



● 一九九一年 ● 总期第76期

7

# 電子

744

# 與 電腦



OEMs Welcome!

• ELECTRONICS AND COMPUTERS •



# 中华学习机汉字处理 (CPC) 卡

## 电脑写作的好帮手



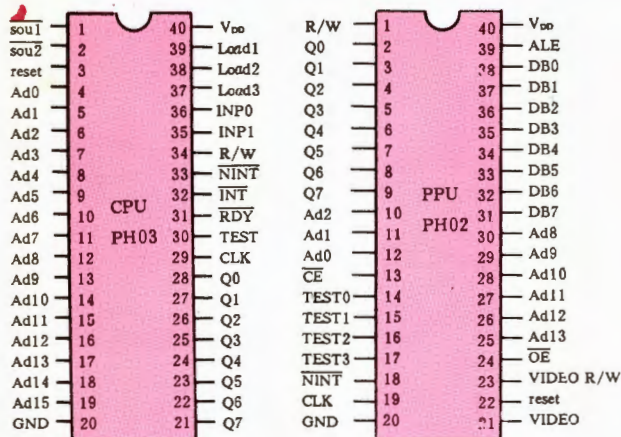
- 汉字全屏编辑
- 打印机驱动
- 五笔字型输入

- 北京科盾新技术开发公司 研制
- 《电子与电脑》读者服务部 经销

中华学习机汉字处理卡 (以下简称CPC卡) 是专为CEC-1型中华学习机设计的五笔字型输入、打印机驱动和汉字全屏编辑的多功能卡。插上该卡后, 中华学习机在汉字状态下, 除可以使用原有的“区位”、“拼音”两种汉字输入方法外, 还可使用“五笔”方式输入。同时还可使用CPC卡上的汉字全屏编辑软件以及打印驱动功能。CPC卡虽然使中华学习机增加了三项功能, 但未改动和减少原中华学习机的任何功能。

CPC卡具体使用方法见本期有关内容。





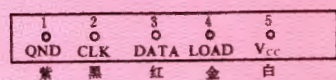
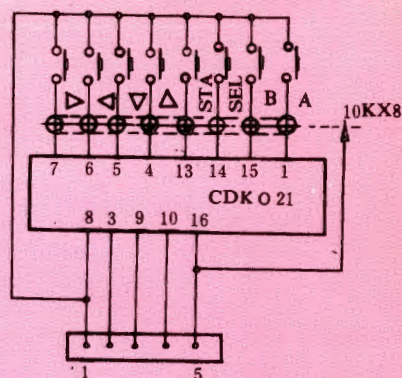
表一 CPU 引脚功能及其正常工作电压

引脚号	信号名称	信号说明	正常工作电压(V)
1	sou1	第一音频信号输出	0
2	sou2	第二音频信号输出	0.5
3	reset	复位输入端。由一个低电平信号控制,使 CPU 结束当前工作而返回程序开始时的监控状态。	1.5
4	Ad0	Ad0~Ad15 地址位 0~15。这些地址线在系统内对存储器和 PPU、游戏节目卡进行寻址,以取出需要的信息及把运算结果送给它们。	4.2
5~19	Ad1~Ad15	地址线	
20	GND	地线	0
21~28	Q7~Q0	Q7~Q0 为输入/输出数据总线,共有 8 位。CPU 与 PPU、存储器、游戏节目卡交换信息,都是通过数据线传递的。	0
29	CLK	时钟信号输入端,频率高达 21.47727MHz	0.1
30	TEST	测试点,生产厂家测试用。	0
31	RDY	总线准备好信号输出端。	2.7
32	INT	中断请求信号输入端。	4.2
33	NINT		
34	R/W	读写存储器及游戏卡信息控制信号输出端。	4.2
35	INP1	手柄操作控制信号输出端。	4.2
36	INP0		4.2
37	Load3	加载控制信号输出端。	0
38	Load2		
39	Load1		
40	VDD	电源	5

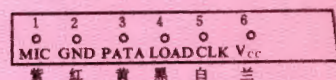
表二 PPU 引脚功能及其正常工作电压参考值

引脚号	信号名称	信号说明	正常工作电压(V)
1	R/W	读写存储器、游戏卡信息控制信号输出端。	4.2
2~9	Q0~Q7	Q0~Q7,8 位双向数据总线	0
10	Ad2	地址输入线。供 CPU 查找 PPU 片内存储器内容时寻址使用。	4.2
11	Ad1		
12	Ad0		
13	CE	片选信号输出端。低电平有效。	5
14	TEST	测试点	0
15	TEST		
16	TEST		
17	TEST		
18	NINT	非屏蔽中断信号输入端。	0
19	CLK	时钟信号输入端,频率 21.47727MHz	4.2
20	GND	地线	0
21	VIDEO	视频信号输出端。	1.3
22	reset	复位信号输入端,低电平有效。	3.9
23	VIDEO R/W	视频信号读写控制信号输入端。	3.9
24	OE	允许视频信号输出控制端。	3.9
25~30	Ad13~Ad8	地址线	3.9
31~38	DB7~DB0	地址/数据线。双向。	3.9
39	ALE	地址锁存信号输出端。	0
40	VDD	电源电压输入端	5

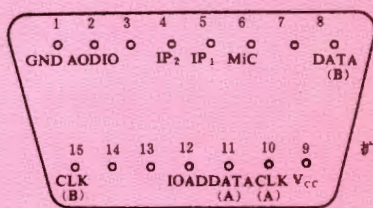




### 手柄接口



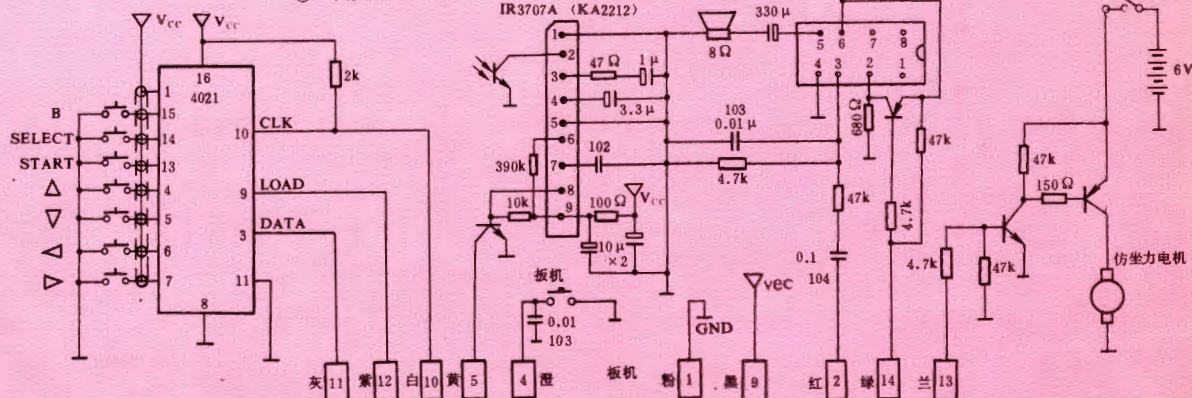
7



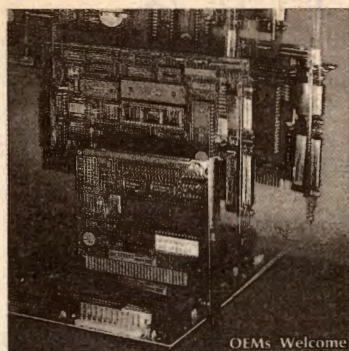
1



#### ④ 手枪电路







• ELECTRONICS AND COMPUTERS •

一九九一年

总期第76期

# 電子與電腦

## 目 录

### • 综述 •

80年代微机发展回顾及90年代展望(续) ..... 陈志雄(2)

### • PC 用户 •

用未公布的dBASE III命令开发图形功能 ..... 金林樵(4)  
谈dBASE III“模糊”查询 ..... 班艳玲(5)  
FOXBASE编程技巧两则 ..... 贺文华(5)  
恢复误删除文件的原理和方法 ..... 朱大公(6)  
微机绘制平面曲线 ..... 魏新民(9)  
若干子目录下共用一个dBASE III(或WS)软件包 ..... 龚文彪(11)  
小议PC-DOS的拷贝命令 ..... 刘杰(11)  
能读多位数的计算机程序 ..... 周礼森(12)  
模拟钢琴演奏 ..... 冯剑秘(12)  
如何对中文DOS的汉字库进行直接读写 ..... 刘旭华(13)  
NC—新型文件管理工具软件 ..... 杨进(14)

### • 学习机之友 •

为无驱动器系统设置虚拟磁盘 ..... 刘庆丰(15)  
巧用LOGO解难题 ..... 廖大勇(16)  
修改CEC汉字系统程序 ..... 周易(16)  
浅谈计算机在中专数学中的应用 ..... 王志平 顾正刚(17)  
数据块的后移 ..... 冯端品(18)  
从QXC卡读取汉字字模 ..... 牛海峰(18)  
ProDOS磁盘操作系统入门 ..... 廖凯(19)  
在APPLESOFT系统下实现HSCRN功能

..... 李齐(21)

BASIC语言中几个命令功能的扩展 ..... 那履弘(21)

### • C语言初阶讲座 •

第六讲 宏指令的用法 ..... 李文兵(23)

### • 学用单片机 •

单片机旅馆客房门卫系统 ..... 傅士淦(27)

### • 电脑巧开发 •

MSX-EPROM读写/擦除器 ..... 刘园(32)

### • 维修经验谈 •

任天堂系列游戏机检修实例 ..... 林华春(34)

让PC机接上家用彩电

——CGA—PAL卡即将问世 ..... 编辑部(35)

微计算机维修工具

——8088系统板故障诊断卡(续) ..... 董珊(36)

磁盘驱动器修复一例 ..... 雷勇(38)

### • 初级程序员级水平考试辅导讲座 •

汉字dBASE III使用技巧(中) ..... 王路敬(39)

### • 新书与软件 •

TANGO印刷线路板自动布线软件包 ..... (45)

电子工业出版社新书介绍 ..... (45)

电子工业出版社软件部发行CCED V3.0中文字

表编辑软件 ..... (46)

### • 读者联谊 •

致参加“普及型PC个人用户软件交流联谊活动”的用户

..... 农科院计算中心(47)

机械电子工业部电子工业出版社主办

编辑、出版：《电子与电脑》编辑部

(北京173信箱 邮政编码：100036)

印刷：北京三二〇九厂

国内总发行：北京报刊发行局

国内统一刊号：CN11—2199

邮发代号：2—888

国外代号：M924

出版日期：每月23日

主编：王惠民 副主编：王昌铭

责任编辑：施玉新

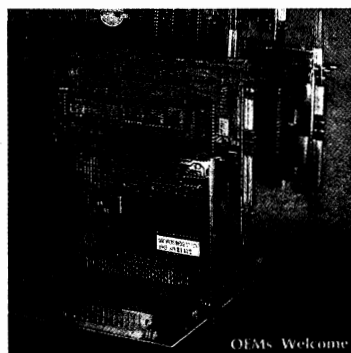
订购处：全国各地邮电局

国外总发行：中国国际图书贸易总公司

(北京399信箱 邮政编码100044)

广告经营许可证：京海工商广字147号

定价：0.95元



• ELECTRONICS AND COMPUTERS •

一九九一年

总期第76期

# 電子與電腦

## 目 录

### • 综述 •

- 80年代微机发展回顾及90年代展望(续) ..... 陈志雄(2)

### • PC 用户 •

- 用未公布的 dBASE III 命令开发图形功能 ... 金林樵(4)  
谈 dBASE III “模糊”查询 ..... 班艳玲(5)  
FOXBASE 编程技巧两例 ..... 贺文华(5)  
恢复误删除文件的原理和方法 ..... 朱大公(6)  
微机绘制平面曲线 ..... 魏新民(9)  
若干子目录下共用一个 dBASE III (或 WS) 软件包  
..... 龚文彪(11)  
小议 PC-DOS 的拷贝命令 ..... 刘 杰(11)  
能读多位数的计算机程序 ..... 周礼森(12)  
模拟钢琴演奏 ..... 冯剑秘(12)  
如何对中文 DOS 的汉字库进行直接读写  
..... 刘旭华(13)  
NC—新型文件管理工具软件 ..... 杨 进(14)

### • 学习机之友 •

- 为无驱动器系统设置虚拟磁盘 ..... 刘庆丰(15)  
巧用 LOGO 解难题 ..... 廖大勇(16)  
修改 CEC 汉字系统程序 ..... 周 易(16)  
浅谈计算机在中专数学中的应用  
..... 王志平 顾正刚(17)  
数据块的后移 ..... 冯端品(18)  
从 QXC 卡读取汉字字模 ..... 牛海峰(18)  
ProDOS 磁盘操作系统入门 ..... 廖 凯(19)  
在 APPLESOFT 系统下实现 HSCRN 功能

- ..... 李 齐(21)  
BASIC 语言中几个命令功能的扩展 ..... 那履弘(21)

### • C 语言初阶讲座 •

- 第六讲 宏指令的用法 ..... 李文兵(23)

### • 学用单片机 •

- 单片机旅馆客房门卫系统 ..... 傅士淦(27)

### • 电脑巧开发 •

- MSX—EPROM 读写/擦除器 ..... 刘 园(32)

### • 维修经验谈 •

- 任天堂系列游戏机检修实例 ..... 林华春(34)  
让 PC 机接上家用彩电  
——CGA—PAL 卡即将问世 ..... 编辑部(35)  
微计算机维修工具  
——8088 系统板故障诊断卡(续) ..... 董 珊(36)  
磁盘驱动器修复一例 ..... 雷 勇(38)

### • 初级程序员级水平考试辅导讲座 •

- 汉字 dBASE III 使用技巧(中) ..... 王路敬(39)

### • 新书与软件 •

- TANGO 印刷线路板自动布线软件包 ..... (45)  
电子工业出版社新书介绍 ..... (45)  
电子工业出版社软件部发行 CCED V3.0 中文字  
表编辑软件 ..... (46)

### • 读者联谊 •

- 致参加“普及型 PC 个人用户软件交流联谊活动”的用户  
..... 农科院计算中心(47)

机械电子工业部电子工业出版社主办

编辑、出版：《电子与电脑》编辑部  
(北京 173 信箱 邮政编码：100036)

印刷：北京三二〇九厂

国内总发行：北京报刊发行局

国内统一刊号：CN11—2199

邮发代号：2—888

国外代号：M924

出版日期：每月 23 日

主编：王惠民 副主编：王昌铭

责任编辑：施玉新

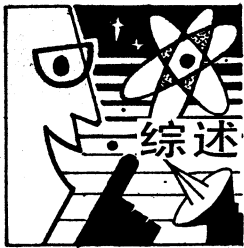
订购处：全国各地邮电局

国外总发行：中国国际图书贸易总公司

(北京 399 信箱 邮政编码 100044)

广告经营许可证：京海工商广字 147 号

定价：0.95 元



# 80 年代微机发展回顾及 90 年代展望(续)

艺高电脑国际有限公司总经理 陈志雄

工作站的总线在功能以及速度上是工作站和微机分别最大的地方。工作站的总线亦如它的微机处理器一样没有一个标准, SUN 的 S-BUS 及 VME BUS 都同时存在 SUN 不同的系统里, 有趣的是 IBM 最新推出的 System6000 是采用 MCA(微通道总线)。System6000 在性能上已可以跟其它的主流工作站竞争。而 System6000 配上 MCA 对它在市场上的影响则虽多作观察才能作出分析。

## 三、显示方式的发展

显示及图形功能是微机及工作站分别最大的地方, 早期 IBM 的单色显示及彩色显示标准性能较差, 单色图形显示方面已被美国 Hercules 公司开发的单色图形显示(720×350)所代替, 而在彩色图形显示方面, IBM 亦不停推出新的标准, 从开始时的 CGA(640×200)到 EGA(640×400)至现时最流行的 VGA(640×480, 800×600, 1024×768)都是在分辨率及显示颜色数量上作出改善, 而图形的发生全靠以微机中央处理器来完成, 所以在速度上微机图形发生比工作站的逊色得多。

反之工作站的生产厂家都有各自的图形标准, 而这些图形的发生都是以独立的图形处理器来完成, 为了不同的需要多数工作站的生产厂家都有其自己设计的大规模图形处理器(如 SUN, DEC, IBM, Silicon Graphics 等), 但新一代现成的图形处理器(如 Intel 的 i860 及 Texas Instrument 的 TMS340 系列)都有很强的性能, 所以图形处理将会转向标准化器件而不用专用芯片, 如 HP/Apollo 的 VRX 图形处理卡便是用上了 Intel 的 i860, 而 IBM 用在它的 X 终端上的图形处理便是 Texas Instrument 的 TMS34010。

在 90 年代微机的显示及图形功能可能会向工作站的方向迈进, 这是在改善微机总线后的合理结果。IBM PS/2 的微通道及 ESIA 总线的设计是便利多个处理器在同一总线上并行运算, IBM 于 90 年 10 月便首先推出在微通道上操作的 XGA(Extended Graphics Array)显示及图形处理标准。新的 XGA 包含了一个高速的 VGA, 一个图形辅助处理器及总线控制功能, 所以 XGA 除了可以往下跟 VGA 的软件兼容外, 新开发的软件可以利用它的辅助处理器来增加图形发生的速度, 而 XGA 的总线控制功能更能在中央处理器工作的当中, 直接访问主机内存, 提高了整个形功能的速度。

在 ELSA 总线上操作的图形处理标准估计亦在今后两三年内推出, 它的功能相信会跟 XGA 大同小异,

但最关键的还是谁能创造一个大家都公认的标准。

## 四、其他关键技术的发展

### 1. 显示器技术

以往显示器都是以 12 英寸及 14 英寸为主流, 而以后的趋势将会多元化, 从 9 英寸至 19 英寸以适合不同的需要, 而 RGB 模拟彩色显示器因价格关系会一步步取代 TTL 数字彩色显示器。90 年代最大的发展可能会是大屏幕彩色液晶显示器, 这是世界关注的一个发展, 因为现时彩色 LCD 技术仍然是掌握于日本厂家的手中, 如彩色 LCD 的价格能够回落至合理水平的话它便极有可能取代了阴极射线管了。

### 2. 存储技术

光盘一直是磁盘的一种威胁, 但直至今在光盘技术还未有可以取代磁盘的突破, 光盘现时在价格、性能、体积上都未能取胜磁盘, 反之磁盘不断有新的发展, 如现时的 2 1/2 英寸 40 兆字节硬盘在便携式计算机上已极为流行, 市场上亦传言新的 1 1/2 英寸硬盘很快便出现。

在脱机后备方面, 光盘还受到 8mm 录象带技术的威胁。现在, 一个小小的 8mm 盒带已可以记录上千兆字节的信息, 所以它正开始成为网络后备资料存储的标准。

光盘技术中, 现时唯一胜出的要算是 CD-ROM 了, 许多厂家利用 CD-ROM 来发送技术手册, 文件和软件。在 90 年代, 我们将可以见到 CD-ROM 在工作站及部份微机中成为一标准配件, 使用 CD-ROM 来传递印刷品的年代已为期不远了。

### 3. 网络技术

IBM 在 80 年代中期推出的“Token Ring”网络系统已证明不成功而现时最流行的差不多都是“以太网”(Ethernet), 但:“以太网”的传输速度只有 10M bit/s, 在 90 年代多媒体通讯开始在计算机上实现时, 网络在速度上遇到的压力将会很大, 尤其是计算机将不单是传送文本, 而且还要传送语音, 图形和图象。相信传送速度达 100M bit/s 的光纤网络会被推动而形成规模经济, 但在此举未成之前, Ethernet 还会继续维持其领导地位。

### 4. 微机的设计技术

半导体技术的进步不单是微处理器功能增加, 对整台计算机的微型化有很大的影响。除了人所共知的存储芯片从 80 年代开始时的 64K, 256K 到现时的 1M 及 4M 外, 环绕着中央处理器的外围 IC 如 DMA



(Direct Memory Access), 存储管理 (Memory Management), 总线控制, 中断控制等等, 都得到集成化。早期的 8088PC/XT 及 80286AT 都需要上百片的集成电路组成, 因为母板上基本计算机功能所需面积太大, 所以许多外围功能如并/串联、显示器、软/硬盘控制器等都需要以插卡的形式来完成。

1985 年美国的 Chips and Technology 公司开始把 286AT 微机的外围芯片集成在 5 片大规模集成电路上, 这方向证明很成功, 很多半导体设计公司便因此而成立加入竞争行列, 如 SUNTAC, VLSI, Zymos, Acer, Twin Head, Opti, Head land 等等。而一些大型公司如 Texas Instrument INTEL 和 Motorola 亦不甘人后相继加入竞争。结果是集成度不断的提高, 从五芯片一组的变成最新有单芯片的 286AT 及 386SX 微机(微处理器, 内存 IC, EPROM 及一些零散的逻辑 IC 不包括在内)。

踏进 90 年代, 有能力的半导体公司如 Intel, AMD 等更会进一步把微处理器和外围芯片都集而为一。估计 91 年美国 AMD 公司将会推出创新的 80286 "Motherchip", 即把 80286 和它的外围电路集而为一的"母片"。可以看得到的是母板上的零件会大量减少, 计算机设计及制造厂家将会把更多的外围功能放在母板上变成基本功能如并/串联接口、软盘控制器、软盘控制器、硬盘接口 (IDE 接口) 等, 这对母板上扩充槽数量上的要求便减少了, 因此计算机的外形可设计得更加小和更加美观。

## 五、市场的发展

根据美国市场研究公司 Datquest 的资料显示, 1990 年全世界微机的总销量为 2500 万台, 估计在 1994 年全世界微机的总销量会增加到 4000 万台总值 630 亿美元即每年有 12% 左右的增长。

在市场占有率方面, 除了超级计算机之外, 大型及小型计算机的市场占有率将会一步步减少, 估计它们的增长只有 6—7% 左右。所以到 1994 年工作站加上微机在市场的总值将会跟小型、大型和超级计算机的总值扯平。相信在 1995 年以后, 以微处理器为基础的计算机在市场总值上将会越过小型、大型和超级计算机。

在微机本身的发展方向, 便携式的生长将会比坐台式的大。在 1989 年坐台式销售量为 2000 万台左右占微机市场的 90%, 估计 1994 年坐台式微机的销售

量为 2500 万台占整个微机市场的 63%, 而便携式微机更会发展得品种繁多, 大致可分为:

1. 膝上形(重量在 5 公斤左右)——在 1994 年市场占有率大概为所有微机的 8% 左右, 总机量 310 万台。

2. 记事簿形(重量在 2 公斤左右)——在 1994 年市场占有率预测为所有微机的 16%, 总机量 640 万台左右。

3. 掌上形(重量在 1 公斤之下)——在 1994 年市场占有率预测为所有微机的 13%, 总机量 520 万台左右。

在微处理器方面, 8088/8086 一向被人认为会被淘汰, 但因为掌上形微机的掘起, 在 92 年开始会重纳增长轨道, 而 80286 微机将会在 90 年代步入负增长, 数量最大而增长亦最高的。据美国 Dataquest 的估计是 80386SX 微机, 而 386DX 和 80486 虽然有稳定的增长, 但在数量上会远远不及 386SX。

以上预测是 Dataquest 公司前一段时间作出的, 但他可能未能及时把 AMD 生产的 80386DX 和 Intel 公司宣布的 80486SX 可能带来的影响估计在内。相信 AMD 在 91 年或 92 年将会开始把 80386DX 降价, 而 Intel 的对策是推出 80486SX (注: 80486SX 是 80486DX 减去了辅助处理器)。在两者竞争的情况下 386SX 能否继续处于领先地位上要看以后的发展, 但得益的最后都是计算机的家用。

经过十年的市场及技术发展, 可以预见的是在 90 年代微机将会更普及化, 而其技术发展亦趋向更微型化及多元化。

在一个统一标准的局限下微机在技术上的发展在 80 年代是追不上工作站, 但在 Intel 80386 32 位微处理器推出之后估计标准微处理器已定型, 只要半导体厂家能生产兼容 80386 的话, 以后的发展可能不会被统一标准所局限。在微机总线有所改良而其他辅助器件在不停的改进下, 微机开始有条件进军工作站市场。但工作站的领导公司亦不停地努力将自己的产品变成一个标准, 但要推进这方面的工作可能比发展技术来得更困难。

无论如何我们在 90 年代将会看到的微机的发展不会比 80 年代的逊色, 差不多可以断言 90 年代微机的功能价格比仍然有一到两个数量级的改善。

## 会议消息

中国计算机用户协会 IBM 微机协会定于今年 10 月 21 日至 25 日在成都望江宾馆召开协会第四届年会(全国微机应用技术交流会。会议交流内容为:

1. 计算机网络与通讯
2. 管理信息系统
3. 办公自动化系统
4. 电子、机械 CAD

### 5. 机电一体化产品

### 6. 计算机文字处理系统

会间作有关技术报告并评选优秀论文, 欢迎协会理事单位、会员单位、微机用户以及从事微机研究开发、生产销售、技术服务的单位派人参加会议。

凡确定参加会议者请于 8 月底前向 IBM 微机用户协会办公室联系。

通讯地址: 北京学院南路 55 号 1314 室

电话: 8316539





## PC用户

# 用未公布的 dBASE III 命令开发图形功能

金林樵

人们普遍认为 dBASE III 本身没有图形功能,因此总是求助于 PASCAL 等有强大图形功能的软件,从 dBASE III 数据库中间接或直接读得绘图数据后,将其显示出来。其实,我们可以借助 dBASE III 的未公布命令来开发它的图形处理能力。

### 一、未公布的命令及函数

在使用 dBASE III 时,我们发现了若干条在 dBASE III 使用手册、参考书等资料中未见公布的命令和函数。借助于它们可在 dBASE III 系统下直接绘出所需的图形。这些命令和函数为:

#### 1. SET SEGMENT TO <段地址>

它用来定义其后的 POKE, PEEK 函数中内存地址所使用到的段地址。

#### 2. POKE <内存地址>, <数据>

它用来将数据写入所指定的内存地址单元中。

#### 3. PEEK (<内存地址>)

它将指定的内存地址单元的内容读出。

后两个函数中内存地址的段地址,由其前的最后一个 SET SEGMENT 命令中所指定的段地址决定。命令中的段地址,内存地址和数据均为十进制数,且不能是变量或表达式。

### 二、屏幕绘图的基本原理

在 PC/XT 的彩色图形显示器下,分配给屏幕的缓冲区为 16K 内存。它位于段地址 47104(&HB800)中,偏移地址为 0 到 16383。屏幕的分辨率为 640×200。缓冲区中每个字节的每一位对应屏幕上的一个图形点。因此,每行 640 个点占用 80 个字节。屏幕显示分偶数和奇数行。偶数行对应的显示缓冲区地址为 0~7999;奇数行分配地址为 8192~16191。因此,在绘制图形时,应注意以下的换算关系:

将显示行除 2 后,结果为整数(设值为 P)时,则其内存地址为:  $P * 80 + \text{INT}(\text{列数}/8)$ ;若不为整数,则说明为奇数行,这时将该数取整(设值为 P1)后,其对应地址为:  $8192 + P1 * 80 + \text{INT}(\text{列数}/8)$ 。例如 POKE 80,255 将在(2,0)至(2,7)间画一水平线。

### 三、图形功能的实现

根据下列数据库结构,绘出直方图。数据库 JIN. DBF 的结构如下:

字段	字段名	类型	宽度	小数位
1	班级	C	4	
2	平均成绩	N	4	

数据库中数据如下:

记录号	班级	平均成绩
1	8911	80
2	8912	75

3	8921	89
4	8922	71
5	8931	85

下附的程序就可根据上述数据库绘出直方图。本例在 IBM PC/XT 上调试通过,并可用屏幕硬拷贝打印出图形。

```

SET TALK OFF
CLEAR
SET SEGMENT TO 47104
Y=140
X=20
N=0
I=0
Y1=100
@ 0,24 SAY "浙江电子工业学校数学成绩示意图"
DO WHILE N<Y1
    AD=(INT(Y/2)-I-1)*80+X
    IF INT(N/2)=N/2
        STOR "POKE"+STR(AD+8192,6)+" ,3" TO PO
    ELSE
        STOR "POKE"+STR(AD,6)+" ,3" TO PO
        I=I+1
    ENDIF
    &PO
    N=N+1
ENDDO
I=0
X1=40
AD=INT(Y/2)*80+X
DO WHILE I<X1
    STOR "POKE"+STR(AD+I)+" ,255" TO PO
    &PO
    I=I+1
ENDDO
@ (Y-Y1)/16-1,X-3 SAY "平均成绩"
@ Y/16-1,X+X1+2 SAY "班级"
USE JIN
L=LEN(班级)+1
DO WHILE .NOT. EOF()
    STOR 平均成绩 TO J
    STOR 班级 TO BJ
    N=1
    I=0
    DO WHILE N<J
        AD=(INT(Y/2)-I)*80+X+L
        IF INT(N/2)=N/2
            STOR "POKE"+STR(AD+8192,6)+" ,255" TO PO
        ELSE
            STOR "POKE"+STR(AD,6)+" ,255" TO PO
        
```

```

I=I+1
ENDIF
&.PO
N=N+1
ENDDO
X=X+L
@ Y/16,X-1 SAY BJ
SKIP
ENDDO
USE
RETURN

```

## 浅谈 dBASE III “模糊”查询

黑龙江友谊县红兴农校 班艳玲

关于汉字 dBASE III 的模糊查询方法很多,但这些方法都不能满足用户的全部要求。如,有的用户要求把查到的信息从打印机上输出,如果用屏幕硬拷贝,则效果不好。为了解决这类问题,本人在编程中发现用 SET FILTER TO (条件) 这条命令,可以满足用户的打印要求,程序如下:

```

set print off
set talk off
clear
use(数据库)
go top
acce "请输入要查人的姓名(只需一字)" to h2
set filter to '&h2.' $ 姓名
copy to bh.dbf
* (注:可在此处插入屏幕显示)
wait "您需打印吗(y/n)?" to h3
if h3="y".or.h3="Y"
clear
wait "按任一键开始打印..."
set device to print
set print on
use bh
go top
? line1
? line2
do while .not.eof()
? line3
? jl
skip
enddo
? line5
?
endif
set print off
retu

```

注:此种方法在 IBM PC/XT 机上通过,其中变量 line1,line2,line3,line4,line5 为表格变量,变量 jl 为记录信息。

## FOXBASE 编程技巧两则

湖南娄底师专 贺文华

### 一、FOXBASE 中 ON 命令的巧用

FOXBASE 中的 ON 命令在指定的条件出现后将引起程序执行的转移。这种条件可能是在程序的执行过程中按了 ESC 键或其它键而引起的。为避免这种程序执行的转移引起对打开的数据库文件及其索引文件造成破坏致使数据丢失,我们可巧用 ON 命令,编制一个有意义的中断处理子程序:

```

SET BELL ON
SET CONSOLE OFF
?? CHR(7)
@ 8,20
@ 8,20 SAY '警告;'
@ 8,30 SAY '系统运行时不得强行中断!'
I=INKEY(5)
@ 8,20
SET CONSOLE ON
RETURN

```

调用这一子程序,只需在主程序的开头设置一个 ON ESCAPE DO ESC KEY

命令即可。当用户企图通过 ESC 键中断程序运行时,此程序将发出警告,稍停后继续程序的运行。

### 二、打印机未准备就绪的预处理

在数据库管理系统中,由于打印机未准备就绪而导致程序出错是一个常见的问题,有趣的是我们可以利用 FOXBASE 中提供的 SYS 函数来预防。

FOXBASE 中函数 SYS 可返回当前打印机的状态。在 MS-DOS 下,如果打印机没有联机或接通电源,将返回信息“OFFLINE”;如果打印机在线,则返回信息“READY”。据此,编制如下程序段来预防由于打印机故障而导致程序运行出错的情况。

```

***** 打印机未准备好处理子程序 *****
DO WHILE .T.
PASS='P'
SET CONSOLE OFF
IF SYS(13)="OFFLINE"
SET BELL ON
?? CHR(7)
DO WHILE UPPER(PASS)(<) 'Y' AND UPPER(PASS)(<) 'N'
? "打印机未准备好,重试吗(y/n):"
ACCEPT TO PASS
ENDDO
ENDIF
DO CASE
CASE UPPER(PASS)='N'
SET CONSOLE ON

```



```
EXIT
CASE UPPER(PASS) = 'P'
SET CONSOLE ON
DO 打印程序
```

```
EXIT
ENDCASE
ENDDO
RETURN
```

## 恢复误删除文件的原理和方法

朱大公

磁盘文件被误删除是令人沮丧的事。本文提供的源程序,经汇编和链接生成可执行文件后,用户只需键入文件名,根据屏幕提示进行简单操作,便可恢复被误删除的文件。

为使读者对程序有较好的理解,本文拟从文件目录表(FDT)和文件分配表(FAT)谈起。

DOS 对文件的管理功能,主要是通过 FDT、FAT、磁盘参数表和设备驱动程序实现的,而与恢复被误删除文件有直接关系的是 FDT 和 FAT。

对于容量 360K 字节的软磁盘,DOS 把从相对扇区 5 开始的 7 个连续扇区分配给 FDT。FDT 包含 112 个目录项,因此根目录下最多可存放 112 个文件。一个目录项占 32 字节,分成下面 8 部分(表 1)。

位移	域	长度(字节)
00H—07H	文件名	8
08H—0AH	扩展名	3
0BH	文件属性	1
0CH—15H	保留	10
16H—17H	文件建立或最后修改日期	2
18H—19H	文件建立或最后修改时间	2
1AH—1BH	文件首簇号	2
1CH—1FH	文件长度(字节数)	4

表 1 目录项结构(位移相对于目录项首字节)

图 1 是某软盘上 FDT 的一部分,现结合表 1 和图 1 说明目录项各部分的含义。

3F70:0000	49 42 4D 42 49 4F 20 20-43 4F 4D 27 00 00 00 00	I B M B I O	C O M	....
3F70:0010	00 00 00 00 00 00 00 60-54 07 02 00 80 12 00 00	.....	T	.....
3F70:0020	E5 45 42 55 47 20 20 20-43 4F 4D 20 00 00 00 00	e E B U G	C O M	....
3F70:0030	00 00 00 00 00 00 00 60-68 06 2A 00 80 2E 00 00	.....	h	*. ....
3F70:0040	43 4F 4D 4D 41 4E 44 20-43 4F 4D 20 00 00 00 00	C O M M A N D	C O M	....
3F70:0050	00 00 00 00 00 00 00 60-54 07 24 00 80 45 00 00	.....	T	\$. .E..
3F70:0060	00 F6 F6 F6 F6 F6 F6 F6-F6 F6 F6 F6 F6 F6 F6 F6	. v		
3F70:0070	F6 F6 F6 F6 F6 F6 F6 F6-F6 F6 F6 F6 F6 F6 F6 F6	v v v v v v v v v v v v v v v v v v v v		

图 1 文件目录表 FDT(部分)

文件名域 最多可存放 8 个字符的文件名,不足 8 个字符则以空格码 20H 填充。图 1 中 3F70:00 00 处开始的 8 个字节便存放文件名 IBMBIO,后跟 2 个空格。文件名域的首字节有可能出现下列情况:

00H 表示该目录项未使用过,其后的各目录项均不再被查找。空目录项的其余各字节一般填入 F6,如图 1 的 3F70:0061~007F。

E5H 文件删除标志。该目录项可在建立新文件时被使用。图 1 中 3F70:0020 处的 E5H 表示 eEBUG.COM 文件已被删除。

扩展名域 最多可放 3 个字符的扩展名。不足 3 个字符时以空格码 20H 填充。

文件属性域、DOS 保留区、日期域和时间域,它们与恢复误删除文件关系不大。

文件首簇号域 填入分配给文件的首簇号(按低、高字节顺序)。图 1 中 eEBUG.COM 的首簇号为 02AH。

文件长度域 填入文件长度字节数(按低、高顺

序),图 1 中 eEBUG.COM 的长度为 2E80 字节。

DOS 把 FAT 安排在相对扇区号 1 开始的两个连续扇区,还在随后的两个连续扇区安排一份副本,供 FAT 损坏时使用。分配给文件的磁盘空间以簇为单位,每一簇为连续的两个扇区。根据簇号可找到对应的扇区。文件的簇号信息都包含在 FAT 中。

软盘的 FAT 约有 680 个登记项,每项占 1.5 字节,即 12 位,并从第一项开始按 000H 簇、001H 簇、002H 簇等顺序编号。000H 簇的首字节为磁盘介质标志,对于软磁盘是 FDH,其余两字节保留为 FFH。因此可分配给文件的第一个可用簇为 002H 簇。磁盘上尚未存放文件时,除 000H 簇外,其余各字节均填入 00H。使用 FAT 的困难在于每个登记项的值并非实际簇号。DOS 按下列算法处理以获取实际簇号:

- 1 刚用过的簇号乘以 1.5;
- 2 取乘积的整数部分作为 FAT 中的位移;
- 3 从 FAT 表中指定位移处取出一个字;
- 4 若刚用过的簇号为偶数,取该字的低 12 位;若

刚用过的簇号为奇数,则取高 12 位。

5 上述结果若为 000H,表明该簇为空,便是可分配给文件的下一簇号;若结果为 FF7H,表示该簇已损坏不可用;若结果为 FFFH,表示刚用过的簇号是文件的最后一簇。其它情况则表明该簇已经分配。

←i 簇→			←i+1 簇→		
2	5	6	0	0	2

图 2 文件分配表 FAT 中的簇号

参看图 2,设第  $i-1$  簇为偶数,实际取出的字为 6025H,簇号取低 12 位,为 025H 簇。下一簇号应为

```

3F70:0000  FD FF FF 03 40 00 05 60-00 FF 8F 00 09 A0 00 0B
3F70:0010  C0 00 0D E0 00 0F 00 01-11 20 01 13 40 01 15 60
3F70:0020  01 17 F0 FF 19 A0 01 1B-C0 01 1D E0 01 1F 00 02
3F70:0030  21 20 02 23 F0 FF 25 60-02 27 80 02 29 B0 03 00
3F70:0040  00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
3F70:0050  00 37 80 03 39 A0 03 FF-CF 03 3D E0 03 3F 00 04
3F70:0060  41 20 04 43 40 04 45 60-04 FF 8F 04 49 A0 04 FF
3F70:0070  CF 04 FF EF 04 4F 00 05-51 F0 FF 00 00 00 00 00

```

图 3 文件分配表 FAT(部分)

现在,让我们借助图 1 和图 3 来查找文件 COMMAND.COM 的簇号链。

1 由图 1 可知,文件的首簇号为 024H;

2  $024H \times 1.5 = 036H$ ;以 036H 为偏移量(位移),在 FAT 中取出一个字,为 6025H;

4 由于 024H 为偶数,所以取 6025H 的低 12 位,为 025H。这就是分配给文件的下一簇号;

5  $025H \times 1.5$ ,取整得 037H 作为偏移量,在 FAT 表中取出一个字,为 0260H;

6 由于 025H 为奇数,所以取 0260H 的高 12 位,为 026H,它是分配文件的又一个簇号;

7 重复 2~7 步,当取出的簇号为 FFFH 时,表示文件结束。

由计算得知,文件 COMMAND.COM 的簇号链为 024H、025H、026H、027H、028H、029H、03BH、03CH、03DH、03EH、03FH、040H、041H、042H、043H、044H、045H、和 046H 共 18 个簇。注意 029H 簇以后跳至 03BH 簇,表明文件在磁盘上的存放是不连续的。

删除文件时,DOS 先在 FDT 中找到要删除的文件名,将该目录项的首字节改为 E5H,并根据文件首簇号域提供的首簇号,在 FAT 中查找文件的簇号链,将簇号链的每一登记项填入 000H,而这时该文件的内容仍完整无损地保留在磁盘上,只是不能读出去了。

将本文提供的恢复误删除文件的源程序通过行编辑程序 EDLIN.COM 输入计算机并存盘后,再经过汇编和链接便可生成 .EXE 的可执行文件。设文件名为 HFWJ.EXE,使用时在 DOS 下键入 HFWJ,然后按程序提示将含误删除文件名的软盘插入 B 驱动器,键入待恢复的文件名(大小写均可)并回车。如果恢复文件成功则返回 DOS;若读写出错或找不到待恢复的文件

026H(取出的字为 0260H,取高 12 位)。千万不能认为第  $i$  簇的簇号为 256H, $i+1$  簇簇号为 002H。

DOS 建立新文件时先扫描 FAT,跳过已分配的簇或坏簇,找出第一个空簇(其值为 000H),将该簇号填入 FDT 中相应目录项的文件首簇号域,再继续查找下一个空簇,将其簇号填入第一个可用簇的位置,依此类推,直到分配给文件的簇数满足文件长度的要求,并在最后一个可用簇的位置填入 FFFH。这样在 FAT 中便形成一条簇号链,无论文件在磁盘上的物理地址如何分散,均可正常读写。图 3 是 FAT 表的一部分:

名,程序都有相应的提示和处理。

根据 DOS 的文件管理机制,文件被删除后,一旦在磁盘上建立了新文件,或某些文件长度增加,则不仅簇号链的恢复可能出错,而且误删除文件在磁盘上的信息也将被新文件或其它信息所取代。因此要想使恢复误删除文件获得成功,应在建立新文件或增加某些文件长度以前进行。另外,当磁盘上有多个被删除文件时,恢复误删除文件的成功率也不大。

程序流程如下:

1 将 DOS 返回地址入栈、设置 DS 和 ES 段、提示在 B 驱动器插入软磁盘,等待键盘输入。

2 读 FDT,若反复读至规定次数后仍不成功则提示读 B 驱动器出错,返回 DOS。

3 提示输入误删除文件名,转换成 FDT 目录项的标准格式后存入文件名缓冲区。将文件名中的小写字母转换成大写。

4 目录项超过 112 或其首字节为 00H 时,提示无此文件并询问是否重输入误删除文件名。若键入 Y 或 y 则返回 3,否则返回 DOS。

5 若目录项首字节不是 E5H,修改目录项指针后返回 4。

6 若误删文件名与目录文件名不同,指针指向下一目录项后返回 4。

7 修改恢复目录项文件名的首字节。

8 将 FDT 写回软磁盘,若反复写至规定次数仍不成功,则提示写 B 驱动器出错后返回 DOS。

9 计算误删除文件所需簇数并读入 FAT。

10 取刚用过的簇号乘以 1.5,保留该偏移地址。

11 簇号增一得新簇号并乘以 1.5,得新偏移地址。



12 新簇号为奇数则按新偏移地址取一个字的高 12 位;新簇号为偶数则取低 12 位。若所得之值为 000H,则返回 11。

13 簇数减 1。若未减至零,根据刚用过簇号的奇

偶性,在指定的偏移地址填入新簇号后返回 10。

14 所需簇数已减至零。根据刚用过的簇号的奇偶性,在指定的偏移地址填入 FFFH,将 FAT 写回 B 驱动器后返回 DOS。

SSG	SEGMENT PARA	STACK 'STACK'	JG	CNZF	DIV	BX
	DB	32 DUP('S')	SUB	AL,20H	CMP	DX,00
SSG	ENDS		CNZF:	STOSB	JZ	CNCUS
CSS	SEGMENT		LOOP	QUZF	INC	AX
	ASSUME DS,CSS,ES,CSS,CS,CSS		LEA	SI,ML-FAT	CNCUS:	MOV CUS,AX
BEGIN	PROC FAR		LEA	BX,WJM0	SUB	SI,04
	PUSH DS		MOV	ML-112,BX	MOV	AX,[SI]
	XOR AX,AX	PAN112:	CMP	SI,ML-112	PUH	AX
	PUSH AX		JB	PAN00	LEA	BX,DXCS
	MOV AX,CSS		JMP	WUWJ	MOV	BYTE PTR [BX+07],02
	MOV DS,AX	PAN00:	MOV	AL,[SI]	MOV	BYTE PTR [BX+10],01
	PUSH DS		CMP	AL,00	MOV	CX,05
	POP ES		JNZ	E5	ZDUB:	PUSH CX
	LEA DX,TISICP		JMP	WUWJ	CALL	DXCS
	CALL TISI	E5:	MOV	AL,[SI]	INT	25H
	MOV AH,08		CMP	AL,229	POP	CX
	INT 21H		JZ	ZSI	POP	CX
	MOV CX,05	ADDSI:	ADD	SI,32	JNC	CUZAN
DUB:	PUSH CX		JMP	PAN112	LOOP	ZDUB
	CALL DXCS	ZSI:	MOV	WJMDZ,SI	JMP	CUO
	INT 25H		INC	SI	CUZAN:	POP AX
	POP CX		LEA	DI,WJM	CUNCU:	MOV QCUHAO,AX
	POP CX		INC	DI	CALL	PYDZ
	JNC TSWJM		MOV	CX,10	MOV	FATDZ,BX
	LOOP DUB		REPZ	CMPSB	MOV	AX,QCUHAO
CUO:	LEA DX,DUBCUO		JZ	HFWMJ	MOV	XCUHAO,AX
	CALL TISI		MOV	SI,WJMDZ	ZENCU:	INC XCUHAO
	JMP JIESU		LEA	DI,WJM	MOV	AX,XCUHAO
TSWJM:	LEA DX,TISIM		JMP	ADDSI	CALL	PYDZ
	CALL TISI	HFWMJ:	MOV	SI,WJMDZ	TEST	XCUHAO,01
	LEA DX,WJM0		LEA	DI,WJM	JNZ	J1
	MOV AH,0AH		MOV	AL,[DI]	ADD	BX,OFFSET ML-FAT
	INT 21H		MOV	[SI],AL	MOV	AX,[BX]
	MOV CX,08		MOV	CX,05	AND	AX,0FFFH
	MOV BX,00	XIEB:	PUSH	CX	CMP	AX,00
	LEA SI,WJM2		CALL	DXCS	JNZ	ZENCU
	LEA DI,WJM		INT	26H	DEC	CUS
QUZIFU:	MOV AL,[SI]		POP	CX	CMP	CUS,00
	CMP AL,2EH		POP	CX	JZ	XIEFFF
	JZ CNSP1		JNC	LIN	CALL	TIANCH
	CMP AL,0DH		LOOP	XIEB	JMP	CUNCU
	JZ CNSP2		LEA	DX,XBCUO	XIEFFF:	CALL TIANSJ
	MOVS		CALL	TISI	JMP	XIEFAT
	INC BX		JMP	JIESU	J1:	ADD BX,OFFSET ML-FAT
	JMP QUZIFU	LIN:	JMP	QUUCUS	MOV	AX,[BX]
CNSP1:	INC SI	WUWJ:	LEA	DX,TISIW	AND	AX,0FFF0H
	SUB CX,BX		CALL	TISI	CMP	AX,00
	MOV AL,20H		MOV	AH,01	JNZ	ZENCU
	REPZ STOSB		INT	21H	DEC	CUS
	MOV CX,03		CMP	AL,59H	CMP	CUS,00
	MOV BX,00		JZ	JMP0	JZ	XIEFFF
	JMP QUZIFU		CMP	AL,79H	CALL	TIANCH
CNSP2:	SUB CX,BX		JNZ	JMP1	JMP	CUNCU
	MOV AL,20H	JMP0:	JMP	TSWJM	XIEFAT:	MOV CX,05
	REPZ STOSB	JMP1:	JMP	JIESU	ZXFAT:	PUSH CX
	MOV CX,11	QUUCUS:	MOV	SI,WJMDZ	CALL	DXCS
	LEA SI,WJM		ADD	SI,28	INT	26H
	MOV DI,SI		MOV	AX,[SI]	POP	CX
QUZF:	LODSB		INC	SI	POP	CX
	CMP AL,61H		INC	SI	JNC	JIESU
	JB CNZF		MOV	DX,[SI]	LOOP	ZXFAT
	CMP AL,7AH		MOV	BX,1024	LEA	DX,XBCUO

JIESU;	CALL	TISI	TEST	QCUHAO,01	ML-FAT DB	1792 DUP('ML')
BEGIN	RET		JNZ	J1-SHL	WJM0 DB	13
TISI	ENDP		OR	AX,XCUHAO	WJM1 DB	00
	PROC		MOV	[BX],AX	WJM2 DB	13 DUP(00)
	MOV	AH,09	AND	AX,OFFFH	WJM DB	11 DUP(00)
	INT	21H	JMP	RE-TCH	WJMDZ DW	00
	RET		J1-SHL; MOV	CL,04	ML-112 DW	00
TISI	ENDP		SHL	XCUHAO,CL	CUS DW	00
DXCS	PROC		OR	AX,XCUHAO	FATDZ DW	00
	MOV	AL,01	MOV	[BX],AX	QCUHAODW	00
	LEA	BX,ML-FAT	SHR	AX,CL	XCUHAODW	00
	MOV	CX,0007H	RE-TCH; RET		DUBCUO DB	0AH,0DH, '读 B:驱错! \$'
	MOV	DX,0005H	TIANCH ENDP		TISIM DB	0AH,0DH, '请输入
	RET		TIANJS PROC			误删除的文件名: - \$'
DXCS	ENDP		MOV	BX,FATDZ	TISIW DB	0AH,0DH, '无此文件!'
PYDZ	PROC		ADD	BX,OFFSET ML-FAT	DB	0AH,0DH, '要重新输入
	MOV	BX,AX	MOV	AX,[BX]		文件名吗? (Y/N)- \$'
	SHR	AX,01	TEST	QCUHAO,01	TISICP DB	0AH,0DH, '在 B:驱插入
	ADD	BX,AX	JNZ	J1FFF		含误删除文件的磁盘!'
	RET		OR	AX,OFFFH	DB	0AH,0DH,
PYDZ	ENDP		JMP	FFF		'按任意键继续... \$'
TIANCH	PROC		J1FFF; OR	AX,OFFF0H	XBCUO DB	0AH,0DH '写 B:驱错! \$'
	MOV	BX,FATDZ	FFF; MOV	[BX],AX	CSS ENDS	
	ADD	BX,OFFSET ML-FAT	RET		END	BEGIN
	MOV	AX,[BX]	TIANJS ENDP			

## 微机绘制平面曲线

长春财政学校微机室 魏新民

用微机绘制平面曲线时,往往要花很多时间设计程序。笔者在此给出一个用 BASIC 语言设计的通用程序,可在配有中分辨率(640×200)彩色显示器的 IBM-PC、XT、286 及兼容机上使用,现将程序的功能及使用方法说明如下。

此程序的通用性很强,主要体现在:1、对于任一方程,只要能用 BASIC 中的自定义函数来表示,都能画出图象;2、此程序适用于直角坐标方程  $y=F(X)$ 、极坐标方程  $\rho=\rho(\theta)$  及参数方程  $X=\varphi(t)$ ,  $y=\psi(t)$  三种类型的曲线;3、能在同一坐标系中同时画出多条(不超过 30)曲线,而且对于每一个方程,变量的取值区间可以任选。

用此程序画出的曲线准确直观,在坐标轴上都标有刻度,相邻两刻度的距离  $I$  由方程中变量的取值区间决定,比如,对于方程  $y=x^2$ ,当  $x \in [-1,1]$  时,  $I=0.1$ ,当  $x \in [-5,5]$  时,  $I=1$ ,当  $x \in [-9,9]$  时,  $I=5$ ,图画画出后在屏幕的左上角显示出  $I$  的值。

此程序使用方便,只要在运行前,先根据曲线的方程依次在程序中的 101~130 程序行之间写入  $m$  ( $m$  为方程个数)个自定义函数语句与 GOTO 语句(行号必需连续),运行后在屏幕的提示下输入“描点数(隐含值为 60)”、“图象个数(隐含值为 1)”、“变量取值区间”等信息即可,具体方法如下:

1、直角坐标方程,比如  $y=\sin x$ ,  $y=x\sin x$  与  $y=x+\sin x$ ,需写入如下三行:

```
101 DEF FNY(X)=SIN(X):GOTO 200
```

```
102 DEF FNY(X)=X * SIN(X):GOTO 200
```

```
103 DEF FNY (X)=X+SIN(X):GOTO 200
```

图 1 是画出的三条曲线(描点数 60,区间  $[-8,8]$ )

2、极坐标方程,如  $\rho=\theta$  与  $\rho=4\sin 4\theta$ ,需写入如下两个程序行:

```
101 DEF FNY(T)=T:GOTO 200
```

```
102 DEF FNY(T)=4 * SIN(4 * T):GOTO 200
```

对于直角坐标与极坐标方程,函数名必需是 FNY。图 2 为描点数 100,区间  $[0,8]$  和  $[0,6.284]$  的两条曲线。

3、参数方程,比如  $\begin{cases} x=t^2-2 \\ y=t \end{cases}$ ,需写入如下程序行:

```
101 DEF FNX(T)=T * T - 2:DEF FNY(T)=T:
GOTO 200
```

函数名必需是 FNX 与 FNY。

如方程中含有可变参量而且要画出当参量取不同值时方程所对应的曲线,或者要画出含有间断点的函数在其连续区间上的图象时,需要在 190 行中写一个自定义函数语句,对于前者,还需在 190 行中加一个 INPUT 语句。比如,对于含两个可变参量  $A$ 、 $B$  的极坐标方程  $\rho=A\cos\theta+B$ ,应将 190 行写成

```
190 DEF FNY(T)=A * COS(T)+B:INPUT A,B
```

最后要说明的是,图象画出后,反复敲 A 可使屏幕在中分辨率与高分辨率之间转换,敲 C 可改变颜色,敲 S 可将图象缩小,敲 R 重新运行,敲 Q 退出,若敲 L 键,则将图象纵向伸长,这时如进行屏幕拷贝,在打印机上得到的是标准图象。



```

5 KEY OFF;SCREEN 2;CLS
10 LOCATE 4,20;INPUT "描点数(30--300)";N%;IF
N%<30 OR N%>300 THEN N%=60
15 LOCATE ,20;INPUT "图象个数";M%;IF M%<1
THEN M%=1
20 LOCATE ,20;INPUT "方程类型(1--直角坐标;2
--参数方程;3--极坐标)";LX;LX=INT(LX);IF LX<1
OR LX>3 THEN LX=1
25 LOCATE ,20;INPUT "如方程中含可变参数或间断点
请键入 y";FT$
30 DIM X(M%,N%),Y(M%,N%),U(M%,2),X%
(M%,N%),Y%(M%,N%),O(2)
40 I%=0;II=0;CC=0;FF=0;IF M%=1 THEN 60
50 LOCATE ,20;INPUT "在同一区间画图吗 y/n";A$;
IF A$<>"Y" AND A$<>"Y" THEN 70
60 LOCATE ,20;INPUT "变量取值区间[u,v]";U,V;H
=V-U;GOTO 70
70 I%=I%+1;IF FT$="Y" OR FT$="y" THEN
180
80 ON I% GOTO 101,102,103,104,105,106,107,108,
109,110,111,112,113,114,115,116,117,118,119,120,
121,122,123,124,125,126,127,128,129,130
180 LOCATE ,20;PRINT "第";I%,"组参量 ";
200 IF M%=1 OR A$="y" OR A$="Y" THEN 280
210 LOCATE ,20;PRINT "区间 ";I%;;INPUT U
(I%,1),U(I%,2)
220 IF LX<>1 THEN 280
230 IF I%=1 THEN U1=U(1,1);V1=U(1,2)
240 FOR J%=1 TO 2;IF U(I%,J%)<U1 THEN U1=U
(I%,J%)
250 IF U(I%,J%)>V1 THEN V1=U(I%,J%)
260 NEXT J%
280 IF A$="Y" OR A$="y" OR M%=1 THEN 290
ELSE U=U(I%,1);V=U(I%,2);H=V-U
290 ON LX GOTO 300,310,320
300 FOR J%=0 TO N%;X(I%,J%)=J%*H/N%+
U;Y(I%,J%)=FNY(X(I%,J%));NEXT;GOTO 330
310 FOR J%=0 TO N%;T=J%*H/N%+U;X(I%,
J%)=FNX(T);Y(I%,J%)=FNY(T);NEXT;GOTO 330
320 FOR J%=0 TO N%;T=J%*H/N%+U;L=FNY
(T);X(I%,J%)=L * COS(T);Y(I%,J%)=L * SIN(T);
NEXT
330 IF I%=1 THEN P=Y(I%,0);Q=Y(I%,N%);IF
LX<>1 THEN U1=X(I%,0);V1=X(I%,N%)
340 FOR J%=0 TO N%;IF Y(I%,J%)<P THEN P=Y
(I%,J%)
350 IF Y(I%,J%)>Q THEN Q=Y(I%,J%)
360 IF LX=1 THEN 390
370 IF X(I%,J%)<U1 THEN U1=X(I%,J%)
380 IF X(I%,J%)>V1 THEN V1=X(I%,J%)
390 NEXT
400 IF I%<M% THEN 70
410 IF A$="N" OR A$="n" OR LX<>1 THEN U=
U1;V=V1;H=V-U
420 IF U=0 AND V>0 THEN A$=60;UA%=0;GOTO

```

```

440
430 IF U<0 AND V<=0 THEN A%=600;VA%=638;
ELSE A%=-U/H*540+60
440 IF -U/(A%-20)>V/(618-A%) THEN K=(A%
-20)/(-U) ELSE K=(618-A%)/V
450 HL%=638
460 IF P<0 AND Q<=0 THEN B%=20;QA%=0;GOTO
480
470 IF P>=0 AND Q>0 THEN B%=160;PA%=180;
ELSE B%=Q/(Q-P)*140+20
480 IF B%-5*K*P/12<=180 AND B%-5*K*Q/
12>=10 THEN KX=K;KY=5*KX/12;GOTO 510
490 IF -P/(180-B%)>Q/(B%-10) THEN K=(180
-B%)/(-P) ELSE K=(B%-10)/Q
500 KY=K;KX=12*KY/5
510 A1=A%;UA1=UA%;VA1=VA%;HL1=HL%;
KX1=KX;KY1=KY;KX2=KX;KY2=KY
520 CLS;IF U<0 AND V>0 THEN UA%=A%+KX*U
-16;VA%=A%+KX*V+16;GOTO 540
530 IF U<0 THEN UA%=A%+KX*U-16;ELSE VA%
=A%+KX*V+16
540 H1%=(HL%-VA%+UA%)/2;UB%=UA%;
UA%=H1%;A%=A%-UB%+H1%;VA%=VA%-
UB%+H1%
550 IF P<0 AND Q>0 THEN PA%=B%-KY*P+8;
QA%=B%-KY*Q-8;GOTO 570
560 IF P<0 THEN PA%=B%-KY*P+8 ELSE QA%=
B%-KY*Q-8
570 H1%=(190-PA%+QA%)/2;UB%=QA%;QA%
=H1%;B%=B%-UB%+H1%;PA%=PA%-UB%+
H1%
580 IF ABS(U)>ABS(V) THEN US=ABS(U) ELSE US=
ABS(V)
590 DW=1;O(1)=5;O(2)=2;CLS
600 II=0;WHILE DW<KY<5;DW*O((II MOD 2)+
1);II=II+1;WEND
610 II=0;WHILE DW<KY>20;DW=DW/O(2-(II
MOD 2));II=II+1;WEND
620 IF V<>0 THEN FOR I%=A% TO VA% STEP DW *
KX * V/ABS(V);PSET(I%,B%-1);NEXT
630 IF U<>0 THEN FOR I%=A% TO UA% STEP DW *
KX * U/ABS(U);PSET(I%,B%-1);NEXT I%
640 IF ABS(P)>ABS(Q) THEN US=ABS(P) ELSE US=
ABS(Q)
650 PRINT "I=";DW
660 LINE(A%,QA%)-(A%,PA%);LINE(A%-2,
QA%+4)-(A%,QA%);LINE(A%+2,QA%+4)-(A%,
QA%)
670 LINE(UA%,B%)-(VA%,B%);LINE(VA%-8,
B%-1)-(VA%,B%);LINE(VA%-8,B%+1)-(VA%,
B%)
680 IF P<>0 THEN FOR I%=B% TO PA% STEP -DW
* KY * P/ABS(P);PSET(A%+2,I%);NEXT I%
690 IF V<>0 THEN FOR I%=B% TO QA% STEP -DW
* KY * Q/ABS(Q);PSET(A%+2,I%);NEXT I%

```

```

700 FOR I%=1 TO M%:FOR J%=0 TO N%:X%(I%,
J%)=A%+KX*X(I%,J%):Y%(I%,J%)=B%-KY*Y
(I%,J%):NEXT

```

```

710 IF A2$("<z") AND A2$("<z") THEN 730
720 FOR J%=0 TO N%-1:LINE(X%(I%,J%),Y%
(I%,J%))-(X%(I%,J+1),Y%(I%,J+1),3-(I%
MOD 3):NEXT :GOTO 740

```

```

730 FOR J%=0 TO N%-1:LINE(X%(I%,J%),Y%
(I%,J%))-(X%(I%,J+1),Y%(I%,J+1)):NEXT
740 NEXT I%

```

```

750 BB$=INKEY$:IF BB$("<") THEN 750
760 BB$=INKEY$:IF BB$("<") THEN 760
770 IF A2$("<z") THEN 790

```

```

780 IF BB$="C" OR BB$="c" THEN CC=CC+1:
CC1=INT((CC1+1)/2-1):COLOR 1+(CC1 MOD 15),CC
MOD 2:GOTO 750

```

```

790 A%=A1:UA%=UA1:VA%=VA1:HL%=HL1:
KX=KX1,KY=KY1

```

```

800 IF BB$("<a") AND BB$("<A") THEN 830
810 IF A2$="z" THEN A2$="g":SCREEN 2 ELSE A2
$="z":KX=KX/2:A%=A%/2:VA%=VA%/2:UA%=
UA%/2:HL%=319:SCREEN 1:COLOR 0,1

```

```

820 GOTO 520

```

```

830 IF BB$="S" OR BB$="s" THEN FF=FF+1:IF
FF/5 THEN KX=KX/1.2^FF:KX1=KX,KY=KY/1.2^
FF:KY1=KY:IF A2$="z" THEN KX=KX/2:HL%=319:
GOTO 520 ELSE 520 ELSE 840 ELSE 850

```

```

840 FF=0:A%=A1:UA%=UA1:VA%=VA1:HL%=
HL1:KX=KX2:KY=KY2:KX1=KX2:KY1=KY2:A2$=
"g":SCREEN 2:GOTO 520

```

```

850 IF BB$("<L") AND BB$("<I") THEN 900

```

```

860 IF A2$="z" THEN SCREEN 2:A2$="g"

```

```

870 IF B%-KY*P*6/5<=189 AND B%-KY*Q*
6/5>10 THEN KY=KY*6/5:GOTO 520

```

```

880 IF -P/(180-B%)>Q/(B%-10) THEN K=(180
-B%)/(-P) ELSE K=(B%-10)/Q

```

```

890 KY=K:KX=2*K:GOTO 520

```

```

900 IF BB$="r" OR BB$="R" THEN RUN ELSE IF
BB$="q" OR BB$="Q" THEN END ELSE 750

```

## 若干子目录下共用一个 dBASE Ⅲ (或 WS) 软件包

湖北供销学校 龚文彪

利用 DOS3.3 中的 APPEND 命令和 PATH 命令配合设置就能实现在若干子目录下共用一个 dBASE Ⅲ (或 WS) 软件包。

假如在 PC/XT 机的根目录下有许多子目录,如 DOS、DB3、GZ(工资)、RS(人事)、CW(财务)等,只要把 dBASE Ⅲ 系统文件 (DBASE.EXE 和 DBASE.OVL) 放到子目录 DB3 中,其他子目录只放 dBASE Ⅲ 编的各种应用程序 (如工资程序、人档程序、财务程序等),只要在 AUTOEXEC.BAT 文件中添加如下两条命令:

```

PATH C:\; C:\DOS; C:\DB3; C:\GZ; C:\RS; C:\CW
C:\DOS APPEND C:\; C:\DOS; C:\DB3; C:\GZ; C:\
RS; C:\CW

```

修改后存盘,重新启动机器,这时你就可在任意子目录下运用 dBASE Ⅲ 编的各种应用程序。即达到在若干子目录下共用一个 dBASE Ⅲ 软件包的目的。

说明:(1)APPEND.EXE 是 DOS3.3 外部命令,放在 C 盘的子目录 DOS 中。

(2)把 WORDSTAR 字处理软件放在 DB3 子目录中,在任意子目录下用 WS 编辑各种文稿或编制、修改程序,甚感方便。其他软件包也可这样做。

## 小议 PC-DOS 的拷贝命令

辽宁师范大学物理系 刘杰

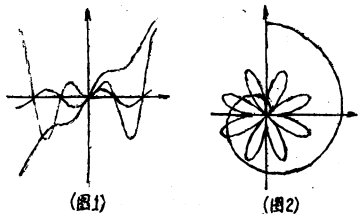
PC-DOS 有关文件拷贝的命令有:

DISKCOPY, COPY, SYS, BACKUP, RESTORE, 其中以前二者用得最多,各命令的使用不尽相同。

(1)DISKCOPY 磁盘拷贝命令是将源磁盘一模一样的复制到目的磁盘上,磁盘上的每条磁道和每个磁区的内容都一样。若目的盘未格式化,则在拷贝的同时也对目的盘格式化。它不仅能拷贝显含文件,也能拷贝隐含文件 IBMDOS.COM, IBMBIO.COM。该命令只能用于同性质磁盘间,如软盘之间,硬盘之间,不能用于软硬盘之间或不同性质驱动器之间拷贝。

(2)COPY 拷贝命令可把源磁盘显含文件的全部或部分拷贝至目的磁盘,不能拷贝隐含文件 IBMDOS.COM, IBMBIO.COM,也不能将目的盘格式化,COPY 命令能重新安排各文件在磁盘上的位置并可用于不同性质磁盘间的拷贝。

(3)SYS 是将隐含文件 IBMDOS.COM, 及 IBM-



(图1)

(图2)

BIO.COM 系统程序从第一个磁盘转移至第二个磁盘。在进行格式化时,如果要预先留下位置给系统程序,可使用 FORMAT /B。如果系统程序位置已被其它程序占用,该命令无效。

(4)BACKUP、RESTORE 是配对使用的。BACKUP 是将一个磁盘上的一些文件,转移至第二个磁盘作为备份,通常从硬盘转移至软盘,以防硬盘损坏而作备用。隐含、显含文件均可备份。BACKUP 形成的目的文件,加上了一些控制数据,这些数据是为 RESTORE 而设计的,备份盘上的文件要由 RESTORE 命令恢复后,方可正常使用,而由 COPY 命令形成的目的文件与源文件一模一样。这一点与 COPY 命令不同。

## 能读多位数的计算机程序

贵州 中国人民银行剑河支行 周礼森

贵刊在创刊号刊出的《“会念数”的程序》以及在八七年第二期刊出的《计算机能读出你输入的数》,为计算机模拟人读数的程序设计提供了思路。但所给程序均有其局限:一是两文均用拼音方式读数不直观,二是读数范围仅局限于 1~99 或 1~999 之间。

下面是笔者设计的程序,利用汉字方式能读出  $10^{-15}$ ~ $10^{-16}$  之间的任意整数和小数,读数规则完全符合人们的读数习惯。当你用键盘输入一个数时,计算机立即在屏幕上用汉字显示这个数的读法,如键入“10801.45”这个数,屏幕立即显示:一万零八百零一点四五。程序可在任何由 CCDOS 支持的微机上运行。

```
10 INPUT "请输入要判读的数";X#;A#=INT(X#)
20 IF A#=0 THEN X#=X#/10;GOTO 30
30 X$=STR$(X#);A$=STR$(A#);K=LEN(A$)-1;L=LEN(X$)-1;B=16-K
40 B$="零一二三四五六七八九";C$="千百十万千百十亿百千十万千百十个"
50 DIM C$(L),C(L),D$(L),B$(L)
60 IF A#=X# THEN E$=A$;Q=K;GOTO 80
70 E$=X$;Q=L
80 GOSUB 800
90 GOSUB 1000
100 IF A#=X# THEN 120
110 PRINT "点";FOR I=K+1 TO Q;PRINT D$(I);NEXT
120 END
800 FOR I=1 TO Q
810 IF I=K+1 THEN 850
820 C$(I)=MID$(E$,I+1,1);C(I)=VAL(C$(I));D$(I)=MID$(B$,2*C(I)+1,2)
830 IF I>K THEN 850
840 B$(I)=MID$(C$,2*(B+I)-1,2)
850 NEXT;RETURN
```

```
1000 FOR I=1 TO K
1010 IF K=1 THEN 1090
1020 IF C(I)=0 THEN 1050
1030 IF I=K THEN 1090
1040 PRINT D$(I)+B$(I);GOTO 1100
1050 IF K>8 THEN IF B$(I)="万" AND C(I-1)=0 THEN IF C(I-2)=0 AND C(I-3)=0 THEN 1100
1060 IF B$(I)="亿" OR B$(I)="万" THEN PRINT B$(I);GOTO 1080
1070 IF I=K THEN 1100
1080 IF C(I+1)=0 THEN 1100
1090 PRINT D$(I);
1100 NEXT;RETURN
```

run

请输入要判读的数? 9000000000000189

九百万亿零一百八十九

run

请输入要判读的数? 0.000876

零点零零零八七六

## 模拟钢琴演奏

重庆电力学校计算机八一班 冯剑秋

在 PC BASIC 中,一般演奏音乐的方式,是采取按预定的简谱编程序来实现的。换一首歌谱,就要另编一个程序,显得很麻烦。为了能充分利用 IBM PC 的丰富音乐功能,我编了一个小程序,使计算机能用键盘按简谱“弹”出悦耳的乐音,就像钢琴一样。

在本程序中,如果运行时你键入 C、D、E、F、G、A、B (相当于 C 调中的多、瑞、咪、发、梭、拉、西)七个键中的任意一个,便能发出相应的简谱的音。假如你要演奏高音多,那么按一下 M 键,再按 C 键,那么便是高音了。现在又要演奏中音咪,那么按一下 N 键,便回到了中音段,再按一下 E,便就是中音咪。假如又要演奏低音,那么便又按一下 N,就到了低音段,再按相应的键便可以发出低音了。要结束演奏,按字符“O”键便退出了。程序中 C\$ 中的字符串“L16”可以改变,它表示音乐的音长为 1/16 拍。读者可以按简谱把“L16”中的“16”改为其它的音长。便可以控制发音的长短。

可以看出,使用这种方法,可以随心所欲演奏任意的曲子,具有很高的趣味性,您不妨一试。

```
30 A$=INKEY$
40 IF A$="O" THEN END
50 IF A$="M" THEN B$=">";L$="T";GOTO 80
60 IF A$="N" THEN B$="<";L$="T";GOTO 80
70 L$="F"
80 IF L$="T" THEN A$=INKEY$;C$=B$+"L16"+A$
90 IF L$(">" "T") THEN C$="L16"+A$
100 PLAY C$
110 GOTO 30
```



# 如何对中文 DOS 的汉字库进行直接读写

湖南 刘旭华

笔者为了在西文 DOS 下直接显示汉字,对中文 DOS 的汉字库进行了分析,发现 CCDOS2.12H 的汉字库的记录号与汉字区位码存在下面关系。

将区码减 1 再乘以 94,然后加位码,就是本字对应库中的记录号,比如汉字“师”,区位码为 4206,对应的记录号  $R = (42-1) \times 94 + 6 = 3860$

对于 CCDOS2.1 等汉字系统的库,与上略有不同,对应关系如下(QW 为汉字区位码):

$$Q_1 = \text{INT}(QW/100)$$

$$Q_2 = QW - Q_1 \times 100$$

$$Q_1 = \begin{cases} 6 & Q_1 = 9 \\ Q_1 - 8 & Q_1 > 15 \end{cases}$$

$$R = (Q_1 - 1) \times 94 + Q_2 \quad (R \text{ 为记录号})$$

根据这些关系,我们可以直接在西文 DOS 下读取对应的汉字,并将其显示出来。

程序(一)是一个汉字显示程序,可以任意放大,并变换出许多花样。当不放大时,本程序采用了一个小技巧,显示速度很快(见 390 句)。

程序(二)是将 CCDOS2.1 的汉字显示字库转换到 CCDOS2.13H 的字库中去的一个小程序,喜欢 CCDOS2.1 的粗体字的人不妨一试。

程序一

```
1 XX=10:YY=10:YK=YY
3 INPUT KL,YL
20 KEY OFF:CLS
30 OUT 985,10
40 DIM A%(18):A%(0)=16:A%(1)=16
50 OPEN"C:\213\hzk16" AS#1 LEN=32
60 FIELD #1,32 AS A$
70 LOCATE 10,10:QW=4206:GOSUB 310
75 XX=XX+4*KL
80 LOCATE 10,12:QW=2083:GOSUB 310
85 XX=XX+4*KL
90 LOCATE 10,14:QW=2429:GOSUB 310
95 XX=KL+4*KL+XX
100 LOCATE 10,16:QW=5448:GOSUB 310
265 YY=YY+1
280 K$=INKEY$:IF K$="" THEN 280
290 CLOSE:END
310 XC=XX
320 Q=INT(QW/100):W=QW-Q*100
330 REM >49 THEN JL=(Q-1)*94+W
340 JL=(Q-1)*94+W
350 GET #1,JL
360 FOR I=1 TO 16
361 XX=XC
365 YY=YY+(YL-1)*2+1
```

```
370 A%(I+1)=CVI(MID$(A$,2*I-1,2))
371 IF KL=1 AND YL=1 THEN 380
375 GOTO 1000
380 NEXT
381 YY=YK
382 IF KL=1 AND YL=1 THEN 390
385 RETURN
390 PP=POS(0)*8-8:CC=CSRLIN*8-8:PUT(PP,CC),A%
400 RETURN
1000 X=A%(I+1):C$=""
1005 IF X<0 THEN X=(X)+2^15
1010 B$=STR$(X-INT(X/2)*2):B$=MID$(B$,2,1):C$=B$+C$
1020 X=INT(X/2)
1030 IF X=2 THEN 1010
1040 IF X=1 THEN C$="1"+C$
1045 IF LEN(C$)<15 THEN FOR U=1 TO 15-LEN(C$):C$="0"+C$:NEXT
1047 F$=MID$(C$,1,7):G$=MID$(C$,8,8):C$=G$+F$
1050 LL=LEN(C$):FOR O=1 TO LL
1060 SS$=MID$(C$,O,1)
1065 IF KL=1 THEN RR=0
1066 IF KL>1 THEN RR=KL*2
1070 XX=XX+2*(KL-1)+1:IF VAL(SS$)=1 THEN CIRCLE(XX,YY),RR/2:IF KL>1 THEN PAINT(XX+RR/2-1,YY),1
1080 NEXT
2000 GOTO 380
```

程序二

```
10 SCREEN 2,0
15 DIM A%(18):A%(0)=16:A%(1)=16
30 WIDTH 80:CLS
40 NA1$="C:\213\hzk16":NA2$="c\213\celib"
41 LOCATE 20,40:PRINT "WAITING....."
45 OPEN NA2$ AS#1 LEN=32
46 OPEN NA1$ AS#2 LEN=32
50 FOR I=1601 TO 8793
60 IF INT(I/100)=I/100 THEN 420
80 FIELD #1,32 AS A$
90 Q=INT(I/100):W=I-Q*100
100 IF Q=9 THEN Q=6
340 IF Q>15 THEN Q=Q-8
350 GET #1,(Q-1)*94+W
380 FIELD #2,32 AS B$
385 LSET B$=A$
390 Q=INT(I/100):W=I-Q*100
400 PUT #2,(Q-1)*94+W
420 NEXT I:CLOSE
```

# NC—新型文件管理工具软件

杨 进

NC—Norton Commander 是由 Peter Norton Computing 于 1986 年推出的适用于 IBM-PC 及其兼容机的新型文件管理工具软件。以其小巧的体积,灵活的使用,方便的操作而独树一帜,某些功能甚至超过了长期居统治地位的 Pctools。

NC 由四个文件组成:NC. EXE(主运行文件),NC. INI(运行环境设置文件),NC. MNU(用户定义功能键菜单文件),NC. EXT(用户定义非执行文件处理文件),总长度约为 102K,比起 Pctools 4.0 以上的版本要小得多,但在文件管理上完全可以和 Pcsell 5.0 媲美,现将其主要特点介绍如下:

一、双窗口显示,直观方便。NC 在屏幕上开辟了两个窗口,可同时显示,两个磁盘或路径的文件目录,这两个窗口的显示行数,显示方式皆可方便调节,可满足不同的需要。用 TAB 键可转换当前窗口,用 CTRL-U 可交换两个窗口的内容,还有其他一些功能键可通过 F1 获得提示帮助。值得一提的是,NC 还支持独特的 EGA 状态下可显示 43 行,更加方便对大量文件的同时处理。

二、对子目录的操作简便迅速。用过 Pctools 的人都知道,当目标盘的树目录结构十分庞大时,要想改变当前路径往往是十分费时费力的,相比之下 NC 读取目录树的速度就快得多,同时进入子目录也只需把光标移到子目录名上,然后一按回车即可,而不必象 Pctools 那样,每次改变都要查找甚至重读目录树,节省了大量时间。

三、良好的 DOS 界面。当屏幕左下角有光标闪烁时,可直接输入 DOS 命令或者运行文件名,如 DIR, TYPE, BASIC 等。但此时应保证在 NC 的同一目录中或 A 盘或 C 盘根目录中有 COMMAND. COM 文件,另外要保证运行文件的存在,否则 NC 将给出出错提示。(若事先用 PATH 命令指明了 COMMAND. COM 的路径亦可。)

四、灵活的用户自定义功能,这也是 NC 最独到之处。NC. MNU 是一个由用户定义功能键的文件。其基本格式为:

功能键名:注释  
命令

我们可以用 NC 的 EDIT 功能或 COMMAND 菜单中的 Menu file edit 功能编辑一个如下的 NC. MNU 文件:

D: Directory of drive A

dir a: /p

F1: Pctools  
c:\tools\pc  
F2: Personal Edit II  
c:\tools\pe2  
F3: Norton Utilities  
d:\ni\nu

编辑完毕,存盘退出,再按 F2 键进入 USER 菜单,即可使用自定义的功能键。此文件的含义为:按“D”键则列 A 盘目录,按“F1”键则运行 C 盘 TOOLS 目录中的 Pctools,余类推。应注意,用户自定义的功能键只有在进入 USER 菜单以后才有效。

NC. EXT 是另一个由用户自定义的文件,它的作用在于告诉 NC 如何处理非执行文件。在使用 NC 时,当把光标移到一个可执行文件上时,一按回车,NC 就自动执行该文件。当把光标移到一个非执行文件上时,一按回车,NC 就查找 NC. EXT 文件,根据这一类非执行文件在 NC. EXT 中的定义进行处理,其基本格式为:

后缀:命令(运行文件名)!. 后缀(当命令不要求时可省略)

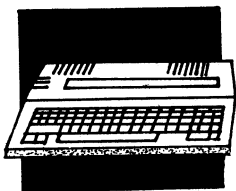
我们可以利用 COMMAND 菜单中的 extension file edit 功能编辑这样一个 NC. EXT 文件:

BAK: TYPE !. BAK  
BAS: C:\BAS\GWBASIC !  
PRG: C:\DBA\DBASE!  
\*: ren no runfile for this extension

编辑完存盘退出,就可使 NC 按这一定义进行处理。其含义为:当文件为 .BAK 文件时,则显示其内容;为 .BAS 文件时,则调用 C 盘 BAS 目录下的 GWBASIC 运行此文件;为 .PRG 文件时,则调用 C 盘 DBA 目录下的 DBASE 运行此文件;为其它未被定义的后缀时,则显示“rem no runfile for this extension”。

需要说明的是:如果用 NC 的 EDIT 功能编辑 NC. EXT 文件,在编辑完存盘后,必需先退出 NC,然后重新进入 NC,才能使 NC 按新的定义处理,否则将仍然按照原来的定义执行。

另外,对于 NC 的显示方式,用户可根据自己的需要或习惯进行设置,设置完毕后,利用 NC 的 OPTION 菜单中 Save setup 功能将其存入 NC. INI,下次使用 NC 时,即可直接按新设置显示。



## 学习机之友

# 为无驱动器系统设置虚拟磁盘

刘庆丰

目前与苹果机兼容的各类学习机已大量进入家庭。但不少用户均未配置驱动器,只用录音机作外存,这种结构使全机的性能大为降低,且使大量在 DOS3.3 系统下开发的软件均不能运行。但这类学习机均有扩充 16k 存储器,完全可以用软件的方式将其生成为一台虚拟磁盘机。

下面即是我参考有关资料编写的虚拟磁盘程序。首次使用前先将带有驱动器机子上录下的 \$9D00.\$BFFF 间的数据读入内存,再键入或读入程序一,并继续键入:B7B7:20 9A BE NADC0:20 61 BD NB3AF:BA E5 E5 F2 C6 A0 D3 CF C4 AD CE D8 N9CFD:4C 00 BD.将 \$9CFD~\$BFFF 间的数据录下来,就成为一盒“虚拟磁盘引导带”了。

用法:将此虚拟磁盘引导程序读入内存并从首址运行,屏幕将显示“DOS-XNP W:BDAD”等字符,此时整机已在 DOS3.3 系统管理下工作,可以使用所有的 DOS 命令了,使用方法与效果和驱动器上完全相同且速度快得多。本系统为用户提供了 60 扇区(15k)容量(每次 CATALOG 时均会自动显示剩余扇区数值),共可容纳 21 个文件。虚拟盘储满后将显示 DISK FULL,此时及关机前若需将“盘”内信息脱机保存,可键入 CALL48557 或 BDADG,屏幕将显示 BSAVE “DOS”,A \$3FFD,L \$4057 字样。移动光标划过这些字符并回车,同时启动录音机,就将全“盘”内容整体存入了磁带。这个命令是为小蜜蜂(XMF-I)学习机设置的,读回磁带时只需使用 BRUN“XNP”命令就可将所有信息重新装入虚拟磁盘。对于 CEC 等其他苹果兼容机,只需对程序作如下改动就可在监控状态下存带,读回时改用 3FFD.8054R 命令并从首址运行:

BE75: A0 B3 C6 C6 C4 AE B8 B0 B5 B4 D7 A0

程序一

BD00-20 58 FC A2 18 BD 21 BD  
BD08- 9D 05 04 CA D0 F7 A2 25  
BD10- BD 3A BD 9D CF 03 CA D0  
BD18- F7 20 51 A8 20 D4 A7 4C  
BD20- 19 BF AA AA AA A0 D8 CE  
BD28- D0 AD C4 CF D3 A0 AA AA  
BD30- AA A0 A0 A0 D7 BA C2 C4  
BD38- C1 C4 A0 4C BF 9D 4C 84  
BD40- 9D 4C FD AA 4C B5 B7 AD  
BD48- 0F 9D AC 0E 9D 60 AD C2  
BD50- AA AC C1 AA 60 4C 51 A8  
BD58- EA EA 4C 59 FA BF 9D 38  
BD60- 00 A9 00 85 40 20 86 BD  
BD68- 18 B9 F2 B3 F0 0E 0A 90  
BD70- FB 48 E6 40 D0 02 E6 41  
BD78- 68 18 90 F0 88 D0 E9 A6

BD80- 40 A5 41 4C 24 ED 85 41  
BD88- AD EF B3 0A 0A 85 06 A4  
BD90- 06 60 B1 3C 91 42 E6 42  
BD98- D0 02 E6 43 A5 3C C5 3E  
BDA0- A5 3D E5 3F E6 3C D0 02  
BDA8- E6 3D 90 E6 60 A9 00 85  
BDB0- 42 85 3C A9 40 85 43 A9  
BDB8- D0 85 3D A9 FF 85 3E 85  
BDC0- 3F A0 00 2C 80 C0 20 92  
BDC8- DD 2C 88 C0 A9 00 85 42  
BDD0- 85 3C A9 70 85 43 A9 D0  
BDD8- 85 3C A9 FF 85 3E A9 DF  
BDE0- 85 3F A0 00 20 92 BD 2C  
BDE8- 82 C0 A2 00 BD 75 BE 20  
BDF0- F0 FD E8 E0 19 90 F5 A9  
BDF8- 00 85 42 A9 80 85 43 A9  
BE00- 22 85 3C A9 BE 85 3D 85

BE08- 3F A9 74 85 3E A0 00 20  
BE10- 2C FE A9 4C 8D FD 3F A9  
BE18- 00 8D FE 3F A9 80 8D FF  
BE20- 3F 60 8D 81 C0 8D 81 C0  
BE28- A9 00 85 42 85 3C A9 D0  
BE30- 85 43 A9 40 85 3D A9 FF  
BE38- 85 3E A9 6F 85 3F A0 00  
BE40- 20 2C FE 8D 89 C0 8D 89  
BE48- C0 A9 00 85 42 85 3C A9  
BE50- D0 85 43 A9 70 85 3D A9  
BE58- FF 85 3E A9 7F 85 3F A0  
BE60- 00 20 2C FE 8D 82 C0 A2  
BE68- 00 BD 8F BE 20 F0 FD E8  
BE70- E0 0A 90 F5 60 A0 C2 D3  
BE78- C1 D6 C5 A2 D8 CE D0 A2  
BE80- AC C1 A4 B3 C6 C6 C4 AC  
BE88- CC A4 B4 B0 B5 B7 A0 D2

NBDF4:0C

有了这套虚拟磁盘系统,在很大程度上相当于配置了一台驱动器,不仅可进行各种顺序、随机文件的实际操作,更重要的是大量在 DOS 系统下开发的软件或操作系统,几乎都能够使用了。比如 DOS3.3 系统主盘上的一些非常有用的工具软件,如 RENUMBER(更改行号及程序合并)、CHAIN(程序链接)等,都可将其存放于虚拟盘中,供随时调用。若软件占内存较大,虚拟盘无法容纳时,还可只将中途需读盘的部分存于虚拟磁盘内,而将主体程序仍存于内存,这样能运行的软件范围就很广泛了。

由于本系统并不存在一个实际的“磁盘”,所以在同别人交换软件时仍需通过磁带进行。至于如何将 A 类 BASIC 文件和 B 类二进制文件存入磁带,相信大家都已很熟悉,但存储 T 型文本文件的方式却很少有人提到,而 T 型文件在 DOS 系统中又是一类非常有价值的文件。这里给出两段小程序,程序二可将 T 型顺序文件变成内存中的二进制信息并给出首末地址,以便按 B 类方式存储,程序三则将内存中的二进制信息重新按指定的文件名生成 T 型文件。

使用本系统还需注意以下几点:1. 由于 16k 卡已为虚拟磁盘所占用,所以如整数 BASIC、MIT LOGO 等一些需占用 16K 卡的系统将不能载入,DOS 的 INT 命令也不能使用,否则将显示 LANGUAGE NOT AVAILABLE 信息;2. 从首址启动及 INIT 命令都将使虚拟盘全部清零,所以使用这两个命令一定要慎重,当然也可利用这种特性来快速清“盘”;3. 本程序存放于 DOS 系统的 RWTS 区,不另占任何空间,故对各种系统的运行均无任何影响。但有个别软件运行中直接调用 RWTS 来驱动磁盘,在本系统中运行将有一定困难。



```

BE90- C1 CD AD D8 CE D0 A0 A1
BE98- A1 00 84 48 85 49 AD F4
BEA0- B7 F0 58 C9 04 D0 03 4C
BEA8- 0C BF 48 AD EC B7 C9 13
BEB0- 90 04 E9 04 B0 F8 C9 0F
BEB8- B0 04 69 04 90 F8 8D 78
BEC0- 04 E9 0F AA AD F0 B7 85
BEC8- 3E AD F1 B7 85 3F BC 76
BED0- BF B9 83 C0 B9 83 C0 BD
BED8- 7A BF 0D ED B7 85 47 A9
BEE0- 00 85 46 68 4A B0 0B A0

```

#### 程序二

```

10 FOR I=768 TO 781:READ A;POKE I,A;NEXT DA-
TA 230,6,208,2,230,7,96,165,7,166,6,76,65,249
20 ONERR GOTO 100
30 POKE 6,0; POKE 7,32;D$=CHR$(4)
40 INPUT "NAME? ";A$
50 PRINT D$ "OPEN" A$
60 PRINT D$ "READ" A$
70 GET B$;B=ASC(B$)+128
80 GOSUB 150;GOTO 70
100 PRINT D$ "CLOSE"
110 B=0;GOSUB 150

```

```

BEE8- 00 B1 3E 91 46 88 D0 F9
BEF0- F0 09 A0 00 B1 46 91 3E
BEF8- 88 D0 F9 2C 82 C0 AD 75
BF00- BF 8D F7 B7 AD 97 BF 8D
BF08- F6 B7 18 60 20 19 BF A9
BF10- 00 85 48 20 EA A2 4C 8B
BF18- 9F A9 FF 85 47 A0 00 84
BF20- 46 98 2C 83 C0 2C 83 C0
BF28- 91 46 2C 8B C0 91 46 88
BF30- D0 F3 C6 47 A5 47 C9 D0
BF38- B0 E7 A0 0D B9 7E BF 99

```

```

BF40- 74 E0 88 10 F7 A0 07 B9
BF48- 8C BF 99 30 E0 88 10 F7
BF50- A9 7A 8D 27 E0 A0 03 B9
BF58- 94 BF 99 00 E0 88 10 F7
BF60- A9 11 A0 02 8D 01 E3 8C
BF68- 02 E3 88 8D 01 E2 8C 02
BF70- E2 2C 82 C0 60 60 00 08
BF78- 08 08 D0 D0 E0 F0 FF FF
BF80- 00 00 FF FF 00 00 FF F0
BF88- 00 00 FF FF 11 01 00 00
BF90- 23 10 00 01 04 11 03 00

```

```
120 PRINT " $ 2000- $ ";:CALL 775
```

```
130 END
```

```
150 POKE ( PEEK(6)+PEEK(7) * 256),B;CALL 768;
```

RETURN

#### 程序三

```

10 D$=CHR$(4);A=8192
20 INPUT "NAME? ";A$
30 PRINT D$ "OPEN" A$
40 PRINT D$ "WRITE" A$
50 B=PEEK(A);A=A+1;IF B=0 THEN 80
60 PRINT CHR$(B);:GOTO 50
80 PRINT D$ "CLOSE"

```

## 巧用 LOGO 解难题

四川 廖大勇

一九八六年全国青少年计算机程序设计竞赛题中,有一道“四狗互追”的难题,原题如下:

有一个正方形场地,在四个角上有四条猎犬,分别为 A,B,C,D,鸣枪之后它们同时起跑;并以相同速率 A 追 B,B 追 C,C 追 D,D 追 A,试编程在屏幕上显示它们跑出的轨迹(即跑过的路线)。由于 BASIC 语言无递归功能,编写的程序都要牵涉到较多较深的数学知识,难度较大,不易为学生理解掌握。

用 LOGO 语言的递归功能来解此题,只需两个简单的过程,简洁明了,直观易读,充分体现出了 LOGO 语言简单易懂、化难为易的神奇本领!程序如下:

```

TO DOG
ASPECT 1 MAKE "A 120 PU FD 60 LT 90 FD 60 PD RE-
PEAT 4 (LT 90 FD :A) RT 180 HT DOG1 :A
TO DOG1 :A
IF :A<3 THEN STOP
REPEAT 4 (FD 3 PU FD :A-3 PD RT 90) FD 3
RT (ATAN 3 :A-3) DOG1 SQRT(:A-3) * (:A-3)+3
END

```

## 修改 CEC 汉字系统程序

四川 泸州二中 周易

CEC-I 中华学习机的汉字系统管理程序被固化在辅存 ROM 的 \$EC00~\$FFFF 一段地址中,无法用监控命令 L 进行查看,更无法对其进行修改,使 CEC 机汉字系统的扩展很不方便。

下面这段机器语言程序专门为有志于 CEC 汉字系统扩展的微机爱好者而设计,它将辅存 ROM 中的汉字管理程序移到了 RAM 内存中的 \$2C00~3FFF 这段地址中,并将其中的绝对地址改成了相对于 2C00 的地址,使用户更容易使用。现在不仅可查看它,更可以大胆地修改调用它为自己所用了。

利用这个方法,可使 CEC 具有同 STC 汉字系统相似的汉字处理功能,还可配合 F<sub>4</sub>,F<sub>5</sub> 键的扩展功能,使 CEC 汉字系统得到进一步的强化和扩充。

运行方法:在监控下键入 21CDG 即可。

```

21CD- A9 00 85
21D0- 06 85 08 A9 EC 85 07 A9
21D8- 2C 85 09 8D 0B C0 8D 06
21E0- C0 20 AB C3 20 ED 21 20
21E8- B9 C3 4C 00 22 A0 00 A2
21F0- EC B1 06 91 08 C8 D0 F9
21F8- E6 C7 E6 09 E8 D0 F2 60
2200- A9 00 85 06 A9 2C A6 EC
2208- A0 00 B1 06 C9 20 F0 0D

```

2210- C9 4C F0 09 C8 D0 F3 E6  
 2218- 07 E8 D0 EE 60 A5 06 48  
 2220- A5 07 48 98 48 8A 48 C8  
 2228- D0 06 E6 07 E8 D0 01 60  
 2230- C8 D0 06 E6 07 E8 D0 01

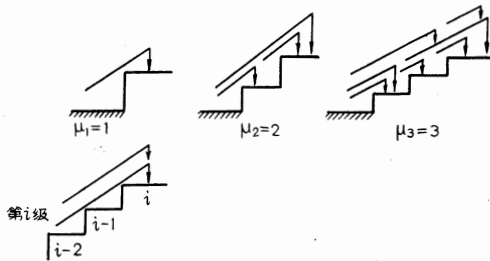
2238- 60 B1 06 C9 EC 90 0B 29  
 2240- 3F 91 06 68 68 68 68 4C  
 2248- 14 22 68 AA 68 A8 68 85  
 2250- 07 68 85 06 4C 14 22

## 浅谈计算机在中专数学中的应用

无锡无线电工业学校 王志平 顾正刚

在中等专业学校,如果将计算机课程的学习穿插在一些基础课教学中,尤其是有机地与数学结合起来,可增加它的趣味性和实用性。

例如在学习 BASIC 语言时,可以结合数学中一些较为繁琐和头痛的问题来讲解。这样,既可巩固所学的计算机知识,又可使学生感到学有所用,从而进一步激发学习热情。下面举两个简单的例子。



(一)在加深理解变量的概念的时候,希望学生学会使用变量循环更新的方法,我们就结合了中专数学中计算数列某项这一繁琐的问题。为了增加它的趣味性,设置了如下例题:

某数学教师上课时,问一同学他每次上楼梯一步跨几个台阶?同学回答:有时一步跨一个台阶,有时一步两个台阶。老师又问:按这种走法,从一楼走到三楼一共40级台阶,可有多少种方式走到三楼?学生哑口无言。这时老师就分析了这一题目:

跨上第*i*级只能有两种方式,即从第*i*-1级跨一步;从第*i*-2级跨二步。因此, $\mu_i = \mu_{i-1} + \mu_{i-2}$ 。这样,实际得到了如下一个数列:

1, 2, 3, 5, 8, ...

该数列从第三项起,每一项都等于前两项之和。这一问题就是要求出该数列的第40项。这是件令学生头痛的问题。下面利用计算机分析解答。

设置两个变量A和B,开始令A=1, B=2。采用变量更新A=A+B, B=B+A。更新一次后,A和B的值分别为3和5,正好是数列的第三、四项。如此更新下去可得数列的各项。因此,本问题的程序如下:

```
5 LET A=1;B=2
10 FOR I=1 TO 19
20 LET A=A+B;LET B=B+A
```

```
25 NEXT I
30 PRINT "THE WAY=";B
35 END
RUN
THE WAY=1.6558E+08
```

(二)在线性方程组一章中,为了学会线性方程组的解法,结合中专教材《数学》第四册,可以讲解用高斯消元法解线性方程组的程序。

[例]用高斯消元法解联立方程组:

$$\begin{aligned} 3X + 2Y + Z &= 14 \\ X + Y + Z &= 10 \\ 2X + 3Y - Z &= 1 \end{aligned}$$

在这里可与同学们讲解高斯消元法的程序:

```
10 DIM A(3,4)
20 FOR I=1 TO 3;FOR J=1 TO 4
40 READ A(I,J)
50 NEXT J;NEXT I
70 FOR J=1 TO 3
80 IF A(J,J)<>0 THEN 150
90 FOR I=J TO 3
100 IF A(I,J)<>0 THEN 140
110 NEXT I
120 PRINT "No unique solution or no solution"
130 GOTO 610
140 GOSUB 300
150 LET C=1/A(J,J);GOSUB 400
170 FOR I=1 TO 3
180 IF I=J THEN 210
190 LET C=-A(I,J);GOSUB 500
210 NEXT I
220 NEXT J
230 FOR I=1 TO 3
240 PRINT "X(";I;")=";A(I,4)
250 NEXT I
260 GOTO 610
300 FOR K=J TO 4
310 LET B=A(J,K);A(J,K)=A(I,K);A(I,K)=B
340 NEXT K;RETURN
400 FOR I=J TO 4
410 LET A(J,I)=A(J,I)*C
420 NEXT I;RETURN
500 FOR S=J TO 4
510 A(I,S)=A(I,S)+C*A(J,S)
```

```

520 NEXT S;RETURN
600 DATA 3,2,1,14,1,1,1,10,2,3,-1,1
610 END
RUN✓
X(1)=1
X(2)=2
X(3)=7

```

对于任意元线性方程组都可以用以上程序求解,只需将数组 A(3,4)及相应的 FOR-NEXT 语句中循环终值修改为合适的数值即可。

通过以上的简单例子,我们体会到将计算机和数学有关内容有机地结合在一起,一定可以收到良好的教学效果。实际上,也可以将中专的计算机教学渗透到其它一些学科中去,这样可以使学生学会利用计算机解决一些实际问题,为走上工作岗位后普及应用计算机打下良好的基础。

## 数据块的后移

成都商业学校 冯端品

在监控系统中有个移动数据的 M 命令,用它把数据块向前移动是没有任何问题的,但是用它把数据块向后移动,当源地址和目标地址发生交叉时,会丢失数据。分析 M 命令的执行过程,原来它在移动数据块时,先移第 1 个数据,然后移第 2 个数据,……直到最后一个数据。这样向前移是对的,向后移就可能发生问题。那么我们只要想办法在向后移数据块时,把次序反过来,先移最后 1 个数据,然后移倒数第 2 个数据……直到第 1 个数据,就不会发生问题了。

我用机器语言编了一段程序,解决了把数据块向后移动的问题。为了应用方便,程序地址是浮动的。

把程序调入内存任何方便的位置,然后把程序的开始地址送入 \$3F9、\$3FA 单元,低位在前,高位在后。当需要把数据块向后移动时,在监控状态下输入

目标地址<源数据块首址 末址 CTRL-Y✓

```

0300- A5 3C 85 FA A5 3D 85 FB
0308- A5 3F 85 FD A5 3E 85 FC
0310- 38 E5 FA 85 08 A5 FD E5
0318- FB 85 09 A5 44 18 65 08
0320- 85 08 A5 45 65 09 85 09
0328- A0 00 B1 FC 91 08 A5 FC
0330- C5 FA D0 07 A5 FD C5 FB
0338- D0 01 60 A6 FC D0 02 C6
0340- FD C6 FC A6 08 D0 02 C6
0348- 09 C6 08 D0 DD F0 DB

```

## 从 QZC 卡读取汉字字模

北京机械工程学院附中 牛海峰

QZC 汉卡的功能很强,加装后许多指标已经略高 CEC 机一筹。它还留了两个语句,使用户自行定义其功能。这两个语句分别是 KEY 和 PLAY,开机初始化后被定义为:KEY 是追回刚被 NEW 掉的程序;PLAY 是用来在 PST 表内查找某行程序所在地址。下面介绍用 PLAY 语句显示出汉字字模的方法

用 MON A9000✓进入汇编过程,键入以下程序:

```

9000- CALL 2B0E
      PUSH HL
      LD A,(7894)
      CALL 4E6C
      LD HL,7AB2
      LD B,20
900F- LD A,(HL)
      CALL 5080
      INC HL
      DJNZ 900F
      POP HL
      RET

```

然后修改 PLAY 的指针,执行:MONW7991;C3,00,90✓指针即修改完毕。进入高显状态,执行 PLAY 16,01✓即看到显示出:“啊”,然后显示啊的点阵字模。扩展后的 PLAY 语句也可用在程序中。

我在使用 QZC 卡编程中陆续发现了一些该卡的机器语言子程序入口,现公布如下:485CH:进入 SC 状态,使用寄存器 A、B。程序中利用 HALT 命令防止闪烁,原理是在场消隐期间改变状态。

5080H:把 A 中的二进制数换成 ASCII 字符印出

5189H:设原 A 寄存器存有 mm,L 寄存器存有 nn,本子程序把 A 和 L 的值以“mmnn-”的形式印出。

4E12H:进行 SC 的状态切换。(SC1↔SC0)

40CCH:把 ASCII 代码串换算成数值,入口时 HL 指向代码串首,处理后 HL 指向串尾+1,结果存于 WRA1 和 DE 中。

4E6CH:汉字输出。入口时,A 内存入区码值,E 内存入位码,调用后,对应汉字将打印在当前位置上。

48A1H:把对应汉字字模从汉卡读出,送入以 7AB2H 开始的缓冲区中,区长 32 字节,A、E 同上。

2B0EH:此程序是在原驻机系统中的,用户用来扩展语句非常方便,功能是专用于测试:“数据①,数据②”形式的代码串,并将数据①存于 7894 和 7897 中,数据②存于 A、E 中。



# ProDOS 磁盘操作系统入门

廖 凯

## 前 言

目前中华学习机已经普及,广大用户对 DOS3.3 已比较熟悉,而对 ProDOS 磁盘操作系统却很陌生。我应编辑部的要求,将向大家系统地介绍 ProDOS 系统,使大家能够迅速地掌握 ProDOS 系统。此介绍只适合于中华学习、APPLE 机及其兼容机的用户。

## 第一章 ProDOS 简介

### 一 ProDOS 的功能

ProDOS (Professional Disk Operating System 专业磁盘操作系统)是 1984 年由苹果公司为克服 DOS3.3 所存在的缺陷而设计的一种高级磁盘操作系统。它是目前 APPLE 机功能最强的磁盘操作系统,它不仅包括了 DOS3.3 的一切功能,而且还具有以下功能:

1. 快速磁盘存取——8K 字节/秒,是 DOS3.3 的 8 倍。
2. 大容量存储——支持磁盘容量可达到 32M 字节,每一文件容量可达到 16M 字节。
3. 多层次文件结构——使用与 UNIX 相同的树形分层目录结构。
4. 文件类型可达 256 种,包括用户自定义类型。
5. 支持时钟/计时卡,自动给文件加上日期和时间。
6. 加强的 BASIC 指令。
7. 提供用户可改变的四个中断进入点。
8. 方便的 MLI (机器语言接口)调用。
9. 支持 RAM 磁盘。
10. 与 APPLE III 的 SOS 操作系统相兼容的文件结构。

### 二 ProDOS 的运行条件

ProDOS 系统必须具备以下条件才能真正运行:

1. 驻机 ROM 中必须是 APPLESOFT BASIC 语言解释程序。
2. 至少有 64K RAM 的内存。
- \* 3. 必须是 APPLE 公司的原装机。

注:这是指原版 ProDOS 系统盘而言,如果用户使用的是中华学习机或苹果兼容机,可参阅《电子与电脑》1991 年第 6 期的《ProDOS 系统盘的修改》一文,将 ProDOS 进行修改。修改后的 ProDOS 系统盘可在满足前二个条件的任何兼容机上运行。

### 三 ProDOS 的结构

ProDOS 系统经核心程序及解释程序提供有效的存储管理和中断处理。这核心程序即 MLI (Machine Language Interface 机器语言接口),它包含四个部分:

1. 命令处理程序:由解释程序接收信息并确定所

使用的 ProDOS 命令类型,如果命令有效,那么处理程序发送命令到操作系统内的另一部分,如果命令无效,则得到一个错误信息。

2. 分层文件管理程序:由 MLI 执行命令的发送、记载引导磁盘的磁道、发送数据到磁盘驱动程序、处理打开的文件及进行磁盘的管理。分层文件管理程序按逻辑次序在磁盘上存储数据。

3. 磁盘驱动程序:通过磁盘驱动设备进行数据读写,其磁盘驱动设备有 DISK II, DuoDISK 和 RAM 磁盘。

4. 中断处理程序:ProDOS 允许将四级中断处理程序同时装入内存,并按优先级予以响应,独立工作。这给联网和通信以及实时控制创造了有利条件。

ProDOS 系统盘在市场上有两种版本:ProDOS USERS. DISK V1. 0. 1 及 ProDOS USERS. DISK V1. 1. 1

在系统盘上主要有四个系统文件,它们是:  
1. PRODOS 是 ProDOS 系统的核心,它用于控制计算机与输入/输出(I/O)设备信息的流程,并经 BASIC 解释程序与用户沟通。

2. BASIC. SYSTEM 是磁盘操作命令解释程序,它提供加强的 BASIC 命令。BASIC. SYSTEM 解释程序根据缓冲区空间的自动分配情况,提供存储管理,随着文件的打开与关闭,它将自动向上或向下移动用户可用的内存地址上界 HIMEM。

3. FILER 是磁盘及文件管理程序。它用于磁盘或文件的拷贝、比较以及磁盘的格式化和文件的上锁、改名等操作。

4. CONVERT 是转换程序,它可将文件在 ProDOS 和 DOS3.3 之间进行转换。

### 四 ProDOS 的启动

ProDOS 系统盘具有自启动功能。当你将系统盘插入驱动器后,开机即可进入主菜单。或者你可以使用 PR# 或 IN# 命令来启动系统盘。在启动系统盘时,ProDOS 首先将 PRODOS 文件装入内存(首地址为 \$2000)并执行。MLI 确定计算机存储容量并移动 MLI 到存储器较高的位置,而后确定所有在槽口上的设备并建立系统整体页面。随后 MLI 将 BASIC. SYSTEM 文件装入内存(首地址为 \$2000)并执行。BASIC. SYSTEM 于是移动自己本身并更新系统整体页面。系统整体页面是解释程序与操作系统之间的通信环节。在系统整体页面内部是一个系统位元图(bitmap),它记载存储器从 \$0000 到 \$BFFF 各页的状态。解释程序决定各页是否显示或使用。在一个 64K RAM 的计算机上,ProDOS 系统占用内存 25K 字节,留给用户 35.5K 字节用于编程。

## 第二章 文件目录结构

ProDOS 使用多级目录结构,这给用户带来很多方便,它可以使用户按任务的不同层次、不同领域,建立多层次的分目录。如学校由校、系、年级、班级四层次组成,一个学校可以按其层次组织方式,采用多级目录结构形式,方便管理其所属的全部文件,这种多层次的树形结构有一个根目录,它是整个目录结构的基础。在根目录下,既可以是目录文件,也可以是普通文件。

ProDOS 这种分层文件结构,对于存储容量大的磁盘(例如 5M 温式硬盘)特别有用,而它在 5 寸软磁盘上用途则有限,因为这种软磁盘容量不大。

### 一 文件目录

每个 ProDOS 磁盘都有一个卷名,是在磁盘被格式化时所给予的。在列磁盘目录和进入主菜单时,磁盘卷名会被显示出来。在根目录下共可存放 51 个文件。

DOS3.3 采用的是一级目录结构。所谓一级目录结构是指把系统中的所有文件都建立在一张目录表中,结构比较简单。但是一级目录结构存在不少的缺点,首先如果系统中有很多文件,要查找一个文件需要扫描整个目录表,使得查找目录时间增加;其次是不方便用户对文件的访问。一级目录结构的文件系统规定,用户不能给文件起相同的名字,给多用户带来很大的不便,因此在多用户的情况下,采用二级目录结构较为方便。二级目录结构是由一个主目录和它所管理的若干子目录组成。文件系统采用二级或多级目录结构后,就能够为用户提供文件可以同名和一个文件可以有多个不同名字的这种便利。

现在让我们看一个 ProDOS 使用子目录的例子:

```
/DATA
  NAME      TYPE      BLOCK  MODIFIED
SPREADSHEET DIR        1      NO DATE
WORD. PROCESSOR DIR    1      NO DATE
DATA. BASE    DIR        1      NO DATE
```

这磁盘的卷名为 DATA,用子目录把资料分为 SPREADSHEET,WORD. PROCESSOR 和 DATA. BASE。我们可以在 SPREADSHEET 目录内存放空白表格文件。其它两个子目录也同样,你可以由 TYPE 项目得知这些文件是目录文件(类型为 DIR)。此时你可能需要一个字处理文件的列表,并找三个名为 CORP. POLICY,FRED 和 GENERAL. DATA 的文件,如下:

```
/DATA/WORD. PROCESSOR
  NAME      TYPE      BLOCK  MODIFIED
CORP. POLICY TXT        6      NO DATE
FRED        TXT        2      NO DATE
GENERAL. DATA TXT      4      NO DATE
```

要在目录结构内观看其它文件,你必须按顺序给出正确的文件名。在上面的例子中,我们通知 ProDOS 寻找名为 DATA 的卷目录和名为 WORD. PROCESSOR 的子目录,计算机在

WORD. PROCESSOR 目录内读取指定文件并显示出来。

注意在上述的例子(/DATA)中,文本文件并未列出,列出的只是子目录,这是因为 ProDOS 一次只能显示一级目录。在这种情况下,它是目录结构的第一层。

### 二 路径名(PATHNAME)

当用户要访问多级目录中的某个文件时,需要用该文件的“路径名”来标识文件。所谓文件的路径名是指从主目录(根目录)出发,一直到所要找的文件,把途径的各目录名连接一起而形成的。每个目录名之间必须用“/”分开。例如:

```
/DATA/WORD. PROCESSOR/FRED
```

路径名前面的第一个“/”代表主目录(根目录)。该例子表示 ProDOS 首先寻找卷目录 DATA,然后找子目录 WORD. PROCESSOR,最后找到文件 FRED。如果你输入的路径名不对或文件名不在这儿,那么将会显示:PATH NOT FOUND。

### 三 部首(PREFIX)

在许多实例中,你要存取一个多级目录的文件时,需要连续打入路径名,这很麻烦而又费时。由于这个缘故,所以 ProDOS 提供一个部首命令 PREFIX。命令 PREFIX 用于定义一个路径名(PATHNAME)。当使用相同的路径名时,我们可以用 PREFIX 命令定义路径名:PREFIX/DATA/WORD. PROCESSOR/

这允许我们在 WORD. PROCESSOR 子目录内存取任何文件,除输入整个路径名的情况外。

例如我们要访问 FRED 文件,只要输入 FRED 就可以了。换句话说,在使用 PREFIX 以后,我们只要输入文件名即可。

注意在输入的文件名前面没有“/”。

如果在 WORD. PROCESSOR 目录中加入一个子目录 CONTRACTS,我们仍能在部首下存取所有文件或使用一个部分路径名,例如 PREFIX 仍设为/DATA/WORD. PROCESSOR/,要从 CONTRACTS 子目录中访问 LABOR 文件,我们就需输入:CONTRACTS/LABOR

注意在子目录 CONTRACTS 前面没有“/”,如果再加一个“/”,那么路径名将为/DATA/WORD. PROCESSOR//CONTRACTS,这是错误的,即使在设置部首时后面未附“/”,ProDOS 也会自动加上。

在存取一个磁盘或文件时,ProDOS 只寻找卷名而不是驱动器号,这在写数据到数个驱动器时会很有用。不过 ProDOS 仍保留 DOS3.3 的存取方式。

### 四 目录名和文件名的限制

ProDOS 要求一个目录名和文件名必须以字母开头,并且长度不能超过 15 个字符。文件名和目录名内的字符必须是相连的。ProDOS 将允许在一个文件名和目录名内使用大小写字母、数字和一个句号。

ProDOS 路径名的最长为 64 个字符,包括斜线。

注意:你不能 RUN 或执行目录文件。(未完待续)

## 在 APPLESOFT 系统下 实现 HSCRN 功能

四川绵阳师专附中 李齐

在 CP/M 操作系统支持下运行的 GBASIC 语言为高分辨作图提供了一条 HSCRN 语句,它可以检查屏幕上的点是否显示。这对我们编写动画和游戏程序无疑是一大帮助。

可是在无 CP/M 操作系统下能否实现 HSCRN 功能呢?我们知道,APPLE-Ⅱ 计算机的高分辨率图形方式有两页。它们的内存区分别是 \$2000—\$3FFF 和 \$4000—\$5FFF。下面我们主要就第一页进行讨论。

高分辨率图形的内存区中的一个字节对应屏幕上的一个显示单元,一个显示单元含有七个点。字节的低七位同七个点一一对应。如果某一点显示。那么这个点的对应位为1,否则为0。字节的高位是彩色代码,我们这里不必深究。

另外还需要注意,显示单元与内存单元的地址排列对应关系并不是完全按照顺序的。内存单元的内容在显示时的次序是把屏幕分成三个部分,然后依次填入。所以不能直接用 X、Y 坐标得出与显示单元对应的内存单元,但可以用下面的公式求出。

$$AD = 8192 + [\text{INT}(Y/8) \bmod 8] \times 128 + \text{INT}(Y/$$

$$64) \times 40 + (Y \bmod 8) \times 1024 + \text{INT}(X/7)$$

可以看出,要实现 HSCRN 功能,必须进行二进制数处理和地址计算。由于汇编语言处理二进制数的功能较强;而用 BASIC 语言进行复杂计算较为方便,所以采用汇编语言和 BASIC 语言相结合的方式。

以下是程序清单。读者可先将程序1输入内存,程序必须从 \$0300 开始装入,而将程序2做为子程序放在自己的程序中,需要时调用。AD 是点的显示标志,程序运行后 AD 的值如为1则表示显示,如为0则表示未显示。所以通过 AD 的值就可以看出点是否显示,从而实现了 HSCRN 的功能。在调用子程序前,应先将需要判定的点的纵坐标和横坐标的值分别赋给变量 Y 和 X。不要忘了在程序2后面加上 RETURN。

程序1

```
0300-    AD  13  03          LDA  $0313
0303-    AE  14  13          LDX  $0314
0306-    F0  05          BEQ  $030D
0308-    4A          LSR
0309-    CA          DEX
030A-    4C  06  03          JMP  $0306
030D-    29  01          AND  # $01
030F-    8D  13  03          STA  $0313
0312-    60          RTS
```

程序2

```
1000  AD=8192+(INT(Y/8)-8*INT(Y/64))*128+
      INT(Y/64)*40+(Y-8*INT(Y/8))*1024+INT
      (X/7)
1010  AD=PEEK(AD);POKE 787,AD;AD=X-7*INT
      (X/7);POKE 788,AD;CALL 768;AD=PEEK(787)
```

## BASIC 语言中几个命令功能的扩展

郭殿秋

中华学习机和苹果Ⅱ微机(以下简称学习机)的 BASIC 语言中 GOTO, GOSUB, ON...GOTO, ON...GOSUB 以及 RESTORE 命令,可以通过一段短短的解释程序进一步扩展其功能。扩展功能之后,不影响原来命令的用法,不占用 BASIC 语言通常所用的空间,而且可以增加编程技巧,使之完成一些特殊的功能。本文旨在介绍扩展上述命令功能的实用方法。

### 1. 原解释程序的局限性

在学习机的浮点 BASIC 语言中, GOTO 语句是由 GOTO 命令和数值行号两部分构成的。由于它不能像整数 BASIC 语言那样用表达式表示行号,因此 GOTO 语句仅能实现绝对转移,而不存在使程序产生分枝的功能。GOSUB 命令也仅能调用以数值表示的、固定行号的子程序,这样就限制了程序的灵活性。

ON...GOTO(GOSUB)语句虽然可以实现程序的分枝,但是这种分枝仅仅是单结点的多枝树结构,分枝的数量增多表达起来不太方便,语句也较长。而且在

ON...GOTO(GOSUB)语句中作为判别分枝的开关变量的取值仅能是自然数。(例如 ON K GOTO...中的 K。)如果开关变量的值较为离散,那么就不得不再次为开关变量赋值,或是用空行号添补行号序列中与自然数序列不对应的部分。实际上这样的程序结构只要用 GOTO 命令和变量行号组成的语句就可以替代。

ON...GOTO(GOSUB)语句中的数值行号序列也可以扩展成为变量序列。这样通过开关变量、行号变量两级变化,可以构造出具有多结点、多分枝树和网结构的程序。同样,在条件转移语句 IF...THEN GOTO(GOSUB)语句中,用变量表示行号,可使该语句由条件单分枝结构(二叉树)变为条件多分枝结构。

在学习机的 BASIC 语言中,恢复数据命令 RESTORE 不能带参数。如果要想有选择地重读某些数据,势必要空读很多数据,且空读的次数不得有误。在一些微机中(例如 IBM-PC 机),RESTORE 命令可以同其后的行号组成语句,这样就可根据需要来设置数



据指针,这在分析处理数据的程序中具有重要意义。我们可以对此语句的功能进行扩展,完成与 IBMPC 机类似的功能;而且 RESTORE 命令后面的行号还可以用表达式来表示,这样就会给编程带来很大的方便。

## 2. 解释程序的建立及说明

扩展上述功能的解释程序共用56条汇编指令,占用第3页起始的119个内存单元;反汇编如下

```
0300- C9 B4      CMP      # $B4
0302- D0 2F      BNE      $0333
0304- 20 B1 00   JSR      $00B1
0307- 20 F8 E6   JSR      $E6F8
030A- C9 AF      CMP      # $AF
030C- F0 03      BEQ      $0311
030E- 4C EF D9   JMP      $D9EF
0311- 20 B1 00   JSR      $00B1
0314- 48         PHA
0315- A5 A1      LDA      $A1
0317- 85 06      STA      $06
0319- C6 06      DEC      $06
031B- D0 04      BNE      $0321
031D- 68         PLA
031E- 4C 33 03   JMP      $0333
0321- 20 B1 00   JSR      $00B1
0324- 20 7B DD   JSR      $DD7B
0327- 20 52 E7   JSR      $E752
032A- 20 B7 00   JSR      $00B7
032D- C9 2C      CMP      # $2C
032F- F0 E8      BEQ      $0319
0331- 68         PLA
0332- 60         RTS
0333- 48         PHA
0334- 20 B1 00   JSR      $00B1
0337- 20 7B DD   JSR      $DD7B
033A- 20 52 E7   JSR      $E752
033D- 68         PLA
033E- C9 B0      CMP      # $B0
0340- D0 1A      BNE      $035C
0342- A9 03      LDA      # $03
0344- 20 D6 D3   JSR      $D3D6
0347- A5 B9      LDA      $B9
0249- 48         PHA
034A- A5 B8      LDA      $B8
034C- 48         PHA
034D- A5 76      LDA      $76
034F- 48         PHA
0350- A5 75      LDA      $75
0352- 48         PHA
0353- A9 B0      LDA      # $B0
0355- 48         PHA
3056- 20 41 D9   JSR      $D941
0359- 4C D2 D7   JMP      $D7D2
035C- C9 AB      CMP      # $AB
035E- F0 14      BEQ      $0374
0360- C9 AE      CMP      # $AE
0362- F0 03      BEQ      $0367
```

```
0364- 4C 95 DD   JMP      $DD95
0367- 20 1A D6   JSR      $D61A
036A- A5 9B      LDA      $9B
036C- A4 9C      LDY      $9C
036E- 38         SEC
036F- E9 01      SBC      # $01
0371- 4C 50 D8   JMP      $D850
0374- 4C 41 D9   JMP      $D941
```

这段解释程序可利用 BASIC 语言的 & 命令,因此在使用前要将本段程序的首地址送入3F5等单元,即

```
03F5- 4C 00 03   JMP      $0300
```

这段解释程序可在监控状态下键入,亦可在汇编状态下键入。对监控及小汇编不很熟悉的读者,可用 BASIC 语言将其输入。程序如下:

```
400 FOR I = 1 TO 119:READ A
410 POKE 767+I,A:NEXT
420 POKE 1013,76:POKE 1014,0
430 POKE 1015,3:END
440 DATA 201,180,208,47,32,177,0,32
450 DATA 248,230,201,175,240,3,76,239
460 DATA 217,32,177,0,72,165,161,133
470 DATA 6,198,6,208,4,104,76,51
480 DATA 3,32,177,0,32,123,221,32
490 DATA 82,231,32,183,0,201,44,240
500 DATA 232,104,96,72,32,177,0,32
510 DATA 123,221,32,82,231,104,201,176
520 DATA 208,26,169,3,32,214,211,165
530 DATA 185,72,165,184,72,165,118,72
540 DATA 165,117,72,169,176,72,32,65
550 DATA 217,76,210,215,201,171,240,20
560 DATA 201,174,240,3,76,149,221,32
570 DATA 26,214,165,155,164,156,56,233
580 DATA 1,76,80,216,76,65,217
```

这段 BASIC 程序最好写在 HELLO 程序中,以便一开机就可获得扩展这些命令的功能。

## 3. 解释程序的语法及说明

我们附加的这段解释程序使用了 & 命令作为引导符。因此在程序中用到扩展的功能时,相应的语法如下:

```
& 'GOTO<表达式>
& GOSUB<表达式>
& ON<表达式>&GOTO<表达式>
& ON<表达式>&GOSUB<表达式>
& RESTORE<表达式>
```

注1. GOTO 和 GOSUB 后面的表达式可以是可表达行号的任何形式的变量、表达式或数值行号,如果采用数值行号则与原来的功能相同。如果变量为小数,那么将截去小数部分;如果变量对应的行号不存在,那么仍给出原错误提示。

注2. &RESTORE 命令后的表达式不可省略,恢复全部数据时仍用原命令或给出第一个 DATA 语句行号;表达式值无对应行号时,将设置该值后面的行。

注3. &ON...&GOTO(GOSUB)语句中的第二个 & 命令如果省略,那么后面只能用数值行号。

## 第六讲 宏指令的用法

李文兵

典型的 C 源程序的编译过程是:用预处理程序处理带 # 号的宏预处理指令(简称宏指令);进行语法分析,产生目标代码;进行代码优化处理。

本讲,我们介绍 C 宏指令的用法。

### 1. #define 与 #undef

(1) #define 这条宏指令叫宏定义。它用得比较多,也非常有用。其使用格式如下:

#define <标识符> <字符串>

这里出现的标识符也叫宏名, #define 的功能是:用字符串置换宏名。

```
A>type exp6-1.c
#define message "This is a message"
main()
{ printf(message); }
A>exp6-1
This is a message
```

宏名可以是带参数的宏

```
A>type exp6-2.c
#define dmax(x,y) (x>y)? x:y
main()
{ int a=3,b=2;
  printf("%d\n",dmax(a,b)); }
A>exp6-2
3
```

```
A>type exp6-3.c
#define begin {
#define end }
#define DMAX(x,y) x>y? x:y
main()
begin
double x=1000.0,y=200.0;
long lx=11,ly=21;
int i;
printf("d-max=%lf\n",DMAX(x,y));
printf("l-max=%ld\n",DMAX(lx,ly));
end
```

```
A>exp6-3
d-max=1000.000000
l-max=21
```

宏定义经预处理的结果叫宏展开,练习 6.3 的宏展开程序是

```
main()
{ double x=1000.0,y=200.0;
  long lx=11,ly=21;
  int i;
  printf("d-max=%lf\n",x>y? x:y);
  printf("l-max=%ld\n",lx>ly? lx:ly);
}
```

宏展开程序再经过语法分析,产生目标代码,代码优化,之后,经过链接程序 link 把程序的目标码和集中存放诸如 printf(),scanf() 等函数目标码的库链接起来,才使程序变为可执行程序。

带参数的宏的调用只是宏置换,因此,执行速度比较快;而预处理展开后的程序要占存储空间,故带参数的宏要比函数需要较多的存储空间。

宏名还可以嵌套使用,如练习 6.4 所示。

```
A>type exp6-4.c
#define a 10
#define b 20
#define c 30
#define v a*b*c
main()
{ printf("staiseki %d\n",v); }
```

```
A>exp6-4
staiseki 6000
```

使用宏定义应注意以下几点:

- ① # 前不能有空格或标识符。
- ② 宏指令不是 C 语句,故其末端不能有分号(;),否则,分号被看作是字符串的一部分。
- ③ 宏名不能用双引号引起来,否则不能置换。
- ④ 宏指令一般放在文件开头部分,且只在该文件内有效;出了这文件,该宏指令全失去作用。

### (2) #undef 取消宏定义,其用法是

#undef 标识符

其中标识符是由宏定义指令定义过的宏名。预处理程序检测到取消宏定义后,便使对应的宏定义失效。

```
A>type exp6-5.c
#define PAI 3.14159
main()
{ float r,l,a,v;
  r=4.0;
  l=2.0*PAI*r;
  #undef PAI
  a=PAI*r*r;
  v=(3.0/4.0)*PAI*r*r*r;
  printf("R=%f\n",r);
  printf("L=%f\n A=%f\n V=%f\n",l,a,v);
}
```

从宏展开可以看出,在取消宏定义指令 #undef PAI 后的宏名 PAI 不能展开。因为预处理程序遇到这条宏指令后,便使宏定义 #define PAI 3.14159 失效。

### 2. #include

这条宏指令叫宏包含,其用法有两种:

```
#include "文件名"
#include <文件名>
```

遇到这两条指令之一,预处理程序就把文件名所指定的文件引进到本指令所在的源文件中,即用文件名所指定的文件内容替换#include指令行。

这两种宏包含指令是有区别的。遇到前者,预处理程序首先检索当前的文件目录有无该文件,若没有检索到,再检索C编译系统中的目录。遇到后者,预处理程序只检索C编译系统的目录。宏包含的实例如下:

```
A>type u.h
#define begin{
#define end }
#define DMAX(x,y) x>y? x:y
C>type a.exp6-6.c
#include "a.u.h"
main()
begin
    double x=1000.0,y=200.0;
    long lx=11,ly=21;
    int i;
    printf("d-max=%lf\n",DMAX(x,y));
    printf("l-max=%ld\n",DMAX(lx,ly));
end
A>exp6-6
d-max=1000.000000
l-max=21
```

我们可以事先编好具有各种功能的标题文件(扩展名为.h的文件)或其它文件,然后再用#include把它们包含到当前文件里使用。

stdio.h是用得最多的标题文件,内容大致包括:符号定义、标准输入设备的定义及常数、函数、类型的定义等。在根目录下可用如下命令查看。

```
C>CD TC\include
C>type 'stdio.h'
```

标准输入输出函数 getchar() 和 putchar() 就是在该文件里定义的,故凡是要使用这两个函数的文件,其开头部分必须有宏包含:

```
#include <stdio.h>
或 #include "stdio.h"
```

getchar() 的功能是从操作系统确定的标准输入设备输入一个字符。其特点是:没有参数;类型为整型;返回值为字符的 ASCII 码。用法:

```
int c;
c=getchar();
或 char c;
c=getchar();
```

putchar() 的功能是从操作系统所确定的标准输出设备上输出一个字符,其特点是:必须把输出字符作为参数给出;其类型为整数;没有返回值。

下面是用这两个函数输入输出的最简单的例子:

```
A>type exp6-7.c
#include "stdio.h"
main()
{ int c;
  c=getchar();
```

```
    putchar(c);
}
A>exp6-7
3 (仿效返回)
3 (执行结果)
A>exp6-7
w
w
下面是使用这两个函数连续输入输出的程序:
A>type exp6-8.c
#include "stdio.h"
main()
{ for(;;)
  putchar(getchar());
}
```

```
A>exp6-8
abc. (仿效返回)
abc. (执行结果)
```

### 3. 条件编译

这类宏指令有: #if、#ifdef、#ifndef、#endif、#else、#elif。它们按一定方式组合,构成条件编译程序结构。下面介绍这些程序结构。

(1) #if~#endif 结构 这种结构的用法如下:

```
#if 常量表达式
语句;
#endif
```

如果常量表达式的值不为0,则 Turbo C 编译该程序模块;否则,不编译该程序模块。如练习 6.9 所示。

```
C>type exp6-9.c
#define MAX 10
#if MAX==0
#define MAX 5
#endif
main()
{ int i=6;
  do {
    printf("i=%d\n",i++);
  } while (i<MAX);
}
```

由于 MAX 先用#define 定义为 10,其条件编译模块不被编译,故 MAX 仍保持为 10,执行结果如下:

```
C>exp6-9
i=6
i=7
i=8
i=9
```

如把#define MAX 10 删去,则如练习 6.10。

```
A>type exp6-10.c
#if MAX==0
#define MAX 5
#endif
main()
{ int i=2;
  do { printf("i=%d\n",i++);
```

```

    } while(i<MAX);
}
A>exp6-10
i=2
i=3
i=4

```

(2) #if ~ #else ~ #endif 用法如下:

```

    #if 常量表达式
        语句 1;
    #else
        语句 2;
    #endif

```

如果常量表达式的值不为 0, 则编译语句 1; 否则, 编译语句 2, 如练习 6.11 所示。

```

A>type exp6-11.c
#define MAX 100
#if MAX>10
    #define MAX 10
#else
    #define MAX 5
#endif
main()
{
    int i=6;
    do { printf(" i=%d\n", i++);
        } while(i<MAX);
}

```

因为设定 MAX=100, 根据 #if ~ #else ~ #endif 的功能, 使 MAX=10, 故程序执行结果如下

```

A>exp6-11.exe
i=6
i=7
i=8
i=9

```

(3) #if ~ #elif ~ #endif 用法如下:

```

    #if 常量表达式 1
        语句 1;
    #elif 常量表达式 2
        语句 2;
    #endif

```

这里, #elif 可进行多次重复, 如练习 6.12。

```

A>type exp6-12.c
#define MAX 4
#if MAX > 10
    #define MAX 10
#elif MAX > 3
    #define MAX 3
#elif MAX > 2
    #define MAX 2
#endif
main()
{
    int i=0;

```

```

do { printf(" i=%d\n", i++);
    } while(i<MAX);
}
A>exp6-12
i=0
i=1
i=2

```

(4) #ifdef、#ifndef 这两条宏指令是用来识别宏名是否被定义过了。它们均可取代 #if, 构成条件编译结构。对于 #ifdef, 当其后跟的标识符被 #define 定义过时为真; 对于 #ifndef, 只有其后跟的标识符没有被 #define 定义过才为真。如练习 6.13 和 6.14;

```

A>type exp6-13.c
#define MAX 4
main()
{
    #ifdef MAX
        printf("define MAX\n");
    #else
        printf("not defined MAX\n");
    #endif
}
A>exp6-13
define MAX
A>type exp6-14.c
#define MAX 4
main()
{
    #ifndef MAX
        printf("define MAX\n");
    #else
        printf("not defined MAX\n");
    #endif
}

```

```

A>exp6-14
not defined MAX
4. #line

```

这条宏指令可用于修改 Turbo C 中所定义的 --LINE-- 和 --FILE-- 等宏名的内容, 用法如下:

```

    #line 行号 "文件名"

```

注意该行号为源程序中当前行号, 文件名为当前编译的文件名。

```

A>type exp6-15.c
main()
{
    printf("line=%d\n", --LINE--);
    {
        #line 1000 "sub-block"
        printf("line=%d %s\n", --LINE--, --FILE--);
    }
}
A>exp6-15
line=2
line=5 A:\EXP6-15.C

```

从执行结果可以看出, 尽管我们在该程序 4 行中所指定的文件并非当前编译的文件 exp6-15.c, 但编译程序还是把 --FILE-- 和 --LINE-- 分别按当前编



译文件名和源程序中当前行号进行处理的。再做一个练习。

```
C>type exp6-16.c
#line 100
main()
{ printf("LINE=%d\n",--LINE--);
}
```

```
C>exp6-16
LINE=3
```

5. #pragma 用法:  
#pragma 名字

在 Turbo C 中有两个名字可使用,即:

(1) #pragma inline 该宏指令对于 Turbo C 来说,表示程序内含有 inline 汇编码,等效于指定编译任选项-B。

(2) #pragma warn 该宏指令与许可编译任选项-wxxx 所指定的 xxx 相同,但由 #pragma warn 所指定的 xxx 优先。例如,若编译如下程序,则会因变量 i 不确定而发生警告。

```
C>type exp6-17.c
/* pragma warn -def */
main()
{ int i;
  if (i>=100) printf("i=%d\n",i);
}
```

编译如下程序则不会出现警告。

```
C>type exp6-18.c
#pragma warn -def
main()
{ int i;
  if (i>=100) printf("i=%d\n",i);
}
```

6. #error

这条宏指令的用法是:

#error 错误信息

在编译过程中,一旦遇到这条宏指令,便会中止编译,读该宏指令,显示出错误信息后而停止。

```
C>type exp6-19.c
main()
{ printf("Before #error");
  #error error-message
  printf("After #error");
}
```

信息窗口显示内容如下:

```
Compiling C:\EXP6-19.C;
Error C:\EXP6-18.C 3; Error directive;
error-message in function main
```

7. 内部宏名

ANSI 标准规定了 5 个内部宏名,即:--LINE--

--FILE--,--DATE--,--TIME--,--STDC--。

Turbo C 增补了 10 个内部宏名,即:--CDECL--

--COMPACT--,--HUGE--,--LARGE--,--MEDIUM--,--MSDOS--,--PASCAL--,--SMALL--,--TINY--,--TURBOC--。

下面说明各内部宏名的定义值。

--LINE--存放当前编译的行号

--FILE--存放当前编译的文件名

--DATE--存放源文件被编译为目标文件的日期,存放形式为月/日/年。

--TIME--存放源文件被转换为目标文件的开始时间,以时:分:秒形式存放。

--STDC--若存有 10 进制常数 1,则代表使用的环境工具是标准工具程序;否则,为非标准工具程序。

--CDECL--使用标准 C 调用协议,即没有使用 pascal 选择项时,--CDECL--有定义;否则无定义。

--TINY--,--SMALL--,--COMPACT--,--MEDIUM--,--LARGE--,--HUGE--这 6 个内部宏名是表示存储模式的,因此,它们当中只有对应于编译中定义的存储模式的一个才被定义。

--MSDOS--使用 MS-DOS 版 Turbo C 时被定义为 1。

--PASCAL--使用 pascal 调用协议时,--PASCAL--被定义;否则不被定义。

--TURBOC--存有使用的 Turbo C 的版本号。

```
A>type exp6-20.c
```

```
main()
{ printf("%s %d %s %s\n",--FILE--,
  --LINE--,--DATE--,--TIME--);
  printf("Program being compiled using the ");
  #ifdef --TINY--
    printf("tiny model");
  #endif
  #ifdef --SMALL--
    printf("small model");
  #endif
  #ifdef --COMPACT--
    printf("compact model");
  #endif
  #ifdef --MEDIUM--
    printf("medium model");
  #endif
  #ifdef --LARGE--
    printf("large model");
  #endif
  #ifdef --HUGE--
    printf("huge model");
  #endif
  printf("\n");
}
```

```
C>exp6-19
```

```
EXP6-19.C 3 Jan 01 1980 00:16:20
Program being compiled using the small model
```



# 单片机旅馆客房门卫系统

杭州旅游职业学校 傅士淦

旅馆饭店的客房安全服务素为管理者所重视,也为住宿旅客所关切。客房服务员为搞好安全服务付出了辛勤劳动。但有时旅客出入客房频繁,稍为怠慢即被认为有服务不周之嫌;倘使偶有疏忽,客房财物短少,即造成企业和个人的损失。因此,客房安全管理实为一项繁琐又重要的工作。

现用单片微电脑设计控制的门卫系统,可以对客房实行有效的安全管理。它忠于职守,不徇私情,能可靠地代替人们的劳动。

旅馆客房门卫系统功能:

旅客进入客房时,需在系统键盘上键入六位数字密码。经电脑核实无误,指示灯亮,房门自动打开;十五秒钟后房门自动关闭,指示灯灭。如有外力阻碍开门或关门,系统将发讯号报警。出室者只须按室内开门键,门即自动打开十五秒钟并关闭。内存密码 100 组,可供管理者调换选用。

一、系统硬件介绍:

系统硬件电原理图如图 1。

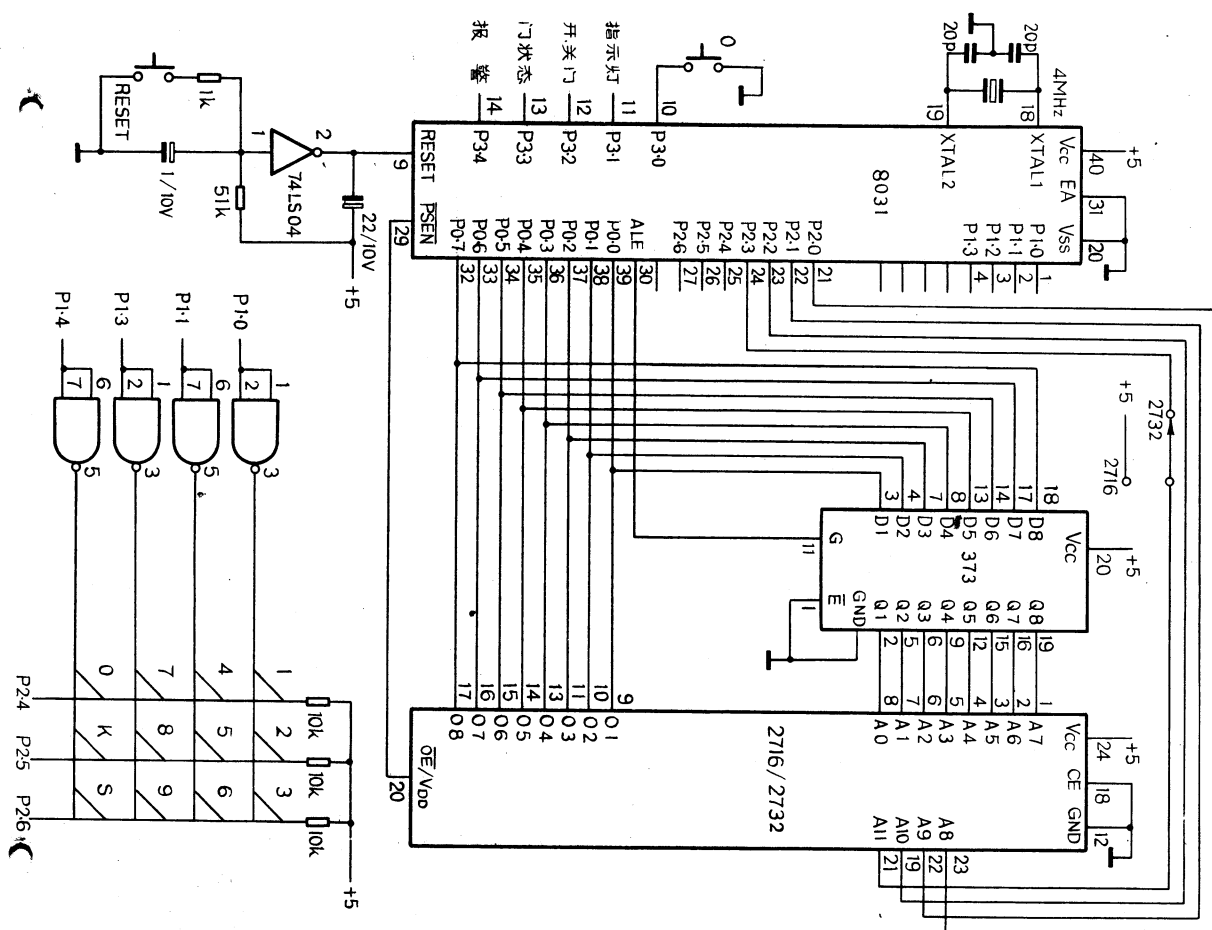


图 1

键盘用矩阵联接编码方式。键盘上设置十三个按键。其中一个复位键,十个数字键,二个命令键。

## 二、系统软件介绍：

系统软件流程框图如图 2。

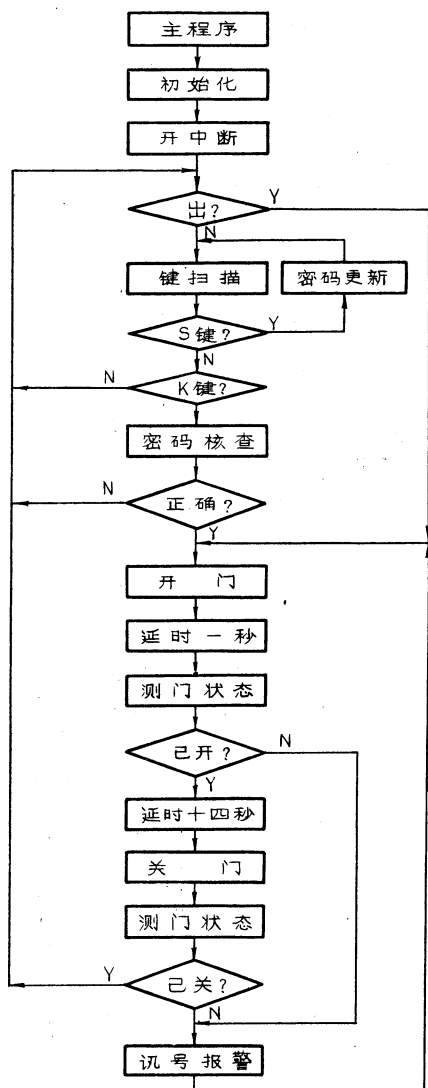
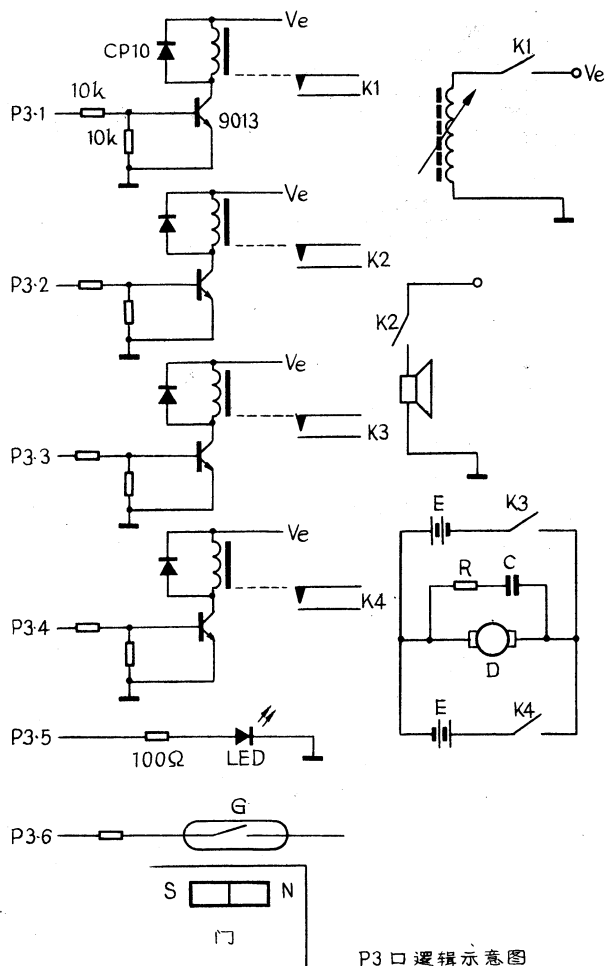


图 2

系统程序由主程序、键盘扫描子程序密码输入子程序密码核查子程序及中断服务子程序组成。



### P3口逻辑示意图

利用单片机机内定时器,设置 100SM 溢出中断一次。中断服务子程序对中断次数计数,并把计数值置于 R6 中,用来延时计时。

### 三、系统工作说明:

复位键 RESET 可中断程序运行,使程序返回入口地址 0000H,也可用此键中止报警。

用户首先选定内存密码,可用命令键 S。把已写入 EPROM 的 100 组密码,选择任一组作当前密码用。在键盘上输入 0~99 数字,按下 S 键,便确定当前密码组的首址。此键由管理人员专用。(装置此键不必外露)

命令键 K 是执行键,当六位数字密码输入后,按下 K 键,系统就进行核查工作。若输入密码数字正确无误,置 P3.1、P3.3、P3.5 高电平,执行启锁、开门、亮指示灯动作。K1 吸合,接通电磁铁 M,启开锁舌。K3 吸合,直流电机 D 正转,经传动机构变速,打开门户。延时一秒钟,置 P3.3 低电平,关电机。此时检测 P3.6 位,若是高电平,因舌簧管 G 仍被装在门上的磁铁吸合,门未被打开。置 P3.2 高电平,发讯号报警。若检测到 P3.6 位低电平,则门已被打开。延时十五秒,置 P3.4 高电平,K4 吸合,直流电机反转驱动门户关闭。延时

一秒再次检测门的状态,若门已关,置 P3.1、P3.4、P3.5 低电平,执行以下工作;释放电磁铁使锁舌合上、关电机、关指示灯。若门未关上则发信号报警,直至外界干预。

系统另有一个出门键 OUT,装于客房内。按下此键,P3.0 位接地。系统检测到 P3.0 低电平状态,即转入自动开门、关门等动作。

#### 四、核查控制程序清单:

```

TEXT:      MOV R0, #10H      ;输入密码存放首址
           MOV DPTR, #0300H  ;内存密码组存放首址
           MOV R2, #06H      ;核查六次
           MOV A, 20H        ;密码组代码
TX0:      MOVC A, @A+DPTR    ;取密码
           XRL A, @R0        ;核查
           JZ TX1            ;相符继续核查
           SJMP MAIN         ;不相符返回循环
TX1:      INC R0              ;
           INC DPTR          ;
           DJNZ R2, TX0      ;
TX2:      SETB P3.1          ;开锁
           SETB P3.3          ;开门
           STEB P3.5          ;亮指示灯
           CLR A              ;
           MOV R6, A          ;
TX3:      MOV A, R6           ;延时一秒钟
           CJNE A, #0AH, TX3 ;
           CLR P3.3          ;关电机
           JNB P3.6, TX4      ;检测门状态
           SJMP TX8           ;未开报警
TX4:      MOV A, R6           ;门已开延时十五秒
           CJNE A, #0A0H, TX4 ;
           SETB P3.4          ;关门
TX5:      MOV A, R6           ;延时一秒
           CJNE A, #0AAH, TX5 ;
           CLR P3.4          ;关电视
TX6:      JB P3.6, TX7        ;测门状态
           SJMP TX8           ;未关报警
TX7:      CLR P3.1           ;门已关上锁
           CLR P3.5          ;熄指示灯
           SJMP MAIN         ;返回循环
TX8:      SETB P3.2          ;报警讯号
           SJMP TX8

```

Addr	Opcode	Line	Source Line
		1	ORG 0000H
0000H	010D	2	AJMP MAIN
		3	ORG 000BH
000BH	2112	4	AJMP JSW
000DH	C2B1	5	MAIN: CLR P3.1
000FH	C2B1	6	CLR P3.1
0011H	C2B3	7	CLR P3.3
0013H	C2B4	8	CLR P3.4
0015H	C2B5	9	CLR P3.5
0017H	C2B6	10	CLR P3.6
0019H	758160	11	MOV SP, #60H

Addr	Opcode	Line	Source Line
001CH	758901	12	MOV TMOD, #01H
001FH	E4	13	CLR A
0020H	FE	14	MOV R6, A
0021H	F520	15	MOV 20H, A
0023H	758AB0	16	MOV TLO, #0B0H
0026H	758C3C	17	MOV TH0, #3CH
0029H	753010	18	MOV 30H, #10H
002CH	D28C	19	SETB TR0
002EH	20B002	20	MP0: JB P3.0, MP1
0031H	0167	21	AJMP TXZ
0033H	1195	22	MP1: ACALL KENY
0035H	FC	23	MOV R4, A
0036H	740B	24	MOV A, #0BH
0038H	9C	25	SUBB A, R4
0039H	4002	26	JC MP2
003BH	8004	27	SJMP MP3
003DH	11CA	28	MP2: ACALL INP
003FH	80ED	29	SJMP MP0
0041H	1195	30	MP3: ACALL KENY
0043H	B40C02	31	CJNE A, #0CH, MP4
0046H	800D	32	SJMP TEXT
0048H	1195	33	MP4: ACALL KENY
004AH	B41406	34	CJNE A, #14H, MP5
004DH	7811	35	MOV R0, #11H
004FH	7920	36	MOV R1, #20H
0051H	11DC	37	ACALL JUB
0053H	012E	38	MP5: AJMP MP0
0055H	11E8	39	TEXT: ACALL JPC
0057H	7811	40	MOV R0, #11H
0059H	7A06	41	MOV R2, #06H
005BH	E520	42	MOV A, 20H
005DH	93	43	TX0: MOVC A, @A+DPTR
005EH	66	44	XRL A, @R0
005FH	6002	45	JZ TX1
0061H	80CB	46	SJMP MP0
0063H	08	47	TX1: INC R0
0064H	A3	48	INC DPTR
0065H	DAF6	49	DJNZ R2, TX0
0067H	D2B1	50	TX2: SETB P3.1
0069H	D2B3	51	SETB P3.3
006BH	D2B5	52	SETB P3.5
006DH	E4	53	CLR A
006EH	FE	54	MOV R6, A
006FH	EE	55	TX3: MOV A, R6
0070H	B40AFC	56	CJNE A, #0AH, TX3
0073H	C2B3	57	CLR P3.3
0075H	20B602	58	JB P3.6, TX4
0078H	8017	59	SJMP TX8
007AH	EE	60	TX4: MOV A, R6
007BH	B4A0FC	61	CJNE A, #0A0H, TX4
007EH	D2B4	62	SETB P3.4
0080H	EE	63	TX5: MOV A, R6
0081H	B4AAFC	64	CJNE A, #0AAH, TX5
0084H	C2B4	65	CLR P3.4
0086H	20B602	66	TX6: JB P3.6, TX7



0089H	8006	67		SJMP TX8	00D6H	8003	122		SJMP IN2
008BH	C2B1	68	TX7:	CLR P3.1	00D8H	753016	123	IN1:	MOV 30H, #16H
008DH	C2B5	69		CLR P3.5	00DBH	22	124	IN2:	RET
008FH	809D	70		SJMP MP0			125		
0091H	D2B2	71	TX8:	SETB P3.2	00DCH	E6	126	JUB:	MOV A, @R0
0093H	80FC	72		SJMP TX8	00DDH	540F	127		ANL A, #0FH
		73			00DFH	FB	128		MOV R3, A
0095H	7A08	74	KENY:	MOV R2, #08H	00E0H	18	129		DEC R0
0097H	7B80	75		MOV R3, #80H	00E1H	E6	130		MOV A, @R0
0099H	EB	76	EN0:	MOV A, R3	00E2H	C4	131		SWAP A
009AH	F590	77		MOV P1, A	00E3H	54F0	132		ANL A, #0F0H
009CH	C3	78		CLR C	00E5H	4B	133		ORL A, R3
009DH	13	79		RRC A	00E6H	F9	134		MOV R1, A
009EH	FB	80		MOV R3, A	00E7H	22	135		RET
009FH	E5A0	81		MOV A, P2			136		
00A1H	F4	82		CPL A	00E8H	E520	137	JPC:	MOV A, 20H
00A2H	5470	83		ANL A, #70H	00EAH	540F	138		ANL A, #0FH
00A4H	7002	84		JNZ EN1	00ECH	FB	139		MOV R3, A
00A6H	DAF1	85		DJNZ R2, EN0	00EDH	EB	140		MOV A, R3
00A8H	03	86	EN1:	RRA	00EEH	A4	141		MUL AB
00A9H	03	87		RRA	00EFH	FB	142		MOV R3, A
00AAH	541F	88		ANL A, #1FH	00F0H	75F010	143		MOV B, #10H
00ACH	4A	89		ORL A, R2	00F3H	C5F0	144		XCH A, B
00ADH	FA	90		MOV R2, A	00F5H	CA	145		XCH A, R2
00AEH	9000BE	91		MOV DPTR, EN5	00F6H	A4	146		MUL AB
00B1H	7B0C	92		MOV R3, #0CH	00F7H	2A	147		ADD A, R2
00B3H	EB	93	EN2:	MOV A, R3	00F8H	CB	148		XCH A, R3
00B4H	93	94		MOVC A, @A+DPTR	00F9H	5320F0	149		ANL 20H, #0F0H
00B5H	B50A02	95		CJNE A, 0AH, EN4	00FCH	7820	150		MOV R0, #20H
00B8H	EB	96	EN3:	MOV A, R3	00FEH	26	151		ADD A, @R0
00B9H	22	97		RET	00FFH	CB	152		XCH A, R3
00BAH	DBF7	98	EN4:	DJNZ R3, EN2	0100H	3400	153		ADDC A, #00H
00BCH	80FA	99		SJMB EN3	0102H	FA	154		MOV R2, A
		100	EN5:	DB #08H	0103H	7903	155		MOV R1, #03H
		101		DB #05H	0105H	7F00	156		MOV R7, #00H
		102		DB #09H	0107H	EB	157		MOV A, R3
		103		DB #11H	0108H	2F	158		ADD A, R7
		104		DB #06H	0109H	FD	159		MOV R5, A
		105		DB #0AH	010AH	EA	160		MOV A, R2
		106		DB #12H	010BH	39	161		ADDC A, R1
		107		DB #07H	010CH	FC	162		MOV R4, A
		108		DB #0BH	010DH	8D82	163		MOV DPL, R5
		109		DB #13H	010FH	8C83	164		MOV DPH, R4
		110		DB #0CH	0111H	22	165		RET
		111		DB #14H			166		
		112			0112H	758AB7	167	JSW:	MOV TL0, #0B7H
00CAH	7830	113	INP:	MOV R0, #30H	0115H	758C3C	168		MOV TH0, #03CH
00CCH	E6	114		MOV A, @R0	0118H	0E	169		INC R6
00CDH	F9	115		MOV R1, A	0119H	32	170		RETI
00CEH	EC	116		MOV A, R4			171		
00CFH	F7	117		MOV @R1, A			172		ORG 0300H
00D0H	09	118		INC R1			173		DB #01H
00D1H	E9	119		MOV A, R1			174		DB #02H
00D2H	F6	120		MOV @R0, A			175		DB #03H
00D3H	B41602	121		CJNE A, #16H, IN1			176		DB #04H

177 DB # 05H  
178 DB # 06H  
179 DB # 07H  
180 DB # 08H  
181 DB # 09H

ERROR 0

182 DB # 00H

## 中华学习机汉字处理(CPC)卡使用说明

中华学习机汉字处理卡(以下简称 CPC 卡)是专为 CEC 一 I 型中华学习机设计的五笔字型输入、打印机驱动和汉字全屏编辑的多功能卡。插上该卡后,中华学习机在汉字状态下,除可以使用原有的“区位”、“拼音”两种汉字输入方法外,还可使用“五笔”方式输入。同时还可使用 CPC 卡上的汉字全屏编辑软件以及打印驱动功能。CPC 卡虽然使中华学习机增加了三项功能,但未改动和减少原中华学习机的任何功能。

CPC 卡具体使用方法如下:

### 安装办法

将中华学习机电源关掉。把 CPC 卡插入扩充槽内(器件面向用户)。如使用 CPC 卡上的汉字全屏编辑软件,还需在磁盘驱动器中插入一张带有 DOS3.30 系统的软盘。然后打开显示器和中华学习机的电源即可使用下述功能。

### 一、打印功能(PR #1):

为方便用户,中华学习机 CPC 卡在中西文方式下均保留原单一打印卡的全部功能,且操作方法不变。

在西文方式下,当准备好打印机之后,只需直接键入“PR #1”便可接通打印机。此时 CPC 卡为用户提供了字符打印及高分辨率图形拷贝功能。

#### ※打印西文字符

用于打印程序清单、程序运行结果以及其它信息。

在此方式下,原打印控制符均不变。如 PRINT CHR \$(20)可使输出的西文字符在正体/斜体之间转换;PRINT CHR \$(23)可使输出的大写字母变为小写。其它如加重打印、缩小打印、行间距控制等【ESC】序列控制符均可照常使用。

#### ※打印图形

西文方式下接通打印机后,直接按 CTRL-Q(【西文】)键或用 BASIC 语句 PRINT CHR \$(17)即可按用户设置的参数打印出高分辨率图形。

参数设置方法如下:

$$N = P + B + I + S + L$$

其中:

P=0 不打印  
P=1 打印第一页  
P=2 打印第二页  
P=3 两页并排打印  
B=0 直接打印原图  
B=4 两页与(AND)运算后打印  
B=8 两页或(OR)运算后打印  
B=16 两页异或(EOR)运算后打印  
I=0 正相打印  
I=32 反相打印  
S=0 一倍大小打印

S=64 二倍大小打印  
L=0 打印全屏图形  
L=128 打印一行图形

用户按需要选择各参数并相加后,使用 POKE1913,N 即可确定打印参数。在打印过程中,可按 CTRL-C 中断打印。

汉字方式下的打印设置也保持不变:

#### ※设置打印方式

POKE1659,N

N=0 为不打印,以屏幕作为输出显示设备。N=1~15 为打印,以打印机作为输出设备,所打印字型与 N 的值有关。共有 15 种大小字型,N=1 最小,N=15 最大。

#### ※设置字间距

POKE1687,N

N 为所打印的字符及汉字之间的字间距离,N 的取值为 0~255,以点为单位。

#### ※设置行距

POKE1915,N

N 为打印行之间的点距离,N 的取值为 0~255。

#### ※设置行打印字数

POKE2043,N

N 为每行所打印的汉字个数。每一汉字占两个字符位置。N 的取值为 0~255。

### 二、五笔字型输入功能(PR #8)

用户键入“PR #8”后,中华学习机自动进入汉字状态,并将屏幕编辑软件调入 16K 扩充 RAM。此时中华学习机可使用五笔字型输入法进行汉字输入,并可使用屏幕编辑软件进行文字编辑。

在汉字状态下,按【F4】键便可进入五笔字型输入方式。此时屏幕底端的状态提示行显示“五笔!”。除“Z”键外的 25 个小写字母,将作为五笔字型编码输入而显示在屏幕底端的状态提示行上。大写字母和数字符号将作为普通输入而显示在屏幕光标位置上。

CPC 卡提供了国标 GB-2312(80)16 区~87 区两级汉字(共计 6763 个汉字)编码,并有一、二、三级简码功能。例如:“啊”字可用编码:KBSK(原码)、KBS(三级简码)、KB(二级简码)三个编码进行汉字输入;“国”字可用编码:LGYY(原码)、LGY(三级简码)或 L(一级简码)三个编码进行汉字输入。同时,该卡还提供了二千六百多个双字词组供用户使用。这是其它中华学习机软、硬五笔系统所不具备的。

当在输入过程中出现重码的情况时,该卡将会在屏幕底端显示出所有的重码字,用户可使用数字键选择所需要的字。如果用户所要选择的为第一个重码字,不必用数字键选择,

(下转 47 页)



# MSX—EPROM 读写/擦除器

刘 园

MSX 微电脑在我国有较多的用户。为了扩充其功能,使之用于开发应用特设计 MSX—EPROM 写入器与擦除器。

**1. 硬件结构** 本写入器是利用了具有三个八位端口的可编程接口芯片 8255 和相应的控制程序构成的,其硬件电路如图 1 所示。其中 8255 的数据线分别与 MSX 总线的  $D_0 \sim D_7$  相连, $A_0A_1$  作为内部寄存器的选择线也接至 MSX 总线的  $A_0A_1$ , $\overline{IO}/\overline{RQ}$  和  $A_7$  经过二极管  $D_1D_2$  相或后接至 8255 的片选端作为选片信号,这样就有效地避免了和 MSX 机内的端口发生选址冲突。8255 的读写控制线分别接至 MSX 的总线  $\overline{WE}$  和  $\overline{RD}$  端,以便 CUP 对其进行相应的操作。 $\overline{RESET}$  是 8255 的复位信号,高电平有效,故接地。端口 A 接至待写 EPROM 的数据线  $D_0 \sim D_7$  构成写入器的数据总线。端口 B 接至待写 EPROM 的  $A_0 \sim A_7$  构成写入器的低八位地址线。端口 C 的  $PC_0 \sim PC_6$  接至待写 EPROM 的  $A_8 \sim A_{14}$ ,构成写入器的高七位地址线,而端口 C 的  $PC_7$  经过  $BG_1$ 、 $BG_2$  产生相应的编程脉冲加入  $V_{PP}$  端,其中  $D_4$  是为了保证读片时  $V_{PP}$  为高电平而设置的。

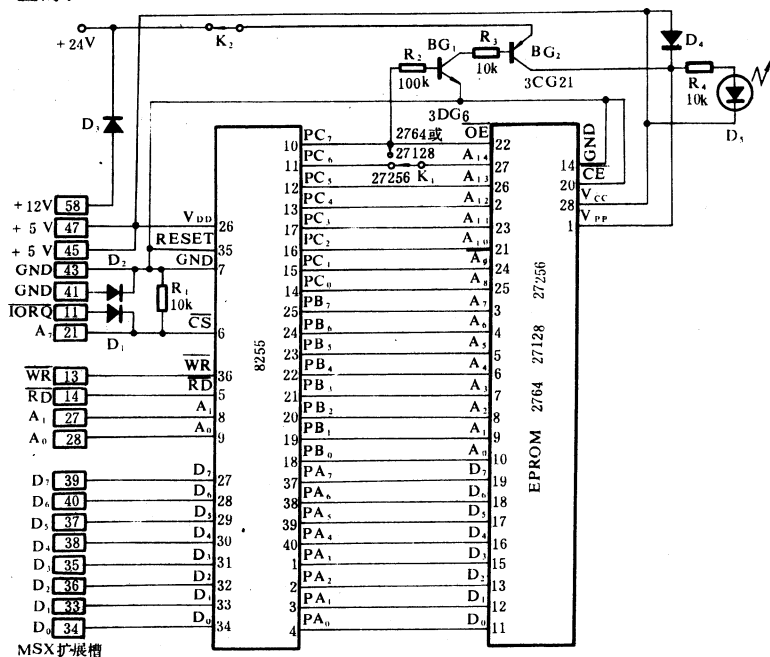


图 1

$D_5$  为发光二极管,当对 EPROM 进行写操作时发光,用来指示编程情况, $D_3$  为芯片所用编程电压不同而设计。当要写入的芯片为 12V 编程电压时无需外加编程电压,此时 12V 电压由 MSX 总线提供。当要写入的芯片为 24V 编程电压时,需在 24V 端加上电压,此时机内 12V 电压因  $D_3$  反编被切断,24V 将做为编程电压提供给芯片。开关  $K_2$  用来接通或断开编程电压以防误操作而带来不必要的麻烦。 $K_1$  用来选择 27256, 27128 或 2764 图 1 中所置位置的读写 27256,这样本写入器就可完成对 2764,27128,27256 三种不同的大容量 EPROM 进行读写和校验。

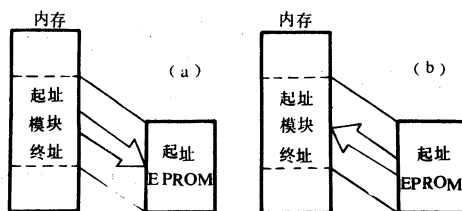


图 2

**2. 软件** 本写入器是在磁带版《监控汇编》的支持工作,当要对 EPROM 进行读写操作时,首先要 MSX 机用 RUN“CAS:”调入监控汇编,然后用“CMDMON”命令进入监控,用“X”命令在 BC 寄存器对中放入要操作的 EPROM 首址;在 DE 寄存器对中放入模块区首址;在 HL 寄存器对中放入模块区的尾地址,为以下各种操作做好准备。

在 A 寄存器中放入 1H 表示要对 EPROM 进行“写”操作,此时若敲入“GD400”就可把模块中的数据写入到 EPROM 中,如图 2(a)其工作过程见图 3 左边框,首先由  $PB_0 \sim PB_7$  和  $PC_0 \sim PC_6$  输出地址信号,把要写入的数据通过  $PA_0 \sim PA_7$  送到待写的 EPROM,然后通过  $PC_7$  输出约 2ms 左右的编程控制脉中,一个单元即完成了写操作,修正指

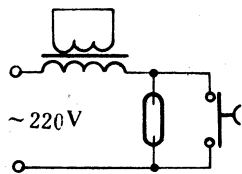


图 4

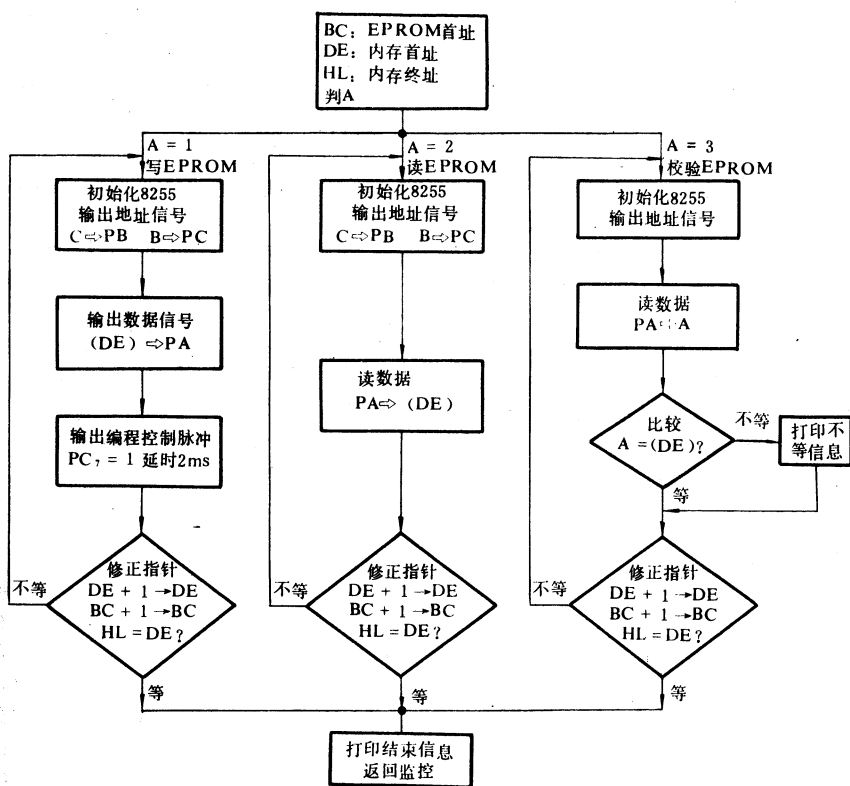


图 3

针后循环上述操作,直至把所有的数据写完。

在 A 寄存器中放入 2H 表示要对 EPROM 进行读操作,此时若敲入“GD400”就可把 EPROM 中的数据装入模块如图 2(b)。工作过程参见图 3 中框,也是先通过  $PB_0 \sim PB_7$  和  $PC_0 \sim PC_6$  发出地址信号而后从  $PA_0 \sim PA_7$  上读取 EPROM 的数据放入指定的模块单元中,修正指针后循环读到指定的长度。

在 A 寄存器中放入 3H 表示要对 EPROM 进行校验。此时若敲“GD400”即检查 EPROM 中的内容和指

定模块中的内容是否完全相等,当有不等内容时将打印出不等单元的地址和该单元的内容,如此反复循环到结束。工作过程见图 3 右框,和读出 EPROM 时大致相同,只是读出的数据不是放入指定的单元。而是和该单元进行比较后做判别处理。

以上所说的模块是指内存的某一区域。写入器所用软件主要由 ADDRES,  $V_{PP}$ , ERROR 和 LDINC 四个

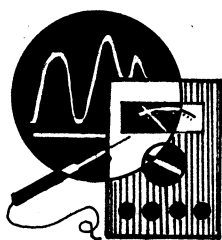
子程序构成,它们的功能分别是输出地址信号;产生编程脉冲控制信号;打印出错单元地址及该单元内容;修改指针判结束。

3. 擦除器 实际上就是一个普通的日光灯电路,电原理如图 4,考虑到体积和价格等因素没有加入时间控制,但完全可满足业余条件下擦除 EPROM。其中镇流器可用 XD7 信号灯的变压器将其次级短路后代用,紫外管可采用 250W 水银灯泡改制,首先将灯泡外壳轻轻敲破,取出其中的芯子,即可代替紫外管。而后用白铁皮制作一个大小合适,便于放置 EPROM 和紫外管的支架,安好变压器和起动按钮,按图检查无误即可通电试验。正常后可把铁架装入一个铝制小菜盒中,这样一个体积小巧,又很实用的擦除器即告制作完毕。使用时,把要擦除的

芯片放在支架上正对紫外管,启动紫外管发光,然后盖上盒盖约 30 分钟左右(不同 EPROM 擦除时间不同)即可把 EPROM 擦除干净。写入芯片时,首先要检查是否擦除干净(即检查所有单元是否为 FFH)然后即可按上述方法进行写入等操作。

有关 MSX—EPROM 读写/擦除器的程序清单,可向编辑部函索,收复印及邮挂费 2.00 元。





## 维修经验谈

# 任天堂系列游戏机检修实例

林华春

目前各种型号任天堂系列游戏机,虽然其型号繁多,但从电路原理上区别大致可分为三类。以 CPU、PPU 时钟晶振频率为例:① 26.601712MHz 单时钟晶振,此类机线路简单,图像质量好,游戏卡兼容性好;② 26.601712MHz 和 21.47727MHz;双时钟晶振,此类机线路略为复杂,图象质量较好,游戏卡兼容性稍差;③ 21.251465MHz,单时钟晶振,此类机线路复杂,增加了制式变换电路,图像质量较好,游戏卡兼容性好,但维修因缺少配件,比较困难。本文结合实际检修例子,用简单仪器、工具(如万用表、逻辑笔、示波器等)加实际经验的检修方法,来判断、排除游戏机电路故障,以此来介绍如何掌握游戏机的检修方法。在检修中还要配备一盘性能良好的游戏卡,作为测试卡使用(文中的故障说明均以第一代“魂斗罗”为例)。

**例一、故障现象:**有光栅,无图象,无伴音(单时钟晶振)。

**分析与检修:**联机通电后,电视机荧光屏上有宽屏幕光栅出现,此时可确定 5V 供电电源及 PPU 电路基本正常,再用万用表检测电脑板上各 IC 电路电源端是否有 5V 电压,经检测正常。因为有光栅,所以可确定晶振电路正常,然后用示波器检测 CPU65272 的②脚是否有晶振脉中,经检测正常,用万用表测 CPU③脚电压为 4.1V,用手按一下复位按键 RESET,表针回摆后复位,证明复位端正常。断电后测与 CPU 联接的 6116 静态存储器和 74LS139 译码器 IC 电路各脚对地电阻值,(万用表放 RX1K 档),检测后发现 74LS139④脚对地电阻仅为十几欧,与正常值相差较多,断开其外围电路再测量,仍为十几欧,换一片 74LS139 后开机,图像、伴音恢复正常。

**例二、故障现象:**与例一相同。

**分析与检修:**用上述方法检测后,没有发现故障点。反复按动 RESET 复位按键后,电视荧光屏上出现一些不规则的图案或颜色变化,此时可确定图像处理 PPU 电路正常,故障是由 CPU 及存储器电路引起,当用逻辑笔测 74LS139 第④脚时,为高电平,而测 6116 的⑩脚 CS 片选信号端时,为开路状态,断电后用万用表测两脚引线间电阻值,发现已开路,用细软线将两脚之间联接后开机,一切恢复正常。

**例三、故障现象:**一台灰白游戏机,无型号,与例一相同。

**分析与检修:**用上述方法检测后,没有发现异常,更换 CPU 后故障依然存在,此时可确认 CPU 电路正常,仔细观察电视荧光屏上的光栅,发现并不十分清

晰,随即更换 PPU6528PIC 电路,换新后开机,故障排除。

**例四、故障现象:**一台红白游戏机,游戏程序可正常进行,但在图像画面中产生一些多余的图案。

**分析与检修:**图像的出现是由 PPU 电路执行了 CPU 电路传输的信号后而产生的,从画面上看只有一些多余图案,但主画面并不破碎,可确定是 CPU 数据口电路损坏,换新 CPU87007 后,故障排除。

**例五、故障现象:**图像破碎,静画面游戏程序能正常进行,(背景画面称为静画面,手柄控制的人物为动画面)。

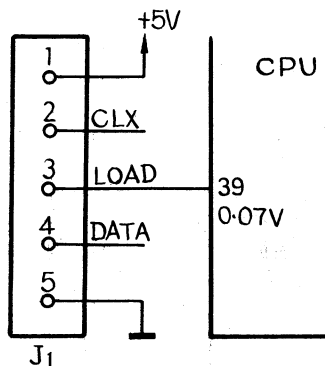
**分析与检修:**由上例故障分析可知,图像是由 PPU 电路及所属外围电路产生的,故其损坏后,图像即破碎,首先检查游戏机上联接游戏卡的插座,测量时发现有几个脚接触不良,用烙铁重新焊接后,故障排除。除此之外,74LS373 锁存器,视频随机静态存储器 6116 和其联接印刷电路板损坏,都能产生该故障。

**例六、故障现象:**一台小泰山 301 型游戏机,自动游戏程序正常,画面清晰,但不能启动。

**分析与检修:**按常规方法检查,用新手柄控制盒替代旧手柄控制盒,仍不能启动,用万用表电压档测 I 号手柄插座各脚电压值,发现第③脚无 0.07V 电压,参见附图。顺其引线测量 CPU 的③脚也无 0.07V 电压,将③脚外围电路断环节后测量仍无 0.07V 电压,故确定是 CPU 电路损坏,换新后故障排除。出现上述故障的原因是其它修理人员带电拔插手柄插头。

**例七、故障现象:**一台红白“小太阳”737 游戏机,故障现象同例六相同。

**分析与检修:**首先测 I 号手柄插座各脚电压值,发现③脚无 0.70V 电压,但测量 CPU③脚却有 0.07V 电压,断电后用万用表测插座至 CPU③脚间的连线,发现



例六附图

断线,联接后故障排除。此故障是人为将机器摔在地上把引线摔断。

例八:故障现象:一台任天堂 LT-900A 游戏机,故障现象同例六。

分析检修:正常情况下,把 I 号手柄拔下,插在 II 号手柄插座上,游戏节目能够启动。用此方法处理后,游戏节目能启动,这说明 I 号手柄及电路正常,由此可判断是电脑板上的 74HC368 损坏,换新后故障排除。也可以用上面方法检测 II 号手柄是否正常,此时可在 II 号手柄中加一个启动信号。

例九:故障现象:一台红白游戏机,按 I 号手柄上

启动按键 STAR 后,游戏节目不能启动,按其它键无反应,只有按 B 键后图象上选择单双人游戏的符号上下跳动。

分析与检修:为区别是手柄电路还是电脑板上 74HC368 损坏,可先换一个新手柄试验一下,经试验故障如初,可确定是 74HC368 损坏,测试后发现 74HC368 已损坏,换新后用原机手柄试验,故障仍然出现,所以怀疑是手柄有故障,经测量发现手柄电缆中有一根引线断线,联接后故障排除,对于这种故障,要分开检查,否则会造成故障判断困难。

## 让 PC 机接上家用彩电

### CGA—PAL 卡即将问世

本刊接到不少打算购 PC 机的读者来函。寻求 PC 机接彩电的 PAL 卡。鉴于目前国内尚无厂商供货。为了及时满足经济条件并不富裕的购机者的需要。本刊组织多年来一直支持本刊工作的朋友,开发出 CGA—PAL 卡。目前已完成设计工作,有待组织制作。为了较准确的进行投入,请有兴趣购买的读者速来函同我们联系,以便有的放矢地安排制作。

有关 CGA—PAL 卡的功能,简介如下:

CGA—PAL 适配器是一种复合卡,可直接接 CGA (640×200)彩色显示器;也可以接 PAL 制式或 NTSC 制式的彩色电视机(从电视机的视频输入端口输入;如果彩电上没有视频口,则需加射频调制器)。

CGA—PAL 卡有两种基本工作方式:字母数字方式(A/N)和可寻址所有点的图形方式(APA)。在 A/N 方式中,显示器可以工作于 40 列×25 行方式,以适应低分辨率显示器,也可以工作于 80 列×25 行方式,用来驱动中分辨率监视器。在上述两种方式中,均使用 8×8 的字符框定义字符,小写字符使用 7×7 的字符框定义,比大写字符低一行,大写,小写字符适用于所有

的工作方式。每个字符都可以使用 16 个前景颜色和 8 个背景颜色中的一种组合。

CGA—PAL 卡包括 16KB 的存储器,可存储多达 8 屏显示信息。在 80 列×25 行显示方式时,可存储四屏显示信息。全部 16KB 存储器均可直接由 CPC 直接寻址,使软件在管理屏幕时具有极大的灵活性。

在 APA 工作方式时,有两种可用的分辨率:一是中分辨率彩色图形方式(320 图素×200 行)。在中分辨率分式时,每个图素可以在四种颜色中选择一种使用。背景颜色可以是 16 种颜色之一。

在 A/N 方式时,字符由一个 ROM 字符发生器形成,字符发生器中包括了 256 种不同字符的点阵映象。

CGA—PAL 卡也可工作在黑白显示方式,此时能得到高分辨率方式,字符属性包括负象显示,闪耀和增亮功能,因为接口中全部 16KB 存储器都被用来定义图象的亮与不亮状态。

估计售价:350.00 元(注意:请勿汇款,仅来函表示购买意向及要求,以供我们估计市场需要量参考)。

## 介绍一种理想的“单片机教学实验系统”

“北京市单片机应用技术协会为进一步做好单片机的推广服务工作,近期集中了一批高校教师及有经验的技术干部研制了“单片机教学实验系统”。目前“BJS-51 系列单片机教学实验系统”的基本部分已经完成。“BJS-98 系列单片机教学实验系统”及功能扩展、综合应用部分正进行中。

这种教学实验系统区别于目前市场上一些立足于某仿

真器产品开发的教學系統,它完全从教学的实际需要出发,具备实验内容多、覆盖范围广、结构灵活、性能可靠、价格低廉等优点,是一种理想的教學实验设备。

凡对该产品有兴趣者,请与北京虎坊路十三号北京市单片机应用技术协会盛志纯联系。

邮编:100052

# 微计算机维修工具

## —8088 系统板故障诊断卡(续)

董 珊

### 五 8253 定时/计数控制器的检查

PC/XT 机系统板中,8253 定时/计数控制器完成三项工作:①为电钟提供一恒定的时间基准,由通道 0 每隔 0.055 秒产生一个输出信号(每秒产生 18.2 次),用以请求时钟中断,8088 响应该中断,用以计算一天的时间。②定时对 RAM 刷新,每隔 15.12 $\mu$ s 由通道 1 发出一输出信号,向 8237 的通道 0 提出申请,8237 的通道 0 负责实施刷新工作。③控制扬声器的音频,由通道 3 实现。

在测试卡的标准显示结果中,可以看到诊断程序对 8253 定时器的通道 0,通道 1 进行了测试,其中“8253 CHAN1……OK”,是 8253 通道 1 的测试结果,“8253,CHAN0……OK”,表示对 8253 通道 0 的检测结果。如若发现错误,8253 的诊断结果会有若干不同的情况,我们大体上可以将这若干情况分为两类:“并发性故障”和非并发性故障。所谓“并发性故障”,是指在 8253 显示出故障信息的同时,其它部分也同时报告出错误信息,特别是对 8237 和开关值部分,例如下述情况:

CPU……OK	SW1	1	2	3	4	5	6	7	8
8237 REG…ERROR		1	0	0	1	1	1	1	1
8237 CHANO…OK	SW2	1	2	3	4	5	6	7	8
8253 CHAN1…ERROR		1	1	1	1	1	1	1	1

在 8253 通道 1 有故障时,8237 的寄存器也出现了故障,并且开关值 SW1 的值亦不正确,这是一种典型的“并发性故障”。一般说来,这么多部分因损坏的可能性不大,所以首先检查它们的公用电路,例如 8237 和 8253 在数据传送时都经过的数据缓冲驱动器 U<sub>15</sub>(LS-245),片选信号的输入控制器 U<sub>45</sub>(LS-138)和控制传送驱动器 U<sub>17</sub>(LS-244)。所谓“非并发性故障”,是指在显示的信息中,只有 8253 一项出现错误。这种情况,一般是 8253 芯片本身或它的输入、输出控制部分有故障。首先,反复启动几次机器,查看故障信息是否固定不变,若现象固定,可检查 8253 的 9 脚、15 脚、18 脚它们分别是通道 0、1、2 的时钟输入,匀应有脉冲信号,若为恒定电平,可能为 U<sub>21</sub>损坏。若输入正常,可查 8253 的 21 脚片选信号( $\overline{CS}$ )是否有负脉冲,若没有,可能是 U<sub>45</sub>有故障。当 8253 的时钟输入,片选信号都正常以后,可查 U<sub>73</sub>,因为即使 8253 发出了正常信号刷新内存,但 U<sub>73</sub>不好,诊断程序也无法运行。假如这些匀是好的,可断定 8253 芯片损坏。在替换 8253

时,要选用 8253-5 的型号,这是因为其它型号的 8253 芯片时钟频率不一样,在 PC/XT 机上不能正常工作。

### 六 8237DMA 控制逻辑的维修

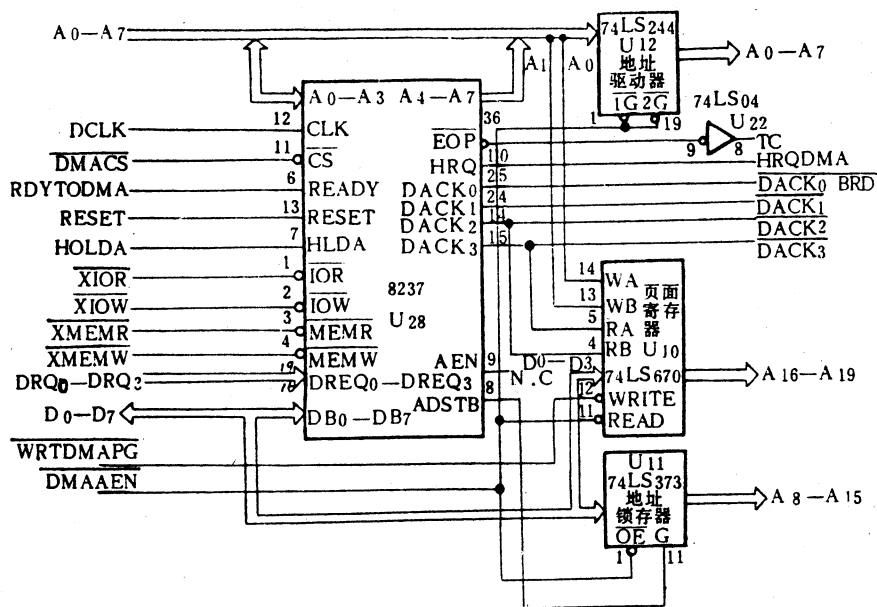
IBM PC/XT 的 DMA 控制逻辑由 DMA 控制电路和 DMA 应答电路两部分组成。DMA 控制电路由 8237DMA 控制器(U<sub>28</sub>),DMA 页面寄存器(U<sub>10</sub>),DMA 地址锁存器(U<sub>11</sub>)和 DMA 地址驱动器(U<sub>12</sub>)组成如图六所示,8237 有四个通道,通道 0 用于动态 RAM 刷新;通道 1 未用;通道 2 用于软磁盘的 DMA 操作,通道 3 用于硬磁盘的 DMA 操作。故只有通道 0 用在系统板上所以诊断卡的程序只对通道 0 作了全面的检查。

在发现 8237 报出故障信息时,一般应该首先检查 8237 和输入信号,8237 的 13 脚为 RESET 信号,在开机时应有一个正向脉冲信号,用以消除 8237 的内部寄存器。8237 的 1 脚( $\overline{IOR}$ )2 脚( $\overline{IOW}$ )应有脉冲信号。上述三个信号出现故障,不是短路就是有断路。8237 的另一个输入端是 11 脚( $\overline{CS}$ ),这里应为负脉冲信号,否则可查 U<sub>45</sub>和 U<sub>17</sub>。8237 的 12 脚为时钟输入信号,它的频率为 4.77MHZ,但它高电平的保持时间为 77ns,若时钟输入为恒定电平或波形不好,8237 出现故障,这时可查一下 U<sub>74</sub>和 U<sub>89</sub>,若二者都好,可能是延时芯片 TD2 有故障。

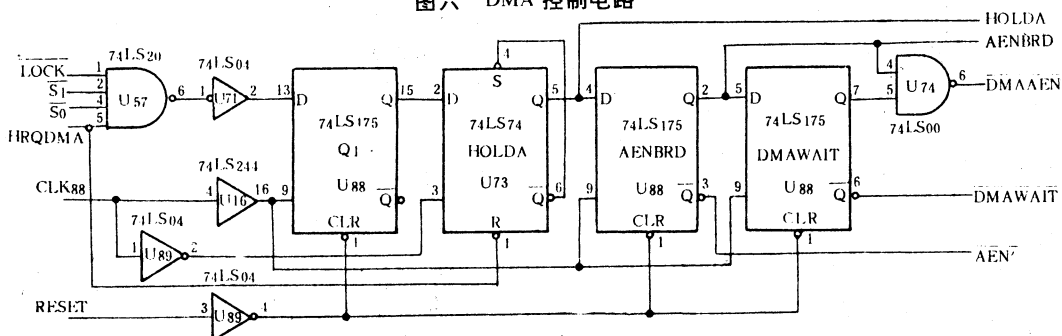
同时还应检查 8237 的 6 脚即 READY 信号,这里应该是脉冲信号,否则应检查 U<sub>88</sub>的引脚 8 和 U<sub>70</sub>是否正常。

若上述部分未见异常,应对 DMA 应答电路进行检查,诊断程序是通过通道 0 的响应情况来观察它的,当 8237 的引脚 19 有高电平或脉冲时,8237 的引脚 10 应有高电平或脉冲信号输出,否则就是 8237 坏,若 8237 的 10 脚正常,应检查 8237 的 7 脚是否收到高电平的 HLDA 保持请求响应信号,若 HLDA 是低电平,说明 DMA 应答电路有故障,可查 U<sub>57</sub>引脚 2、4,它们是 8088 送来的状态信号,应为脉冲, U<sub>57</sub>引脚 1 为 LOCK 信号应为高电平或脉冲,这时 U<sub>57</sub>-6 应为低电平或脉冲, U<sub>88</sub>的引脚 13 应为高电平或脉冲, U<sub>73</sub>的 5 脚应输出高电平或脉冲,所以当 HOLD 信号不正常时,一般是 U<sub>57</sub>、U<sub>71</sub>、U<sub>88</sub>、和 U<sub>73</sub>四个芯片中某一个损坏,如图七所示。

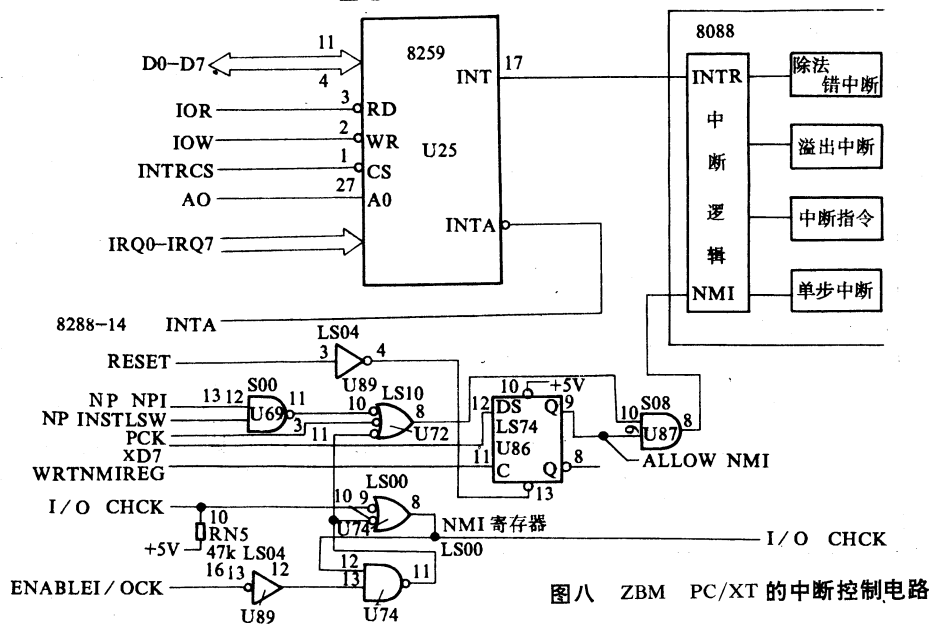
若 8237 的输入信号和 DMA 应答信号都是好的,则 8237 有故障的可能性很大。注意在更换 8237 芯片时,应使用 8237A-5 型号的芯片。



图六 DMA 控制电路



图七 DMA 应答电路



图八 ZBM PC/XT 的中断控制电路



8237 给出的错信息可能在查找故障点时比较复杂,应仔细的测量上述各点。这里我们举一个实例。从显示屏上见到如下信息:

```
8237      CHAN0.....ERROR
MEMORY    .....0000 FF ERROR
8237      STADUS bit  ERROR
```

内存给出的 0 体 FF 不太可能 8 个芯片都坏, 8237 的 0 通道错,表示内存刷新电路不正常,进一步说明不是 RAM 本身错,我们根据这些信息发现在 DMA 应答电路中 U<sub>57</sub>-6 脚输出波型不对,低电平在 2V 左右,而 U<sub>57</sub> 的输入匀正常,换掉 U<sub>57</sub> 后,马上恢复正常。

#### 七 对 8259 中断控制部分的检查

8259 中断控制器管理的是可屏蔽中断 INTR,这是由外部设备申请的中断,负责选择优先级最高者向 CPU 发出中断申请。利用诊断卡可以检查 8259 的内部寄存器,及中断请求响应情况。

例如发现错误信息:“8259 REG...ERROR”便可知这是测试程序对 8259 的内部寄存器进行检查时,发现有错误,这时应查一下 8259-1 脚(CS)是否为低电平或页脉冲,引脚 27 的地址线 A。也应有脉冲,同时 8259 的 2 脚(WR)和 3 脚(RD)也必须有负脉冲。若 CS 恒为高电平,应检查 U<sub>45</sub>和 U<sub>17</sub>;若 WR 信号为恒高电平,可检查 U<sub>13</sub>是否有故障。PC/XT 的中断控制电路见图八。

当基本 16K 内存测试通过后,测试卡中的检测程序将设置中断向量,并进行中断响应情况的测试,若发现错误会有如下的信息显示:

```
“8259 INTRR FLAG ERROR”
```

这时可以检查 8259-17 脚 INT 中断信号是否有高电平(或脉冲)输出,若只有低电平则 8259 损坏;若有高电平或脉冲输出,则可检查 8288 的引脚 14 是否有低电平或脉冲信号输出,若无,则一般是 8288 损坏,少数情况也可能是 8088 故障。

在更换 8259 时也要注意选用 8259AC-2 型号的芯片。

在对 8259 进行检查时同样会发生“并发性故障”例如显示:

```
8237 REG..... ERROR
8237 CHAN0..... ERROR
8253 CHAN1..... ERROR
8259 REG..... ERROR
```

造成这种现象的原因一般是公用部分故障例如 U<sub>15</sub>损坏的可能性最大。凡属于并发性故障,应首先在图纸中找到它们的公用电路,仔细测量,效果更好。

#### 八、开关读取不正常的检查

在测试卡测试结果的显示中,有 SW1 和 SW2 二项开关值的检查结果,其中 PC/XT 机只有 SW1,SW2

在 PC 机检测时才有实际意义。

这一部分的检测结果,反映的是 8255 并行接口控制器的工作情况,在 PC/XT 中,8255 共有三个接口,PA 口负责读取键盘输入扫描码;PB 口负责扬声器的控制、奇偶校验控制;PC 口负责 SW1 开关状态的读取。

在开关值读取不正常时,首先应检查开关本身是否有故障,方法是改变开关的设置状况,再观察测试结果,便可判断开关是否损坏。当开关值报出错误信息而且其它部分显示均正常时,例如:

```
CPU..... OK SW1 1 2 3 4 5 6 7 8
8237 REG... OK      1 1 1 0 1 0 0 1
8237 CHAN0... OK SW2 1 2 3 4 5 6 7 8
8253 CHAN1... OK      1 1 1 1 0 0 0 0
8259 REG... OK
```

这种情况下可以检查一下 U<sub>41</sub>(SL244),如果开关和 U<sub>41</sub>都正常则一般是 8255 的 PC 口有故障。

至此,我们可以看到 8088 系统板诊断卡在维修系统板时的作用,它几乎覆盖了系统板上的全部功能,并且在信息报告中将故障按功能分类,对缩小故障围范,提高工作效率有很大帮助,而且你越是对机器的电路熟悉,使用该诊断卡的时间越长,你越感到它确实是一个得力的系统板维修工具。

除此之外,从标准显示信息报告中还可知道,诊断卡还可对打印机并行卡和 RS232 串行卡进行类似的故障诊断,这里不再一一赘述。

## 磁盘驱动器修复一例

雷 勇

一台驱动器与主机正常连接,开机后驱动器呼呼转动不停,红灯不熄,无法正常引导 DOS,修理步骤如下:

1. 换上一台好驱动器,结果一切正常,这说明问题出在驱动器上。

2. 打开驱动器,发现磁头小车不能顺利平稳地滑动,仔细观察,注意到有两片写保护纸粘住了,以致不能正常工作,用镊子将其小心夹出,驱动器即恢复正常工作。

反思:现在驱动器和写保护片被广泛地使用,但是有些写保护片质量有问题,很容易卡在驱动器里面,这些又是不易被人注意的细节,很容易造成事故。

因此,在这里敬请广大用户注意:在每次使用盘片时应检查写保护片是否牢固,这样能够防止不出意外事故。

# 初级程序员级水平考试辅导讲座

## 汉字 dBASE III 使用技巧(中)

王路敬

23. 检索命令 FIND、SEEK 的功能上存在的共同缺陷是什么? 怎样解决?

无论是 FIND 还是 SEEK 命令,在功能上都不尽完善。它们存在的共同缺陷是:

(1)只能查找首记录,而不能进行连续的顺序查找;

(2)只能在索引文件上进行检索,而不能查找库文件。

利用顺序查找命令 LOCATE 和 CONTINUE 配合起来使用,可以进行连续的顺序查找。这种查找既可以在索引文件内进行查找,也可以在库文件内进行查找。在索引文件内进行查找时,是按着索引文件的记录逻辑顺序进行查找的;而在库文件中则是按照库文件的记录进行查找的;且可以实现连续的查找。

24. 在数据库文件及在索引文件中进行查询时, FIND、SEEK 和 LOCATE、CONTINUE 命令在功能上有何不同?

不同的地方见表 1—1 所示

表 1—1 FIND、SEEK 和 LOCATE、CONTINUE 功能对比

功 能 命 令	能否在库文件中 进行查询		能否在索引文件中进行查询						
	能否 定域 查询	能否 连续 查询	能否 定域 查询	能否 连续 查询	非索 引关 键字	字 符 型 关 键 字	数 字 型 关 键 字	日 期 型 关 键 字	组 合 关 键 字
FIND	不能	不能	不能	不能	不能	能	能	不能	能
SEEK	不能	不能	不能	不能	不能	能	能	能	能
LOCATE CONT — INUE	能	能	能	能	能	能	能	能	能

25. 使用快速检索命令 FIND 应注意什么问题?

检索命令 FIND 只能对索引文件进行各种信息查找,其目的是为了提提高信息检索速度。使用该命令时应注意下列几个问题:

(1)查找的字符串无需加引号

用户查找的目标信息可以是字符串,在用 FIND 命令时,字符串可不加界限符‘ ’、“ ”、□,而将字符直接写在 FIND 命令后面。

例如:

• FIND 高级工程师

DISP

(2)FIND 命令只能查找满足条件的首记录

(3)FIND 命令可以查找部分字符串

例如:

• FIND 高级

• DISP

(4)查找内存变量时,必须先用宏代换命令 & 将内存变量替换成字符串之后,才能进行查找。

例如:

• 条件=“高级工程师”

高级工程师

• FIND & 条件

• DISP

(5)不能查找非索引关键字

如果在操作中 FIND 查找的是没有索引的关键字,dBASE III 将回答“未发现”的错误信息。

(6)不能查找日期型索引关键字。

26. 检索命令 SEEK 比 FIND 功能上有哪些扩展?

SEEK 是 dBASE III 新增添的命令,可以在索引的数据库文件中,检索关键字与给定表达式相一致的首记录。它克服了 FIND 命令查找数字型关键字要经过多次变换,手续繁琐和不能查找日期型索引关键字的二大缺点。

例 1. 当索引关键字为数字型时,查找内存变量,不需进行任何变换,而直接进行查找,十分方便。

• 条件=45

45

• SEEK 条件

• DISP

同一问题,而 FIND 命令必须先用 STR()函数将数字型内存变量转换成字符型内存变量,然后再用宏代换 & 命令将字符型内存变量转换成字符串。最后再用 FIND 命令进行查找。

• 条件=45

• 条件=STR(条件,2)

• FIND & 条件

• DISP

例 2. 索引关键字为日期型时,SEEK 命令扩大了 FIND 命令的功能。

表 1—2 SEEK 和 FIND 功能比较一览表

异 同 命 令	字符型 索引关键字		数字型 索引关键字		日期型 索引关键字	
	常量	内存变量	常量	内存变量	常量	内存变量
FIND	直 引 不 引	接 用 加 号 要 代 宏 换	直 引 不 引	接 用 加 号 先 将 数 字 转 为 字 符 型 再 代 换	不 能	不 能
SEEK	不 能	不 用 代 宏 换	直 引 不 引	接 用 加 号 无 需 代 宏 换	不 能	能 够

• 日期=CTOD('09/12/81')

09/12/81

- SEEK 日期
- DISP

而 FIND 根本没有上述功能。

但是 SEEK 命令不能查找字符串,只能查找内存变量,这点在使用时应予注意。

借此机会小结 SEEK 和 FIND 命令在功能和应用上的异同点。见表 1—2。

#### 27. 索引文件有哪些特点?

归纳起来有以下几点:

- (1)索引文件占用磁盘空间少。
- (2)可实现多种不同形式的快速查询。
- (3)建立不同数据库之间的关联。

(4)打开一个数据库的同时可以打开七个与它相关的索引文件,录入数据时索引文件同时增加,并按逻辑顺序排列。

(5)当索引文件打开后,删除数据库全部记录时,对应的索引文件记录也一同删除,但建立的逻辑顺序不变。

#### 28. 索引文件和排序文件在使用上有何不同?

排序也称分类,按照某一关键字段来重新排列数据库中的数据,可递增或递减。使用的命令是 SORT,这个命令的作用是由打开的数据库文件再产生一个新的数据库文件,该文件中的记录包括了原文件中符合范围条件的所有记录,这些记录在新的数据库文件中首先按命令给的第一个关键字段的顺序进行排列,如果第一个关键字段的数据值相同,记录又按第二个关键字段进行排列等等。在使用排序文件时,排序文件并不改变原数据库,而是以原数据库文件为蓝本,按照给出的关键字段及用户指定的排序方法,建立一个新的,按某一关键的顺序排列的数据库。使用几次 SORT 命令就建立几个这样的数据库,所以它要占用许多存储空间。

与排序文件相比,索引文件是根据一个选定的关键字建立起来的,“关键字”是一个字段,或者是几个字段的组合。由于各记录均有与此关键字对应的项,便可根据这些项的值而对各记录按逻辑顺序进行排列。索引文件就是按此逻辑顺序号记录着各关键字项及对应的记录号的,当数据库文件打开时,若对应的索引文件也已打开,则对数据库的记录进行列表输出时,便可根据索引文件的内容将数据库内的记录不按原来的物理顺序,而由关键字规定的逻辑顺序列出来。

由于索引文件是依赖于数据库文件而存在的,所以检索某个索引文件时,应该在打开该索引文件的同时也必须打开与之相对应的数据库文件。INDEX 命令产生的索引文件,所占空间比原数据库文件要小。

#### 29. 打开索引文件有哪二种方法?二者在使用上有何区别?

有两种方法打开索引文件,一种是用 USE 命令,另一种是用 SET INDEX TO 命令。

使用 USE 命令打开与库相关的索引文件,一次可以打开一个,也可以打开多个,但最多能打开 7 个索引文件。被打开的 7 个索引文件可以一次打开,也可以分多次打开。

使用 SET INDEX TO 命令同样可以打开索引文件。二者不同的是:USE<库文件名>INDEX<索引文件名>是将库文件与其相关的索引文件一次打开,而 SET INDEX TO<索引文件名表>可先打开库文件,其后当需要时,另行打开索引文件。

#### 30. 删除标志对库文件索引有什么影响?

在 dBASE III 中有一删除标志记入开关的设置命令:

SET DELETED ON/OFF

当为 SET DELETED ON 状态时,删除标志就起作用了,被加删除标志的记录就好像不存在一样,对记录计数、查找、复制等操作都不考虑这些加删除标志的记录。要是对其索引,对加删除标志记录也进行索引,但操作时不考虑这些加了删除标志的记录。

如果断开删除标志开关,则索引文件又恢复其原来的面貌。

所以删除标志开关接通状态,带删除标志的记录被隐蔽起来,当删除标志开关断开之后,这些被隐蔽起来的记录又重新被显露出来。但是不管删除标志是处于接通或断开状态,当对库文件进行索引时,带删除标志的记录和未加删除标志的记录一样,都被索引了。

#### 31. DELETE, ZAP, DELETE FILE, ERASE 在使用上有什么区别?

DELETE 是给要删除的记录加删除标记,这样做的目的显然是为了慎重,真正地删除库文件中删除 DELETE 指定的这些记录,还必须调用其他命令。

ZAP 是快速清除库文件内容,这种删除是不可恢复的。但库文件结构仍然存在。

DELETE FILE 可以删除任何文件,但不能删除已打开的文件。

而 ERASE 是 dBASE III 新增加的一条命令,其功能与 DELETE FILE 一样。但在 dBASE II 中 ERASE 是清屏命令。

#### 32. COPY, COPY TO, COPY FILE 在使用上有什么不同?

COPY 命令是 PC-DOS 的内部命令。它可在 dBASE III 系统下用 RUN 或! 命令来调用,进行文件复制,运行速度比在 PC-DOS 系统要快。

COPY TO 是在 dBASE III 系统里的命令它可以进行:

(1)库文件结构的拷贝

操作:

- USE<库文件名>
- COPY TO<新库文件名> STRUCTURE

(2)库文件内容的拷贝

操作:

- USE<库文件名>
- COPY TO<新库文件名> [范围][FOR<条件>][FIELDS<字段名清单>]

(3)生成数据格式文件

操作

- USE<库文件名>
- COPY TO<数据格式文件名> SDE/DELIMITED [范围][FOR<条件>][FIELDS<字段名清单>]

而 COPY FILE 对 dBASE III 定义的各种文件进行复制。但不能对已打开的文件进行复制。

#### 33. DISPLAY, DISPLAY MEMORY 和 DISPLAY FILE 在使用上有何不同?

DISPLAY 是信息显示命令之一。它可以显示库文件结构,也可显示库文件或索引文件的内容。该命令的最大特点具有分页显示功能。即一次只显示一个屏幕画面,叫一页。一页的容量相当于一个屏幕显示的信息,10 个记录。

DISPLAY MEMORY 用来显示已定义的内存变量。DISPLAY FILE 用来显示磁盘上的文件目录。

例如显示当前盘上的库文件

操作:

- DISPLAY FILE

#### 34. DELETE, PACK, ZAP 在功能上的区别是什么?

DELETE 命令是给记录加删除标志“\*”，并不是真正的删除。DELETE 命令指定的记录，这样做的目的显然是为了慎重。这种逻辑上的删除可以用 RECALL 命令还原成不带删除标志的记录。

PACK 命令则是从数据库文件中清除掉所有带删除标志的记录，真正实现了对库文件记录的最后删除，但速度较慢。

ZAP 命令则是快速清除库文件内容，从已打开的库文件中删除所有记录，这种删除是不可恢复的。

#### 35. SUM 和 TOTAL 命令有什么区别?

SUM 命令可以在当前工作库文件或当前工作索引文件内，对用户指定范围内满足给定条件的记录的数字型字段进行求和运算。

它可以对所有记录的所有数字型字段求和，也可按给定的条件对满足条件的记录的数字型字段求和。在求和时可以利用索引文件或排序文件来加快求和速度。

而 TOTAL 命令是将已打开的库文件的某些数字型字段，按照关键字相同的原则，进行汇总，汇总结果放在一个汇总库内，而汇总库也是一个库文件。它必须是按照关键字进行排序或索引过，换言之，TOTAL 命令执行之前必须按照关键字建立排序或索引文件。而 SUM 命令没有这个要求。TOTAL 命令的结果是一个库文件，SUM 命令的结果是一个数值。

#### 36. dBASE III 的 RUN 命令应用在哪些方面?

dBASE III 的 RUN 命令是在 dBASE III 状态下，允许用户在 dBASE III 命令文件中执行扩展名为“.EXE”、“.COM”的文件和 CC-DOS 的一些命令，一旦执行结束便返回到 dBASE III 状态下。使用该命令，要求用户内存有足够空间，若有效内存空间不够，则外部程序调不到内存中来。一般要求内存空间为 64KB 较好。

该命令格式：

• RUN <外部文件名> <参数>

或

! <外部文件名> <参数> (“!”之后不能有空格)

其中<外部文件名>指以“.EXE”、“.COM”为扩展名文件或 DOS 的内部、外部命令。<参数>指源程序名，扩展名或者子目录名等。

RUN 命令为 dBASE III 提供了十分方便的操作系统接口，为用户提供了一个良好的处理环境。该命令可应用于：

##### (1) 在 dBASE III 状态下编辑源程序

在 RUN/! 命令下直接调用 EDLIN 或 WS 等编辑程序。使用方法如下：

RUN EDLIN 源程序名.扩展名

这样便进入了行编辑程序状态，即达到了编辑源程序的目的，又避免了数据丢失(因 dBASE III 的字处理命令所编辑的文件最大长度 4KB)及用其他方法时来回进入。退出 dBASE III 状态所造成的时间上的浪费。

##### (2) 可使用多个目录共享一个 dBASE III 版本

如 dBASE III 在一子目录 USER1 中，现欲在与 USER1 同层次 USER2 子目录中使用 dBASE III，则可用 RUN 命令调用：

• RUN CD\USER1

• RUN CD\USER2

这样便可达到子目录 USER1 与 USER2 共享一个 dBASE III 版本的目的。

##### (3) 与外部语言程序连接

可以通过 RUN 命令调用其他高级语言程序。对编译类高

级语言，经过链接可得到 .EXE 文件，故可直接调用。而对解释型语言不能直接运行其程序，但是可以先调用该语言的解释程序再运行其有关应用程序。

例如，要调用名为 A1.BAS 的 GWBASIC 程序，可进行如下操作：

• RUN GWBASIC A1

若要返回到 dBASE III 状态下，则在 A1 程序的出口处使用 SYSYSTEM 即可。

##### (4) 执行 DOS 的一些命令速度快

在 dBASE III 状态下能很方便地使用 DOS 的一些命令，速度比在 DOS 系统下要快得多。

#### 37. 数据库文件用完为什么要关闭?

如果忘记关闭库文件，一般情况下不会出现什么大问题，因为 dBASE III 具有自动关闭库文件的功能。但在某些特殊情况下，不关闭文件，就会出现一些我们意想不到的“怪现象”。

例如，对一个软盘上的库文件进行数据修改，修改完后，未关闭文件就从驱动器抽出，再重新进入 dBASE III，结果会发现要修改的记录内容并未变化。其原因就是因为对库文件内容的修改是在计算机内存中进行的，并非修改完即自动转入磁盘，只有在关闭文件之后，才能将修改内容存到磁盘文件中。

再例如，删除操作：

• USE <库文件名>

• DELETE ALL FOR RECNO() = 4

3 记录 deleted

• PACK

×××记录 已拷贝

上述操作过程，似乎 3 个记录已被删除，但这种删除也只是在内存中进行的，并未真正地写到磁盘上，因为还未关闭文件。如果我们误认为记录真的被删除了，拿走了磁盘后就退出系统，或换另一张盘，作其他数据操作，我们再打开经过删除操作的该文件，发现记录号 = 4 的记录并未删除掉。

因此，当对库文件内容进行过某些数据操作之后，特别是变动了库文件内容，应及时关闭库文件，以确保这种数据操作的有效性。

#### 38. 关闭数据库常用的方法有哪些?

在汉字 dBASE III 中，输入和修改数据经常使用的命令：APPEND [BLANK]，INSERT [BLANK] [BEFORE]，BROWSE，DELETE，PACK 等。有关这几条命令的执行特点是，每执行一条命令后，才将前一条命令执行的结果真正写入磁盘。

关闭数据库常用的方法有：

(1) 在单工作区方式下，打开一个数据库文件，利用打开新文件时，自动关闭以前已打开的数据库的功能，关闭前一个文件。

(2) 使用不带参数的 USE 命令，关闭现用数据库文件。

(3) 使用 CLOSE DATABASE 命令，关闭所有打开数据库文件、索引文件和格式文件。

(4) 使用 CLEAR ALL 命令，关闭所有打开的数据库文件，同时释放所有的内存变量。

(5) 使用 QUIT 命令。该命令对数据进行一次清理，该存盘的存盘，所以在退出 dBASE III 之前应执行该命令，而不应强行关机。

以上几种办法，可根据不同的需要选择使用。

39. 调试程序过程中要进行其他操作为什么必须先关闭数据库?

在 dBASE III 状态下调试命令文件的过程中,若将数据盘放在软盘驱动器中,并且未关闭所有的数据库,这时若在该驱动器中换入另外的盘片,执行其他的操作,可能会将该盘上所有的内容全部删除(包括子目录),而会写上刚刚插入的数据盘上的全部内容,但无真正内容。尤其是运行 DOS 命令时很容易发生这一现象,造成数据丢失。

存在这一现象的主要原因是在对数据库进行操作时,打开数据库后,便可对它进行存入、修改、删除、显示等操作,这一系列工作不是在当前使用的驱动器上进行的,而是在微机的内存中进行的。操作完毕,所有的结果还在内存中,必须在完成关闭数据库操作时,才将该内容存到盘上。否则盘上的内容不变。当然,如果退出 dBASE III 状态,微机将自动关闭所在数据,释放所有内存变量,然后回到 DOS 状态,这一过程一般用户是感觉不出来的。

万一发生上述现象,将被破坏的磁盘不要再进行其他操作,想办法进行恢复。

#### 40. 退出 dBASE III 系统时为什么一定要执行 QUIT 命令?

主要原因是由于 dBASE III (包括 dBASE II) 的许多命令并不是立即执行的,尤其是涉及外部设备的命令。比如 CTRL+W 的修改存盘,有时不马上执行操作,即是参照内存存储器 RAM 的使用等情况由系统随机管理。dBASE III 退出系统命令 QUIT 对系统环境进行一次清理,该存盘的存盘,对原有的信息还要进行一次反存处理。可见,退出 dBASE III 时一定要执行 QUIT 命令,切不可为了一时的方便不执行它而退机,或是直接把机器的电源关掉,否则常常会致使一些不该发生的事情发生。有时数据的丢失就是因为忽略了这一点。

#### 41. CLOSE DATABASES 和 USE 命令在功能上有什么区别?

CLOSE DATABASES 命令可以同时关闭所有工作区内的所有库文件。如果现在已经开辟 10 个工作区,要关闭 10 个工作区内的 10 个库文件只要执行:

• CLOSE DATABASES

命令一次就可以了。

而 USE 命令只能关闭当前工作区的库文件或索引文件。所以,要关闭 10 个工作区内的 10 个库文件,则要调用 10 次 USE 命令。

显然,CLOSE DATABASES 要比 USE 效率高。

#### 42. 为什么当执行 QUIT 命令时,有时系统显示:

"Invalid command . com

Insert command . com disk in drive A

Abort, Retry, Ignore?"错误信息?

出现这个问题往往是内存空间小所引起的。对于一般 512KB 的内存,当调入汉字库和打印驱动程序后,进入 dBASE III,这时内存空间仅剩 10KB 左右,当运行程序时很可能将 COMMAND . COM 的高端已破坏。因为当开机启动时,首先从系统盘中读出引导记录,然后将 IBMBIOS . COM 和 IBMDOS . COM 两个隐含文件读入内存,进行 DOS 初始化,最后将 COMMAND . COM 读入内存,并赋予之控制权。COMMAND . COM 对本身进行初始化后,便将其核心部分自举到内存的高地址端,这部分约占内存 12KB 左右,而在内存低地址端仅留一小部分,这部分包括处理 INT 22H, INT 23H, INT 24H 以及检查高端部分是否被覆盖。当用户应用程序调入时,系统并不对 COMMAND . COM 部分进行保护,若应用程序较大,则可能将其 COMMAND . COM 的高端部分覆盖掉。这样的内存分布可使用户得到最大的可能的内存空间。当退出用户应用程序返回到操作系统之前,COMMAND . COM 的驻留部分检查

其高端部分是否被覆盖,若已覆盖,则重新从当前盘读出 COMMAND . COM 的高端部分自举到内存的高地址端,然后回到系统提示。同样的道理,从 dBASE III 退出时,系统发现若 COMMAND . COM 的高端已被破坏,则重新在当前盘调入 COMMAND . COM 的高端,如果系统启动盘和当前 dBASE III 系统盘不是同一版本的操作系统,因此就出现问题里的显示信息。

#### 43. 宏代换函数一般应用在哪些方面?

##### (1) 代替一条命令

当程序中要重复用到某一条命令时,可将该命令存放到一个内存变量中,在需要时用宏代换引用即可,从而省去了反复书写同一命令的麻烦。例如程序中要反复用到删除某一条记录的命令,则先将删除命令存入内存变量之中:

• STORE "DELECTE RECORD" TO D

在需要删除第 5 条记录的位置,使用宏代换

• &D 5

在需要删除第 2 条记录的位置,使用宏代换

• &D 2

这等效于使用了 DELECTE RECORD 5 或 DELECTE RECORD 2

##### (2) 代替字段名

当字段名是一个变化的量时,应把字段名存入内存变量之中,在需要的时候用宏代换引用。例如,对于一个图书管理系统,应该在给出书名,或者给出作者名,或者给出编号的情况下都能找出要找的书。那么是否要为每一种情况设计一段程序呢?实际上没有这个必要,只要一段程序就可以解决问题:

ACCEPT "请输入要查找的字段名:" TO F

ACCEPT "请输入要查找的字段内容:" TO S

LOCATE ALL FOR &F=S

若某甲输入的字段名是"书名",输入的字段内容是"dBASE III 综合应用",则 LOCATE 命令的条件表达式经代换后就变成 FOR 书名="dBASE III 综合应用"。若某乙输入的字段名是"作者名",输入的字段内容是"王秉湖",则代换后的条件表达式为 FOR 作者="王秉湖"。可见,用宏代换可以将一个表达式变为多个。

##### (3) 代替文件名

当文件名是一个变化的量时,就必须用到宏代换。例如,根据用户需要在程序运行中打开某个数据库文件时,则应该使用如下命令:

• ACCEPT "请输入库文件名" TO FILE

• USE &FILE

##### (4) 将字符型数据转换为数据型数据

& 函数可将一个字符型数字转化为一个数字型量,具有类似 VAL() 函数的功能。

例如: • STORE "123.45" TO X

• Y=&X

• ? TYPE("Y")

N

##### (5) 代替字符串

对程序中重复出现的字符串,用 & 函数可减少程序量;人机对话时,& 的使用可增加程序的灵活性。例如:主菜单有 8 个功能模块供用户选择,有两种编写方式:一种用 CASE 语句,8 个模块就必须对应 8 段 CASE 语句;第二种用 & 函数方法,仅用很少语句就能实现上述功能:



DO WHILE .T.

PH=" "

@ 5,1 SAY "请送功能模块号:" GET PH

IF PH(="5"

DO 文件名 &PH

EXIT

ELSE

LOOP

ENDIF

ENDDO

这种宏代换的方法,当功能模块增加后,程序也无须做什么修改。

#### (6)形成多维数组

编制程序过程中,使用数组变量能大幅度地缩减程序。尤其是编制任意二维统计模块时没有 A(I,J)数组的应用,程序的实现会相当复杂。在 dBASE III 中,系统用“.”区分宏代换变量结束。

例如:

A12="HOW ARE YOU?"

I="1"

J="2"

? A&I. &J

这里 A&I. &J 表示了 A12 变量,用“.”分割了 I 和 J 两个变量的代换。多维数组也同样可用此法实现。

#### (7)在 FIND 命令中使用变量

FIND 命令中不能直接使用变量,必须用宏代换来引用。

例如:

• STORE "张平" TO NAME

• FIND & NAME

通过这种变通即可在命令中使用变量。

尽管宏代换的用途很多,但只要了解它的本质,就可以掌握要领。它的本质就是去掉字符型内存变量的引号。去掉引号的对象,若在命令行的位置上,则被当作一条命令;若在字段名的位置上,则被当作一个字段名;若在文件名的位置上,则被当作一个文件名等等。

宏代换尽管很有用,但它要降低程序的执行速度。这是因为程序在执行包含宏代换的语句行时是先代换而后执行。

#### 44. 使用宏代换应注意哪些问题?

(1)宏代换的字符型变量不允许有前置空格,否则,系统认为语法错误,而与后续的其他变量间,一定要留空格。

例如:

• 运算符="\*"

• STORE 44 &.运算符 4 TO A

44 \* 4

(2)变量宏代换后,原变量内容是数字,就变成数值;是字符串,就失去了字符串的引号。某些情况下,变量宏代换后仍需要有引号,为解决此矛盾,采用双层引号将变量内容括起来,但双层引号不能相同。

(3)在 DO WHILE 语句的条件表达式中不能用宏代换。

(4)& 函数后必须紧跟一个已定义过的字符型存储变量,而不能是 D 型、N 型和 L 型。

#### 45. 调用 dBASE III 的函数时需要注意什么问题?

dBASE III 有 36 种函数,函数的一个重要特性就是函数类型。函数可以被当作一个常数、变量或表达式在程序中进行各种操作运算。但是任何一种数据操作运算都必须遵循着一条

原则:即只有相同类型的数据才能进行操作。在函数调用时需注意以下几点:

(1)函数本身的意义。每种函数都完成一定的功能,回送一个确定的数值因此可以把函数看成一种常数,变量或表达式,参加各种数据操作运算。

例如:

• ? SQRT(4)函数值为 2,这里 2 是函数的回送值。

(2)每种函数都具有一定的数据类型或属性,在对这些函数进行各种数据操作时,一定要注意每种函数的数据类型,否则就要出现操作语法错误。

(3)每种函数都要求一定数据类型的参数,不能出错。

例如:SQRT(B)就是错误的,因为参数“B”不是数字型,而是字符型数据。

(4)如果回送值是一个简单数值,它可以和其他数据进行运算。

例如:2+SQRT(4)

#### 46. 人机会话命令 WAIT、ACCEPT、INPUT 在用法上有什么不同?

人机会话命令 WAIT、ACCEPT、INPUT 三个命令在命令文件中的使用不同的地方是:

WAIT:作暂停或输入单个字符,不需要回车,快速响应命令。

例如:• WAIT "打印报表否?(Y/N)" TO W

显示:打印报表否?(Y/N)。于是,根据需要按 Y 或 N 键快速响应。

如按 Y,则打印。

• ? W (变量 W 显示 Y 的结果)

ACCEPT:输入多于一个字符时使用。

例如:• ACCEPT "请输入姓名:" TO N

显示:请输入姓名:(可根据具体姓名输入,然后在变量 N 中存储)

INPUT:除能输入字符和数字型数据外,还可输入逻辑型数据。但字符型需加引号,不如 ACCEPT 方便。INPUT 常用于输入数字型或输入逻辑型数据。

例如:• INPUT "请输入(0~5)选择功能:" TO M

显示:请输入(0~5)选择功能:

#### 47. READ 命令和 GO 命令怎样配合,对指定记录进行屏幕修改?

要修改某个记录,先用 GO 命令将记录指针移至该记录处,然后再用 READ 命令对其进行修改。用 READ 可以修改一条记录中的所有数据项,也可以只修改部分数据项,只要把格式式设计好,操作是比较容易的。

操作步骤:

(1)打开数据库文件 • USE <库文件名>

(2)打开格式文件 • SET FORMAT TO <格式文件名>.FMT)

(3)将记录定位 • GO RECORD n

(4)用 READ 命令调用格式文件

• READ

(5)利用全屏编辑键进行屏幕编辑修改

#### 48. RETURN、RETURN TO MASTER、QUIT、CANCEL 命令有什么区别?

在 dBASE III 命令文件结束方式的控制中,RETURN 为程序逻辑终止命令,当命令文件执行到 RETURN 命令时,自动地逻辑终止本次程序的运行。

若该程序是由其他主程序调用的,那么该子程序终止后,则自动地返回主程序;如果该程序是由操作员通过键盘输入调用的,那么该子程序终止之后,又自动返回键盘操作工作模式。

RETURN TO MASTER,在这种命令方式下,使得子程序无论在哪一层,当执行到 RETURN 时,都自动返回到最外层的主程序,而不是返回调用该程序的相邻主程序。最外层主程序是 dBASE III 模式下调用的第一个主程序。

QUIT 命令是退出 dBASE III 系统的命令。QUIT 命令的执行,将关闭 dBASE III 系统中的一切文件,其中包括库文件、命令文件等,并自动返回操作系统。

而 CANCEL 命令则是取消程序运行。在程序执行过程中,如果遇到 CANCEL 命令,就自动终止命令文件的执行,返回键盘操作工作模式。

49. SET TALK, SET STEP, SET ECHO, SET DEBUG 等命令在用法上有什么区别? 如何配合使用?

SET TALK ON/OFF 命令是 dBASE III 将执行的每一条命令结果显示/不显示在屏幕上,作为人机会话的开关控制。系统默认状态是 ON。

SET STEP ON/OFF 命令是 dBASE III 的一条程序调试命令。在程序运行过程中,每执行一条命令都自动停止/不停止,系统默认状态是 OFF。

SET ECHO ON/OFF 命令是 dBASE III 程序调试工具之一,其作用是可将运行中的命令文件的命令清单,通过屏幕显示或通过打印机打印出来。ON 状态时显示,OFF 状态时不显示。

而 SET DEBUG ON/OFF 命令也是 dBASE III 调试工具之一。它具有自动接通或切断打印机的功能。

SET DEBUG 应该和 SET ECHO、SET STEP、SET TALK 配合使用,使 ECHO、STEP、TALK 回送的命令、信息通过打印机打印机出来。

例如:

- SET DEBUG ON
- SET ECHO ON

此时,ECHO 给出的程序命令清单是送给打印机打印出来的。在这种状态下,当执行某个程序时,就可以对命令文件进行错误诊断。诊断方法是 ECHO 命令在程序运行的同时,给出命令文件中的每条命令,供检查是否出错。因为 DEBUG 是处于 ON 状态,所以 ECHO 送出的程序命令是通过打印机打印出来,而运行的结果,却是通过屏幕显示出来。一旦发现运行的结果有错,就可以从程序命令清单中查找哪些命令可能有错。

50. dBASE III 变量有哪几种? 它们之间有什么异同点?

dBASE III 有二种变量:内存变量和字段名变量。

内存变量被用来存放程序运行过程中所要用的各种中间数据。因此内存变量是一种临时工作单元,需要用时可以临时定义,不用时又可以释放掉,每个内存变量都是独立的。

内存变量有四种类型:

- (1) 字符型内存变量——用来存储字符串。字符串长度  $\leq 254$ 。
- (2) 数字型内存变量——用来存储数字。
- (3) 逻辑型内存变量——只能存储逻辑“真”或逻辑“假”。
- (4) 日期型内存变量——存储日期。日期总长度是 8 个字节。

字段名变量是构成库文件的最基本数据单元。同一个库

文件中的各个数据项在物理意义上是有关联的。字段名变量的数据类型有字符型、数字型、逻辑型、明细型以及日期型。

51. 内存变量的主要用途是什么?

内存变量是独立存储于数据结构之外的有名数据项,存储于内存变量区。内存变量区中的数据用来控制程序执行过程中各个进程的执行,并且每个数据项都有一个内存变量名指向它。这样一组内存变量就好像一个总是打开的没有任何字段结构的数据库文件。

内存变量最主要的用途有如下一些:

- (1) 保存经常使用的常数;
- (2) 保存操作者键入的信息;
- (3) 作为数据进入数据文件的缓冲区;
- (4) 保存表达式和函数的结果;
- (5) 作为计数器的暂存器。

内存变量的建立和赋值是在同一命令内完成的。内存变量的类型就是它们的数据类型,可分为字符型、日期型、逻辑型、数值型。

52. 建立和赋值内存变量的命令有哪些?

表 1—3 归纳了建立和赋值内存变量的命令。

表 1—3 建立和赋值内存变量的命令

命令	所能处理的内存变量的类型
= (赋值命令)	C, D, L, N①
ACCEPT TO	C
AVERAGE TO	N
COUNT TO	N
INPUT TO	C, D, L, N
RESTORE FROM	C, D, L, N
STORE	C, D, L, N
SUM TO	N
WAIT TO	C

表中 C 表示字符型内存变量, D 表示日期型内存变量, L 表示逻辑型内存变量, N 表示数字型内存变量。

53. 使用内存变量时应注意什么?

当许多内存变量被来回地建立和恢复时,内存变量区可能会被划成碎片。这样会使实际可用的内存变量区越来越小。更为严重的是,当没有内存变量区而又企图建立内存变量时,就会导致危急情况。如果你的内存变量出了毛病而又找不出原因,就应该考虑到这一点。此时,应把所有活动内存变量保存到一个内存变量文件上去,然后从这个文件上重新调入内存,这样就可以重新构造内存变量区,消灭碎片现象。相应的操作命令及格式如下:

SAVE TO <内存变量文件名>  
RESTORE FROM <内存变量文件名>

在汉字 dBASE III 中,内存变量分为两大类,一类为局部型变量,一类为全局型变量。在圆点“.”提示符状态下设定的所有变量都被自动定义为全局型变量。在命令文件中设定的变量,如果不加说明,则一律被认定为局部变量。局部变量只在同一级命令文件中有效。一旦遇到 RETURN 命令,局部变量将全部自动释放。因此,对于在程序各级命令中都要使用的变量,都必须事先用 PUBLIC 命令说明为全局变量。

若已把一个变量设定为全局变量,则不能先给定初值,否则系统将给出错误信息。但对一个已经说明为全局型的且已赋过值的变量,可以在另一级命令文件中重新用 PUBLIC 说明之。



## 新书与软件

## TANGO 印刷线路板 自动布线软件包

在计算机迅速发展的今天,各种应用软件也越来越多,已成为科研、生产等工作中不可缺少的重要工具。计算机辅助设计和设计自动化是当今电子工程技术必不可少的重要支柱之一,印制线路板图的设计自动化是其中重要的一部份,它彻底解决了传统手工绘图效率低、错误多、集成化低、质量不高的缺点。

印制线路板在电气设备中,从人造卫星的航天电子设备到家庭最小最简单的电器都无法离开印制线路板,它起着各种规模的电子元件和机械结构的双重连接作用,因而印制线路板的正确设计是电子工程中一项极为重要的基础部件,它必须体现优良的电气性能、又具有合理的工艺结构,二者缺一不可,否则达不到预期目的。

TANGO 印制线路板自动布线软件包是美国 ACCEL Technologies Inc 在 1987 年推广的最新一代自动布线软件包,

该软件包的设计者具有多年印制线路板图设计的工作经验,并充分吸取了其它印制板图设计 CAD 的优点,因而 TANGO 软件包可以满足众多工程技术人员在实际工作中的要求。它可以在有限的面积内,根据设计者的要求设计出数量庞大和多种线宽的电气连接线、多种规格和形状的焊盘和连接孔,精确地设计出符合高频中要求的特定阻抗的印制线路板和微带电路。

TANGO 软件包包括电原理图编辑 TANGO—SCHEMATIC、电路板设计 TANGO—PCB、线路板自动布线设计 TANGO—ROUTE 等。

为了便于广大电子工程技术人员迅速掌握有关 TANGO 的用法。本刊将出版《印制线路板图的计算机辅助设计》专著,供 TANGO 用户参考(并可提供 24 针打印机,如 1724, 3070……等打印驱动软件)。

## 电子工业出版社新书介绍

《电子测量仪器选购指南》定价 35.00 元

本《指南》汇集了我国和美、英、法、日、西德等十多个国家的主要电子测量仪器厂商生产的有关产品,给出每一种仪器的生产厂商型号、主要技术性能。

《最新简明中外集成电路互换型号手册》定价 18.00 元

本手册收集中国、美国、日本、西德等 150 家集成电路和元件厂家,45000 个最新型号互换信息及技术参数、管脚应用图,提供中外各种集成电路型号互换资料、接口电路,运算放大器脚图,国内外集成电路型号互换表,中外电阻、电位器、电容器。适合电子产品设计,家电维修的计算机和半导体研究、元器件供销工作人员及大专院校师生参考。

《家用电器元器件手册》(精装) 定价 56.00 元

《传感器敏感元器件大全》(精装) 定价 64.00 元

《标准集成电路数据手册——TTL 集成电路》(精装)定价 31.50 元

《标准集成电路数据手册——运算放大器》(精装)定价 32.00 元

《国内外功率晶体管实用手册》上(精装)定价 21.00 元

《国内外功率晶体管实用手册》下(精装)定价 45.00 元

以上六种图书是大型的、完整的电子工具系列丛书。可供研究、设计、生产、技术开发等部门有关专业的工程技术人员、大专院校有关专业教师、学生、实验室科研及广大无线电、电子爱好者、家用电器、电子修理行业的同志阅读、查阅。

《dBASE 应用解疑》定价 5.00 元

《IBM PC 编程指南》定价 7.00 元

《DOS/BIOS 使用详解》定价 8.00 元

《MS—DOS 高水平程序设计》定价 5.50 元

以上四种图书是 dBASE 应用、IBM PC 机、C 语言和汇编语言使用必备用书、该书编排合理、启发性强、由浅入深、为计算机系统开发提供了得力工具、是开发人员开阔思想解决疑难、提高水平的必读之书。

邮购办法:邮局汇款 北京万寿路电子工业出版社发行部邮购科

银行汇款 开户行北京市工商银行翠微路分理处帐号 661036—40

注:购书清单写在汇款单“附言栏”内写明书名、册数、收书人详细地址和邮政编码。

本部电话:81.3693 电挂:3101 邮政编码:100036

# 电子工业出版社软件部发行 CCED V3.0 中文字表编辑软件

软件开发:朱崇君

CCED 是针对中文编辑的特点开发而成的编辑软件。特点:除吸收了各种字处理软件的优点外,还增加了表格处理功能和数据运算功能。使用户在同一文件中既可做表格,又可写文章。CCED 还能直接调用 dBASE 的数据文件,形成各种表格输出。

在文字处理方面,CCED 除了具备一般编辑软件的基本功能(如:删除、恢复、搜索、替换、复制、对中等)外,还包括了 EDLIN、PE 及汉字 Wordstar 的全部优点,集 PE 中跨越文件的各种块操作功能与 Wordstar 中的排版功能、信函批处理功能于一身。CCED 中的块操作可以实现文件中或文件之间文字块的复制、移动或覆盖。块打印可帮助您完成诸如根据通信录打印贺年卡的任务。段重组可支持中英文混合排版,在保持单词完整性的前提下,排版结果可使行的首尾对齐,并妥善地处理了行首、尾的标点符号。CCED 还具有 EDLIN 中的上行复制功能,并将其扩展到全屏幕操作。

在表格编辑方面,用户可在通过光标画线制表,也可以利用程序本身的功能自动地生成表格。对于画好的表格,可以调整宽度。在表格中插入一行时,纵向表格线能自动保持连贯。在字处理中所用的编辑功能均可以在表格编辑中使用。更必表内数据时,采用线保护状态,可使原表格保持不

变。为了能同其它软件交换数据,CCED 所制作的表格文件为纯粹的文本文件,而不带任何隐含控制符。

在数据计算方面,用户可在编辑版面的任何位置列算式,进行计算。对于表格中的数据还可以列写公式按行或列成批计算。

CCED 的另一个突出特点是能够调出 dBASE 数据库,产生各种报表输出。dBASE 是微机上常用的数据库系统,其报表打印一直很困难,往往需要用户编制许多繁琐的程序才能实现。而所编的程序常常是针对某一种报表,一旦要改动或产生新的报表,就得修改或新编程序。使用 CCED 可使 dBASE 的报表自动输出。无论结构多么复杂的报表,用户只要用 CCED 画出一个报表的样子(称之为样本表格),就能从数据库中提取数据对其填表输出。样本表格可以长期保存,亦可随时修改。最初的样本表格还可以由程序自动生成,其操作十分简单。生成样本表头时,字段名能够自动排成整齐。由于采用了这种样本表格技术,更为管理系统开发提供了方便。

联系地址:北京 173 信箱电子工业出版社软件部  
邮政编码:100036

## 北京华源技贸商行 经营:中华学习机、苹果机系列产品

中华学习机 910—930 元(保修三年)  
天坛学习机 995 元(保修一年)  
软盘驱动器:东芝 590 元  
Superb 640 元  
中华学习机各种软件 800 余种  
软盘:(学生盘)、ACT、DATAACE、3M、Maxell janus)  
2.5~7.5 元  
单色显示器 520—550 元  
中华学习机扩充卡 打印卡 Z80 打印卡  
驱动卡 双驱动卡等

游戏棒(开关量,模拟量) 47—72 元

中华学习机各种资料、书籍 30 余种

凡在本部购买学习机者,将得到保修、免费培训的优惠,欢迎来函索取目录,本部代修学习机、驱动器,并可为学校配置中华学习机网络。批发面议。

地址:北京东城沙滩五四大街 52 号 邮政编码:100009  
电话:512.7308 开户行:工商行崇外分理处中都信用

社

帐户:华源技贸商行 帐号:中都 03040

联系人:于民 张小键

## 浙江温州市精密电子仪器经营部

地址:温州市飞霞南路龟湖路  
邮编:325004 电挂:5072 电话 336072

开户:工商五马辨 帐号:1567892

供应(单价)EPROM2716/32/64/128 均 18 元;27256,25 元;RAM6116,15 元;6117,10 元;6264,32 元;8085/8031,15 元;Z80CPU,7.8 元;CTC/PIO 均 8.8 元;SIO,12 元;DMA,20 元;8039/8035/8255,均 10.5 元;8155,15 元;8243,12.5 元;8253,14 元;8279,22.5 元;8257/8259,18 元;8251,16.5 元;CD4508/4514/4515,4.8 元;光隔 P541—1/521—1,2 元;P541—2,3.5 元;P521—3,6 元;P521—4,7.5 元;40P 集成插座,0.65 元;晶振 2M,2.8 元;3.579545M,2 元;4M,3.2 元;2N3055,6.9 元;2SC1008,0.65 元,全桥 5A/50V,2.6 元;10A/50V,8.32 元;2uF/1KV,9 元,数字电压表 250V/电流表 5A,均 143 元。以上邮资 2 元/次,另有数字式实验多用仪,功能有(1)1MHZ 频率计(2)100PF~10000uF 电容计(3)三路可调稳压电源 1.2V~26V/1.5A 数显及±5V~±15V/1.5A,价 496 元,邮资 12.5 元/台,另有系列 74LS,CD4000 电路及数百种其它晶体管,集成电路,继电器,阻容等均为进口器件,函索价目表者,请用贴足邮票信封写清回邮地址寄我部。

### 求 援

• PC—8100 图纸

联系地址:新疆哈密桥梁厂薄如兰转

张子芳

邮政编码:839001

• 有关 MZ—731 选配磁盘驱动器的方法  
及资料

联系地址:江苏油田运输船一、李志明

邮政编码:225261

# 致参加“普及型 PC 个人用户 软件交流联谊活动”的用户

农科院计算中心

自从今年一月份“电子与电脑”杂志刊登组织“普及型 PC 机个人用户软件交流联谊活动”以来,得到了社会各界人士,尤其是“电子与电脑”杂志用户大力支持和充分的肯定,作为联谊点的挂靠单位,我们中国农科院计算中心培训部全体同志能有这样一个为广大用户直接服务的机会,能为 PC 机个人用户获得自己需要的软件,提高用户的应用水平作一些力所能及的工作感到十分高兴。

到目前为止,我们先后收到来自各行各业 PC 机个人用户要求参加该项活动的已达 120 多人。大家积极主动地提供了自己拥有的软件清单;反映了希望通过联谊活动得到自己所缺乏的软件愿望和要求;提出了在使用中碰到的许多困难和问题;对搞好联谊活动提了殷切的希望。大家的积极支持是我们办好联谊活动的根本条件。针对大家的来信、来函、来电,我们已组织人将有关材料进行了计算机管理。归纳起来大家普遍的要求:一是有了机器,由于硬件资料缺乏,对其性能了解较少,操作使用尚有一定困难,所以希望提供有关的硬件资料信息。二是有了机器,由于缺乏软件,机器的功能不能很好的发挥,利用率低,所以希望相互交流交换软件。三是有了软件由于缺乏软件使用说明的资料,致使软件不能得到利用,所以希望提供有关软件使用的资料来源。针对上述问题和大家的普遍要求,下一步联谊活动的搞法我们准备作如下的安排:

## 1. 关于 PC 机系统硬件资料,哪些用户需要什么样的资

(上接第 31 页)

只要继续输入下一字的编码,此字就会被自动选中作为输入。

用 CPC 卡上的五笔字型输入法输入的汉字,不需经任何处理即可方便地用于编程、打印或用于 CPC 卡上的屏幕编辑软件。

## 三、汉字全屏编辑功能

使用 PR#8 进入汉字后,按【F5】键即可进入汉字全屏编辑状态,进行文字的插入、删除等编辑,并可存盘或打印。编辑功能如下:

光标上移	
↑	
光标左移 ←	→ 光标右移
↓	
光标下移	
【CTRL-N】	在光标所在行开一新行
【CTRL-D】	删除光标所指字符
【CTRL-X】	删除光标所在行
【CTRL-I】或【TAB】	插入/覆盖状态转换键
【CTRL-Q】【西文】	快速/正常编辑转换键
【QUIT】、【T】	设置字号
【QUIT】、【D】	删除整个文件

料,希望直接来函给我培训部说明,目前我部拥有 120 种微机各种硬件、软件资料,北京市用户可直接来我培训部选购,外地用户可邮购,但资料的名称邮政编码、单位地址、收件人姓名务必写清楚。

2. 关于软件的交流,大家所需的软件覆盖面较大,既有系统类软件,也有应用软件类,既有文字处理方面的软件,也有统计、制表、制图等方面的软件,既有输入方法,也有教学软件,甚至不少用户需娱乐方面的游戏软件等等。所以单靠我们这里一个点很难满足大家的要求,但作为大家常用的软件我们陆续公布常用软件清单(附后见 48 页),供大家交流。原则仍然是北京市用户可持软盘直接由我部获取。外地用户写明所需软件名称,办理邮寄,这是一种办法。我们已初步与《电子与电脑》杂志编辑部商量决定在今年年底举办一次北京地区用户的联谊会,通过联谊活动交流使用的经验、交流软件。借此摸索经验,然后扩大联谊会范围和内容。什么时候举办,以通知为准。

3. 关于使用中的问题。我们正在把大家使用中的问题进行整理、归纳,准备将带有普遍性的问题组织有关方面专业人员,通过《电子与电脑》杂志这一园地回答给用户。

“第一批常用软件清单”附后:

联系人:中国农业科学院计算中心培训部王路敬

邮政编码:100081

通讯地址:北京海淀区白石桥路 30 号

【QUIT】、【S】	文件存入磁盘
【QUIT】、【L】	文件调入内存
【QUIT】、【P】	打印盘上文件
【QUIT】、【QUIT】	退出屏幕编辑

当打印文件时,系统会询问打印参数。打印参数如下:

T:字号 C:字距 N:行距 H:行内字数

回答时,各参数用“,”分开。取值范围为:1≤T≤15;0≤C≤254;0≤N≤254;0≤H≤254。软件默认值为:T=6,C=4,N=4,H=40。另外字号可在编辑文章时直接设置,从而实现同一文章多种字号。

在屏幕编辑软件中也可使用五笔字型输入法。使用方法同前介绍。

中华学习机 CPC 卡五笔字型输入法采用独特的检索方式,使时间和空间的占用都减到最低。从时间来说,当用户输入完五笔码后,基本感觉不到输入后的编码检索时间即可出字,为实现高速盲打创造了条件;从空间来说,CPC 卡只占用了用户空间高端的 \$9000~\$91FF 的 512 个字节,并用 HIMEM 保护起来。对用户来说,不会有什么明显影响。

中华学习机 CPC 卡集三种重要功能为一体,使用户能在中华学习机上以低代价换得高效的文字处理功能。随着 CPC 卡今后不断的改进,中华学习机也将被升级为中文处理机。



常用交流软件清单

版本号	适用机型	所需盘数	版本号	适用机型	所需盘数	版本号	适用机型	所需盘数	版本号	适用机型	所需盘数
CCDOSS	A	1				MSC			版本号	适用机型	所需盘数
CCDOS	A	1	硬盘加密	A&C	1	COBOL I	A&C	1	V2.0	A&C	9
CCDOS	A	1	加密、拨号	A&C	1	FORTRAN	A&C	1		A&C	2
CCDOS	C	1	解密软件(专用)	A&C	1	MS-FORTRAN	A	2		A	2
CCDOS	A&C	5	解密软件(专用)	A	1	FOXBASE PLUS	A&C	2	V2.00	A&C	2
GWDOSS	A&C	7~8	WS(字处理)	C	1	DBASE II	A	1		A	1
LLDOS(双音双拼)	A	2	WS	D	1					C	1
MSDOS	C	1	WS	286	1	DBASE II PLUS	A&C	6		A&C	6
CLIB24 * 24(汉字库)	A&C	6	XE	V2.8	2	全汉化 DBASE II	V1.01	1		A&C	1
CLIB32 * 32	A&C	11	XE	CEGA	2	汉字编译 DBASE II		2			2
CLIB48 * 48	A&C	12	HW(字处理)	A&C	1	编译 DBASE II		2		C	2
各种打印驱动程序	A&C	25	HW	CEGA	1	DBASE II 辅助系统		4		C	4
高级打字驱动程序(PS)	A&C	1	CW(词语输入方法)	CEGA	2	DLTIL		7		C	7
LP24	A&C	1	CVICH SINNER		5	INFORMIX		4		C	4
对折打印	A&C	6	LOTUS(图形管理系统)	A	1	DOFRMAT		1		C	1
IBM 检测程序	A&C	1	LOTUS	C	3	编译 PASCAL		2			2
	A&C	1	LOTUS(教学盘)	C	1	R:BASE 4000(库管理系统)	V1.11	2		C	2
PCTOOLS(汉化)	A&C	1	MAKER(报表汉化)	V2.1	A	日常事务处理及生物统计		4		A	4
PCTOOLS(汉化)	A&C	1	汉化 BCT 制表		A	统计		1		C	1
PCTOOLS(西文)	A&C	1	CCED 汉字编辑软件	A&C	3	DUTIL		2			2
PCTOOLS(西文)	A&C	4	高级智能制表软件(80元)	A&C	1	STATPAK 统计	V3.11	1		A&C	2
BOX(工具箱)	A&C	5	AUTO HCAD(汉化计算机辅助设计)	V2.2	A&C	STATISTICS 多功能统计		1		A&C	1
DMA(磁盘管理恢复文件)	A&C	1	AUTO CAD		A&C	科研管理软件		2		A&C	2
五笔字型	A	2	ACAD(汉化计算机辅助设计)	V2.17	A&C	日常管理软件		2		A&C	2
五笔字型造字软件	A	2	OFFICE		A&C	通用工资管理软件		1		A&C	1
五笔字型导硬盘	A	1	OFFICE		A	财务报表管理软件		1		A&C	1
五笔字型	C	1	OFFICE		C	卫生统计程序软件包		2		A&C	1
五笔字型学习盘	单显	2	OFFICE	D8-286B	2	人事软件		2		A&C	2
	彩显	2	OFFICE	286	2	中文干部管理档案		1		A&C	1
条形码学习软件		1	GCLISP(汉化)		CEGA	通用财会记帐;银行对帐软件		2		A&C	2
长城笔形码输入方法	GWBX-CGEA	1	TURBO PROLOG(系统盘)		A&C	检测 77 种;消 12 种病毒	V1.0	1		A&C	1
COPYWRITE	A&C	2	TURBO PASCAL	V5.0	A&C	检测 78 种;消 13 种病毒	V1.0	1		A&C	1
GW87	A&C	1	TURBO C	V2.0	A&C	检测 79 种;消 17 种病毒	V1.0	1		A&C	1
						WORDPERFECT(从国外引进)		5			5

注:适用机型一栏内凡标有“A”均适用于 IBM-PC、PC/XT,标有“C”的为彩色显示器高分辨率下运行的软件。