个很好的程序。但在实际应用中,它却是一个低效率的排序程序。

为方便说明,现给出一个未排序的表(6, 7, 3, 4, 8)并将表的数据写成一竖列(如图 1)。

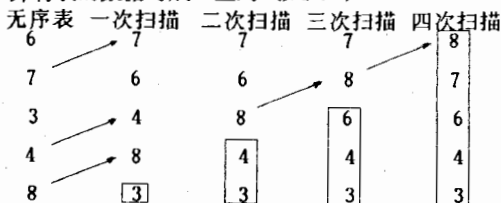


图1

(注:方框内的数据为经过某一次扫描后已排好序的数据)

由图 1 可见,其排序过程(降序)是:

第一次扫描时:

四次比较	(1) 6和7比较	逆序	交换位置
	(2) 6和3比较	顺序	不交换
	(3) 3和4比较	逆序	交换位置
	(4) 3和8比较	逆序	交换位置

可见,完成第一次扫描后,表最底下的数据“3”就排好了。

第二次扫描时:

四次比较	(1) 7和6比较	顺序	不交换
	(2) 6和4比较	顺序	不交换
	(3) 4和8比较	逆序	交换位置
	(4) 4和3比较	顺序	不交换

这样,经过第四次(为 $n-1$ 次)扫描后,便得到了一个排好序的表(见图 1 最右边一列)。

实际应用中,并不是所有的数列都得经过 $(n-1)$ 次的扫描才能完成排序工作的。如图 2 所示,数列(21, 15, 4, 17, 6)只做 2 次的扫描就已完成排序工作了。

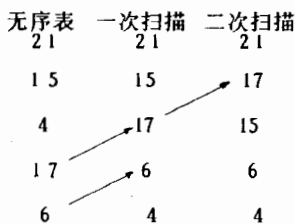


图2

通过图 1 和图 2 数列排序过程的分析,我们不难发现,冒泡排序程序(程序 1)排序工作效率低的原因有两个方面:

1、由于对任意数列(即使是图 2 中这样的数列)都必须经过 $(n-1)$ 次的扫描后才结束排序工作,因而使得像图 2 中这样的数列多做了没有必要的扫描次数,浪费了机时,导致排序效率的降低,这显然是不合理和不经济的。

2、由于对每一次扫描都规定了必须要做 $(n-1)$ 次的比较,这样一来,使得在前次扫描中已排好了序的那些数据,在后面的各次扫描中仍要做无谓的重复比较,造成机时更大的浪费,排序效率再次被降低了。

从图1我们可以清楚地看出,数列经过第一次扫描(做 $n-1$ 次的比较)后,表中最后(下)一个数据“3”就排好了。显然,做第二次扫描时该数据就没有必要再参与比较了。换言之,做第二次扫描时不需要再做 $(n-1)$ 次的比较,而只要做 $(n-2)$ 次的比较就可以了。同理,做第三扫描时只需做 $(n-3)$ 次的比较,……,做第 $(n-1)$ 次扫描时,只做1次的比较即可。通过对图1数列排序过程的分析,我们可以总结出这样的规律:对任意数列的排序,每经过了一次扫描后,总会有一个新的逆序数据被排好序,而且无论降序或升序排序,首先被排好序的是表最底下的那个数据,随后是表次最底下的那个数据,最后才是表顶那个数据。这一规律清楚地表明,后一次的扫描比前一次的扫描可减少一次比较。但程序1未能解决这一问题,所以它的排序效率是低效的。

为了提高数据的排序效率,人们对程序1作了改进,改进后的程序如下:

程序2

```
100 B = 0
110 FOR J = 1 TO N-1
120 IF A (J) >= A (J+1) THEN 150
130 T = A (J): A (J) = A (J+1): A (J+1) = T
140 B = 1
150 NEXT J
160 IF B = 1 THEN 100
```

程序2的优点是针对程序1存在的第一个问题,将程序1的外循环计数控制扫描次数改为由条件来控制扫描次数。其基本思想是:当某一次扫描时,若发生数据交换(尽管只有一次),就说明表并未完全排好序,因此继续做下一次扫描。否则,说明表已完全排好,结束排序工作。这一改进,可使数列的扫描次数由数列本身的情况确定,而不再像程序1那样非要做完 $(n-1)$ 次扫描不可。由于程序2可减少某些数列(如图2中数列)的扫描次数,所以排序效率得到了提高。但程序2的缺点是,它只是解决了程序1存在的第一个问题,对程序1存在的第二个问题未有加以改进。因此,当数列为最坏情况(每次扫描都有发生数据交换,做完 $(n-1)$ 次扫描才能完成排序工作)时,由于扫描次数不能减少,这样,程序2和程序1的扫描次数相同,排序效率一样,仍是低效率的。

三、提高冒泡排序效率的有效方法

为了克服现有冒泡排序程序存在的低排序效率问题,真正实现冒泡排序的高效率。笔者在通过对这些程序存在低排序效率的原因作深入剖析的基础上,对程序2作了进一步的改进。改进后的程序如下:

程序3

```
100 I = 1
110 B = 0
120 FOR J = 1 TO N-I
130 IF A (J) >= A (J+1) THEN 160
140 T = A (J): A (J) = A (J+1): A (J+1) = T
```

150 B = 1

160 NEXT J

170 I = I + 1

180 IF B = 1 THEN 110

程序3与程序2比较,改动虽不大,但排序效率却相差很大。其根本原因是程序3能彻底地克服了程序1存在的不足,因而即使是在数列最坏(扫描次数无法减少)情况下,也能获得较高的排序效率。为检验改进方法的有效性,笔者在APPLE II微机上利用随机函数产生了10~100之间三组随机整数(每组100个数据)和100~1000之间两组随机整数(每组200个数据),然后将每组数据分别使用程序1、程序2和程序3进行排序。各程序对各数列所花的排序时间如表一、表二所示。

表一 排序效率比较 N = 100

排序时间 数列	程序1	程序2	程序3
第一组数据	2分28秒	2分14秒	1分28秒
第二组数据	2分29秒	2分15秒	1分26秒
第三组数据	2分33秒	2分18秒	1分32秒

表二 排序效率比较 N = 200

排序时间 数列	程序1	程序2	程序3
第一组数据	10分15秒	9分32秒	6分7秒
第二组数据	10分16秒	9分57秒	6分8秒

从以上两表可见,改进后的程序(程序3)大大地缩短了排序时间,使排序效率得到了显著的提高,而且数据量越大,越显示出它的优越性。

四、结束语

本文提出的对冒泡排序程序的改进方案具有如下优点:(1)对原冒泡程序改动小,但对提高排序效率十分有效。(2)对任意数列(即使数列为最坏情况),都具有较高的排序效率。(3)程序的设计思想对使用其他计算机语言编写的冒泡排序程序的改进同样适用。■

灯 谜

江苏泗阳县委机要室: 陈复军

谜面

1. 满园春色关不住 (打一计算机术语)
2. 一分为二 (打一计算机术语)
3. 天生一对, 地就一双 (打一电子方面名词)
4. 炸坝行洪 (打一邮电术语)

(谜底在本期找)

将 CCBIOS2.13H 的显示扩展模块 联接于其它汉字系统

山东潍坊检验所计算机室 李志刚

CCBIOS 2.13H 可以直接在屏幕上显示多种字体的 24 点阵彩色汉字，这给在应用程序中设计标题带来了很大方便。2.13H 的这个功能是由 INT10E.COM (或 INT10A.COM) 显示扩展模块实现的。笔者通过对该模块分析得知，其显示 24 点阵汉字的方法是通过调用 INT 78H 中断的 0CH 功能在屏幕上写点实现的，在程序中还用到了 INT 7EH 中断。这两个中断实际上是 ROM BIOS 的 INT 10H 和 INT 16H，2.13H 将其修改为 INT 78H 与 INT 7EH。因此只要加一个辅助程序就可以将该功能移植到其它汉字系统中。

这个程序所要做的工作是将 ROM BIOS 的 INT 10H 和 INT 16H 分别设置为 INT 78H 和 INT 7EH，以便联接 INT10E.COM，用来显示 24 点阵汉字，对于该汉字系统来说 INT 78H 和 INT 7EH 必须没另做它用。设置 INT 78H 和 INT 7EH 只要在进入汉字系统以前执行一下 INT78.COM 即可，该程序可用 DEBUG 来建立其过程如下：

```
C>DEBUG
-A100
4321: 0100 PUSH DS
4321: 0101 MOV AX, 3510
4321: 0104 INT 21
4321: 0106 MOV AX, ES
4321: 0108 MOV DS, AX
4321: 010A MOV DX, BX
4321: 010C MOV AX, 2578
4321: 010F INT 21
4321: 0111 MOV AX, 3516
4321: 0114 INT 21
4321: 0116 MOV AX, ES
4321: 0118 MOV DS, AX
4321: 011A MOV DX, BX
4321: 011C MOV AX, 257E
4321: 011F INT 21
4321: 0121 POP DS
4321: 0122 INT 20
4321: 0124
-N INT78.COM
-R CX
CX 0000
:24
-W
```

Writing 0024 bytes

```
-q
C>
```

但我们在使用中发现在某些汉字系统下，调用 ROM BIOS 的 INT 10H 不能正常工作，解决的方法是将 INT 78H、INT 7EH 分别设置成与该汉字系统的 INT 10H 和 INT 16H 相同的入口地址，这种方法适用于所有的汉字系统，但显示 24 点阵汉字的速度要比前一种慢一些。其设置方法是在进入汉字系统后执行 INT 78 命令。

设置好的 INT 78H 与 INT 7EH 后，要显示 24 点阵汉字还需要以下模块来支持：

HZK24F	24 点阵仿宋体字库
HZK24H	24 点阵黑体字库
HZK24K	24 点阵楷体字库
JZK24S	24 点宋体字库
HZK24T	24 点图形字符库
INT 10E.COM	EGA 显示功能扩展模块
INT 10A.COM	CGA 显示功能扩展模块
FILE 16B.COM	读 16 点阵字库模块
FILE 24A.COM	读 24 点阵字库模块
ZF24.COM	读 24 点阵字符模块

这几个文件都是 CCBIOS 2.13H 系统盘中提供的，可将其考入硬盘。进入汉字系统后可按以下顺序键入所有命令或加到启动汉字系统的 BAT 文件中。

C> INT78; 如用 ROM BIOS 的 INT 10H 与 INT 16H 应在进入汉字前执行该命令。

C> INT 10E; CGA 用 INT10A

C> FILE16B

C> FILE24A 1SFHK

C> ZF24 3

至此就可以显示 24 点阵汉字了，具体使用方法参见 2.13H 使用说明，另外以上文件启动后如再启动 2.13H 中的 PRT.COM 就可使用 2.13H 打印系统，来打印宋、仿宋、楷、黑四种字体多种字号的汉字。■

袖珍机用户的喜讯

本中心特向袖珍机用户推荐 SHARP PC-E500 袖珍计算机, PC-E500 可以替代 PC-1500 机。

PC-E500 机内存 32K, 使用 BASIC 语言, 最大可扩充到 96K, 可选配窄行打印机、515P 绘图机, 还可选配磁盘机。

PC-E500 主机有串行口, 可以直接与具有 RS-232C 串行口仪器、仪表及计算机联接通讯。

PC-E500 主机可以直接与各种红外测距仪连接, 当作电子手簿使用。本中心生产有 E500 机与日本托卜康全站经纬仪、与日本测距舍、与美国 MD81 等红外测距仪连接的电缆线及软件出售。

PC-E500 机具有文件管理系统, 用户可将内存划出一空区域, 作为文件区使用, 文件区可以存放多个程序和数据文件, 从文件区调进调出程序或数据既高速又准确, 可免去使用磁带机录制的麻烦。

PC-E500 主机单价 1550 元 / 台

串行口连接电缆 120 元 / 条

广州袖珍计算机技术服务中心

地址: 广州东风东路 745 号 邮码: 510080 电话: (020) -7751025-235 电挂: 5051

一种用于 51 单片机防止程序“跑飞” 的软件抗干扰办法

重庆航天机电设计院汽电室 刘志明

近年来, 单片机, 尤其是 51 系列单片机在我国各行各业的参数检测与实时控制中得到了广泛的运用。同其它工控微机一样, 单片机在工作过程中也常常会由于电压的投入与切除、瞬间短路以及电网串进来的干扰脉冲引起 CPU 的工作紊乱。CPU 的 PC 值发生跳变——即程序“跑飞”, 使程序进入非预定区域而无秩序运行。此时, 如不迅速采取措施, 就有可能引起灾难性后果。

笔者在 51 单片机的实际运用中, 采用了一种简单、有效的软件抗干扰法, 较好地解决了上述问题。该方法的实施很方便, 因为一般的单片机应用系统中, 工作程序都不很长! 数据区也有限。因此, 我们可以在有用的程序及数据区之外的所有内存单元中, 全部填入字节 02H, 这样, 一旦程序飞入这些区域, 也就是系统受干扰, 程序“跑飞”

时, CPU 便会自动执行一条无条件转移指令——LJMP0202H, 并在 0202H 这个地址上, 设置一条无条件转移指令 LJMP address, 跳转到系统正常工作所进行的循环中。此种软件抗干扰办法, 克服了一般的软、硬抗干扰办法即一旦 CPU 受干扰, 便复位 CPU, 使其 PC 值为“0”这一不足之处。因为有些系统有它自己特定的工作循环, 一旦受干扰后, 是不允许复位 CPU 跳转到“0000H”单元地址执行程序的。

上述软件抗干扰方法, 具有很大的实用价值, 也适用于其它种类的微处理器。在“多功能定量包装秤”的监控程序中应用了此种抗干扰办法, 通过现场运行表明系统干扰性能有明显提高, 效果良好。■

1991 年总目录

专论

保护知识产权的重大措施	(3-2)
计算机的法律保护	(3-2)
认真学习与运用计算机软件保护条例	(4-2)
软件开发中的版权保护—兼容软件取得版权的途径	(4-2)
“Look and Feel”的争论及其影响	
—计算机软件相似性引起的版权纠纷	(5-2)
国库业务电脑化的进程与思考	(6-2)

网络通讯

Novell 网络共享汉字的调用	(4)
APPLE 与 IBM PC 的单向串行通讯	(5)

电脑应用

GDMIS 信息管理网络	(1-4)
微机网在发射机状态监控中地线隔离的实现	(1-6)
利用电子计算机编译密码	(1-8)
如何在 NOVELL 网无盘工作站安装虚拟盘	(1-10)
微型计算机遥控遥测无人值守广播	
电视发射机系统	(2-2)
视野测试	(2-4)
微机数据处理工作的排序方法研究	(2-5)
利用计算机打印条形码程序	(3-5)
可限定时间的操作训练程序	(3-6)
选频激光器微机控制系统电路设计	(3-7)
汉字注音的计算机实现	(3-9)
一个塑料膜厚度微机检测控制系统分析	(4-5)
半人工变量计算机单形寻优	(4-8)
PASCAL 程序自动生成技术	(4-13)
直方图的显示与打印	(4-14)
ADC 直接与 PC 接口连接的电路	(4-16)
BASIC 语言动画技术应用实例	(5-5)
适用于大型应用软件的软件保密硬卡的研制	(5-8)

软件纵横

MS-DOS 彻底剖析 (九) IBMBIO 总体结构	(1-11)
MS-DOS 彻底剖析 (十) IBM BIO 模块数据结构详析 (上)	(2-11)
MS-DOS 彻底剖析 (十一) IBM BIO 模块数据	

结构详析 (下)	(3-11)
棒极了, DESK-LINK	(3-15)
弹出式和下拉式菜单程序包的设计	(3-16)
AR3240 打印机驱动程序	(4-17)
《苹果全能系统》最新版本功能简介	(4-19)
结构系统模块层次图	(4-20)
彻底剖析 (十二) BIO 常驻模块总体结构	(5-9)
西文 DOS 下的汉字彩色显示	(5-16)
Turbo 系统盘中实用程序 Mcalc 介绍	(5-18)
山外青山楼外楼——比 dBASE IV 更新的 Foxpro	(6-7)
MS-DOS 彻底剖析 (十三)	
MS-DOS 层间调用算法的实现	(6-10)

大学生之页

在 PC DOS 中同时打开二十个以上文件	(1-15)
隐含文件列目录的最简方法	(1-18)
用 DEBUG 恢复内存文件	(1-19)
为可执行文件设置口令	(2-16)
硬盘安全分区的实现	(3-17)
分页符号的消除二法	(3-19)
再谈加快按键连发频率	(3-19)
技巧三则	(4-22)
dBASE IV 的结构化查询语言 SQL 简介	(4-25)
虚盘的动态建立和撤销	(5-19)
热键激活式 TSR 编程的通用格式	(6-14)

使用与维修

浪潮 LC 0530-D 系统板故障检修一例	(1-20)
OKI-8320 打印机维修一例	(1-20)
ANSI.SYS 汉字滚屏乱符的解决办法	(1-21)
IBM PC / XT 及兼容机系统板维修一例	(1-21)
SENDON UPS-600 型维修一例	(1-34)
AR-2463 打印机故障检修一例	(2-18)
PC 机总线故障检测	(2-19)
硬盘不能启动的软维修方法	(2-20)
软盘 0 磁道损坏文件后的恢复	(2-22)
9P.EXE 中的错误及更正	(2-22)
对《电源维修一法》的几点看法	(2-23)
M-1724 打印机的电源故障排除与使用	(3-20)
注意事项	(3-20)
由电阻电容引起的微机故障	(3-21)
掉电保护及自动恢复实用电路和程序	(3-22)
中华学习机 CEC—I 驱动器维修二例	(4-27)

对《硬盘不能启动的软维修方法》	
一文的补充	(4-28)
速修微机显示器三例	(5-18)
采用 82C11 芯片并行打印机接口的维修 ...	(5-21)
一种方便修复故障盘数据的实用程序	(5-23)
打印机驱动程序的修改	(5-24)
改 GWBASIC 使其适用于	
CCBIOS2.13H	(5-25)
袖珍机新秀—SHARP PC-E500	(6-16)
硬盘的软维护	(6-18)
带电插拔打印机电缆造成机器故障的	
修复	(6-22)
CITIZEN CKP-5240 打印机维修一例	(6-23)

计算机辅助教学

计算机模拟化学中的酸碱滴定	(2-23)
计算机辅助教学课件设计	(3-24)
计算机考试评价分析	(4-29)
PC 机上试题库图形文件的压缩、	
还原和显示	(5-25)
以新观点来改进高校计算机教育	(6-24)

中华学习机

软化中华学习机使用汉字的“硬环境”	(1-22)
一种简便的进制转换方法	(2-24)
解拆 CEC-I 五笔字型一例	(2-25)
中华学习机图文显示及打印技巧	(3-26)
给 CEC-I 中华学习机增加一个磁盘机	
接口	(3-28)
汉字图形全屏幕绘图	(4-35)
电算 Fermat 大定理	(4-36)
CEC-I、APPLE 机磁盘螺旋轨的	
原理与制作	(5-31)
中华机配用 24 针汉字打印机的方法	(5-33)
CEC-I 中文状态下 CATALOG 的改进	
.....	(6-25)
保持在中文状态下操作	(6-26)

中学天地

操纵杆的键盘模拟	(1-25)
AAPPLE II PLUS 中断系统剖析及其在软件加密中的	
应用	(1-26)
又一个苹果机图形放大的机器码	
程序	(1-28)
用 LINE 语句在屏幕上作彩色图象	(1-29)
APPLE 经验二则	(2-25)
对电算《Waring》程序问题的修正	(2-26)
一个屏幕快速绕卷程序	(2-27)

新编 CHAIN 程序介绍	(3-29)
改进 BLOAD 命令	(3-29)
谈 BASIC 的特技清屏	(4-31)
LQ-1600K PRINTER 的压缩打印	(4-33)
近似计算与精确判断之间的矛盾	(4-34)
如何使九针打印机字迹更清晰	(5-29)
弹性碰撞程序	(5-29)
电脑剪贴画	(5-30)
APPLE 图形压缩系统	(6-26)
也谈快速随机读取 READ 语句	(6-27)
一个小发现—6502 的奇怪转移指令	(6-28)

竞赛辅导

一九九〇年广东省青少年计算机程序设计竞赛试题	
分析及其选解(上)	(2-44)
一九九〇年广东省青少年计算机程序设计竞赛试题	
分析及其选解(下)	(2-28)
递归算法	(3-30)
一九九一年广东省青少年“科特杯”信息学	
(计算机) 竞赛试题	(4-37)
炸弹与迷宫	(6-29)

病毒防治

谈谈计算机病毒的防范措施	(1-30)
IBM 系列微机数据维护技术研究	(1-31)
毛毛虫病毒的分析及排除方法	(2-30)
“428”病毒的分析、诊断及防治	(2-32)
一类感染.COM 文件病毒的消除方法	(2-34)
226 种计算机病毒介绍	(3-33)
用 DEBUG 解除所有主引导记录型病毒 ...	(4-38)
解剖电脑病毒的基本方法	(4-39)
旅游者病毒的分析及排除方法	(4-42)
利用逻辑零扇区的 I/O 参数表修复硬盘	
主引导扇区的分区参数	(5-35)
以毒攻毒—boot 区型病毒的症治	(5-36)
介绍强有力的反病毒软件包	
Turbo Anti-virus V6.80a	(5-39)
V2000 病毒的工作原理与清除	(6-33)
“侵略者”病毒分析及防治	(6-35)
一种精巧的病毒—世纪病毒	(6-37)
广大一号及其两个变种病毒	(6-38)

电脑用户

DEBUG 可恢复备份文件	(1-17)
EXE 文件与 COM 文件的比较	(1-35)
具有多级放大功能的屏幕图形	
拷贝程序的设计	(1-37)
一种简便的报表输出方法	(1-38)

通用多库统计汇总程序的设计	(1-39)
CC-DOS2.13H 使用 CEGA 卡上之硬盘库	(1-43)
对 CCDOS4.0 (即西山 DOS) 的修改与补充	(2-41)
程序倒运行的实现方法	(2-42)
汉化 Turbo Pascal 3.01A, Supercalc 3, Wordstar 显示行参数修改	(2-45)
TURBOPASCAL 程序中调用 TURBO 模块应注意的问题	(2-47)
电子系统交流电源的配置	(2-49)
BASIC 数据文件的压缩技术	(3-10)
小型机与微机数据通讯中的数据格式转换	(3-39)
一种有效的硬盘加密方法—口令码加密	(3-41)
CH 机上的一个键盘绘图程序	(3-42)
IBM PC BASIC 语言矢量字库及其应用	(3-44)
用中文 WS 进行高级打印	(3-45)
用 TYPE 命令直接分页打印 WORDSTAR 文本文件	(3-46)
《五笔字型》系统下汉字制表符的快速输入	(3-46)
Foxbase+ 中 Readkey 函数的妙用	(3-47)
数据库查询窗口上、下滚动屏幕显示的实现方法	(3-48)
内存控制块与 COMMAND 下移的实现	(3-49)
备份文件结构信息的查询	(3-50)
设立存取硬盘的用户权限	(4-44)
《WS》实现多页自动打印的方法	(4-46)
如何实现“FOXBASE+”共享	(4-47)
活动式磁盘保护	(4-47)
PC-DOS 磁盘引导扇区内容注释	(4-48)
有效地扩充软盘容量	(4-49)
另一种消除分页符号方法	(5-4)
用高级语言在 IBM-PC 机单显上作画	(5-41)
一个用 TuroBo C2.0 编写的文件删除实用程序	(5-42)
EGA 卡图形方式下屏幕的滚动和横向移动	(5-43)
解决在内存 640K 的 IBM-PC 机上不能运行 COBOL 的两种方法	(5-44)
四通 2401—微机数据库文书管理系统	(6-39)
M1724 打印机中 DIP 开关的设置对打印机的影响	(6-39)
一种通用的外部命令口令设置法	(6-41)
谈谈 DOS 与 XENIX 系统共享硬盘	(6-43)
提高昌泡排序效率的有效方法	(6-45)

将 CCBIOS 2.13H 的显示扩展模块联接于其他汉字系统	(6-47)
---------------------------------------	--------

游戏乐园

游戏图形图象的巧妙再利用	(2-36)
DIGGER 画面解析	(4-50)
幻方百字诀	(5-34)
牌洗几次最好?	(6-30)
单片机演奏乐曲	(6-31)

万花筒

音乐程序	(2-38)
将乐曲输给电脑	(3-32)

单板机与单片机

TP-180 单板机 256KEPROM 的扩展	(5-45)
一种用于 51 单片机防止程序“跑飞”的软件抗干扰方法	(6-48)

资料

1000VA 在线式 UPS 性能比较表	(5-46)
3kVA、5kVA 在线式 UPS 性能比较表	(5-47)

服务台

1991 年广东省部分县、市计算机应用软件程序员联合水平考试初级程序员级考试大纲	(1-46)
计算机爱好者软件库	(1-47)
软件库	(2-48)
软件库	(3-52)
软件库	(4-51)

1991 年总目录	(6-49)
-----------------	--------

广告索引

广州白云山电源设备厂 CWY 系列高抗干扰稳压电源	(1-封面、封二)
广州先锋电子经营部	(1-封底)
智通电子公司	(1-封三)
广州市科教电脑商场微机配置价格一览表	(1-1)
广东省教育服务公司供应陕西计算机厂中华学习机	(1-2)
PC-1500 机镍镉电池组大量供货	(1-2)
广州经济技术开发区夏港综合服务公司	(2-封面)
电脑杂志社广州经济技术开发区科技开发经营部开业	(2-封二)
广州白云山电源设备厂 CWY 系列高抗干扰稳压电源	(2-I)

PC1500 计算机新型 128K 电子记录模块开 始供应	(2-II)
广东省教育服务公司供应陕西计算机厂中华 学习机	(2-II)
广州市科教电脑商场微机配置及价格 一览表	(2-III)
当今国内最理想的高抗干扰稳压净化电源产 品	(2-封三)
南虹牌单色终端显示器	(2-封底)
广州白云山电源设备厂 CWY 系列高抗干扰稳压 电源	(3-封面、封二)
广州市科教电脑商场微机配置及价格 一览表	(3-I)
JANUS 永远使顾客满意	(3-II)
晨光智能仪器股份有限公司	(3-III)
高能计算机公司简介	(3-IV)
广州达奇电脑技术有限公司简介	(3-V)
华南计算机公司小型机厂	(3-VI)
广州市计算机公司天河技术服务部	(3-VII)
广东外贸电脑公司天河产业区经营部	(3-VIII)
新华电脑中心	(3-IX)
广州天河计算机公司	(3-X)
广州袖珍计算机技术服务中心简介	(3-XI)
学习英语的好帮手—您身边的好伙伴	(3-XII)
供应陕西计算机厂中华学习机	(3-XII)
电脑杂志社科技开发经营部	(3-封三)
华杰技术开发公司	(3-封三)
广州超级电脑配件有限公司	(3-封底)
广东省计算中心	(4-封面)
电脑杂志社科技开发经营部	(4-封二)
广州白云山电源设备厂 CWY 系列高抗干扰稳压 电源	(4-I)
广州科教电脑商场微机配置及价格 一览表	(4-II)
广州袖珍计算机技术服务中心	(4-III)
华杰技术开发公司	(4-III)
广州经济技术开发区夏港综合服务公司网络 工程部	(4-IV)
中国科学院广州电子技术研究所智通电子 技术公司	(4-52)
中外合资广州华源电子 有限公司	(4-封三、封底)
广州白云山电源设备厂 CWY 系列高抗干 稳压电源	(5-封面、封二)
广州科教电脑商场微机配置及价格 一览表	(5-I)
汕头教学仪器设备公司、汕头市教学 仪器厂	(5-II)

赛宝星河公司	(5-III)
肇庆自动化(广州)公司简介	(5-IV)
广州天河电子器材经营部	(5-V)
机箱电源总汇	(5-VI)
天翔九针打印机总汇	(5-VII)
智通电子技术公司	(5-VIII)
自然码广东总代理—电脑杂志社 ...	(5-48、封三)
思索电脑公司	(5-封底)
3M 数据记录产品中国特约经销	(6-封面)
长安中华学习机 CEC-E	(6-封底)
广州白云山电源设备厂 CWY 系列高抗干扰稳压 电源	(6-I)
电源机箱总汇	(6-II)
博森企业有限公司向各用户单位致以 新年问候	(6-III)
广州科教电脑商场进口微机配置及价格一览表	(6-IV)
广州市天河思索电脑公司简介	(6-28)
袖珍机用户的喜讯	(6-48)
电脑杂志社科技开发经营部	(6-封二)
自然码广东总代理——电脑杂志社	(6-封三)

简 中国计算机学会教育与培训专业 讯 委员会 1991 年年会在渝举行

中国计算机学会教育与培训专业委员会 1991 年年会于 10 月 28 日至 30 日在重庆市重庆大学隆重举行。

出席这次大会的有来自全国 24 个省、(市)、自治区计 150 位代表。会议期间专委会下属八个学组分别举行了学术座谈交流,同时进行了专委会的换届改选工作。经到会代表民主协商并一致通过,选出了以重庆大学校长吴中福教授为主任委员的新一届专委会班子,其中常委共 16 人,委员名额 91 人。

重庆市有关领导出席了开幕式并向大会召开表示祝贺。

大会期间,原专委会副主任委员、中国电子工业总公司中国电子科技开发公司副总经理王亚民同志向代表们做了题为“谈谈我国 90 年代计算机行业的发展和人才培养”的专题学术报告。

大会收到学术论文 150 余篇,经各学组择优推荐,其中 72 篇收入本年会议文集。

西南财经大学经济管理系副教授周启海对他的论文“计算机程序设计教育的新捷径”做了进一步阐述。他提出程序设计教育应由非结构化过渡到结构化,尔后则过渡为“同构化”,用“同构化”的教育可以使学习者掌握本质的和共同的东西而达到由此及彼,触类旁通。

这次会议并对以往为专委会工作做出开拓性贡献的老专委会部分成员颁发了荣誉证书以表彰他们的无私奉献精神。

(吴安栋)