

电脑 4 1991

中国软件行业协会会刊

創能

SUPERCAT

桌面排版輕印系統

SC999



SC 999 高級文字處理系統，能一機多用。既是高性能電腦打字機，又可作一般的計算機，具有圖文處理功能，實現圖文混排，達到圖文并茂的效果。打字功能與四通MS 2401 打字機兼容，但價格比四通MS 2401 便宜30%，該系統適用於辦公、人事、財務、信息和檔案管理等。

主要功能

編輯：調入文件、設置密碼、塊操作、尋找/替換、屏幕、模擬打印字符升高、分行分頁、右齊、左齊、多窗口、計數器、自動制表、其他。

輸入：拼音、雙拼、圖標置位、五筆字型、層次四角、表型碼、電報明碼。

字體：宋體、黑體、楷體、（繁簡）

字號：標準字號（0-7）、8×8至480×480點十萬種無級平滑縮放、長扁特大多種字型、自定義字型。

修飾：斜體、轉擊、虛體、空心、升高、角標、行居中、鋼點、網紙、加粗、反視、前景、背景等多種修飾。

文件：WS、MS-2401保持兼容，文件可相互轉換。

广东省计算中心技术服务部

地址：广州市连新路省科学馆内

电话：340642 邮码：510033 传真：340508

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，购配件时另加邮费 50 元 / 件）

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
ST-225 硬盘(40M)带卡	1850	ST-100M 硬盘带卡	3100
超级多功能卡(AT)	260	超级多功能卡(XT)	280
多功能卡(XT)	190	双频显示卡	180
单频显示卡	170	286-16 主机板(带 1MRAM)	940
调制解调器(300~2400 波特率)	2400	60M 磁带机	3850
3M DC600A 60M 磁带	235	双星 SS-CEGA 汉卡	1900
SUPER PUCII 型汉卡	1900	SUPER-CEGA 汉卡	1800
TVGA 显示卡	850	机械式鼠标器	480
光电式鼠标器	580	Novell 8 位星形网卡	750
Novell 8 位星形汉字网卡	1200	Novell 16 位星形网卡	1050
Novell 16 位星形汉字网卡	1950	Novell 8 位总线网卡	820
Novell 8 位总线汉字网卡	1300	Novell 16 位总线网卡	1184
Novell 16 位总线汉字网卡	2154	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

开户银行：广州农行德政北所 账号：247—612008

电话：330644 邮码：510055

Safe Protec 安全保護盤

本產品的國際註冊專利號：DBP 3603694

世界首創國際專利 **Safe protec** (安全保護盤) 為信息記錄提供了方便可靠和安全的使用方法，它高質量和卓越的性能是您的最佳選擇。

一. 優點：

- (1) 5. 25"軟盤片的數據寫保護採用了硬件開關，不需用膠紙粘貼，使用方便。
- (2) 硬件開關，不易脫落，避免了由于寫保護膠紙粘貼不牢，而損壞計算機的驅動器。
- (3) 硬件開關，裝配牢固，不易破壞已存入磁盤中的軟件程序。
- (4) 硬件開關，略加處理，即可防止計算機病毒的侵入和傳染。
- (5) 設計新穎採用了鮮艷的紅、綠兩色指示，標志醒目，不易出錯。
- (6) 與所有 5. 25"軟磁盤片完全兼容。

二. 使用方法：

右邊保護窗口的硬件開關稱“保護開關”，頂部的硬件開關稱“鎖緊開關”。

保護：不允許往盤上寫入信息。

按照紅色箭頭指示的方向和步驟操作。

- (1) 將保護開關向下撥，關閉寫保護缺口。
- (2) 將鎖緊開關向右推，鎖住保護開關。

保護開關和鎖緊開關均顯示紅色，表示此盤處于“寫保護”狀態。

開啓：允許往盤上寫入新的信息。

按照綠色箭頭指示方向和步驟操作。

- (1) 將鎖緊開關向左撥，打開鎖緊開關。
- (2) 將保護開關向上推，打開保護缺口，表示該盤處于讀寫正常狀態，允許往盤上寫入新的信息。

三. 質量：

採用優質，高檔原材料特制加工 70% 的漏碼門坎測試，終身質量保證。

中外合資廣州華源電子有限公司

地址：廣州市石牌華南師範大學微
電子所四樓。

營業電話：516911 轉 3040

郵政編碼：510631

電報掛號：4880 華源

世界首創·安全保護盤

safeprotect®

使用方便·安全可靠

包裝新穎·質量上乘



广州白云山电源设备厂

CWY系列高抗干扰稳压电源

我厂是生产稳压器、变压器、变压器铁芯的省电子局定点厂，具有十多年的生产历史，是我国生产各类电源设备及其配件的骨干企业。所生产的CWY系列高抗干扰交流参数稳压电源是我厂84年研制成功的国内首创稳压电源。通过国家技术部门鉴定。技术性能优越，比国内其他类型稳压电源有更明显的优越性。86年获国家科技进步奖。三相抗干扰稳压电源也由中国科学院广州分院通过技术鉴定。88年获产品专利权。89年获中国科学院技术进步三等奖。

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

购买时，请认明商标，提防有人剽窃本厂技术制造伪劣产品，使用户造成不必要的损失。

本产品已向中国人民保险公司办理了全国范围产品责任保险，用户可放心使用。

单相抗干扰稳压器系列

一、型号及规格

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

二、主要技术参数

●输入电压单相交流220V50Hz

●电压稳定度

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

- 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 ± 100 伏） $< 10 \sim 30\text{ms}$
- 有效功率可达到80~90%
- 音频噪声低，位于1.6米距离处50dB

全國範圍產品責任保險



三相抗干扰稳压器系列

一、规格：3KV, 6KV, 10KV, 15KV, 30KV

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

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

●音频噪声低：位于1.6米距离处52db。

●效率高：89%。比一般电子交流稳压器效率高1/5。这对节能很有意义。

本单相、三相电源已与VAX11/750, VAX11/785, PDP11/44连机试验。运转正常，性能良好，广泛适用于计算机、自动控制设备，电子显微镜，X光CT，核磁共振断层扫描仪等高、精、尖设备等。与国外同类产品性能相等。

厂长：贝远娥 副厂长、工程师：陈自如 厂址：广州市沙河同和 邮政编码：510515

开户银行：广州农行白云营业所 账号：80-431032 电挂：0839 电话：705764转23 705665转343

注册商标 质量三包 欢迎来人来函订购 多谢惠顾

高抗干擾
安全可靠
功能特殊
電腦必備

优质电脑请君选购

本公司自推出美国 AD 系列微机以后，其功能、质量均受用户好评，不论在银行、酒店、发电厂、船厂、化工厂、国家机关都有她的足迹。本公司将 AD 系列及其它型号微机的功能、最新价格列表如下：

型号	CPU	内存	硬盘	软驱	显示器	保修期	价格
AD-386 / 33	80386 / 33	4MRAM	100M	1.2M+1.44M	彩色 VGA 1024 × 768	二年	21500 元
AD-386 / 25	80386 / 25	2MRAM	100M	1.2M+360K	同上	二年	19700 元
AD-386 / 20	80386 / 20	2MRAM	100M	1.2M+360K	同上	二年	18500 元
AD-386 / 16	80386SX / 16	2MRAM	100M	同上	同上	二年	15500 元
AD-286 / 16	80286 / 16	1MRAM	40M	1.2M+360K	同上	一年	10500 元
AD-286 / 12	80286 / 12	1MRAM	40M	1.2M+360K	同上	一年	9600 元
AD-PC / XT	8088-2	640K		360K+360K	14 寸单显	一年	2950 元
AD-PC / XT	8088-2	640K		同上	12 寸单显	一年	2850 元
AST-386 / 33	80386 / 33	4MRAM	110M	1.2M+360K	彩色 VGA	一年	30500 元
AST-386 / 25	80386 / 25	2MRAM	110M	同上	同上	一年	27500 元
AST-386 / 20	80386SX / 20	2MRAM	40M	同上	同上	一年	18500 元
AST-386 / 16	80386SX / 16	2MRAM	40M	同上	同上	一年	17500 元

广州经济技术开发区
夏港综合服务公司网络工程部
地址：广州市天河路 33 号之七
电话：752397
邮编：510075

专论

- 认真学习与运用计算机软件保护条例 (2)
软件开发中的版权保护—兼容软件取得版权的途径 王桂海 (2)

电脑应用

- 一个塑料膜厚度微机检测控制系统
分析 徐巨善 (5)
半人工变量计算机单形寻优 胡泽滋 (8)
PASCAL 程序自动生成技术 符云清 (13)
直方图的显示与打印 陈文彬 (14)
ADC 直接与 PC 接口连接的电路 周仲玉 (16)

软件纵横

- AR3240 打印机驱动程序 王月成 (17)
《苹果全能系统》最新版本功能简介 钟华宁 (19)
结构系统模块层次图 万希宁等 (20)

大学生之页

- 技巧三则 钟卫等 (22)
dBASE IV 的结构化查询语言 SQL
简介 董祖明 (25)

使用与维修

- 中华学习机 CEC—I 驱动器维修
二例 陈华昌 (27)
对《硬盘不能启动的软维修方法》
一文的补充 刘益林 (28)

计算机辅助教学

- 计算机考试评价分析 张鲁闽等 (29)

中学天地

- 谈 BASIC 的特技清屏 侯 胜 (31)
LQ—1600K PRINTER 的压缩打印 任绥海 (33)
近似计算与精确判断之间的矛盾 廖庆平 (34)

中华学习机

- 汉字图形全屏幕绘图 黄文贵 (35)
电算 Fermat 大定理 陈君佐 (36)

竞赛辅导

- 一九九一年广东省青少年“科特杯”信息学
(计算机) 竞赛试题 (37)

病毒防治

- 用 DEBUG 解除所有主引导记录型
病毒 王全国 (38)

- 解剖电脑病毒的基本方法 熊虎 (39)
旅游者病毒的分析及排除方法 胡向东 (42)

电脑用户

- 设立存取硬盘的用户权限 杜建春等 (44)
《WS》实现多页自动打印的方法 陈复军 (46)
如何实现“FOXBASE+”共享 王培福 (47)
活动式磁盘保护 梁耀旋 (47)
PC-DOS 磁盘引导扇区内容注释 陈兴东 (48)
有效地扩充软盘容量 雷 军等 (49)

游戏乐园

- DIGGER 画面解析 宋令本 (50)

服务台

- 软件库 (51)

广告索引

- 广东省计算中心 封面
电脑杂志社科技开发经营部 封二
广州白云山电源设备厂 CWY 系列高抗干稳压
电源 I
广州科教电脑商场微机配置及价格一览表 II
广州袖珍计算机技术服务中心 III
华杰技术开发公司 III
广州经济技术开发区夏港综合服务公司网络
工程部 IV
中国科学院广州电子技术研究所智通电子
技术公司 (52)

- 中外合资广州华源电子有限公司 (封三)、(封底)

编 辑 出 版

《电脑》编辑部
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:516911-3273)

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

8 月 25 日

刊 号

ISSN 1002-9613
CN44-1188TP

继《中华人民共和国著作权法》及其实行条例生效之后,我国国务院又公布了《计算机软件保护条例》(今年10月1日起施行)。这是我国软件工作的又一重要法律依据。这个条例解释了计算机的定义;指明了软件著作权的内容,给予著作权人充分的受保护权利;提出了计算机软件的登记管理;软件侵权应负的责任等等,这些都说明我国对知识产权、计算机软件保护的重视。

我国广大的软件工作者,必须充分认识著作权法与计算机软件保护法条例对国家,单位以及个人的重大关系。如果说,过去在软件保护,包括开发、出版发行、装机、复制、使用、拷贝等方面,还不是那么明确地有法可依的话,那么,今天这种情况要结束了。现在,人们应当而且能够在比较完整的法律保护和法律监督之下开展与软件有关的工作了;靠软件能手与法盲于一身的不正常现象,也应该逐渐成为过去了。每个软件工作者都应当认真学习与运用我国政府给予我们的法律武器,使自己的软件在国内、国外都处于合法地位;同时,也要认真地依法抑制他人侵权。这些,都是我们应肩负的责任。可以预期,本条例将给我国计算机软件带来一个新的创作繁荣时代。■

软件开发中的版权保护

——兼容软件取得版权的途径

华南师大微电子所 王桂海

我们在本刊上一期就著作权法以及计算机软件的法律保护作了讨论。现在,趁《计算机软件保护条例》公布之际,进一步讨论兼容软件如何取得版权保护的问题。为什么要专门讨论这个问题呢?从计算机软件保护条例来看,对软件的保护是多方面的:开发过程、出版、软件装入、使用、合法拷贝的处置等方面,都受到特定的保护。兼容软件在开发过程中比较容易引起侵权纠纷。因为兼容软件与被兼容软件形式上往往很相似,可能被指控侵权。当然,指控并不一定成立,这就要看兼容软件的作者如何去为自己的作品取得法律保护了。为说明这个问题,我们首先花一些篇幅来讨论兼容软件的性质、特点。

(一) 兼容技术和兼容软件

兼容技术是为解决计算机软件的可移植性,使之适应更多新机种而发展起来的。一般称为 Program-compatible Mainframe (程序兼容主机) 或 Plug-compatible Manufactures (接插兼容生产商), 统称 PCM。当某一程序可以不加修改地在两台不同的机器上运行,产生同样的结果,则认为这两台机(对这一程序)是相互兼容的。当然,程序并不限于一个,而是某类程序的集合。所以,说得详细一点:

设有计算机 A (包括 A 的系列机,系列机内的兼容,不属本文讨论范围,下同), 对应有它的软件集合 P_A , 包括系统软件 Pa_1 , 应用软件 Pa_2 , 记为

$$P_A = \{Pa_1, Pa_2\} \quad (1)$$

$$Pa_1 = \{Pa_{11}, Pa_{12}, \dots, Pa_{1n}\} \quad (2)$$

$$Pa_2 = \{Pa_{21}, Pa_{22}, \dots, Pa_{2n}\} \quad (3)$$

(3) 最后的“...”表示应软件开发的延续性。

由于集合中的每一个软件都是以 A 机为基础开发的,属于 A 机,记为:

$$P_A = \{Pa | \forall e \in A\} \quad (4)$$

同样,对不属于 A 系列的计算机 B (包括 B 的系列机,下同),我们有:

$$P_B = \{Pb | \forall e \in B\} \quad (5)$$

现在,如果做到 P_A 中的一部分可以不加修改地在 B 机中运行,得到在 A 机中——对应运行时同样的结果;而且,在 P_B 中的一部分,可以不加修改地在 A 机中运行,得到在 B 机中——对应运行时同样的结果,则一定有交集 P_C 。以 P_C 表示既能在 A 机中,又能在 B 机中运行且得到同样结果的一类软件集合,有:

$$P_C = P_A \cap P_B = \{P_C | P_C \forall e \in A \text{ 且 } P_C \forall e \in B\} \quad (6)$$

具备这个性质,则称计算机 A 与计算机 B 相互兼容。在 PCM 技术中,若 A 机是原有的, B 机是以 A 机为目标而制造的,则称 A 为“目标机”,而 B 则是 A 的“兼容机”。其性质是向后兼容(已有的程序可以不加修改地运行于其后推出的机器)。

若 A 机的全部程序都可以在 B 机中运行,

$$\text{即 } P_A = \{Pa | \forall e \in A \text{ 且 } \forall e \in B\} \quad (7)$$

则称为 B 对 A 全兼容。这时, (6)、(7) 均成立, $P_A \subset P_B$, 且 $P_A \subseteq P_C$ 。若只有 (6) 成立,则是部分兼容。这里,有几点要说明:

1) 以上各表达式的归属关系,都仅限于技术方面,未涉及法律(知识产权)问题。

2) 兼容是基于扩大软件的适用范围,首先是用户的

要求,但同时也是为计算机有更广阔的市场,所以是生产商所推行的。

3) 从软件归属关系来看,向前兼容(即在后推出的B机上新开发的软件,要专门适合前面已有的A机)没有存在的意义。因为若某程序是面向A机的,即可归入 P_A ,没有必要称之为“在B机上开发的,向A机兼容的软件”。

4) 软件的兼容范围总是有限的。可以证明,当B机不完全等于A机(记为 $A \approx B$)时,不可能有全兼容($P_A \subseteq P_C$)。

设有一程序 $P_A' \subset P_A$,专门用于检查A机各个部件线路级以上的情况。这个程序,当然能且只能在A机中运行。若 $B \approx A$,则 P_A' 不可能不加修改在B中运行。式(7)不成立,故 $P_A \not\subseteq P_C$ 。所以,当且仅当 $B=A$ 才能完全兼容,否则,只能是部分兼容。由于兼容的概念是指两种不同的机器,所以,本文讨论的就是部分兼容引起的问题。

如何实现部分兼容呢?

从现行计算机技术来看,它的必要条件是B与A机在系统结构(Computer architecture)上一致

记为: $B \approx A$; 但 $B_{(ca)} = A_{(ca)}$

即B是与A相异的但在系统结构上一致的兼容机。

$B_{(ca)} = A_{(ca)}$ 是指下列几个方面一致:

1. 指令系统,包括指令的操作类型和格式,指令的控制机构和指令间的排序。
2. 数据类型和数据结构。
3. 寻址方式,包括最小寻址单元和地址运算。
4. 中断机构,包括中断码、中断地址分配。
5. 输入输出结构,包括对I/O设备的访问方式,I/O数据的“源”与所传送的数据量,I/O操作的结束与出错指示等。
6. CPU状态,包括处理机状态字,控制寄存器等。
7. 操作员环境,包括控制台面板、键盘以及显示器等。

以上各方面一致,则A、B两机可以实现高级语言、应用程序和操作系统等方面有相当程度的兼容。

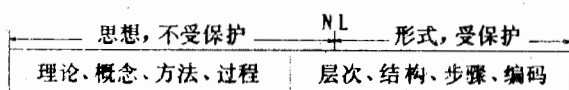
对于微机来说,要做到系统结构一致即 $B_{(ca)} = A_{(ca)}$ 则更直捷些,如果B机采用了和A机相同的处理器来作CPU,便自然实现了指令系统、数据类型与数据结构、输入输出接口、CPU状态、寻址方式等方面相同。如果采用厂商提供的标准接口片子作为I/O的硬件接口,再在BIOS(基本输入输出系统)这个层次上采取相同的中断编号和参数地址,则这两种机器将在很大程度上实现兼容。由此可见,兼容的实现,是以硬件上相接近为前提的。

由软件的可移植性要求,推动了兼容技术的发展,不但使用户(拥有现成软件)受益,更使兼容生产商取得巨额利润。当兼容是由同一系列机(同一生产商)来实现

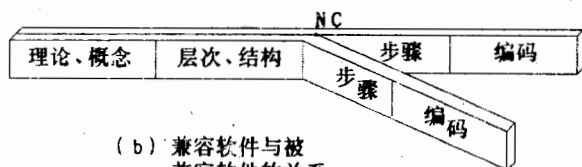
时,不会有权益的纠纷。问题是兼容生产商多是同行的竞争者。这些竞争者用不着花目标机的厂商那么多的力气,生产出兼容机,直接采用目标机已积累起来的丰富软件资源,使得兼容机在性能价格比上拥有明显的优势,夺去目标机的一部分市场。目标机的投资者自然会运用知识产权这一法律武器去维护自身的权益,遏制兼容产业的发展。

从兼容技术发展的二十多年情况来看,硬件实现兼容较易避免法律纠纷。因为兼容厂商可以采用第三方厂商生产的元部件,通过自己的设计组成性能与目标机相接近(即做到 $B_{(ca)} = A_{(ca)}$)的新机器,而又采用不同于目标机的商标,这样,便可能避免触犯专利法和商标法,较易取得合法地位。但仅靠硬件(裸机)仍不能立即实现兼容,要靠软件来作补充,尤其是与裸机相连接的操作系统(包括BIOS和其它方面的OS)。只有这些方面问题都解决了,才能使大量现行的应用软件直接在新机种上运行。由于许多国家已把软件列入著作权法的保护范围,兼容软件因与目标机软件形式上相雷同(版权法只保护形式不保护思想),已经在这两者之间,导致一系列的法律诉讼事件。美国各级法院,受理了一批这类案子。最近报导,欧洲共同体(EC)在拟议中的版权法准则正要大大限制兼容者对已有机器接口的利用,禁止制造兼容产品。但欧洲可互通系统委员会(ECIS)则持相反的看法,这方面的争论,正在欧洲展开。

关于软件的法律保护和兼容软件的关系,我们提出一套模型来描述(图1):



(a) 法律的尺度



(b) 兼容软件与被兼容软件的关系

图 1 兼容软件与法律尺度

图1a,是法律尺度的描述,著作权法的特点是保护作品的表现形式而不保护思想,而什么算是表现形式,不同的法律有着不尽相同的解释。图1a中的 N_L 点,表示某种著作权法提出的保护范围分界点;图1b是原软件与兼容软件的形式比较。兼容软件与原软件总有相同之处;除死复制外,也总有不同之处,相同到什么程度,由图1b的 N_C 点反映出来。根据它们之间的异同,确定出 N_C 点后,以图1a的法律模型去衡量,应当可以作出裁决。

但事实上,由于“思想”和“表现形式”这个概念在计算机软件中的解释,还有许多争议,法律的运用便有相当的弹性: N_L 点有时向左滑(对兼容软件来说是增加了获得法律地位的障碍),有时向右滑(兼容软件易于取得法律承认),这便给兼容软件开发者提供了发展余地。下面,我们将讨论兼容软件将如何为自己取得法律保护。

(二) 兼容软件的作者取得版权的途径

从版权法的讨论我们知道:独立创作;作品必须以某种客观形式表现出来;该作品能够不须作者参加而以某种方式复制,是版权作品的基本特点。对计算机软件来说,后两特点是显而易见的,不论是兼容软件还是被兼容的软件,都具有这些特点。在已确定后两点的前题下,独立创作才是问题真正的焦点。只要作品是“独立创作”的,便一定取得版权。但是,独立创作怎样证明呢?我们可以从下面模式去阐明(图2)。



作品享有版权的充要条件是独立创作,独立创作的充分条件是作品形式上独立或者是创作过程独立(行为独立)。

图 2

图2表示:一个作品,被承认是独立创作的话,则一定被承认享有版权,独立创作是取得版权必要的和充分的条件。独立创作又是怎样被承认的呢?作品在形式上独立,独创性便自然被承认,这是充分条件(但并不是必要条件)。除此之外,如果创作过程是独立的话,尽管形式上和已有的作品极为相似,也不妨碍这一作品获得版权。于是,“行为独立”便是独创性的又一个充分条件(当然,也不是必要条件)。因此,如果一个作品要取得版权,要么是作品形式上是独立的,不与其它作品相似,自然被承认是独创作品;要么能证明作者写作过程中,是独立创作而不是对已有的作品抄袭,这同样可以取得版权。显然,后一种途径是要经过法律证实的。

但正如前所述,兼容软件与被兼容软件总是有相似的地方。我们以兼容的操作系统为例,设目标机的操作系统为 $A_{(os)}$, 兼容的操作系统为 $B_{(os)}$ 。 $B_{(os)}$ 如果被 $A_{(os)}$ 的作者控诉,也因为是与 $A_{(os)}$ 相似而引起的。要使 $B_{(os)}$ 取得版权,一是要能充分说明 $B_{(os)}$ 并没有与 $A_{(os)}$ 相似到在法律上被认定为侵权的地步(即能说明 N_C 落在 N_L 的左边)。或者,不管相似到什么程度,但能证明作者在写作过程中是独立完成的。

对于因“相似”而引起纠纷的问题,我国计算机软件保

护法第三十一条规定:“因下列情况之一而引起的所开发的软件与已经存在的软件相似,不能构成对已经存在的软件的著作权的侵犯:

(一) 由于必须执行国家有关政策、法律、法规和规章;

(二) 由于必须执行国家技术标准;

(三) 由于可供选择的表现形式种类有限。”

应当说,即使不是由于本条所列举的三种情况而引起的相似,也不见得就一定构成侵权,只有相似到把 N_C 点推到 N_L 点的右边,才有可能被指控。所以,通过第一种途径来使 $B_{(os)}$ 取得版权的话,我们可以做下列的工作:

- 列举 $B_{(os)}$ 和 $A_{(os)}$ 的相异点(论证“并不完全相同”),相异点越多,越有利。

- 列举 $B_{(os)}$ 作出的“并非毫无意义的改进”(the non-immaterial variations),这样做的目的是要强调本程序的“创造性”成份。

- 列出有关政策、法律、法规和规章的限制(援引我国法律条例)。

- 列举 $B_{(os)}$ 中有关公有知识和标准化的应用。

- 指出思想与表达形式的“同一性”(the form embodied with the idea)。思想与表达形式当然是两个不同的概念,一般是能够分开的,这也正是版权保护的根据。但有时候,思想与表达形式是不可分的。既然思想是不受保护,那么,其不可分割的表达形式也无法被保护了。1978年,美国 Synercom 技术公司控诉 University 计算公司对其程序的输入格式侵权。被告对法院提出反问:如果不采用原告程序的命令与步骤(order and sequence,被认为是形式之一),能否取得原告中的思想呢?被告举出一个例子,说这种顺序有如汽车变速杆的“H”型结构那样,只有设计者作出的图纸本身,才具有版权,但“H”这种形式,和思想是分不开的,不应限制他人采用。法官在无法分开 Synercom 的程序输入格式的思想与形式之后,驳回了原告。再讲得通俗一点,我们可以举另一个例子,如要防积水,一般会采用“人”字型盖顶的结构,版权保护的只应是以“人”字型构成的某个建筑或其它具体的结构的图纸本身,但对“人”字的形式,因为和泻水原理是不可分的,所以,不应实行保护。这就是“思想-形式同一性”的原则。

- 指出硬件的限制和表达某种思想的唯一性。兼容软件是以兼容的硬件为基础的,硬件相似必然导致软件中的相似,这并不是软件工作者的抄袭行为。在某种情况下,由于限制太多,导致两个程序表现形式十分相似甚至这种形式是唯一的表达,这都应当是允许的。软件保护条例中提到“由于可供选用的表现形式种类有限”,在本文作者看来,“有限”这样的提法只有理解为“唯一的或在常识范围(而非诡辩)所指极少的”才是正确的。因为“有限”这个概念,有时可以理解为很大的数目(有兴趣的读者可以参看

一个塑料薄膜厚度微机检测控制系统分析

徐巨善

本文叙述英国引进的流涎薄膜生产线塑料薄膜厚度微机检测控制系统。重点分析系统数据采集、数据处理及控制各部分的结构原理,说明系统对塑料薄膜厚度的扫描检测和双闭环控制的工作过程。

1、概述

具有双挤出机的流涎薄膜生产线,用于生产平面塑料薄膜,其结构如图1所示。图中挤出机A、B将熔融后的塑料,通过机头的模口流出,成形的塑料薄膜经主冷辊和次冷辊冷却,在导向辊引导下,通过扫描器进行检测,处理辊进行电火花处理,扩张辊1、2平整,再由拉辊等进行张力控制和测量,最后由卷取机成卷后装箱。

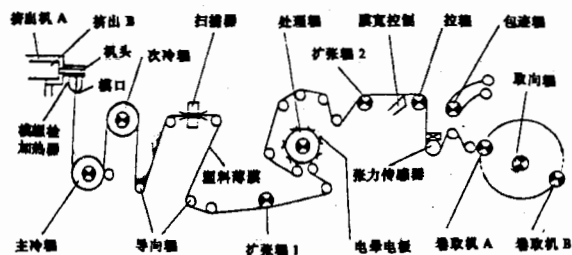


图1 流涎薄膜生产线结构简图

薄膜厚度微机检测控制系统是塑料薄膜生产线质量检测、显示和控制的关键设备,对于保证产品质量、降低生

著名的计算机科学家 D·Knuth 的 Mathematic and Computer Science Coping With Finiteness 一文),而且,只要硬件设备确定,存储容量确定,不论容量多大,总可以认为是“有限”的,如果这样去理解,这个“有限”实在是“太宽容”了。

以上这些,都是从“不相似”即“形式上独立”来表明 $B_{(OS)}$ 应享有版权。但如果这些方面,形式上独立仍不够充分的话,便要求助于“行为独立”的途径了。

$B_{(OS)}$ 作者与 $A_{(OS)}$ 的“无接触”,便是“行为独立”最好证明。但要证明 $B_{(OS)}$ 作者与 $A_{(OS)}$ “无接触”,则要有一番手续,要求符合一定的法律规定。这便是所谓“Cleanroom”(隔离室)的做法。这种做法简单来说便是:由与 $A_{(OS)}$ 有接触的人,充分消化了 $A_{(OS)}$ 的目的、要求、分析问题的方法、算法等(以上都是不受版权保护的),把它们变为一份额对软件 $B_{(OS)}$ 要求的任务说明书下达给另外一部分与 $A_{(OS)}$ 从未接触过的人。这时,由于对 $A_{(OS)}$ 的了解,仍然停留在思想阶段,所以不会有软件开发中的侵权问题,接受任务的人,也就是 $B_{(OS)}$

产成本,提高经济效益具有重大的作用。其主要功能是对塑料薄膜的厚度进行检测与控制,实现双闭环回路控制。由 β 源扫描器作测厚传感器,将薄膜横截面各处的厚度分别与设定厚度进行比较,并分别调整对应于横截面各处厚度的偏差,使其成为设定值;又根据薄膜横截面各处厚度的平均值调整生产线的工作,使其平均厚度成为设定值,从而保证薄膜厚度均匀。另外,测定并监视生产线的各种参数,除产品的厚度、长度、卷数外,还有各流涎辊的速度、温度、挤出机的电流、速度和压力,模螺栓加热功率等,保证生产线正常运行。

2、系统的结构原理

本检测控制系统由单板计算机系统、扫描检测控制电路和模螺栓加热控制电路等部分组成,如图2所示。

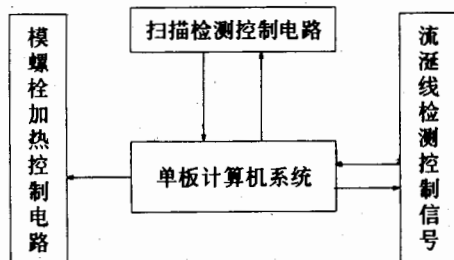


图2 微机检测控制系统框图

2.1 扫描检测控制电路

扫描检测控制电路主要用来控制扫描器沿2.4m宽的

的实际开发者,应仅限于在此基础上进行具体软件的制作,包括模块设计、流程图、具体程序的编写等。在这期间内, $B_{(OS)}$ 的作者应受到隔离,即不能接触到 $A_{(OS)}$ 的任何具体材料。如果 $B_{(OS)}$ 作者对 $A_{(OS)}$ 的有关思想、概念不清楚而要提问的话,信息的传递要经过专门的机构审查。而且,最后要能充分说明(1)所提供的信息足以成为 $B_{(OS)}$ 完成任务的基础(编出 $B_{(OS)}$ 不须靠其它渠道);(2) $B_{(OS)}$ 的作者有能力完成这个任务;(3)时间和财力上有充分保证, $B_{(OS)}$ 任务的被完成是符合情理的。这些资质对说明作者享有版权都很重要,否则,好比希望别人相信薛蟠是“葬花词”的作者那样,是一定要落空的。

侵害他人著作权,我们视之为剥削别人劳动成果;在英语中,抄袭(pirate)原意是海盗行为。可见,不论是国内还是国外,抄袭都受到鄙视,是非法的。我们既不允许别人对自己作品侵权,也不应在开发软件过程中搞抄袭,即使是开发兼容软件,也要严格防止这一点。(参考文献从略)

薄膜, 横向来回移动, 并在不同位置 (90 个采样点) 测定薄膜的厚度, 即对 90 个不同扫描位置的薄膜厚度进行测试, 取得薄膜厚度及其对应位置的采样信号。

扫描检测控制电路由扫描控制器、扫描器端箱和扫描器马达、扫描器 (β 源测厚传感器)、扫描器位置传感器、 β 源开和 β 源闭指示灯及薄膜在线检测器、长度传感器等组成, 如图 3 所示。

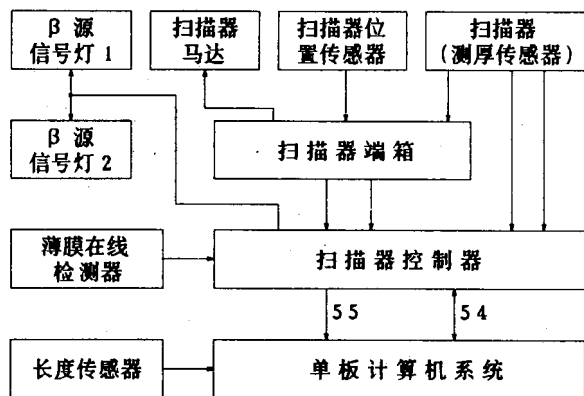


图 3 扫描检测控制电路框图

塑料薄膜生产线作业时, 单板计算机系统通过数据信号电缆 54, 对扫描控制器进行程序控制, 通过模拟信号电缆 55 对扫描检测控制电路提供直流电源, 并读取扫描器及其位置传感器的采样信号。扫描器控制器中包含启动按钮、逻辑控制继电器, 其逻辑信号控制扫描器端箱中的马达控制继电器, 使马达实现正转、反转及快速反向转动, 由此实现扫描器正向、反向及快速反向移动。在正向、反向移动时, 对薄膜进行采样。扫描器的薄膜厚度采样信号及其对应位置采样信号, 同时送单板机进行数据处理。 β 源开 β 源闭指示灯指出 β 放射源快门开关的开闭状态, 且只有在正反向扫描移动时 β 源开指示灯亮。薄膜在线检测器控制扫描器采样只能薄膜在线 (即薄膜通过扫描器) 时才能进行。长度传感器检测薄膜的长度。

扫描检测控制电路中还具有诊断装置, 利用诊断装置可以检查各种直流电压值, 模拟扫描检测薄膜及其对应位置的采样信号, 由此检查并排除故障, 确保电路正常工作。

2.2 单板计算机系统

单板计算机系统是检测控制系统中的专用设备, 由单板计算机箱、显示器、键盘和打印机等组成。显示器有彩色和单色两种, 可同时使用, 以便在不同位置观察生产线的工作情况。系统只配工作键盘, 以专门命令读取数据。在系统软件中读取设置生产线各工作参数的系统设置页, 在用户软件中读取设置产品参数的作业设置页。打印机用来打印系统设置页和作业设置页。操作系统是开机时从软盘读入内存的。

单板计算机箱由单板机主板、接口板、电源板及 5 寸软盘驱动器等组成。接口板主要是直流输入、直流输出及交流输出光电耦合电路。电源板提供 $\pm 15V$ 、 $+5V$ 、 $+24V$ 及 $\sim 24V$ 电压。单板机主板是微机系统的主体部分, 其结构如图 4 所示。

中央控制器和时钟电路主要由 CPU 芯片 CP2171B-15、CP2171B-14、CP2171B-18, 地址数据总线接口电路 CP2151B-05 和控制总线接口电路 CP2161B-02 等组成。CP2151B-05 与 16 位地址锁存器 LS534 和 16 位双向数据缓冲器 LS640 相连, CP2161B-02 与控制总线缓冲器 LS240 和锁存器 LS374 等电路相连, 输出各种地址、数据及控制信号, 并读取状态信息。至于时钟电路由 20MHz 晶体振荡器经 8 分频及移相后, 经 TTL-MOS 接口驱动到 CPU 总线电路, 并用作系统各种定时信号。

动态存储器电路主要由动态存储器控制器 D8203、DRAM 芯片 TMS6164 及输出锁存器 LS373 等组成。DRAM 电路的寻址空间为 0000-FFFFH, CPU 或 DMA 控制器对 DRAM 提供地址和读写控制信号。在 20MHz 时钟同步下, D8203 将高低各 8 位地址分别送 DRAM 地址输入端, 并在行列选通和读写信号控制下, 对 DRAM 存取数据。

扩充存储器主要由存贮系统参数的 E^2PR OM D2816A、用户使用的静态存储器 D4016-C 和显示存储器 D4016-C 等组成, 扩充存储器及 I/O 端口由标志寄存器所控制, 而标志寄存器由 DRAM 和 I/O 公用的一个地址所写入。扩充存储器及 I/O 端口地址由多片 PROM 及其他译码电路所组成。 E^2PR OM 还专门配有写入电路。

软盘控制器 FD1791 和软盘驱动器的控制信号与一般软盘控制器类似, 其数据经缓冲锁存器与 FD1791 接口。存盘时, 数据在 FD1791 中进行串一并转换, 再在存取逻辑电路中进行预补偿处理, 然后存盘; 读盘时, 调制数据在压控振荡

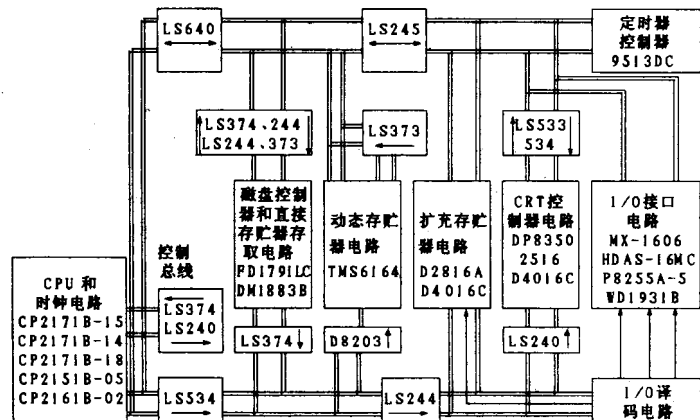


图 4 单板机主板结构框图

器和相位比较器控制下送入 FD1791, 经数据分离和串-并转换, 再写入存贮单元。

DMA 控制器电路主要由 DMA 控制器 DM1883B 和控制逻辑所组成, 用于 DRAM 和软盘驱动器之间数据传送。首先由 FD1791 发出 DMA 请求信号到 DM1883B 的 DMA 请求端, 然后由 DM1883B 向 CPU 发出总线请求信号。CPU 向 DM1883B 发出总线响应信号时让出总线控制权。在 DM1883B 的地址、存贮器读写及 I/O 读写信号的控制下, 实现 DRAM 与软盘之间的数据传送。传送结束时将总线控制权交还 CPU。

CRT 控制器电路主要由 CRT 控制器 DP8350、显示存贮器 D4016C、字符和字符属性发生器 2516 及缓冲锁存器等组成。显示时, 在 DP8350 控制下, 显示内容从显示存贮器读出, 其中字符代码送字符发生器产生字符点阵, 由移位寄存器串行输出; 字符属性代码送字符属性发生器, 产生字符点阵的前景和背景彩色信号。在多路转换电路控制下送出单色、红绿蓝及同步信号, 分别由单色和彩色显示器显示。CRT 控制器电路实际产生字符点阵为 7×10 , 字符大小为 5×7 , 显示屏幕为 80 列 \times 24 行。

串行接口电路有四个, 主要使用异步接收发送器 WD1931 和 MC1488、MC1489 等组成的 RS232 接口, 需要时可选用光电隔离器件。用于系统打印机、键盘和模螺栓控制器等。

并行接口电路有三个, 由 P8255A-5 及双向缓冲器等组成, 可接收或发送 72 个位信号。主要用于扫描控制电路数字信息的输入输出, 以及发送生产线主冷辊直流电机的控制信号。

模拟接口电路主要由 16 通道输入模拟多路器 MX-1606 和 16 通道输入、12 位数据输出的模拟接收系统芯片 HDAS-16 等组成, 是一个 32 通道输入、12 位数据输出的接收系统。主要用来接收生产线各工作参数送 CRT 显示。

2.3 模螺栓加热控制电路

单板机接收扫描检测控制电路对薄膜厚度及对应位置的采样数据, 并进行数据处理。模螺栓加热控制电路接收单板机处理后的数据, 控制对应位置模螺栓的加热功率, 使薄膜厚度调整为设定值。

模螺栓加热控制电路的结构如图 5 所示。主要由模螺栓控制器、通道控制继电器及模螺栓加热器三部分, 其中模螺栓控制器包括数据接收、存贮、比较电路, 移位寄存器和输出电路, 控制和时钟电路等。

单板机将薄膜厚度数据处理后, 通过 RS232 接口送到模螺栓控制器。在接收信号控制下, 数据接收电路将数据存入存贮电路。当 64 位数据存入存贮器时, 控制电路将存贮器各单元的数据分别与比较电路中的计数器的计数值 0 进行比较, 将比较结果送 64 位移位寄存器。由于存贮器 64 位数据均大于比较计数器值 0, 故比较电路输出 64 个 0 电平。在控制电路控制下并行输出, 经输出缓冲器分别控

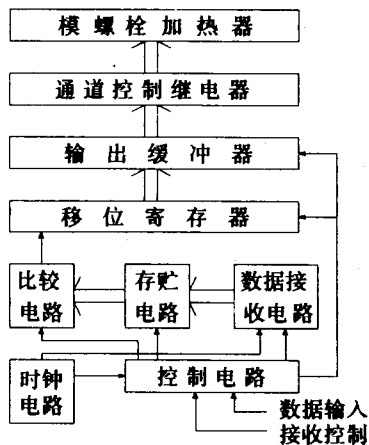


图 5 模螺栓加热控制电路框图

制 64 个通道继电器, 使 $\sim 220V$ 电源对模螺栓加热器进行加热供电。这就使塑料挤出机机头的模口间隙减小, 使塑料薄膜的厚度变薄。随后, 比较计数器的值加 1, 又分别与存贮器 64 个数据进行比较, 其结果又去控制模螺栓加热器。对移位寄存器输出 0 电平的位所对应的模螺栓提供加热功率, 而移位寄存器输出 1 电平的位所对应的模螺栓不加热。接着比较计数器的值又加 1, 再比较、送出加热控制信号。如此下去, 直到比较计数器的值为 FFH。然后, 比较计数器又从 0 开始重新进行比较, 不断循环。

由此可见, 在比较过程中, 存贮器 64 位数据是保持不变的, 而比较计数器的值由 0 到 FFH 循环变化。假如存贮器 64 位值均为 80H, 则比较计数器的值 $< 80H$ 时, 存贮器的值相对较大, 比较后输入移位寄存器的都为 0 电平; 当比较计数器的值 $> 80H$ 时, 存贮器的值相对较小, 比较后输入移位寄存器的均为 1 电平。因此, 在比较计数器之值 $< 80H$ 期间模螺栓得到加热功率, 而 $> 80H$ 期间不对模螺栓加热。这样, 就有 50% 时间对模螺栓提供加热功率。

假如由于环境温度等影响使薄膜厚度大于设定值, 则扫描检测处理后存入存贮器的数据就稍大于 80H, 对应的模螺栓加热时间就会略加延长, 使这处薄膜厚度恢复为设定值; 假如薄膜厚度小于设定值, 则扫描检测处理后存入存贮器的数据就稍小于 80H, 对应模螺栓的加热时间就会略有缩短, 也使该处薄膜厚度恢复为设定值。这样, 就实现了薄膜厚度的自动调整, 使其成为设定值。当然自动调整是在人工调整后使模口间隙, 也就是使薄膜厚度偏差较小的情况下才能实现。

3. 结束语

本检测控制系统是塑料薄膜厚度进行双闭环检测控制系统, 其过程如图 6 所示。

本系统是一个完整的微机工业检测控制系统, 所采用的数据采集、处理及反馈控制等都是 80 年代的先进技术。

半人工变量计算机单形寻优

贵州工学院机械系 胡泽滋

提要: 本文主要研究线性规划中引入松弛变量单纯形优化的计算机解法。文中重点地分析, 论述了这种寻优的原理、方法、步骤; 详尽地给出了比较完整又具有通用性的优化程序。此外, 还通过一个实例说明了该程序的具体应用和注意事项, 并给出优化结果所需的数据与参数。

本优化程序是在 AST-286 机上通过的。

主题词: 半人工变量

单纯形法寻优

线性规划的单纯形法, 实质上就是按顺序从一个基本可行解到另一个基本可行解一步一步地前进的过程, 每一步都使在一定约束条件下的最小值减小; 经过有限步之后, 达到再也不能改进的时候, 这一过程就停止。亦即当约束矩阵经过有限次转换后无处转换时, 寻优过程就此结束。这是一个非常精巧而极有效的数值方法, 即使含有上千甚至更多的约束条件, 用现代微型数字计算机也能在几分钟之内求出解答。虽然线性规划主要用于运筹学以及将其定量方法用于社会经济问题; 但在一定情况下它也是解决工程动态控制问题的很有效的方法。

HO 与 Noton 提出过用线性规划来研究加拿大国民经济管理的报告, 他们用的是线性经济模型。虽然原来的模型有 11 个状态方程。由于对于每个小的时间间隔都要重新确定各个变量, 又由于要引入其他的约束不等式, 约束总数竟达 130 个之多。可见用一般的线性规划解动态问题是很麻烦的。但是, 由于单纯形法很有效, 加上微型数字计算机的高效和快速计算功能, 解上述问题只需 12 秒钟; 因此, 这种方法还是普遍受到人们注意的。

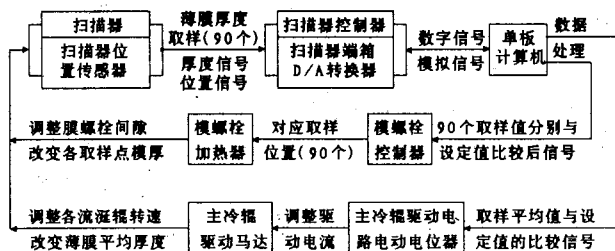


图6 薄膜厚度闭环检测控制原理图

不仅可广泛用于各种塑料、纸张等平面薄膜产品的检测和控制在, 而且对我国自行研究设计同类产品具有良好的鉴借作用。

本文是在解剖分析广州塑料二厂引进的流涎薄膜生产线基础上写成的, 在此对工厂领导、师傅的支持和帮助表示感谢。

一、原理与方法

所谓单纯形法, 就是线性规划的每一个可行解是由一组基所组成的, 从最初的一组基开始, 通过将这组基线性相关的其他变量进行替换, 从而产生一组新基, 这组新基导致得到一个新的较好的目标函数可行解。这样反复进行下去, 直到求得最优解为止。

有时, 根据已知的约束条件, 找不出明显的基, 这时需人为地引进松弛变量, 亦即人工变量, 以便得出最初的一组基。为了缩短计算过程, 我们只对部分约束条件加入人工变量, 而对另一些条件则不加入人工变量, 这就是半人工变量。

本程序运用上述方法来求解线性规划问题, 主要归结为解决两个问题。其一是: 有了一组基以后, 在有可行解的情况下, 如何选择一个与这组线性相关的变量进行替换, 以最快的速度, 趋向于最优解。同时, 对其他变量也要作相应的变换; 其二是如何确定目标函数是否已达到最优解(极大值或极小值), 以及如何区别线性规划不可行及无界。

设给定目标函数为:

$$f(x) = c_1 x_1 + c_2 x_2 + \dots + c_n x_n = cx = \max \quad (1)$$

给定约束条件为:

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n \leq b_1 \\ \dots \dots \\ a_{m1}x_1 + a_{m2}x_2 + \dots + a_{mn}x_n \leq b_m \end{cases} \quad (2)$$

且 $x_i \geq 0$

若要把以上约束条件写成等式, 即如下:

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n + a_{1n+1}x_{n+1} = b_1 \\ \dots \dots \\ a_{m1}x_1 + a_{m2}x_2 + \dots + a_{mn}x_n + a_{mn+m}x_{n+m} = b_m \end{cases} \quad (3)$$

其中 $x_{n+1}, x_{n+2}, \dots, x_{n+m}$ 即引入的人工变量。

(3)式可写成如下形式

$$p_1 x_1 + p_2 x_2 + \dots + p_n x_n + p_{n+1} x_{n+1} + \dots + p_{n+m} x_{n+m} = p_0$$

其中:

$$p_i = \begin{bmatrix} a_{1i} \\ a_{2i} \\ \vdots \\ a_{mi} \end{bmatrix} \quad \text{其中 } i < n$$

$$p_{n+j} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \quad \text{其中 } j \leq m \text{ (这时令(3)式中的 } a_{n+j} = 1)$$

$$p_0 = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{bmatrix}$$

对于(3)式存在一组基称为 B , 则令:

$$B = [p(1)p(2)p(3)\cdots p(m)] \quad p(i) \neq p_j$$

那么(3)式中的任一个向量 P_k , $k=1, 2, 3\cdots n+m$

都可由 B 来线性表出, 即

$$P_k = Y(1)_k p(1) + Y(2)_k p(2) + \cdots + Y(m)_k p(m) \\ = BY_k$$

其中

$$Y_k = \begin{bmatrix} Y(1)_k \\ Y(2)_k \\ \vdots \\ Y(m)_k \end{bmatrix} \quad K=1, 2, 3\cdots, n+m \quad (4)$$

(4)式又可以写成:

$$Y_k = B^{-1}P_k \quad k=1, 2, 3, \cdots, n+m \quad (5)$$

由线性变换, (3)式可写成

$$\theta(1)P(1) + \theta(2)P(2) + \cdots + \theta(m)P(m) = P_0 \quad (6)$$

其中 $P(i)$ 为新基 B 的各个分量, 而令

$\theta(i) = X(i) \quad i=1, 2, 3, \cdots, m$ 是变量向量在某组基下形成的各分量。

下面我们来推导如何找出替换变量 θ'_k , 产生一个新基。

在(6)式两边分别加上 $\theta'_k P_k$ 项得:

$$\theta(1)P(1) + \theta(2)P(2) + \cdots + \theta(m)P(m) + \theta'_k P_k \\ = P_0 + \theta'_k P_k$$

将 $P_k = Y(1)_k p(1) + Y(2)_k p(2) + \cdots + Y(m)_k p(m)$ 代入上式整理后得:

$$P_0 = (\theta(1) - \theta'_k Y(1)_k)P(1) + (\theta(2) - \theta'_k Y(2)_k)P(2) \\ + \cdots + (\theta(m) - \theta'_k Y(m)_k)P(m) + \theta'_k P_k \quad (7)$$

在(7)式中, 右边有 $m+1$ 项分量, 而初始基 B 只有 m 项分量, 所以有一项应该消去, 这样可以获得一组新基。我们选择某一项趋近零最快的进行变换, 即取

$$\theta'_k = \frac{\theta(1)}{Y(1)_k} = \min \left\{ \frac{\theta(i)}{Y(i)_k} \right\} \quad Y(i)_k > 0 \quad i=1, 2, \cdots, m \quad (8)$$

若将第1个分量消去, 其他分量也应进行相应的变换, 其变换公式为:

$$\theta'(i) = \theta(i) - \frac{\theta(1)}{Y(1)_k} Y(i)_k, \quad i=1, 2, \cdots, m \quad (9)$$

这样初始基 B 中的 $P(1)$ 分量将由 P_k 来替换, B 中的其他分量也相应有所变化, 从而得到一个新基。

现在再来看看, 新基的变换, 对目标函数有什么影响:

在初始基下, 目标函数表示为:

$$f(x) = C(1)\theta(1) + C(2)\theta(2) + \cdots + C(m)\theta(m)$$

若列向量 P_i 是一个经过替换后新基的一个分量, 则由(7)式可得:

$$\theta'(1) = \theta(1) - \theta'_i Y(1)_i, \quad \theta'(2) = \theta(2) - \theta'_i Y(2)_i, \quad \cdots$$

而在新基下的目标函数为:

$$f'(x) = C(1)(\theta(1) - \theta'_i Y(1)_i) + C(2)(\theta(2) - \theta'_i Y(2)_i) \\ + \cdots + C(m)(\theta(m) - \theta'_i Y(m)_i) + C_i \theta'_i \\ = C(1)\theta(1) + C(2)\theta(2) + \cdots + C(m)\theta(m) \\ - C(1)Y(1)_i \theta'_i - C(2)Y(2)_i \theta'_i - \cdots \\ - C(m)Y(m)_i \theta'_i + C_i \theta'_i \\ = f(x) - \theta'_i (C(1)Y(1)_i + C(2)Y(2)_i + \cdots \\ + C(m)Y(m)_i - C_i)$$

令 $Z_i = C(1)Y(1)_i + C(2)Y(2)_i + \cdots + C(m)Y(m)_i$, 则

上式化为:

$$f'(x) = f(x) - \theta'_i (Z_i - C_i) \quad (10)$$

对于(10)式来说, 其中 $\theta'_i \geq 0$ 。在求极大值的情况下, $Z_i - C_i$ 必须为负值时, P_i 才可以作为一个新基的分量而引入, 否则 $Z_i - C_i$ 为正值, P_i 就不可以引入。因为引入 P_i 的目的就是使 $f'(x)$ 取值增大而趋于极大值。若所有引入的人工变量都为零, 则对应的基底就是最优解, 即使 $Z_i - C_i$ 无负值为止, 此时目标函数取极大值。

同样, 在求极小值的情况下, 要求 $Z_i - C_i$ 必须取正值。当变换 $Z_i - C_i$ 都不为正时, 目标函数取极小值。

由(8)式和(9)式知道, 基的变换是在 $Y(i)_k > 0$ 的情况下进行的。若 $Y(i)_k < 0$, 则变换的 $\theta(i)$ 值将越来越大, 最终使目标函数趋于无穷。若 $Y(i)_k = 0$, 则会使 θ'_k 取任意值, 引出无穷多组解, 即线性规划问题不可行。

半人工变量单纯形法就是在出现约束条件“ \geq ”符号时, 引入人工变量, 而出现“ \leq ”或“ $=$ ”的约束条件时, 则不引入人工变量。

在一般情况下的约束条件为:

$$\begin{cases} a_{11} x_1 + a_{12} x_2 + a_{13} x_3 + \dots + a_{1n} x_n \leq b_1 \\ a_{21} x_1 + a_{22} x_2 + a_{23} x_3 + \dots + a_{2n} x_n \leq b_2 \\ \dots \quad \dots \quad \dots \quad \dots \\ a_{i1} x_1 + a_{i2} x_2 + a_{i3} x_3 + \dots + a_{in} x_n \leq b_i \\ a_{i+11} x_1 + a_{i+12} x_2 + a_{i+13} x_3 + \dots + a_{i+1n} x_n \geq b_{i+1} \\ \dots \quad \dots \quad \dots \quad \dots \\ a_{m1} x_1 + a_{m2} x_2 + a_{m3} x_3 + \dots + a_{mn} x_n \geq b_m \end{cases}$$

其中 $1, 2, \dots, i$ 项是含有 \leq 符号的约束条件; 而自 i 到 m 项约束条件是含有 \geq 符号。当把约束条件化为等式处理时, 则 i 项以前各个约束条件要引入一个带正号的人工变量而 i 项以后各个约束条件要引入一个带负号的人工变量, 即为:

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n + x_{n+1} = b_1 \\ a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n + x_{n+2} = b_2 \\ \dots \quad \dots \quad \dots \\ a_{i1}x_1 + a_{i2}x_2 + \dots + a_{in}x_n + x_{n+i} = b_i \\ a_{i+11}x_1 + a_{i+12}x_2 + \dots + a_{i+1n}x_n - x_{n+i+1} = b_{i+1} \\ \dots \quad \dots \quad \dots \\ a_{m1}x_1 + a_{m2}x_2 + \dots + a_{mn}x_n - x_{n+m} = b_m \end{cases}$$

若目标函数为: $f(x) = C_1x_1 + C_2x_2 + \cdots + C_nx_n$

则可写成矩阵 $[A]$:

$$A = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} & 0 & b_1 \\ a_{21} & a_{22} & \dots & a_{2n} & 0 & b_2 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ a_{i1} & a_{i2} & \dots & a_{in} & 0 & b_i \\ a_{i+11} & a_{i+12} & \dots & a_{i+1n} & -1 & b_{i+1} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ a_{m1} & a_{m2} & \dots & a_{mn} & -1 & b_m \\ C_1 & C_2 & \dots & C_n & & 0 \\ Z_1 & Z_2 & \dots & Z_n & Z_{n+1} & Z_{n+2} \end{bmatrix}$$

在 A 阵中第 $m+1$ 行是目标函数的系数, 第 $m+2$ 行是引进的人工变量, 且有:

$$Z_n = - \sum_{l=|n|}^{\infty} a_{ln},$$

$$Z_{n+2} = - \sum_{l=1}^n b_l,$$

$$Z_{n+1} = - \sum_{l=1}^n (-1)^{l+1} \frac{1}{l} \left(\frac{1}{2} \right)^l$$

二、程序结构特点

本程序采取模块式结构与菜单选择技术。程序模块包含：引导程序块、输入程序块、菜单、运算程序块、输出程序块。

引导程序块是后续各程序块的先导并说明本程序的结构特点, 输入程序块按顺序输入变量数 $N1$, 约束条件

数 M ，约束矩阵 $[A]$ 各系数，目标函数各系数，以及半人工变量数组 $B(I)$ 各值。约束矩阵 $[A]$ 的输入运用特殊显示法去掉连续竖排与“?”号的缺陷使之符合数学矩阵排列的直观形式；目标函数、半人工变量数组 $B(I)$ 的输入也采用特殊显示法而获得直观的效果。其中的数组 $U(M+2, M+2)$ 为单位矩阵，程序变换时， A 阵不变而改变 U 的元素内容。而数组 $X1(M+2, M+2)$ 相当于输出值 $Y(i)_k$ ，数组 $X(M+2)$ 用以存放所有的解。数组 $S(3 * N1 + 2)$ 用以存放中间运算单元。本程序中使用 I, J, K, P, R, Q, NN, L 等均为简单变量。菜单可以选择不同的运算与输出：计有屏幕显示、打印输出、再次运行、退出等。每项功能使用后可自动返回主菜单或退出。运算程序块简洁明快，输出程序块使用方便。整个操作过程均用屏幕提示，并配以不同色彩的变换。使用户能在轻松愉快的环境中顺利完成各种枯燥乏味的计算。

三、运算实例

今有一目标函数为: $f(x) = 5x_1 + 3x_2 + 2x_3$

而其约束条件是: $4x_1 + 5x_2 + x_3 \geq 10$

$$5x_1 + 2x_2 - x_3 \leq 10$$

$$3x_1 + 8x_2 + 2x_3 \leq 12$$

$$-x_1 + 5x_2 + 4x_3 \geq 1$$

此时变量数 $N1 = 3$ (即有 x_1, x_2, x_3 三个变量)

而约束条件数 $M = 4$ (即有四个不等式)

根据上述原理将约束方程引入半人工变量后变为:

$$A = \begin{cases} 4x_1 + 5x_2 + x_3 - x_4 = 10 \\ 5x_1 + 2x_2 - x_3 + 0x_4 = 10 \\ 3x_1 + 8x_2 + 2x_3 + 0x_4 = 12 \\ -x_1 + 5x_2 + 4x_3 - x_4 = 1 \end{cases}$$

即 \leq 引入人工变量, \geq 不引入人工变量

因此[A]各系数输入的排列应如下:

4	5	1	-1	10
5	2	-1	0	10
3	8	2	0	12
-1	5	4	-1	1

相应目标函数变为: $f(x) = 5x_1 + 3x_2 + 2x_3 + 0x_4 + 0x_5$

因此, 目标函数各系数的输入排列应如下:

5 3 2 0 0

而数组 $B(I)$ 的输入排列根据引入的半人工变量应

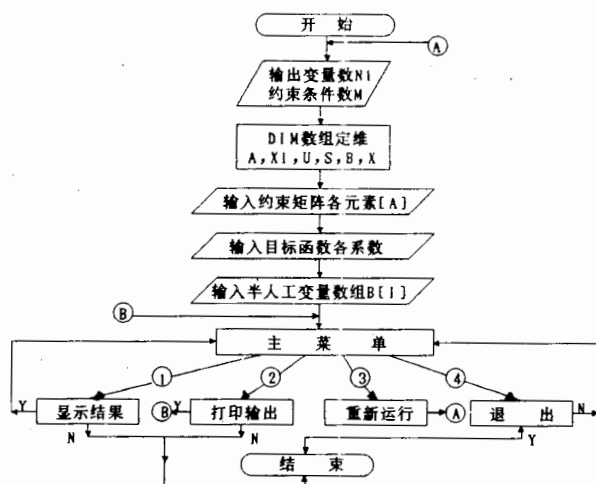
运算结果为:

OBJECTIVE FUNCTION IS

MIN = 10.588824

$$X(1) = 1.764706$$
$$X(2) = 0.5882359$$
$$X(3) = 0$$

四、程序框图与程序清单



```

10 CLS
20 KEY OFF: SCREEN 0,1: COLOR 14,1,15: WIDTH 80
30 LOCATE 2,20: PRINT "*****"
40 LOCATE 3,20: PRINT "THE PROGRAM OF LINEAR PROGRAMMING"
50 LOCATE 4,20: PRINT " "
60 LOCATE 5,20: PRINT "SOLUTION IN SIMPLEX METHOD"
70 LOCATE 6,20: PRINT " "
80 LOCATE 7,20: PRINT "Quizhou College of Engineering"
90 LOCATE 8,20: PRINT " "
100 LOCATE 9,20: PRINT "Mechanical engineering department"
110 LOCATE 10,20: PRINT " "
120 LOCATE 11,20: PRINT "Research section of"
130 LOCATE 12,20: PRINT " "
140 LOCATE 13,20: PRINT "fluid dridt and control engineering"
150 LOCATE 14,20: PRINT " "
160 LOCATE 15,20: PRINT "Develop by Huzzei"
170 LOCATE 16,20: PRINT " "
180 LOCATE 17,20: PRINT "March 1991 years"
190 LOCATE 18,20: PRINT "*****"
200 LOCATE 22,7: INPUT "number of variables N1=";N1
210 LOCATE 22,45: INPUT "number of constraints m=";m
220 DIM A(M+2,2 * N1+2),X1(M+2,M+2), U ( M+2,M+2) , S (3 * N1+2), B(2 * N1+2),X(M+2)
230 CLS
235 COLOR 12,9,4
240 Q=0
250 LOCATE 7,7:PRINT "enter matrix[A]"
260 NN=8
270 FOR I=1 TO M
280 NN=NN+2:LOCATE NN,20
290 FOR J=1 TO N1+2
300 INPUT;"",A(I,J)
310 NEXT J
320 PRINT
330 NEXT I
340 CLS

```

```

345 COLOR 13,14,2
350 LOCATE 7,7:PRINT"enter objective functiot"
360 PRINT
370 FOR I=1 TO N1+2
380 INPUT;"",A(M+1,I)
390 NEXT I
400 PRINT
410 LET F=1
420 LET P=M+1
430 CLS
435 COLOR 10,6,8
440 LOCATE 7,7:PRINT"evaluation of B(I)"
450 PRINT
460 LET R=0
470 FOR I=1 TO M
480 INPUT;"",B(I)
490 LET R=R+B(I)
500 NEXT I
510 CLS
515 COLOR 15,4
520 LOCATE 6,25:PRINT "***** MAIN MENU *****"
530 LOCATE 7,25:PRINT " "
540 LOCATE 8,25:PRINT " " 1 display " "
550 LOCATE 9,25:PRINT " " " "
560 LOCATE 10,25:PRINT " " 2 print output " "
570 LOCATE 11,25:PRINT " " " "
580 LOCATE 12,25:PRINT " " 3 Again run " "
590 LOCATE 13,25:PRINT " " " "
600 LOCATE 14,25:PRINT " " 4 Eixt !!! " "
610 LOCATE 15,25:PRINT " " " "
620 LOCATE 16,25:PRINT "*****"
630 LOCATE 20, 25:INPUT "choose number(1—4);I
640 CLS
650 ON I GOTO 660 1900,2180,2190
660 IF Q=0 THEN GOTO 680
670 GOTO 1600
680 Q=1
690 LET N=N1+1
700 IF R=0 THEN GOTO 980
710 FOR I=1 TO M+2
720 S(I)=0
730 S(I)=A(I,N1+2)
740 A(I,N1+2)=0
750 NEXT I
760 R1=0
770 FOR I=1 TO M
780 IF A(I,N1+1)> <-1 THEN GOTO 830
790 R1=R1+1
800 IF R=1 THEN GOTO 830
810 A(I,N1+1)=A(I,N1+1)
820 A(I,N1+1)=0
830 NEXT I
840 N=N1+R
850 FOR I=1 TO M+1
860 A(I,N+1)=S(I)
870 NEXT I
880 IF R=0 THEN GOTO 910

```

```

890 F=0
900 P=M+2
910 FOR J=1 TO N+1
920 A(P,J)=0
930 FOR I=1 TO M
940 IF B(I)><1 THEN GOTO 960
950 A(P,J)=AP,J)-A(I,J)
960 NEXT I
970 NEXT J
980 FOR I=1 TO P
990 U(I,I)=1
1000 FOR J=1 TO P
1010 IF J=I THEN GOTO 1030
1020 U(I,J)=0
1030 NEXT J
1040 NEXT I
1050 E=.000001
1060 W=-1
1070 FOR I=1 TO P
1080 B(I)=N+I
1090 NEXT I
1100 FOR I=1 TO P
1110 X(I)=0
1120 FOR J=1 TO P
1130 X(I)=X(I)+U(I,J)*A(J,N+1)
1140 NEXT J
1150 NEXT I
1160 IF F=1 THEN GOTO 1200
1170 IF X(P)+E<0 THEN GOTO 1210
1180 P=P-1
1190 F=1
1200 W=0
1210 K=0
1220 R=0
1230 FOR J=1 TO N
1240 S(J)=0
1250 FOR I=1 TO P
1260 S(J)=S(J)+U(P,I)*A(I,J)
1270 NEXT I
1280 IF S(J)-R+E>0 THEN GOTO 1310
1290 R=S(J)
1300 K=J
1310 NEXT J
1320 IF K=0 THEN GOTO 1600
1330 R=1E+31
1340 L=0
1350 FOR I=1 TO P
1360 X1(I,K)=0
1370 FOR J=1 TO P
1380 X1(I,K)=X1(I,K)+U(I,J)*A(J,K)
1390 NEXT J
1400 IF X1(I,K)-E<0 THEN GOTO 1470
1410 IF X(I)-E<0 THEN GOTO 1470
1420 IF I>M THEN GOTO 1470
1430 S(I)=X(I)/X1(I,K)
1440 IF S(I)-R=> THEN GOTO 1470
1450 R=S(I)
1460 L=I
1470 NEXT I
1480 IF L=0 THEN GOTO 1650
1490 FOR J=1 TO P
1500 U(L,J)=U(L,J)/X1(L,K)
1510 NEXT J
1520 FOR I=1 TO P
1530 IF I=L THEN GOTO 1570
1540 FOR J=1 TO P
1550 U(I,J)=U(I,J)-UL,J)*X1(I,K)
1560 NEXT J
1570 NEXT I
1580 B(L)=K
1590 GOTO 1100
1600 IF F=1 THEN GOTO 1690
1610 LOCATE 5,20:PRINT "infeasible"
1620 LOCATE 5,20:INPUT "Are you return menu or Exit(Y/N)";G$;CLS
1630 IF G$="Y" OR G$="y" THEN 520
1640 END
1650 LOCATE 5,20:PRINT "unbound"
1660 LOCATE 5,25:INPUT "Are you return menu or Exit(Y/N)";G1$;CLS
1670 IF G1$="Y" OR G1$="y" THEN 520
1680 END
1690 Y=-X(M+1)
1700 IF Y>0 THEN GOTO 1730
1710 LOCATE 5,20:PRINT "objective function is Max=";-Y
1720 GOTO 1640
1730 LOCATE 5,20:PRINT "objective function is Min=";Y
1740 FOR I=1 TO N1
1750 S(I)=0
1760 NEXT I
1770 FOR I=1 TO M
1780 J=B(I)
1790 S(J)=X(I)
1800 KK=7
1810 NEXT I
1820 FOR I=1 TO M1
1830 KK=KK+2
1840 LOCATE KK,30:PRINT "X(";I;")=";S(I)
1850 NEXT I
1860 LOCATE 20,45:INPUT "Are you return menu or Exit(Y/N)";Z$;CLS
1870 IF Z$="Y" THEN 520
1880 IF Z$="y" THEN 520
1890 END
1900 IF F=1 THEN GOTO 1990
1910 LOCATE 10,25:LPRINT "infeasible"
1920 LOCATE 10,25:INPUT "Are you return menu or Exit(Y/N)";O$;CLS
1930 IF O$="Y" OR O$="y" THEN 520
1940 END
1950 LOCATE 10,25:LPRINT "unbound"
1960 LOCATE 10,25:INPUT "Are you return menu or Exit(Y/N)";O1$;CLS
1970 IF O1$="Y" OR O1$="y" THEN 520

```

PASCAL 程序自动生成技术

四川重庆大学计算机系 符云清

程序正确性证明和自动生成是人工智能的一个分支。它能实现软件开发的自动化或半自动化,提高程序开发效率,缩短软件开发周期。在实际中特别是专家系统中有广泛应用和重要意义。在专家系统中,若能让知识库和推理机制自动或半自动生成,则能使领域专家在没有知识工程师的帮助下也能建立自己的专家系统,这将提高开发速度,减小开发费用。⁽¹⁾

笔者在开发“SKJ-II 数控线切割机床程序自动生成”软件时,便采用 PASCAL 程序自动生成技术。用户只需输入加工图形或对应坐标值,机器便自动生成加工程序。这在加工复杂工件时,既可避免繁杂计算,又能避免因大意而产生的错误。由于许多读者可能不熟悉该机床的指令系统,笔者现以一个简单的例子来说明该技术要点。

众所周知,程序自动生成关键要解决两个问题:一、功能模块的生成;二、功能模块的加载。⁽²⁾利用文件读写功能可以很方便实现第一问题,对于第二个问题可利用 TURBO PASCAL 的拼接技术,即通过 { \$ i filename.ext } 来实现文件加载。为进一步说明问题,现举例如下:实际中,我们可能对大量的函数进行某种操作,这些函数都千差万别,我们不可能针对每个函数表达式写一个过程,这是不明智的,也是不现实的。若能使应用软件根据不同要求生成相应功能函数,然后同处理模块拼接,则省时省力。附录程序 generator 通过与用户交互,利用文件读写功能生成相应功能模块 func,在应用程序中通过 { \$ i func } 将功能块拼接到应用程序中,以后编译,运行应用程序,便对应用程序进行相应处理,每运行

generator 便可以得到不同的功能块,为了提高效率,我们可以先生成所有的功能块,然后一起加载,这样只一次编译,运行便大功告成。

上面介绍的是一个极其简单的例子,目的在于以一斑,达到抛砖引玉的作用。总之,程序自动生成是一个很有发展前途和实用价值的领域;随着人工智能和 5GL (第五代语言)的发展,程序自动生成技术将进一步推广、发展。这对计算软件开发来说,无疑是一大幸事。

```
program generator (input, output); {程序自动生成器}
uses dos;

var
f: text;
strings: array[1..10] of string;
para, typ: array[1..5] of string;
fname, ftype, fn: string;
i, num: integer;
begin
assign (f, 'func.pas'); rewrite (f);
writeln('input function name'); readln (fname);
writeln('input function type'); readln (ftype);
writeln('input how many parameters'); readln (num);

for i:=1 to num do
begin
writeln('parameter', i, ':');
readln (para[i]);
writeln('type', i, ':');
readln (typ[i]);
end;
writeln('the expression of function is:'); write (fname, '
-'); readln (fn);
```

```
1980 END
1990 Y=-X(M+1)
2000 IF Y>0 THEN GOTO 2030
2010 LOCATE 10,25: LPRINT "objective function is Max="; -Y
2020 GOTO 2040
2030 LOCATE 10,25: LPRINT "objective function is Min="; Y
2040 FOR I=1 TO N1
2050 S(I)=0
2060 NEXT I
2070 FOR I=1 TO M
2080 J=B(I)
2090 S(J)=X(I)
2100 NEXT I
2110 FOR I=1 TO N1: LPRINT
2120 LOCATE I+12,30: LPRINT "X(";I);S(I)
2130 NEXT I
2140 LOCATE 3,8: INPUT "Are you return menu or Exit (Y/N)"; K
$:CLS
2150 IF K$="Y" THEN GOTO 520
```

```
2160 IF K$="Y" THEN 520
2170 END
2180 RUN
2190 LOCATE 5,10: INPUT "Are you sure"; A$
2200 IF A$="Y" OR A$="Y" THEN GOTO 2170
2210 GOTO 510 (该程序文件名: ZD23.PRG)
```

参考文献

- [1] M. Noton, MODERN CONTROL ENGINEERING. Pergamon Press 1972
- [2] HO.D.and NOTON. M "computation of control with a Canada Econometric Model". to appear in applied Economics.
- [3] An-min chung "Linear Programming" May 1962
- [4] 周明德主编, 微型计算机 IBM-PC 系统原理及应用, 清华大学出版社, 1985 年
- [5] 计算所十七室八室编, MS-DOS 磁盘操作系统
- [6] David.A.Lien. The BASIC Handbook

直方图的显示与打印

九龙海关技术处 陈文彤

直方图可以直观、明确地表示统计数据的变化趋势和分布状况。在考虑用电脑处理直方图时笔者发现,使用工具软件可以在微机或图形终端方便地绘出直方图,而要在字符终端或打印机上绘出该图,则略为复杂。事实上,关键在于打印机和字符终端只能以字符,而不能以点和直线(曲线)组成直方图图形。由此,笔者经过试验,使用 COBOL 语言,编制了一个直方图通用打印子程序 General pr image sub。程序约 200 余行,由主程序调用可以在 STAR2463 打印机上绘出较为美观的直方图。(对其它类型打印机而言,仅需修改程序中的打印控制码即可)。以下介绍其设计思想和使用方法。

一、设计技术

一幅直方图可以看作是在坐标系第一象限内若干纵向轴列“COLUMN”(简称“纵列”)的序列,它可以由读入的对应于各纵列的顶点系列决定。通常,这些顶点是按其 X 坐标的大小逐个读入的,因此可以在终端上逐列显示每个顶点对应的纵列。而对于打印机来说,由于打印头不能退行,即只能逐行打印,这增加了编程的复杂性。解决的方法是建立一个类似虚拟打印屏幕作用的数据缓冲二维数组:

```
01 print_image.
```

```
05 print_line occurs 75 times pic X(100).
```

程序逐个读入顶点数据,将之转换为 75 行 100 列的虚拟屏面 Print image 上的直方图形式。最后才在打印机上将建立在 Print image 上的直方图逐行打印出来(恰好打满一页标准窄行打印纸)。

```
strings (1) := 'function '+fname+'(';
for i := 1 to num-1 do strings (1) := strings (1) + para (i) + ' ';
+typ (i) + ' ';
strings (1) := strings (1) + para (num) + ':'+typ (num) + ')';
+fntype+'';
strings (2) := 'begin';
strings (3) := ' '+fname+': = '+fn+'';
strings (4) := 'end;';
for i := 1 to 4 do writeln (f, strings (i));
close (f);
writeln('enter turbo compile run utility program!');
end
```

```
A>
A> 功能模块 func 为:
A> type func . pas
function func_name (x1: integer; x2: real; x3: real):real;
begin
func_name := x1+2*x2-x3;
end;
A>
```

从直方图打印的美观着眼,纵列之间应该分离,纵列本身应该连续。经比较,笔者选用了反像显示的空格字符用以组成纵列“黑柱”,同时设定行间压缩状态,即设定送纸量为 1/8 英寸。结果打印出的直方图如附图所示。显然其他字符的打印效果均不如以上方法。

二、调用方法

作为可调用的子程序,general_pr_image_sub 的调用参数是:

① 直方图数组: Print_image.

② 横、纵坐标的刻度大小: Hori_interval, Vert_interval.

③ 横、纵坐标的起始刻度: Hori_init, Vert_init.

④ 每一次读入的纵柱顶点的横、纵坐标: Hori_value, Vert_value.

⑤ 直方图打印标题: Top_headline.

⑥ 调用结果: Result.

调用功能类型: Action_flag

Action_flag = "I" ————— 初始化数组 Print_image.

Action_flag = "R" ————— 读入一个纵柱顶点坐标到 Print_image.

Action_flag = "P" ————— 将 Print_image 内直方图送至打印机.

主程序的调用方法是:

```
Call "general_pr_image_sub" using
....., "I".
```

A>

```
A> type utility2 . pas { 实用程序调用功能模块 }
program file2 (input, output);
var
a: integer;
b, c, d: real;
($ifunc . pas)
begin
writeln('x1 = x2 = x3');
readln (a, b, c);
d := func_name (a, b, c);
writeln('d = ', d);
end.
```

(该程序文件名: ZC18 . PRG)

注:

[1] 朱稼兴 《专家系统与现代管理》北京航空航天大学出版社 1990 年

[2] 顾景文 《PC BASIC 程序自动生成技术》,《计算机时代》1990 年第 1 期。


```

Perform varying I from 1 by 1 until I >
Number_of_columns
    送参数值到各调用参数处
    Call "general_pr_image_sub" using print_image,
    hori_interval,
    vert_interval, hori_init, vert_init, hori_value,
    vert_value, top_headline, result, "R"
End-Perform.
Call "general_pr_image_sub" using
    ..... , "P".

```

三、主要算法

(1) 子程序的三个调用功能完成以下工作:

① Action_flag = "I" ————— 将二维数组 Print_image 每一元素填以空格字符。

② Action_flag = "R" ————— 读入一顶点坐标, 并在二维数组 Print_image 上绘出相应纵柱形式 (Column)。

```

Move (vert_value - vert_init) / vert_interval to column_height. /* 计算纵柱高度 Column_height
Move ((hori_value - hori_init) / hori_interval) * 2 +
14 to column_position.
/* 计算纵柱位置 Column_position
Move 70 to disp_line. /* 纵柱底部行坐标
Perform test after until column_height < 1
move "*" to print_line(disp_line)(column_height:1)
/* 全黑字符标志
subtract 1 from disp_line subtract 1 from
column_height
End-Perform.

```

③ Action_flag = "P" ————— 将 Print_image 内直方图送至打印机。

按调用参数在 Print_image 上用图形字符绘出坐标系框架。

打印标题汉字。

```

Perform varying line_register from 1 by 1 until
line_register > 70
display print_line(line_register)(1:11) no /* 打印纵轴
Perform varying column_register from 12 by 1 until
column_register > 100

```

```

If print_line(line_register)(column_register:1) = "*"
display " " reversed /* 黑白反现的空格 (全黑)
Else
display print_line(line_register)(column_register:1)
End-If
End-Perform

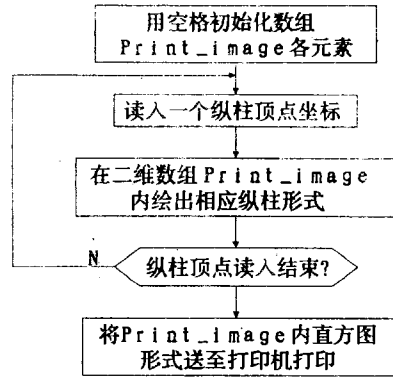
```

换行回车

End-Perform.

打印横轴和横轴下的注解与数字。

(2) 以流程图图表示其工作流程:

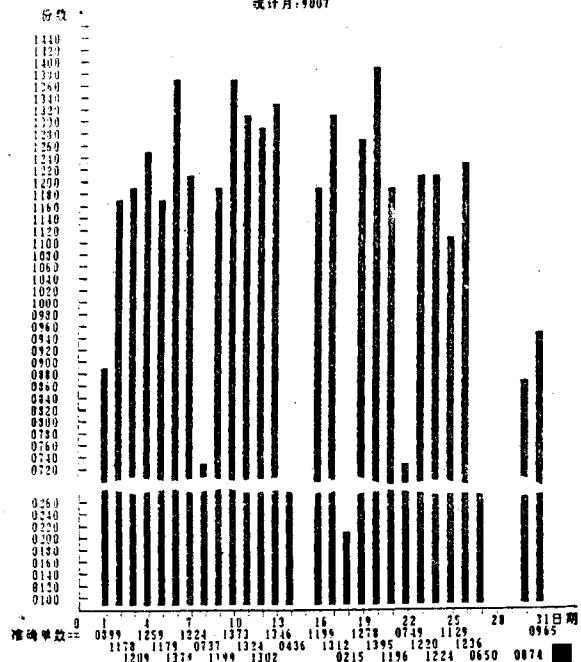


本程序绘制的直方图中, 纵柱的高度按字符高度递增的, 其数值递增单位等于纵轴刻度长(Hori interval), 若想更精确的话, 可以用打印机提供的功能——点影象, 用全黑部分高度不同的点影象代替各纵柱顶点的全黑字符。另外, 如果在打印纸上按横的方向打印直方图, 也可以不采用数组 Print_image 的方法, 但要比本方法麻烦的多。最后, 对于在字符仿真终端上打印直方图, 方法基本相同, 而且由于可以全屏幕显示, 它可以不经数组 Print_image, 直接在屏上绘图。

四、应用实例

在九龙海关 H883 报关自动化系统的理单子系统中, 系统在统计分析每日或每月在文锦渡海关申报并理单的报关单的数量情况后, 使用本子程序以直方图形式将其提供给业务人员和领导参考, 得到较好的评价, 本文在后面附上一张打印出的直方图。对于本子程序来说, 由于它的简单与实用, 在使用电脑分析数据的变化趋势或分布状况方面, 将会得到更多的应用。

理单报关单每日统计 直方图



ADC 直接与 PC 接口连接的电路

常州电子计算机厂 周仲玉

当将一个模-数转换器 (ADC) 与一台个人计算机连接时, 必须在 PC 计算机的扩充 I/O 槽里插上一块 A/D 插卡, 而在下面将介绍一种新颖简单的将 ADC 与个人计算机连接的办法, 它的最大优点是无须打开机器的机壳, 而且在 PC 机的插槽已经用完的情况下仍然可以连接。

绝大部分个人计算机, 不管是从 8088 CPU 的便携式 PC 机, 还是 80386 CPU 的高档 PC 机, 它们都配有一个连接打印机的平行接口, 可以利用该接口来连接一个 ADC, 图 1 是一个以 PC/XT/AT 的 25 芯 D 型插座为基础的, 仅用两个集成块便构成一块 A—D 转换卡的电路图, 通过 PC 的数据输出口 378H 对模数转换器 (AD7572, MAX162, MAX172) 进行控制, 通过 379H 口的状态输入线读取模数转换数据。

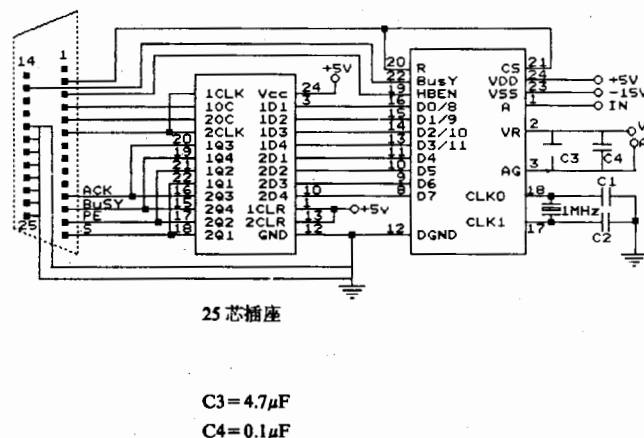


图 1

由于 25 芯插座只有 5 个引脚与口 379H 相连, 所以图 1 中 12 位的转换数据要读三次, 这可以通过对 D 型触发器 74LS878 进行控制来完成。除 BUSY 信号外, 其它信号都可以通过输出口 378H 产生, 见图 2 所示。

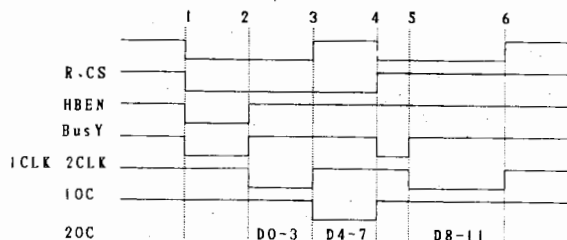


图 2

采用这种原理将 ADC 与 PC 相连的另一个好处是灵活性大, 如 12 位的 ADC 也可以换成 10 位或 8 位的

ADC, 如果同时连接几个模数转换器, 那么只要利用口 378H 输出的其它几位便能实现。

如果打印机和 A—D 转换器同时连在一个输入输出口, 那么可以使用一个 74LS157 来控制输入口 379 的输入线数据 (见图 3), 可以采用输出口 37AH 的 SLCTIN 来作为选择打印机还是 ADC 的控制信号。

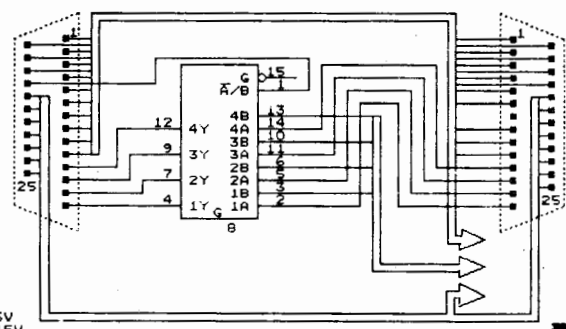


图 3

(上接 43 页)

CS: 114H 改为未染毒前的 IP 值

CS: 116H 改为未染毒前的 CS 值

(11) -W

把修改后的 111 文件存盘。

(12) -q

退出 debug 状态, 返回 DOS 的“A>”提示符

(13) A>copy C: 111 C: \xxx*.EXE

则该 EXE 文件已不带旅游者病毒, 且可正常运行。

笔者有一个 JIEDU.EXE 文件, 未感染旅游者病毒时, 文件长 14432, 感染旅游者病毒后, 文件长度为 15655, 文件增长 1223 字节。按上述方法消毒后, 所得的文件长 14432, 与原文件长度一致, 且可正常运行。

另有一个 HD.EXE 文件, 未感染旅游者病毒时, 文件长 8826 字节, 感染旅游者病毒后, 文件长 10055, 文件增长 1229 字节, 按上述步骤消毒后, 文件长 8832, 比原文件增长 6 字节, 可正常运行该文件。

文件长度经解毒后出现一些误差的原因是消毒过程中对文件实际占扇区数及文件长度 mod (512) 作简化处理, 避免繁杂计算而引起, 但不影响原文件正常运行。

上述对感染旅游者病毒的 COM 文件及 EXE 文件的消毒步骤, 与另一种文件型病毒——毛毛虫 (1575) 病毒非常相似, 读者可参考 1991 年第 2 期《电脑》杂志上胡向东所写文章:《毛毛虫病毒的分析及排除方法》, 从中可以领略到排除文件型病毒的许多带共通性的要领和步骤。

我们可以根据用 debug 清除旅游者病毒的步骤, 编写出检测及排除旅游者病毒的解毒软件来。

AR3240 打印机驱动程序

福建省漳浦县税务局 王月成

AR3240 打印机具有很强的打印功能,但其控制码一般都是以 ASCII 码 26,27,28 开头,这样,在用字处理软件(如 WPS)打印文本文件就无法控制打印机,若用通常的打印驱动程序则既降低了打印速度,又无法充分利用打印机的所有功能,给应用带来了很大麻烦。为此,本人设计了 AR3240 打印机驱动程序,它在保留原有的所有在程序中发送的控制码外,加入可在文本文件中输入的打印控制码。

打印机控制码的编写规则为:控制码必须用二个“,”(“,”的 ASCII 码为 96,与“~”同在一键)作为起止标识符,用“^”代替原打印控制码的 ASCII 码 26,用“\”代替 ASCII 码 27,用“~”代替 ASCII 码 28,其它与原有打印控制码同,若控制码需发送 ASC 码则用方括号“[”和“]”把数值括起来,若发送的 ASCII 码对应字符可显示,亦可用该字符代替(ASCII 码 96 为标识符“,”除外)。

AR3240 打印机驱动程序输入:

```

C:\>DEBUG
-A
35E0:0100 JMP 0199
35E0:0103 CMP AL,60
35E0:0105 JNZ 0122
35E0:0107 CS:
35E0:0108 CMP BYTE PTR[011F],B3
35E0:010D JZ 0116
35E0:010F CS:
35E0:0110 MOV BYTE PTR[011F],B3
35E0:0115 IRET
35E0:0116 CS:
35E0:0117 MOV [011F],AL
35E0:011A IRET
35E0:011B ADD [BX+SI],AL
35E0:011D ADD DL,AL
35E0:011F MOV BL,C9
35E0:0121 NOP
35E0:0122 CS:
35E0:0123 CMP BYTE PTR[011F],B3
35E0:0128 JNZ 0136
35E0:012A PUSH DX
35E0:012B XOR DX,DX
35E0:012D PUSHF
35E0:012E CALL F000:EFD2
35E0:0133 POP DX
35E0:0134 IRET
35E0:0135 NOP
35E0:0136 CMP AL,5C
35E0:0138 JNZ 013ED,A
35E0:013A MOV AL,1B
35E0:013C JMP 012A
35E0:013E CMP AL,5E
35E0:0140 JNZ 0146
35E0:0142 MOV AL,1A
35E0:0144 JMP 012A
35E0:0146 CMP AL,7E
35E0:0148 JNZ 014E
35E0:014A MOV AL,1C
35E0:014C JMP 012A
35E0:014E CMP AL,5B
35E0:0150 JNZ 0157
35E0:0152 CS:
35E0:0153 MOV [011E],AL
35E0:0156 IRET
35E0:0157 CMP AL,5D
35E0:0159 JNZ 016D
35E0:015B CS:
35E0:015C MOV AL,[011D]
35E0:015F CS:
35E0:0160 MOV BYTE PTR[011D],00
35E0:0165 CS:
35E0:0166 MOV BYTE PTR[011E],C2
35E0:016B JMP 012A
35E0:016D CS:
35E0:016E CMP BYTE PTR[011E],C2
35E0:0173 JNZ 0177
35E0:0175 JMP 012A
35E0:0177 CMP AL,30
35E0:0179 JNB 017C
35E0:017B IRET
35E0:017C CMP AL,39
35E0:017E JLE 0181
35E0:0180 IRET
35E0:0181 PUSH BX
35E0:0182 PUSH AX
35E0:0183 CS:
35E0:0184 MOV BL,[011D]
35E0:0188 MOV AL,0A
35E0:018A MUL BL
35E0:018C MOV BL,AL
35E0:018E POP AX
35E0:018F ADD AL,BL
35E0:0191 SUB AL,30
35E0:0193 CS:
35E0:0194 MOV [011D],AL
35E0:0197 POP BX
35E0:0198 IRET
35E0:0199 MOV AH,09
35E0:019B MOV DX,01B0
35E0:019E INT 21
35E0:01A0 MOV AX,2517
35E0:01A3 MOV DX,0103
35E0:01A6 INT 21
35E0:01A8 MOV AX,3100
35E0:01AB MOV DX,0199
35E0:01AF INT 21
35E0:01B0 DB "AR3240打印机驱动程序使用示例:",
35E0:01D8 DB "有级放大8*8 [96]
\c[8][8]-
35E0:01FEDB "无级放大65*65 [96]
-c[65][65] [96]
35E0:0228 DB "设十二点空距 [96] P
[96] ",D,A
35E0:023E DB "福建省漳浦县税务局王
月成",D,A,"$"
35E0:025B
-R CX
CX 0000
:160
-N AR3240.COM
-W
Writing 0160 bytes
-Q
C:\>
另: 附后台打印程序
C:\>DEBUG
-A
356A:0100 JMP 0179
356A:0102 DB "C:\LPRINT.COM",0
356A:0110 STI
356A:0111 PUSH AX
356A:0112 PUSH BX
356A:0113 PUSH CX
356A:0114 MOV CX,0001
356A:0117 CS:
356A:0118 MOV BX,[0107]
356A:011C CS:
356A:011DMOV AL,[BX]
356A:011F MOV AH,00
356A:0121 INT 17
356A:0123 INC BX
356A:0124 CS:

```

356A:0125 CMP BX,[0105]	356A:0197 NOP	356A:0209 JZ 0206
356A:0129 JGE 0136	356A:0198 MOV SI,0082	356A:020B DEC SI
356A:012B LOOP 011C	356A:019B MOV DL,AL	356A:020C MOV DX,SI
356A:012DCS:	356A:019D MOV AH,05	356A:020E MOV AX,3D00
356A:012E MOV [0107],BX	356A:019F INT 21	356A:0211 INT 21
356A:0132 POP CX	356A:01A1 LODSB	356A:0213 JNB 021E
356A:0133 POP BX	356A:01A2 CMP SI,DI	356A:0215 MOV DX,0250
356A:0134 POP AX	356A:01A4 JB 019B	356A:0218 MOV AH,09
356A:0135 IRET	356A:01A6 MOV DL,0D	356A:021A INT 21
356A:0136 MOV AX,000C	356A:01A8 MOV AH,05	356A:021C INT 20
356A:0139 INT 17	356A:01AA INT 21	356A:021E MOV BX,AX
356A:013B CS:	356A:01AC MOV DL,0A	356A:0220 MOV DX,0250
356A:013C MOV CL,[0104]	356A:01AE MOV AH,05	356A:0223 MOV CX,FFFF
356A:0140 CMP CL,01	356A:01B0 INT 21	356A:0226 MOV AH,3F
356A:0143 JLE 0157	356A:01B2 INT 20	356A:0228 INT 21
356A:0145 DEC CL	356A:01B4 MOV DX,0260	356A:022A JNB 022E
356A:0147 CS:	356A:01B7 MOV AH,09	356A:022C JMP 0215
356A:0148 MOV [0104],CL	356A:01B9 INT 21	356A:022E ADD AX,0250
356A:014CCS:	356A:01BB MOV AX,0C01	356A:0231 MOV [0165],AX
356A:014DMOV WORD PTR[0107],0250	356A:01BE INT 21	356A:0234 MOV WORD PTR[0107],0250
356A:0153 POP CX	356A:01C0 CMP AL,31	356A:023A MOV CX,AX
356A:0154 POP BX	356A:01C2 JB 01E5	356A:023C MOV AX,251C
356A:0155 POP AX	356A:01C4 CMP AL,39	356A:023F MOV DX,0110
356A:0156 IRET	356A:01C6 JA 01E5	356A:0242 INT 21
356A:0157 PUSH DS	356A:01C8 SUB AL,30	356A:0244 MOV DX,CX
356A:0158 PUSH ES	356A:01CAMOV [0115],AL	356A:0246 MOV AX,3100
356A:0159 MOV AX,0000	356A:01CD MOV DX,0102	356A:0249 INT 21
356A:015CMOV DS,AX	356A:01D0 MOV AX,3D01	356A:024B DB "文件不存在",D,A,"\$"
356A:015E MOV AX,[0072]	356A:01D3 INT 21	356A:0260 DB "后台打印程序使用说明",D,A
356A:0161 MOV ES,AX	356A:01D5 MOV BX,AX	356A:0276 DB "格式: LPRINT份数路径\文件名",D,A
356A:0163 MOV AH,49	356A:01D7 MOV DX,0100	356A:0296 DB "LPRINT字符串",D,A
356A:0165 INT 21	356A:01DAMOV CX,026F	356A:02ABDB "注: 若LPRINT后没参数, 则提示使用说明和选择",D,A
356A:0167 MOV WORD PTR[0070],FF53	356A:01D1MOV AH,40	356A:02D9DB "后台打印速度,每秒打印N*(N1-9)个汉字",D,A
356A:016DMOV WORD PTR[0072],F000	356A:01DFINT 21	356A:0307 DB "本程序在AST-286微机,AR3240 打印机上调试通过",D,A
356A:0173 POP ES	356A:01E1 MOV AH,3E	356A:0335 DB "研制人: 福建省漳浦县税务局王月成",D,A
356A:0174 POP DS	356A:01E3 INT 21	356A:0359 DB "请输入打印速度N=?\$"
356A:0175 POP CX	356A:01E5 INT 20	356A:036F -R CX
356A:0176 POP BX	356A:01E7 DEC SI	CX 0000
356A:0177 POP AX	356A:01E8 MOV CX,000A	:026F
356A:0178 IRET	356A:01EB LODSB	-N LPRINT.COM
356A:0179 MOV SI,0082	356A:01ECCMP SI,DI	-W
356A:017CMOV BL,[SI-02]	356A:01EEJZ 0198	Writing 026F bytes
356A:017F MOV BH,00	356A:01F0 CMP AL,20	-Q
356A:0181 LEA DI,[BX+SI]	356A:01F2 JZ 0200	C<>
356A:0183 MOV BYTE PTR[DI-01],00	356A:01F4 AND AL,0F	
356A:0187 CMP SI,DI	356A:01F6 XCHG AL,CH	
356A:0189 JNB 01B4	356A:01F8 MUL CL	
356A:018B LODSB	356A:01FAADD AL,CH	
356A:018CCMP AL,31	356A:01FCMOV CH,AL	
356A:018E JB 019B	356A:01FEJMP 01EB	
356A:0190 CMP AL,3A	356A:0200 MOV CL,CH	
356A:0192 JB 01E7	356A:0202 MOV [0104],CL	
356A:0194 JMP 019B	356A:0206 LODSB	
356A:0196 NOP	356A:0207 CMP AL,20	

该程序文件名为 ZA32.PRG, 拷入本期源程序盘 (编号为 P91-4) 中, 邮购办法参阅第 页小启。 ■

《苹果全能系统》最新版本功能简介

钟华宁

· 笔者自去年在《电脑》杂志第5期推出《苹果全能系统 V2.0》以来,很受读者欢迎。一些热心的读者纷纷来函提出对新版本的建议和旧版本存在的失误与不足。为了感谢众多读者的支持,笔者经过几个月的努力,终于研制出《苹果全能系统 V3.1 版本》,献给广大的苹果机爱好者。

《苹果全能系统 V3.1》(以下简称 V3.1), 在 2.0 的基础上更上一级台阶。除了修改 2.0 存在的失误及保持 2.0 兼容性好、占用内存少的优点外, 还继续扩展了多条实用语句, 并且在目前所有同类系统中首创扩展函数功能, 这可谓又来一个飞跃。V2.0 的主要功能读者可查阅《电脑》杂志 1990 年第 5 期, 在此不再赘述。本文仅就 V3.1 增加的功能介绍给读者。

首先，V3.1 中增加一条宏代换命令 **MACRO**，与 **dBASE** 中的 **&** 命令非常相似，用户若巧妙运用的话，会发现其妙用无穷，它能将平时一些非常复杂甚至用 **BASIC** 无法办到的事情轻而易举的解决，如将计算机模拟成计算器，在程序执行过程中增加、删除或修改程序自身，而不使程序停下来等。绘图方面，高分辨率的字符可在屏幕的任意位置、任意大小、任意颜色显示；**HIRES** 命令可为用户对高分辨率进行简单的编辑或为其它的绘图指令如 **TAKE**、**HPLOT** 等提供坐标信息；还有一个 **ACT** 命令可帮助用户制作动画程序，它允许任意多个物体同时运动，有单页或双页交替切换显示，而且数据完全透明，用户可方便进行修改。

本系统共增加了 7 条函数：LOGIC 函数可对两数进行与、或、异或、循环左移或右移五种逻辑运算；SCR 函数可判断高分辨率或文本某一坐标上某一点是否发光或字符的代码；KEY\$ 可将按键的代码赋给指定变量；HEX\$ 和 \$ 两个函数可实现 16 进制与 10 进制数之间的相互转换；SCAN 函数可快速查找内存中指定的数值，字符甚至是 BASIC 指令的代码。VAR 函数可获得任一变量的存储地址。

DOS 中的 BLOAD 或 BRUN 命令可显示文件的起始地址, 结束地址和长度, 若调用的文件可能会破坏系统时, 系统会提示用户最好放弃该操作 (用户可不理睬)。有较强的自我保护能力。

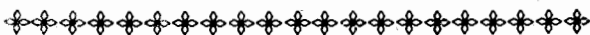
V3.1 采用向上兼容的方式, 除一两条命令外基本同 V2.0 兼容, 占用用户内存也仅 $\frac{1}{2}K$, 还有许多空间供有能力用户扩展。

V3.1 与目前流行的 ACE 软件功能相似, 两者比较各有千秋, ACE 是一个专门的绘图与动画设计软件, 在这方面比 V3.1 要稍胜一筹, 但她只能使用其本身的 BASIC 语言 (与 Applesoft 不兼容), 无法使用监控, 用户的汇编语

言程序不能调入，兼容性差和透明度低是她的致命弱点，而这些正是 V3.1 的成功之处，以下是 ACE 与 V3.1 功能上的一些比较，仅供参考：

	ACE	V3.1
兼容性	差	好
汉字系统	系统带有软汉字系统	使用 CEC-I 硬卡汉字
占用用户内存	大	少
透明度	低	高
应用程序及演示程序	多而且相当出色	少, 简单
函数扩展功能	无	7 条
删除原 Appiesoft 命令	24 条及所有 DOS3.3 命令	无
新增及扩展命令	33 条	35 条

V3.1 出售价格与 2.0 相同, 不加密 50 元/套, 加密: 30 元/套。已购买 V3.1 版本前任意版本的用户可五折优惠, 说明书: 2 元/份。■



宝迪牌打印机色带系列产品面市

为迎合国内市场的需要，广东省计算所天河电脑公司推出价优质良的宝迪牌全系列打印机色带。

宝迪色带特别采用色泽深润的油墨配方及日本制造的高密度优质尼龙带印染而成。带盒设计, 则充分利用CAD 电脑设计, 生产过程, 全面采用日本自动机械设备。因此, 该色带具有打印效果鲜明清晰, 使用寿命长久, 长时间打印后, 依然保持原来色泽等特点。

为顾及各方面用户的需要,宝迪色带各种品种规格齐全,应有尽有。特别是四色色带,对于打印彩色文字和图像,更能发挥彩色打印机的功能。

目前, 该色带在广州天河科技街 155 号及江南大酒店 2603 室有批发零售。

结构系统模块层次图

华中理工大学 万希宁 杨家新

管理信息系统通常使用 dBASEIII 小型关系数据库作为系统开发的主要语言。由于企业组织包含的管理职能较多, 软件系统的规模也相对较大, 为了使系统既达到预定的功能要求, 又具有清晰的结构, 通常采用结构系统分析和设计方法设计软件系统, 用结构程序设计实现系统功能。

系统模块化程度越高, 其模块数量也越多。基于以下两点原因, 需要在系统开发过程中不断地跟踪实施情况。第一, 软件开发人员和负责人必须要随时了解系统开发的进度和结构; 第二, 在系统实施过程中系统结构仍在不断地演化和调整, 在系统文档的建立中, 要保证文档与系统同步。但是整理和跟踪系统结构所要付出的代价将是较大的, 也是繁琐的。为了能方便地取得系统的程序模块结构层次图, 这里编制了一个自动分析模块结构和调用关系的软件。

一、程序结构

自动识别系统程序模块的层次和调用关系是一个字处理过程, 如果处理方法运用不当, 需要利用多个中间文件进行过渡, 处理过程也较复杂。笔者曾见过几个类似的软件, 使用的中间文件较多, 源程序较长, 程序也复杂。

这里提供的源程序不到二百行, 见附源程序, 整个处理过程由四部分组成, 分别描述如下:

第一部分为 1~33 行, 其功能是获取信息系统的主控模块名, 以决定层次结构图的树根。

第二部分为 34~68 行, 其目的是从系统的当前模块结构中读取一个程序模块, 以待分析。

第三部分为 69~94 行, 对读入的程序模块进行分析, 识别和记录下层模块, 并自动绘出一个层次编号, 其中第 88~94 行用于寻找本层次的下一模块。

第四部分为 95~114 行, 其目的是将所有的分析结果按层次结构排列, 模块名在各行中列的缩进位置表明了其在系统中的地位, 而每个程序模块的下层子模块, 均列于该模块之下, 如图一。

在处理过程中, 只使用了两个数据库文件, 一个用于存放源程序模块文件内容, 另一个用于记录分析所得的系统层次结构, 这两个数据库文件的结构在附源程序 9~13 行的备注中给予了说明。

二、程序的使用

本程序是在 IBM-PC 的 dBASEII PLUS 环境下运行, 使用时, 要求用户通过键盘送入待分析的信息系统主控模块名。在程序运行期间, 当前正在被处理的子模块名会被显示在屏幕上, 直到系统中所有的程序模块都处理完毕。

三、有关源程序的说明

本程序除了以上功能, 还可执行以下几项任务, 一是这些任务的实现较简单, 二是因源程序的篇幅问题, 因此在源程序中略去了这部分内容, 但是这并不影响附源程序的完整性。以下对这几个功能作一简要说明。

首先, 程序允许用户输入系统所在的磁盘子目录名。处理时, 只要将输入的子目录名加到程序模块名之前即可。

其次, 本程序可以将系统中所有模块、层次关系以层次图的格式输出, 对于包装在过程文件中的模块, 也能一一列出其层次关系。有关过程文件的处理程序约二十余行, 可以简单地插在源程序的第 71~72 行或 85~86 行之间。

另外, 分析结果的输出, 可以按用户自己的习惯处理, 只需将这段程序附加到源程序的末尾。在附源程序中, 只简单地用一个命令行指出了分析结果的存储位置。需要的话, 可以用 LIST 命令直接输出。

主控模块	CWMAIN.prg	* 财务系统主控模块
1	CWENTER.prg	* 输入数据
11	CWENTSUB.prg	* 输入数据子程序
2	CWMODIFY.prg	* 修改凭证数据
21	CWMODSUB.prg	* 修改模块子过程
3	CWSEEK.prg	* 查询模块
4	CWREPORT.prg	* 输出报表
41	CWB1.prg	* 利润表
42	CWB2.prg	* 资金平衡表
43	CWB3.prg	* 资金流转表

图一 输出结果

附: 源程序

```

1: * 文件层次图 DHIP01.prg 1990.11.10
2: SET TALK OFF
3: SET SAFE OFF
4: FN=" "
5: P=" "
6: PN=1
7: STORE 0 TO LK, L
8: SELE A
9: USE DHIP01 && 过渡文件 结构 AA BB CC
10: ZAP && CHAR 10 12 80
11: SELE B
12: USE DHIP0 && 层次文件 结构 AA
13: ZAP && CHAR 90

```



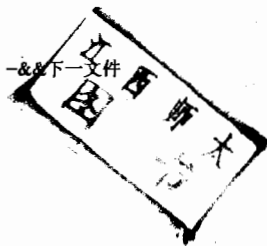
```

14: DO WHILE .T.
15: CLEAR
16: @1, 20 SAY "*****文件层次图*****"
17: @4, 23 SAY "主文件名: "GET FN PICT "XXXXXXXX"
18: READ
19: SELE A
20: P = " "
21: FN = TRIM (FN)
22: IF LEN (FN) = 0
23: RETU
24: ENDIF
25: @6, 30 SAY FN
26: IF .NOT. FILE ("&FN..PRG")
27: LOOP
28: ELSE
29: EXIT
30: ENDIF
31: ENDDO
32: APPE BLANK
33: REPL AA WITH P, BB WITH FN + ".prg", && 层号, 文件名
    写入 A 区
34: DO WHILE .T.
35: SELE A
36: SET FILT TO LEN (TRIM (AA)) = L && 设置层深
37: COUN ALL TO RC
38: IF RC = 0
39: EXIT
40: ENDIF
41: GO TOP
42: DO WHILE .NOT.EOF()
43: FN = BB && 取下一文件名
44: @6, 30 SAY FN
45: P1 = AA
46: PN = 1
47: REC = RECN()
48: IF LEFT (FN, 1) = CHR (38)
49: REPL CC WITH "-宏代换文件名-"
50: SKIP
51: IF EOF()
52: EXIT
53: ENDIF
54: LOOP
55: ENDIF
56: IF .NOT.FILE (FN)
57: REPL CC WITH "-在过程或不在系统-"
58: SKIP
59: IF EOF()
60: EXIT
61: ENDIF
62: LOOP
63: ENDIF
64: SELE B

65: APPEN FROM &FN SDF &&读源程序
66: GO TOP
67: SELE A
68: REPL CC WITH LEFT(LTRIM(B >AA, 40)&&备注写入 A 区
69: SELE B
70: DO WHILE .NOT.EOF()
71: MAA = LTRIM (UPPE (AA))
72: IF LEFT (MAA, 2) = "DO" &&寻找源程序名
73: MAA = LTRIM (SUBS (MAA, 3))
74: LAA = LTRIM (LEFT (MAA, 4))
75: IF LAA = "WHIL".OR.LAA "CASE"
76: ELSE
77: MAA = TRIM (LEFT (MAA, 8))
78: SELE A
79: P = TRIM (P1) + STR (PN, 1) &&生成层号
80: APPE BLANK &&层号, 文件名 写入 A 区
81: REPL AA WITH P, BB WITH MAA
82: PN = PN + 1
83: SELE B
84: ENDIF
85: ENDIF
86: SKIP
87: ENDDO
88: ZAP
89: SELE A
90: GO REC
91: SKIP
92: ENDDO
93: L = L + 1
94: ENDDO
95: SELE B
96: ZAP
97: SELE A
98: SET FILT TO
99: INDE ON AA TO DHIP01
100: GO TOP
101: DO WHILE .NOT.EOF()
102: SP = LEN (TRIM (AA))
103: MAA = AA + SPAC (SP * 4) + BB + SPAC (20 - SP * 4) + CC
104: SELE B
105: APPE BLANK
106: REPL AA WITH MAA
107: SELE A
108: SKIP
109: ENDDO
110: SELE B
111: GO TOP
112: REPL AA WITH STUF (AA, 1, 8, "主文件名")
113: @8, 20 SAY "结果存于文件 DHIP0.DBF"
114: RETURN

(该程序文件名: ZC22.PRG)

```



技巧三则

北京航空航天大学 90 级研究生 钟卫 吴雨

(一)高级硬盘管理系统 ADM 解锁一法

ADM 是一种比较常用的硬盘管理系统,它管理硬盘的主要特点是:1.将硬盘分成任意个区,用户要回答登录号 ID 和口令,方可使用硬盘的某个区或几个区;2.可以限定用户使用某个分区的权限:无权使用、只可读、可读/写;3.登录号为 0 的用户为超级用户,只有 0 号用户方有权限修改硬盘分区,增删用户,修改用户权限及用户口令。因此,ADM 系统是将大容量硬盘进行小型机管理的良好工具。

一、问题提出

ADM 系统不是完美的,例如对于超级用户,口令就绝对不可以出错,如果超级用户口令记错或遗忘,整个硬盘将无法被修改,而只有使用低级格式化硬盘工具才能重新分配硬盘,但这样将使硬盘数据丢失。

高级硬盘管理系统 ADM 加/解锁源程序清单

```
#include <dos.h>
#include <stdio.h>
#include <string.h>

#define SCREEN_WIDTH 160
#define KEYOFFSET 1009
typedef unsigned char byte;
typedef unsigned int word;

main (word argc, byte * * argv) {
    byte onoff = 2;

    changemode (3);
    dispstr ("Powerful Anti ADM (Tm) V1.00+", 0, 0, 0xb);
    dispstr ("CopyRight [c] By ZhongWei WuYu, (R)
    MACRO LINKer Inc., BUAA, 1990.", 0, 1, 0xc);
    setcursor (0, 4);
    if (argc == 2) {
        if (strcmp(strupr(argv[1]), "/ ON") == 0) onoff = 1;
        else if (strcmp(strupr(argv[1]), "/ OFF") == 0)
            onoff = 0;
    }
    if (onoff == 2) {
        do change (onoff);
        if (onoff == 1) dispstr ("[DONE] HardDisk
        PASSWORD turn ON!", 0, 3, 0xe);
```

```
else dispstr ("[DONE] HardDisk PASSWORD turn
OFF!", 0, 3, 0xe);
    }
    else do error ("[USAGE] ZWADM < / on > < / off > ");
    }

do change (byte c) {
    byte buff[1024];
    word oldes = ES;

    __AH = 0x10;
    __DX = 0xFFFF;
    geninterrupt (0x13);
    __ES = DS;
    __AX = 0x202;
    __BX = (word)buff;
    __CX = 0x1;
    __DX = 0x80;
    geninterrupt (0x13);
    if (AH) do error ("[ERROR] Cannot Read the
    HardDisk!");
    if (buff[KEYOFFSET] != 0 || buff[KEYOFFSET] == 1)
        do error ("[WARN] Not found ADM on the
    HardDisk!");
    buff[KEYOFFSET] = c;
    __AX = 0x302;
    __BX = (word)buff;
    __CX = 0x1;
    __DX = 0x80;
    geninterrupt (0x13);
    if (AH) do error ("[ERROR] Cannot Change the
    HardDisk!");
    __AH = 0x10;
    __DX = 0xFFFF;
    geninterrupt (0x13);
    __ES = oldes;
    }

do error (byte * error message) {
    dispstr (error message, 0, 3, 0xe);
    exit (1);
}
```

```

changemode(int mode){
    __AH=0;
    __AL=mode;
    geninterrupt(0x10);
}

dispstr(byte * str, word Lcol, word Urow, word color){
    register no, col, len = strlen(str);
    byte line[160];

    for (no=col=0; col<len;){
        line[no++] = str[col++];
        line[no++] = color;
    }

    movedata    ( DS,      (word)line,      0xB800,
    Lcol * 2+Urow * 160, len+len);
}

setcursor(word col, word row){
    __AH=2;
    __DL=col;
    __DH=row;
    __BH=0;
    geninterrupt(0x10);
}

```

本程序就可在不知超级用户口令的情况下,对ADM管理的硬盘进行解锁,使所有分区对用户开放,而且可重新修改任意用户口令。

二、实现

ADM管理硬盘的状态有两种:(1)开锁状态,此时任何用户不须输入登录号和口令,即可使用整个硬盘;(2)关锁状态,即用户必须输入自己的登录号及口令,方可使用硬盘,即使从软盘启动机器,也无法使用整个硬盘。当然改变硬盘状态的权利在超级用户。

我们对硬盘两种状态下的绝对第0道分别保存并进行比较,找到了其中不同的一个字节,这个字节就是ADM判断硬盘开/关锁的关键字节,这样,就可以用程序来控制硬盘的开/关锁状态。

三、程序说明

程序中使用了功能号为10H的INT13H中断。因为ADM管理硬盘时,对硬盘绝对0道进行只读保护,从硬盘启动后,不能对绝对第0道进行写操作,而调用10H功能可将硬盘复位,使绝对第0道可写。从这点上来说,ADM管理的硬盘对感染分区表的病毒有免疫功能。

具体的程序见后,使用INT13H的3H功能对绝对第0道的关键字节修改为开锁状态,即达到解锁目的,使整个硬盘对用户开放。

(二) 用 PCTOOLS 对文件或目录

隐藏加密的新方法

目前,常用的对文件/目录名隐藏加密的方法有明显的缺点,即在PCTOOLS等常用高级工具软件中,隐藏后的文件/目录仍会被显示出来,因而起不到隐藏加密的目的。

下面介绍一种新的文件/目录隐藏加密方法,可以克服一般方法的不足,而对PCTOOLS等常用工具软件同样起到隐藏效果。

一、文件/目录名隐藏加密原理

我们知道,每个文件/目录的存贮都对应于目录区的某一文件目录项,该文件目录项的数据结构如下:

字节 0	7	8	10	11	12	15
文件名	扩展名	属性	保留			
保留	时间	日期	起始簇号	文件长度		
字节 16	2122	2324	2526	2728	31	

隐藏加密就是首先将文件目录项的属性字节(第11字节)置为隐藏方式(若是文件,则为27H,若是目录则为17H);其次将文件长度的最高字节(第31字节)置为FFH,这样使其长度大于 10^9 字节,PCTOOLS等自动不显示。值得注意的是,若被隐藏的文件/目录不在根目录下,则须从根目录起逐一计算其子目录的入口簇号,直至找到欲隐藏的文件/目录项所在的盘簇区,然后进行修改。

二、隐藏加密步骤

下面以隐藏C:/BIN/HIDDEN子目录为例,具体说明隐藏加密的全过程(使用PCTOOLS V4.21)。

STEP 1)、已进入PCTOOLS,按<F3>,按E,选择欲隐藏文件/目录所在盘符C;

STEP 2)、已进入DISK EDIT,按<F2>,按R;

STEP 3)、已进入ROOT目录区,找到BIN所在的目录项,并记下第26,27字节的入口簇号,设为A、B(均为十六进制),转换为对应的十进制 $N=B*256+A$;

STEP 4)、按<F2>,按C,输入N的值;

STEP 5)、已进入BIN的子目录区,找到HIDDEN所在的目录项,按<F3>将其第11字节改为17H,将第31字节改为FF,按<F5>;

STEP 6)、已完成,退出PCTOOLS。

这样,该子目录已被隐藏加密,对DIR, COPY, RMDIR命令以及PCTOOLS等工具软件可起到隐藏作用,而其它对该目录及文件的操作却不受影响,因而有较好的加密效果。有兴趣的朋友不妨一试。

(三) 用 C 语言测试 CPU 的运行速度

软件在编制时常常要考虑到该软件的通用性、兼容性。而对不同的机型如XT、286、386、486运行速度相差

很大, 所以许多实时软件在运行中都要考虑使用延时技术, 对不同的 CPU 速度使用不同的延时循环。这样, 首先要测试 CPU 的运行速度。

一、原理

我们知道, PC 系列微机的系统时钟都是由一块通用的定时器芯片 8253 产生, 且标准的系统时钟频率为 18.2HZ。每当一个系统时钟中断到来时, 都要将 0000:046CH 中的内容加 1。简单的测试 CPU 运行速度的方法就是: 记下两次系统时钟中断之间的空循环次数, 再推算出被测 CPU 与标准 PC 机 CPU 运行速度的倍数关系, 以此来确定延时的时间, 附具体程序在后。

二、程序说明

函数 CHANGEMODE() 为改变显示模式

函数 DISPSTR() 为显示字符串函数, 采用对屏幕显示缓存器的直接存取操作方法。

函数 SPEED() 为测试 CPU 速度函数, 返回值为: (被测 CPU 速度 / 标准 PC 机速度) * 100, 传递参数记录两次系统时钟中断之间的循环次数。

用 C 语言测试 CPU 速度的源程序清单

```
#include <mem.h>
#include <dos.h>

typedef unsigned char byte;
typedef unsigned int word;

main() {
    byte str[80], ti = 1;
    long stime = 1;

    changemode(3);
    setcursor(80, 25);
    dispstr("SUPER MACRO SPEED Test (TM) Version 0.89", 0, 0, 7);
    dispstr("CopyRight (c) by ZhongWei WuYu, (R) MACRO LINKer Inc., BUAA, 1990.", 0, 1, 7);
    dispstr("The SPEED of this computer compare with the standard IBM PC (100%) is:", 0, 3, 11);
    dispstr("_____ (_____, 0, 5, 9);

    dispstr("[Wait] Press any key to QUIT.....", 0, 7, 0x0d);
    do {
        sprintf(str, "Test times: %3u SPEED: %4u%% Structures: %8lus", ti++, speed(stime), stime);
        dispstr(str, 12, 5, 0x1e);
    } while (!kbhit() || ti < 255);

    dispstr("[Done] Thank you for using SUPER MACRO SPEED Test.", 0, 7, 10);
```

```
setcursor(0, 8);
}

speed(long * times) {
    unsigned far * begintime = (unsigned far *) 0x46C,
    endtime;
    long count = 0;
    for (endtime = * begintime; endtime == * begintime;);
    for (endtime = * begintime; endtime == * begintime; count++);
    return (100 * (* times - count) / 1991);
}

changemode(word mode) {
    __AH = 0;
    __AL = mode;
    geninterrupt(0x10);
}

dispstr(byte * str, word Lcol, word Urow, word color) {
    register no, col, len = strlen(str);
    byte line[160];

    for (no = 0, col = 0; col < len; col++) {
        line[no++] = str[col++];
        line[no++] = color;
    }

    movedata(DS, (word)line, 0xB800, Lcol * 2 + Urow * 160, len + len);
}

setcursor(word col, word row) {
    __AH = 2;
    __DL = col;
    __DH = row;
    __BH = 0;
    geninterrupt(0x10);
} (该程序文件名: ZC15.PRG)
```

小启

本刊有软件库目录, 欢迎广大爱好者来信索取, 每份收资料费 0.2 元 (用邮票代款)。来信索取请附贴有足够邮票和写有详细地址的回邮信封, 多谢合作。

dBASE IV 的结构化查询语言 SQL 简介

中山大学计算机系 董祖明

dBASE IV 是 dBASE 系列数据库软件的最新产品。它是以 dBASE III plus 为基础, 增加和改进了几百条命令, 增加了全新的用户接口、示例查询 QBE 和结构化查询语言 SQL。本文向读者介绍 dBASE IV 最引人瞩目的新颖之处——结构化查询语言 SQL。

许多著名的关系数据库系统, 如 DB2, SYSTEM R, ORACLE 等都支持 SQL。SQL 业已成为关系数据库查询语言的既成事实的工业标准。dBASE IV 顺应国际新潮流, 增加了功能强大, 使用方便的结构化查询语言 SQL。SQL 是集合级的非过程化语言, 因为用户只要指出“做什么”(WHAT)而不是“怎么做”(HOW), 即用户只要说明想要什么数据, 而并不用规定获得数据的过程。换言之, 查找数据的“导航”过程是由系统自动完成的。正因为如此, SQL 能为用户提供高效率的工作环境。

一. SQL 的基本概念

SQL 的基本概念有数据库, 表, 行, 列等, 与 dBASE 的概念基本对应, 如下表所示。

SQL	dBASE	标准关系术语
=====	=====	=====
数据库	一组相关的文件	一组相关的关系
表	数据库文件 DBF	关系
行	记录	元组
列	域, 字段	属性

表一 SQL, dBASE 和标准关系术语对照

熟悉 dBASE 的读者可用 dBASE 概念理解相应的 SQL 概念。需要特别解释的是 SQL 中的数据库包含一个应用中用户建立的有关各个表和系统为更有效地处理数据而为他们建立的一组目录表。

SQL 命令(或称为语句)由若干称为子句的部分组成, 以分号(;)结束。其中包括有限的一组关键字。有些关键字, 如 SELECT, INSERT, UPDATE 作为命令的开头; 而有些关键字, 如 FROM, WHERE 作为命令中子句的开始。另外一些关键字, 如 DISTINCT, NOT, AND, OR 等, 用说明子句中的可选项或连接子句中的几个子部分。象 SELECT 等命令甚至可以嵌套在其它命令中使用。

SQL 支持 8 种数据类型:

SMALLINT	-99999 至 999999
INTEGER	11 位正整数或 10 位负整数加一位符号
FLOAT(x, y)	0.1E-307 到 0.9E+308。其中 x 为总位数, y 为小数点位置
DECIMAL(x, y)	压缩十进制数, x 位数字

NUMERIC(x, y)	($0 < x < 20$), y 位小数 同上, 但 x 包括小数点位置
CHAR(n)	长为 n ($n < 255$) 的字符串
LOGICAL	逻辑值 .T. 或 .Y. 和 .F. 或 .N.
DATE	日期值 {mm/dd/yy}

下面介绍如何用 SQL 命令定义, 存贮, 检索和修改信息。

二. dBASE IV SQL 命令的用法

下面通过众所周知的供应商-零件数据库的例子来说明如何用 SQL 命令定义, 存贮, 检索和修改信息。

供应商-零件数据库由由供应商表 S, 零件表 P 和供货表 SP 三个表组成。表 S 包括供应商号 S#, 供应商名 SNAME, 供应商状态 STATUS 及供应商地址 CITY; 表 P 由零件号 P#, 零件名 PNAME, 颜色 COLOR, 重量 WEIGHT 和存放地址 CITY 组成。表 SP 有供应商号 S#, 零件号 P# 和供货数量 QTY 三列。

1. dBASE IV SQL 的启动与退出

dBASE IV 提供 dBASE 和 SQL 两种命令接口。在 dBASE 的点提示下打入命令

SET SQL ON

即进入 SQL 状态, 提示符变为“SQL.”。也可在 CONFIG.DB 中设置 SQL=ON, 则进入 dBASE 时直接进入 SQL 方式。退出 SQL 状态并返回 dBASE 状态的命令是

SET SQL OFF

SQL 为了更好地管理数据, 把一组相关的表置于一个数据库中, 必须先为一个应用创建数据库:

CREATE DATABASE spdb;

要在数据库 SPDB 中工作还必须打开数据库:

START DATABASE spdb;

现在就可在 SPDB 中使用各种数据定义和数据操纵的命令了。

退出时必须先关闭当前的 SQL 数据库:

STOP DATABASE;

删除一个废弃的数据库的命令是:

DROP DATABASE spdb;

2. SQL 数据定义命令

CREATE 命令用于定义基本表, 索引和视图(导出表)。

2.1 基本表的定义和删除

用 CREATE TABLE 命令定义表 S, 表 P 和表 SP。

CREATE TABLE s

(s# CHAR(5), sname CHAR(20), status

SMALLINT, city CHAR(15));

CREATE TABLE p

(p# CHAR(6), pname CHAR(20), color
CHAR(6), weight SMALLINT, city CHAR(15));

CREATE TABLE sp

(s# CHAR(5), p# CHAR(6), QTY INTEGER);

dBASE IV 的 SQL 不支持 ANSI SQL 中列 NOT
NULL 的概念。

用 ALTER TABLE 命令可在一个现有的表中增加一
列

ALTER TABLE s ADD discount SMALLINT;

撤消一个基本表的命令是

DROP TABLE 表名;

被撤消的基本表上的所有索引和视图也自动地撤消。

2.2 索引的建立与撤消

在供应商表 S 的供应商号 S# 上建立索引 XS

CREATE UNIQUE INDEX xs ON s (s#);

同样, 在表 P 和表 SP 上建立索引 XP 和 XSP

CREATE UNIQUE INDEX xp ON p (p# DESC);

CREATE UNIQUE INDEX xsp ON sp (s#, p#);

任选 UNIQUE 规定不允许索引基本表中有两条记录
录的索引字段(或字段组合)同时取相同值。索引字段后可
加上 DESC 表示降序排列(省缺值为升序 ASC)。

撤消索引的命令为

DROP INDEX 索引名;

2.3 视图的定义与撤消

SQL 中的视图是从基本表(实表)导出的虚表, 它没有
自己独立的物理存贮。假设要建立基本表 S 的一个视图,
它由 S 中满足 STATUS > 15 的行组成

CREATE VIEW good_suppliers

AS SELECT s#, status, city

FROM s

WHERE status > 15;

视图 good_suppliers 实际上是基本表 S 的一个“窗口”
。它是动态的: 基本表 S 的变化将自动地同时在窗口上反
映出来。用户对表和视图的使用是基本相同的, 用户甚至
不必知道表 S 就可象使用表一样使用视图
good_suppliers。

视图的用途之一是提供了逻辑数据独立性, 当数据库
发生增长和调整时, 通过视图可保持总体信息不变。另
外, 视图允许不同用户同时以不同方式看待相同的数据。
而且视图也使用户的理解简化了。再有就是视图为隐藏
的数据提供了自动安全保护。

撤消视图的命令为

DROP VIEW 视图名;

3. SQL 的数据操纵命令

SQL 的数据操纵命令有数据检索命令 SELECT, 数据

修改命令 INSERT, DELETE 和 UPDATE。

3.1 数据检索命令 SELECT

假设表 S, P, SP 中已经存有一部分数据。

SELECT 命令包括 SELECT, FROM, WHERE 等
子句。SELECT 子句指出要“投影”的列, FROM 子句给
出一个表或多个表的“连接”, 而 WHERE 子句表明“选择”
条件。

列出表 S 中的所有行:

SELECT * FROM s;

命令中的 * 是表 S 中所有列名的缩写。

列出居住在巴黎且状态 > 20 的供应商号及状态:

SELECT s#, status

FROM s

WHERE city = 'Paris' AND status > 20;

列出供应的所有零件的零件号并消去多余的重复号:

SELECT DISTINCT p# FROM sp;

稍为复杂的查询包含了关系系统的“连接”操作。查询
处于同一城市的供应商和零件的全部信息组合:

SELECT s.*, p.*

FROM s, p

WHERE s.city = p.city;

同名字段必须用相应的表名区别。

最后简单介绍 SQL 的子查询或嵌套查询。子查询一般
用来表示谓词 IN 和 EXIST 要检索的值的集合。如, 列出
供应零件 P2 的供应商名:

SELECT sname

FROM s

WHERE s# IN

(SELECT s#

FROM SP

WHERE p# = 'P2');

3.2 插入命令 INSERT

最简单的插入是往表中插入单个记录。例如, 将零件
P7 加进表 P 中:

INSERT INTO P (p#, pname, color, city, weight)

VALUE ('P7', 'IC80386', 'Black', 'GuangZhou', 2);

也可把 INSERT 命令的 VALUE 子句换成一个子查询,
从一个表中选出一批记录插入到另一个表中:

CREATE TABLE guangzhou_s

(s# CHAR(5), city CHAR(15));

INSERT INTO guangzhou_s

SELECT s#, city

FROM s

WHERE s.city = 'Guangzhou';

此例先建立包含供应商号 s# 和城市 city 的“广州”供应商表
guangzhou_s, 然后从表 S 中选出满足 city = 'Guangzhou'
的记录 s# 和 city 字段加进 guangzhou_s 表中。

3.3 删除命令 DELETE

删除表中所有记录的命令是:

中华学习机 CEC—I 驱动器维修二例

广东中山市第一中学 陈华昌

故障一：开机后驱动器指示灯亮，但不能读和写。

检修方法：先用磁盘驱动器清洗盘将驱动器清洗一次，若故障解决，则为磁盘驱动器磁头脏所致。若故障仍然存在，此时应检查一下磁盘驱动器本身有无故障，最简单的方法是用一台好的驱动器来替换试一试。若驱动器本身无故障，则先检查主机中固化磁盘驱动器引导程序的 ROM (U40) 是否损坏或接触不良。若无问题，再检查控制逻辑 U41 (74LS259) 和时基电路 U45 (NE555) 工作是否正常，主要测量有无开启主机信号 (测 U41 第 10 脚，U45 第 3 脚) 和控制调制解调电路工作方式的信号 (测 U41 第 11、12 脚)。若这些元件没有问题，则检测读数据电路 U39、U42 工作是否正常，若其中一个损坏，磁盘驱动器也不能够读写。

故障二：开机后，磁盘驱动器指示灯亮，但不能引导 DOS，听到“滴嗒、滴嗒...”声。

检修方法：换上另一台好的驱动器，开机，故障消失，启动正常。说明是驱动器故障，打开驱动器机盖，用

手正、反转动步进电机的驱动轴，磁头装置可以平滑地前进和后退，这就说明步进电机的驱动机械没有故障。再检查驱动器主控制线路板上的焊点有无虚焊点，电阻、电容等分立元件有无明显的损坏。经检查，发现在连接线旁边的一只 0.22μF 涤纶电容烧焦，更换此电容。开机，听不到“滴嗒、滴嗒...”声，但还是不能引导 DOS，这就说明主控制线路上出故障。由于主控制线路板上的集成电路个数不多，所以先检查集成电路，而集成电路又固定焊在底板上，不可能将全部集成块焊下来逐个进行测试，所以可采用静态电阻测试比较法来判断哪块集成电路有故障，具体做法是：将有故障的主控制线路板上的相应集成块的静态阻值与另一块好的主控制线路板上的相应集成块的静态阻值进行比较，看看是否相同，以确定集成块是否好坏。对比结果发现原主控制线路板上的 74LS08 有些脚的静态阻值与正常的静态阻值相差较大，所以断定这集成块有故障。换上一块好的 74LS08，开机，故障消失。 ■

DELETE FROM 表名;

删除表中满足一定条件的一些记录可用以下命令。例如，删除住在广州的状态 < 20 的供应商：

DELETE

FROM S

WHERE city = 'Guangzhou' AND status < 20;

涉及到多个表的删除可利用子查询。如将居住在广州的供应商的 SP 记录删除，试用以下命令：

DELETE

FROM sp

WHERE 'Guangzhou' =

(SELECT city FROM s WHERE s.s# = sp.s#);

3.4 更改命令 UPDATE

更改单行记录应在 WHERE 子句中指定关键字段的值。下面的命令把零件 P2 的颜色改为黄色，重量加 5：

UPDATE p

SET color = 'Yellow', weight = weight + 5

WHERE p# = 'P2';

更改多行记录的语句同样简单。把广州的供应商状态增加一倍。

UPDATE s

SET status = 2 * status

WHERE city = 'Guangzhou';

上述修改命令 INSERT, DELETE, UPDATE 在涉及到数据完整性时必须小心，象大多数别的关系系统一样，系统并不维护完整性，此重任由用户承担。

4. 其它命令

SQL 的命令使用简单但功能强大。除了以上命令外，还包括有关安全性和授权的 GRANT 和 REVOKE 命令，有关事务处理的 COMMIT 和 ROLLBACK 命令以及其它实用命令。SQL 命令中可以使用 SUM, AVG, COUNT 等系统函数。另处部分 dBASE IV 的命令和函数也可用在 dBASE IV 的 SQL 中。

本文的供应商-零件数据库的例子简单地说明了 SQL 的大部分操作。实际上，SQL 在构造更复杂的查询时显得更为有用和强大。

dBASE IV 的 SQL 实现了 ANSI SQL 标准的大部分条文。由于微型计算机本身资源的限制实现的策略，dBASE IV 的 SQL 相比于它的 DBASE 命令在效率上略逊一筹。dBASE IV 对硬件的要求较 ORACLE 低，同时 dBASE 数据库拥有广大的用户，这将使 SQL 在微型计算机上流行成为可能。而且大量在 SQL 在开发的数据库应用软件也移植到微机中。

SQL 凭着本身令人无法抗拒的魅力，很快会赢得广大微机数据库用户的心。 ■

对《硬盘不能启动的软维修方法》一文的补充

山东建筑工程学院计算中心 刘益林

贵刊 1991 年第二期徐云彪同志所写《硬盘不能启动的软维修方法》一文非常实用, 该文从主引导扇区的角度介绍了如何通过覆盖主引导扇区来修复不能启动的硬盘, 但硬盘不能启动并不仅与主引导扇区有关, 因此, 对该文做以下补充。

硬盘不能启动, 从逻辑性方面来说主要与以下因素有关:

1. 系统设置有误 (因误击键或其它原因使系统的实际配置与设置不符)。
2. 主引导扇区遭破坏。
3. DOS 的引导记录遭破坏。
4. DOS 系统的两个隐含文件 IBMBIO.COM 及 IBMDOS.COM 遭破坏, 丢失或不在正确的位置。
5. COMMAND.COM 文件丢失或版本不对。

对第一个因素只要检查一下系统设置并予修正就可 (注: 该因素只与 286 机及以上型号机器有关)。

对于第二个因素可按徐云彪同志所介绍的方法即覆盖主引导扇区恢复硬盘。

主引导扇区位于物理 1 扇区, 由主引导程序和分区表两部分组成, 主引导程序负责将活动分区引导程序调入内存并赋予控制权, 分区表说明硬盘划分的情况, 由 FDISK 命令形成, 负责为硬盘在内存建立基本的 I/O 参数表。

需要说明的是, 大部分的系统不识硬盘的情况 (即: 从硬盘不启动系统, 而从软盘启动后又不能切换到硬盘, 或虽能切换到硬盘却不能读写硬盘数据) 主要与硬盘主引导扇区的分区表有关, 无论从硬盘或软盘引导系统, 总要将硬盘分区表的有关信息读入内存, 以便为硬盘在内存建立一张基本 I/O 参数表。因此, 如硬盘分区表无活动分区标志 (即四个可能分区的首字节均为 00H), 或无 "55AA" 分区自举有效标志, 或虽有活动分区 (其首字节为 80H), 但该分区有关的信息 (如: 分区的起始及结束地址) 却不对等, 都会使 I/O 表无法建立而造成系统不识硬盘。解决的方法是: 取一同型机器的主引导扇区覆盖之, 或直接修改分区表。

对于第三方面的原因也可通过覆盖法得以解决。

DOS 的引导扇区位于逻辑 1 扇区, 由 FORMAT 命令形成, 主要有四部分组成: JMP 代码, 版本商号, BPB 表及 DOS 引导程序, 其中 13 字节的 BPB 表及后续的三个字节记录了硬盘的有关重要信息, DOS 引导程序负责将 DOS 的系统文件 IBMBIO.COM, IBMDOS.COM 调入内存, 因此, 如该 DOS 引导记录遭破坏也会造成硬盘不能启动。修改的基本方法是: 取一好的 DOS 引导记录写入

欲修复硬盘的 DOS 引导扇区即可 (DOS 版本及硬盘型号应相同, 注: 也可用 FORMAT C: /S 重新格式化硬盘, 但会丢失硬盘数据)。

对于第四方面的原因一般说用 DOS 的 SYS 命令重新传输系统即可, 但有时传输并不成功或虽传输成功但系统仍不能启动, 原因何在呢?

DOS 从硬盘自举时, 其引导程序要从硬盘根目录的第一登入项 (簇号 0002) 及第二登入项寻找 IBMBIO.COM 及 IBMDOS.COM, 如找到则调入内存, 否则显示:

NON-SYSTEM DISK OR DISK ERROR

REPLACE AND STRIKE ANY KEY WHEN READY

并处于循环等待状态, 因此, 该两个系统文件必须占领根目录的首项及第二项, 并要占用连续簇, 如该两位置被其它文件侵占也会造成系统无法从硬盘启动。因此若无法用 SYS 传输系统或虽传输成功却仍不能从硬盘启动, 则应查看根目录表首项及第二项是否已被其它文件占领, 若是则只要交该文件删掉再传输系统即可。查看根目录表的方法也十分简单, 用 PCTOOLS 即可实现, 进入 PCTOOLS 后选 F3 进入磁盘特殊服务菜单, 再敲 E (EDIT 项) 选编辑功能, 机器询问盘符时键入 C, 则首先调入内存并显示在屏幕上的是 DOS 引导扇区, 再敲 F2 则屏幕上出现一小菜单, 敲 R 即将文件目录表的首扇区显示在屏幕上, 查看 ASCII 码栏即知那些文件占据了根目录表的首项及第二项 (注: 每个目录项占 32 个字节, 记录了一个文件的有关信息, 其中第 26, 27 字节说明了该文件的起始簇号), 予以删除即可。

对于第五方面的原因, 用 COPY 命令重新拷入 COMMAND.COM 文件即可。

如用户对造成硬盘不能启动的原因一时搞不清楚, 可对以上五个因素逐一检查, 确认后再修改, 有关查看及覆盖主引导目录及 DOS 引导高录的方法已有许多文章作过介绍, 这里就不再多说了。 ■

小广告

黑龙江阿城市继电四号信箱阿科信协计算机部向您赠各类 APPLE, CEC 软件, 软盘双面录制 4.80 元, 每次邮包费 3 元。并可交换软件, 详情附 4 角邮资即寄。

计算机考试评价分析

南京军区军医学校 张鲁闽 魏江平

提要：考试评价的定量分析是深入教学改革的重要环节。考试评价的定量分析主要有考试成绩和考题质量两个方面，本文介绍了其定量计算的公式，介绍了其在计算机上的实现方法和软件的用法。

随着教育事业的发展，考试作为评价教育质量的方法之一，已成了教学过程中不可缺少的重要环节。而对考试的评价，又为学校的教学改革，教员教学方法和学生学习方法的改进，考试题库的择优建立，进一步提高教学质量提供反馈信息。考试评价主要有考试成绩和考题质量两个方面，在这两个方面的评价中，采用了许多统计学的量化标准，使得考试的评价定量化，科学化。但是，由于这些统计学的定量指标需要处理大量的考试数据，需要进行大量的计算，以致对考试的评估往往不能及时进行或不得不放弃。为此，应用计算机能快速地进行数据处理的功能，我们研制了计算机考试评价分析系统。该系统能使不懂得计算机的人员方便地进行数据采集，数据处理，并打印出各种分析数据，形成报表。该软件对考试的评价具有实用价值。以下，我们对该系统做进一步的介绍。

一、考试评价分析的方法

(一) 考试成绩定量分析的计算

考试成绩进行下列项目的统计处理：

1、统计成绩各分数段的人数及占考试人数的百分比。自动选出成绩最高分和最低分。

2、绘制成绩分布直方图。直方图能直观地反映考生考试成绩的分布情况。

3、计算平均分 \bar{X} 。平均分反映了考生的平均水平。

计算式如下：

$$\bar{X} = \frac{1}{n} \sum_{i=1}^n X_i$$

X_i ：为第*i*名考生分数。

n ：参加考试的学生数。

4、计算标准差 S 。标准差反映考生成绩的离散程度。系统设定标准差的正常值域为 $8 \leq S \leq 12$ 。标准差的计算式为：

$$S = \sqrt{\frac{\sum_{i=1}^n (X_i - \bar{X})^2}{n-1}}$$

5、计算变异系数 CV 。变异系数也反映了考生成绩的离散程度，且可用于不同班次成绩离散程度的比较。其计算式为：

$$CV = \frac{S}{\bar{X}} \times 100\%$$

6、计算总评成绩 Q 。总评成绩是和平均成绩、标准差有关的经修定后的平均成绩，其计算式为：

$$Q = \bar{X}(e^{\frac{100S}{\bar{X}} - 11} - 0.8) \cdot (\frac{100S}{\bar{X}} - 13.7)$$

7、计算偏度系数（不对称系数） P 。偏度系数是对一次考试正态性检验的标志。如果出现明显负偏，则分数贬值；如果出现正偏，说明考试过难。系统设定标准正态分布时，偏度系数值域为 $-0.2 \leq P \leq 0.2$ ；负偏态分布时，偏度系数值域为 $P < -0.2$ ；正偏态分布时，偏度系数值域为 $P > 0.2$ 。偏度系数的计算式为：

$$P = \frac{\sum_{i=1}^n (X_i - \bar{X})^3}{nS^3}$$

8、计算峰度系数 F 。峰度系数也是对一次考试正态性检验的标志。设定标准峰态分布值域为： $2.2 \leq F \leq 3.8$ ；低峰态值域为 $F < 2.2$ ；尖峰态值域为 $F > 3.8$ 。峰度系数的计算式为：

$$F = \frac{\sum_{i=1}^n (X_i - \bar{X})^4}{nS^4}$$

9、计算单项潜力系数 K 。该系数反映了学生可以挖掘的学习能力，其计算式为：

$$K = \frac{M - V}{M} \times 100\%$$

M ：某次考试满分。

V ：某次考试实得分。

10、计算信度指数 R 。信度指数 R 反映了该次考试的可信性。系统中选用的计算公式为Kuderrichardson公式：

$$R = \frac{K}{K-1} \left(\frac{S^2 - \sum_{i=1}^K P_i Q_i}{S^2} \right)$$

K ：考题数。

P_i ：第*i*题的难度指数。

Q_i ：误答率 $Q_i = 1 - P_i$

信度指数要求大于0.7。

(二)、考题质量定量分析的计算

考题质量分析进行下列项目统计：

1、计算考题的难度指数 P 。难度指数反映了考生正确回答某一题的比率，其计算式为：

$$P = \frac{H + L}{N \times F}$$

H: 按成绩高低排列的前30%学生(高分组)该题的得分和。

L: 按成绩高低排列的后30%学生(低分组)该题的得分和。

N: 高低分两组学生的总人数。

F: 该题的满分。

2. 计算考题的区别指数 PR, 该指数反映了考题区别优劣考生的能力。其计算式为:

$$PR = \frac{H-L}{N \times F} \times 2$$

3. 考题按难度指数分类。分类依据设定为表-1。题目按难度指数分类的依据仅供分析者参考。

表-1 题目按难度指数分类表

难度指数	题目类型
$P > 0.85$	容易题
$0.5 \leq P \leq 0.85$	一般题
$0.2 \leq P \leq 0.49$	较难题
$P < 0.2$	难题

4. 考题按区别指数分类。分类依据设定为表-2。题目按区别指数分类的依据仅供参考。

表-2 题目按区别指数分类表

区别指数	题目类型
$R > 0.4$	好题
$0.3 \leq R \leq 0.4$	较好题
$0.15 \leq R \leq 0.3$	可用题
$R < 0.15$	慎用题

上述考试评价的计算公式是我们参考许多计算方法后选用的, 这些选用的计算方法是得到教育界的承认的。

三、系统的设计

(一) 硬件配置

本系统适应在 IBM-PC/XT 或其兼容机上运行。系统要求如下硬件配置: IBM-PC/XT (或 PC 机) 一台, 标准键盘一台, 监视器一台, 一台能与 IBM-PC 系列机适配的打印机。

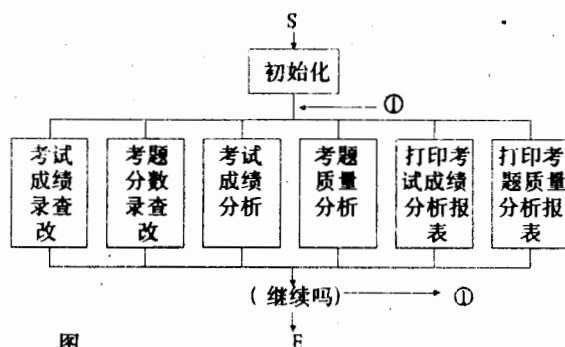
(二) 软件配置

CCDOS2.0 以上版本的磁盘操作系统, CDBASE-III 1.0 以上版本, BASICA 和与打印机适配的驱动程序。

(三) 系统功能

在 DBASE-III 数据库的基础上, 该软件能对要分析的考试成绩和考题分数进行录入、修改、查询, 并自动生成变长记录库文件。在数据录入后, 能自动进行考试成绩和考题质量的分析, 并存贮分析后的数据。能根据需要打印考试成绩和考题质量分析报表。其功能框图如下图:

在该系统的设计中, 考虑到不同考试的人数和题数是不同的, 在数据库文件的组织上, 我们采取了变长记录的方法, 从而大大地节省了外存贮空间, 这是该系统的特点之一。



图

三、系统的使用

1. 系统的安装:

该软件由含 CCDOS 的启动盘、DBASE 盘和系统程序盘所构成。在含有硬盘的 IBM-PC 机及其兼容机上, 系统使用前, 应先在硬盘上建立名为 KS 的子目录, 进入 KS 子目录后, 把 DBASE 盘和系统程序盘中的所有文件拷入子目录中, 软件即安装完毕。以后只需在 A 驱动器中插入启动盘, 冷启动或热启动, 即可自动进入考试评价分析系统。在只有双软盘驱动器的 PC 机上, 在 A 驱动器上插入启动盘, 冷启动或热启动后, 只需按屏幕提示进行操作即可。

2. 系统的使用

该软件使用前, 必须进行数据的收集准备。考试成绩分析要准备的数据主要有: 考试人数, 各学员考分, 考试科目名称。考题质量分析要准备的数据主要有: 考题数量, 考试成绩按高分到低分排列后前 30% 的学生 (高分组) 和后 30% 的学生 (低分组) 各考题的实际得分, 各考题的满分值。

软件使用时, 当启动系统进入主菜单时, 首先应选择考试成绩录查改 (功能 1) 和考题分数录查改 (功能 2), 把需处理的数据正确地采集完毕。再选择考试成绩分析 (功能 3) 和考题质量分析 (功能 4), 进行数据的统计计算, 数据经过上述处理后, 方可选择打印考试成绩分析报表 (功能 5) 和打印考题质量分析报表 (功能 6)。由于每次考试数据分析后的结果是存在磁盘上的, 故报表打印功能可选用多次, 以便打印多份报表。

3. 使用注意事项

(1)、使用时应注意根据屏幕菜单和提示进行操作。

(2)、对某次考试进行分析时, 应给该考试一个编号。如 "860HL" 表示 86 级 1 班护理考试, 编号的前 2 个字符表示年级, 第 3、4 个字符表示班次, 第 5、6 字符表示考试科目。编号长度不能超过 7 个字符。

(3)、功能 5 和功能 6 的选用必须在功能 1、2、3、4 进行后才可进行。

(4)、系统运行时, 遇到故障, 可以重新启动系统, 重新运行。应用电脑技术进行考试评价分析, 是教育现代化的一种手段。考试分析中所计算出来的一些统计学指标, 能及时反馈考试中的信息, 为教学改革提供参考数据。但

谈 BASIC 的特技清屏

大连商业学校 侯胜

在程序编制过程中,尤其对于趣味性程序较浓的程序,屏幕画面的清除方式是非常重要的,每个设计者都希望有个新颖的色彩斑斓的清屏方式。

CLS 指令所产生的清屏效果使屏幕上的画面在瞬间突然消失,实在是缺乏艺术性。如果我们用 LINE 语句,并配上颜色来设计清屏方式,就可在屏幕上产生犹如舞台上幕的徐徐拉开,关闭,升起,放下的效果。本文所提供的三个程序中,均有实现上述效果的方法。程序 1-2 中的 20-40 语句可实现从右向左徐徐闭幕,若要改变幕的颜色可更改 30 语句中的颜色参数。程序 2-2 中的 20-50 语句可实现从两端向中间徐徐闭幕的效果。程序 3-2 中的 20-110 语句可实现从屏幕的左上角斜着向下角徐徐清屏的效果。由于屏幕不是正方形,因此用 3 个循环分成三部分才能使 LINE 语句始终按 45 度角方向清屏。在第一个循环中,对角线端点坐标为 0-200,在第二个循环中,对角线右端点的横坐标从 201-320,纵坐标始终为 0,在第三个循环中对角线右端点的横坐标始终为 320,纵坐标为 0-200,不难看出,对这三部分程序略加改动即可实现从左向右,从上向下,从下向上,从右下角向左上角清屏。

为了实现一个画面被清除另一个画面被显示出来,可以采用图形文件。但是,采用图形文件前一个画面是在瞬间被清除掉的。如何实现一个画面被徐徐清除的同时另一个画面又徐徐的显示出来呢?笔者通过研究找到了一个切实可行的办法。即将背景画面在屏幕上的点坐标,按徐徐显示的方面和顺序存入一个随机文件。如果在徐徐清屏的同时读随机文件的点坐标,并按读取的点坐标值画点,即可实现背景画面随着徐徐清屏而徐徐的显示出来。

现对笔者提供的三个程序做如下说明:

是,这一系统的提出只是初步的。我们希望,通过交流,使这一系统进一步完善,使对考试的评价做的更准确,更科学。

参考文献

- 《考试学引论》辽宁人民出版社 1987 年版
- 《教学质量与教学效果的评价方法》卫生部科教司 1989 年版
- 《汉字 DBASE III 数据库管理系统及其应用》炮兵技术学院 1986 年版
- 《DBASE (II III) 程序设计应用指南》中科院计算所 17 室 1986 年版
- 《教育研究》1983 第 2 期

程序 1-1 是前期制作过程,即制作的随机文件。20 和 30 语句规定背景汉字,用户可按自己的需要更换。20 和 30 语句中的汉字是不能直接用做背景的,必须经放大,并画在屏幕中用户要求的位置上,才可做为背景画面。40 语句中的自定义函数决定了汉字的放大倍数和在屏幕上所处的位置。修改自定义函数中的常量可以变汉字的大小及在屏幕上的位置。80 语句是取下 20 和 30 语句在屏幕左上角显示的小汉字的点坐标,100 语句将取得的点坐标经自定义函数转换成该点在放大后的汉字,即真正的背景中的坐标(放大后的点已变成小矩形),110 语句将该点(小矩形)的点坐标写入随机文件,120 语句显示该点(小矩形),130 语句将屏幕左上角小汉字已取过坐标的点擦去。60 和 70 语句中的循环变量 X 和 Y 的终值,取决于 20 和 30 语句所用的屏幕的大小。150 语句是表示一系列取点坐标结束,为程序 1-2 中的 120 语句准备条件。

程序 1-2 为应用部分,执行这部分程序由 20-40 语句产生由右向左徐徐闭幕的效果,50-70 语句又使幕由左向右徐徐拉开。当拉倒要同时显示背景的是方时,执行 80 语句打开由前期制做程序制做好的名为 CU1 的随机文件,100 语句将幕拉开一点后,马上由 110 语句到随机文件中读取背景中点(小矩形)的坐标,再由 130 语句按取的坐标画点(小点形),如此反复进行即可实现边拉幕边显示背景的效果。170-190 语句将幕继续向右拉。

程序 2-1 和 2-2 的作用是将幕从两侧向中间徐徐关闭,然后从中间向两端徐徐拉开,同时徐徐显示背景。

程序 3-1 和 3-2 的作用是将幕从左上角向下角方向关闭,再从左上角向下角拉开,同时显示背景。在程序 3-1 的前期制做部分中,从 60 语句开始取点坐标,它不同于程序 1-1 和程序 2-1 的取点方式,前二个程序都是自上而下呈直线取点,而程序 3-1 是自上而下向左呈斜线取点。由于屏幕左上角两行小汉字的最大点纵坐标为 33,70 语句可控制当横坐标 X 大于 34 时,所用的最大纵坐标为 33。执行部分的原理,与前二个程序相同,不再重复。

改变前期制做部分中取点坐标的顺序和方向,即可得到不同的背景显示顺序,再配上相应的清屏方式,即可得到其它种特技清屏。有兴趣的读者不妨一试。

笔者所提供的程序均在 IBM PC/XT 机上通过。

```
5 REM 程序 1-1
10 SCREEN 1:COLOR 12,4:KEY OFF:CLS
20 LOCATE 1,1:PRINT "大连商业学校"
30 LOCATE 2,1:PRINT "微机小组设计"
40 DEF FN X=X*2+32:DEF FN Y=69+2*Y-1
50 OPEN "cul.dat" FOR OUTPUT AS #1
60 FOR X=0 TO 128
```

```

70 FOR Y=0 TO 33
80 D=POINT(X,Y)
90 IF D=0 THEN 140
100 A=FNX:B=FNH
110 WRITE#1,A,B
120 LINE(A,B)-(A+1,B+2),3,BF
130 PRESET(X,Y)
140 NEXT Y
150 WRITE#1,0,0
160 NEXT X
170 CLOSE

```

```

5 REM 程序 1-2
10 SCREEN 1:COLOR 17,4
20 FOR I=320 TO 4 STEP -4
30 LINE(I,0)-(I-4,200),2,BF
40 NEXT I
50 FOR I=0 TO 30 STEP 2
60 LINE(I,0)-(I+2,200),0,BF
70 NEXT I
80 OPEN "cul.dat" FOR INPUT AS #1
90 FOR I=32 TO 288 STEP 2
100 LINE(I,0)-(I+2,200),0,BF
110 INPUT#1,A,B
120 IF A=0 THEN 150
130 LINE(A,B)-(A+1,B+2),3,BF
140 GOTO 110
150 NEXT I
160 CLOSE
170 FOR I=288 TO 320 STEP 2
180 LINE(I,0)-(I+2,200),0,BF
190 NEXT I

```

```

5 REM 程序 2-1
10 CLS:SCREEN 1:COLOR 12,4:KEY OFF:H=0
20 LOCATE 1,1:PRINT "大连商业学校"
30 LOCATE 2,1:PRINT "微机小组设计"
40 DEF FNY=69+2 * Y-1:DEF FNH=H * 2+32
50 OPEN "cx2.dat" FOR OUTPUT AS #1
60 FOR X=64 TO 0 STEP -1
70 H=X
80 GOSUB 200
90 H=64+65-X
100 GOSUB 200
110 NEXT X
120 CLOSE
130 END
200 FOR Y=0 TO 33
210 D=POINT(H,Y)
220 IF D=0 THEN 270
230 A=FNH:B=FNH
240 LINE(A,B)-(A+1,B+2),3,BF
250 WRITE#1,A,B
260 PRESET(H,Y)
270 NEXT Y
280 WRITE#1,0,0
290 RETURN

```

```

5 REM 程序 2-2
10 SCREEN 1:COLOR 17,4
20 FOR I=0 TO 160 STEP 2
30 LINE(I,0)-(I+2,200),2,BF
40 LINE(320-I,0)-(320-I-2,200),2,BF
50 NEXT I
60 OPEN "cx2.dat" FOR INPUT AS #1
70 FOR I=0 TO 128 STEP 2
80 LINE(161-I,0)-(161-I-2,200),0,BF
90 INPUT#1,A,B
100 IF A=0 THEN 130
110 LINE(A,B)-(A+1,B+2),3,BF
120 GOTO 90
130 LINE(162+I,0)-(162+I+2,200),0,BF
140 INPUT#1,A,B
150 IF A=0 THEN 180
160 LINE(A,B)-(A+1,B+2),3,BF
170 GOTO 140
180 NEXT I
190 CLOSE
200 FOR I=0 TO 32 STEP 2
210 LINE(32-I,0)-(32-I-2,200),0,BF
220 LINE(286+I,0)-(286+I+2,200),0,BF
230 NEXT I
240 END

```

```

5 REM 程序 3-1
10 SCREEN 1:COLOR 12,4:KEY OFF
20 LOCATE 1,1:PRINT "大连商业学校"
30 LOCATE 2,1:PRINT "微机小组设计"
40 DEF FNA=A * 2+32:DEF FNY=69+2 * Y-1
50 OPEN "tu-4.dat" FOR OUTPUT AS #1
60 FOR X=0 TO 160
70 IF X>34 THEN A2=33 ELSE A2=X
80 FOR Y=A2 TO 0 STEP -1
90 D=POINT(X-Y,Y)
100 IF D=0 THEN 150
110 A=X-Y:A=FNA:B=FNY
120 WRITE#1,A,B
130 LINE(A,B)-(A+1,B+2),3,BF
140 PRESET(X-Y,Y)
150 NEXT Y
160 WRITE#1,0,0
170 NEXT X
180 CLOSE

```

```

5 REM 程序 3-2
10 SCREEN 1:COLOR 17,4
20 FOR I=0 TO 200
30 LINE(I,0)-(0,I),2
40 NEXT I
50 FOR I=201 TO 320
60 LINE(I,0)-(I-200,200),2
70 NEXT I
80 FOR I=200 TO 0 STEP -1
90 LINE(320,200-I)-(320-I,200),2
100 NEXT I

```


LQ-1600K PRINTER 的压缩打印

甘肃庆阳长庆一中 任绥海

LQ-1600K 型打印机【FS S n1 n2】命令中, n1 和 n2 分别为字符的左间距和右间距。中文字符的右间距默认为 3, 可以通过置零来压缩。应该注意的是这里的零不是“0”这个数字, 而是 ASCII 代码表中的第一个值为“0”的字符, 即 CHR\$(0)。这个字符在键盘上找不到, 屏幕上也无法显示, 所以在编辑软件中是无法实现的。只能用高级语言中的转换函数来取得。如在 BASIC 中, 我们可以编写程序, 利用文件操作功能来实现文本文件的压缩打印。初始化打印机时, 用如下命令:

```
LPRINT CHR$(27)+"+"+"0": '行距
LPRINT CHR$(28)+"S"+CHR$(0)
+CHR$(0): '列间距
```

王玉中老师在《软件报》90 年 8 月 18 日发表的小汇编程序, 把【FS S 0 0】用低级语言的形式送给了打印机。但他疏忽了有些编辑软件每次打印时, 都要初始化打印机, 如果控制码不能同打印内容在同一次打印中送出, 本命命令将失去作用, 字符间距仍然是 3。

当然, 如果你会使用动态调试程序 DEBUG, 也可以将 CHR\$(0) 输入到要打印的文本文件中。具体方法是: 在编辑时把 CHR\$(0) 先用两个空格符或其他可显示字符代替, 进入 DEBUG 后将其 16 进制码转换成“00 00”即可。

另外, 我们还发现, 用 ASCII 代码表中第一个可显示字符也就是 CHR\$(1), 可以勉强代替 CHR\$(0)。它比 CHR\$(0) 大了一个点, 但比 CHR\$(3) 小了两个点, 接近于实践。

CHR\$(1) 可以在键盘上按 CTRL A 得到, 也可以通过按 ALT 1 来获得, FS 则可以通过按 ALT 28 得到。屏幕上显示时应该是这样的:

这种方法用于控制行距时, 压缩效果更为明显。如能与“ESC 3”合用, 还会使行距小于 0, 从而实现重叠打印。

以下是几种打印效果的对比:

行距列距用默认值

官	知	止	兮	神	歌	行	肉	眼	闭	兮	心	眼	开
务	清	策	者	无	愿	事	无	远	虑	者	育	近	忧

行距用 ESC+O 控制, 列距用 CHR\$(0)控制

官	知	止	兮	神	歌	行	肉	眼	闭	兮	心	眼	开
务	清	策	者	无	愿	事	无	远	虑	者	育	近	忧

行距用 ESC+O 控制, 列距用 CHR\$(1)控制

官	知	止	兮	神	歌	行	肉	眼	闭	兮	心	眼	开
务	清	策	者	无	愿	事	无	远	虑	者	育	近	忧

```
110 FOR I=0 TO 100
120 LINE(I,0)-(0,I),0
130 NEXT I
140 OPEN "tu-4.dat" FOR INPUT AS #1
150 FOR I=101 TO 200 STEP 2
160 LINE(I,0)-(0,I),0:LINE(I+1,0)-(0,I+1),0
170 INPUT#1,A,B
180 IF A=0 THEN 210
190 LINE(A,B)-(A+1,B+2),3,BF
200 GOTO 170
210 NEXT I
220 FOR I=200 TO 318 STEP 2
230 LINE(I+1,0)-(I-200+1,200),0:LINE(I+2,0)-(I-200+2,200),0
240 INPUT#1,A,B
250 IF A=0 THEN 280
260 LINE(A,B)-(A+1,B+2),3,BF
270 GOTO 240
280 NEXT I
290 FOR I=120 TO 218 STEP 2
300 LINE(320,I-120+1)-(I+1,200),0:LINE(320,I-120+2)-(I+2,200),0
310 INPUT#1,A,B
320 IF A=0 THEN 350
330 LINE(A,B)-(A+1,B+2),3,BF
340 GOTO 310
350 NEXT I
360 FOR I=220 TO 320:LINE(320,-120+I)-(I,200),0:NEXT I
370 CLOSE (读程序文件名: ZC2.PRG)
```

近似计算与精确判断之间的矛盾

湖北省煤炭工业学校 廖庆平

电子数字计算机数字表示的离散性和实现计算的代数性决定了数值计算结果的近似性。

我们知道，计算机字长越长，所能表示的数越多，也就是说，计算精确度越高。我们用的苹果机（包括中华学习机）的字长只有八位；因此在计算过程中必然有舍入误差。但是在关系表达式中数值之间的比较却是准确无误的精确判断。我们对于这两者之间的矛盾往往在编程时容易忽视，导致我们在调试程序时莫名其妙，为查找错误浪费不少时间。

为了方便说明问题，请看下面的程序：

```
10 A=1.11: B=1.24: C=8.75: D=0.46
20 M=A+B+C+D: PRINT "M="; M
30 INPUT "A+B+C+D="; N
40 IF M=N THEN 60
50 PRINT"输入错误": GOTO 70
60 PRINT"输入正确":
70 END
```

执行结果为：

```
RUN
M=11.56
A+B+C+D=? 11.56
```

输入错误

实际上，我们从键盘上输入正确数据 11.56，但计算机却显示“输入错误”。为分析原因，在程序中加一个语句：

```
35 PRINT "M-N="; M-N
```

再执行后的结果为：

```
RUN
M=11.56
A+B+C+D=11.56
M-N=-3.27252903E-09
```

输入错误

从运行结果可以看出，M、N 显示的值是相等的，但两者之差为-3.27252903E-09，并非完全相等。

原因在于 20 语句中的近似计算与 40 语句的精确判断发生了矛盾，导致执行结果出错，把 40 语句修改一下，再看运行结果。

```
40 IF ABS (M-N) < 0.00001 THEN 60
RUN
M=11.56
A+B+C+D=11.56
M-N=3.7252903E-09
```

输入正确

这样，执行结果才是正确的。

另外，在步长型循环程序中，当步长值为小数时，为保证程序的正确执行，最好给终值加上一个微量。请看下面的程序：

```
10 FOR I=0.1 TO 1 STEP 0.1
20 P=P+1: NEXT: PRINT "P="; P: END
RUN
```

P=9

我们知道，P 值应该等于 10，程序又没有错，是什么原因呢？

将 10 句改为：

```
10 FOR I=0.1 TO 1.00001 STEP 0.1
```

再看执行结果：

RUN

P=10

这样才是正确的结果。

还有其它情况，比如：不要以为 N 是一个自然数，则 $N \wedge 3$ 也一定是整数；反之，不要希望完全平方数的算术平方根也必定为正整数。

总结：在判断两个数值是否相等时，禁止用等号作比较，应该用两个数值之差的绝对值是否小于某一微量来判断。

电脑用户佳音

CEC-I 中华学习机游戏杆(九针插头) 48 元/个
APPLE II 苹果机游戏杆(十六针双列插头) 48 元/个
IBM 游戏杆(十五针插头) 58 元/个

以上游戏杆二模拟量二开关量，使用进口日本万向电位器，计算机按键，进口插头。现货供应，款到即发。已含邮资。

现货供应电脑资料书：

1. 中华机，APPLE 机“常用工具软件的功能和使用”，每本 4.00 元。
2. 中华机，APPLE 机 DOS 操作系统详解，每本 9.00 元。
3. 苹果机加解密大全，每本 10 元。

已含邮费，款到即寄。购买游戏杆写明机型，详细地址，联系人，邮政编码。单位购买有正式税务发票报销，声明开票单位。

广州市解放北路桂花岗东 1 号

广州师范学院外语系电教室

联系人：王德安

电话：668410 或 663804 转 247

BB 机：广州 181 电话台呼 687028

汉字图形全屏幕绘图

四川省攀枝花市攀钢冶金工业学校 黄文贵

为了给图形加注文字、字符或符号，人们通常使用造型表的方法来实现。中华学习机本身具有 6763 个汉字和 618 个符号及标准打印格式。人们能否直接在绘图中调用它们呢？回答是肯定的。我利用报刊、杂志上已发表的文章，实现了这一目的。在此，向这些作者表示感谢。

为方便读者使用，我将机器语言全部转换为 BASIC 程序。根据把高分辨第二页图形搬到第一页的思路，我们就可以在第一页上绘图，在第二页（即汉字状态下）输汉字。即让图形搬家，使汉字进入图形。我们只要处理好这两页的相互关系，就能在绘图过程中，随时调用汉字，即方便，又省事。请见程序 890~910 语句，只有 38 个字节，用 870 语句调用这段机器语句，就能把第二页汉字搬到第一页的图形中。

请读者亲自使用一下此程序，不仅能体会到国产汉字学习机的优点，而且能赏识到汉字图形的方便之处。相信，在此基础上，会有更好的汉字绘图软件出现。

现简单介绍程序的使用情况。运行程序后，主要分为三部分：第一部分为汉字菜单，占用高分辨的第二页来介绍各个键的绘图功能。第二部分为输汉字状态，同样在第二页进行，用拼音、区位法输入汉字。第三部分为绘图状态，占用高分辨第一页。这三部分互不影响，各占用一个屏幕。用户在使用过程中可随时调用任一部分，如：在绘图时，按 Q 键，就能见到菜单。再按 Y 键，就能输汉字。汉字输完后，自动加到图形中，见程序 790~880 语句。这正是在图形中加汉字的一个重大突破。

另外，在打印、存盘、调盘的过程中，都有详细的中文提示。打印时，能够进行正相、正相放大、反相和反相放大打印。存盘和调盘可以实现压缩图形或不压缩图形功能。注：压缩图形的子程序为 920~1020 语句，由用户任意选择。

注：

① 编制这种程序的关键地方 20、30、140、620、870 语句。

② 为缩短程序，存盘和调盘程序综合在一起，读者可单独编成两个子程序。

③ 再结合报刊已发表的“汉字缩小和放大”、“汉字显示技巧”等程序，会使该程序更增特色。

10 GOSUB 890

20 HGR

30 PR#3: PRINT: PRINT CHR\$(18): HOME: HTAB 8

40 PRINT "汉字图形全屏幕绘图": VTAB 3

50 PRINT TAB(5); "U-左上"; TAB(15); "I-上"; TAB(24); "O-右上"

60 PRINT TAB(5); "J-左"; TAB(24); "L-右"

70 PRINT TAB(5); "N-左下"; TAB(15); "M-下"; TAB(24);

": -右下": VTAB 7

80 PRINT TAB(3); "A-画图"; TAB(11); "Z-擦图"; TAB(19);

"C-清图"; TAB(27); "Q-保留"

90 PRINT TAB(3); "W-打印"; TAB(11); "S-存盘"; TAB(19);

"D-调盘"; TAB(27); "E-结束"

100 VTAB 10: HTAB 8

110 INVERSE:PRINT "加汉字否? (Y/N)";:NORMAL:GET K\$

120 IF K\$="Y" THEN GOSUB 790

130 X=15: Y=16: B\$="Z"

140 POKE-16300, 0: POKE-16302, 0

150 R=PEEK(-16384)

160 IF R>127 THEN R\$=CHR\$(R-128): POKE-16368, 0: GOTO 200

170 HPLLOT X, Y: HCOLOR=0: HPLLOT X, Y: HCOLOR=3

180 IF B\$="A" THEN HPLLOT X, Y

190 GOTO 150

200 IF R\$="A" OR R\$="Z" THEN B\$=R\$: GOTO 400

210 IF R\$="I" THEN Y=Y-1: GOTO 350

220 IF R\$="M" THEN Y=Y+1: GOTO 350

230 IF R\$="J" THEN X=X-1: GOTO 350

240 IF R\$="L" THEN X=X+1: GOTO 350

250 IF R\$="U" THEN X=X-1: Y=Y-1: GOTO 350

260 IF R\$="N" THEN X=X-1: Y=Y+1: GOTO 350

270 IF R\$="O" THEN X=X+1: Y=Y-1: GOTO 350

280 IF R\$="," THEN X=X+1: Y=Y+1: GOTO 350

290 IF R\$="C" THEN 20

300 IF R\$="Q" THEN 30

310 IF R\$="W" THEN 430

320 IF R\$="S" THEN C\$="存盘": D\$="BSAVE": GOTO 580

330 IF R\$="D" THEN C\$="调盘": D\$="BLOAD": GOTO 580

340 IF R\$="E" THEN 730

350 IF X<0 THEN X=0

360 IF Y<0 THEN Y=0

370 IF X>279 THEN X=279

380 IF Y>191 THEN Y=191

390 IF X=0 OR X=279 OR Y=0 OR Y=191 THEN GAME

400 IF B\$="A" THEN HPLLOT X, Y: GOTO 150

410 HPLLOT X, Y: HCOLOR=0: HPLLOT X, Y

420 HCOLOR=3: GOTO 150

430 PR#3: PRINT: HOME: PRINT CHR\$(18)

440 VTAB 2: PRINT "请选择打印方式:"

450 VTAB 4: HTAB 7

460 PRINT "1-正相 3-正相放大"

470 VTAB 6: HTAB 7

480 PRINT "2-反相 4-反相放大";

490 GET K\$: K=ASC(K\$)-48

500 IF K<1 OR K>4 THEN 470

510 IF K=2 THEN K=K+31

520 IF K=3 THEN K=K+62

530 IF K=4 THEN K=K+93

540 PR#1: POKE 1913, K: PRINT CHR\$(17): PR#0

```

550 PR#3: PRINT: VTAB 9: HTAB 13
560 PRINT "打印完毕!";: PRINT CHR$(18): GET K$
570 GOTO 140
580 PR#3: PRINT: HOME: VTAB 2
590 PRINT "请输入"; C$; "的文件名:";
600 INPUT ""; N$: VTAB 4
610 PRINT "按 Y 键压缩图形"; C$;: INPUT ""; K$
620 CALL 43089: VTAB 7: HTAB 2
630 PRINT "正在"; C$; ", 请等等....."
640 IF K$ < > "Y" AND R$ = "D" THEN PRINT CHR$(4);
D$; N$; A$2000: GOTO 710
650 IF K$ < > "Y" THEN PRINT CHR$(4); D$; N$; A$
2000, L$2000: GOTO 710
660 IF R$ = "D" THEN PRINT CHR$(4); D$; N$
670 POKE 230, 32: POKE 0, 0: POKE 1, 96
680 IF R$ = "D" THEN CALL 5632: GOTO 710
690 CALL 5756
700 PRINT CHR$(4); D$; N$; "A24576, L"; PEEK(0)
+PEEK(1) * 256-24575
710 VTAB 9: HTAB 2
720 PRINT C$; "结束!";: GET K$: GOTO 140
730 PR#3: PRINT: CALL 43089: HOME
740 VTAB 4: HTAB 11: PRINT "按任意键结束"
750 VTAB 6: HTAB 4
760 PRINT "Y 键继续使用磁盘中的文件";: GET K$: PRINT
770 IF K$ = "Y" THEN PRINT CHR$(4); "PR#6"
780 HOME: VTAB 5: HTAB 13: PRINT "再见": VTAB 9: END
790 POKE 942,140:POKE 251,253:POKE 252,237:CALL 50030
800 HOME: PRINT "请输入汉字 (或字符)"
810 VTAB 3: INPUT ""; K$
820 POKE 942,129:POKE 251,245:POKE 252,237:CALL 50030
830 INPUT "需要放在第几行"; V
840 INPUT "需要放在第几列"; H
850 HOME: PRINT CHR$(18)
855 IF V=11 THEN 885
860 VTAB V: HTAB H: PRINT K$;
870 CALL 797
880 RETURN
885 A=PEEK(54): B=PEEK(55): POKE 54, 0: POKE 55, 3
886 POKE 36, 0: POKE 37, 10: PRINT K$
887 POKE 54, A: POKE 55, B: CALL 797
888 RETURN
890 FOR I=768 TO 834: READ Q: POKE I, Q: NEXT I
895 DATA 32, 171, 195, 201, 127, 240, 16, 72, 173, 203, 3,
201, 2, 104, 9, 128, 176, 5, 41, 127, 32, 125, 239, 32, 13,
238, 76, 185, 195
900 DATA 169, 0, 133, 0, 169, 32, 133, 1, 169, 0, 133, 2,
169, 64, 133, 3, 160, 0, 177, 2, 17, 0, 145, 0, 200
910 DATA 208, 247, 230, 3, 230, 1, 165, 1, 201, 64, 144,
237, 96
920 FOR I=5632 TO 5879: READ Q: POKE I, Q: NEXT I
930 DATA 165, 230, 9, 4, 133, 6, 162, 1, 134, 4, 160, 0,
132, 2, 132, 5, 132, 8, 177, 0, 208, 24, 230, 0, 208
940 DATA 2, 230, 1, 177, 0, 133, 8, 230, 0, 208, 2, 230, 1,
177, 0, 133, 7, 165, 7, 198, 8, 164, 2, 145, 5
950 DATA 232, 232, 224, 192, 144, 18, 230, 2, 164, 2, 192,
40, 144, 8, 198, 4, 48, 21, 160, 0, 132, 2, 166, 4, 32

```

```

960 DATA 90, 22, 164, 8, 208, 217, 230, 0, 208, 189, 23, 0,
1, 208, 185, 96, 138, 41, 192, 133, 5, 74, 74, 5, 5, 133
970 DATA 5, 138, 133, 6, 10, 10, 10, 38, 6, 10, 38, 6, 10,
102, 5, 165, 6, 41, 31, 5, 230, 133, 6, 96, 160
980 DATA 1, 132, 4, 132, 3, 136, 132, 2, 165, 230, 9, 4,
133, 6, 132, 5, 177, 5, 208, 2, 9, 128, 162, 1, 134
990 DATA 8, 133, 7, 164, 2, 166, 3, 232, 232, 224, 192,
144, 15, 200, 192, 40, 144, 6, 198, 4, 48, 27, 160, 0, 132
1000 DATA 2, 166, 4, 134, 3, 32, 90, 22, 177, 5, 208, 2, 9,
128, 197, 7, 208, 6, 230, 8, 208, 212, 198, 8, 72
1010 DATA 160, 0, 166, 8, 240, 14, 224, 4, 176, 10, 165, 7,
32, 239, 22, 202, 208, 250, 240, 13, 152, 32, 239, 22, 138
1020 DATA 32, 239, 22, 165, 7, 32, 239, 22, 104, 36, 4, 16,
165, 96, 145, 0, 230, 0, 208, 2, 230, 1, 96
1030 RETURN
(该程序文件名: ZC29.PRG)

```

电算 Fermat 大定理

汕头四中 陈君佐

Fermat P·de (1601·8·17—1665·1·12) 是法国著名数学大师, 他终生偏爱数学, 1637 年提出大定理:

$$X^n + Y^n \neq Z^n \quad (n \geq 3)$$

三百五十多年来, 很多数学家, 为证明它的正确性, 花了大量的心血。

1991 年《大自然探索》第二期, 刊出河北省地震局黄诗炎工程师《应用“尾数域”组合规律证明“费尔马大定理”》, 我觉得这是用初等数论作出的有益探索。

四年前, 我也曾用 LASER—310 微机, 编了一个 BASIC 程序, 用 RUN 运行它, 一天半, 不停地观看荧屏, 有见结果, 反面“验证” $X^n + Y^n = Z^n \quad (n \geq 3)$ 的不可性。

事实上, 在一天半的时间内, 电脑作了万万亿次寻求, 就是找不到, 符合上式要求的答案。现在, 我校有了四台 IBM PC/XT, 我又用它们编程:

```

10 REM "FERMAT BIG - BAS"
20 PRINT "费尔马大定理"
30 INPUT "大数="; DS: FOR N=3 TO 10
40 K=DS/(1/N)
50 FOR X=1 TO K
60 FOR Y=X TO K
70 FOR Z=Y TO K
80 IF X^N+Y^N=Z^N THEN 200
90 NEXT: NEXT: NEXT: NEXT
100 END
200 PRINT DS; N; X; Y; Z
210 GOTO 90

```

我, 同样等了半天, 不见一个结果。

“难道程序有电脑查不出的错误吗?”

我把 30 的 N=3 TO 10 改为: N=1 TO 10

电脑就显示:

```

x=1 y=1 n=1 z=2
x=1 y=2 n=1 z=3
:

```

可见, 原程序并没有错误。BASIC 程序是正确的, 那么就“验证”: $x^n + y^n = z^n \quad (n \geq 3)$ 是不可能成立的——费尔马大定理是正确的。 ■

一九九一年广东省青少年“科特杯”信息学(计算机)竞赛 初赛试题

本机上机编程, 调试, 二小时完成。

试题 (共五题):

1. 请在键盘上分别输入下面的两个程序段 A 和 B, 并按下面的要求完成: (20 分)

(1) 请给程序段 A 和 B 中的变量 I, K 赋给初值 (仅知道 I, K 是整数, 并且两程序段 A 和 B 执行结果 I, J, K, T 的值分别是 -1, -2, 1, 2), 请在答卷上写出所有可能的 I, K 的初值。

(2) 在 A~H 格中填入正确的语句 (每格只填一个语句, 不含“:”号, 不含任何转向语句; 所用的变量只允许是 I, J, K, T 或 K0)

程序 A:

```
10 I = _____ : T = 0: K = _____
20 I = I - 1: J = I + 1
30 IF I > J THEN K = K - 1: GOTO 70
40 IF I > -J THEN 70
50 T = T + I: I = I + T
60 GOTO 40
70 IF I < -K THEN 100
80 I = I - K: K = K - 1
90 GOTO 70
100 IF K > -2 THEN 20
110 PRINT I, J, K, T
120 END
```

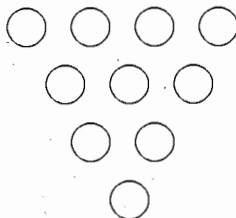
程序 B:

```
10 I = _____ : T = 0: K = _____
20 FOR K0 = (A) _____
30 I = I - 1: J = I + 1
40 IF I > J THEN K = K - 1
50 FOR (B) _____
60 IF I < J THEN T = T + I: I = I + T
70 (C) _____
80 NEXT: (D) _____
90 FOR (E) _____
100 IF I > K THEN I = I - K: K = K - 1
110 IF I > K THEN J = J - 1
120 NEXT: (F) _____
130 (G) _____
140 NEXT (H) _____
```

```
150 PRINT I, J, K, T
160 END
```

2. 试编写一行程序, 实现将任意 N 进制整数 X 转换为相应的十进制数 D. (20 分)

3. 将 1~10 这 10 个自然数不重复, 不遗漏地填入如下图中的 10 个圈内, 使除最上层外, 其余各层每个圈中的数等于对应上一层的相邻两个圈中数之差, 试编写一个程序, 找出所有填数的方案. (20 分)



4. 假设 A, B, C, D, E 五个城市间旅费如下表所列, 左边一列是起点, 上面一行是终点, 即从 A 到 B 的旅费为 7, 从 B 到 A 的旅费是 6, 某人想从一城市出发游览到各市一遍, 再回到出发地, 而所用旅费最省, 试编写一个程序, 求出从各地出发回到出发地的所有最佳路线. (20 分)

	A	B	C	D	E
A	0	7	3	10	15
B	6	0	5	13	12
C	4	8	0	5	10
D	9	11	6	0	11
E	17	14	9	8	0

5. 用计算机设计马赛克 (纸皮石) 的图案, 已知有一块正方形的纸皮, 面积为 $N \times N$ (N 为偶数), 又有 $N^2/2$ 个长为 2, 宽为 1 的长方形的瓷片, 请你把这些瓷片摆设在这张纸皮上, 要求恰好把纸皮全部复盖, 试编一程序, 能打印出所有的不同方案, 要求每种摆设不能重复 (所谓重复是指经过旋转一定角度或反转过来时相同). (20 分)

复赛试题

本机上机编程, 调试, 不限语言, 四小时完成。

试题:

有一个含 $N \times N$ 格的迷宫, 每一格或者是空, 或者是实, 如果有一个位于迷宫的一空格 (x, y) 中, 则他仅能到达相邻的空格 (指上、下、左、右)。

现有一人从 $(1, 1)$ 始点出发, 目标是 (N, N) , 他

随身带有 K 个炸弹 ($0 < K \leq N$), 一个炸弹的威力能将与他相邻的一个实格炸成空格。

编一程序, 求出 r 个被炸的实点 (格) 位置 ($0 < r \leq k$) 和此人从始点到目标的路径, 并且 r 是所有满足条件中的最小值。

要求:

用 DEBUG 解除所有主引导记录型病毒

武汉测绘科技大学九州电脑公司 王全国

大麻、中国炸弹等系统病毒，能够感染到硬盘的主引导记录中，很多查病毒的软件都发现不了，因而难以解除，更何况新的主引导记录型的病毒层出不穷。目前解这类病毒，大多数人都是将硬盘重新低级格式化，带来了许多不必要的麻烦和损失。一般的主引导记录型病毒都有一个特点，就是用软盘启动后仍可进入 C 盘。由此可见，被感染的硬盘的主引导记录中，分区表没有被破坏。依此特点，我们可用下法消除所有的主引导记录型病毒。

用编辑器将附表的程序输入计算机，取名为 DEUP.COM。

用无毒的 DOS 盘启动，再键入命令

DEBUG<DEUP.COM

即可消除所有主引导记录型病毒。如果机器无病毒，也不会出现意外。

本段程序是 DEBUG 命令序列文件。首先，填充 100H 到 2FFH 这一段空间以备后用，然后用 A 命令从 100H 编一个硬盘的主引导程序，接着读硬盘的主引导记录到 400H 处，再将分区表搬至 2BEH 处，这样就构成了一个适合你的机器的主引导记录，最后将它写回你的硬盘。

注意在运行前仔细核对每条指令，其中空行都是必不可少的。本程序在 DOS 版本 3.30 上通过。

用这种方法还可解决一部分硬盘无法启动的问题。

```
F 100 L 200 0
A 100
CLI
XOR     AX, AX
MOV     ES, AX
MOV     DS, AX
MOV     SS, AX
MOV     SP, 7C00
STI
MOV     SI, 7C00
MOV     DI, 0600
MOV     CX, 0200
CLD
REPZ
MOVSB
JMP     0000: 061E
MOV     CX, 0904
MOV     SI, 07BE
CMP     BYTE PTR [SI], 80
JZ      013F
ADD     SI, +10
LOOP    0124
MOV     SI, 0684
LODSB
OR      AL, AL
```

(1) 输入 N*N 的迷宫时，应先画出一个全是实点的迷宫（参见图 (a) 的例子，B 表示实点，N 值应足够大，N 至少 >= 7），(10 分)

(2) 该迷宫能由使用者移动光标选择任一点，并能通过某个键（自定）使此点（格）由实变空，或由空变为实。(20 分)

(3) 完成题目所要求的任务，要求所设计程序具有通用性，良好的结构并选择好的算法。输出结果时，在输入图（即利用 (2) 的功能作出的某一个迷宫）中显示被炸的点（格）及路径，要求写出你所用的测试数据，运行结果应能显示（打印）类似图 (b)，(c) 的迷宫图形。(50

分)

注意：图 (b) 仅是根据 (2) 的功能作出的一个迷宫例子，图 (c) 是在图 (b) 上完成的结果，用“-”表示路径，用“*”表示被炸的点（格）。

(4) 写出必要的文档材料：(20 分)

写出该软件的使用说明；

对程序中的变量，函数作必要的说明；

写出编程思路（算法思想）和流程图；

讨论：简单说明能否找出更高效的算法。

(5) 对显示效果好，程序结构好，算法效率高的程序酌情加分（最高可加至 20 分）。

输入 N=5, K=3

B	B	B	B	B
B	B	B	B	B
B	B	B	B	B
B	B	B	B	B
B	B	B	B	B

(a)

输入后:

B			B	
	B		B	B
	B			B
	B	B		
				B

(b)

结果是:

B		-	*	-
	B	-	B	B
	B	-	-	B
	B	B	-	
-	-	-	-	B

r=1 (c)

解剖电脑病毒的基本方法

湖南日报社 熊虎

我们以准备工具、采取样本、解剖病毒这样的顺序来进行研究病毒的工作。

准备工具：解剖病毒时，应尽量避免将工具程序调入硬盘中使用，因为硬盘无法绝对的防写。请将你所挑选的工具程序 COPY 到几张磁盘上，贴上防写纸以防工具盘受感染。如果你平常都是以硬盘中的 DOS 来开机，而使用其中的 DOS 公用程序，那么你最好快作一张可以用来开机的 DOS 磁盘，将 DOS 公用程序全部 COPY 进去，贴上防写纸备用。读者必须要确定这套存入磁盘的 DOS 是“干净”的，如确定不了，可用 ML.COM (或 MEM) 检查一下 DOS 和工具软件。将这些工具程序一一地执行 (勿执行常住程序)，每执行一个程序后，就执行 M1.COM，如果程序还留在 RAM 里，则这个程序极有可能常有病毒。例如：

这个方法虽然不是百分之百有效，但是至少检查之后你可以比较安心地使用。

采取样本：所谓的采取样本，也就是取得所要解剖的病毒，下面介绍几种取样的方法：

a. 磁区抄录法：从病毒藏身的磁盘里直接将病毒

COPY 出来，这是最安全又直接的方法之一，适用于采取“开机型”病毒样本，但是需要特殊的工具软件，而且需对

A>mi/v-先做自我检查

Memory Info v5.19

Copyright 1989 Central Point Software, Inc. All rights reserved.

Conventional memory: Total: 640K 带有病毒的程序会留在

Largest executable program: 565K RAM 中，所以若 MI 本身就

Type Paragraphs Bytes Owner 带有病毒，RAM 中就有两份。

.....

Prog 0EDD-1040h 5696 0EDDh COMMAND / v .

& using interrupt 23h-24h, 2Eh.

Prog 1059-1224h 7360 1059h GMOUSE c:\ut\GMOUSE.COM

& using interrupt 0Ch, 10h, 33h, FFh.

Prog 27AF-282Eh 2048 27AFh M1 A:\M1.COM / v

& using interrupt 08h, 21h-22h. 两份! 有病毒!

Prog 2837-9FFFh 479K 2837h M1 A:\M1.COM / v

正常情况下只会看到一份 M1

• Press a key to resume...

```
JZ 013C
MOV AH, 0E
INT 10
JMP 0131
STI
JMP 013D
MOV DI, SI
DEC CX
JZ 0151
ADD SI, +10
CMP BYTE PTR [SI], 80
JNZ 0141
MOV SI, 0684
JMP 0131
MOV SI, 0005
MOV DX, [DI]
MOV CX, [DI+02]
MOV BX, 7C00
MOV AX, 0201
INT 13
JNB 016F
XOR AX, AX
INT 13
DEC SI
JNZ 0159
MOV SI, 06B5
JMP 0131
```

```
POP CX
CMP WORD PTR [7DFE], AA55
JZ 017D
MOV SI, 069C
JMP 131
MOV SI, DI
JMP 0000: 7C00
DB "Invalid partition table ", 0
DB "Missing operating system ", 0
DB "Error loading operating system ", 0

A300
MOV AX, 201
MOV BX, 400
MOV CX, 1
MOV DX, 80
INT 13
INT 3

G=300
M 5BE 5FF 2BE
A300
MOV AX, 301
MOV BX, 100
```

G=300

Q

(该程序文件名: ZC30.PRG)

电脑病毒的藏身方式有相当的了解。此方法我们留待日后讨论。

b、“记忆体”抄录法：先让病毒启动查出病毒的中断向量，将该中断向量所指向的“记忆区”（segment）内容全部或部分抄录到指定的档案文件里。这种方法是“活体”（已启动）的样本，而且只需 DEBUG 一个工具软件就可以办到，但涉及的技巧颇为复杂，而且必须对病毒有相当的了解才行，我们也留待稍后再谈。

c、感染取样法：先让病毒软盘启动，（注意，不可在有硬盘的系统上启动），然后让病毒感染你备好的磁盘或文件。这个方法简单有效，但需谨慎。实施方法如下：

1、以“干净”的 DOS 开机，FORMAT 两张 360K 的磁盘，一张注明“对照磁盘”，一张注明“取样磁盘”

2、用 DEBUG 制作一个小小的取样程序文件，存入“取样”磁片。例如：

```
A>debug
-a 100
7E31:0100 mov ax,4c4c
7E31:0103 int 21
7E31:0105
-n b:\test.com←档案文件命名，取样磁片在 B:中
-r ex←设空长度
CX 0000
S: ←
-w←写入磁片
Writing 00005 bytes
-q
A>dir b:\test.com
Volume in drive B is TEST_VIDISK
Directory of B:\
TEST COM 504-08-90 11:11a
1 File(s) 265728 bytes free
```

（将此程序 COPY 一份存在“对照”磁片里，贴上防写纸。取样磁片不贴防写纸）。

3、执行疑似带毒的程序文件，再执行取样的程序文件。（即 test 回车）。然后关机让 RAM 里的病毒消失（要关电源，不能按 ALT CTRL DEL），再以干净的 DOS 开机，查看取样程序文件是否被感染。

4、以疑似带毒的磁片开机，再以 DIR、COPY 等命令读写“取样磁片”。

```
A>xyz.exe←执行疑似带病毒程序文件
.....
A>b:test←执行取样程序文件
(重新开机)
A>dir b:test.com 档案文件变大便是被感染了
Volume in drive B is TEST_VIDISK
Directory of B:\
TEST COM 1818 04-08-90 11:11a
1 File(s) xxxxxx bytes free
A>
```

```
A>dir b:←可多执行几次(取样片放在 B:)
```

```
Volume in drive B is TEST_VIDISK
```

```
Directory of B:\←取样磁片在 B:中
```

```
TEST COM 1818 04-08-90 11:11a
```

```
1 File(s) xxxxxx bytes free
```

```
A>copy con b:tmp.txt←写入一个小档案文件
```

```
abcdc^Z
```

```
1 file(s) copied
```

```
A>del b:tmp.txt←再将该档案文件删除
```

5、以干净的 DOS 开机，再用 SDUMP（或 DEBUG、PC SHELL）查看“取样磁片”的第 0 磁区与“对照磁片”内容有何不同。正常情况下，这两张磁片顶多只有几个 bytes，与磁盘编号（DOS 4.0 以上才有）会有所差异，除此之外，若还有许多不同的地方，则表示你已经取得“开机型”病毒的样本了。

解剖技巧：我们来谈谈用 SOURCER 解剖病毒的特别技巧。（SOURCER 软件使用方法附后）

读者如果是初次使用 SOURCER，最好只做 I、O、N、C 的设定，先选几个较小的 EXE、COM 来试试。（I、O、N、C 的设定，为 SOURCER 软件特定功能）

根据笔者经验，SOURCER 很少将数据误认为程序，但经常将一小段程序误认为数据。这种情形往往发生在程序里使用了特殊的 JUMP 方式、移位方式、或者调整了 ES、DS 等寄存器之值时，而使 SOURCER 发生误解。

读者若将 A 功能表中的 a、b、c、f 选项设为 on（字母较亮），可以减少这种误解，但仍无法完全避免。所以读者可以先令 SOURCER 将档案文件载入，再特别指定起始地址，而以 fragment 型的形式来汇编该段的机器码。

前面讲述的感染取样法取得的开机型病毒是以独立磁区形式存于磁盘中。其程序文件必定由磁片的第 0 磁区内开始，所以读者需由磁区内容开始汇编。必须先使用 DEBUG（工具软件）内容制成档案文件，再由 SOURCER 进行反汇编。例如：

```
A>debug
-1100,1,0,1←入口地址:B:磁片第0磁区起,一个磁区
-n b:vstart.com←档案命名
-r cx
CX 0000 (一磁区=512byte)
200 ← 档案长度
-w
Writing 0200 bytes
-q
A>dir a:vstart.com
Volume in drive B is N
Volume Serial Number is 054C-19D6
Directory of B:\
VSTART COM 512 04-09-90 9:22a
1 File(s) 361472 bytes free
```

需提醒读者的是, 虽然我们将档案文件命名为 .COM, 但是用 XOURCER 汇编时, 欲需指定 zero start, 因为此磁区的内容是由 CS: 0000 (IP=0000), 而非由 CS: 0100 开始执行。

我们用 .lst 档文件输出, 程序列出后才会有地址及原机器码可以对照, 如下列所示:

```

..... (000A 地址)
=000A data-0018c cqu 0Ah
=000C data-0019c cqu 0Ch
=000E data-0020c cqu 0Eh
=000F data-0021c cqu 0Fh
.....

6DAA:038D75 09      jnz loc_0008
6DAA:038F8E 06 000F  movs,ds:data_0021c
6DAA:0393 2E:FF 2E 000A jmpdword ptr cs:data_0018c
6DAA:0398      loc_0008:
6DAA:0398 83 C4 06      add sp,6      常数名
                        标名
程序的入口地址      机器码

```

如上列所示, SOURCER 会自动将程序中用到的一些记忆地址设为常数而赋予代名。如 loc-xx、data-aa 都是。阅读程序时必须注意学时的 CS、DS、ES 为何, 如上例的 moves, ds: data_0021c 指令, 与当前的 DS 值有绝对的关系。

病毒程序里常会使用一些特殊的方法来跳跃、移位或返回。如上例 OFFSET 0393 地址上的 JMP 指令, 是取 CS: 000A 开始的 4 个 bytes 为 CS: 1p 做 jump 的动作。这种技巧的确骗过了 SOVR CER。需提醒读者的是, 任何一个记忆位置的地址表示法都不是唯一的, 如 3000: 010A 与 3010: 000A 就是同地址。所以, 读者想想上例中若 CS:00A 存的地址是 6D8A: 0598, 则 JMP 的目的地是哪里呢? 没错, 就是 loc-0008 只为了往下跳一个 byte? 不: 这样做是为了调整 CS! 像这种情形, 如果上面没有 jnz loc-0008, SOURCER 就可能将下面的机器码都当成 data3。

阅读程序时, 了解某一地址数据, 常数, 子程序或程序段的功能时, 应立即将其代名更改为较易联想的代名, 例如: drv_n (磁盘机号)、load_file; 等。再利用 PE2 的字串替换功能, 将整个档案文件中所出现的该代名一次全部换掉, 如此有助于下面的程序。当这些代名逐渐被取代后, 整个程序的轮廓就渐渐呈现出来, 可读性便越来越高, 当然也就越容易 Trace 了!

附: 解剖病毒的工具软件的使用方法。

(一) DEBUG 的用法:

DEBUG 可将病毒程序编译成汇编语言, 我们以前面取样的样本为例, 谈谈 DEBUG 的 Vnussemble 操作: (以下为盲目操作, 为的是取得档案列表, 否则可在屏幕上显示出来)。

A> debug b: test.com> b: test.lst←输出转向到此

—u100 L71A←档案长度 (16 进制)

(请盲目键入这列命令)

```

7E85: 0100 E99200 JMP 0195
7E85: 0103 7355 JNB 015A
7E85: 0105 4D DEC BP
7E85: 0106 7344 JNB 014C
7E85: 0108 6F DB 6F
7E85: 0109 7300 JNB 010B
-q 盲目键入
A> 看得到了

```

包含所键入的命令都不会在屏幕上显示出来

如上所述, 第一个指令是跳至 0195 去执行, 接下来的程序码其实应该是数据资料了, 如 "4D73446F73" 应该是 "MSDOS", DEBUG 欲误以为是机器码, 这是因为 DEBUG 完全不去理会程序的流程, 而只一味地由上而下地汇编。SOURCER 这个工具程序则不同, 它会自动追踪程序的流程, 辨别程序或数据资料。接下来我们来介绍这个工具软件的使用方法。

SOURCER 的用法:

SOURCER 的启动有两种方法:

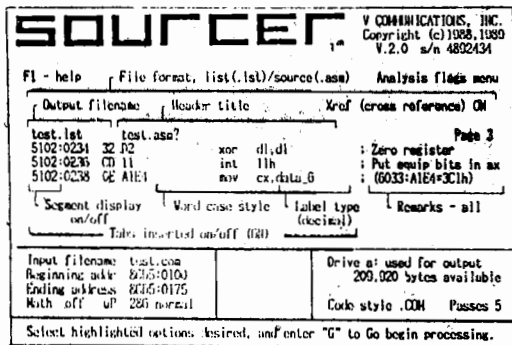
A> SR B: TEST.COM

指定要汇编的程序名。程序后必须加上 COM 或 EXE

A> SR

不预先指定。

进入 SOURCER 后, 画面如下:



画面中间部分便是其输出的实例。图中较亮的字母则是选项设定功能的 Hot Key, 用途如下:

I 键: 输入程序文件名: 若启动时指定了文件, 则此项显示的就是该文件的文件名。(如上图所示)。按下 I 键后, 可由键盘输入欲汇编的程序文件名 (后缀不一定要 COM 或 EXE)。按下 Enter 后, SOVR CER 便会将该文件载入 RAM, 并且在下面两项显示出该文件载入的起始地址。

B 与 E 键: 起始地址。指定汇编起始范围内的记忆体

其一是检测内存中断向量表 0000: 0084H 中 int 21H 的入口是否为 01E1H。

其二是用一个未染病毒的已知长度的 COM 文件或 EXE 文件 (例如用一个未染病毒的 COMMAND.COM) 放进 A 盘, 然后用 DIR 命令显示 A 盘中该文件的目录, 例如 DIR A: *COM。

如此重复 DIR 命令若干次, 若该文件长度改变 (有时需要用不带毒的系统盘重新启动后, 再用 DIR 命令才可以看到该文件长度的改变), 即 COM 文件增长 1220 字节或 EXE 文件增长 1223 (或稍多几个字节), 则可判断该计算机有旅游者病毒。

这种方法是判断文件型病毒通用的一种方法。笔者正是利用这种比较长度的方法发现这种新病毒的。

要判断一个 COM 文件是否带有旅游者病毒, 可以用 debug 把该 COM 文件调入内存 CS: 100H 处。

则可以看到带毒的 COM 文件在 CS: 100H 处的首语句为: JMP XXXX 的三字节跳转命令, 然后用 T 命令, 可看到程序执行的第二个语句是一个由二字节组成的 JMP XXXX 命令。再执行一次 T 命令, 则可看到程序执行的第三个语句是 mov al, 43 (机器码 B043)。最后我们从指令所在段内地址后退 13 个字节, 再用 d 命令观察, 可看到 2A 2E 2A 00 43 4F 4D 45 58 45, 即“* * * COM EXE”, 则可判定该 COM 文件已感染上旅游者病毒。

要判定一个 EXE 文件是否带有旅游者病毒, 则可以用 debug 命令把该 EXE 文件调入内存, 然后用 r 命令, 可看到带毒的 EXE 文件的首语句为:

```
XXX: 0138 B045 mov al, 45
而 sp 值为 0A10H, IP 值为 0138H。
```

如果不具有这种特征, 该文件就没有带旅游者病毒。

用 debug 清除感染上旅游者病毒的 COM 文件步骤如下:

(1) 用不带毒的写保护的软盘启动 (假设该软盘有不带毒的 debug.com 文件) 进入“A>”提示符。

(2) A>debug C: XXXX.com (假设要消毒的 COM 文件在 C 盘)

进入 debug 提示符“-”状态。

(3) 按前面介绍过的方法, 判断该 COM 文件是否带有旅游者病毒, 若有, 则继续以下步骤。

(4) -dcs: 101 102

则可以看到这两个字节组成的一个字 (低位在前, 高位在后) 实际上是原文件未感染旅游者病毒前的文件长度减去 3。

不妨设该字的值为 y, 原文件长度 Z=y+3。

(5) -R CX 更改文件长度为 Z, 输入 Z 值, 再回车。

(6) 原 COM 文件的头三个字节被病毒转移到原文件末尾后面偏移量为 01AH 处, 故不妨设 X=Z+01AH+100H (因该 COM 文件起始于 CS:

100H), 则可以用 m 移动命令。

-m CS: X L3 CS: 100

把原来的三个字节移回到原来 CS: 100 处。

(7) -W 写盘。

(8) -q 退出: 则所得的 XXXX.COM 是一个不带旅游者病毒的 COM 文件。

笔者一个 DOS3.31 的 COMMAND.COM, 原文件长为 25332, 感染上旅游者病毒后, 文件长变为 26552, 加长 1220 字节, 用上述方法消毒后, 文件长 25332, 与原来一样, 且可正常运行。

另有一个 cdebug.com, 原文件长 11904, 感染上旅游者病毒后, 文件长变为 13124, 文件长度加长 1220, 按上述步骤消毒后, 文件长 11904, 与原来一样, 且可正常运行。

用 debug 清除感染上旅游者病毒的 EXE 文件步骤如下:

(1) 用不带毒的写保护的软盘启动 (设 A 盘有不带毒的 debug.com 文件), 进入“A>”提示符。

(2) A>debug C: xxxxx.EXE (假设要消毒的 EXE 文件在 C 盘)。

进入 debug 提示符“-”状态。

(3) 按前面介绍过的方法, 判断该 EXE 文件是否带有旅游者病毒, 若有, 则继续以下步骤:

(4) -d CS: 107 10E

记录该 EXE 文件未带毒前的重要参数, 其中 CS: 109H 有原 CS 值, CS: 107H 有原 IP 值, CS: 10BH 有原 SS 值, CS: 10DH 有原 SP 值。

(5) -q

退出 debug 状态, 回到“A>”提示符。

(6) A>copy C: xxxxx.EXE C: 111

要修改 XXXX.EXE, 先要把该文件拷贝为中间文件, 此处取名为 111。

(7) A>debug 111

把中间文件调入内存 CS: 100H 处。

(8) -R CX

把显示出来的 EXE 文件长度低 16 位字节数减去 04C7H, (若不够减, 则还要用 R BX 命令修改 BX, 即用 BX-1 赋值给 BX, 然后用 10000H+CX-04C7H, 计算新的 CX), 修改文件 111 的长度。

(9) -d CS: 104

取得 CS: 104H 中保存的该 111 文件所占实际扇区数, 用该数 (占 2 个字节) 减去 2, 得到消毒后的文件所占实际扇区数 (比原带毒文件少占 2 个扇区)

(10) 用 e 命令修改该 EXE 文件的 5 个重要参数, 其中:

CS: 104H 改为消毒后的文件所占实际扇区数

CS: 10EH 改为未染毒前的 SS 值

CS: 110H 改为未染毒前的 SP 值

(下转 16 页)

设立存取硬盘的用户权限

上海灯头二厂 黄 昇
上海电气联合公司 杜建春

随着微机的应用的日益普及, 电脑信息的安全显得日趋重要, 同时非法存取和非法拷贝愈演愈烈, 建立安全措施很有必要。通常建立硬盘口令字在硬盘主引导程序或引导程序一级, 本程序 disklock 是中断驻留程序, 在新的磁盘 I/O 中断程序中设立口令, 不更改主引导扇区和引导扇区, 以免和一些病毒程序混淆。DISKLOCK 启动后, 不输入正确口令则用户只有读权限, 即不能更改盘上信息 (包括 A 驱动器和 B 驱动器), 同时也阻止硬盘和软盘间的文件拷贝。另外, 口令输入方式具有隐蔽性, 只有建立 DISKLOCK 的人才知道怎样才能输入口令。

建立过程:

C> MASM DISKLOCK.ASM

C> LINK DISKLOCK.OBJ

C> EXE2BIN DISKLOCK.EXE DISKLOCK.COM

最后将 DISKLOCK 放入硬盘根目录中的 AUTOEXEC.BAT 中, 则每次从硬盘启动机器, 存取权限就建立。运行 DISKLOCK 之后, 用户只有读权限。当同时按下 CTRL 和 F2 键, 屏幕会提示输入口令, 口令正确后, 用户具备存取权限。如欲设置只读权限, 再同时按下 CTRL 和 F2 键即可。

源程序中有键盘扫描码的注释, 用户可更改键码, 自行定义控制。现将一些控制键的扫描码列出, 供参考:

键	Shift	Ctrl	Alt
F1	54h	5eh	68h
F2	55h	5fh	69h
F3	56h	60h	6ah
F4	57h	61h	6bh
F5	58h	62h	6ch
F6	59h	63h	6dh
F7	5ah	64h	6eh
F8	5bh	65h	6fh
F9	5ch	66h	70h
F10	5dh	67h	71h

源程序如下:

```
disk lock
code segment
    assume cs:code,ds:code
    org 100h
start: jmp init
old13 dd ?
old16 dd ?
```

转初始化处理
存放原 INT 13H 中断向量
存放原 INT 16H 中断向量

```
flag13 db 1
str1 db 'Open lock'
str9 db 'Close lock'
jdx db ?
password db 1fh,16h,12h,2eh
input db 4 dup(?)
please db 'Please input password:'
cursor dw 0
new13 proc far
    push AX
    push dx
    push es
    push bx
    push di
    push cx
    push ds
    cmp ah,05
    je my
    cmp ah,03
    je my
    ori: pop ds
    pop cx
    pop di
    pop bx
    pop es
    pop dx
    pop ax
    assume ds:nothing
    jmp old13
my:
    cmp flag13,1
    jne ori
    pop: pop ds
    pop cx
    pop di
    pop bx
    pop es
    pop dx
    pop ax
    iret
new13 endp
new16 proc far
    assume cs:code,ds:code
    sti
    or ah,ah
    jz my16
    jmp ori16
my16: pushf
    assume ds:nothing
    call old16
```

新的 INT 13H 中断程序
保护现场
判断是否对硬盘写操作
是, 转新的 INT 13H 处理
否, 恢复现场
转原 INT 13H 处理
测试锁标志是否置位
否, 转原 INT 13H 处理
是, 跳过写操作
直接返回
新的键盘中断程序
测试是否有键盘输入
是, 转 my16
否, 转原中断程序
调用原中断程序

```

cmp ah,5fh          ;测试输入键的扫描码是否 5Fh
je asc              ;是,转 asc
jmp ret16           ;否,退出中断

asc:
cmp al,00h          ;测试输入键的 ASCII 码是否 00h
je process          ;是,即 Ctrl+F2 键,转 process 处理
jmp ret16           ;否,退出中断

process:
push dx
push ax
push es
push bx
push di
push cx
push ds
push si
push cs
pop ds
push cs
pop es
call read cur       ;读当前光标位置
mov word ptr cursor,dx ;并保存
cmp flag13,1        ;测试锁当前状态
jne active          ;非激活,则转激活处理
jmp do              ;已激活,转 do 使之不激活

active: mov flag13,1 ;激活标志置位
jmp sent            ;转显示

do:
mov dh,0
mov dl,30
call locate         ;光标定位
lea si,please
mov cx,22
call displ          ;显示 "Please input password."
mov cx,4
mov si,0
mov di,52

entry:
pushf
assume ds:nothing
mov ah,0
call old16          ;调用原键盘中断,等待输入口令
push cs
pop ds
push cs
pop ds
mov input[si],ah
inc si
inc di
call locate
loop entry
cld
mov cx,2
lea si,password
lea di,input
rep cmpsw           ;比较口令是否正确
jne sent            ;否,转 sent
mov flag13,0        ;是,清除标志位

```

```

sent:
cmp flag13,1        ;测试锁标志位是否置位
je act              ;是,转 act
lea si,str1         ;否,将显示 'Open Lock'
jmp goon
act: lea si,str9      ;将显示 'Close Lock'
goon:
mov dh,1
mov dl,30
call locate         ;光标定位
mov cx,11
call displ          ;调用显示子程序
push cs
pop ds
mov dx,word ptr cs:cursor
call locate

popret:              ;恢复现场
pop si
pop ds
pop cx
pop di
pop bx
pop es
pop ax
pop dx

ret16:ret            ;中断返回

ori16:assume ds:nothing
jmp old16

new16 endp
displ proc           ;显示子程序
mov ah,0eh
mov bh,0
disp10:mov al,[si]
int 10h
inc si
loop disp10
ret
displ endp

locate proc          ;光标定位子程序
mov ah,02
mov bh,0
int 10h

locate cursor
ret

locate endp
read cur proc        ;读当前光标位置子程序
mov ah,3
int 10h
ret

read cwnp
init:                ;初始化处理
assume es:nothing
push cs
pop ds
mov ax,3513h
int 21h              ;取 INT 13H 中断向量
mov word ptr old13,bx

```


《WS》实现多页自动打印的方法

江苏泗阳县县委机要室 陈复军

文字处理工作中，很多用户都习惯用 WS 进行编辑，但在进行多页打印时，常会遇到这样的问题：第二页及以后各页的打印位置很难与第一页保持一致，每打印一页往往要用手工来调整，操作麻烦，耽误时间，打印效果也欠佳。其实正确地运用 WS 中的三个点命令，是完全可以实现多页的自动打印。现介绍如下：

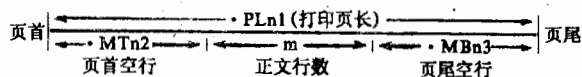
一、三个点命令

- PL n 设每页行数 = n
- MT n 设页首与正文之间空行数 = n
- MB n 设页尾与正文之间空行数 = n

在编辑过程中，不少人都只使用 • PL n 命令，而忽略了 • MT n 和 • MB n 命令的作用。其实这三个命令是有联系的，在编辑和打印过程中，同时起到控制作用。

二、三个点命令的关系

为了便于说明，首先介绍一下三个点命令的相互关系(见图)：



• PL n 包括三个方面：每页页首空行数、页尾空行数和正文行数，即： $n_1 = n_2 + m + n_3$ 。对于某种字型而言，要实现自动打印，必须使 n_1 等于该字型的打印页长（一页打印纸上能打印的行数）。

WS 中 n_1 、 n_2 、 n_3 的默认值分别为 66、3、8，如只使用 • PL n 命令，则 $n_1 = 3 + m + 8 = m + 11$ ，所以编辑时，设置的行数必须比实际行数大 11，否则就不能得到要求的每页行数。笔者认为，只使用 • PL 命令来控制每页行数是不能正常发挥 WS 的功能作用的，不仅会给操作带来许多麻烦，而且对打印速度和打印机的寿命也有一定影响。使用中，最好将三个点命令互相配合起来使用。

三、多页的自动打印

编辑时，首先用 • PL n 命令设置打印页长（某类点阵字型的打印页长一般是固定的），再根据要求的每页正文行数得到页首、页尾的空行总数，然后用 • MT n、• MB n 命令来分配这些空行（同时可调整正文在打印纸上的前后

位置），从而实现无需人工干涉的自动打印。

例：若某字型打印页长为 33，要求每页打印 20 行，可进行如下设置（三种类型）：

• PL33	• PL33	• PL33
• MB 10	• MT 5	• MT 1
(MT 取默认值 3)	(MB 取默认值 8)	• MB 12

此外，用 • MT0 能消除页首空行，适用于起始定位要求较高的场合，如打印蜡纸等；• MB0 可消除分符符号。

四、说明

1. 编辑时，应遵循 $n_1 = n_2 + m + n_3$ ；• PL n 命令应由打印页长决定，而不能再用正文行数加 11 得到；

2. 在 2.13 等系统中，空行与字符行的走纸长度是不同的，且具有调整行距命令，所以本文无法提供具体的参考数据。用如下程序进行测试，可方便地得到各种字型和行距下三个点命令的数据。

```
10 REM 测试 • PL、• MT、• MB 数据程序
20 KEY OFF: CLS
30 INPUT "您习惯使用的每页行数:", A
40 LPRINT CHR$(27) + "I" + "*" + "*"
50 LPRINT
60 PRINT "请进纸至第二页页首, 按任一健!";
70 A$ = INPUT$(1)
80 FOR I = 1 TO A: LPRINT USING "###行"; I: NEXT
90 CLS: PRINT "如走纸至第二页页尾, 请按 ESC 键!";
100 PRINT "按其它键打印机走一空行"
110 A$ = INPUT$(1): IF A$ = CHR$(27) THEN 140
120 B = B + 1: LOCATE 7, 1: PRINT "走"; B; "空行";
130 LPRINT: GOTO 110
140 CLS: PRINT "每页行数: " A
150 PRINT "• PL 应设置为: ", A + B
160 PRINT "• MT、• MB 分配总行数为: ", B
170 END
```

注：

1.40 行用来设置打印字型，如：CCDOS2.1，* 用 A - P 代替；CCDOS2.13 中，40 行用 * 40 LPRINT"@* & * "代替，* 分别表示字型和行距。

```
mov word ptr old13[2],es ;保存 INT 13H 中断向量
lea dx,new13
mov ax,2513h
int 21h ;设置新的 INT 13H 中断向量
mov ax,3516h
int 21h ;取 INT 16H 中断向量
mov word ptr old16,bx
mov word ptr old16[2],es ;保存 INT 16H 中断向量
push cs
```

```
pop ds
lea dx,new16
mov ax,2516h
int 21h ;设置新的 INT 16H 中断向量
lea dx,init
add dx,100h
int 27h ;驻留退出
code ends
end start (该程序文件名: ZC3.PRG)
```

如何实现“FOXBASE+”共享

四川重庆食品公司 王培福

一、问题的提出

“FOXBASE+”以其强大的功能，极快的运行速度，命令文件较好的保密手段等优势，深受广大计算机用户的欢迎。要做到运行速度快，又要节省存储介质，这便需要我们采取相应的措施加以利用。下面本人就这一问题谈点自己的做法，供同行们参考。

二、建立自己的目录。

通常，为了便于文件管理，常常需要建立自己的子目录。为叙述方便，假设我们在某台微机上用“FOXBASE+”开发了财务管理系统，该系统存放在 C 盘的“CW”子目录内，工资管理系统，存放在 C 盘的“GZ”子目录内，等若干应用系统。把“FOXBASE+”存放在 C 盘的“FOX”子目录内。

三、实现共享的方法。

1、利用 DOS 命令“PATH”实现共享。“PATH”命令的用途是在指定的目录里搜寻当前目录中未找到的命令或批处理文件，有三两种方法实现共享。

①自动执行。在批处理文件 AUTOEXEC.BAT 中加上“PATH”命令。

```
C>TYPE C:\AUTOEXEC.BAT
```

命令行

```
PATH C:\; C:\FOX; .....
```

②手动执行。在操作系统提示符下输入：

```
C>PATH C:\; C:\FOX; ..... (驱动器可以是任意的)
```

做①②两步任一步操作后，再进入应用系统子目录，输入：

```
C>CD\CW 回车
```

```
C>MFOXPLUS<应用程序名<设为 CW.PRG 下同
```

>>回车

③建立应用系统批文件 CW.BAT。

```
C>COPY CON C:\CW.BAT
```

```
ECHO OFF
```

```
PATH C:\; C:\FOX.....
```

```
C:
```

```
CD\CW
```

```
MFOXPLUS CW.PRG
```

ECHO ON

^Z

执行批文件 CW.BAT。

C>CW.BAT 回车 (驱动器可以是任意的)

2、进入“FOXBASE+”交互式状态后输入：

```
.RUN C: 回车 (把驱动器转入 C 盘)
```

```
.RUN CD\CW 回车 (相当于 C>CD\CW 回车)
```

```
.DO CW.PRG 回车
```

通过以上处理后，不必在每个应用系统中装入“FOXBASE+”文件就可运行“FOXBASE+”所开发的应用程序，并可节约大量磁盘空间，也克服了使用软盘驱动器执行速度慢的缺点，有兴趣者可以试一试。

活动式磁盘保护

南京动专计算机系 梁耀旋

软磁盘是现代微型计算机最常用的磁存储设备，它具有存储容量大、可长期保存信息、使用方便等优点，而 5.25 英寸磁盘是目前微型计算机中使用最为广泛的软磁盘，它以其特有的结构深受广大用户的欢迎。使用过这种软磁盘的用户都了解，这种磁盘的侧面有一个保护缺口，采用不干胶粘纸封法写保护，是保护和防止用户在误操作时或病毒侵害等造成的数据损失，能有效地保护数据信息。但在一些经常要进行数据读写的场合里，用户为了防止自身的误操作而造成的数据损失，又要提防着病毒的破坏，不得不对其工作磁盘的保护封口频繁的开关，这样不但增加了封口粘纸的消耗，而且给用户带来了许多不便，为此，能否在磁盘安装一个活动的保护封口，以实现灵活的保护作用。

通过对 5.25 英寸磁盘结构的解剖，及其与磁盘驱动器间的工作原理分析，磁盘内部保护缺口位置留有一定的空间，磁盘驱动器的控制电路读写逻辑控制磁头产生读写所需的电流，按程序控制命令，在向磁盘存入数据信息时，电路首先检测工作磁盘的保护缺口是否被封，如写保护就禁止写电流通过磁头，使数据不能写入磁盘，只能读出，综合其结构和工作原理，我设计一种在磁盘内部装入一张保护片代替外保护封口粘纸的方法。(图 1) 采用上下推动活片，打开和封闭磁盘保护缺口，以达到对磁盘的写保护作用，方法简单易行，制作容易，安装方便，而且对磁盘的外形和损度都很小，实用性强。

制作方法和过程：

- (1) 在磁盘壳侧面和背面按 (图 2) 示的尺寸用刀片各划开两条 15mm 的直缝 (注意：磁盘壳背面要划穿两层塑料片，小心不要划穿壳正面的塑料片)。
- (2) 利用一张废磁盘壳的塑料薄片 (或其它类似的非透明塑料薄片) 按照 (图 3) 示的形状尺寸裁剪出来，并且折

2、60 行、90 行中的页首、页尾是指打印纸的折线处与打印头的中间位置对齐。

3、某些字型的打印页长是相同的，只需测试一种字型。如 CCDOS2.13 系统下，40 点阵中的黑体、楷体、仿宋体等。

PC-DOS 磁盘引导扇区内容注释

江苏省地震局 陈兴东

PC-DOS 磁盘引导扇区是联接计算机硬件和操作系统的桥梁,它包含有引导整个计算机操作系统的引导程序和磁盘参数。每张磁盘不管它是否是操作系统盘都有引导扇区。正因为这样所以有一些病毒就拿它作文章,病毒通过修改引导扇区及转移引导扇区来驻留和传染病毒。因此,为了更好地研究病毒、操作系统、磁盘结构,了解引导扇区的内容就十分必要了。下面就介绍一下几种磁盘的引导扇区。每个扇区均为 512 字节(十六进制为 200H,编号从 000H 到 1FFH)。

1. 硬盘主引导扇区

硬盘主引导扇区可用如下方法进行读写,注意,在写时一定要保证引导扇区有用信息没有被破坏。

C>DEBUG↓

-A100↓

XXXX:0100 MOV AX,0201↓

XXXX:0103 MOV BX,0200↓

XXXX:0106 MOV CX,0001↓

其中AH中值02表示读,若AH中为03表示写,AL中数为要读的扇区数,由于引导扇区仅占一扇区,这里取1;

BX中的数表示数据存放的起始地址;

CL中的数低六位表示从某头某柱的第几个扇区开始读写,CH和CL中的高两位共十位组成要读写的柱号;

XXXX:0109 MOV DX,0080↓

XXXX:010C INT 13↓

XXXX:010E INT 20↓

XXXX:0110 <CTREL>+<C>

-G=100↓

Program terminated normally

-NBOOT↓

-RCX↓

CX 0000

200↓

-W200↓

Writing 0200h bytes

-Q↓

C>DEBUG↓

DL中的数表示某个磁盘,00表示A盘,01表示B盘,80表示第一个硬盘,81表示第二个硬盘,DH中的数表示要读写的头号;

读或写扇区中断;

程序终止;

给读或写文件命名;

修改文件长度;

写 200H 字节到 BOOT 中。

这时文件 BOOT 中即为引导扇区的内容。

我们可以利用这个方法在软盘上保存计算机硬盘的主引导扇区,在硬盘主引导扇区遭到破坏的情况下,从软驱启动系统并把引导扇区写到硬盘上,这样硬盘就不会因为主引导扇区的破坏而不能使用。方法如下:

好。

(3) 安装保护片,参考(图4)示,首先将保护片插入磁盘壳侧面直缝里,同时将折面插入背面双层直缝内,将其同时推入磁盘内部(注意:保护片比直缝长些,这是为防止保护片的脱落,要将它压入)。

(4) 上下推动保护开关,检查保护片是否完成打开和封闭保护缺口。

至此,一张带活动式保护口的磁盘就制作完成,它将给你的工作带来方便,提高你的工作效率,有兴趣的朋友不妨制作使用。

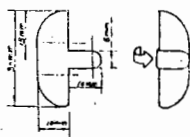


图3

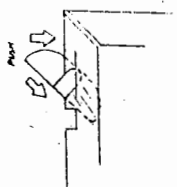


图4

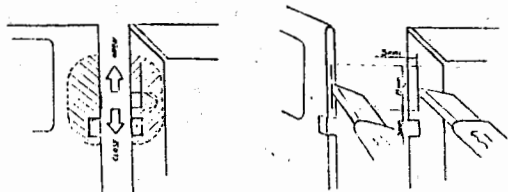


图1

图2

有效地扩充软盘容量

武汉大学 雷军 张彤

前一阵子，不少报刊上都登载了一些文章，介绍如何用高密驱动器把低密盘格式化成 1.2M。经过我们反复试验，结果并不理想。一般的软盘格式化时有相当多的坏道，即使勉强格式化成功，存储的数据也不稳定。现在，已经很少有人愿意把低密盘格式化成 1.2M。

如何在确保数据安全的情况下，有效地扩充软盘容量呢？下面我们将配合两个工具程序介绍扩充的方法。

(一)400K (在低密驱动器上把低密盘格式化成 400K)


```

-A100 ↓
XXXX:0100 MOV AX,0201 ↓
XXXX:0103 MOV BX,0200 ↓
XXXX:0106 MOV CX,0001 ↓
XXXX:0109 MOV DX,0080 ↓
XXXX:010C INT 13 ↓
XXXX:010E INT 20 ↓
XXXX:0110 <CTREL>+<C>

```

-NA:BOOT ↓

-L200 ↓

$$-G = 100 \downarrow$$

Program terminated normally

 $\rightarrow Q \downarrow$

硬盘主引导扇区存放着硬盘分区情况表,以及引导硬盘活动分区的引导程序。分区情况如下(整个硬盘共可存放4个分区,前面的十六进制数字为在引导扇区中的位置):

一般的低密盘格式化后有 360KB, 共 40 个磁道, 每道 9 个扇区。目前的软盘一般来说多格式化几个磁道, 并不困难, 使用上也沒麻烦。400K 这个工具程序就是出自这种思想, 把普通低密盘格式化成 40 道, 每道 10 扇区。

400K.ARC 中包含三个程序, 即 400KDISK.BIN, 400KFORM.COM, READ-ME.NOW

使用方法是：

①把 400KDISK.BIN 复制到硬盘根目录或系统盘的

标志,即 1FEH 中的 55H,1FFH 中的 AAH,若此两字节不是此内容,硬盘即无法使用,通过改变这两个字节的内容可对硬盘进行简单的加密。

2. 硬盘分区引导扇区和软盘引导扇区

硬盘分区引导扇区和软盘引导扇区内容大同小异,这里放在一起介绍。存取这样的引导扇区可用 DEBUG 和 PCTOOLS,为了得到反汇编程序,就只能用 DEBUG 了。存取方法如下:

C>DEBUG ↓

-L100 X 0 1

其中 X 为 0、1、2..., 0 为 A 驱, 1 为 B 驱, 2 为第一个硬盘等。

引导扇区的第 0、1 字节为转移指令,第 2 到第 A 字节为版本注释,从 00B 到 035 为磁盘状态信息。1FEH、1FFH 中为磁盘活动标志 55H、AAH。磁盘状态信息见下

00B~00C 每扇区字节数(低字节在前,下同):

00D 每簇扇区数:

00E~00F 引导块所占扇区数;

010 FAT(文件分配表)个数:

011~012 磁盤上最大可放文件數目;

013~014 磁盘总扇区数:

015 磁盘性质: F8 — 360K, F9 — 1.2M, F10 — 5.25M, F11 — 硬盘等;

016~017 FAT 所占扇区数:

018~019 每柱(道)扇区数;

01A~01B 磁盘头数.

由上表可以看出, 决定磁盘容量的是扇区字节数和磁盘总扇区数的乘积, 由于 PC-DOS 每扇区为 512 字节, 因此, 整个硬盘分区的最大容量为 $FFFFH \times 200H = 33554432$ 字节, 也就是大约 32M 字节。

由于篇幅限制,主引导扇区和分区引导扇区的引导程序在这里就不作注释了,欢迎大家自己去阅读,这样可以提高编程水平。因为这些程序都编制得十分巧妙。

1BE	分区 1	开始	引导指示符	头	扇区	柱面
1C2	分区 1	结束	系统指示符	头	扇区	柱面
1C6	分区 1	相对扇区	低字	高字		
1CA	分区 1	扇区数	低字	高字		
1CE	分区 2	开始	引导指示符	头	扇区	柱面
1D2	分区 2	结束	系统指示符	头	扇区	柱面
1D6	分区 2	相对扇区	低字	高字		
1DA	分区 2	扇区数	低字	高字		
1DE	分区 3	开始	引导指示符	头	扇区	柱面
1E2	分区 3	结束	系统指示符	头	扇区	柱面
1E6	分区 3	相对扇区	低字	高字		
1EA	分区 3	扇区数	低字	高字		
1EE	分区 4	开始	引导指示符	头	扇区	柱面
1F2	分区 4	结束	系统指示符	头	扇区	柱面
1F6	分区 4	相对扇区	低字	高字		
1FA	分区 4	扇区数	低字	高字		

以上引导指示符占 1 字节(8 位), 头占 8 位, 扇区占 6 位, 柱面占 10 位, 引导扇区最后两个字节存放着硬盘活动

根目录下，并在 CONFIG.SYS 配置文件中加一行：

DEVICE=400K.DISK.BIN

②使用 400K.FORM.COM 格式化软盘，键入命令：

400FORM A: /E

不到一分钟就可把软盘格式化成 400K。

这样做好后，你还必须清楚以下问题。这里假定你有两个 360K 的软驱和一个硬盘。

A 指一个普通的 360K 软驱

B 指一个普通的 360K 软驱

C 指硬盘

D 指一个 360K 和 400K 软驱 (E 是 B 驱的另一个逻辑盘号)

访问 400K 的软盘请用 D 盘和 E 盘。

试与一般格式化的盘比较一下优劣：

360K 的软盘 400K 的软盘

360KB 的磁盘空间 400KB 的磁盘空间

格式化约需 3 分钟 不到 1 分钟

访问磁盘速度慢 相当于前者的 3 倍

(二) 800K (在 高密驱动器上把 低密盘格式化成 800K)

360K 的盘共有 40 道，每道 9 个扇区。高密驱动器可提高软盘道密度，即把软盘格式化成 80 道，位密度基本上不变。这样，就可格式化出 800K。

800K.ARC 中包含三个程序，即 800K.COM，800K.DOC，800K.FORMAT.COM。

使用 800K.FORMAT 与 DOS 的 FORMAT 方法一致，如：

800K.FORMAT A:

800K.FORMAT 将把软盘格式化成 80 道，每道 10 扇区。BOOT 扇区中包含了修改后的 BIOS 磁盘参数块，并将 1 份 FAT 标志成坏簇。

一旦用此法格式化完毕，在版本高于 3.0 的 DOS 环境中使用这类磁盘，从原理上讲，没有问题。然而，一些 BIOS ROM 程序拒绝这类特殊操作。为了解决兼容性问题，800K.COM 这个 TSR 驻留内存的程序中断 13H，接管所有的磁盘读写调用。在几种机器上试用都通过了。

在 A 盘上安装 800K.COM，则键入命令 800K / A 在两个软驱上安装，则键入命令 800H / A / B 拆卸该程序 800K / U。

你可以先试试你的 ROM BIOS 是否可以直接读 800K 的盘。如果能够，那将是相当舒服。作者在 Big Blue! 版的 BIOS 上 PASS! 作者发现在标准的 AST 机上也可直接使用。

这两个小工具在软盘上压缩存储，使用时请用随盘带的 UNPAOK 程序展开，即 UNPACK 800K，UNPAOK 400K。

这两个小工具安全有效地扩展了软盘容量，推荐大家使用！

DIGGER 画面解析

天津市海洋技术研究所 宋令本

digger (淘金者) 是 PC 及其兼容机上传播最广的游戏。它的结构严谨、音乐活泼有趣，特别是它能给操作者留以设计和思考的余地，令人百玩不厌。

digger 也有不尽人意的地方，那就是画面有限，当把所有面都过去以后，它就会把已玩过的画面再拉回来，虽然其它参数都已改动，但仍令人有些扫兴。这里笔者就介绍一下 digger 的画面结构，使读者可以根据自己的想法设计出成百上千种不同的画面来。

用 PCTOOLS 找到 digger.com 文件的第 108 扇区 (有些为 123 扇区)，可以看到由在 0112 (0070) 这行尾部 S 字母开头的、由大写字母 S、B、H、V、C 和空格组成的字符块，这就是 digger 画面的数据区。这里 S 代表圆洞，H 代表水平通道，V 代表垂直通道，B 代表金袋，C 代表钻石，空格则是未开挖的部分。digger 画面是由 15×10=150 这样的元素组成的。所以画面数据区也就是由几个这样 150 组成的。读者可以先在纸上设计好画面，然后译成字母，填入数据区即可。例如要将 150 个空格填入数据区开头，则第一面便成了空白；填入 300 到 450 的位置则第三面成了空白，另外，每面至少有一个钻石 (C)，至多有七袋金子 (B)。

小资料

adapter 适配器

adapter plug 插接适配器

address 地址

address format 地址格式

addressing capacity 寻址能力

addressing mode 寻址方式

address path 地址路径

address space 地址空间

auto loader 自动装入程序

autodoping 自动掺杂

autoindexing 自动变址

automata theory 自动机理论

automatic coding 自动编程

automatic error detection 自动错误检测

automatic interrupt 自动中断

automatic loader diagnostic 自动装入诊断程序

automatic programming 自动编程

automatic reset 自动复位

automatic sequencing 自动排序

软件库

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

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

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

三、订购软件清单的每个软件后, 请注明片数及单价。

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

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

六、汇款: 广州市石牌华南师范大学内电脑杂志社收。

注: 方括号内数字为片数, 圆圈内J指游戏杆, K指键盘。

工具类

IT123 硬盘管理实用程序库Hard Disk Utilities (1) 这个程序库收集了20多个用作硬盘管理的实用程序, 其中包括修改文件/目录属性的ALTER程序; 显示硬盘上未有备份文件的BACKSTAT程序; 按序重新整理目录程序CATALOG; 进行隐藏目录操作的三个程序: 进入秘密目录程序CDSECRET, 创建秘密目录程序MDSECRET, 删除秘密目录程序RDSECRET; 硬盘读写磁头安全定位程序DISKPARK; 恢复丢失扇区程序; 显示硬盘空闲区程序; 菜单式拷贝程序GCOPY, 菜单式删除程序GDEL; DOS命令编辑程序NDOSEDIT; 置文件属性为只读程序READONLY; 还原文件属性为可读写状态程序READWRIT; 按指定型式和不同格式列文件目录程序SDIR5; DOS命令路径重置程序SEARCH; 显示目录树程序TREED; 删除文件和恢复被删除文件程序VDL和VNDEL; 磁盘扇区检查程序VIEWDISK; 在盘空间所有路径查找文件程序WHEREIS; 拷贝“只读”文件程序WRITE; 建立只读文件程序WRTP; 日志程序DIARY; 口令程序PASSWORD等。以上内容丰富的硬盘管理程序使用方便, 功能齐全, 是你工作上的好帮手。

IT124 数学计算器CALL1.01版(1) 在您使用PC机时, 有时需要进行一些数学计算的工作, 你也许在此时会想到, 桌旁再放一台小型的计算器该多好啊! CALL软件将帮您的忙, 这是一个功能齐全的数学计算器, 运行时您可以按照屏幕提示, 在多功能的窗口进行数学计算, 成为您得心应手的工具。

IT125 压缩软件PKARC及解压还原软件PKXARC (3.6版) (1) 当你需要节省存储空间和处理时间, 你可以使用压缩软件, 尤其是在利用电话线, 通过Modem进行远距离传送时, “时间就是金钱”, 为了节省费用, 你最好是先将你要传送的文件(直至整个磁盘内文件)进行压缩后(能将原文件压缩30%~60%)再作传送, 接收者只要用解压还原软件就可进行解压还原, 恢复原文件, 使用时只要按下PKARC回车或PKXARC回车, 屏幕就会展现出压缩(或解压)操作的帮助菜单, 其操作十分简单。本软件是PC用户所必备的工具软件, 读者不妨一试。

IT126 PC绘图软件(PC-GRAPH) (1) 30元/片 这是一个可利用由PC-FILE产生的数据库或报表文件的数据在屏幕上用用户所指定的区域绘制出相应图形的软件。该软件必须与PC-FILE数据库系统配合使用, 需要彩色显示器及IBM或与IBM兼容的彩色/图形显示卡, 最好能有一台可打印图形的打印机(如无, 则不能打印图形)。

IT127 全屏幕编辑器与具有扩展功能的电子表格软件(1) 这是一个包含有一个全屏幕编辑器AE和一个电子表格的软件, 使用它用来进行源程序的编辑及英文文字处理, 具有方便、实用的优点, 同时该软件内还包括电子表格程序AC, 用它来制作有关数字的表格, 显得十分便利。

IT128 菜单生成系统CHOICE 这是一个多功能的菜单生成程序, 它能设置、显示菜单, 在菜单下可以运行批文件、外部程序和DOS命令, CHOICE允许每一个菜单包含32个项目, 允许在命令项中使用变量, 该软件设置有安全保护措施, 你可以利用这个软件, 重新编排您的磁盘上的程序, 并能以菜单的方式管理它们。

IT129 PC通讯软件COMMIX 这是一个常驻内存, 为PC用户提供完整的通讯功能的软件, 它可通过电缆, 将两台PC机连接起来, 利用该软件实现通讯及打印机共享, 还可利用Modem(调制解调器)通过电话线或专线, 实现多台PC机的远距离通讯。

IT130 激光打印机驱动程序(LSU5.02版) 这是一个专供使用激光打印机的实用程序, 它可以为激光机设置各种不同参数, 并能提供多种不同字体供激光机使用。

IT131 图形符编辑器ICON MAKER 这是一个专门用于编辑图形符的软件, 与软件库中IT103软件各有特色, 读者不妨一试。

IT132 打印控制软件LIST V1.0 LIST是程序设计的一个有用工具, 它能帮助程序员打印一个符合要求的文件, 包括自动标记页码、页数、标题, 始末打印页以及其他一些要求的类型特征, 该软件用于打印程序源码是十分便利的。

IT133 电视化学价游戏 Video chem Video chem是使用编译BASIC为中学生设计学习化合价(chemical Valences)的教学游戏软件, 它包括三大部分: 概述原子基本的知识的Introduction; 引入化合价概念的Practice Lab; 选择原子进行实验的VCHEM, 确定原子的分类的SLAB及其化合价的LAB, 该软件是菜单驱动, 容易使用。

IT187 PCTOOLS V7.0最新版, 功能最强。(高密6张/300元)

- BT9 EE SYSTEM V2.0 300元一套[10]
- BT10 TANGO 电路制作 300元一套[9]
- BT11 TANGO 电路制作 330元一套[11]
- BT13B 汉化AUTO CAD 9.03 制图软件(要8087处理器)[10]
- BT14 MATHCAD V2.0 数学作图软件(35元/2片)[2]
- BT15 MAPEDBT V1.0(划地图用)[2]
- BT33 英语翻译助手有说明盘(350元/套)[14]
- BT36 TURBO LIGHTING 英语单词拼写检查[2]
- BT37 GRE 研究生英语考试[4]
- BT123 TOEFL 英语[10]
- BT54 WINDOWS V3.00 AT机, 386机用, 有多种管理功能[19]
- BT65 LQ1600K 配 AUTOCAD 9.03 或 2.6 以上作图的打印驱动程序[1]
- BT71A PCTOOLS V6.0 最新版, 有多种功能(120元/套)[6]
- BT123 TURBO ASSEMBLER V2.01 汇编语言(60元/2片)[2]

智通电子技术公司

(CCS IBM PS/2 中国南方维修中心)

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

公司竭诚为各界用户服务, 并提供:

- 一、负责对购买 AS/400 小型机的用户进行出国培训前的初级培训。
 - 二、人材培训, 技术咨询, 各种计算机资料和软件。
 - 三、3⁺以太网, Novell 网, Token Ring 网, D-Link 网等计算机网络系统的软、硬件开发、安装、销售。
 - 四、工业自动化系统, 遥测遥控系统, 电子工程等软、硬件开发、研究、设计、工程。
 - 五、手写汉字识别系统, 汉字文本识别处理系统, 汉字排版及轻印刷系统。
 - 六、中国科学院的各种新产品, 各种进口元器件、数字仪表、通讯设备、示波器等。
 - 七、销售美国 AST, COMPAQ, PS/2 及各种兼容 486, 386, 286, PC 微机, AS/400 小型机, IBM RISC SYSTEM/6000 工作站等。
 - 八、高抗干扰电源, UPS 不间断电源, 各种打印机、中英文打字机, 各种电子电脑配件。
- 公司地址: 广州市先烈中路 100 号大院内电子研究所一楼

公司电话: 753245-4131、1107, 775600-610

邮政编码: 510070

图文传真: 753247

联系人: 李海华, 姚智鹏, 李建

门市部地址: 广州市天河科技街第二座

171/176 号

门市部电话: 510450

邮政编码: 510630

联系人: 欧德

