

# 電腦 1990

080330

中国软件行业协会会刊

## CWY系列交流參數 高抗干擾穩壓器

CWY FAMILY PARAMETRIC  
AC POWER REGULATOR

商標

全國範圍產品責任保險



三相系列



單相系列

高抗干擾  
安全可靠  
功能特殊  
電腦必備

廣州白雲山集團公司

廣州白雲山電源設備廠

廣州市沙河同和 郵政編碼：510515 電話：705764 —23 705665 —343 電掛：0839

# 广州白云山电源设备厂

## CWY系列高抗干扰稳压电源

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

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

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

本产品已向中国人民保险公司办理了全国范围产品责任保险，用户可放心使用。

### 单相抗干扰稳压器系列

#### 一、型号及规格

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

#### 二、主要技术参数

●输入电压单相交流220V50Hz

●电压稳定度

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

全國範圍產品責任保險



●无过压危险，有自动短路保护特性

●总谐波失真度 $<4\%$ （开关式电源负载）

●对电网振铃干扰或尖脉冲干扰扼制能力符合国际计算机电源要求

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

●有效功率可达到80~90%

●音频噪声低，位于1.6米距离处50dB

### 三相抗干扰稳压器系列

一、规格：3KV、6KV、10KV、15KV、30KV

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

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

●音频噪声低：位于1.6米距离处52db。

●效率高：89%。比一般电子交流稳压器效率高 $1/5$ 。这对节能很有意义。

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

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

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

中国科学院广州电子技术研究所

## 智通电子公司

(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<sup>+</sup>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, AS/400 小型机在国内的销售和维修业务。

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<sup>+</sup>以太网, Novell 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、美国 AST, COMPAQ, PS/2 及各种兼容 486, 386, 286, PC 微机, AS/400 小型机等。
- 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 号大院内电子所实验楼五楼

公司电话: 753245-4131, 775600-610

邮政编码: 510070

图文传真: 753247

联系人: 李海华, 姚智鹏

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



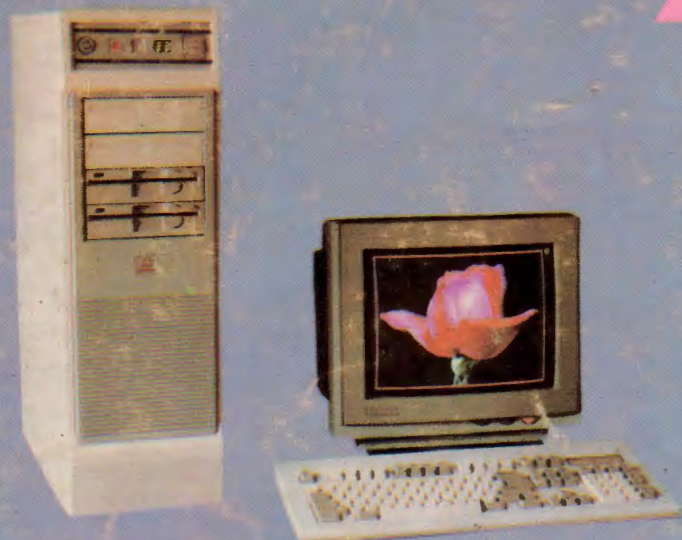
**AD-386/33型 零售价32000元**

- 80386-33 CPU 2 M RAM 64K 高速缓冲
- 1.2M+1.44M软盘驱动器
- 100M硬盘
- 1024×768彩色显示器及VGA卡
- 软件、资料各一套
- AD-386系列机有25型、20型、16型等，配置与上相同

零售价：AD-386/25型 28500元

AD-386/20型 27500元

AD-386SX/16型 19900元



**AD-286/16型 零售价14800元**

- 80286 CPU 1M RAM
- 1.2M+360K 软盘驱动器
- 40M硬盘
- 1024×768彩色显示器及VGA彩卡
- 软件、资料各一套



本公司讲求信誉，用户第一，AD-386系列机全部免费保修两年，AD-286系列机全部免费保修一年。本公司售出机器除免费保修期间之外，其余均终身保用，如有维护，只收零件费用。本公司在南京、长沙、北京、哈尔滨都设有代理点。

## 广州先锋电子经营部

地址：广州市天河路33号之七

电话：752397 邮政编码：510075

经理：柯庆

# 广州市科教电脑商场微机配置及价格一览表

TP3  
57

机 型 配 置	PC		XT		说 明
CPU	8088V20	8088V20	8088V20	8088V20	可插 8087
频率	4.77 / 13MHz	4.77 / 13MHz	4.77 / 13MHz	4.77 / 13MHz	机箱按钮加速
内存	640KB	640KB	640KB	640KB	可扩至 1MB
软驱	360K × 2	360K × 2	360K × 2	360K × 2	日产 TEAC
硬盘			20MB	20MB	可选 40MB
显示	高 / 中分单显(双频)	CGA640 × 200	高 / 中分单显(双频)	CGA640 × 200	单显 12", 彩显 14"
串 / 并	2 / 1 多功能	2 / 1 多功能	2 / 1 多功能	2 / 1 多功能	五功能
键盘	101 键	101 键	101 键	101 键	XT / AT 两用
价格	3500 元	5800 元	5300 元	7600 元	
机 型 配 置	CT286				
CPU	80286-16	80286-16	80286-16	80286-16	可插 80287
频率	10 / 22MHz	10 / 22MHz	10 / 22MHz	10 / 22MHz	按钮加速、液晶显示
内存	1MB	1MB	1MB	1MB	可扩至 4MB
软驱	1.2M+360K	1.2M+360K	1.2M+360K	1.2M+360K	可选 1.44M3.5"盘
硬盘	20MB	40MB	40MB	80MB	可选 100MB 以上
显示	CGA640 × 200	EGA640 × 350	TVGA1024 × 768	TVGA1024 × 768	可选 VGA800 × 600
串 / 并	2 / 1	2 / 1	2 / 1	2 / 1	可选四用户卡
键盘	101 键	101 键	101 键	101 键	AT / 386 两用
价格	11000 元	13000 元	13700 元	15700 元	
机 型 配 置	386SX		386DX		
CPU	80386SX-16	80386SX-16	80386-25	80386-33	可插 80387
频率	15 / 25MHz	15 / 25MHz	20 / 33MHz	20 / 58MHz	按钮加速、液晶显示
内存	2MB	2MB	2MB	2MB	可扩至 16MB
软驱	1.2M+360K	1.2M+360K	1.2M+360K	1.2M+360K	可选 1.44M3.5"盘
硬盘	40MB	80MB	80MB	100MB	任意
显示	EGA640 × 350	TVGA1024 × 768	TVGA1024 × 768	TVGA1024 × 768	任意
串 / 并	2 / 1	2 / 1	2 / 1	2 / 1	可选四用户卡
键盘	101 键	101 键	101 键	101 键	386 / AT
价格	18000 元	20000 元	22000 元	25000 元	

地址：广州市天河东路商业街东座 25-27 号

通信：广州市 1753 号邮政信箱

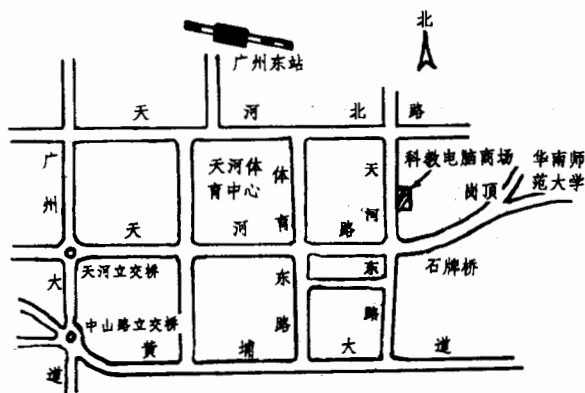
邮编：510620

电话：511197、755147

电挂：8360 FAX：511197

开户银行：广州中国银行天河支行

帐号：271-015110027



# PC-1500 机镍镉电池组大量供货

本中心有大批原装法国电池，经本中心监制组装成套，专为 1500 机用户服务。

电池组有三种规格：500mA，600mA，700mA，用户可任选其中一种，安装时，打开 CE-150 打印机后盖，取出原镍镉电池组，换上新买的电池组即可使用。

5GNY-500	500mA / 小时	70 元 / 组
5GNY-600	600mA / 小时	80 元 / 组
5GNY-700	700mA / 小时	90 元 / 组

注：邮购者免收邮费。

经实际试验表明 5GNY-700 电池组较原 1500 机内新的电池组，连续打印时间长 30% 左右，比用手工焊电池组要长 50% 以上。

广州市东风东路 745 号 **广州袖珍机技术服务中心**

电话：(020) 751025-235      电挂：5051      邮码：510080

银行及账号：广州建行天河支行环市东办    208-2612123-72





# GDMIS 信息管理网络

中山大学计算机科学系 罗源明

引言:

GDMIS 信息管理网络系统是在中山大学软件研究所所长姚卿达教授的主持下完成的。本文在他的倡议、指导下写成,借此机会对姚教授表示衷心的感谢。

香港某集团有限公司是一个大型企业,集团总公司下属有多个部、室和四十多家子公司、孙公司。他们主要经营地产、金融、化工、机械、电子、纺织、土产、陶瓷、粮食、蔬菜、水产和旅游等十几个行业的许多种商品,内营外销的商业活动规模大、范围广,且直接面对复杂多变的港、澳及海外市场。为了提高企业的现代化管理水平,从根本上提高企业的素质,促进商品的流通和资金利用,降低消耗,提高效率,创造企业更高的经济效益和社会效益,在集团公司中开发了 GDMIS 信息管理网络系统,它一方面用于处理集团内部的数据和各种资料,另一方面又可以和国际商业信息网连接,采集各种商业信息,供经营决策之用。该系统开通运行半年多来,得到了用户的认可和好评,对企业的经营活动起到很好的促进作用。实践证明,该系统的开发是成功的。由于篇幅限制,本文仅讨论该网络的设计、结构、主要技术指标和功能。

## 1 网络设计的原则

在设计 GDMIS 信息管理网络系统时主要遵循下述原则:

### 1.1 企业的管理模式

GDMIS 信息管理网络既照顾到企业的现行管理模式又考虑到其今后的发展和改变。企业的现行管理体制主要是垂直方向的管理。在集团总公司的组织、领导、协调下由各部、室、子公司、孙公司自主经营,独立核算,横向联系较少,是一种集中与分散相结合的管理模式。信息的采集、加工处理、存储、报表制作、传递绝大部分都是手工进行的。这种模式显然不能适应企业的现代化管理的需要,不能适应企业的日益发展的经营活动,必须完善、改进其信息处理及管理的体制和方式。建立企业的计算机网络系统,就能很好地提高信息处理的能力、速度和质量,加强集团企业各部门、各子公司、各孙公司之间的联系,增进企业内部的了解,有利于发挥企业的整体力量,增强企业的活力。近期还能对市场预测、企业管理、经营决策提供信息和方法,从长远考虑,可以建立以文本库、方法库、数据库、模型库和知识库为基础的新一代决策支持系统,从根本上改善企业的管理和经营,促进企业的发展。

### 1.2 信息流向与信息流量

网络系统必须适应企业信息流量和信息流向的要求,这是又一个必须遵循的原则。企业的信息流向以纵向为

主,即集团总公司 $\leftrightarrow$ 子公司 $\leftrightarrow$ 孙公司之间是主要流向。而各部室之间、各子公司之间、各孙公司之间的横向联系现时还不密切,今后将得到适当的加强。企业的信息流量从总的来看属中等程度,流量并不是很大,但又有相对集中的时候,例如每个月的月底(或月初)要逐级呈报各种月报统计数据,又如每天的早上和下午要发布或查询“天天行情”、“商业行情”和“金融信息”,象这些时候,网上的信息流量就较为集中。信息流量的另一个特点是实时性的查询和传递。例如总经理需立刻查询某客户的有关资料要马上知道某方面的市场行情,供分析、决策之用时,总经理就在自己的网络终端上打入查询命令,系统要能立刻响应,提供有关的资料。

### 1.3 标准化

采用国际网络标准协议,使系统能与国际商业信息网连接,可以接入 X.25 公共数据分组交换网和 IBM 的 SNA 网。

### 1.4 开放式

系统采用标准的协议和接口,使系统扩充方便,可以接入不同机种,如 PRIME、VAX、MV 等。

### 1.5 连结现有的计算机资源

集团总公司的各部、室、大部分的子公司近几年来都先后购置了以 IBM-PC/XT、AT 为主体的微机系统,并在本单位、本部门的一些专项业务中开发了应用软件。新建的网络系统必须尽可能地让这些微机系统都能进网,原先开发的软件基本上能继续运行,不会造成浪费。

### 1.6 保密和安全

网络资源的保密和安全是网络设计中必须认真考虑的一个方面。GDMIS 信息管理网络系统面对错综复杂的外部环境,所以保密性和安全性是用户特别关心的问题。我们从网络的拓扑结构、连网技术、网络资源的配置、网络的管理、用户的登录、用户的权限等多个方面采取强有力的措施,力求网络资源有较好的保密性能,力求网络资源安全可靠。

### 1.7 适应投资的规模

企业给 GDMIS 信息管理网络系统的首期专项投资金额是有限的。网络系统既要适应企业的近期发展的需要,又必须具有较好的可扩充性和向上兼容性。随着企业经营的发展,业务范围的拓宽,对网络系统的投资也会逐渐增加,网络就要能很方便地扩充。根据以上的考虑,GDMIS 信息管理网络系统采用图一所示的网络结构。

## 2 网络的基本结构

网络的结构基本上可分为两大部分:一是 Ether (以太网)部分,二是集中型的微机网络部分。



总公司的电脑中心由 PRIME 2350 和 PRIME 6150 小型机以及 4 个 LTS 300 和两套 AT 机连接为 Ether (以太网) 结构, LTS 300 再连接集团总公司的部、室的微机系统。两台小型机的总的磁盘容量有 4000MB, 另有两台磁带机, 足以存贮企业需要的信息。各子公司的微机系统则通过 MODEM (调制解调器)、程控电话交换机、ICS 3 和 PRIME 6150 主机相连接。

### 2.1 局部网主机控制器 (LHC 300)

LHC 300 (LAN HOST CONTROLLER 300) 是一种把计算机联结到网络上的设备。它是一种智能设备, 能够执行程序, 分担主机的通讯作业。当它运行 NTS (Network Terminal Service) 软件时, 可支持 128 个终端设备。LHC300 采用 80186CPU, 其内存容量有 64KB-128KB。

### 2.2 局部网络终端服务器 (LTS 300)

LTS 300 (LAN Terminal Server 300) 是一种网络接口设备, 它能接 8 个异步通讯的 RS-232c 接口到 IEEE802.3 局部网络 (LAN) 上, 该局部网络采用 CSMA/CD 传输规程。LTS 300 既可以使用 NTS 协议又可以使用 TCP/IP 协议, 当然不能同时使用两种协议。NTS 协议允许 LTS 300 与 PRIME 的 50 系列机连接, 该 50 系列机配置为网络终端服务器并与 LAN 连接, NTS 允许一台主机有效地与多个终端连接; TCP/IP 协议的效率比 NTS 协议低, 但它允许 LTS 300 不仅与 50 系列机连接, 而且可以和其它 PRIME 机器连接, 甚至与其它支持 TCP/IP 协议的机器连接。

### 2.3 智能通讯子系统 (ICS 3/2)

ICS (Intelligent Communications Subsystems) 是一种智能通讯设备。其中 ICS 3 是以一个 16 位的 Z-8001 微处理器为基础的控制装置。它支持异步通讯, 支持一组异步通讯线同时工作。在 ICS 3/2 上能运行程序, 分担主机的通讯业务, 以利于减轻主机的负担, 提高主机的资源利用率。

### 2.4 远程结点 (RN)

RN (Remote Node) 是一种远程结点。它是各子公司的 IBM-PC/XT、AT 或其兼容机通过符合国际标准和香港电话公司确认的 MODEM、程控电话交换机与 PRIME 6150 的 ICS 3 相连接, 构成集中型网络的结点。在远程结点上通过运行 CTEMu-p 或 DATACOM 汉字通讯与仿真软件, 就能使该微机系统与主机通讯或成为主机的智能终端。CTEMU-p 是用 IBM-PC 汇编语言设计的中、英文通讯与仿真软件, 它可以在 PC-COS 和 CCDOS 操作系统的支持下运行。其主要功能有下述几个方面:

(1) 能把 PC 微机仿真成远地的 PRIME 6150 主机的中文或英文终端, 并可以使用 CCDOS 支持的任何一种汉字输入方法。

(2) 能用自动拨号方式 (ALT-C) 与远程主机建立通

讯连接, 也可以用人工拨号的方式 (ALT-G, ALT-D) 与远地的指定的电话号码线路建立连接, 在人工拨号时, 还可以由人工设定信息块的大小及传输速率。

(3) 可以自动拆除与远地主机的电话线路的连接。

(4) 支持与远地机器间的文本文件和二进制文件的传送。

(5) 支持 PC 机与 PC 机, PC 机与其它机器间的通讯和文件传输。

(6) 用户还可以根据自身的需要, 增加数据格式转换与加密传输功能, 用于不同场合的数据采集和通讯。

### 2.5 TCP/IP AT 机网卡

#### 3 网络的主要技术指标

网络的拓扑结构: GDMIS 信息管理网络采用总线型与集中型相结合的拓扑结构

通讯方式: 双线全双工, MODEM (自动应答)

波特率: 1200-2400BPS

总线传输速率: 10MB/s

传输规程: CSMA/CD 并支持 CCITT X.25 (公共数据网), IBM SNA (远程)

最远传输距离: 数百公里

上网工作站: 几十个

代码体系:

ASCII (英文)

GB 2312 (汉字国家标准, 用 BIG-5 繁体码的可转换)

公司代码

会计科目代码

贸易统计代码

客户代码

智能终端: IBM-PC/AT, 屏幕 9/25 行, 标准的 RS-232C 接口, 可运行 CCDOS 及各种汉字软件

#### 4 网络的主要功能

GDMIS 信息管理网络的主要功能有下述几个方面。

##### 4.1 数据采集电脑化

由于总公司的电脑中心和大部分的子公司的微机系统构成一个计算机网络系统, 只要在中心主机和远程微机结点上运行相应的软件, 就能实现数据 (信息) 采集电脑化, 使数据的采集准确、快速和规范。

##### 4.2 实现数据的电子传递

由于 GDMIS 网络系统支持结点和主机间的文本文件和二进制文件的传送, 就实现了企业内部数据的电子传递, 就能使企业在经营和决策过程中传递必要的、规定的各种数据时快速、准确、可靠。

##### 4.3 提供良好的资源共享

建立起完善的计算机网络, 支持统一的、集中与局部处理相结合的数据库及其应用系统, 实现软件资源和数据资源的良好共享, 减少软件开发的重复劳动, 节约投资,

# 微机网在发射机状态监控中地线隔离的实现

华南理工大学计算机系

李仲麟 刘国

## 摘要

本文就一个具体微机网在发射机状态监控中如何进行地线隔离为例子，阐述了微机应用于工业控制中地线隔离的必要性，介绍了利用光电耦合方式进行电信号模拟量、开关量隔离的实现方法。

微机应用于工业控制，应该重视对地线的处理。尤其是多机联网，在系统设计时必须把它作为一个重要的因素加以考虑。这个问题不解决好，将直接影响到整个系统的可靠性。这里以一个对发射机状态实施监控的微机网为例，介绍一些地线隔离的实现方法。

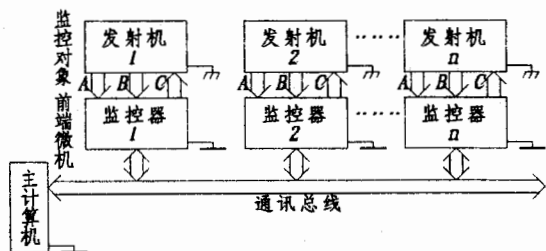


图1 系统示意图

上图是以实际系统为背景而画出的简图。仅就监控对

象而言，除了几十台型号各异的发射机外，实际系统还包含有发射天线交换器，(通讯)线(发射)机交换网，电屏、温度、频率测量仪等等。这里为讨论方便而又不失一般性，只画出主要的监控对象—发射机。

图中，发射机工作电压的参考电位用符号 $\phi$ 表示，即发射机地。计算机网工作电压的参考电位用符号 $\uparrow$ 表示，即计算机地。每个监控器都有8路模拟量输入(A)，24路开关量输入(B)和24路开关量输出(C)。

取自发射机的模拟量和开关量以电信号的形式作为监控器的输入信号，监控器不断地对发射机的工作状态进行监测。同时，监控器发出的控制信号送至发射机，实施各种控制。由于发射机与监控器之间有电信号传送，所以要考虑，发射机地与计算机地能否共地的问题。

监控对象的地能够直接地与计算机的地相连吗？这个问题对于工业控制中的微机应用有着普遍意义。显然，如果确认两个地是能够直接连在一起，那么电信号的传送是简单明了的。但要确认能否直接共地，必须对监控对象和现场环境进行了解分析。

发射机的工作电压高、电流大，机内某些工作点的电压至少在7000伏以上，工作电流达几十安培。特别是启动

提高数据资源的利用率，提高企业的经济效益。

## 4.4 联机查询检索

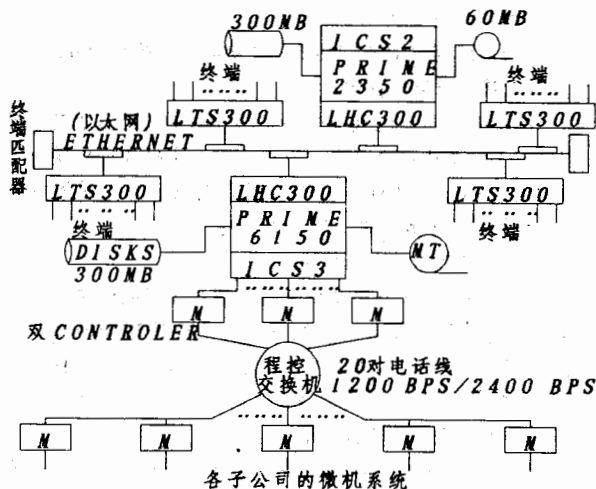
企业有了网络系统，就为企业内部的各单位、各部门提供多种信息服务，此外还能以适当的方式给企业外的单位提供某些有偿的信息服务。

## 4.5 与其它网络系统互连

GDMIS网络系统符合国际网络互联标准，因此可以方便地与大东电报局的DIALCOM系统、香港贸易发展局的HORTS以及英美的商业信息网络系统连接，从中吸收更多有用的来自企业外部的信息，为企业的经营和决策服务。

## 4.6 支持决策分析

企业网络的建立，可以不断地采集企业内部以及企业外部的经营行情，建立综合的数据库，为总公司和各子公司的管理和决策提供完整、准确、及时和有效的信息服务。在信息集中控制和管理的基础上，充分地利用信息资源，改变传统的落后的管理方式，实现现代化的管理与经营的方法。在市场预测、企业管理、经营决策等多方面提供有价值的信息，提供好的方法和模型。■



图一

机器上高压的瞬间,将产生非常强大的浪涌电流,整个发射台周围空间各种高频、工频干扰的场强也很大。

事实上,由于种种原因,各发射机的地并不是真正的大地电位,各发射机的地之间有电压差。当诸如开发射机,启动某些大负载机器,上高压等情况发生时,都会引起地扰动。若计算机地与发射机地简单地直接相连,那么不但各种高频、工频干扰会通过地线干扰计算机的正常工作,而且地扰动对计算机内工作电压只有几伏的IC来讲,其工作的可靠性势必受到严重的影响,甚至可能造成灾难性的损坏。计算机网内从主机到各监控器之间是用通讯总线互联的,网络中各点的地不一致将影响通讯的可靠性,更有甚者,网络中任意某一台监控器被浪涌电流损坏都有可能通过通讯总线波及整个系统。因此,计算机地与发射机地之间要严格地进行隔离,而计算机与发射机之间电信号的相互传送应采用耦合方式进行。

考虑隔离与不考虑隔离,从这两个不同点出发,将导致系统整体设计,硬件电路设计和软件编制产生不同的结果。要做到计算机与发射机彻底隔离,就必须把监控器与发射机的所有连线都隔离,这些连线包括模拟量输入线、开关量输入线、开关量输出线以及某些信号线、控制线和电源线等。

#### 模拟量输入线的隔离

每台发射机都有8个模拟量要送到监控器实行监测。这些模拟量是与发射机共地的。对模拟量输入线实施隔离的具体做法如下图所示。

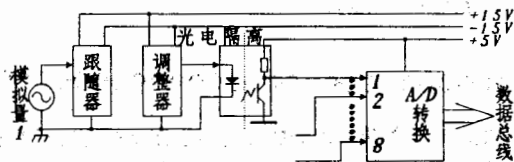


图2 模拟量输入线的隔离

可以看到,光电隔离元件的左部是与发射机共地的,而光电隔离元件的右部是接计算机地的。跟随器和调整器的工作电源是由监控器的两组(+15V, -15V)独立电源提供,这两组电源的地与发射机地相连。跟随器有很高的输入阻抗,主要是考虑到有些模拟量例如7000多伏的高压是经过阻值很高的电阻分压后获得的,若与模拟量连接的器件阻抗低,那么不但不能进行正确的测量,而且会影响发射机的正常工作。调整器有两个作用,其一是把模拟量不论是正信号或负信号都统一地变换成正信号,其二是把信号进行放大并调整到光电隔离元件的线性工作区。真正起到隔离作用的是光电隔离元件,它的工作原理是:当元件内的二极管有一定的电流流过时,二极管就发光,通过光对三极管发生作用,使三极管导通。实验表明,光电隔离元件内二极管电流与三极管电流在一定的区域内是线性关系的,但元件间的参数有一定的离散性,同时,由于光电隔离元件的工作过程是一个电—光—电的转换过程,

其工作频率受到限制,一般在几KHz范围之内。

显然,利用光的耦合作用来传送逻辑信号是完全没问题的,而且,只要电路设计合理,我们还可以充分利用光电隔离元件的线性工作区,达到既实现模拟量值的传送又方便调试的目的。在不降低系统性能的前提下,用光电隔离元件替代隔离运算放大器,将大大降低成本,收到显著的效益。

#### 开关量输入线的隔离

每台发射机可有多达24路的开关量要传送到监控器。对开关量输入线实现隔离的示意图如下:

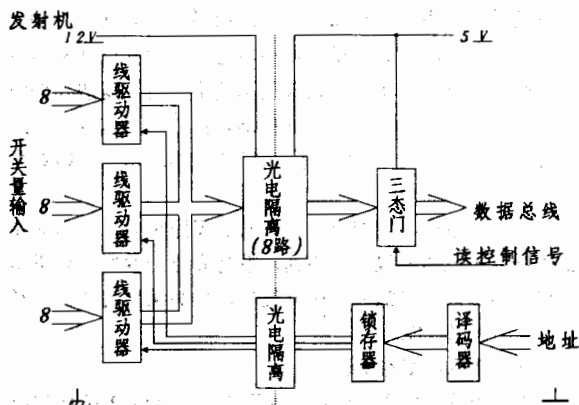


图3 开关量输入线的隔离

这个电路设计有两个特点:第一,24路输入共用8只光电隔离元件,而不是采用每路用一只光电隔离元件。显然前一种方法要比后一种节省元件。当输入的路数增多时,这个特点更为显著。以该系统的天线交换器来说,开关量输入为120路。若每路用一只光电隔离元件,那么就要120只,数目相当可观。事实上,微机总线一次只能读入其中8路。把所有的输入线按8路为一组,分成若干组,每次读入一组,这样做是合理的。至于每次读入是选通哪一组,可用片选信号来控制。当然,片选信号也必须进行隔离。第二,光电隔离元件的工作速度比起微机CPU的工作速度要低很多。在读入时,如果直接使用CPU发出的地址脉冲信号和读脉冲信号是不行的。解决问题的办法是:在硬件上,增加一个锁存器,把由地址信号经译码器所产生的片选信号锁存起来。在软件上,采用两次读入的办法。具体的读入过程如下:第一次预读。由监控器发出的地址信号经译码后产生片选信号,该片选信号被锁存起来,在地址信号撤消后,片选信号仍然保持有效。片选信号经光电隔离元件作用到某一片被选中的线驱动器上,于是对应的8路输入信号经过光电耦合反映到三态门的输入端。第一次预读的指令发出后,要延时一段时间,务必让输入信号真正到达三态门的输入端上。第二次真读。由监控器发出读命令,打开三态门,让输入信号经总线读入监控器内。

还有一点要提一下的是,所有的线驱动器不能使用计

# 利用电子计算机编译密码

陈庆祥

在现代社会中,密码通信被广泛地应用于各种重要情报的传输,它不仅在军事上发挥着重要的作用,在各种非军事领域里也大量采用密码来进行通信。

密码的质量,直接涉及其被破译的概率,对于保证通信内容的安全至关重要。直到二次世界大战结束前,编制密码的工作还主要依靠手工完成,因而易于被破译。例如中途岛海战后,东条英机召回山本五十六的密电被美国海军截获并破译,山本的座机起飞不到半小时,就遭到美国航空母舰上起飞的多架飞机的准确拦截,一声炮响,使这位曾经不可一世的“日本海军之神”的山本总司令立即葬身海底!

二次大战后,随着电子计算机的崭露头角,密码的编制质量有了飞速的提高。下面,就将介绍一种用计算机编制和解译密码的方法,它极其简单,然而编出的密码却具有极高的保密性能。

## 一、随机加密的原理

我们知道,任何字符输入计算机后,都以 ASCII 代码的形式存储在内存中(包括汉字内码,每个汉字以两个以上的 ASCII 代码表示。),由于 ASCII 代码是大家都认得的通用代码,因此,原文产生的 ASCII 代码是一组明码。怎样才能使明码变为无法辨认的密码呢?最简单的办法就是拿一组毫无规律的随机正整数与之相加,密码就编好

了。译解时只需要用同一组随机正整数与密码相减,密码就恢复成明码。

编制密码的方法是有了,对方接收密码后怎样译解它呢?这一组随机数是编制密码时,机器运行过程中“即兴”产生的,接收方当然无法预先约定,而且在发报时,也决不能连随机数也送过去,否则敌方会很快破译。因此,让接收方能准确地取得同一组随机数是解决问题的关键。为讲清这个问题,请先看下面一段程序的运行结果:

```
10 FOR I=1 TO 3:READ X
20 PRINT "RND("X")=" RND(X)
30 FOR J=1 TO 3
40 PRINT "RND("I")=" RND(I)
50 NEXT J:PRINT:NEXT I
60 PRINT "RND(O)=" RND(O)
70 PRINT "RND(-1.7E38)=" RND(-1.7E38)
80 DATA -1,-2,-1
```

```
RUN
RND(-1)=2.99196472E-08
RND(1)=.738207502
RND(1)=.272707136
RND(1)=.299733446
RND(-2)=2.99205567E-08
RND(2)=.273385388
RND(2)=.621966945
RND(2)=.369081192
RND(-1)=2.99196472E-08
RND(3)=.738207502
RND(3)=.272707136
RND(3)=.299733446
RND(0)=.299733446
RND(-1.7E38)=.253076494
```

计算机一方的 5V 电源。它们所需的电源必须另外提供。

## 开关量输出线的隔离

每台监控器有 24 路开关量输出线要连至相应的发射机。这些线的隔离如下图所示。

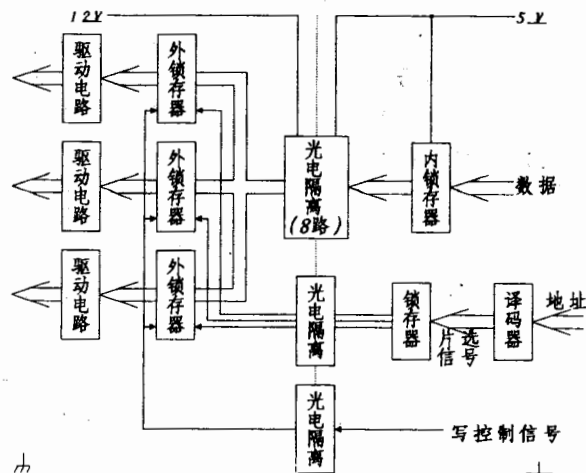


图4 开关量输出线的隔离。

这里,对开关量输出线的隔离与上面介绍的对开关量输入线的隔离有很多地方是类似的。首先,24 路输出是共用 8 路光电隔离元件;其次,输出的过程也是分两步走:第一步,用输出指令把输出信号先锁存在内锁存器,经过一段时间延时,让输出信号通过光电隔离元件到达外锁存器的输入端,同时片选信号也到达选中的外锁存器。第二步,发出专门的写控制信号,把输出信号锁在外锁存器,这些信号经驱动电路去控制发射机的开关机及进行各种工作方式选择等等。

值得注意的是,写控制信号也要进行隔离。此外,外锁存器还起到控制保持作用。因为它是由发射机供电的,即使监控器电源消失,也不会影响发射机的状态。 ■

## 小广告

请邮购 SHARP M<sub>2</sub>-731 计算机汇编及反汇编语言和中文说明书,价 38 元,增有 SHARP 公司未公开发表的 102 条指令,款到发货。

汇款请寄:南宁市第二水文队 黄学克(收)

邮政编码:530007



从程序 1 的运行结果可以弄清随机函数 RND (X) 的以下特性: (1) 当  $X > \Phi$ , RND (X) 的值是 (0, 1) 间的一个随机数; (2) 当  $X < \Phi$ , 得到的是一个固定的随机数序列 ( $X > \Phi$  时, TND (X) 的值与 X 大小无关); (3) 当  $X = \Phi$  时, RND (X) 值与才使用过的 RND (X) 值相同。由于不同的负数 X 可作为“种子”, 启动不同的随机数序列, 因此, 它完全可以作为密码钥匙, 解决刚才提出的问题。另外, 变动的随机数范围又是另一把密码, 发送和接收密码的双方, 只要记住预先约定的两个整数, 就可以取得完全相同的随机数序列, 进行密码的编制或译解工作。

## 二、随机密码的编译程序实例

```
5 REM 计算机密码编制/解译程序.
10 DIM X(255):ONERR GOTO 320
20 INPUT "请选择处理方式: 加密/译密 (1/2)"; W
30 IF W < 1 OR W > 2 THEN 20
40 INPUT "请输入密钥 M1, M2 (M1 < 1E38, M2 < 873)";
M1, M2
50 IF M2 > 872 THEN 40
60 K = RND (-M1):HGR2
70 ON W GOSUB 100,200
80 INPUT "是否继续? (Y/N)"; W$
90 IF W$ = "Y" THEN 20
99 END
100 REM SUB-1 (编制密码子程序)
110 INPUT "请输入原文: "; X$
115 PRINT "加密结果 (三位密码) 如下:"
120 FOR I = 1 TO LEN(X$)
130 X(I) = ASC(MID$(X$,I,1))
140 R = INT(M2 * RND(1))
150 X(I) = X(I) + R
160 PRINT RIGHT$(STR$(1000 + X(I)),3); " ";
170 NEXT I:PRINT
180 RETURN
200 REM SUB-2 (解译密码子程序)
205 PRINT "请输入密码:"
210 FOR I = 1 TO 4
220 GET B$:PRINT B$;
225 B = ASC(B$)
240 A$ = A$ + B$:NEXT I
250 R = INT(M2 * RND(1))
260 Y = VAL(A$) - R
270 Y$ = Y$ + CHR$(Y):A$ = ""
275 IF B = 13 THEN 290
280 GOTO 210
290 PRINT "译出原文结果是: ";
300 PRINT Y$:Y$ = ""
310 RETURN
320 PRINT:CALL -198
330 PRINT "选择处理方式或密钥, 密码有误!"
340 PRINT "请检查后重新开始.": GOTO 20
```

RUN

请选择处理方式: 加密/译密 (1/2) 1  
请输入密钥 M1, M2 (M1 < 1E38, M2 < 863):  
19901006, 397

请输入原文: 按 3 号方案执行

加密结果 (三位密码) 如下:

373	080	222	442	377	108	320	411
376	323	518	262	105	166	255	361
159	186	302					

是否继续? (Y/N) Y

请选择处理方式: 加密/译密 (1/2) 1

请输入密钥 M1, M2 (M1 < 1E38, M2 < 863):

6.1357942E37, 833

请输入原文: The enemy's fort fell into our hands.

加密结果 (三位密码) 如下:

589	572	591	131	295	482	432	754
867	060	168	137	721	576	619	311
419	901	871	878	119	606	385	724
567	493	459	882	927	274	854	132
588	181	492	670	208			

是否继续? (Y/N) N

RUN

请选择处理方式: 加密/译密 (1/2) 2

请输入密钥 M1, M2 (M1 < 1E38, M2 < 863):

19901006, 397

请输入密码:

373	080	222	442	377	108	320	411
376	323	518	262	105	166	255	361
159	186	302					

译出原文结果是: 按 3 号方案执行

是否继续? (Y/N) Y

请选择处理方式: 加密/译密 (1/2) 2

请输入密钥 M1, M2 (M1 < 1E38, M2 < 863):

6.1357942E37, 833

请输入密码:

589	572	591	131	295	482	432	754
867	060	168	137	721	576	619	311
419	901	871	878	119	606	385	724
567	493	459	882	927	274	854	132
588	181	492	670	208			

译出原文结果是: The enemy's fort fell into our hands.

是否继续? (Y/N) N

程序 2 比较简单而且结构清晰, 无需再作解释。下面仅对其功能与运行情况作简单介绍。

该程序可同时处理中、英、日、俄等文字及其他字符, 而且具有编制和译解密码的双重功能。运行第一次, 打入 1, 进入编码状态, 输入密钥数 19900922, 397, 原文为中文, 第二段文字为英文, 中文意义是: 敌军要塞已被我攻陷。(其第一密钥为 6.1357942E37, 是一个很大的数) 输出三位密码 (每个汉字占三码)。运行第二次, 输入 2, 进入译码状态, 再输入与先前完全相同的密钥和密码 (输完最后一个数字再按回车, 数字间空格照按), 就很快获得原文。

## 三、关于密码质量的讨论

用上面程序 2 编译密码十分简便、迅速, 而且具有极高的保密性能。我们简单地算一算, 密钥 M1 的种数是

# 如何在 NOVELL 网无盘工作站安装虚拟盘

深圳建博电子仪器公司开发部 廖文忠 周良军

虚拟盘是 DOS3.X 版本提供的。通过设备驱动程序 VDISK.SYS 或 RAMDRIVE.SYS 安装在内存的模拟磁盘驱动器。利用到 Novell 网无盘工作站上是非常有效的。它不仅为用户提供了一个可访问的本地工作盘，减少了与网络的通信，减轻了网络的负担，提高了用户的工作效率；更重要的是，它可以安装在系统的扩展内存，把 CCDOS 字库装载到虚拟盘，解决因系统基本内存（DOS 系统最大基本内存为 640K）的限制，而不能同时运行 CCDOS 和 DBASE III plus、FOXBASE 等大型内驻软件的问题（除非在无盘工作站增加汉卡）。

然而，我们在为用户安装 Novell 网过程中发现，在无盘工作站上安装虚拟盘，网络不予承认，一旦工作站远程启动完毕，虚拟盘就消失。为此，我们认真地分析了 Novell 网操作系统及其界面外壳程序，发现问题的症结是，Novell 网的工作站驱动程序通过检查 BIOS 参数区硬件设备配置字的最低位（该位表示系统内是否装有软盘驱动器），确定网络默认驱动器的盘号（A-Z）。如果最低位为 1，则系统内配置了软盘驱动器，网络把 F 指定为其默认驱动器（对 DOS3.X 而言），在此之前的字母都留给本地使用，如果本地盘号超过 F，网络将从本地最后一个盘号后分配网络逻辑驱动器；如果最低位为 0，则网络默认驱动器指定为 A，并认为无本地盘存在，因此，无盘工作站上安装的虚拟盘自动失效。

为了解决这个问题，我们修改了工作站驱动程序 IPX.COM。下面是我们在 NetWare SFT2.15 上修改并实现的全过程。其它 Novell 操作系统可按此方法作相似修改。

①将一张系统盘插入有盘工作站的 A 驱动器中，拷贝

工作站驱动程序 IPX.COM、NET?.COM、VDISK.SYS（或 RAMDRIVE.SYS），并在该盘上建立 CONFIG.SYS，其中包括语句：

DEVICE=VDISK.SYS（或 RAMDRIVE.SYS）360  
512/E

其中参数的设置可参看 DOS 资料。

②对 IPX.COM 进行修改：

```
A>C: DEBUG IPX.COM
-a 100
XXXX: 0100 JMP 280 (跳转到空白区)
XXXX: 0103
-a 280
XXXX: 0280 PUSH ES
MOV AX, 0
MOVES, AX (建立辅助段)
ES:
MOV AL, [410] (取系统配置低字节)
OR AL, 01 (修改)
ES:
MOV [410], AL (存回)
POP ES
JMP 622E (IPX.COM 原第一语句)
```

-W

-q

③从有盘工作站进网，确认上面的盘片在 A 软驱中后，执行 DOSGEN 以生成远端启动用的 NET\$OS.SYS；

Y>MAPF:=SYS:SYSTEM

Y>MAPG:=SYS:LOGIN

Y>G:

G>F: DOSGEN

生成好 NET\$OS.SYS 后，打开任何无盘工作站可自动安装一个 360K 的虚拟盘。

RND(X) 的 X 可取负数的范围，这在程序 1 中已验证过了，是  $1 \sim 1.7 \times 10^{38}$ ，密钥 M2 的种数与密码位数有关，因为 ASCII 字符的代码值为 0~127，最大三位数为 999，所以 M2 的值最大为 2~872，这是三位密码的第二密钥范围，四位密码为 2~9872，五位以上依次在最高位加 9。由此可知，组合密钥的种数极多，以三位码为例，可达  $872 \times 1.7 \times 10^{38} = 1.4824 \times 10^{41}$  种，这是一个令人难以置信的天文数字。对于破译者来说，即使知道这种编密方法，动用每秒运算 1 亿次的巨型机，以每秒钟运行一遍程序计（实际上是办不到的），要逐一试完所有密钥共需要  $4.8 \times 10^{26}$  年。更何况密码位数还可增加，以及对同一段文字还可以用不同密钥反复加密，这样，密码被破译的概率实际

上完全等于零。读者可以试一试，只要输入密钥稍有差错甚至输入的密码某一数字有一位不对，便可导致 270 语句中 CHR\$ 函数自变量范围超值而中断运行，或者打印出一些乱七八糟的字符出来（汉字或英文单词必须每个 ASCII 代码毫无差错才能译出），为此，程序中设置了自动纠错语句来处理上述情况。

从以上分析可知，本文介绍的方法已有相当的实际应用价值，由此也可见电子计算机高强本领之一斑！

（本文程序在 APPLE II 及 CEC-I 上通过，由于不受汉字系统类别等方面的限制，可方便地移植于其他机型）。

# MS-DOS 彻底剖析 (九)

## IBMBIO 总体结构

郭嵩山 陈 政

【编者按】本刊自八九年第三期开始，以连续 8 期的篇幅，刊载了中山大学计算机软件研究所郭嵩山老师撰写的 MS-DOS COMMAND 模块彻底剖析，深受读者欢迎。不少读者来信要求继续刊载 DOS 其他模块剖析的文章。为满足读者的需要，本刊将从本期开始，在软件纵横栏目上，继续以连载的方式，刊载郭老师撰写的 MS-DOS IBMBIO 模块的彻底剖析。

**提要：**本系列文章从操作系统结构的角度，对 MS-DOS 的 IBMBIO 模块进行深入的研究和剖析。研究和剖析工作是在兼顾到 MS-DOS 多个版本特点的基础上，以目前流行的 3.3 版本为例来进行的。其研究方法是采用由笔者所倡导的以模块主程序为主线、以数据结构为中心的系统软件分析的方法，期望能让读者更快地理解。

本系列文章是在作者全面深入地剖析 MS-DOS 多个版本的基础上撰写的，文章里提供的不少资料是目前一般文献及技术资料中少有介绍，期望对那些希望深入了解 MS-DOS 的磁盘 BIOS，并试图在此基础上进行系统一级的研究、开发和改造的读者有所帮助。

为适应不同对象读者的需要和考虑到杂志的特点，本系列将以一组相对独立而整体上又有关联的文章形式推出。

众所周知，IBMBIO.COM 模块是组成 MS-DOS 操作系统的三大模块之一，它作为层次结构的次底层，供 DOS 内核调用，以驱动固化在 ROM-BIOS 中的中断例程，控制硬件工作。IBMBIO 模块以磁盘隐文件 IBMBIO.COM 形式存在盘中，它的内部又由标准设备驱动程序，初始化程序 (SYSINIT) I 和初始化程序 II 所组成，而初始化程序 I 又由其主体和重装入程序两个模块所组成，整个 IBMBIO 的模块结构以及与 MS-DOS 其他模块的关系如图 1 所示。

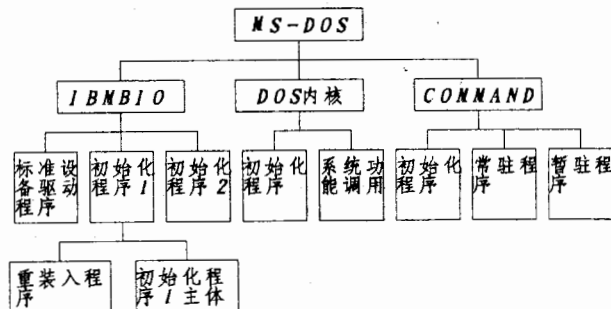


图 1 MS-DOS 的模块结构

### 一、MS-DOS 的引导过程

MS-DOS 从磁盘引导入内存，大体上经历了四个阶

段。在这四个阶段中，将由 MS-DOS 不同的模块起主导作用。下面，将结合 MS-DOS 各模块在初始化过程中在内存位置的变化来阐述初始化过程中各个阶段的特点。

#### 1. 装入 IBMBIO.COM 阶段

这个阶段的任务主要由 MS-DOS 引导记录来完成，其实现过程如下：

1) 系统加电后，由硬件设置，控制自动指向内存 ROM-BIOS 的 FFFF:0000H，进行硬设备的自测，执行固化在 ROM-BIOS 中的系统自举中断 (INT 19H) 处理。

2) 将位于 0 面 0 道 1 扇区的引导记录读到 0000:7C00H 开始的内存，并将控制转移到引导记录。

3) 引导记录首先检查盘目录中是否有 IBMBIO.COM 和 IBMDOS.COM 这两个隐含文件。若无，提示用户换上系统盘并重新引导，若正确，把 IBMBIO.COM 文件读入内存 0070:0000H 开始的内存区，然后将控制转移到 IBMBIO 模块。

在这阶段中各模块在内存变化如图 2 所示。

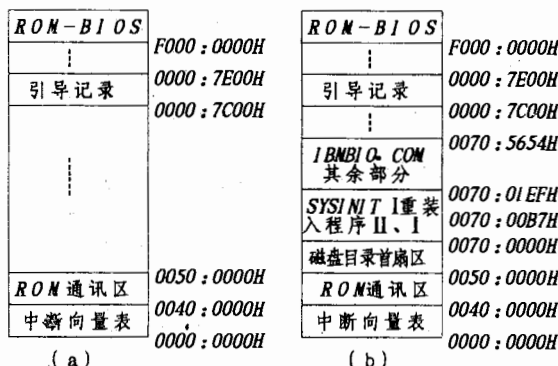


图 2 装入 IBMBIO.COM 阶段内存的变化

其中：a)、装入引导记录后的内存

b)、装入 IBMBIO.COM 后的内存

#### 2. 装入 IBMDOS.COM 阶段

这个阶段的任务由 IBMBIO 初始化程序 (SYSINIT I) 来完成，其实现过程如下：

1) 首先执行重装入程序 (位于 IBMBIO.COM 文件偏移 0000~01EEH 处), 建立为重装入 IBMBIO 而设置的 I/O 参数表, 并将自身 (偏移 00B7~1EEH) 移至 0000: 7D00H 处, 将 FAT (文件分配表) 读入到内存 0000: 8000H, 计算 IBMBIO.COM 文件除去重装入程序后的长度, 然后根据 FAT 表的内容, 将 IBMBIO.COM (除重装入程序) 读入到内存 0070: 0000H。之所以需要重新装入 IBMBIO 的原因是引导记录读入 IBMBIO.COM 文件是按照文件在磁盘上是连续分布的假设而进行的。这样就有可能因假设不正确而导致错误的读入。所以, 必须在读入 SYSINIT I 后运行重装入程序, 建立 FAT 表在内存的映象, 然后根据 FAT 表把 IBMBIO.COM 正确装入。

重装入程序执行后各模块在内存的位置如图 3 所示。

ROM-BIOS	F000:0000H
:	
磁盘 FAT 表	0000:8000H
:	
重装入程序 (II)	0000:7E38H
:	
IBMBIO.COM 其余部分	0000:7D00H
磁盘目录首扇区	0070:0000H
ROM 通讯区	0050:0000H
中断向量表	0040:0000H
	0000:0000H

图 3 重装入程序运行后的内存

ROM-BIOS	F000:0000H
:	
磁盘 FAT	DDDD:0000H
:	
IBMDOS.COM	5B9:75CFH
:	
	5B9:0000H
	3B9:1FD4H
SYSINIT II	3B9:7F1H
中断服务程序	03B9:0000H
SYSINIT I	0070:348FH
:	
根据配置取舍后保留数据及过程	0070:29FAH
标准设备驱动程序	0070:1B58H
磁盘目录首扇区	0070:0000H
ROM 通讯区	0050:0000H
中断向量表	0040:0000H
	0000:0000H

图 4 装入 IBMDOS.COM 阶段完成后的内存

注: PPPP=RAM 最高可用地址-每一个 FAT 所占字节

2) 完成各物理端口 (通讯口、打印机口及时钟) 初始化并根据设备物理配置的情况, 修改并重置各种参数, 为各个块设备生成各自的块设备控制块 (BDCB-Block Device Control Block), 修改相应的中断例行程序入口及标准设备驱动程序命令代码的入口。

3) 定位 IBMBIO 的标准设备驱动程序, 并根据硬件

配置对 IBMBIO 设备驱动程序部分进行取舍, 为其进入工作状态作准备。

4) 将 IBMDOS.COM 文件读入到 IBMBIO.COM 之后, 即从 5B9: 0000H 开始的内存区。

5) 控制转移到 IBMDOS.COM

这个阶段任务完成后, 其各模块在内存的位置如图 4 所示。

3、IBMDOS 初始化及内核定位阶段

这个阶段的任务由 IBMDOS 的初始化程序来完成。

1) 将 IBMBIO 初始化程序 II (SYSINIT II) 移至内存高端, 以腾出空间来定位 DOS 内核。

2) 为磁盘文件管理建立一系列参数表和控制块。

3) 将 DOS 内核定位于紧靠内存 IBMBIO 常驻部分之后, 即从 3B9: 0000H 开始的内存。

4) 控制转移到 IBMBIO 的 SYSINIT II。

这个阶段任务完成后, 各模块在内存位置如图 5 所示。

:	
SYSINIT II	CS:0000H
:	
DOS 内核	
取舍后保留的数据和过程	70:yyyyH
标准设备驱动程序	70:1B58H
	70:0000H

图 5 IBMDOS 初始化及内核定位后的内存

其中: yyyy 值根据取舍后情况而定。

4、装入 COMMAND.COM 阶段

ROM	
COMMAND 常驻程序	
用户区	
COMMAND 常驻程序	
可安装的设备驱动程序	
堆栈运行环境	
磁盘路径表 (DPAT)	
BUFFER 运行环境	
FCBS 运行环境	
FILES 运行环境	
新增的磁盘参数表 (DPT)	
新增的设备驱动程序链	
DOS 内核	0070:yyyy
标准设备驱动程序 (含取舍后保留内容)	0070:0000H
DOS 通讯区	0050:0000H
ROM 通讯区	0040:0000H
中断向量表	0000:0000H

图 6 MS-DOS 引导成功后的内存

本阶段前半部分工作由 IBMBIO 的 SYSINIT II 来完成, 后半部分由 COMMAND.COM 初始化程序来完成。



1) 为 SYSINIT II 建立 DOS 运行环境, 也即建立程序段前缀 PSP、内存管理块 (MCB) 以及构造磁盘路径表 (DPAT), 并修改 INT 2FH (多路中断) 和 INT 24H 中断向量, 便于多路中断及严重出错中断返回。(以上配置仅为 SYSINIT II 而建立的, 以后还会为用户程序而重新建立新的配置)。

2) 调用 DOS 内核功能, 运行系统盘上的 CONFIG.SYS 文件, 创建操作系统运行环境, 其中包括建立新增加的磁盘参数表、FILES 运行环境、FCBS 运行环境、BUFFER 运行环境、磁盘路径表 (DPAT) 和堆栈运行环境等。

3) 加载可安装的设备驱动程序, 修改设备链, 使之与标准设备驱动程序链连接起来。

4) 调用 DOS 的 EXEC 功能, 将命令解释程序 (shell) 加载到内存 (缺省的 shell 为 COMMAND.COM 文件)。

5) 完成 COMMAND 初始化工作, 控制转移到 COMMAND.COM 暂驻程序, 进入命令状态。至此, 引导 MS-DOS 成功, 其内存分配如图 6 所示。

## 二、IBMBIO 各子模块功能概述

如前所述, 在 IBMBIO 初始化的各个阶段, 根据系统的物理配置及初始化过程的逐步完成, 将逐步释放 IBMBIO 的 SYSINIT 程序所占有的内存空间, 也就是说, 在初始化过程中, IBMBIO 的各个子模块在内存中的位置是有所变化的, 为便于读者学习、研究和改造, 表 1 列出了 IBMBIO 的各个子模块的地址范围、功能以及它们在初始化前后在内存位置的变化。

在表 1 中, IBMBIO.COM 文件上的偏移地址值实际上是初始化刚开始时装入到 0070: 0000 开始的内存值, 但它在初始化后是完全改变的, 读者是无法直接从 0070: 0000H 开始的内存中读到。但这个内容可通过将 IBMBIO.COM 文件读入内存用户区, 然后用 DEBUG 反汇编读出。不同的是, 读入内存用户区各模块的偏移地址比表 1 中 IBMBIO.COM 文件上的偏移值 (也即初始化时刚装入到 0070: 0000H 的地址值) 增加 100H, 这 100H 个 Byte 是为程序段前缀 PSP 所设置的。

注: 本系列文章是中山大学自然科学基金资助的有关操作系统研究的科研课题成果之一。陈学军、朱国庆同志参加了本项目的研究。

表 1 IBMBIO 各子模块地址、功能和初始化前后内存的变化

IBMBIO.COM 文件上偏移	工作时的地址	初始化结束后的地址	功能及说明
0000~1EFH	0000: 7D00~7EEH	被冲掉	重装入程序, 执行时先把自身搬至 0000: 7D00~7EEH, 然后读入 FAT, 并按 FAT 将 IBMBIO.COM 文件从偏移 1EFH 开始的内容重新读入内存 0070: 0000H 处
1EF~7CAH	0070: 0000~5DBH	同左	标准设备驱动程序的数据区
7CB~7D5H	0070: 5DC~5E6H	同左	标准设备决策例行程序
7D6~88AH	0070: 5E7~69BH	同左	标准设备中断例行主程序
88B~89AH	0070: 69C~6ABH	同左	INT 29H 服务程序
89B~93EH	0070: 6AC~74FH	同左	标准控制台设备驱动程序
93F~945H	0070: 750~756H	同左	INT 1BH 服务程序
946~9D4H	0070: 757~7E5H	同左	标准辅助设备驱动程序
9D5~A9DH	0070: 7E6~8AEH	同左	标准列表设备驱动程序
A9E~BB2H	0070: 8AF~9C3H	同左	标准时钟设备驱动程序及其数据区
BB3~1152H	0070: 9C4~F63H	同左	块型设备驱动程序 (一) (01H、02H、04H、08H、09H、0FH 子命令处理及一些子过程)
1153~13B8H	0070: F64~11C9H	同左	INT 13H 服务程序扩充部分 (一) 及其数据区
13B9~14CBH	0070: 11CA~12DCH	同左	块型设备数据区 (一)

续上表

14CC~19FAH	0070: 12DD~180BH	同左	块型设备驱动程序 (二) (13H、17H、18H 子命令处理及一些子过程)
19FB~1A31H	0070: 180C~1842H	同左	INT 2FH 服务程序 (一)
1A32~1A53H	0070: 1843~1864H	同左	块型设备驱动程序 (三) (一些子过程)
1A54~1A92H	0070: 1865~18A3H	同左	INT 19H 服务程序所用数据区
1A93H~1ACEH	0070: 18A4~18DFH	同左	块型设备数据区 (二) (BPB 指针组)
1ACF~1C09H	0070: 18EOH~1A1AH	同左	修改中断向量
1C0A~1C15H	0070: 1A1B~1A26H	同左	块型设备驱动程序 (四) (00H 子命令处理)
1C16~1C40H	0070: 1A27~1A51H	同左	INT 2FH 服务程序 (二)
1C41~1CE1H	0070: 1A52~1AF2H	同左	块型设备驱动程序 (五) (一些子过程)
1CE2~1D47H	0070: 1AF3~1B58H	同左	块型设备驱动程序数据区 (三)
1D48~1D99H	0070: 1B59~1BAAH	同左	为硬盘预留的第一个块设备控制块
1D9A~1DEBH	0070: 1BAB~1BFCH	同左	为硬盘预留的第二个块设备控制块
1DEC~1F03H	0070: 1BFD~1D14H	同左	块型设备驱动程序 (六) (0DH、0EH 及支持高容量软盘驱动器操作等一些子过程)
1F04~1F45H	0070: 1D15~1D56H	同左	INT 13H 服务程序扩充部分 (二) 及其数据区
1F46~209EH	0070: 1D57~1EAFH	同左	块型设备驱动程序 (七) (支持高容量软盘驱动器操作等一些子过程)
209F~27E5H	0070: 1EB0~25F6H	同左	为硬盘分区预留的 23 个块设备控制块
27E6~292EH	0070: 25F7~273FH	同左	INT 13H 服务程序扩充部分 (三)
292F~2A10H	0070: 2740~2821H	同左	时钟处理过程 (一) (支持高级时钟功能)
2A11~2A31H	0070: 2822~2842H	同左	INT 6CH 服务程序
2A32~2BE8H	0070: 2843~29F9H	同左	时钟处理过程 (二) (支持高级时钟功能)
2BE9~2CA6H	0070: 29FA~2AB7H	被冲掉	初始化 SYSINIT I 之数据区
2CA7~367EH	0070: 2AB8~348FH	被冲掉	初始化程序 SYSINIT I
367F~3E6FH	3B9: 0000~7F0H	被冲掉	中断服务程序, 用于在形成堆栈运行环境时, 修改各中断操作
3E70~3F72H	3B9: 7F1~8F3H	被冲掉	初始化 SYSINIT II 数据区 (一)
3F73~5287H	3B9: 8F4~1C08H	被冲掉	初始化程序 SYSINIT II
5288~529AH	3B9: 1C09~1C1BH	被冲掉	初始化时的 INT 24H 服务程序
529B~5753H	3B9: 1C1C~1FD4	被冲掉	初始化时 SYSINIT II 数据区 (二)

# 在 PC DOS 中同时打开二十个以上文件

中山大学计算机及应用八九级 林粤舟

在 PC DOS 的各种说明书和参考书中,都说明一次打开的句柄不可超过二十个,除去 DOS 占用五个作为标准输入出,标准错误设备,标准打印设备和辅助设备外,用户可打开的文件数最多为十五个。但在其系统配置文件 (config.sys) 中,又有一条命令 FILES=XXX,说明同时可打开的文件数。笔者取 XXX 为 30, DOS 没指出错误,但就是不能同时打开超过 15 个文件。用一些内存显示程序显示出 DOS 占用内存增大了,这说明 DOS 为这 30 个句柄开辟了缓冲区,但为什么就不能使用呢?

原来,在程序段前缀 (PSP) 偏移量为 18h 的地方,有一块长度为 20 的区域(如附一所示)存放着 DOS 的文件表(这一部分未写入文档,一般书籍介绍 PSP 时均写这一部分为 DOS 保留)每一个打开的句柄要在这占一个位置(一个字节)。由于这一部分仅有 20 个字节, PSP 其他部分又存放着其他内容,无法扩展,以至于虽然 DOS 在内存中为每个打开文件开辟了缓冲区,但由于 PSP 的限制,无法使用。

```

psp: 24— 1
psp: 25— 1
psp: 26— 1
psp: 27— 0
psp: 28— 2
psp: 29— 3
psp: 30— 4
psp: 31— 5
psp: 32—255
psp: 33—255
psp: 34—255
psp: 35—255
psp: 36—255
psp: 37—255
psp: 38—255
psp: 39—255
psp: 40—255
psp: 41—255
psp: 42—255
psp: 43—255
psp: 44— 80
size = 20
address = 32027: 24
pspseg = 32027

```

## 附一

为了突破 DOS 的限制,先要分析一下文件表的结构。DOS 对每一个文件缓冲区给予一个编号,称为 DOS 内码,第一个文件缓冲区给予 03,第二个文件缓冲区给予 04,以此类推,内码 00 已分配给 AUX, 01 为 CON, 02 代表 PRN。这个 DOS 内码对用户是透明的,文件表存放

的就是 DOS 内码,(若内码为 FFh,表示未使用),而文件句柄的内容是相对于文件表头部的偏移量。每当 DOS 打开一个新文件,它就在文件表中搜索第一个 FFh,若找到(比如在 (PSP+18h) + Ah 处),则再找第一个未使用的文件缓冲区,若找到,则把这个缓冲区的编号(DOS 内码)填入文件表中,然后把 Ah 作为文件句柄传给用户;若两者之一有一个未找到,则返回错误信息指出打开文件太多。示范程序 1 就是显示在打开两个文件后文件表的内容。从打印结果可以看到,前五个字节(即前五个句柄)已被赋予了 00~02 的值,这与 DOS 提供的五个标准句柄相对应。第六个字节的 03 对应于打开的设备 lst,第七个字节为 04,对应于打开的第一个文件 f1, 05 对应于打开的文件 f2,其余字节为 FFh,表示未使用,若这时关闭 f1,将会看到第七个字节变为 FFh。

```

program 示范程序 1;
uses dos,printer;
var i,j:word;
begin
for i:=24 to 44 do
begin
j:=mem[prefixseg:i];
writeln(lst,'psp:',i:4,'—',j:4);
end;
assign(f1,file1):rewrite(f1);
assign(f2,file2):rewrite(f2);
for i:=24 to 44 do
begin
j:=mem[prefixseg:i];
writeln(lst,'psp:',i:4,'—',j:4);
end;
writeln(lst,'size =',memw[prefixseg:50]);
write(lst,'address =',memw[prefixseg:54]);
writeln(' ','memw[prefixseg:52]);
close(f1);close(f2);
writeln(lst,'pspseg-',prefixseg);
end.

```

明白了文件表的结构,就可以想办法使用超过 15 个的文件。在这里我采用的是类似覆盖的技术,虽然每个打开的文件对应于一个 DOS 内码(这是 DOS 规定,无法改变),但可用程序使多个 DOS 内码对应于同一个文件句柄,比如文件 f1, f2, f3 对应 DOS 内码 a1, a2, a3, a1, a2, a3 同时使用句柄 b,每当使用 f1 时,把 a1 填入 (psp+18h) + b 处,然后引用句柄 b 即可对文件 f1 进行操作,而把 a2 填入句柄 b 中,然后引用句柄 b 即可改对 f2 进行操作,一旦 a3 填入 (psp+18h) + b 中,以后一切引用句柄 b 的操作均作用在文件 f3 上。用此技术其实只适于使用了一个句柄就可以实现对所有文件设备的访问,为了

提高效率,可以采用部分覆盖技术,一些常用的文件单独占用一个句柄,而不常用文件尽可能精心组合,共用一个句柄,使句柄更换尽可能少,以提高效率。

示范程序 2 示范如何用以上技术一次打开 40 个文件,这个程序用 TURBO PASCAL 5.0 编译成功,用 4.0 和 5.5 也同样可以。程序中使用了一个标准常量 prefixseg 给出 psp 的段址,这是 TP5.0 提供的(TP5.0 为 TURBO PASCAL 简称),在其他语言中可能没有,可以通过系统功能调用号 62h (在 DOS3.0 以上)或 51h (在 DOS2.0 以上)来获得。在使用这个程序前,应使配置文件 (config.sys) 中有 FILES=40,以保证 DOS 开辟了足够的缓冲区。这个程序打开的 40 个文件共用 \$5 这个文件句柄 (由常量 handle 说明)。过程 newfile open 用于打开一个新文件后保存此文件的 DOS 内码,这过程在 open 之后被调用。过程 closefile 用于取消一个 DOS 内码,将空间让出给其他程序使用,此过程在每个 close 之后调用。过程 usefile 将相应 DOS 内码填入句柄 handle 中,在每次引用文件前调用一次,确保以后对句柄 handle 操作作用于正确文件。这三个过程完全结构化,修改这三个过程就可以实现前面所讲的部分覆盖技术。主程序除了在适当地方添加对三个过程 (newfile open, use file, closefile) 外,其余部分与使用小于 15 个文件,不采用此技术的程序一样。以前的程序只需作些小修改即可同时打开 20 个文件以上。这个程序虽使用 TP5.0 编写,但其原理很容易在其他语言中实现。

以上所讲的采用覆盖技术来实现的只是一种方法,另一种方法可能更简单一些。在 psp 中另有三个有用的区域。一个为 psp+50,在此存放着文件表的大小;另一个为 psp+52,存放着文件表地址的偏移量 (offset);还有一个为 psp+54,存放着文件表地址的段址。由示范程序 1 打出的结果可以看到这三个地址存放的值。由于 psp 大小的限制,文件表在 psp 中无法扩展,但我们可以从内存中开辟一块足够大的地方,将这三个地址都改成合适之值,这样就可以实现同时打开不超过 255 的文件。

注 1: 在本文中出现的诸如 psp+54 中 psp 代表程序段前缀所在地址。

注 2: psp 的这 20 个字节对 DOS 是保留的,未写入文档,不保证在任何操作系统下可行,笔者在 DOS2.00, DOS2.10, DOS3.00, DOS3.10, DOS3.20, DOS3.30 下均可行。

```
program 示范程序 2;
uses dos;
const
```

```
maxfilenumber = 40;
handle =
5;
```

```
type
```

```
dosnumrecord = record
dosstring :string;
doshandle :word;
dosnumber :byte
end;
var
dosnum:array [6..maxfilenumber] of dosnumrecord;
type
psptype = record
reserved1:array [1..22] of byte;
parentpspseg:word;
filetable:array [1..20] of byte;
reserved2:array [44..49] of byte;
fhtablesize:word;
fhtableaddress:pointer;

end;
var
psp: ^psptype;
result:byte;
i:integer;
f:array [6..maxfilenumber] of text;
st:string;
procedure writel;
begin
writeln;
writeln('Press Return to continue..');
readln;
end;
procedure newfileopen(filename:string;var result:byte);
var
i,j:integer;
begin
j:=0;
for i:=maxfilenumber downto 6 do
if dosnum[i].dosnumber=255 then j:=i;
if j=0 then result:=255 else result:=0;
if j<>0 then begin
with dosnum[j] do
begin
dosnumber:=psp^.filetable[handle+1];
psp^.filetable[handle+1]:=255;
dosstring:=filename;
doshandle:=handle;
end;end;
end;
procedure closefile(filename:string;var result:byte);
var i:integer;findbol:boolean;
begin
for i:=6 to maxfilenumber do
```



```

if dosnum[i].dosstring = filename then
begin
dosnum[i].dosnumber := 255;
findbol := true;
end;
if findbol then result := 0 else result := 255
end;
procedure usefile(filename:string;var resule:byte);
var i:integer;findbol:boolean;
begin
for i:= 6 to maxfilenumber do
if dosnum[i].dosstring = filename then
begin
psp^.filetable[dosnum[i].doshandle+1] := dosnum[i].dosnumber;
findbol := true;
end;
if findbol then result := 0 else result := 255
end;
procedure closehandle(filename:string);
var i:integer;
begin
for i:= 6 to maxfilenumber do
if dosnum[i].dosstring = filename then psp^.
filetable[dosnum[i].doshandle+1] := 255;
end;
begin
for i:= 6 to maxfilenumber do dosnum[i].dosnumber := 255;
psp := ptr(prefixseg,0);
writeln('Ready to Open 6..',maxfilenumber,' Files. ');
writeln;
for i:= 6 to maxfilenumber do
begin
str(i,st);
st := 'file'+st;

```

```

assign(f[i],st);
rewrite(f[i]);
newfileopen(st,result);
if result = 255 then begin writeln('No Such File');halt(1) end;
writeln('File',i,' Created. ');
end;
writeln;
writeln('Ready to Write to 6..',maxfilenumber,' Opened Files. ');
writeln;
for i:= 6 to maxfilenumber do
begin
str(i,st);
st := 'file'+st;
usefile(st,result);
if result = 255 then begin writeln('No Such File');halt(1) end;
writeln(f[i],st);writeln(f[i],st);writeln(f[i],st);closehandle(st);
writeln('File',i,' Written. ');
end;
writeln;
writeln('Ready to Close 6..',maxfilenumber,' Opened Files. ');
writeln;
for i:= 6 to maxfilenumber do
begin
writeln('File',i,' Closed. ');
str(i,st);
st := 'file'+st;
usefile(st,result);
if result = 255 then begin writeln('No Such File');halt(1) end;
close(f[i]);
closefile(st,result);
if result = 255 then begin writeln('No Such File');halt(1) end
end;
writeln;
writeln('Done');
end. ■

```

## DEBUG 可恢复备份文件

柳州铁路材料总厂 单昶贤

使用 DOS 的 BACKUP 命令产生的软盘或硬盘的备份是不能直接使用的, 因此用 RESTORE 命令重存如果出错而终止, 系统就不能使用, 一次我厂做物资统计报表时, 用 RETSORE 命令把多个备份文件从软盘重新存储到硬盘, 不幸 01 号盘的某个文件意外出错, 使得重存无法完成, 造成不能工作。报表文件有 30 个, 若重建, 则工作量相当庞大。笔者用调试 (DEBUG) 程序对备份文件进行分析后发现: 备份文件实质上是在原文件之前增添了 128

字节的控制参数, 从第 129 字节起即为原来文件内容, 只要稍加修改, 即可恢复其本来面目。

修改过程如下:

```

A> DEBUG B: 201.POL ;调试备份盘上的报表文件
-R CX ;修改文件长度
CX: CC80
: CC00 ;文件长度减去 128 字节
-N B: 201.POL ;重新命名
-W CS: 180 ;从 CS: 180 处开始写盘
-Q ;退出 DEBUG 状态

```

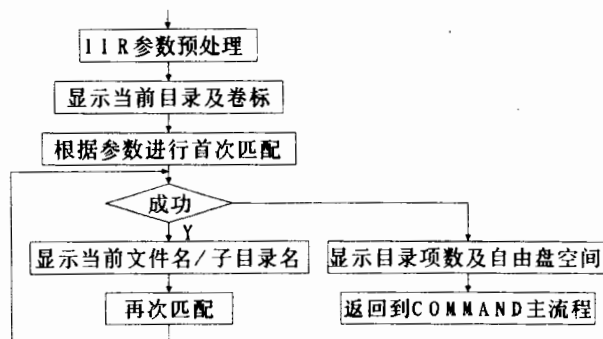
完成上述步骤, 文件 201.POL 即恢复为原来的模样, 可以正常使用了。重复以上步骤即可恢复所有报表文件。避免了重新编制、输入、调试之苦。这种方法不仅适合恢复一般性文件的备份, 还可以恢复文件名为.COM 和.EXE 的备份文件 (注意.EXE 文件应先改为非.EXE 文件)。■

# 隐含文件列目录的最简方法

武汉大学计算机科学系研究生 90 级 万玉丹

大家知道, 在使用 DOS 的 DIR 内部命令时, 隐含文件 (及系统, 只读文件) 都是看不到的。一个典型的情况就是 DOS 的 IBMBIO.COM 及 IBMDOS.COM 这两个文件。那么, 怎样使这类文件象普通文件一样列目录呢? 方法当然很多, 本文介绍的方法应该是最简单的, 因为它只需要修改 COMMAND.COM 的一个字节就够了。

DIR 的服务程序在 COMMAND.COM 中; 以 PC-DOS V3.30 为例: 对 640KB 内存, 启动成功后, 暂驻部份的程序区位于 9B2A: 100—3728, DIR 的处理子程序位于 9B2A: 0ECB—116F。通过分析, 得出其大体流程如下:



注: 所谓匹配, 就是:

1. DIR 无参数或参数为 \*.\* 时, 一般文件都可与之匹配。

2. 参数有字符和通配符时, 只有部份文件可与之匹配。比如 F\*.\*C\*, 就只有 FORMAT.COM 等文件才能匹配。

问题的关键在于如何匹配, 下面是这部分的片断:

```

9B2A:103F C6065500FF MOV BYTE PTR[0055],FF
;将 psp+55 以下作为扩展 FCB

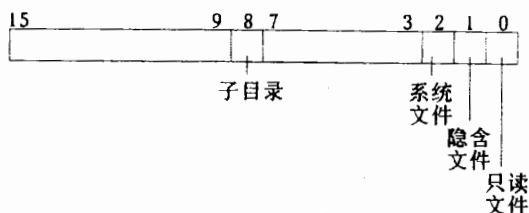
9B2A:1044 C6065B0010 MOV BYTE PTR[005B],10
;属性为子目录或普通文件均可匹配

9B2A:1049 B411 MOV AH,11

9B2A:104B BA5500 MOV DX,0055

9B2A:104E CD21 INT 21
;进行匹配
  
```

其中 9B2A: 1048 字节规定了可参加匹配的文件类型, 结构如下:



注: 各位都是: 1——允许该文件类型参加匹配

0——不允许

可以看到, 由于上面没有让隐含 (系统, 只读) 文件也参加匹配, 故而这类文件无法列目录。

弄清了问题的症结所在, 解决这个问题很简单。只要能使文件属性字节 9B2A: 1048 由 10H 变为 17H 就行了。

不过, 在内存中直接修改是无效的, 需要修改磁盘上的 COMMAND.COM 文件才行。方法如下:

A>debug command.com

—s 0 m m c6 06 5b 00 10

1094:26A4(即寻找相当于 9B2A: 1044 处的那条指令)

—

—e 26a8

1094:26A8 10.17

—

—W

Writing 62DB bytes

—q

重新启动, 再执行 DIR 时, 就可对隐含文件等也列目录了。比如:

A>dir

Volume in drive A has no label

Directory of A:\

IBMBIO	COM	22100	1-01-80	12:02a
IBMDOS	COM	30159	1-01-80	12:06a
COMMAND	COM	25307	1-01-80	12:03a
AUTOEXEC	BAT	5	1-01-80	12:00a
FORMAT	COM	11671	2-02-88	12:00a
DEBUG	COM	15866	9-02-88	12:00a

6 File(s) 243712 bytes free

对于 PC-DOS 的不同版本, 乃至与 PC-DOS 平行的 MS-DOS, 修改的方法是一样的, 因为这些 DOS 版本也是这样匹配的 (指令完全相同!). 读者可按此修改一下 PC-DOS V2.10, 会发现除了指令地址不同外, 效果基本上一样。■

# 用 DEBUG 恢复内存文件

广西师范大学物理系 87-1 班

陈奈菲

在使用 WordStar 等编辑软件编辑文件时,有时会因操作失误退出编辑状态,致使一份刚编辑的文件未能存盘而丢失。若文件较长,要重新建立并非易事,为此,本文介绍一种用 DEBUG 程序重新恢复文件的方法。

一般说来,所编辑的文件内容通常是被放在编辑软件之后的,而多数的编辑软件要比 DEBUG 程序长,若将 DEBUG 程序调入内存取代编辑软件时,并不会破坏原来编辑的文件内容,因此给我们恢复整个文件提供了可能,即使所使用的编辑软件比 DEBUG 程序短,文件的部分内容被覆盖,但仍可部分地恢复文件,使损失减小到最低程度。

恢复文件的基本思想是利用 DEBUG 程序的 D (Display) 命令连续地搜索,寻找所编辑的文件内容,并记下该文件的存放起始地址和结束地址,然后算出文件长度,再用 W (Write) 命令将该文件内容写入磁盘,这样便可恢复丢失的文件。

下面举一例子说明。

调入 WordStar 并建立如下 TEST.TXT 文件:

```

a:aaaaaaaa
bbbbbbbbbb
ccccccccc
dddddcccc
cccccccc

```

随后退出 WordStar,这时我们可用下面的操作恢复上述文件内容。

(1) 在退出 WordStar 后立即运行 DEBUG 程序,千万不可执行其它命令,否则有可能破坏原文件的内容,使文件难以恢复。

(2) 用 D 命令连续显示内存的数据,直到在屏幕右边的显示信息中出现原文件内容为止。本例结果如下:

```

-d 5490:660 6ef
5490:0660 00 00 00 00 00 00 00 .....
5490:0668 61 61 61 61 61 61 61 aaaaaaaa
5490:0670 61 61 0D 0A 62 62 62 62 aa.bbbb
5490:0678 62 62 62 62 62 62 0D 0A bbbbbb..
5490:0680 63 63 63 63 63 63 63 63 cccccccc
5490:0688 63 63 0D 0A 64 64 64 64 cc.dddd
5490:0690 64 64 64 64 64 64 0D 0A dddddd..
5490:0698 65 65 65 65 65 65 65 65 eeeeeeee
5490:06A0 65 65 0D 0A 1A 1A 1A 1A ee.....
5490:06A8 1A 1A 1A 1A 1A 1A 1A .....
5490:06B0 1A 1A 1A 1A 1A 1A 1A .....
5490:06B8 1A 1A 1A 1A 1A 1A 1A .....
5490:06C0 1A 1A 1A 1A 1A 1A 1A .....
5490:06C8 1A 1A 1A 1A 1A 1A 1A .....
5490:06D0 1A 1A 1A 1A 1A 1A 1A .....
5490:06D8 1A 1A 1A 1A 1A 1A 1A .....
5490:06E0 1A 1A 1A 1A 1A 1A 1A .....
5490:06E8 00 00 00 00 00 00 00 .....

```

(3) 记下该文件的起始地址 5490: 0668, 再寻找结束标志, 记下结束地址 5490: 06E8 (用 WordStar 建立的文件是以 Control-Z 即 ASCII 码的 1AH 为结束标志)。

(4) 再键入如下命令:

```

-n test.sav ; 给恢复的内容指定一个文件名 (test.sav)
-h 6e8 668 ; 计算恢复的文件长度 (80H)
0D50 0080
-r bx
BX 0000
; ; 将文件长度的高位写入 BX 寄存器 (00H)
-r cx
CX 0000
:80 ; 将文件长度的低位写入 CX 寄存器 (80H)
-w 5490:668 ; 将指定的内容写入磁盘 (起始地址为 5490:

```

668)

Writing 0080 bytes

-q ; 返回 DOS 状态。

A>

这样, 丢失的文件即可恢复。

上述方法原则上对于丢失的文件内容未被完全破坏的情况都适用。虽然本文是以 WordStar 作示例, 但该方法对于其它的编辑软件同样适用, 有兴趣的读者不妨一试。

(接 46 页)

(三) DBASE Ⅲ 的使用与操作 (10%) 或 (50%)

- 1、基本的数据库操作
- 2、数据库中数据的编辑和修改
- 3、简单命令文件的调试与修改

(四) 数据录入操作 (20%)

1、利用字处理软件在 10 分钟内输入 250 个汉字 (误码率 < 0.45%)

2、利用字处理软件在 10 分钟内输入 800 个英文字符 (误码率 < 0.15%)

说明:

本大纲适用于考初级程序员之在职人员或在学学生 (包括普通中学、职业中学、中专、中技) 其中:

1、下午试题拟分成 A、B 两卷, 主要是对象不同以 BASIC 或 DBASE Ⅲ 侧重不同来区分。

2、上机操作试题拟分成 A、B 卷, 除侧重点不同外, 还考虑机型 IBM-PC 或 APPLE II。

3、上、下午均得合格者, 可进入上机试。笔试、上机操作均合格者, 由市计算机开发与领导小组办公室、市劳动局、市科委发给统一合格证书。 ■

## 浪潮 LC 0530-D 系统板故障检修一例

广西河池氮肥厂计算中心 伍树炎

当前,国内越来越多的单位用上了 286、386 微机,随着应用的普及,不可避免地会碰上各种硬件故障。现将我们在 286 机上的一点经验奉给同行参考。

故障现象:开机后屏幕显示出错信息:

"System CMOS checksum bad-Run Setup"

"Realtime clock error"

并提示:

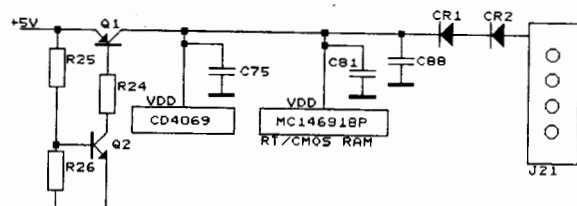
"Press <F1> to resume, <F2> to Setup"

根据提示,键入 F2。重新设置后,机器工作正常;但断电后再开机,又出现上述出错信息。

故障分析与检修:该机采用"SETUP"软件菜单形式的配置选择,存放于系统板上的 RT/CMOS 电路见图:

RT/CMOS RAM 供电原理图

根据出错信息提示,该故障与 RT/CMOS RAM 电路有关,而重新设置后,机器工作正常,说明 RT/CMOS RAM 电路功能正常,但断电后再开机,则又出现出错信息,由此可知该故障与 RT/CMOS RAM 的供电电路有关。查图示 RT/CMOS RAM 部分供电电路,拔下 3.6V 电池测其开路端电压为 0V,换上一只新的 3.6V 电池,插入 J21 插座,关闭主机电源。测 3.6V 干电池在路端电压由原来的 3.6V 降至 2.5V 左右,测电池负载电流高达 20mA,说明电池供电回路有严重漏电,测 Q1 发射极电压为 0V,说明 Q1、Q2 正常;取下干电池,测 Q1 集电极对地电阻,只有 100Ω 左右。先后焊开 C88、C81 的一端,漏电电阻仍为 100Ω 左右;当焊开 C75 的一端时, Q1 集电极对地电阻恢复到趋于无穷大,焊下 C75 测量,其漏电电阻只有 100Ω 左右,用一只 0.01μF 电容换下已坏的 C75,故障排除。



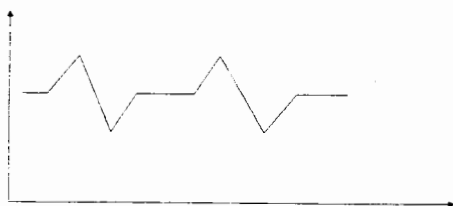
## OKI-8320 打印机维修一例

湖南省电子研究所 胡红

故障现象:开机,电源指示灯一闪即灭,打印机再无任何反应。

分析与检修:由现象首先想到的是电源。打开机盖,开机后再测发现+5V 电压没有,估计由于负载问题使之被保护。取下连到控制板上的电源插头,再测+5V 到地之间的电阻,注意不是机壳地,发现其间只有 12Ω,所以一开机便使+5V 被保护。该种型号的打印机为体积最小的 24 针并自带汉字库的打印机,它的机械和电路都设计得非常精巧,唯一的一块控制板采用单片 8051 为 CPU,外加两片 2732EPROM 及 4 片 4416 动态 RAM (16K×4),以及另外一些芯片,除了 EPROM 采用插座外,其余芯片全部焊在板上。

当发现+5V 对地电阻过小时,不要急于采用割线法,而应先小心拔下两片 EPROM,再测电阻。我已经三次碰到是 EPROM 坏而造成+5V 对地电阻过小。本例也是如此。从好机器上拷贝两片 EPROM 插到板上,注意位置及方向。此时再开机,面板电源灯正常,而告警灯和联机灯(SELECT)不停闪烁,小车纹丝不动。这种现象表明字车驱动或 RAM 有问题。关机,用手移动小车,发现没有什么阻碍,此时就应首先查看 RAM 了。该机用 4 片 4416 组成了 32K 的动态 RAM,用示波器检查它们的各脚信号,行、列刷新及地址线信号波形正常,而数据线 DQ<sub>1</sub> 及 DQ<sub>2</sub> 不正常,如图。此时可以采用分割法来确定到底是哪块 RAM 有问题,换之,正常。由于板上连线很细,希望换片子时注意。



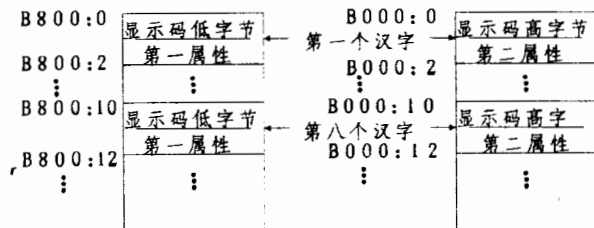
# ANSI.SYS 汉字滚屏乱符的解决办法

中国长城计算机集团公司湖南公司 马国忻

在 DOS 操作系统中, ANSI.SYS 是一个可安装的键盘及屏幕扩展驱动程序, 利用它可加强或增加屏幕显示功能, 可重新对键盘按键进行定义, 有许多应用程序要求操作系统要安装它, 但是在长城系列微机中安装该程序后, 滚屏时汉字会发生错乱, 这是因为 ANSI.SYS 不具备专门处理汉字滚屏功能, 因而发生上述现象。

由 debug 加载 ANSI.SYS 后, 可知程序中滚屏部分位于 2CD~2D4 和 2EB~2F2 两处, 前部分在显示方式 7(MDA)时产生作用, 后部分在其它显示方式下产生作用, 而在 MDA 方式下不能显示汉字, 故只需考虑 2EB~2F2 处, 该程序滚屏采用直接移动显示缓冲区内容的方法, 除 MDA 方式外, 显示缓冲区为 B800 段, 该程序正是在此段内移动信息达到滚屏目的, 但这仅仅只在英文和图形状态下正确。

长城微机具有独特的汉字系统, 它显示汉字的信息除占用缓冲区 B800 段外, 还占用 B000 段 (新系列机, CH 机占用 C000 段), 下图是汉字显示缓冲区示意图, 每个汉字在 B800 段和 B000 段占用相对应偏移处的 4 个字节, 这两段中内容缺一不可, 而且相应位置也不能错动。所以在 ANSI.SYS 中仅滚动 B800 段中的内容便破坏了汉字结构, 出现乱字错字现象。



知道原因后, 仅作小的修改便可解决此问题。在移动 B800 段内容时, 同时将 B000 段内容作相应的移动, 那么屏幕上滚动的便是完整无错的汉字, 现作修改如下:

2EB~2F2 为修改部分, 700~72A 为增加部分。注意, 该 ANSI.SYS 的 DOS 版本为 3.2

```
1112:0700 89E9 MOV CX,BP ;CX←80
1112:0702 F3 REPZ
1112:0703 AB STOSW ;清第一缓冲区最后一行(第25行)

1112:0704 8CC0 MOV AX,ES ;取第一缓冲区段址
1112:0706 2D0008 SUB AX,0800 ;转第二缓冲区段址
1112:0709 8EC0 MOV ES,AX
1112:070B 8ED8 MOV DS,AX ;数据段指向第二缓冲区

1112:070D B98007 MOV CX,0780 ;24行内容含1920字节
```

```
1112:0710 31FF XOR DI,DI ;DI指向第一行
1112:0712 BEA000 MOV SI,00A0 ;SI指向第二行
1112:0715 F3 REPZ
1112:0716 A5 MOVSW ;将第二行至第二十五行上移一行
1112:0717 B8000F MOV AX,0F00 ;作好清第二缓冲区最后一行的准备
1112:071A 89E9 MOV CX,BP ;CX←80
1112:071C F3 REPZ
1112:071D AB STOSW ;清第二缓冲区最后一行(第25行)
1112:071E 8CC0 MOV AX,ES
1112:0720 050008 ADD AX,0800
1112:0723 8EC0 MOV ES,AX ;恢复数据段原址
1112:0725 8ED8 MOV DS,AX
1112:0727 FB STI ;开中断
1112:0728 E9C8FB JMP 02F3 ;转回
1112:02EB FA CLI ;关中断
1112:02EC F3 REPZ
1112:02ED A5 MOVSW ;将第一缓冲区第二行至第二十五行上移一行
1112:02EE 8BC3 MOV AX,BX ;作好清第一缓冲区最后一行的准备
1112:02F0 E9D0D4 JMP 0700 ;跳转
```

## IBM PC / XT 及兼容机 系统板维修一例

岳阳巴陵公司化工区信息站 董纯坚

故障现象: 开机时, 系统自检后, 有时屏幕显示: "PARITY CHECK" 系统死机, 软盘无法启动; 有时自检后, 能正常用机, 但关机再启动时, 故障再现。

故障分析及排除:

不断地开关机, 使故障稳定下来, 根据屏幕可知, 该故障一般发生在内存校验芯片或内存校验电路。首先用一片好的内存芯片替换内存 BANK1 的奇偶位, 故障未消失, 用示波器检查 NMI 作屏蔽中断控制电路, 经查 8088, U87 (S08), U86 (LS74), U72 (LS10) 均未发现问题, 接着检查 8237DMA 电路, 发现扩充命令缓冲器 U13 (LS243) 输出引脚为半高电平, 正常情况该脚信号应为低电平有效, 由此可推断扩充存储器读信号 XMEMR 出错, 造成系统死机, 更换 U13, 故障消失。

# 软化中华学习机使用汉字的“硬环境” ——用户如何修改汉字处理程序

深圳华明电子实业公司技术部 李庆发

CEC-I 中华学习机在 Apple II 的基础上改造并增加了汉字处理功能，它把汉字处理部分的有关程序和汉字库都固化在 ROM 里，为用户提供一个使用汉字的“硬环境”，无需从外存设备引导软汉字系统，方便了用户的使用。但由于它的“硬”，用户在使用上的一些特殊需要，乃至汉字处理程序本身存在的一些不足或缺陷，都要对它进行修改时，便不是那么容易的了。当然有条件备有象 EPROM 擦除器、编程器等设备的用户可以对固化了的程序按新的要求重新烧写。可是烧写之后，下一次修改又要做同样的工作，这种情况特别发生在程序调试过程中，既麻烦又加快 EPROM 器件的失效。

本文作者在中华学习机的开发实践中，时常需要对固化的汉字处理程序进行修改，从中找到了一种行之有效的方法，现将该方法介绍给对中华学习机的深入使用有兴趣的读者。

中华学习机的汉字处理程序固化在辅存 ROM 的高地址部分，其地址范围为 \$EC00-\$FFFF (指 CEC-I 1.1 版本，1.2、1.21 版本在 \$E000-\$FFFF)。由于它与主存 ROM 中的监控程序地址空间存在重叠现象，无法用监控命令 L 直接进行查看，必须用专用程序将它搬至低地址的 RAM 空间。如用 MOVERHZ 程序 (附后) 中第 23 至 34 行的一段指令，将汉字处理程序搬至 RAM 的 \$2C00-\$3FFF 空间内。这时可对这段空间使用所有监控命令，可以发现存放在这里的程序是不可正确执行的，原因是其中的大量跳转指令 (JMP) 使用的是直接地址，搬动程序并未按新的地址空间相应地修改这些直接地址，执行时它将跳至监控程序中对应的地址空间去执行无关的指令，必然出错，所以必须用“地址重定位” (专业术语，它使在某段地址空间的程序，存放在另一地址空间也能正确执行，这里实际是静态地址重定位) 程序进行地址调整。如用程序 CHANGEDADD (源程序附后) 修改跳转指令中的直接地址，同时还要修改一些指向数据存放区的指针和间接寻址指令中的向量，通过地址重定位后存放在 RAM 中的这段汉字处理程序就可正确的执行了 (注意：不要将汉字处理程序存放在 RAM 空间的 \$4000-\$BFFF 中，以避免引起程序执行时的混乱，因为汉字处理程序中进行拼音码表的查找和取字符点阵数据时，需要切换到辅存的这段地址空间去)。用户可根据自己的需要对这段程序方便地进行修改、调用。同时，如果将输入、输出空间 \$C300-\$C3FF 中的连接程序做适当的修改，存放在地址 \$300 开始处，改变连接向量 KSW (\$38、\$39) 和

CSW (\$36、\$37) 以指向新的地址，则完全可用 RAM 中的汉字处理程序替换 ROM 中的汉字处理程序，达到了软化的目的。

上述方法可以满足一般用户的要求，但显然存在两大缺点：第一，它占用了用户的常用空间。第二，容易为用户程序所破坏。能否将汉字处理程序存放在高地址的 16KB 扩充 RAM 中呢？这是完全可以的。程序 MOVERHZ (源程序附后) 就是实现将辅存 ROM 中的汉字处理程序复制到辅存 RAM 的对应地址中，程序 INTERFACE (源程序附后) 则在地址 \$300 开始处形成有关的连接子程序。它编译后的目标程序如下：

• 300LL

0300-	4C 43 03	MP	\$0343
0303-	29 7F	AND	#\$7F
0305-	C9 11	CMP	#\$11
0307-	F0 0A	BEQ	\$0313
0309-	8D 80 C0	STA	\$C080
030C-	20 CE EF	JSR	\$EFCF
030F-	8D 82 C0	STA	\$C082
0312-	60	RTS	
0313-	8D 82 C0	STA	\$C082
0316-	A9 10	LDA	#\$10
0318-	8D B0 C0	STA	\$C0B0
031B-	20 2F FB	JSR	\$FB2F
031E-	20 89 FE	JSR	\$FE89
0321-	4C 93 FE	JMP	\$FE93
0324-	8D 80 C0	STA	\$C080
0327-	20 A3 F7	JSR	\$F7A3
032A-	C9 91	CMP	#\$91
032C-	F0 03	BEQ	\$0331
032E-	4C 0F 03	JMP	\$030F
0331-	8E CA 03	STX	\$03CA
0334-	8C CB 03	STY	\$03CB
0337-	20 13 03	JSR	\$0313
033A-	A9 8D	LDA	#\$8D
033C-	AE CA 03	LDX	\$03CA
033F-	AC CB 03	LDY	\$03CB
0342-	60	RTS	
0343-	A9 00	LDA	#\$00
0345-	8D 7B 06	STA	\$067B
0348-	A9 03	LDA	#\$03
034A-	AA	TAX	
034B-	4D 7B 04	EOR	\$047B
034E-	F0 0D	BEQ	\$035D
0350-	8A	TXA	
0351-	8D 7B 04	STA	\$047B
0354-	8D 80 C0	STA	\$C080
0357-	20 00 EC	JSR	\$EC00
035A-	4C 0F 03	JMP	\$030F
035D-	8D 80 C0	STA	\$C080



```
0360- 20 10 EC JSR $EC10
0363- 4C 0F 03 JMP $030F
```

可见, 新的汉字系统入口点为 \$300, 而新的 CSW 和 KSW 则应指向 \$303 和 \$324。

使用扩充 RAM 这个方法, 可以不必进行地址重定位, 不占用用户的低地址常用空间 (仅用第三页的几十个单元), 通过软开关又可设置辅存 RAM 中程序的写保护, 避免用户程序的无意破坏, 用户对辅存 RAM 中程序的修改, 或增加程序量 (还有 6KB 的辅存 RAM 空间) 也都较为方便。除此之外, 用户也可以从外存设备引导自己的汉字处理程序存放在这个空间里, 利用中华学习机的硬字库形成一个软化了的汉字系统。

下面举出一个修改软化了的汉字处理程序的例子, 以示读者。

例子: 使中华学习机全屏显示十一行。

CEC-I 中华学习机为用户提供十行的屏幕显示区, 而第十一行只能做为状态提示行, 为系统专用, 用户一般难于用到。当然, 可以用各种方法编制一些子程序去使用它, 可还是存在着局限性, 未能象使用其它十行一样方便。这个例子只要修改六个地址单元的值就可让用户使用到十一行屏幕显示区, 各行的使用具备一致性, 并可根据需要随时相互切换十行、十一行, 如显示时用十一行, 输入时用十行。

先在辅存扩充 RAM 中形成一个软化的汉字处理系统, 运行如下程序段形成十一行的屏幕显示区:

```
LDA $C081
LDA $C081
LDA #$0A
STA $F168
STA $F257
STA $F332
LDA #$0B
STA $F264
STA $F359
STA $F411
STA $C082
RTS
```

恢复十行时用如下程序段:

```
LDA $C081
LDA $C081
LDA #$09
STA $F168
STA $F257
STA $F332
LDA #$0A
STA $F264
STA $F359
STA $F411
STA $C082
RTS
```

或者恢复连接向量 CSW 和 KSW 的原值, 让它们指向固化的汉字处理程序, 也能恢复十行显示, 再回十一行时, 再修改这两个连接向量, 指向新的汉字处理程序。在

这里, 如果系统已引导了 DOS 操作系统, 则 CSW 和 KSW 已固定的指向 DOS 的输出、输入程序的入口地址 \$9EBD \$9E81 了, 不能再修改 CSW 和 KSW 的值。如强行将这两个连接向量改至新的汉字处理程序输出、输入入口地址, 那么 DOS 的拦截作用将失效, 系统无法执行 DOS 命令。这时, 只能修改与 \$36、\$37 和 \$38、\$39 等效的一对输入、输出寄存器 \$AA53、\$AA54 和 \$AA55、\$AA56。修改方法是一样的。

有兴趣的读者可以针对 CEC-I 1.1 版中一些不足或缺陷试着修改, 如:

1. 开放 \$7E, 即字符“~”。
2. 恢复“从光标清到行尾”的功能
3. 允许汉字方式下出彩色。
4. 改正退出汉字方式时西文屏幕存在闪烁字符的现象。
5. 改正拼音方式中提示字的闪动。
6. 恢复拼音方式中空格键的功能。

```
1;
2;
3* INTERFACE $0300 *
4;
5;
6;
7;          ORG      $300
8;
9 INTCHZ    JMP      INTO
10;
11CSW       AND      #$7F; 屏蔽高位
12          CMP      #$11
13          BEQ      XW; 是西文键则分支
14          STA      $C080; 禁止 ROM, 允许读扩充 RAM
15
16          JSR      $EFCE; 转扩充 RAM 中的 CSW 处理程序
17
18REROM      STA      $C082; 禁止扩充 RAM, 允许读主 ROM
19
20          RTS
21
22XW         STA      $C082
23          LDA      #$10
24          STA      $C0B0; 允许彩色图形显示
25          JSR      $FB2F; 将 CRT 设定为以 24 行 40 列的格式显示
26
27;
28          JSR      $FE89; 设定键盘为输入设备
29          JMP      $FE93; 设定 CRT 为输出设备
30
31KSW        STA      $C080
32          JSR      $F7A3; 转扩充 RAM 中的 KSW 处理程序
```



### ☆经销及维修中心

广州：广东省计算中心 广州市连新路省科学馆内 电话：340642 邮码：510033

北京：中国铁道科技咨询公司 北京市西直门外大柳树北站 电话：890461 转385 邮码：100081

### △代理商

北京：华通电子有限公司

上海：中国计算机用户协会M68000系列机协会

重庆：中国保利科技有限公司传真机维修中心

西安：国营红华仪器厂

成都：深圳胜利仪器有限公司西南经营部

长春：长春电力电脑技术开发公司

武汉：武汉新大陆公司

哈尔滨：哈尔滨亚高卡科技公司

## 现代宫殿从这里步入

### 访广东省计算中心

随着电脑的普及，拥有雄厚技术力量、大型计算机系统以及能提供周到便利服务的广东省计算中心日益引人注目。

为客户提供超前的先进电子计算机系统以及从买、学、用、修一条龙全方位服务是广东省计算中心在社会服务上的两大优势，而这两大优势的日益发挥使广东省计算中心能从大陆得到越来越多的客户。

一九八六年，传统型的手按字粒打字机开始被新的科技成果—电子打字机所取代。广东省计算中心及时满足大陆用户的迫切需求，率先在国内推出了中英文电子打字机，并以一次销量达四百台的数量而引起同行的注目。

一九八七年，随着大量进口的IBM电脑进入维修期，广大用户急切希望能借助先进的检测设备寻找故障及时修复。广东省计算中心又率先在大陆推出了研制的专门维修IBM电脑的在线测试仪，从而大大方便了用户，得到了用户的一致好评。

一九八八年，广东省计算中心又在已有的科研成果基础上更进一步，与新加坡、香港业商合作，推出适合多种

型号电脑维修用的新型“创能”测试仪，为国内二百多个维修站充实了可靠适时的技术力量。

用户是最公正的“法官”，商品好坏的确全由顾客评说。汕头市统计局由于借助了先进的“创能”测试仪，半小时内便查找到曾花耗两个月时间而查不出的故障所在。

如今，广东省计算中心属下的专业服务机构—技术服务部已以出色的表现而赢得了近千个大陆客户。

为了普及电子计算机，广东省计算中心连续几年在广东各地举办了近百个电脑大专班，引导更多的人步入这通向现代化的殿堂。

眼下，广东省计算中心已组织了逾三十名软件人员专门为企业使用电脑设计编写软件，中共广东省委办公自动化系统也交由该中心设计实施。广东省计算中心以此良好的服务为社会各界的现代化管理创造了条件。

广东省计算中心拥有各种型号的进口电脑系统，在这里，人们可以根据自已的意愿选购到满意的设备。

现代化的光辉前景在向我们招手。眼下，广东省计算中心正在着手拟制广东计算机发展规划，为未来勾划蓝图。广东省计算中心愿与广大客户，业界同行携手努力，共创美好明天。

摘自1990年10月25日《粤港信息报》

# 操纵杆的键盘模拟

广东茂源公司技术工程部 关本立

目前,市面上不少APPLE II上的游戏都是用操纵杆控制的,如将其改为用键盘或键盘操纵杆均可控制,无疑方便了广大用户。

以游戏SPACE EGGS为例,原版只能用操纵杆控制,我通过对其分析和修改,使其能用键盘或操纵杆操作。下面简单介绍修改方法。

我们知道,读操纵杆值前必定会访问\$C070这个单元,而且大多软件中都使用LDA \$C070这个指令访问,因而对大多数用操纵杆操作的软件来说,将整个软件扫描一次,寻找AD70C0这三个字节(LDA \$C070的操作码),就找到了读操纵杆值子程序在软件中的位置。除此之外还要找出判断SW0有无按下的子程序在软件中的位置(有些软件还会用上SW1等),这两种类型均找到后(注意:这两类型的值有可能在软件中多处出现),只要将其改为跳出至自编的读键盘程序,那么就可以用键盘操作了。

修改前首先BLOAD SPACE EGGS,再CALL-151进入监控,在监控状态下键入:

18E4: 20 00 03 N

18EE: 20 00 03

N 1C40: 20 00 03

N 4B00: 20 00 03

再键入以下汇编程序:

\*F3FLLLLL

```
0F3F- A0 00 LDY ##00
0F41- B9 4F 0F LDA $0F4F,Y
0F44- 99 00 03 STA $0300,Y
0F47- C8 INY
0F48- C0 B1 CPY ##B1
0F4A- D0 F5 BNE $0F41
0F4C- 4C 00 10 JMP $1000
```

```
0F4F- A9 01 LDA ##01
0F51- 20 A8 FC JSR $FCA8
0F54- AD 00 C0 LDA $C000
0F57- C9 98 CMP ##98
0F59- F0 F9 BEQ $0F54
0F5B- C9 B0 CMP ##B0
0F5D- 90 14 BCC $0F73
0F5F- C9 BA CMP ##BA
0F61- B0 10 BCS $0F73
0F63- 0A ASL
0F64- 0A ASL
0F65- 0A ASL
0F66- 0A ASL
0F67- 18 CLC
0F68- 69 01 ADC ##01
0F6A- 8D 01 03 STA $0301
0F6D- 20 98 03 JSR $0398
0F70- AD 00 C0 LDA $C000
0F73- C9 93 CMP ##93
0F75- D0 10 BNE $0F87
0F77- AD F4 13 LDA $13F4
0F7A- 49 30 EOR ##30
0F7C- 8D F4 13 STA $13F4
0F7F- 8D 1C 43 STA $431C
0F82- 8D 10 C0 STA $C010
0F85- 10 25 BPL $0FAC
0F87- C9 8B CMP ##8B
0F89- D0 0B BNE $0F96
0F8B- A9 4C LDA ##4C
0F8D- 8D A3 0C STA $0CA3
0F90- A9 78 LDA ##78
0F92- A0 03 LDY ##03
0F94- D0 0D BNE $0FA3
0F96- C9 8A CMP ##8A
0F98- D0 12 BNE $0FAC
0F9A- A9 A0 LDA ##A0
0F9C- 8D A3 0C STA $0CA3
0F9F- A9 00 LDA ##00
0FA1- A0 EA LDY ##EA
0FA3- 8D A4 0C STA $0CA4
```

```
32 PHA
33 INY
34 BNE OPE11
35 INC $07
36 INX
37 BNE OPE11
38 RTS
39 OPE11 INY
40 BNE OPE22
41 INC $07
42 INX
43 BNE OPE22
44 RTS
45 OPE22 LDA ($06),Y
46 CMP ##$EC
47 BCC OPE3
```

```
48 AND ##$3F
49 STA ($06),Y
50 PLA
51 PLA
52 PLA
53 PLA
54 JMP CONTI
55 OPE3 PLA
56 TAX
57 PLA
58 TAY
59 PLA
60 STA $07
61 PLA $06
62 JMP CONTI
```

63 .....  
64 \* END

# APPLE II PLUS 中断系统剖析 及其在软件加解密中的应用

舟山市电力公司变电工区 傅剑

【提要】本文将详细介绍 APPLE II PLUS 微型计算机各种中断方式的响应过程和监控程序内中断服务程序运行情况。并通过实例把中断应用于软件的加解密中。笔者希望本文能使读者对 APPLE II PLUS 系统的了解有所帮助。

我们把计算机能够对一信号线的上升(或下降)有所反映,并能瞬间地将处理器目前的状况保留起来,且迅速地将控制权转移给计算机的某一程序的过程,称为中断反映。被中断运行的程序称主程序,接受控制权的程序称中断服务程序。

在 APPLE II PLUS 中的 6502 处理器有四种不同的中断方式,我们称它们是复位中断 RESET、不可屏蔽中断 NMI、可屏蔽中断 IRQ 和软件中断 BRK。这四种中断方式分别可以通过按“Ctrl-Reset”键使 RESET 线出现正跳

变, NMI 线出现负跳变, IRQ 线出现低电平且状态寄存器允许中断,执行 BRK 指令而得到响应。下面分别介绍这四种中断的响应过程:

## (一) .RESET 中断方式

当按下“CTRL-RESET”键或 RESET 出现正跳变时, 6502 自动地去读存储单元 \$FFFC 所存放的值,并将它送程序计数器 PC 的低八位,接下来 6502 又自动地去读 \$FFFD 存储单元的值,并将它送程序计数器 PC 的高八位。从此以后,程序将从 \$FFFC 和 \$FFFD 所指的地址

```

0FA6- 8C A5 0C STY  #0CA5
0FA9- 20 98 03 JSR  #0398
0FAC- AD A3 0C LDA  #0CA3
0FAF- C9 4C    CMP  #4C
0FB1- D0 10    BNE  #0FC3
0FB3- AD 00 C0 LDA  #C000
0FB6- C9 A0    CMP  #A0
0FB8- F0 03    BEQ  #0FBD
0FBA- A9 3F    LDA  #3F
0FBC- 60      RTS
0FBD- 8D 10 C0 STA  #C010
0FC0- A9 BF    LDA  #BF
0FC2- 60      RTS
0FC3- AD 61 C0 LDA  #C061
0FC6- 60      RTS
0FC7- A4 FB    LDY  #FB
0FC9- AD 00 C0 LDA  #C000
0FCC- C9 88    CMP  #88
0FCE- D0 04    BNE  #0FD4
0FD0- A0 00    LDY  #00
0FD2- D0 06    BNE  #0FDA
0FD4- C9 95    CMP  #95
0FD6- D0 02    BNE  #0FDA
0FD8- A0 68    LDY  #68
0FDA- C4 FB    CPY  #FB
0FDC- D0 01    BNE  #0FDF
0FDE- 60      RTS
0FDF- 90 03    BCC  #0FE4
0FE1- 4C 70 0C JMP  #0C70
0FE4- 4C 86 0C JMP  #0C86
0FE7- AD 10 C0 LDA  #C010
0FEA- 8A      TXA
0FEB- 48      PHA
0FEC- 98      TYA
0FED- 48      PHA
0FEE- A2 00    LDX  #00
0FF0- A0 10    LDY  #10
0FF2- 88      DEY

```

```

0FF3- D0 FD    BNE  #0FF2
0FF5- 8D 30 C0 STA  #C030
0FF8- CA      DEX
0FF9- D0 F5    BNE  #0FF0
0FFB- 68      PLA
0FFC- A8      TAY
0FFD- 68      PLA
0FFE- AA      TAX
0FFF- 60      RTS

```

最后按 CTRL-C 返回 FBASIC 状态, 键入 BSAVE SPACE EGGS+, A\$ F3F, L\$ 7CC1 存盘, 至此修改完成。你可以试一试运行这个游戏, 一定更加有趣吧?

经修改后的游戏控制键如下:

ESC: 游戏暂停, 直到按下除 ESC 外的任意一键为止。

CTRL-S: 关闭或开启游戏声音。

CTRL-J: 设定使用游戏操纵杆操作, 这时操纵杆对游戏的控制和未修改前的使用方法相同。

CTRL-K: 设定使用键盘操作。这时可以用←和→键分别控制向左或向右, 空格键发炮。

0.1.2...9: 改变游戏的速度, 0 为正常速度, 9 为最慢速度。原版的这个游戏也许速度太快了吧? 加上这些键后相信你一定能轻易地打破自己以前的最高纪录!

以上介绍的虽然只是对游戏 SPACE EGGS 的修改, 但其原理对其它用操纵杆软件的修改也是适用的。对现成软件的修改, 大都不采用直接读整个软件的方法, 只要寻找自己感兴趣的部分并进行修改, 那么花的时间就会少很多了。■

开始执行。APPLE II PLUS 微电脑在 \$FFFC 和 \$FFFD 两个存储单元存放着 RESET 的中断向量值是 \$62 和 \$FA, \$FA62 即是 RESET 中断服务程序的入口。

## (二) .NMI 中断方式

在 NMI 中断响应周期中, 6502 将自动进行以下工作:

<1> 将主程序断点值 (即程序计数器 PC 值) 送入堆栈保存, 进堆栈顺序是先高位后低位。

<2> 将标志寄存器 P 的内容送入堆栈保存

<3> 将标志寄存器 P 的中断禁止位 I 置 1, 禁止响应其它 I/O 设备提出 IRQ 中断请求。

<4> 从地址 \$FFFE 和 \$FFFF 的两个单元中取出地址码送 PC, 从此以后, 程序即从此地址码开始工作。地址 \$FFFE 和 \$FFFF 两存储单元中存放着 BRK 中断向量值 \$40 和 \$FA, \$FA40 即是 BRK 中断服务程序的入口地址。

## (三) .IRQ 中断方式

在 IRQ 中断响应周期中, 6502 将自动进行以下工作:

<1> 将主程序断点的值 (即 PC 的值) 送入堆栈保存, 顺序是先高位后低位, 并把标志寄存器 P 的 BRK 指令标志位 B 置 0, 以标和 BRK 中断的区别。

<2> 将标志寄存器 P 的值送入堆栈保存。

<3> 将标志寄存器 P 的中断禁止标志位置 1, 禁止其它 I/O 设备提出 IRQ 中断请求。

<4> 从地址 \$FFFE 和 \$FFFF 的两个单元中取出地址码送 PC, 从此以后, 程序即从此地址码开始工作。地址 \$FFFE 和 \$FFFF 两存储单元中存放着 IRQ 中断向量值 \$40 和 \$FA40 即是 IRQ 中断服务程序的入口地址。

## (四) .BRK 软件中断方式

BRK 是 6502CPU 151 条指令中唯一的软件中断指令, 由于它不需利用硬件线路电平的变化来实现中断, 因此又称 BRK 指令为软件断指令, 当机器执行该指令时, 6502 将自动地进行以下工作:

(1) 将主程序断点的值 (即 PC 的值) 送入堆栈保存, 顺序是先高位后低位指令时, 需注意被送进堆栈保存的断点地址并不是位于 BRK 指令的下一条指令的下一条指令地址, 而是比它多 1。

(2) 将标志寄存器 P 的 BRK 指令标志位 B 置 1, 以表示该中断是执行 BRK 指令引起。

(3) 将标志寄存器 P 的值送入堆栈保存。

(4) 将标志寄存器 P 的中断禁止标志位置 1, 禁止其它 I/O 设备提出 IRQ 中断请求。

(5) 从地址 \$FFFE 和 \$FFFF 的两个单元中取出地址码送 PC, 从此以后, 程序即从此地址码开始工作。地址 \$FFFE 和 \$FFFF 两存储单元中存放着 BRK 中断向量值 \$40 和 \$FA, \$FA40 即是中断服务程序的入口地址。

以上四种中断在得到响应后, 都把控制权转给各自的中断服务程序, 由于四种中断服务程序所执行的动作有相当大的不同, 因此笔者将它们分开来描述。

1. RESET 中断服务程序从地址 \$FA62 开始执行, 动作如下:

FA62	CLD	;清十进位模式
FA63	JSR \$FE84	;设置字符为白底黑字
FA66	JSR \$FB2F	;设置画面格式及显示资料来源
FA69	JSR \$FE93	;设定显示屏幕为输出设备
FA6C	JSR \$FE89	;设定键盘为输入设备
FA6F	LDA \$C058	;置游戏 I/O 插座零号开关输出为 1
FA72	LDA \$C05A	;置游戏 I/O 插座壹号开关输出为 1
FA75	LDA \$C05D	;置游戏 I/O 插座贰号开关输出为 0
FA78	LDA \$C05F	;置游戏 I/O 插座叁号开关输出为 0
FA7B	LDA \$CFFF	;清除扩充 ROM 的开关
FA7E	BIT \$C010	;清除键盘触发信号
FA81	CLD	;清十进制工作模式
FA82	JSR \$FF3A	;调用 BELL 使扬声器发声
FA85	LDA \$3F3	;把 \$3F3 单元的内容存入累加器 A
FA88	EOR # \$A5	;累加器 A 和 \$A5 进行异或运算
FA8A	CMP \$3F4	;检查是否执行过打开电源的启动动作
FA8D	BNE \$FAA6	;若没有启动过, 则转至 \$FAA6 执行加电启动动作
FA8F	LDA \$3F2	;若执行过, 则检查 \$3F2 单元是否为零
FA92	BNE \$FAA3	;若 ≠ 0, 则转移至地址 \$FAA3 开始执行
FA94	LDA # \$E0	;若 = 0, 则取 \$E0 到累加器 A
FA96	CMP \$3F3	;比较地址 \$3F3 的内容和累加器 A 内的 \$E0 是否相等
FA99	BNE \$FAA3	;若 ≠ \$E0, 则转向地址 \$FAA3 开始执行
FA9B	LDY # \$03	;把 \$03 存入寄存器 Y 中
FA9D	STY \$3F2	;若 = \$E0, 则将 \$03 存入地址 \$3F2 中
FAA0	JMP \$E000	;转入 BASIC 语言
FAA3	JMP(\$3F2)	;间接转至 \$3F2、\$3F3 单元所指向的地址 (\$E003)

2. NMI 中断服务程序从地址 \$3FB 开始执行, 该地址内容监控程序不予设置, 可由用户自己编写服务程序, 需注意该地址被 DOS 设定为 JMP \$FF65。

3. IRQ 和 BRK 中断服务程序从地址 \$FA40 开始执行, 动作如下:

FA40	STA \$45	;把累加器 A 的内容存入在零页的存储区 \$45 单元中
FA42	PLA	;将 (P) 从堆栈中取回 A 中
FA44	PAH	;将 (P) 放回堆栈
FA44	ASL A	
FA45	ASL A	
FA46	ASL A	;将 A 左移 3 次, 以判别标志位 B
FA47	BMI \$FA4C	;若由 BRK 指令产生中断, 则转移到 \$FA4C 地址
FA49	JMP (\$3FE)	;若是 IRQ 产生中断, 则间接转移到向量 (\$3FE, \$3FF) 指向的地址, 需注意监控程序对该向量不予设定, 可由用户自己设置。DOS 把该向量设置为 \$FF65, \$FF65; 子程序作用是进入监控命令处理器中。
FA4C	PLP	;退栈恢复 P
FA4D	JSR \$FF4C	;调用 SAV1 子程序保存 X、Y、P、S

FA50	PLA	寄存器在零页的暂存单元中 ;退栈取回主程序断点地址的低位存入 A 中
FA51	STA \$3A	;把断点低位地址存入程序计数器在零页的暂存器中
FA53	PLA	;退栈取回主程序断点地址的高位存入 A 中
FA54	STA \$3B	;把断点高位地址存入程序计数器在零页的暂存器中
FA56	JMP (\$3F0)	;BRK 中断经向量 (\$3F0, \$3F1) 间接转移, 该向量正常是由监控程序设定, 指向 \$FA59 单元, \$FA59 子程序作用是在屏幕上印出 A、X、Y、P、S 的值, 最后进入监控命令处理器中。

### 一.加密实例

从加密的角度来看, 为了保护程序不被 LIST 和永远拥有控制权, 就必须封锁一切可能的中断 (即把中断控制权掌握在程序中), 对于 6502 为 CPU 的计算机而言, 也就是封锁上述四种中断方式。我们在中断响应过程中可以发现, 中断是通过读取 ROM \$FFFA~\$FFFF 单元中的向量把控制权转移给中断服务程序。但由于 ROM 不能被程序改写, 这样使我们无法在 \$FFFA~\$FFFF 单元中截取中断控制权。不过我们可以在 RAM 的第三页截取中断控制权, 这是因为中断服务程序最后转移地址, 总是依靠 RAM 第三页的向量 \$3F0~\$3FF 单元的内容来确定。下面就是笔者运用上述原理, 通过对 DOS3.3 中的 BOOT1 以及主程式的修改来截取中断控制权, 使向量指向重新启动磁盘机。经过修改的系统盘, 能比较彻底地封锁各种中断反应, 如果程序有完善的容错能力, 则该加密系统将永远拥有控制权。

加密步骤如下:

<1> 启动 DOS3.3

<2> 输入命令 "CALL-151", 进入监控模式。

<3> 输入:

```
* A702:4C 00 C6
* 9E70:4C 00 C6
* 9E7C:4C 00 C6 00 C6
* B64A:4C B3 08
* 9E52:00 C6
* B6B3:A9 00 8D F2 03 A9 C6
```

```
8D F3 03 49 A5 8D F4 03 6C FD 08
```

<4> 键入 "CTRL-C" 命令返回 BASIC 模式

<5> 键入 "INIT HELLO" 命令格式化一磁盘

注: 为了防止解密者有意使 DOS3.3 出错 (I/O ERROR) 中断, 本加密程式取消了 DOS 出错功能。本加密程序也适用于 CEC-I 电脑。

### 2.解密实例

中断运行中的加密程序, 导致程序控制权转向解密

者, 是解拆程序的一种重要手段。现在一般的 APPLE II PLUS 电脑都配有 16KRAM 卡, 笔者把 BASIC 和监控程序移入 16KRAM 卡, 然后对监控程序内的中断向量进行修改, 使中断控制权转向解密者设置的入口, 如 BASIC (\$E003) 或监控命令处理器 (\$FF65) 等。在监控中修改中断向量, 可以避开加密者在 RAM 区第三页设定的中断向量, 因此有很好的解密效果。

设置过程如下 (修改中断向量, 使其指向监控命令处理器):

<1> 把 RAM 卡插入 1 号槽

<2> 启动 DOS3.3

<3> 输入 "CALL-151" 命令

<4> 输入: \* C091 C091: 00 回车

<5> 输入: \* D000 <D000.FFFFM 回车

<6> 输入: \* C093 C093: 00 回车

<7> 修改中断向量: \* FFFA: 65 FF 65 FF

65 FF 回车

<8> 输入: \* C090: 00 回车

<9> 输入 "CTRL-C" 命令, 返回 BASIC 模式

<10> 启动加密盘: PR#6 回车

<11> 产生一个中断, 6502 会自动地把程序控制权转向 \$FF65 (即监控命令处理器)。

注: 本方法不适用于 CEC-I 电脑。

参考书籍:

《6502 微处理器及其应用》荣树熙等著

《APPLE II 监督程式剖视》郭重松著

《APPLE II PLUS 微型计算机系统》郑继志等著

《APPLE 磁盘操作系统详解》康蜀屿等著

## 又一个苹果机图形放大打印的

### 机器码程序

武汉市江大路江汉大学 刘红

本刊 90 年第 3 期所登《高分辨率图形放大打印》一文, 程序较简单。但正如文中所述, 由于纵横两方向放大倍率不同, 有使原图形失真的缺点。为此笔者也试编了一个与原图同样的图形放大打印的机器码程序。笔者是在 STAR NP-10 型打印机上运行通过的。如果所使用的是 MX-80、FX-80... 等打印机, 则需要把程序中 \$1FD0 单元开始的 "图形打印命令" 更改为 0A 0D 1B 4B 40 02 00 00, 同时将 \$1E56 单元的 07 改为 06 后, 才能运行。

使用本程序打印图形时, 只要把图形先绘制在高分辨率屏幕第一页或第二页上, 接通打印机电源 (不需要用 PR#1, PR#0 联机及断开)。在监控状态键入 \$1E00G,



屏幕上就显示提问：“PAGE1/2?”，键入页号1或2，打印机就开始打印，直到全图打印完毕，就可断开打印机电源。存在机器中的图形信息，并无影响。

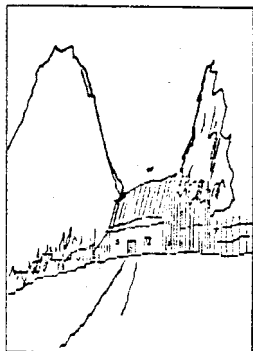
附机器码程序于后：

1E00.1FDF

```

1E00- 20 58 FC A9 B0 A0 1F 20
1E08- 3A DB 20 0C FD 20 ED FD
1E10- C9 B1 F0 0D 90 ED C9 B3
1E18- B0 E9 A9 40 85 E6 4C 25
1E20- 1E A9 20 85 E6 A0 00 A2
1E28- 00 8A 85 FE 20 11 F4 A6
1E30- FE A5 26 9D 00 70 A5 27
1E38- 9D 00 71 E8 E0 C0 D0 E9
1E40- A9 C0 85 4A A9 1F 85 4B
1E48- A9 06 85 4C 20 9A 1F A0
1E50- 00 A9 D0 85 4A A9 07 85
1E58- 4C 20 9A 1F A9 07 85 FB
1E60- A2 BF A9 00 9D 00 73 20
1E68- 79 1F 20 47 1F 5E 00 72
1E70- B0 0C 3E 00 73 3E 00 73
1E78- 3E 00 74 4C 89 1E 3E 00
1E90- 73 38 3E 00 73 38 3E 00
1E88- 74 DE 00 75 D0 08 E0 BF
1E90- D0 01 C8 20 79 1F 20 8E
1E98- 1F 20 8E 1F 20 8E 1F A9
1EA0- 00 9D 00 73 A5 FB C9 07
1EA8- F0 08 CA E0 FF 90 B8 4C
1EB0- B7 1E CA E0 FF 90 AB 20
1EB8- 9A 1F A2 BF 5E 00 74 3E
1EC0- 00 73 20 47 1F 5E 00 72
1EC8- B0 0C 3E 00 73 3E 00 74
1ED0- 3E 00 74 4C E1 1E 3E 00
1ED8- 73 38 3E 00 74 38 3E 00
1EE0- 74 DE 00 75 D0 08 E0 BF
1EE8- D0 01 C8 20 79 1F 20 8E
1EF0- 1F 20 8E 1F 20 8E 1F A9
1EF8- 00 9D 00 73 CA E0 FF 90
1F00- B8 20 9A 1F A2 BF 5E 00
1F08- 74 3E 00 73 5E 00 74 3E
1F10- 00 73 20 47 1F 20 8E 1F
1F18- 20 8E 1F 20 8E 1F A9 00
1F20- 9D 00 73 CA E0 FF 90 DE
1F28- C0 28 D0 0C A9 C8 85 4A
1F30- A9 04 85 4C 20 9A 1F 60
1F38- C6 FB F0 08 A2 BF 20 9A
1F40- 1F 4C 6A 1E 4C 59 1E A9
1F48- 01 85 FA 5E 00 72 B0 0C
1F50- 3E 00 73 3E 00 73 3E 00
1F58- 73 4C 67 1F 3E 00 73 38
1F60- 3E 00 73 38 3E 00 73 DE
1F68- 00 75 D0 08 E0 BF D0 01
1F70- C8 20 79 1F C6 FA 10 D3
1F78- 60 B0 00 70 85 06 B0 00
1F80- 71 85 07 B1 06 9D 00 72
1F88- A9 07 9D 00 75 60 AD C1
1F90- C1 30 FB 8D 00 73 8D 90
1F98- C0 60 84 FC A0 00 AD C1
1FA0- C1 30 FB B1 4A 8D 90 C0
1FAB- C8 C4 4C D0 F1 A4 FC 60
1FB0- 87 8D D0 C1 C7 C5 A8 B1
1FB8- AF B2 A9 BF A0 00 00 00
1FC0- 0A 0A 0D 1B 41 08 00 00
1FC8- 0A 0D 1B 40 00 00 00 00
1FD0- 0A 0D 1B 2A 05 40 02 00
1FDB- 00 00 00 00 00 00 00 00

```



## 用 LINE 语句在屏幕上作彩色图像

台州发电厂热工分场 夏克晃

用 BASIC 的 LINE 绘图语句在屏幕上作彩色图像，其原理是群所周知的，即将一幅图像看作是由无数的色点组成的。因此利用 FOR...NEXT 循环语句对屏幕进行扫描，在扫描过程中，每扫描一个点，即在该点处由 LINE 语句画上一个方块。这些小方块即是图像的色点，而小方块的色彩参数，是根据要画的图像输入的。这样，待屏幕扫描完毕，一幅由不同颜色的小方块组成的彩色图像就显示在屏幕上了。

输入图像每点的色彩参数时，可先在要画的图像上画上小方格，然后逐格输入色彩参数。小方格的多少，影响到屏幕图像的逼真程度，对于中分辨率  $320 \times 200$  的图形显示方式，画上  $160 \times 100$  个小方格，则屏幕图像就较细腻了。当然，对于图像中大块颜色相同的地方，可用 LINE 语句或其它 BASIC 语句直接画出，而无须逐格送色彩参数了。

显示如图所示图像的程序如下：

```

10 SCREEN 1, 0: COLOR 0, 0: CLS
20 FOR X=74 TO 240 STEP 2: FOR Y=0 TO 199 STEP 2:
READ J
30 LINE (X, Y) - (X+2, Y+2), J, BF
40 NEXT Y, X
50 LINE (240, 0) - (319, 199), 2, BF
60 A$=INKEY$: IF A$="" THEN 60
70 END
80 DATA.....

```

10 句置中分辨率图形方式 ( $320 \times 200$ )，并设置背景颜色和调色板。20 句，X 方向从 74 扫描到 240，Y 方向从 0 扫描到 199，步距为 2，并同时取色彩参数 J。30 句画纵横各为 2 的小方块，颜色为色彩参数 J 所设定。50 句画图像中颜色相同的大块红色方块。60 句使图像显示在屏幕上，敲任意键结束。80 句为 DATA 语句，后面的数据为输入的图像的色彩参数。

如想得到该图像的硬拷贝，也可先在屏幕上画出反相图像，然后用图形打印得到硬拷贝。



# 谈谈计算机病毒的防范措施

解放军 54685 部队 薛新文 陈夕坤

在我军指挥自动化迅速发展的今天，如何科学地管理、利用、保护计算机资源，有效地防范计算机病毒的危害，已成为我军搞好指挥自动化建设不容忽视的问题之一。最近，我们单位连续发生了多起由计算机病毒程序毁坏微机系统资源和信息数据库的事件，造成很大的损失，引起了我们对病毒程序危害性的重视。通过对 19 台微机进行普查，共发现了计算机病毒 7 种。在这些微机中，没有被病毒感染的微机仅占普查数的 30.6%，被感染一种病毒的微机占普查数 42.6% 以上，被感染二种以上病毒的微机多达普查数的 26.8%。同时我们又对系统备份盘进行了抽查，发现许多软盘也带有病毒。

要想诊治病毒，就得了解病毒。计算机病毒是一种程序，它能通过修改其它程序，把自身拷贝贴在其它程序之内进行繁殖和生存，从而完成对其它程序的传染。具有很强的传染性、针对性、破坏性、潜伏性和可触发性。我国从国外引进的计算机硬、软件较多，而平时的检“毒”、防“毒”措施还不够完善，病毒程序也随之而不请自来。另外，也有我国计算机爱好者自己研制的一些病毒，这些病毒严重威胁着我军计算机资源的安全。为此，我从以下四个方面谈谈计算机病毒的预防措施。

## 一、加强思想教育，治“毒”先治“本”

由于程序是设计者的代言人，如果程序设计者怀有恶意，那么他所设计的程序就可能对计算机系统构成严重的威胁。1988 年 11 月 2 日，美国计算机爱好者罗伯特·M·莫利斯研制的计算机病毒程序，对 MIL 网 / ARPA 网实施攻击，造成了严重的危害，引起世界上计算机安全专家的震惊。到目前为止，在我国发现的计算机病毒种类已多达七十种，这些病毒在计算机应用中悄无声息地泛滥蔓延，时刻威胁着计算机网络和各个数据库信息的安全。因此，我们应该从抓“人”入手。一是提高思想认识，对病毒的危害性引起足够的重视。我们既要充分认识病毒对微型计算机的破坏性，又要树立战胜病毒的信心。同时，还要作好必要的技术准备。一旦真正遇到病毒侵害时，也能够得心应手地进行处理，避免或尽量减轻病毒危害所造成的损失，而不致于因毫无思想准备而显得束手无策。二是加强职业道德教育，确立全心全意为人民服务的思想。建立优良的道德风尚，提倡严谨的科研作风，树立严肃认真的工作态度。做到不研制、不复制、不传播扩散病毒。发现病毒及时处理，使计算机病毒成为“过街老鼠，人人喊打”。

## 二、加强微机管理，严防“病”从“口”入

根据计算机系统的结构，我们知道微机共有三个“口”，即：键盘输入口、软盘驱动器存、取口和通信口。加强对这三个“口”的管理，是控制微机感染病毒的有效方

法。为此我们必须做到以下三点：

第一，加强对键盘输入“口”的管理。键盘是人—机交互的主要途径，我们通过加强对微机使用的管理，实现专人专机（或定人定机）、专机专用，做到每台机器都有专人负责。禁止外来人员上机操作，以防止通过键盘“口”建立病毒程序。

第二，加强对软盘驱动器的管理。通常计算机有 1~2 个软盘驱动器，有的还带 1~2 个硬盘。在计算机的信息交换过程中，软盘驱动器已成为一个重要的信息交换窗口。严格堵住从软盘驱动器输入“口”带病毒，以确保硬盘资源的安全。因此，每次进行拷盘操作之前，必须首先对有关盘片进行防病毒检测。只有在确认不带病毒的情况下，才能通过软盘驱动器进行拷盘操作。完善对微机使用的管理措施，以防止通过软盘驱动器将病毒程序拷入硬盘系统。

第三，加强对网络通信“口”的安全管理。在计算机网络传递信息过程中，微机是通过两个标准 232C 通信口输入输出信息。因此，如何堵住从此“口”传播病毒，是有效控制病毒在计算机网络中传播，确保计算机网络安全运行的前提。因此，要注意做好以下两个方面的工作：第一、终端上网前必须进行严格检查，发现带病毒的终端不能入网，确保网络安全。第二、定期对网络信息进行安全检查，发现病毒，及时进行清理，以保证微机系统资源的安全。

## 三、加强磁盘管理，防止病毒传播

在计算机技术被广泛应用的今天，各种功能齐全、性能稳定的优秀软件很多，推广普及也快。由于目前的检查、预防病毒的措施还不够完善，许多病毒隐藏于一些优秀软件中，借助于软磁盘而传播。因此，为了有效地控制病毒的扩散，彻底消除病毒的危害，切断病毒的传播途径，我们必须对磁盘载体制订严格的管理措施。

第一，应该加强备份盘的管理。定期做好计算机系统信息、网络信息和重要数据库信息的备份，及时更新备份盘版本，同时对备份的软盘进行防病毒感染处理，贴上写保护，以防备份盘感染、传播病毒。

第二，完善检“毒”、防“毒”措施。应定期对软盘和硬盘进行病毒检测，及时发现病毒源，并立即进行消毒处理，将病毒扼杀于萌芽状态，有效地控制病毒借助磁盘存贮介质在单位中蔓延。

第三，应加强当前工作盘的管理。为了有效地切断病毒程序的传播途径，平时应尽量减少微机间的磁盘复制操作，如确有必要，必须进行复制前的检“毒”、消“毒”工作，在确认没有感染病毒的情况下，方可进行复制移植工

# IBM 系列微机数据维护技术研究

广州市计算机应用研究所 胡向东

随着计算机在财务、机要等部门的进一步应用,随着越来越多的资料贮存在计算机上,随着计算机病毒的流行及花样更新,数据维护技术正越来越受到广大计算机工作者的重视。

造成计算机内的数据读不出来的情况有四大类,第一大类是计算机硬件出了故障,第二大类是系统启动部分出故障,第三大类是文件管理部分出了故障,第四大类是数据实质部分出故障。

下面从这四大类故障谈谈笔者对 IBM 系列微机数据维护技术的一点心得体会,以便抛砖引玉。

## 一、硬件维护

### (1) 属于接触不良的故障

IBM PC 系列微机中最常见的故障是这种类型的故障,由于主机采用插件式结构,因此卡与扩充槽之间,IC 与 IC 插座之间,插头与插座之间常因震动或氧化等原因造成接触不良。

例一,某单位 IBM PC/XT 开机启动不了,笔者用手按紧主机板上的 8284A 时钟发生器后,主机正常启动。

例二,某机不能显示,关机后,压紧显示卡,再开机,显示器正常显示。

例三,某机不显示,关机后,把主机板上的 RAM 压紧,显示器正常显示。

例四,某机不能正常启动,后发现因天长日久,主机内电源输出线接头部分氧化,造成接触不良,后用小刀把

导线接头部分刮新后装上,一切正常。

例五,某硬盘出现 1701 错误,后查出是硬盘电路板上某接头虚焊,造成接触不良,重新焊接后,硬盘恢复正常。

### (2) 属于磁头脏的故障

通常是用清洁盘滴上洗磁头水清洗,但如果仍不见效,则应用小竹枝捆上棉花,沾上洗磁头水或酒精,再用一块小玻璃镜照着来清洗磁头,便可使磁头干净。

### (3) 属于锂电池没电的故障

AT 机主机板上的 CMOS 需要锂电池供电。某 AT 机每次开机都要 SETUP 一次参数才能启动,后更换锂电池后一切正常。

### (4) 属于元器件实质性的故障

此类故障可用替代法或用逻辑笔,万用表等测出,但需要一定的电子线路知识,此处略。

## 二、系统启动部分的恢复

### (1) 硬盘主引导扇区(在硬盘 0 道 0 头 1 区)的恢复

硬盘主引导扇区包括硬盘的主引导程序和硬盘分区信息表二部分,根据笔者对大量的 IBM PC/XT、AT、386 及其兼容机的硬盘主引导程序的调查,这些机型的硬盘主引导程序基本上是相同的,是可以替代的。而硬盘分区信息表又分为二大类,第一种类型硬盘的逻辑 0 扇区定义在 0 道 1 头 1 区,该分区参数表前 4 个字节是 80010100,第二种类型硬盘的逻辑 0 扇区定义在 0 道 0 头 2 区,该分区

作。如果一时没有来得及检查的,在移植后,也应坚持对硬盘和复制盘进行检查,采取补救措施,以防因一时疏忽,造成不可估量的损失。

第四,建立完整的技术档案。为了能及时发现病毒,有效地管理好信息备份盘和硬盘资源,我们应给硬盘和重要的数据信息备份盘建立技术档案,登记有关的参数和各项指标。如:磁盘所剩余的空间、盘上所建的文件个数、长度、运行的速度、记盘时间、系统的性能以及经常发生的故障等等。经常性地对照档案检查,可以及时发现磁盘是否感染病毒,以便及早进行处理。

## 四、加强培训“医生”,及时医“病”消“毒”

有病就得有医生,这是自然界的对立统一规律。对付病毒程序的危害,仅仅采取以上的防范措施还不够,一旦微机感染上病毒后,就必须要有能消除病毒的技术人员。因此,为了能及时给微机医“病”消“毒”,加强微机工作人员的业务培训,提高计算机工作者的业务水平和业务素质是十分必要的。

第一,注重人才的培训。我们应定期在思想品德好的计算机工作者中选拔一些业务水平高、工作能力强、能吃苦、肯钻研的技术人员进行重点培训,使他们及时掌握各种病毒的原理、发作过程及消除方法。同时对新出现的病毒进行研究分析,及时了解当前病毒的动态信息。然后再回到工作第一线,对微机感染的“病毒”进行诊治,确保生“病”的计算机能及时得到医治。

第二,注重抗病毒软件的研制工作。根据目前的实际情况,要想使每个从事计算机工作的人员都能熟练地编制消“毒”软件,是不现实的。因此,为了保证计算机系统的正常安全运行,我们应抓紧研制开发各种能预防病毒、消除病毒危害的应用软件,并及时推广。普及抗病毒程序的知识,提高计算机操作人员的反病毒能力,达到人人都能利用手中现有的消“毒”软件来清除病毒的目的。

总之,计算机病毒虽然危害性很大,但只要掌握了它的基本原理,同时,在思想上重视它,在工作中防范它,病毒的危害也是完全可以避免的。■

参数表前 4 个字节是 80000200。

而对于每一种类型的硬盘分区参数，只因硬盘的容量不同而不同。由于笔者平日注意收集各种不同容量的硬盘分区参数，因此对于恢复硬盘的分区信息表大有帮助。

例一，大麻、广大一号、六·四病毒及打印病毒都会破坏硬盘的主引导程序，笔者编写了解毒程序，用任一 IBM PC/XT 硬盘中的主引导程序去覆盖这些病毒（但保留原机的硬盘分区参数），使感染上病毒的 IBM PC/XT、AT、386 及其兼容机的硬盘恢复正常。（可参考 90 年第一期《电脑》杂志胡向东所写《硬化病毒的分析及排除方法》）

例二，某单位机房突然停电，操作员关主机，关总闸，但没有关 UPS，恢复供电后，操作员开总闸，开主机，造成 UPS 供电的电压不正常，影响硬盘分区信息表的参数，使硬盘不能正常启动。笔者把硬盘的主引导扇区读入内存，然后根据操作员提供的硬盘容量信息仿制了一个硬盘分区信息表，然后写回到硬盘 0 道 0 头 1 区，重新启动后，硬盘恢复正常。（注意，硬盘分区参数表以 55AA 作结束标志）

下面列出的硬盘分区参数，可供参考：

第一类型 10 兆硬盘分区参数：

80 活动分区标识

010100 本分区首扇区所在磁头号、扇区号、磁道号

01 操作系统标志（01 和 04 是 PC DOS，MSDOS；02 是 XENIX 等）

035130 本分区末扇区所在磁头号、扇区号、磁道号

11000000 本分区前已使用的扇区数

F3500000 本分区扇区数

第二类型 10 兆硬盘分区参数：

80 000200 01035130 01000000 03510000

第一类型 20 兆硬盘分区参数：

80 010100 04039165 11000000 07A30000

第一类型 20 兆硬盘，内有 DOS 分区和 XENIX 分区的参数：

80 010100 01004133 11000000 7B510000 （DOS 分区为活动分区）

00 0041340 2038167 D0510000 7B510000 （XENIX 分区为非活动分区）

第一类型 30 兆硬盘分区参数：

80 0101000 4059181 11000000 BBFF0000

(2) 硬盘或软盘逻辑 0 扇区的恢复

逻辑 0 扇区由磁盘 I/O 参数表、磁盘基址表和引导程序这三部分组成。如果逻辑 0 扇区是被病毒所占领，一般可根据该病毒的类型而得知原引导程序藏身之地，然后用 DEBUG 的 L 命令调入内存，再写回到逻辑 0 扇区。

例如在 360K 软盘被打印病毒感染后，原软盘引导程序被藏到 2CEH 逻辑扇区中，我们便可在 DEBUG 中执行：

L 0100 0 2CE 1

W 0100 0 0 1

从而恢复原逻辑 0 扇区。

有时，我们也可以相同 DOS 版本的 SYS 命令来恢复逻辑 0 扇区。

(3) 克服不同版本 DOS 的干扰

有时我们在 DOS3.0 的硬盘中拷入 DOS2.0 的 COMMAND.COM，造成 DOS 启动的不正常，此时我们要用相同 DOS 版本的命令文件 COMMAND.COM 拷回硬盘。

### 三、文件管理部分的维护

要进行这方面的工作，首先要了解贮存文件的软盘和硬盘中，簇号与逻辑扇区号的关系公式。

通常可用以下三种方法来获得该转换公式：

方法一：查 IBM PC/XT (AT) 各类磁盘的基本输入输出参数表，若设簇号为 t，逻辑扇区号为 s，则

360K 软盘中， $s = 2t + 8$

10 兆硬盘中， $s = 8t + 33$

增强型 20 兆硬盘中， $s = 16t + 49$

AT20 兆硬盘中， $s = 4t + 107$

AT32 兆硬盘中， $s = 4t + 153$

方法二：利用逻辑 0 扇区 I/O 参数表提供的每簇扇区数（扇区 0 字节偏移量 0D 处）、保留扇区数（偏移 0E ~ 0FH 处）、每个 FAT 占扇区数（偏移 16 ~ 17H 处）、FAT 数目（偏移 10H 处）、根目录登记项数（偏移 11 ~ 12H 处），按如下公式计算：

$$s = (t - 2) \times \text{每簇扇区数} + \text{保留扇区数} + \text{每个 FAT 占扇区数} \times \text{FAT 表数目} + \text{INT}[(32 \times \text{根目录登记项数} + 511) + 512]$$

方法三：利用簇号与逻辑扇区号的线性关系，设关系式为  $s = at + b$ ；利用 DEBUG 分别把该磁盘中的两个文件调入内存，然后记住每个文件的 8 个连续字节，然后用 PCTOOLS 的 F 查找命令分别查找这两个文件的 8 个字节。PCTOOLS 在每查到一个文件时，都会把该文件所在逻辑扇区号 s 和簇号 t 显示出来，从而可得到方程组：

$$\begin{cases} s_1 = at_1 + b \\ s_2 = at_2 + b \end{cases}$$

从而解出 a 与 b，得出关系式  $s = at + b$

这种方法对于求出一些较特殊的硬盘的关系公式很有效。

掌握了这个转换公式，我们便可以从三个角度研究文件的恢复。

(1) 利用恢复目录来恢复文件

磁盘根目录的首扇区可以从逻辑 0 扇区的 I/O 参数

表中算出:

根目录首扇区逻辑扇区号 =  $1 + \text{FAT 数目} \times \text{每个 FAT 占扇区数}$

而子目录是类似文件形式存在父目录中的。

例一, 某文件 A1.PRG 长度不超过 1 簇, 被 DOS 的 DEL 命令误删除后, 该文件的 FAT 链被清 0, 且该文件的目录项第一个字节变成 E5H。我们用 DEBUG 的 L 命令把该目录项调入内存, 用文件名第一个字母“A”的 ASCII 字符 41H 代替删除标志 E5H, 然后用 W 命令写回磁盘。接着我们用 COPY 命令把修复过的 A1.PRG 拷贝到另一个磁盘上, 则另一个磁盘上所得到的 A1.PRG, 由于原长度未超过一簇而得以全部恢复。

例二, 某文件的目录项被病毒影响覆盖, 在 DIR 命令中已看不到该文件, 但该文件的 FAT 链未被破坏, 我们由原始资料找出该文件的几个特征汉字, 然后用 DEBUG 找出这几特征汉字的机器码, 再用 PCTOOLS 的 F 查找命令找到该文件的起始簇号, 接着用 DEBUG 命令把该磁盘的根目录调入内存, 再按照相应的文件名、起始簇号、文件长度 (可略大于原长度) 仿造出一个目录项, 再写入磁盘, 然后用 COPY 命令把该文件拷贝到另一个磁盘, 则在另一个磁盘上的文件已是一个完整的文件。

例三, 某单位硬盘的根目录中有一个扇区被病毒影响而覆盖, 使该硬盘上一个重要的子目录名所对应的目录项丢失, 从而不能进入该子目录。笔者利用该子目录使用者提供的文件名, 利用 PCTOOLS 的 F 命令查找到该子目录所在逻辑扇区, 然后用 DEBUG 的 L 命令把根目录和该子目录分别调入内存的不同区域 (不妨为 CS: 100 和 CS: 2000), 然后用 DEBUG 的 M 命令把该子目录上的文件目录项移到根目录文件目录项最后一个文件的后面, 然后把修改后的根目录写入硬盘, 即可在根目录上用 DIR 命令看到原子目录上的文件, 避免了子目录上的文件的丢失。

笔者在另一个单位遇到过同样的问题, 另一种解决办法是在用 PCTOOLS 的 F 命令找到该子目录项所在簇号后, 在根目录中仿制出一个子目录项, 子目录名由用户提供, 属性取 10H, 起始簇号取 PCTOOLS 所找到的值, 再把修改后的根目录写入硬盘, 即可恢复子目录上的文件。

注意, 子目录若超过一簇时, 用后一种方法可避免子目录所在扇区不连续的麻烦。

## (2) 利用恢复 FAT 链来恢复文件

DOS 是利用 FAT 链来记录每个文件在磁盘中所分配的位置的。对盘上的每一个簇号, 有一个占 12 位 (1.5 字节) 的登记项 (对 DOS2.X 和 DOS1.X 而言), 在 DOS3.X 中, 每一簇在表中可占 12 位或 16 位, 其值与盘的容量有关。FAT 正本首扇区号 = 1, FAT 副本首扇区号 =  $1 + \text{每个 FAT 占扇区数}$ , 我们往往可以用修复 FAT 链的方法恢复文件。

例一, 某单位受六·四病毒影响, 硬盘中 FAT 区中

逻辑 4 扇区受病毒影响 (可参考 90 年第 6 期《电脑》杂志胡向东所写《大麻、广大一号、六·四病毒及打印病毒》一文) 而覆盖, 造成该硬盘上一个重要的数据库 515 条记录只看到 22 条记录, 由于病毒只覆盖了 FAT 正本, 而问题发现得及时, 故笔者用 DEBUG 的 L 命令把该硬盘的 FAT 副本调入内存, 然后用 DEBUG 的 W 命令把内存中的 FAT 副本写入正本所在扇区, 从而使硬盘恢复正常, 该数据库 515 条记录全部保存。

该例子说明, 受大麻、广大一号和六·四病毒覆盖了 FAT 区正本中的一个扇区, 只要及时把副本写回正本处, 就可挽救硬盘中的文件, 但时间拖得太长, 经多次写盘, 副本中的部分 FAT 链也会失真。

例二, 某一 25 页长的文书文件所在软盘受某种原因影响, 只能看到 3 页, 其余的内容用 WORDSTAR 调不出来。笔者根据原始资料, 记录了断点处的上面一句话和下面一句话, 用 DEBUG 命令找出几个特征汉字的机器码, 然后用 PCTOOLS 的 F 命令分别查出这两句话在磁盘中的对应存储簇号, 然后用 DEBUG 的 L 命令把磁盘 FAT 区调入内存, 然后修改 FAT 表, 使上一句话所在簇号的链指针指向下一句话所在簇号, 然后把改后的 FAT 表写盘 (注意要把操作过程记录, 若失败时也可还原), 从而使上下文件能连接起来。照此办法处理断点, 便可将整个文件恢复。

## (3) 直接对文件所在扇区读写来恢复文件

例一, 某单位由于突然停电的原因, 使某一库文件的 1AH 结束标志丢失而不能对该库文件进行正常操作, 笔者利用用户提供的原始资料, 用 PCTOOLS 的 F 命令找到该库文件末尾所在逻辑扇区, 并用 DEBUG 命令把该扇区调入内存, 然后在文件尾用 E 命令写上 1AH 结束标志, 再把修改后的内容写回磁盘, 便恢复了该库文件的正常操作。

例二, 某单位由于进行了 FORMAT C: 的误操作, 使硬盘的根目录区中的每一个文件都作了删除标志 E5H, 同时 FAT 区全部清 0, 为了挽救硬盘中的一个有几千条记录的数据库, 笔者利用用户提供的原始资料, 用 PCTOOLS 的 F 命令, 首先把该数据库的文件头 (包括文件说明和栏变量说明) 单独组成一个文件写到软盘, 然后交替用 PCTOOLS 和 DEBUG 查找并把库中内容调入内存并以不同的文件名写入软盘, 完成第二步工作。

第三步工作是按照 DBF 文件的格式 (可参考 88 年第 2 期《微计算机应用》胡向东所写《实现 dBASE III 的 DBF 文件与 SUPERCALC3 的 CAL 文件互相转换》一文), 用 DEBUG 命令把文件头和这些软盘上的小文件拼接起来, 然后在文件末尾加上 1AH 结束标志, 再以 .DBF 为扩展名写回到软盘中, 形成一个可在 dBASE III 环境下可以调用的 DBF 文件。

第四步工作是进入 dBASE III 的 “.” 提示符状态, 利

## SENDON UPS-600 型

## 维修一例

柳州市电子计算机软件公司 石成钢

故障现象：市电断电时，UPS 有时能维持计算机继续工作，有时不能。

分析与维修：

UPS 所接计算机为长城 CH-II，其工作电压为 190~240V。

经测量发现，在市电断电前，若 UPS 输出电压大于 210V，则断电后 UPS 能维持计算机工作；若 UPS 输出电压为 200~210V，则断电后不能维持计算机工作。

再测量市电，其电压为 230~240V，在 235V 附近波动。通过分析 UPS 的输入—输出电压曲线可知，在市电电压（即 UPS 输入电压）低于 240V 时，UPS 输出电压大致等于输入电压，即在 230~240V 之间，这时若市电突然断电，瞬间 UPS 输出电压降到 215V，然后回到 225V，停电后计算机工作正常。

若在某时刻市电超过或等于 240V，这时 UPS 内的继电器动作（发出卡嗒一声响），输出电压为 200V，这样可以保护计算机。如果这时停电，则 UPS 输出电压降到 185V，然后回到 220V，但停电后计算机停止工作，问题的症结就在这里。

排除故障的方法是，在 UPS 之前接上交流稳压器，调整交流稳压器的输出为 225V，这样，UPS 和计算机均工作正常。■

## MS-DOS 操作系统结构分析系列教材

## 《命令级结构彻底剖析》简介

本刊自 89 年第三期以来，以连载的方式系统、全面地对 MS-DOS COM MAND 模块进行了彻底的剖析，深受读者的欢迎。杂志社和作者都收到不少读者的来信，给予十分热情的支持和鼓励，并提出许多建设性的建议。但由于篇幅所限和期刊的特点，本刊只能发表多篇相对独立而整体上又有关联的文章，不能将程序注释清单刊出，使读者在阅读文章时产生一定的困难。为此，本刊为满足广大读者的要求，在连载和作者为本科高年级学生编写的教学讲义的基础上，进行了全面的修改和补充，提供完整的程序注释清单，并编成书，以专辑形式出版，以便于读者作为资料备查。

本书可作为大专院校有关专业的教材和教学参考书，也可作为广大计算机工程技术人员在职培训的教材，它是从事微计算机系统和应用开发的工程技术人员必备的参考资料。

本书作为 MS-DOS 操作系统结构分析的系列教材的第一部，由中山大学郭嵩山老师编著。现已出版，每本零售价：15 元，另加邮杂费 1 元（含挂号费）。

俱乐部会员购书注明会员号可获 8 折优惠；购书 10 本以上 9 折优惠，50 本以上 8.5 折优惠，100 本以上 7.5 折优惠（未含邮费）。

## 欢迎邮购

汇款地址：广州市天河 1753 号邮政信箱 萨成熙收。  
填写汇单请用正楷书写，以免误投。■

用 DBASE III 的 APPEND FROM 命令，把软盘上的一个 DBF 文件并接到硬盘上，形成一个大的 DBF 文件。

第五步工作利用 DBASE III 的 COPY 命令把该 DBF 文件复制到硬盘上另一个不同名的 DBF 文件，经过拷贝新的 DBF 文件中，记录个数在说明部分已得以更正。

第六步工作是对照原始资料，对新的 DBF 文件进行修补工作。

笔者用该办法进行过三例的文件恢复，恢复率均在 98% 以上。

## 四、数据实质部分出故障和处理

数据实质部分出故障，只能按情节轻重来处理。

例一，某单位原来用一台机器录入人事资料，后时间紧迫，又动用了另一台机器，结果有一台机器上的数据库有 8000 多条记录，另一台机器上有 2000 多条记录，由于使用的软件是上面统一发放下来的，两台机器所用的数据库名字相同，由于操作员使用 COPY 命令不慎，用 2000 多条记录的库文件覆盖了 8000 多条记录的库文件，使辛苦

了一个多月录入的一万多条记录只能看到 2000 多条记录，加上平日没有备份，上级要资料的限期又快到，情况紧急。笔者利用上述提到对文件所在逻辑扇区直接读写的方法挽救了 8000 多条记录中的后面 6000 多条记录，而该库前面的 2000 多条记录已被另一库文件覆盖，数据实质部分出了故障，已不可能挽救。

例二，某单位送一台主机去维修，顺便要求在硬盘中装配 2.13F 汉字系统，维修人员未问清硬盘中的资料有无备份，便对硬盘低级格式化，笔者前往观察，硬盘中所有资料的实质部分已被清除，无可挽救。

总之，数据维护技术的确是值得人们探讨的一门学问，但是进行这一工作又是一个有风险的工作，一定要把勇于探索与严谨的工作态度结合，万万不能大胆有余而谨慎不足，造成更大的损失。一定要注意对现场的保护及记录，一定要做好备份，做好操作记录，使之有可还原性，然后对症下药，一步一个脚印地做好数据维护工作，才能使自己有所贡献。■

# EXE 文件与 COM 文件的比较

南昌陆军学院计算机室 葛伟华

EXE 文件和 COM 文件同是在 MS-DOS (或 PC-DOS) 环境下的两种二进制代码的可执行文件, 它们在内部结构、存储方式和文件加载装入内存的过程中存在一些差别, 但是在满足一定的条件下, EXE 文件可以转换成 COM 文件。掌握这两种文件的特点, 对于我们编写汇编语言程序会有很大帮助。

## 一、内部结构特征

### 1.COM 文件的内部结构

COM 文件存储在磁盘上时是内存的完全映像, 其文件扩展名为 .COM, 它不包含重新定位的加载信息, 与 EXE 文件相比结构更加紧凑简单, 加载过程和执行速度更快。COM 文件只有一个代码段, 整个文件的长度即是程序本身的长度, 不超过 64KB-PSP 表长-2, 程序可用堆栈范围不超过 64KB-PSP 表长-可执行代码长度, COM 类应用程序可由几个分离的目标模块连接而成, 但是所有的目标模块必须具有相同的代码段名和类别名, 而且数据段、代码段及堆栈段合并成为一个段, 所有子程序都应 NEAR 属性, COM 程序必须在偏移 0100H 处用 ORG 伪指令设置起始地址段。

COM 文件在结构上也存在一定缺陷, 在多用户 DOS 环境下, 不便于生成一个“纯粹”的代码区供几个任务共享, 而 EXE 型文件就从结构上弥补了这个缺陷。

### 2.EXE 文件的内部结构

由 MS-LINK 产生的 EXE 文件以一种特殊形式存于磁盘上, 文件扩展名为 .EXE, 它由两部分组成:

① 文件标题: 包括格式化区和再定位表;

② 装入模块: 包括装入字区和可执行代码段;

两部分的关系见附图 1

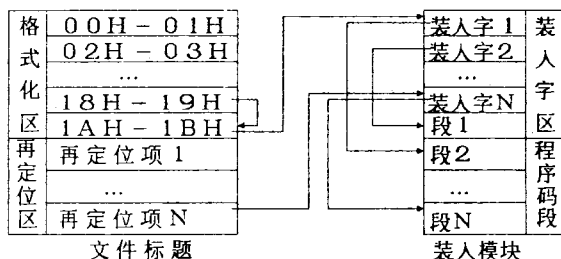


图 1 EXE 型文件结构图

格式化区共有 28 个字节, 每一项包含两个字节, 各项的具体含义请参见附表 1。再定位表紧跟在格式化区之后, 表项的个数等于程序代码段的段数, 它与装入字区的装入字是一一对应的。每个再定位项包含四个字节以高/低字节的形式表示内存的段地址和偏移量。在再定位过程中再定位项所指示的装入字, 在 EXE 文件得到 CPU 控制权

之前需要进行修改, 修改后的装入字内容为其对应程序段的偏移量, 它决定了各程序段在运行时的段值。因此, 改变了装入字就可以实现对程序模块的再定位。

附表 1: 文件标题格式化区结构表

偏移值	说 明
00H~01H	EXE 文件标志, 规定值 4DH, 5AH
02H~03H	EXE 文件最后扇区所包含的字节数 (文件长度 MOD 512)
04H~05H	EXE 文件长度 (包括文件标题), 以 512 字节为单位
06H~07H	再定位表中包括的再定位项数
08H~09H	文件标题长度, 以 16 字节为单位, 用于决定装入模块起始位置
0AH~0BH	装入模块结束端所需最小分配字节数, 以 16 字节为单位
0CH~0DH	装入模块结束端所期望最大分配字节数, 以 16 字节为单位如 0AH~0DH 都为 0, 则程序尽可能被装入高地址处
0EH~0FH	在程序执行之前要装入堆栈段的段偏移值, 此值必须通过再定位来调整
10H~11H	在程序执行之前装入 SP 的值
12H~13H	文件中所有字的负数和
14H~15H	在程序执行之前装入 IP 的值
16H~17H	在程序执行之前装入 CS 的值, 必须由再定位调整
18H~19H	再定位表中第一个再定位项的偏移值
1AH~1BH	由 MS-LINK 产生的覆盖号, 驻留程序为 0000H

EXE 型文件可以由几个目标模块连接而成, 每一模块可使用自己独有的代码段名, 各过程的属性可为 NEAR 或 FAR。生成 EXE 型文件的众多目标模块只能有一个模块带有 STACK 属性段, 并且只有一个程序入口, 它可以位于代码段的任何地址处, 由主模块的伪指令 END 给定。

## 二、内存加载过程说明

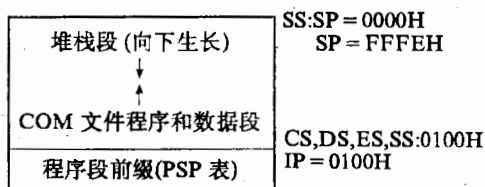
### 1.COM 型文件加载

MS-DOS 在加载 COM 型文件时, 不像 EXE 型文件那么复杂, DOS 不需判别其是否确实为可执行代码, 只是



将其加载到 PSP 表之后, 因此, 在偏移量 0100H 处必须存在一条可执行指令。同时, DOS 将暂存程序区全部分配给 COM 型文件, 当 CPU 控制权由 DOS 传递给 COM 文件时, 所有段寄存器均指向 PSP 表。所以, 当一个 COM 型文件试图通过 EXEC 功能调用另一子过程时, 它必须将自身所占的存储资源压缩到最小程度, 并且保护好堆栈。COM 型文件加载后内存的映象见附图 2。

COM 型文件运行结束后, 可有几种返回 DOS 的方法 (参见 EXEC 功能返回 DOS 方式), 最好使用 INT21H 的 4CH 号功能将返回代码传递给调用它的程序。



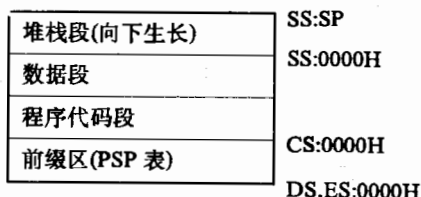
附图 2 典型 COM 文件加载到内存的映象

## 2.EXE 型文件加载

EXE 型文件加载与 COM 型文件加载所利用的机制相同, 虽然 DOS 加载 EXE 文件时数据段, 代码段和堆栈段的顺序有所变化, 但是它们仍是直接加载在 PSP 表后, EXE 型文件加载后的内存映象见附图 3。

EXE 型文件加载过程与 COM 型文件的加载过程相比, 最大差别在于 EXE 文件的再定位过程, 这是由 MS-DOS 的命令解释程序 COMMAND.COM 内提供的再定位装入程序功能完成的, 整个再定位装入过程如下:

- ①在可分配内存的最低地址处建立一个 PSP 表;
- ②将文件标题的格式化区读入内存;
- ③根据装入模块的长度 (文件长度-文件标题长度) 和分配数 (0AH~0DH) 分配内存区段;
- ④根据偏移地址 02H~03H 的内容向下调整文件实际长度;
- ⑤根据高/低装入程序开关的设置, 确定起始段;
- ⑥把装入模块读入以起始段开始的内存中;
- ⑦将文件标题的再定位表读入工作区;
- ⑧把再定位表中每一项的段值和偏移值加到起始段值中, 修改装入模块相应的装入字;
- ⑨根据文件标题的格式化区中相应的值和起始段值, 修改寄存器 SS 和 SP, 修改寄存器 CS 和 IP, 设置寄存器 ES 和 DS。



附图 3 典型 EXE 文件加载后内存映象

一个 EXE 型文件执行结束后, 最好是通过 INT21H 功能将控制返回 DOS, 如使用别的方法通常需将 CS 寄存器指向程序段前缀区 (PSP 表), 这样就可能造成与将来的 DOS 版本不兼容。

## 三、EXE 文件的转化

MS-DOS 提供一个外部命令 EXE2BIN, 可以把没有分段固定的 EXE 型文件转换成和 COM 型文件或其兼容形式, 这样使得文件结构更加紧凑, 调用和执行速度也加快了。但是 EXE 型文件转换成 COM 型文件的格式必须具备下列四项条件:

- ①必须是由 MS-LINK 产生的有效 EXE 格式。
- ②文件驻留在内存的部分, 即实际代码和数据必须小于 64KB。
- ③不能有堆栈 (STACK) 段。对于已有堆栈段的 EXE 文件, 只要将其格式化区的偏移值为 0EH-0FH 的内容改为 0 值, 也可以用 EXE2BIN 进行转换。
- ④程序段必须定义 0100H 的预留区域, 并且加载的首地址须经伪指令 END 语句说明为起始地址。

根据指定的 CS: IP 初始值, EXE 型文件存在两种转换可能:

- ①如果程序中没有指定 CS: IP 的值 (EXE 型文件包含 0000: 0000H, 那么假定为纯二进制码转换, 即不是 COM 型文件, 而是其兼容格式。

在这样情况下, 只有当装入由用户应用程序指定的绝对内存地址时, 结果程序才能使用。

- ②如果把 CS: IP 指定为 0000: 0100H, 则假定该文件当作 COM 型文件运行, 而且用伪指令 ORG 语句把地址指针 IP 置于 0100H, 文件前 100H 字节内容被删除, 这样就可以转换成 COM 文件。

假若 CS: IP 不符合以上准则之一, 或者即使符合, 但存在程序段再定位, 则系统将提示如下信息:

File cannot be converted

总之, EXE 型文件与 COM 型文件之间是存在微妙的差别, 它们各有利弊应用于不同的场合。特别是 EXE 型文件的再定位特性使得 DOS 在内存中安装程序具有更大的灵活性, 与此同时, 还保证了在 DOS 向虚存扩展时保留程序可移值性。因此, 最新的趋向是保留文件的 EXE 格式。

## 参考文献

- 1.《DOS 使用大全》(美) Kris Jamsa
- 2.《IBM DOS 操作系统》谢德煌, 高惟龙
- 3.《DOS 高级程序员使用指南》(美) Ray Duncan
- 4.《磁盘文件管理与加密原理》沈永耀陶铮正

# 具有多级放大功能的屏幕图形 拷贝程序的设计

湖北省水利学校计算机室 吴家富

## 一、引言

目前的各种汉字系统,都具有用针式打印机对屏幕图形进行硬拷贝的能力,而且也出现了许多专门的硬拷贝软件。但是,它们或是针对某一类显示系统而设计的,或选用特定的打印机才能实现。它们所拷贝的图形,或太小、或大小固定,或者变化不够丰富,很难满足对图形拷贝质量要求较高的应用系统的需求。

可望满足各种复杂情况要求的硬拷贝程序,需要解决如下三个问题:

- 1、能对各种流行的显示系统实现正确的硬拷贝,即自适应显示系统。
- 2、能在各种流行的针式打印机上正常实现,即自适应打印机。
- 3、纵、横放大倍数均要可调,且互不影响,因此,也就有多种组合关系。

用得较多的 IBM PC/XT/286/386 及各式兼容机上所配备的显示系统,种类很多。从低档的单色显示系统至高档的彩色 VGA 系统,应有尽有。对于同一单位来说,往往也拥有多种规格的显示系统。

为了使汉字输出美观、快速,许多单位都配置 24 针打印机。

目前,最为流行的 24 针打印机,种类也很多,相互之间多不兼容。

用针式打印机对屏幕图形实现硬拷贝,其实质,就是将 VRAM 中适合于显示要求的横向数据,转换成适合打印机要求的列向数据。

由于各种显示系统的 VRAM 结构不同,与屏幕的对应关系也不同,因此,转换算法也不相同。

这些复杂情况,给研制通用模块,造成了很大的困难。笔者通过大量实践,在此方面作了一些探索,试图在差异之中找出同一性,以实现用一个模块去控制(处理)诸多的硬件系统(显示系统与打印系统)。下面简述主要思想。

## 二、程序设计思想及有关算法

从上述简要分析可知,显示、打印系统是很复杂的,任何针对单一硬件系统而设计的汉字操作系统或硬拷贝程序,都不能在其它类硬件系统上良好运行。

由于大多数硬拷贝程序,都是伴随着一定的汉字操作系统,因此与显示系统的适配问题,应由汉字系统中负责管理显示部分的 INT10H 去完成。我们只用它所返回的有关显示系统的有关参数,分情况处理即可,亦即自适应于

各种显示系统。WHDOS3.3C 汉字系统已做到了这一点,实现了多种显示系统,共用一个显示模块的功能。文中所附 SCRCOPY.ASM 就是基于这个思想而设计的。

对于与打印机的适配问题,应由汉字操作系统中负责管理打印输出部分的 INT17H 去完成。硬拷贝程序只管调用相应位置的指令序列,而不管使用何种打印机。CCDOS2.13F/WHDOS3.3C 汉字系统都成功地解决了多种流行的 24 针打印机共享一个打印模块的问题。

基于上述思想,在编制硬拷贝程序时,主要的问题就是拷贝算法分析了。

### 1、与显示系统的适配

利用新一代汉字系统中提供的 INT10H 之 OFH 号功能,取得当时所用显示系统的有关参数,对各种显示系统分别予以处理。

### 2、与打印机的适配

由新一代汉字系统中提供的 INT17H 去完成。

### 3、打印缓冲区的设置

由于进行屏幕图形硬拷贝时,系统不会进行常规的汉字输出,因此,可与 INT17H 共享打印缓冲区。

### 4、硬拷贝程序的设置

取代原 INT05H,实现图形拷贝。

### 5、列向数据的获得

VRAM 中的内容,为适合显示用的横向数据,每字节高位对应于屏幕左边点,而打印机需要将这些显示数据从左→右按列转换成列向数据。

对于 24 针打印机,在正常情况下(不放大),每一列向数据,可用屏幕上的 24 根扫描线的相应点组合而成。如果要将图形放大 N 倍,则每次所取水平扫描线的数目应减少 N 倍。

具体转换算法,参见附录 ZH-8-BT 子程序。

### 6、一次扫描能处理的扫描线数目的确定。

我们称每转换 24 点列向数据(相应水平扫描线全部被处理)为一次扫描。

设 LINE\_1—进行一次扫描每区应取线数

Y\_KUO—纵向放大倍数(1~3)

CRT\_QU—VRAM 分区数(参见 7(2)表)

则有:

$$LINE\_1 = \frac{24}{Y\_KUO} = \frac{24}{CRT\_QU \cdot Y\_KUO}$$

## 7. 纵、横放大倍数的确定

### ① 横向放大倍数的确定

每条线宽 640 点, 相当于 640 个列向数据, 如果要放大  $N$  倍, 则转换成列向数据量为  $640N$ ; 又因为 24 针打印机最小行宽为 2176 列, 因此, 作为通用模块, 每次处理的数据量不能超过此限, 即  $640N < 2176$ 。

采用整数放大, 可使转换算法简化。因此, 据  $640N < 2176$  可得  $N = \lfloor 2176 / 640 \rfloor = 3$ , 即横向放大倍数可取 1~3。

### ② 纵向放大倍数的确定

为了简化算法, 设定每次转换时, 每区都同时处理相同数目的扫描线。因此, 可用每区一次扫描应处理的扫描线数作为控制数据。

为了求得通用纵向放大倍数, 对各种显示方式下, 用公式  $LINE - 1 = \frac{24}{CRT - QU \cdot Y - KUO}$  计算出一次扫描每区应转换的扫描线数如下表所示。

由上表可知, 纵向放大倍数只能取 1~3

### 8. VRAM 偏移量的确定

设  $ADD-SI-N$ —每扫描一次, VRAM 偏移量则  $ADD-SI-N = 80 \cdot LINE - 1$

### 9. 横向扩展后, 打印机应接收的列向数据量

设  $PRINT-NIESU$  —横扩后, 折成打印机列向数据量

$X-KUO$  —横向放大倍数 (1~3)

则:  $PRINT-NIESU = X-KUO \cdot 640$

10: 屏幕全部被拷贝所需扫描次数

设  $CX$ —全屏拷贝所需扫描的次数

$AX$ —满屏每区扫描线数目

对于 EGA 方式,  $AX = 350$  线

对于 VGA 方式,  $AX = 480$  线

其它方式,  $AX = 100$  线

则:  $CX = \frac{AX}{LINE - 1}$

考虑到不能整除的情况, 取  $CX = [AX / (LINE - 1)] + 1$ 。

## 三、程序实现

基于上述设计思想及算法, 用 8086 汇编语言编制参考程序  $SCRCOPY.ASM$  见附录。由于有较详细的注释, 为节省篇幅, 框图省略。

在  $SCRCOPY$  中, 有几个子程序未给出, 主要是因为它们与  $WHDOS17.EXE$  密切相关, 而且不影响对算法的理解。不过, 很容易地按自己特有的机型予以补齐。下面给出简要说明:

1.  $DATA-H-PRT-8$ ——将  $AL$  的内容转成适用于诸如  $AR2463$  之类的打印机所需要的格式。

2.  $QIN-BUFFER$ ——请打印缓冲区

3.  $TEZE-SUB$ ——将 16 进制参数 ( $CX$ ) 转换成以特征码表示。

4.  $PRINT-YS$ ——向打印机发送图形打印指令 (不含点阵信息)

5.  $AL-PRINT$ ——打印  $AL$  的内容。可直接用  $INT17H$  之 0 号功能实施。

6.  $HJ-TIAO-ZEN$ ——行距处理。

在此子程序中, 也包括了对特征码的处理。

对于诸如  $M2024$ ,  $LQ1000$  之类的打印机, 可直接发送完整的行距指令系列。

在  $SCRCOPY$  中, 还有一关键变量  $PRINT-HJ-2$ , 它是当打印机输出一行 (纵向 24 点) 后, 应走纸的距离 (保证上、下两行恰好相接)。不同的打印机, 此值不完全一样。

## 四、结 束 语

$SCRCOPY$  模块, 具有如下特点:

1. 支持所有的 24 针打印机 (如 2024, 3070, 8320 等)。

2. 支持多种分辨率的屏幕图形拷贝 (如  $640 \times 200$ ,  $640 \times 350$ ,  $640 \times 400$  及  $640 \times 480$ )

3. 纵、横向均可放大 1~3 倍, 共有  $3 \times 3 = 9$  种不同的组合。

缺点: 不支持 9 针, 16 针之类的打印机

不支持彩色硬拷贝。

在设计  $SCRCOPY$  模块时, 得到了湖北省水利水电科研所的黄泽钧高工的指导, 特此致谢!

为节约篇幅, 该文所附  $SCRCOPY$  源程序不在此刊出, 需要本程序的读者可向《电脑》杂志社索取, 本社只收回 10 元的成本及邮资费。■



## 一种简便的报表输出方法

四川重庆市食品公司 王培福

目前,  $FOXBASE$  已成为广大计算机用户进行数据管理的常用软件。在  $FOXBASE$  下, 要输出数字右对齐, 整数部份三位分节, 字段值为零时 (不论是整形还是非整形) 输出空字符串的表格, 方法很多, 如编写一段程序, 将已格式化的数据赋给某一变量, 再输出该变量等方法。但这些方法编写程序比较麻烦, 特别是在输出的数值字段较多的情况下, 更是如此, 而且执行速度慢, 还要占用大量外存空间。笔者在实践中, 找到了一种简便易行的方法, 就能达到上述目的。

$FOXBASE$  提供的“@ <座标> SAY <表达式> [PICTURE <格式>]”命令, 只给出了数字左对齐格式“B”, 而没有右对齐格式。笔者将该命令中的“PICTURE”中的格式置成“@Z 999,999.99”, 成功地输出了数字右对齐, 整数部份三位分节, 数值为零时输出空字符串, 数据的宽度由“@Z”后面的“格式”的宽度而定。此法编程简单, 程序占用字节少, 执行速度快, 输出的表格清晰美观, 效果很好。■

# 通用多库统计汇总程序的设计

中国科技大学 曲少宏

**摘要:** 本文分析了通用多库统计汇总程序的基本思想和实现方法,并给出了全部源程序。该程序适于各类单库或多库的按单字段或多字段无条件或有条件进行分类统计和汇总操作。

## 一、概述

统计汇总是管理信息系统的主要功能之一,无论任何管理部门,都需要报表处理和统计汇总工作,所以统计汇总模块的质量会影响整个管理系统的优劣。怎样设计灵活、快速、有效和通用的统计汇总程序,是本文要讨论的重点。

## 二、基本思想

由于 MIS 的功能大致相同或相似,各功能模块均是对数据库操作,屏幕编辑的格式和方式可以进行统一化和规范化,所以可以采用模块化层次结构设计方法建立对各种不同数据库系统工作的通用模块。对模块的要求首先是独立性强,与外界的联系尽量少,其次是可由其它模块调用。其特点是既有一定柔性又有可重用性。

## 三、实现方法:

见文后的程序清单 TYJJ.PRG.

### 1. 建立逻辑连接

采用 SET RELATION TO <关键字表达式> INTO <别名> 命令,根据两个文件中共有的关键字表达式,把一个当前打开的数据库文件连接到一个已打开的数据库文件中,使主动数据库指针的移动带动被关联数据库指针的移动。当用字段作为关键字关联时,需在被关联库上建立起索引。在 FOXBASE 关系数据库环境下能在 8 个区中顺次建立 7 个关联,即可同时对 8 个库进行统计汇总。

### 2. 生成显示菜单提供选择

当用户输入欲进行统计汇总的数据库名后,其相应的数据描述(字段名 FIELD NAME、字段类型 FIELD TYPE、字段长度 FIELD LEN、小数位数 FIELD DEC),可以采用 COPY STRUCTURE EXTENDED TO <结构描述文件> 命令,自动生成到一个结构描述数据库中,该文件的记录个数等于使用文件的字段个数。此时只需访问这个结构描述文件,系统便会遍历用户输入的库文件结构,把结构信息用菜单定义命令 @PROMPT 和菜单管理命令 MENU TO <内存变量> 关联,即可产生菜单与字段号的一一对应的影射,在屏幕上显示出来就生成了可供用户选择分类字段的菜单。利用全屏编辑功能,任意将光标在字段名菜单上移动,当选用菜单中某一字段时,只要使光标停留在该条上,按回车键即可。

### 3. 数据处理过程中用二维数组定位

由于数据处理的复杂性,有时需对数据库文件中所存贮的大量数据中的一部分进行必要的加工,而 FOXBASE 的特点是只对当前记录进行处理,为此在数据处理以前需准确地确定欲处理的数据在数据库中的位置。

在关系数据库 FOXBASE 的库文件中所存贮的大量数据是一个二维关系模型,因此库文件中的任一数据均可由两个坐标确立,并可由二维数组的二个元素表示:其中第 1 个元素为记录号,第 2 个元素为字段号。因此只要将数据库中各个坐标点的数据存入二维数组中,所有纵向和横向需要计算的逻辑表达式都可一目了然的表示出来。

### 4. 常用的五种统计汇总方式

#### (1) 分类快速汇总和分类条件统计

在数据处理中,往往要求对源数据进行分类统计操作,即按一个或几个关键字字段的取值按条件或无条件进行分类来对单库或多库的所有数值型字段求和。当用户定义条件后,需用库文件过滤命令 SET FILTER TO <逻辑表达式>,将符合条件的所有记录筛选出来,然后用 TOTAL 命令求和,完成分类求和后,只需使用 SET FILTER TO 命令取消过滤器,就会使数据库恢复完整的原样。

#### (2) 条件纵横向统计与计算

在处理复杂形式的报表时,虽然各种报表的计算方法不尽相同,但根据基数求合计数的计算仅有两种:一种是纵向计算,另一种是横向计算。两者都可为单库或多库按条件对单字段或多字段统计计算:求和、求百分比、求平均值、求总数等,并可进行多层次复杂形式求总计、合计、小计、小小计等。

#### (3) 条件单字段或多字段统计与计算

在生成条件统计汇总的过程中,可对满足条件的单库或多库的单字段或多字段的数据项进行统计与计算。统计类型有:求最大值、求最小值、求和、求平均值、求总数等。有时记录数隐含着统计数据,例如统计日处理单据时,只要对记录数统计就会得到每日处理单据的频度。

#### (4) 三维报表统计

基层单位的报表往往和上级机关的综合表的形式一样,当做综合表的汇总工作时,需将具有相同数据结构的基层表,通过 APPEND FROM 命令添加进来,然后用 TOTAL 命令按同类项合并做三维叠加生成汇总表。

#### (5) 月份月累季度年度统计

以记录号为外循环的条件,以字段号为内循环的条件,通过两重循环把数据库中的各个数据逐点存入二维数组中,用 REPLACE 命令把参加合计的各基数之和取代合计数,就可逐月逐季地完成累计操作。

#### 四、结论

本文归纳的五种统计汇总方式,可以满足各种统计计算的需要,其中包括有单库或多库、单字段或多字段、按字段分类或按条件表达式统计、纵向或横向合计、二维报表或三维报表统计、求最大值、求最小值、求和、求平均值、求总数等。本程序的通用性好、适应性强,有很好的应用价值。

• TYJJ.PRG

```
set talk off
set safe off
set devi to scrc
set date ansi
stor 0 to x1,x2,x3
i=1
do while i<=8
j=scr(i,1)
stor "" to zdmb&j,f&j,hx&j,zx&j,hz&j
i=i+1
enddo
dime zd(20),r(30,20)
clear
?'1.无条件分类快速汇总'
?'2.按条件分类快速汇总'
?'3.按条件统计计算'
?'4.按条件纵横向统计计算'
?'5.三维报表统计计算'
acce '选择统计方式' to tf
clear
input '输入库个数(最多 8 个)' to gs
if gs>8
retu
endi
acce '输入关键字的字段名' to gjz
i=1
do while i<=gs
q=subs(str(10+i,2),2,1)
p=chr(96+i)
r=subs(str(11+i,2),2,1)
do case
case i>1
s=chr(96+i)
case i>2
s=chr(95+i)
case i>3
t=chr(94+i)
case i>4
u=chr(93+i)
case i>5
v=chr(92+i)
case i>6
w=chr(91+i)
case i>7
k=chr(90+i)
endc
acce '依次输入库名' to mc&q
sele &q
use &mc&q
if i#1
if i>1.and.i<=8
```

```
index on &gjz to &mc&q
sele &q
use &mc&q index &mc&q
sele 1
set relation to &gjz into &p
endi
sele &q
endi
if tf#5'
zds=fcount()
copy to jgk&q stru exte
if tf='4'
?'下面选择各库参加横向合计的基数;字段,选中回车'
endi
sele &r
use jgk&q
zdm=""
zdc=0
zdmb=""
do while .t.
sele &r
x=1
y=0
go top
do while .not.eof()
@ x,y prompt field name
skip
if y<50
y=y+20
else
y=0
x=x+1
endi
enddo
@ x,y prompt "结束"
menu to cxh
if cxh=zds+1
if zdc=0
clear
retu
endi
else
go cxh
if len(trim(field name))>field len
stor zdc+len(trim(field name))+1 to zdc
else
stor zdc+field len+1 to zdc
endi
dele reco cxh
sele &q
if zdm=""
if i=1
zdmbl=field(cxh)
zdmbl2=field(cxh)
endi
zdm=field(cxh)
else
zdm=field(cxh)
if i=1
zdmbl=zdmbl+''+field(cxh)
zdmbl2=zdmbl2+''+field(cxh)
```

```

endi
endi
if i > 1
zmb1 = zmb1 + '+' + s + '-' -> '+' + field(cxh)
zmb2 = zmb2 + '+' + s + '-' -> '+' + field(cxh)
endi
endi
@ 13,0 say zmb2
if cxh = zds + 1
exit
endi
endd
sele &r
set filt to deleted()
copy to zj&q
set filt to
crea mz&q from zj&q
use
sele &q
use
use mz&q
appe from &mc&q
use
sele &q
use &mc&q
sele &r
use
endi
if i < = gs
i = i + 1
loop
else
exit
endi
endd
sele 1
if .not.tf = '1'.and..not.tf = '3'
acce '输入条件表达式' to tj
set filt to &tj
endi
index on &gjz to &mcl
set index to &mcl
do case
case tf = '1'
i = 1
do while i < = gs
j = str(i,1)
sele 9
use mz&j
total on &gjz to hz&j
@ 24,0 say '汇总库为 hz&j.dbf'
i = i + 1
endd
case tf = '2'
i = 1
do while i < = gs
j = str(i,1)
sele 9
use mz&j
count to x&j for &tj
total on &gjz to hz&j

```

```

@ 24,0 say '汇总库为 hz&j.dbf;'
统计数为: get x&j
i = i + 1
endd
case tf = '3'
?'1.求最大值 2.求最小值'
?'3.求总和 4.求平均值'
?'5.求记录个数 6.退出'
acce '选择统计方式' to fs
sele 1
do case
case fs = '1'.or.fs = '2'
h = 1
do while .not.zmb2 = ''
j = str(h,1)
d = at(',',zmb2)
d1 = subs(zmb2,1,d-1)
d3 = stuff(zmb2,1,d,'')
if len(trim(d3))#0
d2 = subs(zmb2,d+1)
endi
f&j = d1
go top
s = &f&j
go 2
do while .not.eof()
my = str(&f&j)
my = trim(my)
mm = val(my)
if fs = '1'
s = max(s,mm)
endi
if fs = '2'
s = min(s,mm)
endi
skip
endd
@ 24,0 say s
if .not.eof()
zmb2 = d2
h = h + 1
loop
else
exit
endi
endd
case fs = '3'.or.fs = '4'
i = 1
do while i < = gs
j = str(i,1)
sele 9
use mz&j
count to x&j
total on &gjz to hz&j
@ 24,0 say '汇总库为 hz&j.dbf;统计数为: get x&j'
use
if fs = '4'
sele 9
use hz&j
jls1 = recount()
zds1 = fcount()

```

```

l=1
do while l<=jls1
go 1
k=2
do while k<=zds1
zd(k)=field(k)
stor zd(k) to z3
repl &z3 with &z3 / jls1
k=k+1
endd
l=l+1
endd
endi
@ 24,0 say zdmdb2
i=i+1
endd
use
case fs='5'
i=1
do while i<=gs
j=str(i,1)
sele &j
use mz&j
count to x&j
@ 24,0 say '统计数为:' get x&j
use
i=i+1
endd
case fs='6'
retu
endc
case tf='4'
hjm=''
input '输入横向合计字段号' to hj
sele 1
hjm=field(hj)
repl all &hjm with &zdmdb1
jls=reccount()
zds=fcount()
i=2
do while i<=zds
zd(i)=field(i)
k=1
go top
do while .not.eof()
stor zd(i) to z3
r(recn(),i)=&z3
k=k+1
stor spac(10) to z3
skip
endd
i=i+1
endd
clear
do while .not.eof()
display fields &zdmdb1
skip

```

```

endd
clear
input '输入各库横向合计个数' to hg
input '输入各库纵向合计个数' to zg
i=1
do while i<=gs
j=str(i,1)
sele &j
use &mc&j
k=1
do while k<=hg
l=str(k,1)
acce '输入每个库纵向合计相关数组;表达式' to hx&l
@ 24,0 say '二维数组第 1 个元素为行号,第 2 个为字段号'
&hx&l
k=k+1
if k>hg
exit
endi
endd
m=1
do while m<=zg
n=str(m,1)
acce '输入每个库纵向合计相关数组;表达式' to zx&l&zx&n
m=m+1
if m>zg
exit
endi
endd
use
i=i+1
if i>gs
exit
else
loop
endi
endd
case tf='5'
i=2
do while i<=gs
j=str(i,1)
sele &j
use
sele 1
appe from &mc&j
i=i+1
endd
total on &gjz to hzl for &tj
@ 24,0 say '汇总库为 hzl.dbf'
sele 10
use hzl
display all off
use
endc
use
close all
retu

```





## CC-DOS2.13H 使用 CEGA 卡上之硬字库

中国纺机职工大学郑州分校 刘群

CC-DOS2.13H 推出以来,深受广大用户的欢迎.它支持多种显示卡,如:CGA、EGA、VGA、COLOR 400、HERCULES,在显示汉字时,对于没有装汉卡的系统,它可以使用软字库;对于装有 2.13 汉卡的系统,它可以使用其上的硬字库.但是,它没有提供读取 CEGA 卡上的硬字库来显示汉字的接口程序.这就使拥有 CEGA 卡的用户,在使用 DOS2.13H 时,显示汉字不得不使用软字库,这不仅浪费了已有的系统资源,而且软字库又占用了宝贵的系统主存储器.为了解决这一问题,笔者通过对 CC-DOS2.13H 进行剖析,结合 CEGA 卡的特点,编写了使 CC-DOS 2.13H 可以访问 CEGA 卡上的 CGROM (固化汉字库)、读取 CEGA 卡上的汉字字模的程序 FILE9.ASM (程序清单附后).

本程序已在装有 CEGA 卡的 GW-0520DH 以及 GW-286B 上运行通过,也可以在其它装有 CEGA 卡的机器上运行.

为了便于朋友们阅读本程序,现将一些要点说明如下:

1.由于 CC-DOS2.13H 用 INT 7F 读 16 \* 16 点阵显示字库,所以,本程序旨在建立新的 INT 7F 中断.

2.CC-DOS 2.13H 之 INT 7F 为读 16 \* 16 点阵显示字库 (水平点阵)

入口参数: DX = 汉字内码

返回参数: DX = 点阵数据起始段 (偏移 = 0)

3.本程序使用 CEGA 之 ROM BIO 提供的 10 类中断之 16 号功能调用,将汉字之字模传送到字模缓冲区.

入口参数: AH = 16

BP: BX = 字模数据的传送地址

DX = 汉字内码

由于 CC-DOS2.13H 将原来的 ROM BIO 之 10 类中断,修改为 78 类中断,故我们在程序中使用的是 INT 78H.

4.FILE9.ASM 是按 COM 格式编写的,可先将其汇编 (MASM)、连接 (LINK) 生成 FILE9.EXE.然后,用 EXE2BIN 将其转化为 COM 文件.

运行时,只需将原来 CC-DOS2.13H 系统之 AUTOEXEC.BAT 文件中的 FILE1A 2 用 FILE9 替换即可.

FILE9.ASM 清单:

```
cseg segment
org 100h
assume cs:cseg,ds:cseg,ss:cseg
start: jmp begin
buffer db 64 dup(0) ;字模缓冲区
```

```
int 7f proc far ;新的 7F 中断
push bx
push bp
push es
push ds
push si
push di
push ax
push cx
xor bx,bx
mov bp,offset buffer
push bp
mov ah,16h
int 78h ;读 32 字节之字模至缓冲区
xor di,di
xor si,si
mov cx,20h
pop ax
mov ds,ax
add ax,2
mov es,ax
push ax
movsb ;将左半个汉字之字模
push ds ;及右半个汉字之字模
pop ax ;按水平点阵放入缓冲区
add ax,1
mov ds,ax
sub si,1
movsb
push ds
pop ax
sub ax,1
mov ds,ax
loop aa
pop dx
pop cx
pop ax
pop di
pop si
pop ds
pop es
pop bp
pop bx
iret
finish equ
int 7f endp
begin: mov dx,offset int 7f
mov ax,257fh
int 21h ;建立新 7F 中断
lea dx,finish
int 27h ;驻留内存
cseg ends
end start
```

## 一九九〇年广东省青少年计算机程序设计竞赛

## 试题分析及其选解(上)

郭家炜

一九九〇年广东省青少年计算机程序设计竞赛,于一九九〇年七月中旬在深圳市举行。下面就 BASIC 组竞赛试题(该试题已在本刊的一九九〇年第四期刊登了),以及各路选手在解题中出现的一些问题作一个分析,并给出有关的参考答案。

## 笔试部分

第一题:这题难度不大,在竞赛中我发现有些选手由于求胜心急,唯恐考不出水平,心情紧张,故未能细心地分析题意,忘乎所以地急于求解。结果是本来可以得分的题目却得不到。其实只要细心分析题意,边阅读题,边思考,边画,就很容易找到正确的算法。例如为了使问题简单,取  $N=10$ ,把这 10 个自然依次组成的串,首尾相接成了一个环,如图 1。因为这 10 个数具体怎样放我们是不知道的,只是知道第一次取出的  $X_1=1$ ,第二次取出的  $X_4=2$ ……。所以我们把每次取到的数在圆圈里写上相应的数字,然后想象把它拿开,这时自然会有  $X_2$  紧跟  $X_{10}$ ,而  $X_3$  是紧跟  $X_2$ ,第二次取的依题意应取的是第 4 个数  $X_4$ 。……如此下去。我们会直观地发现,若用数组元素  $X(1), X(2), \dots, X(10)$  来描述这 10 个数。那么算法是应该使  $X(1)=1, X(4)=2, X(7)=3, X(10)=4$ ,即每当下标增加 3 后元素值增加 1,但在下一轮取时,由于第 1, 4 个数被拿走了,依题意这时取的应该是  $X_5$ ,而算法应该使  $X(5)=5$ 。显然这时下标增加 5。

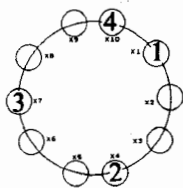


图 1

在程序设计技巧里,可采用计数器  $S$ , 一个指针  $I$ 。当  $I$  所指向的数组元素  $X(I)$  的值为 0 时让计数器  $S=S+1$ , 当  $X(I)$  的值不为 0 时让计数器  $S=S+0$ , 具体通过下面程序的 40 语句实现。请看下面的程序。

```
5 HOME: X=0
10 INPUT "N, M="; N, M
20 DIM X(N): X(1)=1: H=1
30 FOR I=1 TO N
40 S=S+NOT X(I)
50 IF S<>M THEN 80
60 H=H+1: X(I)=H: S=0
70 IF H=N THEN 100
80 NEXT I: GOTO 30
100 FOR Y=1 TO N
```

```
105 PRINT TAB (T*3+1);
110 PRINT X(Y);
120 T=Y-INT(Y/8)*8
125 IF T=0 THEN PRINT
130 NEXT Y: END
```

第二题:这道题难度中等。要求对给定的图形进行分割,问共有多少种不同的分法。(规定旋转和翻转都不算是新的分法)一般的选手都会用回溯搜索的算法来求解。其实,当  $n=1$  时,只有一种分法如图 2。(粗线表示分割轨迹)。



图 2

当  $n=2$ , 就复杂了如图 3

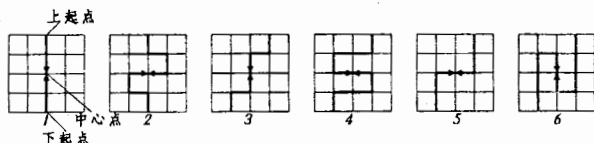


图 3

我们可以看作由上起点,下起点出发开始划分,而划分经过的每个结点用一个二维数组  $A$  表示:  $A(X, Y)=1$  表示划分时经过  $(X, Y)$  点,且都是以图形的中心点对称的。因此只要划分的轨迹沿图形内的直线前进,而不相交,最后会汇合于中心点。这就算是找到一种分法。但能够根据题目的条件确定算法的有效输入,一部分的选手就未能做到。什么是这个算法的有效输入呢?由于旋转和翻转不算是新的分法,所以可以考虑找方格其中的一半为起点开始划分,以及以中轴线为起点划分,第一次离开中轴线时只认定一个方向。正如下面的程序是通过 130 语句,165 语句,来确定算法的有效输入。请看下面的程序。

```
10 INPUT "n="; N: N1=N+1: N2=2*N: N3=N2+2
20 DIM A(N2+2, N2+2), P(100), SP(100)
30 X0=N1: Y0=N1: T=0
35 SX(1)=0: SY(1)=1: SX(2)=-1: SY(2)=0: SX(3)=0: SY(3)=-1: SX(4)=1: SY(4)=0
75 A(X0, Y0)=1
130 FOR J2=N1 TO N2
140 I=1: X=J2: Y=1: P(I)=0
150 P(I)=P(I)+1
160 IF P(I)>4 THEN 280
165 IF (P(I)=4) AND (J2=N1) AND (I=Y) THEN 280
170 X=X+SX(P(I)): Y=Y+SY(P(I))
```

```

174 SP (I) = P (I)
175 IF (X=X0) AND (Y=Y0) THEN 250
180 IF (X>N2) OR (X<=1) OR (Y>N2) OR (Y<=1)
THEN 260
210 IF A (X, Y) = 1 THEN 260
240 A (X, Y) = 1: A (N3-X, N3-Y) = 1: I=I+1: GOTO 150
250 T=T+1
260 X=X-SX (P (I)): Y=Y-SY (P (I))
270 GOTO 150
280 I=I-1: IF I<=0 THEN 300
283 A (X, Y) = 0: A (N3-X, N3-Y) = 0: P (I+1) = 0: SP
(I+1) = 0
290 X=X-SX (SP (I)): Y=Y-SY (SP (I)): GOTO 150
300 NEXT J2
310 PRINT "n="; N;: PRINT: PRINT "Total="; T
330 END

```

第三题：这道题难度中等。它主要测试考生对字符串处理技术中的位串处理的知识。我们知道在 BASIC 语言一般没有位处理功能，因而要把位串的处理设法变成对整数的运算和处理。这就是位串处理算法的难点，我们知道对于一个复杂度较大的算法来说，可以采用分而治之的方法，各个击破。例如在考虑位串的建立时，可以设位串所在的整型字为 WRD，而在 WRD 中存放的内容先放在数组 CB (I)，I 从 1~15 单元中，而每个单元的内容要么为 1，要么为 0。下面程序中的 540 行~570 行就实现了位串的建立。另外，无论在位串的读出，及移位。有一个条件是容易忽视的，它就是  $WRD > = 16384$ ，事实上当 WRD 的第 14 位为 1 时，上述条件成立，当 WRD 的第 14 位为 0 时，上述条件不成立。所以 WRD 的最高位第 14 位，就成了位串处理时的关键位。正如下面的程序中第 400 行~450 行是位串读出的处理。而第 160 行~200 行是位串位移的处理。在位串的逻辑运算方面考生都做得较好，但仍有部分考生对两个位串的逻辑异或 (XOR) 概念模糊而失分。我们知道逻辑变量 A, B 的异或运算可用这个式子表示： $F = A \oplus B$ ，其真值表如下：

A	B	F
0	0	0
1	0	1
0	1	1
1	1	0

由真值表又可把上式改写为  $F = \overline{A}B + A\overline{B}$ 。相应地在程序中求两个位串的逻辑异或时，我们可把其算法描述为：若  $(C1=0 \text{ AND } C2=1) \text{ OR } (C1=1 \text{ AND } C2=0)$  为真，则  $X=X+1$ 。这里 X 是表示异或的值。下面给出完整的程序供参考。

```

10 DIM CB (15)
100 GOSUB 520
110 Q1=WRD
120 GOSUB 520
130 Q2=WRD

```

```

140 W1=Q1: W2=Q2
150 D=0: R=0: T=0: X=0
160 FOR I=1 TO 15
170 C1=0: C2=0
180 IF W1>=16384 THEN C1=1: W1=W1-16384
190 IF W2>=16384 THEN C2=1: W2=W2-16384
200 W1=W1+W1: W2=W2+W2
210 D=D+D
220 IF C1=1 AND C2=1 THEN D=D+1
230 R=R+R
240 IF C1=1 OR C2=1 THEN R=R+1
250 T=T+T
260 IF C1=0 THEN T=T+1
270 X=X+X
280 IF (C1=0 AND C2=1) OR (C1=1 AND C2=0) THEN
X=X+1
290 NEXT I
300 W=Q1: PRINT "WRD1:"; W;: GOSUB 400
310 W=Q2: PRINT "WRD2:"; W;: GOSUB 400
320 W=D: PRINT "AND:"; W;: GOSUB 400
330 W=R: PRINT "OR:"; W;: GOSUB 400
340 W=T: PRINT "NOT:"; W;: GOSUB 400
350 W=X: PRINT "XOR:"; W;: GOSUB 400
360 END
400 TEMP=W
410 FOR I=1 TO 15
420 CB (I) = 0
430 IF TEMP>=16384 THEN CB (I) = 1: TEMP=TEMP-16384
440 TEMP=TEMP+TEMP
450 NEXT I
460 FOR I=1 TO 15
470 PRINT CB (I);
480 NEXT I: PRINT
490 RETURN
520 FOR I=1 TO 15: READ CB (I): NEXT I
530 DATA 0, 1, 1, 0, 0, 1, 0, 1, 1, 0, 0, 0, 0, 1, 1
530 DATA 0, 1, 1, 0, 0, 1, 0, 1, 1, 0, 1, 1, 0, 0, 1
540 WRD=0
550 FOR I=1 TO 15
560 WRD=WRD+WRD+CB (I)
570 NEXT I
580 RETURN

```

## 快讯 广东省将于 5 月 12 日举行

### 初级程序员联合考试

1991 年广东省部分县、市计算机应用软件人员联合水平考试 (初级程序员级) 将于 5 月 12 日 (星期天) 举行，现将《考试大纲》刊登，有关辅导资料可到报名点购买 (地址下详)。

中心考场设在广州市职业技术教育中心，各县、市满 30 名考生以上可设分考场。

报名时间：3 月 1 日~4 月 15 日

报名地点：广州市小北下塘西路 41 号

广州市职业技术教育中心

联系人：何文生、彭兴民、要志东

电话：350464

邮政编码：510050

# 1991 年广东省部分县、市计算机应用软件人员联合水平考试

## 初级程序员级考试大纲

### 一、基本要求

初级程序员应在微机硬、软件基本知识方面达到相当于计算机专业中专(中技)的水平。

- 1、熟悉微型计算机基本工作原理;
- 2、能阅读及编写 BASIC 程序或 DBASE III 程序;
- 3、能熟练掌握 IBM-PC 或 APPLE II 一种微机的操作、维护及常用软件的使用;
- 4、具有一定的中、英文录入能力。

### 二、考试内容

上午试 2 小时(满分 100 分)

#### (一) 硬件基础知识 (35 分)

##### 1、数制及其转换 (8%)

- (1) 二进、十进、十六进制等常用数制及其转换
- (2) 定点数与浮点数

##### 2、机器代码 (7%)

- (1) 原码、反码、补码
- (2) BCD 码与 ASCII 码等常用的编码方式
- (3) 汉字编码初步
- (4) 奇偶校验码的校验方法

##### 3、算术运算与逻辑运算 (10%)

- (1) 二进制数的计算机运算方法
- (2) 逻辑代数的基本运算
- (3) 逻辑式的化简与变换

##### 4、计算机的主要部件 (12%)

- (1) 中央处理器 CPU (运算器、控制器、寄存器组)
- (2) 存储器(内存与外存)功能、种类、特性
- (3) I/O 设备功能、种类、特性
- (4) 总线
- (5) 系统框图、指令和数据流

#### (二) 软件基础知识 (45 分)

##### 1、程序语言基础知识 (5%)

- (1) 机器语言、汇编、编译、解释系统的基础知识
- (2) 常见高级语言种类(机器语言主要应用范围)

##### 2、操作系统使用的基础知识 (21%)

- (1) 操作系统的类型与功能
- (2) 结构与进程的概念
- (3) 文件组织的类型与特点
- (4) 常用文件命令和文件命令之使用

##### 3、基本数据结构 (8%)

- (1) 数组、记录、列表、栈、堆、队列的存贮与操作
- (2) 树的存贮与基本操作

##### 4、基本的程序控制结构 (8%)

- (1) 流程图的标准形式(结构程序框图)

##### (2) 程序的结构化

##### 5、常用应用软件初步知识 (3%)

#### (三) 操作与维护的基本知识 (20 分)

##### 1、计算机及其外部设备的使用

##### 2、操作系统、用户 BASIC 程序或 DBASE III 程序的操作方法

##### 3、常见错误信息(系统错误、程序错误、操作错误)的判断和处理

##### 4、汉字编码、汉字输入方案的基础知识

##### 5、汉字系统的启动、汉字的输入、编辑、保存和打印的操作方法

##### 6、磁盘的规格、种类、用途、使用方法与保护

##### 7、常见硬件故障的判断与处理

下午试 2 小时(满分 100 分)

#### (一) 程序的阅读与编写能力 (70 分)

##### 1、能阅读和理解流程图 (5%)

##### 2、具有阅读 BASIC 程序或 DBASE III 程序能力 (40%)

##### 3、能对流程图、程序的正确性进行分析与纠正 (5%)

##### 4、具有一定编写 BASIC 或 DBASE III 程序的能力 (10%)

##### 5、能了解基本算法 (10%)

查找、更新、排序、合并及字符处理

#### (二) 应用软件的使用 (15 分)

数据库 DBASE III、字处理 WORDSTAR

表处理 LOTUS1-2-3 或 VISICAL

任选其一

#### (三) 专业英语理解能力 (15 分)

##### 1、具有相当于高中毕业生英语程序(词汇、词法和翻译)

##### 2、能正确阅读和理解计算机领域的简单资料

操作考核 1 小时 (满分 100 分)

操作考核采用 IBM-PC 或 APPLE II 机型

#### (一) 基本操作 (20%)

##### 1、操作系统基本命令、树结构目录

##### 2、常用软件的启动与退出

#### (二) BASIC 程序操作 (50%) 或 (10%)

##### 1、BASIC 程序的编辑、保存、运行

##### 2、BASIC 程序的调试、修改

(转 19 页)

欢迎交流

计算机爱好者

# 软件库

欢迎交流

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 片，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，俱乐部会员按会员证反面说明优惠（汇款时请注明会员编号）。邮包费每次收 5 元。

二、邮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、本刊今年第三期以前的目录无效，以软件库的目录为准。

七、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，圆括号内 J 指游戏杆，K 指键盘。

## IBM 机软件 工 具 类

IT103 设计图形符的有力工具 (Graphics Font Design Utility) V3.0 这是一个供 TURBO PASCAL 程序员使用的图形符设计的实用软件。本软件可用于科学表达式（化学式、数学公式、科学表达式上、下标等）、非英文字母以及美术图形符等字符的显示和打印。本软件要求配有与 CGA 兼容的彩显卡，可在 TURBO PASCAL 3.0 及其以上版本的环境下运行。(30 元/片)

IT104 多用途的实用程序库 (MISCELLANEOUS UTILITIES) 这是一个功能齐全，用途广泛的实用程序集，其中包括电子邮件管理程序，产生批处理文件程序，打印日历程序，查询当前时间程序，时钟程序，修整磁盘程序，显示磁盘分配表 (FAT) 程序，检查、转储和维护文件程序，用户日志系统以及建立两个用户区的程序。值得一提的是，运行建立两个用户区的程序，可以暂时挂起当前正在运行的程序，保存系统当前状态，执行另一个程序的操作，这样用户好象有两台电脑在运行一样，这是一个既象多任务而实质上又不是多任务的操作，十分有趣！ [1]

IT105 扩充 DOS 功能程序库 (一) (HOTKEY/ XDOS/ EZ MENU) 本软件包括三组具有丰富功能的扩充 DOS 命令的程序库，其中 HOTKEY 程序组用于重新定义功能键，XDOS 则是具有报警时钟，替代 DIR 命令，DOS 的 HELP 程序，加锁，加密，封打印（改在屏幕输出）等功能，EZ MENU 是一组提供立即运行 EXE、BAT、COM、BAS 文件的菜单，用户通过使用这个菜单来运行程序，更显得方便。 [1]

IT106 DOS 外部命令集锦 (一) (EXTENSIONS#1) 本软件提供了 27 个 DOS 外部命令，大大丰富了 DOS 的命令库，从而令用户使用 MS-DOS 更显得得心应手。本软件所提供的外部程序除了使用方便以外（与一般 DOS 命令使用方法完全相同），还具有在命令操作中一次能指定多个命令参数的特点，这是一般 DOS 命令所少有的。这 27 个外部命令虽然都是些实用校器，但它们将为用户提供便利，并弥补了 MS-DOS 外部命令之不足，这 27 个外部命令包括：指定发声命令 ALARM，查看或修改一个或多个文件属性命令 ATT，测算 DOS 命令或程序运行时间的命令 CLK，连接文件命令 CAT，对文件目录排序命令 DIRSRT，通用搜索命令 GPM，显示盘、系统及内存信息的命令 INFO，将多行数据输入到 DOS 的 pipe 中的命令 INP，强化型列目录命令 LS 等等。 [1]

IT107 数学教学实用软件包 (MATH PAK II 3.0) 这是一种用于数学教学的软件，采用菜单驱动，包括四种进制的数据转换，解一元二次方程，求一阶导数，求二阶导数，极坐标与平面坐标转换，一次方程求解，用矩阵法解二元、三元一次方程，统计计算，求对数等运算，这是一个既能用于教学，也能用于实际数学计算的软件，也是一个面向个人计算机用户和中等、师范、中专等学校学生的软件。 [1]

IT108 最小二乘解软件包 (LISQ) 这是一个用 FORTRAN 语言编写的程序集，包含了求解的几种方法的程序。由于本软件直接提供源程序，更适用于用户学习和使用。(30 元/1 片)

IT109 线性方程组求解软件包 (LEP) 这是一个为求解线性方程组而设计的用 FORTRAN 语言编写的程序集，在本软件包中提供了一些常用的求解线性方程组的方法，并提供源程序，适用于从事数学教学和学习的大学院校教师，大学生和从事数值计算的工程技术人员使用。(30 元/1 片)

IT110 适于 C 语言开发者的 PC 工具箱 (PC-tools) 这是一个为使用 C 语言的开发人员提供的一组类似于 UNIX 软件工具的实用程序，其中包括浏览文件、搜寻字串的 browse 程序，将正文文档打印几次以加深打印黑度的 darken 程序，列出文档的 ASCII 码或 16 进制码的 dump 程序，将多个已排序文档合并成一个排序文档的 merge 程序，屏阅读文档的 page 程序，不用编辑器实现字符串替换的 svtst 程序，快速单词计数的 wc 程序和确认/不用确认删除文档的 rm 程序。 [1]

## 文 书 类

IW15 一种功能强劲易用菜单驱动的字处理器 Freeword 这是一种比 WORDSTAR 还易用的字处理器，它可以使零散的正文左右对齐，能令光标在字、词、行、屏、页间自由跳跃，对一段已作标记的正文执行复制、删除、移动、单词搜索，此外，它还允许在打印一文件的同时，屏幕上继续对另一文件进行编辑作业。 [1]

## 语 言 类

IL43 UNIFORTH (缩略版) 这是一个第四代计算机语言 UNIFORTH 的缩略版软件，该缩略版是从 Unified software systems 开发的专业版 UNIFORTH 选出的一个子集，它支持 FORTH-83 标准，它包括一个主程序和汇编器，屏幕编辑器，该软件由于其精巧，是学习、应用 FORTH 语言的一个十分便利的软件，为便于用户使用，本软件配有约万字的中文说明书。(150 元/片)

IL44 FACILIS 2.0 一种 PASCAL 编译器 FACILIS 是 PASCAL 程序设计语言的扩充子集的 P 一代码编译程序，它基于 Niklaus wirth 的 PASCAL 编译程序，FACILIS 本身是用 PASCAL 编写，在 TURBO PASCAL 上编译。(30 元/1 片)

IL45 PASCAL 自学软件 (Visible pascal) 这是一个可运行观察到其

内部操作的 PASCAL 语言编译系统,它不仅可给用户观察到系统的内部操作,而且有很完美的屏幕编辑功能。本软件附有 19 个示范程序,用户可以在利用它的过程中掌握本软件的使用技巧和功能。 [1]

### 游 戏 类

- IG1 八卦算命游戏 [1]
- IG2 麻将(MJ和MAH两种) [1]
- IG3 游戏集锦 I(CAT, TETRIS.....) [1]
- IG4 CDRAGON 潜艇游戏 [1]
- IG5 King's Quest 王子历险记 [1]
- IG6 Big Top 过障碍 [1]
- IG7 PC-POOL 台球 [1]
- IG8 TE 砌积木 [1]
- IG9 MAIN 拳击 [1]
- IG10 MYCHESS 国际象棋 [1]
- IG11 Burger Time 汉堡包 [1]
- IG12 跳鸟与蛇 [1]
- IG13 PITSTOP II 赛车 [1]
- IG14 十项全能 [1]
- IG15 SERPENTINE 食客 [1]
- IG16 武士道 [1]
- IG17 PARTROOPER 地炮防空 [1]
- IG18 青蛙过河 [1]
- IG19 打伞兵 [1]
- IG20 PRERARE FOR WAR 古代战争 [1]
- IG21 INRILTYATOY 模拟飞行 [1]
- IG22 BABY 教小孩 [1]
- IG23 MOON BUG 月球战车 [1]
- IG24 CONTRA 魂斗罗 [1]
- IG25 TEST DRIVE 悬崖飞车 [1]
- IG26 PIANO SIMULATOR 钢琴模拟演奏 [1]
- IG27 YURBO-BRIDGE 高级桥牌 [1]
- IG28 DOUBLE PRAGON 双截龙 [2]
- IG29 FALCON F-16 战机 [1]
- IG30 THE LAST NINJA 最后的忍者 [2]
- IG31 变形金刚 [4]
- IG32 SO 推积木 [1]
- IG33 中国象棋, 五子棋等 [1]
- IG34 FOOTBALL 足球赛 [1]
- IG35 OPERATION CLEAN STREETS 毒品街(缉毒行动) [2]
- IG36 BLOCK OUT 立体俄罗斯方块 [1]
- IG37 AR 空降游骑兵 [1]
- IG38 圣女贞德 [2]
- IG39 HEAVY METAL 重金属坦克 [1]
- IG40 RAD DUDES 龙之忍者 [2]
- IG41 GP 一级方程式赛车 [2]
- IG42 TECH NOCOP 魔鬼战警 [3]
- IG43 IKARI 史泰龙 [1]
- IG44 BARBARIAN 屠龙记 [1]
- IG45 SHADOW 影之门 [2]
- IG46 RENE GADE 热血高校 [2]
- IG47 VICTORY BOAD 怒号层圈 [1]
- IG48 GUERRILLA 古巴反战 [2]
- IG49 SKY FOX II 星际狐狸战机 [1]
- IG50 TAPPER 酒吧间 [1]
- IG51 NORTH SOUTH 90 南北战争 [1]
- IG52 三国志 [3]
- IG53 游戏集锦 II (NEWDIG, SOP, BRICKS, FLIGHT, PANGO) [1]
- IG55 HEAVY BARRED 无敌神枪 [2]
- IG56 CAVEMAN 原始人奥运会 [4]
- IG57 TOP GUN 壮志凌云 [1]
- IG58 F-19 战机(AT机) [3]

- IG59 游戏集锦 III (空战、掘地虫、吃豆、桌球、CRO、海底大战等) [1]
- IG60 GO 围棋 [1]
- IG61 DGATH SWORD 死亡之剑 [1]
- IG62 TEST DRIVE II 名车大赛 II [2]
- IG63 XENON II 山星异形 [2]
- IG64 WINTER GAME 冬季奥运 [4]
- IG65 SUMMER GAME 夏季奥运 [5]
- IG67 PIRETES 海盗 [2]
- IG68 SPACE QUEST III 宇宙传奇 III [6]
- IG70 BAD STREET 黑街风云 [1]
- IG71 重金属坦克说明盘 [1]
- IG72 BUBBLE 泡泡龙 [2]
- IG73 DEMON'S WINTER 冬之魔(有使用资料介绍,另收1.6元资料费) [3]
- IG74 MIGHT MAGIC II 魔法门(另收使用说明资料费1.2元) [3]
- IG75 SPACE ROGUE 银河英雄 [2]
- IG76 KNIGHTS LEGEND 武士传说(89年版,另收资料费4元) [5]
- IG77 CHAMPIOMS OF KRYNN 克莱恩英雄(90年版,另收资料费2.1元) [4]
- IG78 波斯王子(另收资料费1.2元) [2]
- IG79 COLORADO 寻金记 [4]
- IG80 MI TANK PLATOON 坦克模拟 [3]
- IG81 DYNAMIX PKODUCTION 魔鬼克星 [4]
- IG82 SILENT SERVICE II 死亡潜航 [4]
- IG83 SUPER BIKE CHALLENGE 超级摩托车大赛 [2]
- IG84 GOTO 中途岛之战 [1]
- IG85 LICENCE TO KILL 杀人执照 [1]
- IG86 PATTON VS ROMMEL 巴顿对龙美尔(战争游戏) [1]
- IG87 IF IT MOVES SHOOT IT 终极孤鹰 [4]
- IG88 绿色兵团 [2]
- IG89 RAMBO III 兰博(第一滴血) [2]
- IG90 APOLLO 18 阿波罗登月 [1]
- IG91 MAGIC 未来之魔法 [2]
- IG92 FIRE POWER 坦克大决斗 [2]
- IG93 忍者传奇 [2]
- IG94 KARNOY 卡诺夫(武打游戏) [2]
- IG95 FS3 模拟飞行 [2]
- IG96 飞向北越 [1]
- IG97 688 ATTACK SUB [1]
- IG98 TENNIS 网球 [4]
- IG99 SUPER PINBALL 超级针球 [1]
- IG100 台球 [1]
- IG101 钻石方块(智力游戏) [1]
- IG102 平面俄罗斯方块 [1]
- IG103 MONSTER MATN V1.0 斗兽算术游戏(适合小学生用) [1]
- IG104 罪恶浪潮(286/386机用) [8]

### APPLE 机类

#### 自然、生物、医学类

- AE53 心理学演示
- AE54 生物学模拟(树的生成)
- AE54 生物学遗传学辅助教学
- AE55 物种起源(假设了3100种生命的进化过程)
- AE56 绿色植物的光合作用与新陈代谢(软汉字)

#### 数学类

- AE58 幕函数辅助教学
- AE59 二次函数图象与性质
- AE60 平面解析几何辅助教学
- AE61 立体图与三视图转换