



一九九一年 ● 总期第81期

12

# 電子

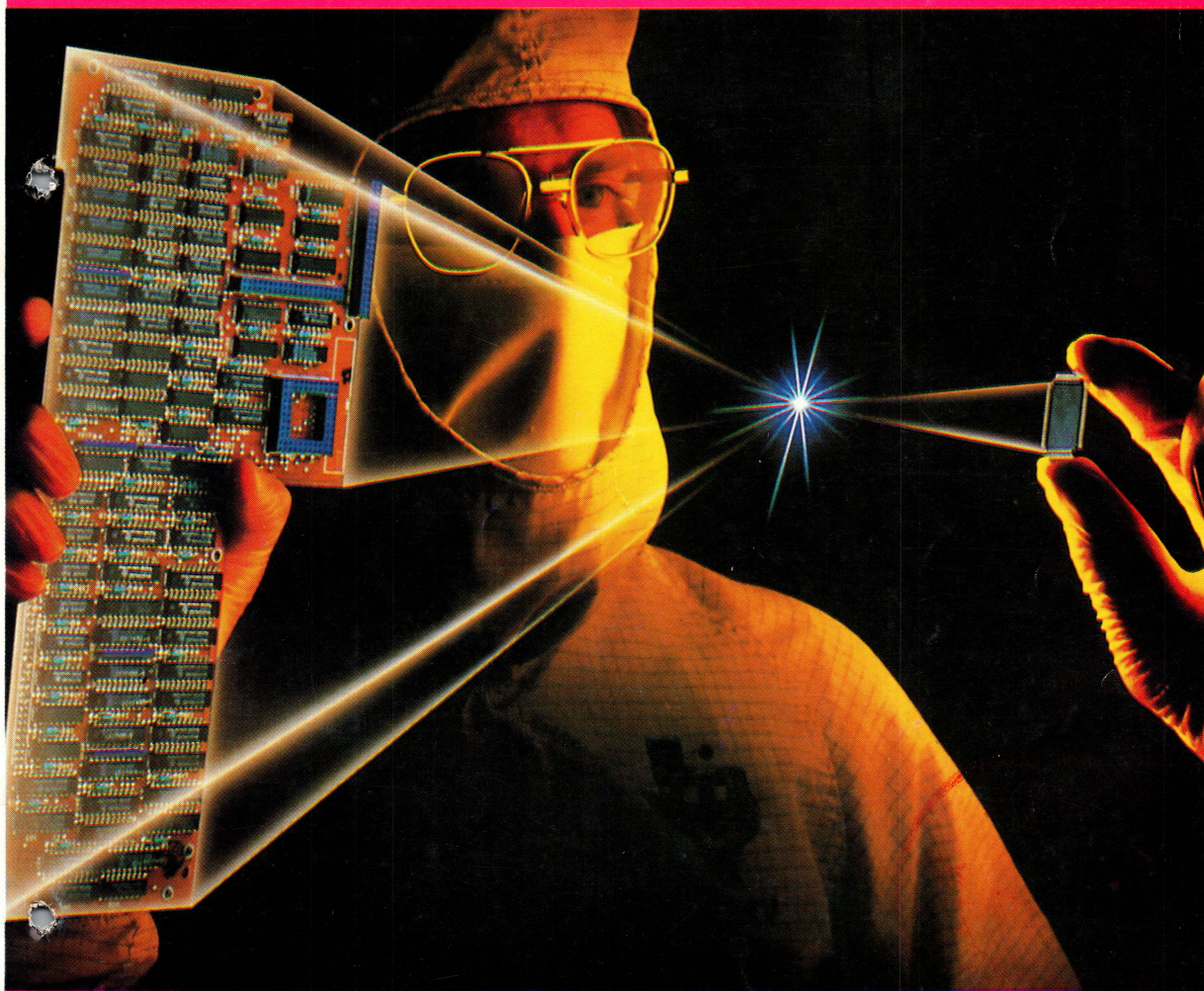
ISSN 1000-1077

744

# 與 電腦

● 电脑心脏

● 电脑心脏



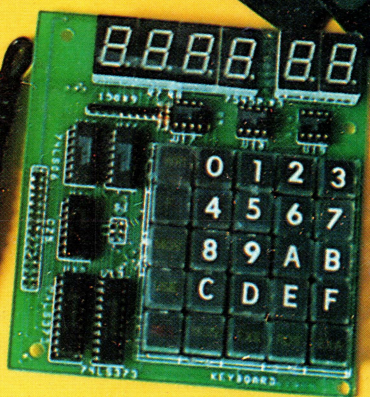
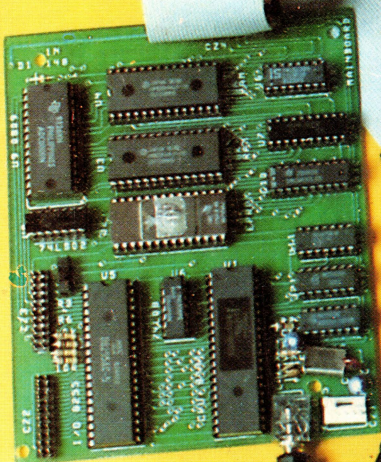
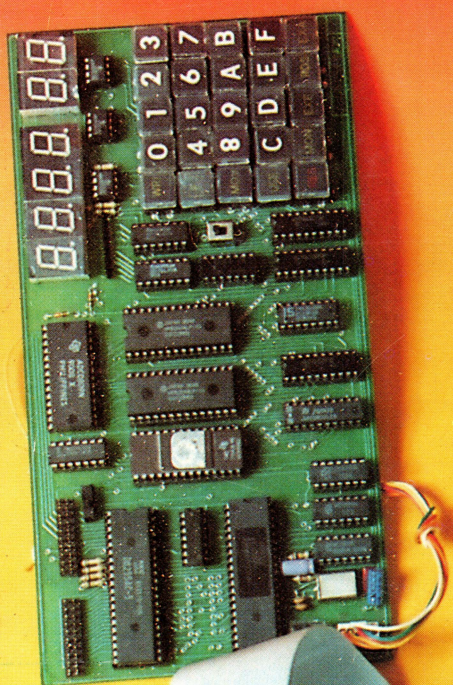
● 电脑心脏

● 电脑心脏

● ELECTRONICS AND COMPUTERS ●

# SCB—1 单片单板机!

武汉市尚吉电子研究所研制开发



武汉市尚吉电子研究所

联系地址: 武汉武昌珞瑜路37—3号

电话: 716138

邮政编码: 430070

电报挂号: 4561

开户银行: 工商银行洪山区办

帐号: 2604—67—016072001

● 国内代号: 2 888 定价: 0.95元



## 中国计算机软件与技术服务总公司(中软总公司)CS&S

市场开发部向您提供下列产品：

电脑大厦

### 长城计算机

产品名称	内存	软盘驱动器	硬盘	显示卡	显示器	零售件
GW0520EM	512K	360K * 1				5950 元
GW0520DH/10	640K	360K * 2	20M	CMGA		9900 元
GW0520/10	640K	360K * 2	20M	CMGA	GW500	12000 元
GW286BH/12	1M	1.44M+1.2M	40M	CVGA	GW500	16500 元
GW286EX/16	1M	1.44M+1.2M	40M	CVGA	GW500	22000 元
GW386SX/16	2M	1.44M+1.2M	40M	CVGA	GW500	26000 元
GW386/20	2M	360K * 1.2M	40M	CEGA	GW500	30000 元
GW386/25C	2M	1.44M+1.2M	40M	CVGA	GW500	33000 元

### 艺高(REGO)系列微机

型号	软盘 (KB)	硬盘 (MB)	显示器	显示控制	串并口	主板 RAM 基本配置	零售参考价 (人民币)
0520/10DM	360×2	NO	14"单显 720×350	主板 Hercules	主板 1串1并	640KB	5600 元
0520/10HM	360×2	20MB	14"单显 720×350	主板 Hercules	主板 1串1并	640KB	7600 元
0520/10CE	360×2	20MB	14"单显 1024×768	CEGA	主板 1串1并	640KB	9900 元
0520/10DE	360×2	20MB	14"彩显 VGA 1024×768	CEGA	主板 1串1并	640KB	12000 元
286/12-BH	360KB, 1.2MB * 1.44MB	40MB	14"彩显 VGA 1024×768	CEGA * CVGA	主板 2串1并	1MB	17500 元
286/16CV	360KB, 1.2MB * 1.44MB	40MB	14"彩显 VGA 1024×768	CEGA * CEGA	主板 2串1并	1MB	22000 元
386SX/16	1.44MB 1.2MB 外挂	40MB * 100MB	14"彩显 VGA 1024×768	CEGA * CVGA	主板 2串1并	1MB	24000 元
386SX/20 DV	1.44MB 1.2MB 外挂	40MB * 100MB	14"彩显 VGA 1024×768	SVGA	主板 2串1并	2MB	25000 元
386SX/20 CV	1.44MB 1.2MB 外挂	40MB * 100MB	14"彩显 VGA 1024×768	CVGA * CEGA	主板 2串1并	2MB	26000 元

打印机 LQ1600K, 5200 元/台, CR3240, 5800 元/台, AR3240, 6500 元/台。  
新教材《SQL 语言初学与入门》配有 SQL 语言中文自学软件 3 片盘, 90 元/套

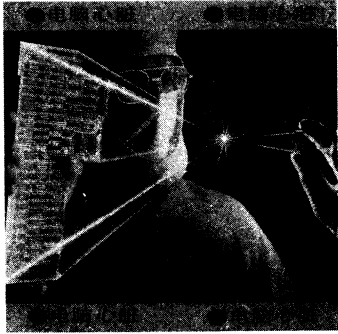
地址：北京海淀区学院南路 55 号电脑大厦，邮编：100081，乘 16 路汽车电脑大厦下车 电话：831.7722—1319  
开户行：中国工商银行海淀分理处，帐号：461187—79 联系人：温友良 寻呼：126 呼 3228



電子

ISSN 1000-1077

與電腦



ELECTRONICS AND COMPUTERS

12

一九九一年

总期第81期

# 電子與電腦

## 目 录

### • 综述 •

新型信息录入技术 一条码技术及其应用 ..... 王枫等(2)

### • PC 用户 •

dBASE III 实现 Foxbase 中 prompt 的菜单功能 ..... 毛维平(4)  
信息系统中模糊查找的设计 ..... 戚靖山 朱玉环(5)  
高密软盘的单驱动器拷贝 ..... 李毅(6)  
用汇编程序显示硬盘主引导记录 ..... 李日林(7)  
PC DOS 内存管理 ..... 陈宏(8)  
CC BIOS 2.13F 的小补丁 ..... 严仲武(9)  
中文弹出式窗口设计 ..... 姜咏江(10)  
一种实用新颖的排序方法 ..... 朱国江(12)

### • 学习机之友 •

多层次全能阴影菜单 ..... 黄文贵(13)  
虚线 ..... 刘裕(14)  
技巧两则 ..... 宋正荣(15)  
改进 MUSIC 的发音 ..... 熊波(15)  
中华学习机 RAM 监控及其应用 ..... 朴一兵(16)  
CEC-I 键控光标作图程序 ..... 陈国良(17)  
ProDOS 磁盘操作系统入门(续) ..... 廖凯(18)

### • C 语言初阶讲座 •

第十讲 文件管理 ..... 李文兵(20)

### • 学用单片机 •

SCB-1 型单板单片机的存储器改进 ..... 江琪(26)

MCS-51 单片机端口结构图示中的一个普遍错误 ..... 王修才(27)

### • 学装微电脑 •

串行 I/O 部件的制作与实验 ..... 易齐干(28)  
“微电脑控制技术”函授学习班总结 ..... 易齐干(30)

### • 电脑巧开发 •

将 MP-1 电脑的时基信号改为晶体振荡 ..... 邓文超(31)

### • 电脑游戏机 •

第二章 FBASIC 的基本语句 ..... 于春(32)

### • 维修经验谈 •

计算机电源修理 ..... 黄培鑫(36)  
四通 MS-2401 打字机使用维修经验 ..... 蔡长安(40)  
修理 CEC-I 型中华学习机软盘驱动器一例 ..... 赵升(41)  
谈谈尼龙打印色带的使用技巧 ..... 林斌(42)

### • 初级程序员级水平考试辅导讲座 •

计算机病毒防范问答 ..... 常晓冀(44)  
SCB-1 单片单板机简介 ..... (48)  
《电子与电脑》1991 年总目录 ..... (53)  
封面 电脑心脏  
封二 电源原理图  
封三 中软总公司市场开发部提供的产品  
封底 SCB-1 单片单板机

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:张丽

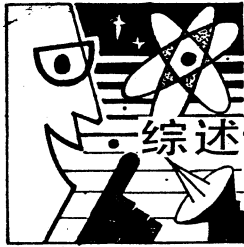
订购处:全国各地邮电局

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



# 新型信息录入技术

## ——条码技术及其应用

机械科学研究院 王枏 高群 董丽珠 赵文 过海春

随着现代科学技术的高度发展,计算机已经步入越来越多的领域,逐步从科研进入民用,进入到人们的生活中。人们已经发现电脑在民航售票处、图书馆、铁路托运站、银行中应用。不久的将来,人们还会发现:在医院挂号、商店买东西,甚至上班报到时,自己会直接地跟电脑打交道。原来由人做的事,现在由电脑做了。

本文将为您介绍条码技术及其应用,希望它能帮助您扩展知识、了解身边发生的事情,从而跟上时代的发展。

### 一、条码的种类及条码系统

条码(BarCode)是一种由一组宽窄不同、黑白相间的线条的排列组合、以代表一定字符的图形编码。这个编码可以是一组数字,或一串英文字母,或数字和字母的混合。数字和字母本身并不重要,重要的是它们所代表的内容,而计算机就是通过编码识别、处理它要完成的工作。

一般计算机工作是通过键盘输入,而条码技术是通过条码译码器和扫描器输入,扫描器的作用象镜子,它把“看到”的条码通过译码器输入给计算机,这样计算机就得到了和键盘输入同样的信息。所以,条形技术作为一种图形识别技术,具有方法简单、可靠性高、信息采集速度快、采集信息量大、信息的对应性强等特点,不仅大大降低了操作人员的工作强度,也减少了人为错误的发生。条码技术的开发,是人们对信息的采集和输入技术的突破。目前,采用自动识别技术,实现信息采集和输入的自动化已成为一种趋势。

由于条码技术的广泛应用,其种类很多,目前国际上使用的条码主要有六种:

1. 通用产品码(Universal Product Code)。该码 1973 年由美国统一代码委员会制定,简称 UPC 码,它主要用于美国、加拿大商品流通领域。

2. 国际物品码(International Article Number)。该码 1977 年由国际物品编码协会制定,简称 EAN 码。它主要用于世界各国和地区(美国、加拿大除外)的商品流通领域,该码与 UPC 码兼容。

3. 2/5 码(2 of 5 Bar Code)。它是一种仅能表示数字信息的高密度条码,它又分矩阵二五码和交叉二五码两种。

4. 39 码(Code 39)。该码由于可以表示英文字母,因而广泛用于工厂自动化和办公自动化领域。

5. 库德巴码(Codabar Code)。该码主要用于邮电、图书管理等领域,在美国该码还用于医疗卫生业。

6. 128 码(Code128)。它是一种与 ASCII 码完全对应的条码。

条码系统(Barcode System)是指由条码符号、条码符号的生成(印刷、打印等方式)、条码符号的扫描以及译码等部分组成的自动识别系统。该系统是以大规模集成电路技术、新

型光电传感技术、精密机械加工技术、印刷、打印技术以及计算机应用技术作为支撑。因此条码系统是一种综合性的高技术系统。

### 二、条码技术的特点

从上面介绍,我们知道条码技术实际上是计算机信息输入的一种方式。在计算机信息输入所采用的多种自动识别技术中,条码技术与其它识别技术相比,其特点表现为下面七个方面:

#### 1. 方法简单、成本低

条码作为一种图形信息技术与其它技术(如文字识别技术)相比,有图形简单,易识别的特点。

#### 2. 信息采集速度快

信息采集速度是衡量信息采集水平的一个重要的指标。由于条码自身的黑白相间的线条正好对应于信号的高低电平,即逻辑信号的“1”和“0”,因而使用条码技术采集信息的速度快,一般译码时间不超过几毫秒,甚至可以更短。如日本东研公司已开发出能读出转动着的电风扇叶片上条码的解读者;

#### 3. 采集信息量大

条码自身具有很大的灵活性,它可进行多种编码,许多条码方法对其信息长度(一个条码所代表字符的位数)从理论上讲不受限制(实现上受限于译码器的能力,如光电扫描的幅度等)。由于条码图形简单这一特点,因而可以相对提高线条的密度;

#### 4. 可靠性高

利用普通键盘输入信息时,其字符的置换率为 800 分之一,且信息的可靠性受人为因素的影响甚大。其它一些自动识别技术,例如,文字识别技术,由于文字图形的不规范和复杂性,对其可靠性有所影响。再如,声音识别技术,在声音识别中,由于人为因素可使其可靠性下降。使用条码技术,其字符的置换率可达百万分之一以上,并且在计算机软件的支持下,其可靠性可进一步提高;

#### 5. 信息的对应性强

在生产、生活活动中,大量的信息是针对某一确定物体的,离开物体,信息本身则失去意义。由于条码可以印(贴)在物体或物体的包装物上,因而使物流信息和物品的对应性得以保证。这是条码的一大特点,因此有人把条码技术称之为物流信息技术;

#### 6. 大大降低劳动强度

手工信息录入是一项劳动强度很大的工作。然而条码技术是一种自动识别技术,在使用固定式条码译码器时,不需要人力,机器自动完成信息采集任务,在使用台式或便携式译码器时,操作人员仅做一些非常简单的工作,如扳动开关

等。因此使用条码技术能大大降低劳动强度；

#### 7. 应用范围大

条码技术自问世十余年来,其发展迅速是人们不曾预料的。在经济发达国家,十余年来,使得条码的有关产品系列化,以适应各方面的需要。如适用自动化流水线的固定式条码译码器,应用于办公自动化及图书管理等的台式译码器以及不受工作现场限制的便携式译码器已被人们广泛使用。目前在国际上,条码技术被广泛用于商业、工业、邮政、生产管理、图书管理、人事管理等各个领域。

### 三、条码技术的应用

70年代美国最早开始应用条码,由于条码技术所具有的诸多优点,因此,自它问世以来,无论是其技术本身还是应用领域都以惊人的速度向前发展和扩展。近几年来,条码技术在许多国家广泛用于各流通领域和生产部门,它所带来的经济效益和社会效益是人们无法估量的。

#### 1. 国外条码技术应用概况

##### (1) 商业

商业是条码技术应用最早的领域,也是当代条码技术应用最广的领域之一。七十年代中期,美国超级市场商曾预言:没有条码的商品将不能摆在超级市场的货架上。今天美国超级市场商出口商标上的印码率已达99%。在日本,现也已普遍采用条码的实时售货系统——POS(Point of Sale)系统。

##### (2) 工业

继条码技术在商业上广泛应用之后,条码技术又开始向工业领域渗透。在美国汽车工业中,协作件的流通规定采用条码技术,它涉及了美国25000家公司。这一措施大大加快了协作件的流通速度,提高了效益。此外,汽车生产装配线也采用了条码技术进行实时控制和工作过程的跟踪。

##### (3) 仓库管理

现代化仓库管理如同超级市场一样,是条码技术的应用领域。在此值得一提的是美国国防部在军用品管理上采用条码技术。美国国防部1982年下达了1189号文件(MIL-STD-1189),要求供应军用品的5万家公司必须在他们的产品上附有规定的条码。现在美国国防领域已成为美国条码技术产品的最大市场,其市场占有率平均为30%。

##### (4) 办公自动化

条码技术在办公自动化方面的应用也发展很快,现在已广泛用于人员管理(如考勤系统等)、图书馆管理、传票处理、档案管理、海关的出入境管理以及机票管理等,在办公自动化领域条码起着逐渐替代键盘的作用。

#### 2. 我国条码技术应用概况

由于条码技术作为一种新技术引入我国时间不长,因此无论是在条码技术的研究、开发方面,还是在条码技术的应用方面,尚处于起步阶段。

### 四、我院的开发成果

在国家“八五”计划中,我院承担了便携式译码器和台式译码器的研制开发。现在,我院已开发出键盘仿真型条码译码器及有关的应用系统,如条码出入管理(考勤)系统和条码档案管理系统等。

#### 1. 键盘仿真型条码译码器

键盘仿真型条码译码器与微机键盘并联,其输出与键盘

输出信号相同,通过键盘接口直接送入计算机,不需要任何硬件扩充及编写专用支持软件,使用起来和键盘一样方便。该译码器可读取ITF码、CODE39码、CODABAR码和JAN/EAN码,可用于IBM-PC、PS/55、PS/2系列的微机,可使用光笔、CCD或激光枪扫描器。

#### 2. 条码出入管理(考勤)系统

条码出入管理系统用于需要进行人员出入情况记录和统计的任何部门和单位。采用条码技术出入管理系统,融合了条码录入技术和计算机管理,使人员的出入管理工作具有手续简单、速度快、实时性强等优点,已成功地用于职工出勤情况管理。

条码考勤系统由一台IBM-PC、一台键盘仿真型译码器、一支光笔(或卡槽)及考勤软件组成。

条码考勤软件由四个子系统组成:

- 实时考勤:随时记录人员的出勤、请假、缺勤等情况的时间和原因。

- 查询:使部门领导随时掌握某部门人员的出勤情况。

- 修改:是某些特定情况下,人工干预时使用的。

- 考勤卡生成:根据不同的需要用激光打印机或针式打印机生成不同格式、特征的胸卡。

条码考勤系统操作简便、使用方便、还在我院试点运行半年以上,深受用户欢迎。

#### 3. 条码在档案管理系统中的应用

条码技术与计算机相结合应用于档案管理,不仅可以提高档案管理的可靠性,促进档案管理现代化、制度化,而且可以大大提高工作效率,同时实现档案借阅、需求情况的实时登录、管理。

档案的收集、整理、鉴定、保管工作都是为了向使用者提供全面、丰富的资料。可以说,档案的使用工作是档案管理的目的。因此,档案馆中的管理主要是指档案使用情况的管理。

在接待室、出纳台和库房各配一台微机,每台微机配有与键盘并连的条码阅读器作为信息输入手段,微机间则用通讯方式进行数据交换。

在这样的系统配置下,各部门完成的主要工作如下:

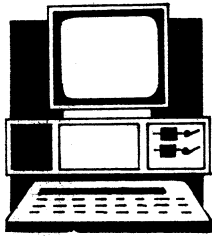
- 接待室:接受读者实时档案目录查询,对索求的在库档案产生出库提单,对索求的在外档案按需要和可能进行预约登记。

- 库房:通过该处微机上所连的译码器读取、入库案卷信息,修改库存数据库。

- 出纳台:办理借、还手续。

此情况下,三处均可实时响应,通过三处中任一处的微机可实时查阅,统计档案出借情况和使用情况。

条码技术作为一种先进的信息采集和输入技术,引入我国的时间虽然不长,但已有了长足的进步。现在专门负责条码登记管理的国家物品编码中心已经成立,最近我国已参加了国际物品编码组织,并计划于“八五”期间使我国6%以上商品使用标有条码的商标;同时,国家也将条码设备作为“八五”重点项目投资开发,为条码技术在我国的发展创造了的条件。随着我国经济建设和科学技术的发展,随着人们对信息采集和输入技术需求的增强,条码技术定将广泛应用于我国国民经济建设的各个领域。



## dBASE III 实现 Foxbase 中 Prompt 的菜单功能

广东佛山绣口总厂 毛维平

### PC 用户

Foxbase 中的...@prompt '...' message '...' 和 menu to (变量)语句给程序设计者们提供了一种简洁的菜单编程方法,特别给用户带来了一种方便新颖的感觉。不少 dBASE 用户也想得到这种享受,然而 dBASE III 没有提供直接实现这种功能的语句。为了解决这个问题,笔者在 PC 机上用 dBASE III 编了一个菜单程序,从操作者界面看与 Foxbase 完全一样。

程序用 INKEY() 函数接收键盘输入的“↑”、“↓”、“↵”键,整个过程只接受这三个键;a1~a5 为显示各模块的字符串,即相当于 Foxbase 中 prompt 语句后面的内容,b1~b5 为各相应模块的说明或下层菜单说明,也就是相当于 Foxbase 中 message 语句后面的内容;P 为 1~5,作为变量的下标,为宏替换提供方便;X 的值决定了所选择的模块;语句 set color to n/w、set color to w/n 设置显示反相和正常显示,当按动“↑”(或↓)时,上(或下)一个模块(字符串)显示为反相,原来模块恢复正常显示,从而实现了光条滚动的效果。若按下“↵”(回车键),表示选中某模块。

```

set exact on
set talk off
set heading off
do while .t.
clear
@ 5,25 say '***** 菜单*****'
a1='1. 输入资料'
a2='2. 修改资料'
a3='3. 查询资料'
a4='4. 打印报表'
a5='5. 退出'
b1='输入出库单. 入库单. 调拨单'
b2='修改各种单据及调整产品库存'
b3='查各种单据. 产品库存. 客户购货情况'
b4='打印各种库存报表. 销售报表'
b5='退出成品仓库管理系统'
@ 7,26 say a1
@ 9,26 say a2
@ 11,26 say a3
@ 13,26 say a4
@ 15,26 say a5
@ 20,15 say '按'+chr(24)+'.'+chr(25)+'键上、下移动光标,按回车键执行选择!'
aao=.t.
do while aao
x=7
p='1'
do while .t.
cm='a'+&p'
bm='b'+&p'

```

```

set color to n/w
@ x,26 say &cm
@ 18,21 say &bm
i=0
do while i=0
i=inkey()
enddo
set color to w/n
@ x,26 say &cm
@ 24,21 say &bm
if i=5
if x=7
x=15
p=5
p=str(p,1,0)
else
x=x-2
p=val(p)-1
p=str(p,1,0)
endif
endif
endif
if i=24
if x>13
exit
else
x=x+2
p=val(p)+1
p=str(p,1,0)
endif
endif
endif
if i=13
aao=.f.
exit
endif
enddo
enddo
clear
do case
case x=7
do 输入模块
case x=9
do 修改模块
case x=11
do 查询模块
case x=13
do 打印模块
case x=15
cancel
endcase
enddo

```



# 信息系统中模糊查找的设计

厦门集美航海学院计算机教研室 戚靖山 朱玉环

在信息系统的研制中经常会碰到通过某一项的已知条件找出相关的其它信息。当这一已知条件是精确值时,即正好是记录中的某一项的值(严格匹配)时,是很容易实现的,但一旦中间插入空格,或者是输入简称(如“上海师范大学”输入“上海师大”)时,则在记录中找不到相应的值。所以,在研制信息系统时,往往要考虑设计具有模糊查找功能的模块。

目前流行的模糊查找的方法,通常是单纯用 AT() 函数来判别输入的已知条件(汉字串)是否是记录中的被检测项值的子串,若是,则显示该项值(汉字串),提问是否是该字符串;若不是,则继续找下一条记录;如果不是被检测项值的子串,则说明不是这条记录,继续找下一条记录。这样查找的局限性很大,使用灵活性也不高。譬如说,严格要求数据库里被测试项的值与输入的已知条件(汉字串)前后及中间不能空格,如要找记录中的单位这一项中“上海师范大学”,输入的已知条件为“上海师”则不行;又譬如,要查找“上海师范大学”,输入“上海师”可以,输入“上海师大”则不行,严格要求输入的已知条件(汉字串)是被检测项中查找的汉字串中的连续汉字组成的子串,这样显然不便于用户的习惯使用,这样设计出来的查找功能只能属于初级阶段的模糊查找。

为此,我们设计了一种功能较强的模糊查找,其特点是:

1. 输入的已知条件(汉字串)和被查找的汉字串允许前后有空格,汉字串中间允许有若干个空格。

如:输入的已知条件(汉字串)为:

“ 上 海 ”

被查找项中的值(汉字串)为:

“ 上 海 师 范 大 学 ”

2. 输入的已知条件(汉字串)可以是要求找到的汉字串中间隔若干个汉字组成的汉字串。

如:要求查到的汉字串为“上海师范大学”

输入的已知条件(汉字串)为:“上海师大”

3. 上述两种情况的混合

如:要求查到的汉字串为:“上海师范大学”输入的汉字串为:“ 上 海 师 大 ”

又如:要求查到的汉字串为:

“上海 师 范 大 学 ”

输入的汉字串为:

“ 上 海 师 大 ”

上述三种情况照样能查找到。

这种模糊查找真正体现出模糊的功能,给用户提供了极大的方便。

该模糊查找的设计思想为:

取出第一条记录的被检测项的值,然后先把被检

测项的值和已知条件(汉字串)都滤去前后的空格,将已知条件的第一个汉字取出,判断是否被包含在检测项中,若不是则退出找下一个记录,若是则截取该汉字后(不包括该汉字)的汉字串做为新的已知条件,截取被检测值该汉字以后(不包括该汉字)的字串作为新的被检测值。同样再滤去各自的前后空格,重复上述过程,直至原来已知条件中的所有汉字都被测试且全部匹配过为止。这时该记录中的测试值就是要查找的汉字串。

步骤:

1. 打开数据库(含被检测项),指针指到起始位置。  
2. 输入已知单位(汉字串)到 CXNR 变量中,并复制一备份在变量 CXNQ 中。

3. 假若输入的已知单位(汉字串)是空串,则返回。

4. 把第一条记录的被测试项的值送入 WX 变量中。

5. 滤去已知条件(汉字串)中的前、后空格,送入变量 ZXNR 中,并把汉字串的长度送入 X 中,当 X 为 0 时则返回。

6. 取出 ZXNR 中最左边的一个汉字,送入变量 E 中。

7. 滤去 WX 中的汉字串的前、后空格并送入 CX 中。

8. 判别 E 中的汉字是 CX 中的子串吗?若是,则把 ZXNR 的长度减 2,并判断是否为 0。若为 0 则表示已知条件(汉字串)中所有的汉字都检测完了,包含在被检测项的值中,那么就显示该检测项的值是否要找的值,是则退出,否则恢复已知条件的初值,重新从下一条记录找起。当 ZXNR 的长度不为 0 时,则截取 CX 中的汉字串从 E 汉字以后到末尾的汉字串送入 WX, ZXNR 中的汉字串也同样去掉第一个汉字以后送入 ZXNR 中,返回到(5)以后继续执行。

当 E 中的汉字不是 CX 中的子串时,则退出后恢复 CXNR 的初值(即已知条件的汉字串),找下一条记录的被检测项的值,送入 WX 变量以后,从第五步开始做起。

假如数据库所有记录中被检测项的值都不满足要求,则显示没有该单位,退回。

附上在“高教论文信息管理系统”中设计的模糊查找 MFCZ. PRG 模块。

```
CLEAR
```

```
SET TALK OFF
```

```
USE YBYYX
```

(YBYYX 为含有被查项的数据库)

```
GO TOP
```

```
CXNR=SPACE(14)
```

(下标为被查项长度)

```
CXNQ=CXNR,
```

```

WX=CXNR
CLEAR
@ 5,30 SAY "输入单位" GET CXNR
READ
CXNQ=CXNR
IF CXNR=""
RETURN
ENDIF
DO WHILE .NOT.EOF()
WX=NAME
DO WHILE .T.
ZXNR=SPACE(LEN(LTRIM(RTRIM(CXNR))))
ZXNR=LTRIM(RTRIM(CXNR))
E=""
X=LEN(ZXNR)
IF X=0
RETURN
ENDIF
E=LEFT(ZXNR,2)
CX=SPACE(LEN(LTRIM(RTRIM(WX))))
CX=LTRIM(RTRIM(WX))
IF AT(E,CX)<>0
X=X-2
IF X=0
CLEAR
@5,30 SAY "工作单位:"
@5,40 SAY NAME
T=""
WAIT "是否(Y/N)?" TO T
IF T="Y"
RETURN
ELSE
CLEAR
EXIT
ENDIF
ENDI
IF LEN(CX)=AT(E,CX)+1
CLEAR
EXIT
ENDI
WX=SUBSTR(CX,AT(E,CX)+2)
CXNR=SUBSTR(ZXNR,3)
LOOP
ELSE
EXIT
ENDIF
ENDDO
SKIP
CXNR=CXNQ
LOOP
ENDDO
CLEAR
@ 5,30 SAY "没有该单位!"
WAIT
RETURN

```

## 高密软盘的单驱动器拷贝

· 北师大二附中高三(2)班 李毅

随着286/386微机的普及,容量为1.2MB的高密驱动器也越来越成为常见的配置了。一般的中档286/386微机上都只配一台高密驱动器,而且这些机器上的内存数量一般也只要一兆左右。这给高密软盘的拷贝带来了极大的不便。我在使用这类机器的时候,也遇到过这样的麻烦。因此,我编制了一个利用硬盘做为缓冲区的高密盘拷贝程序,把它推荐给大家,以便在某种程度上减少麻烦。

程序是用 Turbo C++ 1.0 编写的。但由于没有使用与 C++ 有关的特性,这个程序也可以用 Turbo C 2.0 编译,效果完全一致。由于磁盘的格式化操作要求较高,需要较多的程序才能完成,这里给出的程序就不包括磁盘格式化部分了,也就是说,在使用这个程序之前,请事先将磁盘格式化好(如果磁盘上有内容,即过去曾经使用过,就没有必要再格式化了)。

李毅同学:请函告新地址,以便奉寄稿酬。编者。

程序清单:

```

#include <fcntl.h>
#include <dos.h>
#include <io.h>
#include <process.h>
#include <alloc.h>
#include <conio.h>
#include <ctype.h>
#define buffer_size 7680
#define MAX_SECTOR 2399
int testyn(void);

main()
{
char * buffer;
int secno, nsecno=15;
int handle, done=0;
_fmode=O_BINARY;
buffer=malloc(buffer_size);
puts("High Density Floppy Disk Copy Utility Version 1.01");
puts("Public Domain 1991, by Shadow Information Lab.");
puts("Beijing Normal University, 100875, BJ, CHINA.");
while(!done)
{
puts("\nInsert SOURCE Diskette in drive A.");
puts("Press any key when ready.");
while(!kbhit());
getch();
if ((handle=_creat("C:\\TEMP.BUF",FA_ARCH))==
1)
{

```

```

puts("Sorry! Unable to access buffer on hard disk.");
exit(1);
}
for (secno=0; secno<= MAX_SECTOR; secno+= nsecno)
{
if (absread(0, nsecno, secno, buffer) == -1)
{
puts("Error Access SOURCE Diskette.");
exit(2);
}
_write(handle, buffer, buffer-size);
}
_close(handle);
puts("\nInsert TARGET Diskette in drive A.");
puts("Press any key when ready.");
while (!kbhit());
getch();
if ((handle = _open("c:\\TEMP.BUF", O_RDWR|O_BINARY)) == -1)
{
puts("Sorry! Unable to access buffer on hard disk.");
exit(1);
}
for (secno=0; secno<= MAX_SECTOR; secno+= nsecno)
{
_read(handle, buffer, buffer-size);
if (abswrite(0, nsecno, secno, buffer) == -1)
{
puts("Error Access TARGET Diskette.");
exit(2);
}
}
_close(handle);
printf("\nDo you want another copy(Y/N)?");
done = tstyn();
}
system("DEL C:\\TEMP.BUF");
return 0;
}
int tstyn(void)
{
char ch;
for(;;)
{
while (!kbhit());
ch = getchar();
switch(toupper(ch))
{
case 'Y': return 0;
case 'N': return 1;
}
}
}

```

# 用汇编程序显示 硬盘主引导记录

李日林

硬盘主引导记录是极为重要的系统程序,是使用硬盘必不可少的系统信息区。它包括引导与分区代码、分区表和结束标志“55AA”。但是,由于硬盘主引导记录位于 DOS 分区之外,在 0 磁头、0 柱面的第 1 扇区,所以不能直接用调试程序 DEBUG 访问。为了查看其内容,通常利用 BIOS 提供的 13H 号软中断来读取,再用 DEBUG 中的命令显示,虽然该方法不失为一种可行的手段,但操作、使用不方便,尤其对不熟悉其结构的用户来说,更显得麻烦。

笔者用 8088 汇编语言编制了一段源程序,经汇编、链接后,可以方便地读取硬盘主引导记录,并且直接在屏幕上显示其内容,无须利用 DEBUG 命令,操作、使用都很简单。因注释相当清楚,对程序设计的细节就不再详述。

程序实例 DISPBOOT.ASM

```

output macro a ;显示一个字符的宏指令定义
mov dl, a
mov ah, 2
int 21h
endm
;
stack segment para stack 'stack' ;堆栈段
db 256 dup(?)
stack ends
;
data segment para public 'data' ;数据段
buffer db 512 dup(?) ;定义存放一个扇区的缓冲区
data ends
;
code segment para 'code' ;代码段
start proc far ;汇编程序标准序
assume cs:code, ds:data, es:data, ss:stack
push ds
xor ax, ax
push ax
mov ax, data
mov ds, ax
mov es, ax
;
mov ax, 0201h ;读硬盘主引导记录
mov bx, offset buffer ;存放于定义的缓冲区
mov cx, 0001h
mov dx, 0080h
int 13h
;
mov dh, 0
mov cx, 512

```

```

mov bx,0
lp: push cx ;以下程序段为取一个字节
mov dl,[bx] ;将其转换为 ASCII 码用于显示
push dx
mov cl,4
shr dl,cl
call ascii
pop dx
shl dl,cl
shr dl,cl
call ascii
cmp dh,10h ;每行显示的字节数为10H
jne np
output 0ah
output 0dh
xor dh,dh
jmp np1
np: output 20h
inc dh
np1: inc bx
pop cx

```

```

loop lp ;循环,直至显示完一个扇区
output 0ah
output 0dh
mov ax,4c00h ;返回 DOS
int 21h
start endp
ascii proc near ;子程序,将字符转换为 ASCII 码
cmp dl,9 ;并显示
jb number
add dl,37h
jmp output1
number: add dl,30h
output1: push ax
mov ah,2
int 21h
pop ax
ret
ascii endp
;
code ends
end start

```

## PC DOS 内存管理

海南省商业学校计算机室 陈宏

PC DOS 为了管理内存空间,采用了链式结构,在内存中建立一条由若干内存控制块组成的内存控制块链。该链掌握了整个内存的使用和分配情况,DOS 根据此链就能完成整个内存的管理。内存控制块(MCB)是在 DOS 分配内存时,在所分配的内存块的头十六个字节设置用于描述内存块的特性、段号和容量。MCB 结构如下:

0	1	2	3	4	5	16
标志	分配段号	分配块节数	保			留

1)标志: = 4DH 表示是 MCB 链中的某一块, = 5AH 则是最后的一块。

2)分配段号:说明该内存块已分配的还是空闲的 0000 = 空闲的; XXXX = 程序的 PSP 段址

3)分配块节数:说明该内存块的容量的大小,单位是节,1节 = 16字节。

MCB 链中的第一块在 DOS 引导过程中随同第一个 PSP 建立而建立,相应内存块含有系统开工时的文件缓冲区(FILE)、磁盘缓冲区(BUFFERS)和可安装设备驱动程序(DEVICE),因此该 MCB 容量大小取决于 CONFIG. SYS 的上述三个命令。第二块分给 COMMAND. COM 文件,内容除 PSP 后是 COMMAND. COM 常驻部分的代码和数据。DOS 载入之后,在用户的可用内存低端总存在一块标志为 5AH 的结束 MCB。当 DOS 要载入外部命令或用户应用程序时,EXEC 子功能为程序申请两块内存块,一块放程序的环境块,一块

放程序的 PSP 和程序本身。无论是 EXE 或 COM 文件,载入后 DS、ES 总指向 PSP 段址,如将该指针减 1 就得到含有 PSP 和用户代码和数据内存块的 MCB 段址,文件病毒为在内存中潜伏下来又不能让 DOS 发觉,就直接去修改 MCB,这点与引导扇区病毒很不相同。

各内存块之间的关系是:控制指针  $i+1 = \text{控制指针} + (\text{位移}3)i+1$ ,而 DOS 将第一块 MCB 的指针保留在 DOS 程序段的特定位移内,按此公式便能计算出任何一块 MCB 的段址,通过改写控制块的有关信息达到控制相应内存块的目的,如释放内存(49H)只是将相应的 MCB 位移 1 改成 00,标志为可用块表明是空闲块即可。修改内存 4AH 稍复杂些,要修改相应的 MCB 的大小和重新设立剩余 MCB。Microsoft 公司并没有公布 MCB 结构的细节,直接修改 MCB 要冒一定的风险,当然最好是调用 DOS 为用户提供的关于内存管理的系统调用功能,如申请内存 48H、释放内存 49H 和修改内存 4AH 等,否则 MCB 链一旦断链,整个系统将陷入瘫痪状态并停机(System halt)

运行笔者开发的汇编语言程序的可执行代码便得到当前 DOS 内存的 MCB 情况(如图 1)。图中 NO 项是 MCB 的序号,SEGMENT、SIZE(单位:节)分别是 MCB 的 1-2、3-4 字节的内容,FILE 是掌握该内存块的程序名;PEM 指明内存块的用途,PSP 存放程序 PSP 和程序代码数据;ENV 是环境块;MEM 是既非空闲块也不是 PSP、ENV 的驻留内存块,图中 NO. 6 是 1701 病毒

占用的 MCB。

NO.	SEGMENT	PSP	SIZE	PEM	FILE
1	09F9	0008	817		
2	0D2B	0D2C	211	PSP	\COMMAND.COM 0024(0D2C,0556) 002E(0D2C,0281)
3	0DFD	1197	3	ENV	MCB.EXE
4	0E03	0D2C	10	ENV	\COMMAND.COM
5	0E0E	0E14	4	ENV	C:\PASCAL\DEBUG.COM
6	0E13	0D2C	123	MEM	\COMMAND.COM 0021(0E14,031C)
7	0E8F	0E95	4	ENV	C:\PASCAL\DEBUG.COM
8	0E94	0E95	769	PSP	C:\PASCAL\DEBUG.COM \COMMAND.COM 0001(0E95,08ED) 0003(0E95,08E6) 0022(0E95,0242) 0023(0E95,08E1)
9	1196	1197	36,329	PSP	MCB.EXE C:\PASCAL\DEBUG.COM \COMMAND.COM

Total free memory: 36,329 bytes  
Program terminated normally

图1.

MCB. ASM 主要是利用了内存控制块、程序前缀控制块 PSP 和环境块三者之间的关系而设计的,其中在 PSP+16H 处是父进程的 PSP 段址,子进程可以通过此指针返回父进程,最后会回到命令处理文件 COMMAND.COM,COMMAND.COM 的 PSP+16H,则指向了自己本身,如下图

```

MCB
-D 1196:0 4F
(原 MCB.EXE 的 MCB 和 PSP 内容,程序结束后该内存块被释放,参阅图1)
1196:0000 5A 00 00 E9 8D A2 02 01-53 B4 30 CD 21 5B 3C 02
PSP (1197:0)
1196:0010 CD 20 80 9F 00 9A F0 FE-1D F0 42 02 995 0E 70 02
1196:0020 95 0E 56 05 2C 0D 95 0E-FF FF FF FF FF FF FF FF
1196:0030 FF FF FF FF FF FF FF FF-FF FF FF FF FF 00 0E E8 FF
1196:0040 A7 11 14 00 18 00 97 11-FF FF FF FF 00 00 00 00

-D E94:0 4F DEBUG.COM PSP 段址 0E95=E94+1
0E94:0000 4D 95 0E 01 03 00 00 00-00 00 00 00 00 00 00 00
0E94:0010 CD 20 80 9F 00 9A F0 FE-1D F0 2F 01 2C 0D 3C 01
0E94:0020 2C 0D 56 05 2C 0D 2C 0D-03 04 01 00 02 FF FF FF
0E94:0030 FF FF FF FF FF FF FF FF-FF FF FF FF FF 90 0E C0 2A
0E94:0040 95 0E 14 00 18 00 95 0E-FF FF FF FF 00 00 00 00

-D D2B:0 4F COMMAND.COM 段址 0D2C=D2B+1
0D2B:0000 4D 2C 0D D3 00 59 72 EC-16 1F 80 3E 15 05 FF 72
0D2B:0010 CD 20 80 9F 00 9A F0 FE-1D F0 B2 02 2C 0D 3C 01
0D2B:0020 2C 0D 56 05 2C 0D 2C 0D-03 04 01 00 02 FF FF FF
0D2B:0030 FF FF FF FF FF FF FF FF-FF FF FF FF FF 04 0E 25 08
0D2B:0040 2C 0D 14 00 18 00 2C 0D-FF FF FF FF 00 00 00 00
    
```

在 PSP+2CH 处有环境块的段址,环境块反映的

是 COMMAND.COM 和程序所处周围的情况,由若干以0为结束标志的字符串组成,一般至少有 COMSPEC、PROMPT、PATH 三个字符串组成,在整组环境块结束时以另一个0为结束,在此0字节之后是一组初始化变量,含有指明当前执行程序的驱动器、路径、文件名[扩展名]的 ASCII 串,如下图:(DOS 3.3):

```

D E0E,0 4F
0000 4D 14 0E 04 00 00 00 00-00 00 00 00 00 00 00 M.....
0010 50 41 54 48 3D 00 43 4F-4D 53 50 45 43 3D 43 3A PATH=,COMSPEC=C
0020 5C 43 4F 4D 4D 41 4E 44-2E 43 4F 4D 00 00 01 00,COMMAND.COM...
0030 43 3A 5C 50 4E 53 43 41-4C 5C 44 45 42 55 47 2E, \PASCAL\DEBUG.
0040 43 4F 4D 00 04 D3 EA 8C-C8 03 C2 A3 58 16 A1 16OM. SJ. H. B#X. 1.
    
```

MCB. ASM 由此而获得当前进程名。另外,如在 PSP+2C 处为0则表示环境完全复父进程

MCB. ASM 目的是能及时地了解当前内存的具体情况,便于发现文件病毒入侵的可疑之处,源程序中 ES:DI 一般指向程序的显示区,很容易加上其它功能模块,清单中已有较详细的注释,有兴趣的读者可在此基础上加以改进,实现如删去无用的内存空间,驻留程序、占用大量的内存的汉字库等。

(有关 MCB. ASM 的程序清单,有兴趣的读者可向编辑部函索,收复印及邮挂费2.00元)

## CCBIOS 2.13F 的小补钉

福建省第二电力建设公司 严仲武

CCBIOS 2.13F 是中国计算机系统工程公司于89年9月推出的适用于 IBM-PC 各档微机的一个较成功的汉字操作系统,它以快速、方便和大量特殊的视屏、打印机扩展功能为用户提供方便。按 CCBIOS2.13F 汉字系统《用户手册》介绍,在汉字或词组输入时,用 Alt+Esc 键重选第一个字。但实际使用中发现这时 Esc 键等同于 Alt+Esc 键,使 Esc 失去原作用,给使用带来麻烦,影响用户软件的运行和坚固性。经分析 CCCC.COM 程序,发现处理这个功能的程序段如下:

1. 字输入时:

```

CS:9E9E CMP AH,01 ;是 Esc 或 Alt+Esc 键吗?
9EA1 JNZ 9EC0 ;不是,转9EC0
9EA3 XOR AL,AL ;是,置 AL=0
9EA5 JMP 9D0C ;9D0C 处理
    
```

2. 词组输入时:

```

CS:A757 CMP AH,01 ;是 Esc 或 Alt+Esc 键吗?
A75A JNZ A767 ;不是,转 A767
A75C XOR AL,AL ;是,置 AL=0
A75E JMP A749 ;转 A749 处理
    
```

从这两段程序可见,程序对 Esc 和 Alt+Esc 键作同样处理。但我们希望 Alt+Esc 键用于重选,而对 Esc 键不作特殊处理,保持原功能。为此,把这两段程序改为:

1. 字输入时:

```

CS:9E9E CMP AH,01 ;是 Esc 或 Alt+Esc 键吗?
    
```

9EA1 JNZ 9EC0 ;不是,转9EC0  
9EA3 JMP AA28 ;是,转 AA28处理

## 2. 词组输入时:

CS:A757 CMP AH,01 ;是 Esc 或 Alt+Esc 键吗?  
A75A JNZ A767 ;不是,转 A767  
A75C JMP AA32 ;是,转 AA32处理

为了区分 Esc 和 Alt+Esc 键,需要增补一小段程序,把这一小段程序接在 CCCC.COM 的最后(即分别从 AA28和 AA32开始),该段程序为:

CS:AA28 OR AL,AL ;是 Alt+Esc 键吗?  
AA2A JZ AA2F ;是,转 AA2F  
AA2C JMP 9EC0 ;不是(即是 Esc 键),转9EC0  
AA2F JMP 9D0C; 转9D0C 作重选处理

AA32 OR AL,AL ;是 Alt+Esc 键吗?  
AA34 JZ AA39 ;是,转 AA39  
AA36 JMP A767 ;Esc 键,转 A7667  
AA39 JMP A740 ;转 A749作重选处理

Esc 和 Alt+Esc 键虽然扫描码都是1(即 AH=01),但 Esc 键的 ASCII 码是1B(即 AL=1B),而 Alt+Esc 键的 ASCII 码是0(即 AL=0),AA28和 AA32就是据此区别它们的。

增补后把文件长度 CX 从 A928改为 A93C,然后存盘即可。虽然我们增加了文件长度,但 CCCC.COM 原驻留长度有裕度(为 AB07),因此,增补的程序能安全地驻留内存而不会被覆盖。

# 中文弹出式窗口设计

中国金融学院信息系 姜咏红

弹出式菜单、弹出式窗口,都是使人感觉良好的屏幕设计,但这些技术需要了解较多的显示原理和数据在机器内的存放方式,许多软件技术人员还不善于运用。特别是中文弹出式菜单和窗口,其边界可能要从某些汉字中间穿过,这又涉及到保留半个汉字和将来恢复全字的问题,弄不好就会驴唇不对马嘴。

本文给出弹出式窗口设计的最基本程序,它主要包括窗口区域原内容存储和恢复两个子程序,弹出的窗口是蓝颜色的,可以向其中输入字符和汉字。为了避免程序过长,没有作边界检查。作者相信,只要掌握了本文给出的基本方法,那么通向实用的弹出式窗口及菜单的道路是平坦的。

中文弹出式窗口设计的要点在哪里呢?为了说清这一问题,我们必须介绍一下汉字的显示问题。

采用长城高彩显卡或 CEGA 卡的微机一般都是在字符方式下显示汉字,将汉字作为全角字符处理。假设屏幕可以显示25行,80列,那么屏幕上的内容是内存的某些区域内容通过硬件线路的反映。这些区域称为显示缓冲区。显示缓冲区分为基本显示缓冲区和扩展显示缓冲区两部分。基本显示缓冲区地址规定为 0B8000H,而扩展显示缓冲区的地址随显示卡而不同。CEGA 卡规定的地址是 0B0000H 而长城高彩显卡规定的地址却是 0C0000H。本文的程序是按后者设计的,因为使用后的长城 0520-CH 机的速度较慢,弹出式窗口或菜单的效果更加明显。

不论是半角还是全角字符都有它的显示码。显示码通常由 BIOS 自动生成,它由两个字节组成,低字节在基本显示缓冲区的偶地址,而高字节在扩展显示缓冲区的相同偏移地址处。高字节的第七位即位6是全角字符的标志,此位为1则显示的是全角字符,否则将显示一个半角字符。如果原来显示的是一个汉字,将位6改成0,则将显示这个汉字的左半部分(程序的61行就是完成这一功能的)。字符的属性在两个区各占一字

节,规定它们必须紧接在显示码字节之后。读者不难看出在中西文显示方式下,一个半角字符在基本显示缓冲区占二个字节,而在扩展显示缓冲区也占两个字节,全角字符就应该占8个字节。由于全角字符的显示码也由两个字节组成,故显示汉字要四个字节就够了。对于全角字符,硬件规定后四个字节的內容可以任意。由于占位的需要,这四个字节是不能不要的。为了统一的记录作为全角字符的后四个字节,汉字 BIOS 规定在扩展显示区的显示码位写上 1FH。这样只要查到这一位置是 1FH,就知道包括前面两区内的四个字节的都被某一全角字符使用。利用这一点我们可以查找汉字并对左边界上的汉字进行处理(程序的55-59行)。

窗口技术的最重要问题是对显示缓冲区窗口区域的内容保存和恢复,而左边界上的汉字处理正是弹出式汉字菜单和窗口的难点。左边界上的汉字在生成窗口时变成了只显示左半边,由于底层屏幕的内容是任意的,故这种半角汉字出现的位置完全是随机的。本文程序中使用了行位标记法,即每一行给一个标记字节(程序第2行定义的 S\_LABEL),如果本行有左边界上的汉字,则送 0FFH 给相应的字节,不然这个字节是 0(程序的60行)。有了行位标记,在区域恢复时,就可以先查本行的标记(程序的99行),然后决定是否恢复全角字符(程序的101行),如此就能作到准确无误。

中文窗口技术可以不考虑右边界的汉字,由显示缓冲区的汉字存储方式,读者不难想出这是为什么。

本文给出的源程序是完整的,程序运行时敲“>”产生坐标为(X1,Y1),(X2,Y2)的矩形窗口,敲“<”则使原屏幕恢复,敲回车键结束。41-82行和83-120行的两子程序是通用性子程序,可以用到任何窗口设计程序中,如果用到 CEGA 卡上,只需将扩展显示缓冲区的地址改成 0B0000H。两子程序开始一段都是根据屏幕位置计算内存的地址,了解显示与内存关系的读者一看便知。

1,DATA	SEGMENT	62,C-10:	MOV AX,ES,[SI]
2,S-LABEL	DB 25 DUP(0)	63:	MOV [DI],AX
3,X1	DB 10	64:	INC DI
4,Y1	DB 10	65:	INC DI
5,X2	DB 20	66:	MOV AX,ES,[SI+8000H]
6,Y2	DB 70	67:	MOV [DI],AX
7,BUFFER	DB 4000 DUP(?)	68:	INC DI
8,DATA	ENDS	69:	INC DI
9,STACK	SEGMENT STACK 'STACK'	70:	MOV BYTE PTR ES:[SI],20H
10:	DB 100H DUP(?)	71:	MOV BYTE PTR ES:[SI+1],1FH
11,STACK	ENDS	72:	MOV BYTE PTR ES:[SI+8000H],00H
12,CODE	SEGMENT	73:	INC DL
13:	ASSUME CS,CODE,DS,DATA,ES,DATA	74:	CMP DL,Y2
14,START	PROC FAR	75:	JBE C-1
15:	PUSH DS	76:	MOV DL,Y1
16:	MOV AX,0	77:	INC BL
17:	PUSH AX	78:	CMP BL,X2
18:	MOV AX,DATA	79:	JBE C-1
19:	MOV DS,AX	80:	POP ES
20:	MOV ES,AX	81:	RET
21,C00:	MOV AH,1	82,SAVE	ENDP
22:	INT 21H	83,RESTORE	PROC NEAR
23:	CMP AL,'>'	84:	MOV SI,OFFSET BUFFER
24:	JNZ C01	85:	PUSH ES
25:	CALL SAVE	86:	MOV AX,0B800H
26:	MOV AH,2	87:	MOV ES,AX
27:	MOV DH,X1	88:	MOV BL,X1
28:	ADD DH,1	89:	MOV DL,Y1
29:	MOV DL,Y1	90,C-01	MOV AL,160
30:	INC DL	91:	MUL BL
31:	INT 10H	92:	XOR BH,BH
32:	JMP C00	93:	XOR DH,DH
33,C01:	CMP AL,'<'	94:	ADD AX,DX
34:	JNZ C02	95:	ADD AX,DX
35:	CALL RESTORE	96:	MOV DI,AX
36:	JMP C00	97:	CMP DL,Y1
37,C02:	CMP AL,0DH	98:	JNZ C-010
38:	JNZ C00	99:	CMP S-LABEL [BX],OFFH
39:	RET	100:	JNE C-010
40,START	ENDP	101:	OR BYTE PTR ES:[DI+8000H-2],40H
41,SAVE	PROC NEAR	102:	MOV S-LABEL [BX],0
42:	MOV DI,OFFSET BUFFER	103,C-010	MOV AX,[SI]
43:	PUSH ES	104:	MOV ES,[DI],AX
44:	MOV AX,0B800H	105:	INC SI
45:	MOV ES,AX	106:	INC SI
46:	MOV BL,X1	107:	MOV AX,[SI]
47:	MOV DL,Y1	108:	MOV ES:[DI+8000H],AX
48,C-1	MOV AL,160	109:	INC SI
49:	MUL BL	110:	INC SI
50:	XOR BH,BH	111:	INC DL
51:	XOR DH,DH	112:	CMP DL,Y2
52:	ADD AX,DX	113:	JBE C-01
53:	ADD AX,DX	114:	MOV DL,Y1
54:	MOV SI,AX	115:	INC BL
55:	MOV AL,ES:[SI+8000H]	116:	CMP BL,X2
56:	CMP AL,1FH	117:	JBE C-01
57:	JNZ C-10	118:	POP ES
58:	CMP DL,Y1	119:	RET
59:	JNZ C-10	120:	RESTORE ENDP
60:	MOV S-LABEL [BX],OFFH	121,CODE	ENDS
61:	AND BYTE PTR ES:[SI+8000H-2],0BFH	122:	END START

# 一种实用新颖的排序方法

南京大学 朱国江

几乎所有的 BASIC 语言教程,在介绍名目繁多的排序方法时,其设计思想均采用设置数组、反复比较、不断交换的雷同模式。虽然它们各具特色,但都存在占用内存大、速度慢,无法区分并列名次等问题。

本文介绍的排序程序跳出了传统的框框,因而可以克服不同排序方法的缺点。以 10 个学生单科成绩排序为例,程序及运行结果如后。几点说明:

一、N 为考生人数, M 为满分。N、M 可视实际情况灵活选用,但应和 DATA 语句中存储记录一致。

二、X 记录不同考生分数,兼作数组 A(X), A\$(X) 的下标。其中 A(X)=A(X)+1, 统计考分为 X 的学生人数,是数值相加; A\$(X)=A\$(X)+X\$, 是存储相同成绩 X、不同学号的组合,是字符串连接。例如本例中考 99 分的有二个人,学号分别为 8807, 8809, 则经过 10-60 句处理后, A(X)=A(99)=2, A\$(99)=88078809。

三、设置不同条件语句,分别判断名次。70-140 句的循环,打印排序后的学号、名次和成绩。根据条件语句满足与否,分别执行三个程序段。

· 循环变量 I 控制分数,从满分 120 开始,只要 A(I)=0 成立,就不要。这有两层意思:若存储单元 A(I) 中没有相应成绩的人数,则循环上去,否则执行 J 循环;若没有 80 句条件语句安排,则 J 循环(100-130 句)无法执行。

· 若相同成绩 X 的人数 A(I) 等于或大于 2 时,转向处理并列名次(160-210 句)。

· 当 A(I) 仅为 1 时,执行 100-130 句,其中 C=C+1 作为名次计数器用;而 120 句采用 MID\$ 函数,是将 A\$(I) 中的字符分成 4 个一组的学号依次取出,由于此时 A(I)=1,因而 J=1,所以 MID\$(A\$(I), 4\*(J-1)+1, 4) 中,是从 A\$(I) 的第一个字符开始,取四个字符出来,这正好是成绩为 X 的、仅为一个人的学号。采用 MID\$ 函数,省去交换学号的方法,能保证对应成绩的学号正确打印。

四、安排加、减计数器,巧妙处理并列名次。这由 160-210 句程序段解决。由于此时 A(I)≠1,即相同成绩的人数不止 1 人,因而需要处理同名次问题,方法是在打印一个相同成绩的学号和成绩后,名次必须减 1;而在完成所有相同成绩的同名次人员情况后,计数器应加 1,并返回 140 句。

五、不受人员限制,占用内存少。本程序最大特点是占用内存少,且不受人员限制,这是因为 DIM 语句中仅开辟两个数组 A(M), A\$(M) (下标值仅由 M 决定和人数无关。这就是说,当 M 一定时(如 M=120),只占用内存 240 个,无论是 300 人,还是 3000 人,内存并未增加;而用冒泡法排序时,300 人内存为 720, 3000 人内存为 7200。

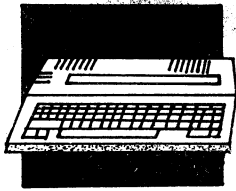
六、占用时机少,执行速度快。传统的排序程序,采用反复比较、不断交换的方法。比较、判断、交换是需要时间的,当信息量足够大时,这种矛盾尤为突出。本程序排序没有采用比较、交换的方法,因而速度特别快。如对一个 3000 人的排序,用冒泡法需时 10 多个小时,而本法不到 4 分钟。

七、修改循环变量初终值,速度还可以提高。例如,单科成绩最高分为 100,则 M 取 100;最低成绩为 40 分,则改 70 句为:FOR I=100 TO 40 STEP-1,这样,由于循环次数大大减少,执行速度可进一步提高。

八、兼有成绩分档、录取划线和检索功能。这主要修改 70 句的循环语句来实现。例如,FOR I=120 TO 100 STEP-1,可以将 100-120 分的成绩分档;FOR I=80 TO 80,可以检索 80 分的人员情况;FOR I=120 TO 70,决定 70 分以上的考生数,作为录取划线的标准等等。而所有这些工作,必须在执行本程序一次后,直接 RUN 70 完成。

```
10 INPUT "n,m=";N,M
20 DIM A$(M),A(M)
30 FOR I=1 TO N:READ X$,X
50 A(X)=A(X)+1:A$(X)=A$(X)+X$
60 NEXT I
70 FOR I=M TO 0 STEP -1
80 IF A(I)=0 THEN 140
90 IF A(I)>=2 THEN 160
100 FOR J=1 TO A(I):C=C+1
120 PRINT MID$(A$(I),4*(J-1)+1,4),C,I
130 NEXT J
140 NEXT I
150 END
160 FOR J=1 TO A(I):C=C+1
180 PRINT MID$(A$(I),4*(J-1)+1,4),C,I
190 C=C-1:NEXT J
210 C=C+1:GOTO 140
230 DATA 8801,96,8802,98,8803,97,8804,98
240 DATA 8805,120,8806,98,8807,99
250 DATA 8808,98,8809,99,8810,95
RUN
n,m=? 10,120
8805      1      120
8807      2      99
8809      2      99
8802      3      98
8804      3      98
8806      3      98
8808      3      98
8803      4      97
8801      5      96
8810      6      95
```





# 多层次全能阴影菜单

学习机之友

四川攀枝花钢铁公司冶金工业学校 黄文贵

我们自编各种各样的程序时,通常都需要菜单,以选择所需的功能。根据程序的内容不同,菜单的项目、每项项目的字符个数也会随之不同;在同一程序中,有时会使用两个以上不同的菜单。阴影菜单是软件中常采用的一种方法,因调试工作较费时间,人们只好不用阴影菜单,而用1~9数字键来选择菜单项目。笔者编制了一菜单程序,基本上不做任何调试,就能适用于各种各样的模块化程序。能实现菜单的“双列显示”、“单列显示”、“多层次菜单”三大功能。现分别介绍如下:

## 一、双列显示

读者只需在11语句中输入“菜单标题”,根据自己所编程序的功能在15~21语句中输入菜单的各项名称。最后修改23语句,就能连接好自己所编的程序。去掉25语句,就能进入正常使用。

## 二、单列显示

对于7个以下的菜单项目,如果需要单列显示时,只需把10语句中的“双列显示”字样改为“单列显示”字样就行了。其它步骤与“一、双列显示”相同,请见500~515语句。

## 三、共同问题

(1)屏幕上显示的实际项目,由13语句中的Q值决定,故用户随时都可在2~14个项目之间任意选择,最多为14个项目。如果使用不完,也不必删去,保留下来,为以后增添用,请见503语句以及508~511语句中的Q\$(8)~Q\$(14)。

(2)为使菜单能显示在屏幕的中间位置,读者需修改12语句中的H值。这也是本程序中唯一需调试的一步,除此之外,读者不需要做任何调试,故此程序的通用性广、灵活性大。现提供一个“双列显示”的参考数字,每项目的字符数分别为4、6、8、10、12(每个汉字为2个字符,即为2~6个汉字)时,可分别取H=8、7、6、5、4。“单列显示”的参考数字,H=17-字符个数/2(即为H=17-汉字个数)。

(3)菜单项目较少时,可在部分项目中输入空格,使项目之间有间隔,以达到屏幕显示上的美观。

## 四、多层次菜单

在同一个程序中,当需要两个以上不同的菜单时,可反复使用10~24语句,请见500~514语句。用23语句和513语句就能分别连接好第一部分程序和第二部分程序。25语句和515语句用作演示,正常使用时,去掉这两语句。读者在演示时,分别选择菜单一和菜单二中的“结束”项目,就能了解到多层菜单的特点。

## 五、共用程序

1000~1020语句是“多层次全能阴影菜单”主要部分,即两个以上不同菜单的共用程序。这一部分已调试好,不必修改。只需完成以上步骤,就能正常使用。

## 六、使用方法

运行本程序后,在菜单状态下,用户只需按动中华学习机键盘右侧的上、下、左、右四个方向键,移动阴影选择自己所需的功能,按回车键就能进入到相应的程序中。

```

5 PRINT CHR$(4);"PR#3";PRINT;DIM Q$(15)
10 Q$(0)="双列显示";REM Q$(0)="单列显示"
11 Q$(15)="菜单一";REM 菜单标题
12 H=6;REM 每个项目4-12个字符分别取H=8-4
13 Q=14;REM 实际菜单数目,最多为14个。
14 REM 14-20语句菜单项目,只修改内容不删除。
15 Q$(1)="查看";Q$(2)="修改"
16 Q$(3)="添加";Q$(4)="插入"
17 Q$(5)="删除";Q$(6)="建立"
18 Q$(7)="打印";Q$(8)="目录"
19 Q$(9)="存盘";Q$(10)="调盘"
20 Q$(11)="加锁";Q$(12)="解锁"
21 Q$(13)="说明";Q$(14)="结束"
22 GOSUB 1000
23 ON Q1 GOSUB 25,25,25,25,25,25,25,25,25,25,25,25,25,25,25,25,500
24 GOTO 22
25 HOME;VTAB 4;HTAB 3;PRINT"第一部分程序:";Q$(Q1);GET K$;PRINT;RETURN
500 Q$(0)="单列显示"
501 Q$(15)="菜单二"
502 H=14
503 Q=7
505 Q$(1)="继续画";Q$(2)="重新画"
506 Q$(3)="图形成";Q$(4)="加汉字"
507 Q$(5)="存磁盘";Q$(6)="结束"
508 Q$(7)="返回";Q$(8)="目录"
509 Q$(9)="存盘";Q$(10)="调盘"
510 Q$(11)="加锁";Q$(12)="解锁"
511 Q$(13)="说明";Q$(14)="结束"
512 GOSUB 1000
513 ON Q1 GOSUB 515,515,515,515,515,515,515,10
514 GOTO 512
515 HOME;VTAB 4;HTAB 3;PRINT"第二部分程序:";Q$(Q1);GET K$;PRINT;RETURN
1000 IF Q$(0)="单列显示" THEN Q2=Q;GOTO 1003
1001 Q2=INT(Q/2)
1002 IF Q/2<Q2 THEN Q=Q+1;Q2=Q2+1;Q$(Q)="";REM 空8格
1003 HGR2;VTAB 1;PRINT TAB(17-LEN(Q$(15))/3);Q$(15)
1004 FOR I=1 TO Q
1005 V=I-(I>Q2)*Q2;VTAB V+2
1006 HTAB H+(I>Q2)*14;PRINT Q$(I)
1007 NEXT I;VTAB 3;HTAB H

```

```

1008 PRINT CHR$(15);Q$(1);CHR$(14)
1009 Q1=1;I=1
1010 K=PEEK(-16384)-128;IF K<0 THEN 1010
1011 V=I-(I>Q2)*Q2;VTAB V+2
1012 HTAB H+(I>Q2)*14;PRINT Q$(I)
1013 Q1=Q1+(K=10)-(K=11)+(K=21 AND Q1<Q)
      *Q2-(K=8 AND Q1>1)*Q2+(K=21 AND Q1>
      Q2)-(K=8 AND Q1<Q2+1)
1014 Q1=Q1+(Q1<1)*Q-(Q1>Q)*Q
1015 V=Q1-(Q1>Q2)*Q2;VTAB V+2
1016 HTAB H+(Q1>Q2)*14
1017 PRINT CHR$(15);Q$(Q1);CHR$(14)
1018 IF K=10 OR K=11 OR K=8 OR K=21 THEN I=Q1;
      GAME
1019 POKE -16368,0;IF K<>13 THEN 1010
1020 RETURN

```

## 虚 线

郑州煤田地质学院 刘 裕

虚线是机械制图中的常用线型。根据国家标准规定,虚线在起始与终端应是线段,不能是间隔;线段与间隔有一定比例关系且应各自大致相等。笔者认为,作为计算机绘图,线段与间隔应各自严格相等,而不能满足于手工绘图的大致相等。现行一些绘虚线程序及绘图机命令所绘虚线总不尽如意,有的会出现终端不是线段的情况,绘图机中的画虚线命令也有这种情况,如日产 MP-1000 型小平板绘图机就是如此;有的虽总能使终端是线段,但最后一线段不是稍长,就是稍短一些,很难保持各自相等;有的虽能按比例变化线段与间隔的长度,但最后一段总是不变,也很难保持各自相等。这种误差虽然很微小,但都不能算是完美的虚线,现存一些弊端给用户带来诸多不便。经过对虚线绘制过程的分析,笔者找到了一个关系式:

$$P_1^* = L \frac{P_1}{(N+1)P_1 + NP_2}$$

$$P_2^* = L \frac{P_2}{(N+1)P_1 + NP_2}$$

依此式编程绘虚线,可以圆满解决上述弊端,使绘虚线工作达到完美境界。

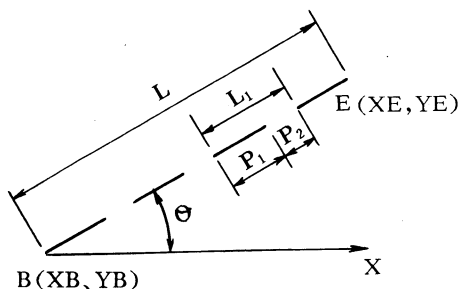


图 1

### 1. 程序设计

设由始点 B 到终点 E 的虚线长为 L, 与 X 轴正方向的倾角为  $\theta$ , 线段部分  $P_1$  可称为实部, 间隔  $P_2$  为虚部, 那么  $L_1 = P_1 + P_2$  可称为一个虚节, 如图 1 所示。

则  $N = \text{INT}(L/L_1)$ , N 为 L 中所含虚节  $L_1$  的整数倍。然后由(1)式求得的  $P_1^*$  与  $P_2^*$  就是绘图时所用的实部与虚部。再配合抬笔、落笔命令驱动画笔, 即可绘出满意的虚线。其 BASIC 程序如下:

```

1400 INPUT "XB,YB,XE,YE=?";XB,YB,XE,YE
1403 P1=12;P2=8
1404 HGR;HCOLOR=3
1405 LX=XE-XB;LY=YE-YB
1410 L=INT(SQR(LX*LX+LY*LY)+0.5)
1415 SI=LY/L;CO=LX/L
1420 L1=P1+P2
1425 N=INT(L/L1)
1430 IF L<(2*P1+P2) THEN PRINT "NO";GOTO 1495
1435 P1=L*P1/((N+1)*P1+N*P2)
1440 P2=L*P2/((N+1)*P1+N*P2)
1445 X1=P1*CO;Y1=P1*SI
1450 X2=P2*CO;Y2=P2*SI
1455 HPOINT XB,YB
1460 FOR I=1 TO N
1465 XB=INT(XB+X1+0.5);YB=INT(YB+Y1+0.5)
1470 HPOINT TO XB,YB
1475 XB=INT(XB+X2+0.5);YB=INT(YB+Y2+0.5)
1480 HPOINT XB,YB
1485 NEXT I
1490 HPOINT TO XE,YE
1495 END

```

### 2. 程序说明:

(1) 1405~1415 行 由始点 B(XB, YB) 与终点 E(XE, YE) 的坐标, 求出虚线长度 L 以及倾角  $\theta$  的正弦和余弦值, 使之能绘任意倾角的虚线;

(2) 1430 行 当 L 长内不够二个实部和一个虚部时, 显示 "NO" 并结束, 否则执行绘虚线工作;

(3) 1435~1440 行 计算出  $P_1^*$  与  $P_2^*$ ;

(4) 1445~1485 行 绘制虚线;

(5) 1490 行 绘制最后一实部;

(6) 该程序可直接在 APPLE II 机上运行显示, 在作为子程序时须删改程序的首部与尾部;

(7) 实用时须适当选取参数  $P_1$  与  $P_2$  的值;

(8) 该程序容易移植到其它机型上, 也容易改写成适用于绘图机的程序。

### 3. 程序变型

机械制图中还有另一种虚线如图 2 所示, 当虚线在实线的延长线上时, 其起始与终端应是间隔, 不能是线段。以往的程序虽也能绘出这种虚线, 但要符合标准则是更加困难。本文所给程序稍加修改就能绘制这种两头虚的虚线, 将相应语句行做如下改动即可:

```

1403 P1=8;P2=12
1470 HPOINT TO XB,YB
1480 HPOINT TO XB,YB

```

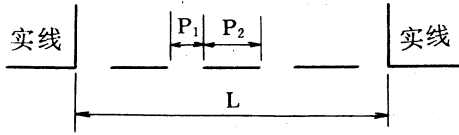


图 2

0314-	60			RTS	
0315-	20	F8	E6	JSR	\$E6F8
0318-	A9	A0		LDA	#\$A0
031A-	E0	28		CPX	#\$28
031C-	F0	06		BEQ	\$0324
031E-	E0	23		CPX	#\$23
0320-	DB	0D		BNE	\$032D
0322-	A9	8C		LDA	#\$8C
0324-	8D	B5	AE	STA	\$AEB5
0327-	8E	EF	B3	STX	\$B3EF
032A-	8E	FE	BE	STX	\$BEFE
032D-	60			RTS	

## 技巧两则

安徽铜陵一中 宋正荣

### 一、跟踪解密初探

有一些软件的程序加密很严格,不论用 CTRL-C 还是 CTRL-RESET 都不能使软件终止运行。CTRL-RESET 还可以导致冷启动、死机等现象。是不是没有办法呢?跟踪启动是一个很好方法,具体步骤如下:

- 1 先将驱动卡中的 BOOT0 移至内存中的 RAM 区。例 \* 8600<C600,C6FFM
- 2 将磁盘放入驱动器中,并在监控状态输入 \* 86F8: A9 00 8D 4A 08 4C 01 08
- 3 待光标出现输入 B747:4C BF 9D

现在已进入 DOS 状态,并且可以列印软件目录,调用程序等工作。

### 二、35 轨与 40 轨磁盘间的通信

《电子与电脑》杂志刊登过如何格式化 40 轨的磁盘。这样虽然容量变大了,但由于按常规方法将 35 轨盘上的文件转移到 40 轨的盘上较繁。使得格式化 40 轨盘好处并不太多。为此我编了个程序。使得这一转化过程十分简单。首先在监控状态下键入程序。用 BSAVE CHANGE, A \$ 300, L \$ 2E。然后用 BRUN CHANGE, A \$ ××××即可。程序是浮动的。可以在任一 RAM 区内运行。

命令 &35 是对 35 轨磁盘操作。&40 是对 40 轨磁盘操作。

0300-	AD	73	AA	LDA	\$AA73
0303-	8D	F7	03	STA	\$03F7
0306-	A9	4C		LDA	#\$4C
0308-	8D	F5	03	STA	\$03F5
030B-	AD	72	AA	LDA	\$AA72
030E-	18			CLC	
030F-	69	15		ADC	#\$15
0311-	8D	F6	03	STA	\$03F6

## 改进 MUSIC 的发音

九江国棉二厂子弟学校初三班 熊波

在 CEC-1 型计算机上有一条音乐语句“MUSIC”,后跟两个参数 X,Y。但要给计算机编一段音乐程序,还需查表对照,得花很长一段时间,使用非常不便。我针对上述问题把 X,Y 两参数简化,X 参数的取值范围是 1-7,与音乐简谱相同。当遇上高音或低音,则在其后跟上一个 H 或 L。这样就形成了音阶参数。音长取值范围是 1-5,1 表示 1 拍,2 表示 2 拍,3 表示 1/4 拍,4 表示 4 拍,5 表示 1/2 拍,如此编起程序来非常方便直观。

例如

5 1 3 | 2 4 3 | 2 2 2 1 2 | 5 — |  
6 7 1 | 1 2 3 | 5 4 3 1 | 2 — |

程序清单如下:

```

10 DIM D(15),E(5);FOR I=1 TO 15;READ D(I);NEXT I;DATA 192,171,152,140,128,114,102,95,84,75,68,62,255,228,205
14 FOR I=1 TO 5;READ E(I);NEXT I;DATA 110,160,30,255,70
15 READ G$,Q
17 IF G$="-1" THEN END
20 A$=RIGHT$(G$,1);B=VAL(G$);A=(A$="H")*(B+7)+(A$="L")*(B+8)+(A$<>"H"AND A$<>"L")*B
30 MUSIC D(A),E(Q);GOTO 15
40 DATA 5L,5,1,5,3,1,2,5,4,5,3,1,2,5,2,5,2,5,1,3,2,3,5L,2
50 DATA 6L,5,7L,5,1,1,1,5,2,5,3,1,5,5,4,5,3,5,1,5,2,2,-1,-1
    
```

# 中华学习机 RAM 监控及其应用

沈阳于洪区吴家荒小学 朴一兵

CEC-I 型中华学习机的 ROM 中固化了与 AP-  
PLE I 机兼容的监控程序,并有所增强,增加了跟踪命  
令 T 和单步执行命令 S,功能较强。但在一些场合就没  
那么方便了,如不能用监控命令解读 CEC-I 的汉字  
管理程序和 16K RAM 语言卡中的内容,其主要原因是  
ROM 中固化的监控程序和上述程序在地址上的冲突。  
为此,本人将 ROM 中的监控移到主存 RAM 中,并进  
行了相应的修改,使其不但能显示内存单元的 16 进制  
内容,还能将其对应的字符显示出来,使用起来非常方  
便。

程序一 将 ROM 中的监控程序移到 \$1800 开始  
的主存 RAM 中,并进行重新设定工作。使用时,将程  
序一输入完后,300G 即可进入 RAM 监控,为了和  
ROM 监控区别,提示符改成 %。

程序二 对主存 RAM 监控进行修改,使其将内  
存中对应的字符显示出来。程序输入完后,1700G 即可  
完成对监控程序的修改工作,然后 BSAVE RAM MON,  
A \$1700,L \$900 存盘,以后使用时只要 BRUN RAM  
MON 即可。

下面简单介绍一下 RAM 监控的应用。

## 1. 剖视 CEC-I 汉字管理程序

在 RAM 监控下,C3ABG 切换到辅存 ROM 中,然  
后就可以用监控命令反汇编汉字管理程序了。注意,因  
内存上的冲突,反汇编的程序不能打印。若要打印,可  
用拷贝贝屏幕的方法打印。

## 2. 解拆文件

LOCKSMITH V6.0 中有一很有用程序,即 BOOT  
TRACE,这个程序对调试及解密 6502 机器语言是一强  
大的工具,但每次使用都要启动 LOCKSMITH V6.0,很  
不方便,因 BOOT TRACE 是存在 16K RAM 卡上。卡上  
无监控程序,不能直接输入 C080 后读出来,把它解拆  
成单个文件还是比较麻烦的,但有了这一 RAM 监控  
后,问题就简单多了,方法如下:

BOOT 起 LOCKSMITH V6.0。

按 A 进入 BOOT TRACE。

输入 RAM 卡位置为 0,

CTRL-Y 进入监控程序。

6 CTRL-P 启动标准 DOS 3.3,

BRUN RAM MON。

C080 / 3000 < D000. FFFFM /, 输入程序三,  
BSAVE BOOT TRACE, A \$2FD0, L \$3040 即可,以后  
使用时只要 BRUN BOOT TRACE。

## 3. 剖视 64K DOS

许多用户喜欢用 64K DOS 磁盘操作系统,如

DAVIE DOS, SUPER DOS V1.6 等,它把用户内存扩充  
了 10K 多,这对于内存小的八位机来说是一大突破。  
但现在提供的操作系统多少都有改进的余地,不少用  
户也在进行这方面的工作,但 64K DOS 是存在 16K  
RAM 卡上的,不能直接用系统提供的监控命令修改。  
利用本文介绍的 RAM 监控就非常方便了。方法很简  
单,BOOT 64K DOS,再 BRUN RAM MON,再 36:F0 1D  
1B 1D /, 然后 C0 80 / 就可以随心所欲地用监控命  
令进行修改工作了。

程序一

```
0300- A9 00 85 3C 85 42 85 06
0308- A8 A9 F8 85 3D A9 FF 85
0310- 3E 85 3F 85 08 A9 18 85
0318- 43 85 07 20 2C FE A9 1F
0320- 85 09 A0 00 B1 06 C9 B9
0328- D0 31 A0 02 B1 06 C9 F8
0330- 90 04 E9 E0 91 06 A5 06
0338- C5 08 A5 07 E5 09 E6 06
0340- D0 02 E6 07 90 DC A9 1D
0348- 8D 54 AA 8D 56 AA A9 1E
0350- 8D BF 1F A9 A5 8D 6A 1F
0358- 4C 69 1F C9 BD F0 CB C9
0360- D9 F0 C7 C9 20 F0 C3 C9
0368- 4C D0 CB 4C 2A 03
```

程序二

```
1700- A9 4C 8D C3 1D A9 1A 8D
1708- C4 1D A9 17 8D C5 1D A9
1710- 1D 8D 54 AA 8D 56 AA 4C
1718- 69 1F 90 03 4C 28 17 A5
1720- 3C 29 07 F0 03 4C B6 1D
1728- 20 56 17 A9 1F 85 24 4C
1730- 3B 17 A5 3C 29 07 D0 03
1738- 4C B3 1D B1 3C 20 46 17
1740- 20 BA 1C 90 ED 60 C9 A0
1748- 90 02 25 32 84 35 48 20
1750- F0 1B 68 A4 35 60 A5 3C
1758- 29 07 D0 0D 18 A5 3C E9
1760- 07 85 3C 90 01 60 C6 3D
1768- 60 A5 3C 29 07 85 08 A5
1770- 3C E5 08 85 3C 60 4C 43
1778- 17
```

程序三

```
2FD0- 2C 81 C0 2C 81 C0 A9 00
2FD8- 85 3C 85 3E 85 42 A9 30
2FE0- 85 3D A9 60 85 3F A9 D0
2FE8- 85 43 A0 00 20 2C FE AD
2FF0- 80 C0 4C 00 E4
```

# CEC— I 键控光标作图程序

安徽蚌埠平板玻璃厂 陈国良

本程序具有方便的键控作图功能:八个方向的留轨迹及不留轨迹的光标运动作图;任意角度的直线、各种矩形、园及椭圆、任意园心角的园弧及各种正多边形;作图过程中随时可更换选择颜色,且速度较快。

操作使用介绍:

(1) U I O    ↖ ↗  
      J K        ← →  
      N M ,    ↙ ↘

分别控制光标移动的八个方向。

(2)按 0~7 八个数字键可在作图过程中随时改选色笔颜色。(由于高分辨彩色设置的限制,实际使用时可能会有部分颜色变异)。

(3)按 Q 键后:落笔,按 W 键后:抬笔,即可通过控制抬落笔选择来实现留轨迹及不留轨迹作图。

(4)画直线 在光标移到直线的起点时按 W 键抬笔,然后移动光标至直线终点后再按一下 F<sub>1</sub> 键,即可画出一条直线。

(5)矩形 光标移到欲画矩形的任一角时按 W 键抬笔,按一下 F<sub>2</sub> 键,然后移动光标至矩形的对角点再按一下 F<sub>2</sub> 键即可画出一个矩形。

(6)园或椭圆 光标移到园心处后按 W 键抬笔,移动光标后按一次 F<sub>3</sub> 键定义 X 轴方向的半径,继续移动光标后再按一次 F<sub>3</sub> 键则定义 Y 轴方向半径,此时即可画出椭圆。如两次定义的半径相等即画出一个园。

(7)园弧 在作园过程中,按除 F<sub>3</sub> 键以外的任意键,光标只移动而不留轨迹,再按 F<sub>3</sub> 键则可控制留轨迹画弧。

(8)园(或椭圆)内接正多边形 在定义园心坐标后按 F<sub>4</sub> 键,此时闪烁光标消失,可按 3~9 的数字键定义欲画多边形的边数(过多边数的多边形已近似园形,故本程序未予考虑),接下来可按照画园(或椭圆)的操作步骤进行即可,不过此时定义的是所画多边形外接园(或椭圆)的半径。

(9)在整个作图过程中,可按 ESC 键退出本次作图回到原始状态以便从头开始作图。

```
10 FOR I=768 TO 778,READ J,POKE I,J,NEXT,SCALE=
1;ROT=0;POKE 232,0;POKE 233,3
20 DATA 1,0,4,0,36,214,111,41,159,54,0
30 HGR2; K=-16384;L=-16368
40 X=139;Y=95;XX=X;YY=Y;D=1;C1=COS(.1);S1=
SIN(.1);N=64;C=3
50 HCOLOR=C; HPLOT 0,0 TO 279,0 TO 279,191 TO 0,
191 TO 0,0
60 P=PEEK (K);POKE L, 0;A=X;B=Y
70 X=X+(P=203 OR P=207 OR P=172)-(P=202
```

```
OR P=213 OR P=206);Y=Y+(P=205 OR P=206
OR P=172)-(P=201 OR P=213 OR P=207)
80 IF X < > A OR Y < > B THEN 170
90 IF P > =176 AND P < =183 THEN C = P-176;GO-
TO 60
100 IF P = 209 THEN D=1;GOTO 60
110 IF P=215 THEN D=0;XX=X; YY=Y;H=0; GOTO
60
120 IF P=129 THEN 220
130 IF P=151 THEN 250
140 IF P=140 THEN 200
150 IF P=148 THEN 230
160 IF P=155 THEN 30
170 X=X+(X<1)-(X>278);Y=Y+(Y<1)-(Y>
190)
180 IF D=1 THEN HCOLOR=C;HPLLOT X,Y
190 XDRAW 1 AT X,Y;FOR I=0 TO10;NEXT ;XDRAW 1
AT X,Y; GOTO 60
200 R(H)=X;H=H+1;R(H)=Y;IF H<2 THEN H=H+
1;GOTO 60
210 H=0;HCOLOR=C;HPLLOT R(0),R(1) TO R(0),R(3)
TO R(2),R(3) TO R(2),R(1) TO R(0) ,R(1);GOTO
60
220 HCOLOR=C;HPLLOT XX,YY TO X,Y;XX=X;YY=Y;
GOTO 60
230 GET A $;P=VAL (A $);IF P<3 THEN 230
240 C1= COS (6. 2832/P);S1= SIN (6. 2832/P);N=P;
GOTO 60
250 R(H)=SQR ((X-XX) * (X-XX)+(Y-YY) * (Y-
YY));IF H=0 THEN H=1;GOTO 60
260 H=0;F=0;X1=1;Y1=0;HCOLOR=C
270 FOR I=0 TO N;T=X1 * C1-Y1 * S1;Y1=X1 * S1+
Y1 * C1;X1=T
280 X=XX-.96 * Y1 * R(0);Y=YY-X1 * R(1)
290 IF X < 1 OR X > 278 OR Y < 1 OR Y > 190 THEN F
=0;GOTO 350
300 P=PEEK (K); POKE L,0
310 IF P< > 151 AND P < > 23 THEN F=0;GOTO
340
320 IF F=0 THEN HPLLOT X,Y,F=1
330 HPLLOT TO X,Y; GOTO 350
340 XDRAW 1 AT X,Y; FOR J=0 TO 50;NEXT ;XDRAW
1 AT X,Y
350 NEXT
360 HCOLOR=0;HPLLOT XX,YY;HCOLOR=C;X=XX;Y
=YY;C1= COS (. 1);S1= SIN (. 1);N=64;GOTO
190
```

# ProDOS 磁盘操作系统入门(续)

廖 凯

## 6. 灵活使用 INPUT

在程序内大量灵活地使用 GET 或 INPUT 语句是很困难的。用 INPUT 或 GET 从键盘得到的许多字符将不能正确操作。例如一个回车键能够破坏一个菜单的结构,它会清除右边的任何字符。下面的程序包含一组能单独或共同使用的输入子程序,它能够阻止输入到程序内的任何字符或字符串显示在屏幕上。经设置 H% 和 V% 的值,可以将一个字符定位在屏幕的任何位置。L 变量用于调节一个字符串的长度。利用这些子程序可以在任何条件下有效地监视所有键盘命令。

第 1 部分设置参数作为用户对“Enter”的反应。第 2 部分在屏幕上定位字符。第 3 部分在屏幕上显示命令。第 4 部分先打印一个反相字符串,然后将字符打印到屏幕。可以使用退回键删除字符。此部分中允许删除键在任何情况下正确操作。第 5 部分最重要,所有键盘输入都来自这部分,可以认为是改进的 GET 语句。

在下面的例子中,任何类型的串可以用大写或小写字母输入,数字值也可以用这些子程序输入。

### 第 1 部分

```
100 D$ = CHR $(4)
110 PRINT D$ ; "PR #3"
120 B=1
130 H% = 20; V% = 10; GOSUB 1000; PRINT "Enter"
140 H% = 30; L = 10; GOSUB 1000; GOSUB 1250
```

### 第 2 部分

```
999 REM POSITION
1000 POKE 1403, H%; VTB V%; RETURN
```

### 第 3 部分

```
1150 REM SCREEN A COMMAND
1155 C% = 0; GOSUB 1500; C$ = CHR $(C%)
1160 IF C% > 64 AND C% < 91 THEN RETURN
1165 IF C% > 96 AND C% < 123 THEN C% = C% - 32; RETURN
1170 IF C% = 27 THEN RETURN
1175 IF C% > 41 AND C% < 58 THEN RETURN
1180 PRINT CHR $(7); GOTO 1150
```

### 第 4 部分

```
1250 REM INPUT A CHARACTER & USE DELETE
1260 INVERSE; FOR I=1 TO L; PRINT " "; NEXT; B$ = " "; L=L+1; NORMAL
1300 FOR I=B TO L
1305 INVERSE; GOSUB 1500; H1% = H%; C$ = CHR $(C%); IF C$ = CHR $(27) THEN RETURN
1310 IF C$ = CHR $(13) THEN NORMAL; RETURN
1315 IF C$ = CHR $(127) THEN 1355
1325 IF C$ >= CHR $(8) AND C$ <= CHR $(21) THEN PRINT CHR $(7); GOTO 1305
1330 IF I=L THEN 1345
```

```
1335 GOSUB 1000; PRINT C$ ; ; H% = H% + 1; B$ = B$ + C$
1340 NEXT
1345 IF RIGHT $(B$, 1) <> CHR $(13) THEN PRINT CHR $(7); GOTO 1305
1350 RETURN
1355 IF LEN (B$) = 0 THEN I=1; PRINT CHR $(7); GOTO 1300
1360 IF LEN (B$) = 1 THEN B$ = " "; C$ = " "; I=I-1; H% = H% - 1; GOSUB 1000; PRINT " "; H% = H% - 1; GOSUB 1000; B=1; GOTO 1300
1370 I=I-1; H% = H% - 1; GOSUB 1000; PRINT " "; ; GOSUB 1000; H% = H1% - 1; B$ = LEFT $(B$, I-1)
1380 GOTO 1305
```

### 第五部分

```
1500 REM GET A KEY
1510 C% = PEEK (-16384); IF C% < 128 THEN 1500
1520 C% = C% - 128; POKE 49168, 0
1530 IF PEEK (-16287) > 127 THEN C% = C% + 128
1540 IF PEEK (-16286) > 127 THEN C% = C% + 128
1550 RETURN
```

### GET A KEY

这个例子中最重要的是第 5 部分。行号 1510 读取由键盘打入的最后字符,如果已按过一个键,那么 PEEK(-16384) 得到一个比 127 大的数;如果没有键按过,重复执行此行。行号 1520 确定哪个键被按过。例如 A 是由键盘最后输入的, PEEK(-16384) 将等于 193, 由 C% 减去 128 得到 65 或 A 的 ASCII 值,并将其值存到 C% 内,而后用 POKE 49168, 0 清除键盘。

### POSITION

第 2 部分用于在屏幕上定位字符。POKE1403, H% 是用于在 80 列显示方式下水平定位字符。HTAB 只能用在 40 列方式下。H% 和 V% 变量是用于设置屏幕位置。注意这部分是用于整个程序定位。

### PROGRAM SET-UP

第 1 部分只说明了必须使用什么变量和什么格式。在行号 130 内,为通过用行号 1000 内的定位子程序而设置 H% 和 V%。行号 140 安置用户的设置,并用 L 变量设置输入字符的长度,程序然后转到行号 1250 的字符输入子程序。

### SCREEN A COMMAND

第 3 部分在这个例子中是不使用的,但对于在屏幕上显示最后按的键是很有用的。用户可以 GOSUB 这部分。行号 1155 设置 C% 为零,从内存清除以前的键,程序并转到 GET A KEY 程序,计算机等候用户输入单键字符。在按过键后,其 ASCII 值被转为一个串变量,

作为以后使用。行号 1160 比较从 A 到 Z 的 ASCII 值, 如果符合条件, 子程序返回。另外, 行号 1165 比较小写字母 A 到 Z 的 ASCII 值, 如果符合条件, 则从 C% 减去 32 以获得大写字母的 ASCII 值。用这种方法用户不必再检查大写和小写。行号 1170 检查 ESC 键, 这是取消一个命令或返回主菜单的一个好方法。行号 1175 将检查数字键是否被按过。如果上面的条件有一个不符, 那么行号 1180 将发出警告并重新执行这部分。

如果你想要使用箭头键, 就需增加如下行号:

```
1176 IF C% = 8 OR C% = 10 OR C% = 11 OR C% = 21
    THEN RETURN
```

在子程序返回后, 你必须用 POSITION 子程序重新定位。

#### DELETE AND PRINT

第 4 部分被分成五个部分: 反相显示并打印一个空串, 打印一个字符, 允许正确的数个字符被打印, 删除一个字符和检测回车键。

行号 1260 打印一个反相字符串。行号 1300 设置字符串的长度。行号 1330 将阻止打印任何额外的字符。行号 1335 根据每个 H% 值打印 C\$。行号 1315 检测 DELETE 键并执行删除操作。行号 1355 检查 B\$ 内的字符是否是当前的。如果不是, 删除则不成功, 重新执行循环。行号 1360 检测 B\$ 的长度, 若符合条件就打印一个空格以有效地删除字符。所有其它字符的删除是在行号 1370。记住要按以下步骤删除一个字符: 先退回, 而后打一个空格, 再退回。行号 1345 测试最右边的字符是否是回车。

#### 7. 数组清除器

在你重新定义一个数组时, 必须清除数组以获得更多的空间。这可以用下面的子程序来实现。

```
1 REM Array Eraser
10 DIM A(100), B(100), C(100)
100 FOR L=768 TO 823
110 READ N: POKE L, N
130 NEXT L
140 CALL 768, A: CALL 768, B
150 END
160 DATA 32, 177, 0, 32, 217, 247, 24, 160, 2, 165, 155,
    133, 66, 113, 155
170 DATA 133, 60, 200, 165, 156, 133, 67, 113, 155, 133,
    61, 136, 56, 165
180 DATA 109, 133, 62, 241, 155, 133, 109, 200, 165, 110,
    133, 63, 241
190 DATA 155, 133, 110, 160, 0, 32, 44, 254, 32, 163, 217,
    76, 152, 217
```

此程序是调用内存地址 768 开始的机器语言子程序来删除 A 和 B 数组。

### 三、屏幕输出

这部分将讨论可在屏幕上显示的不同输出类型。

#### 1. 有关指令

HTAB 控制光标的水平位置(1 到 40)  
VTAB 控制光标的垂直位置(1 到 24)

TAB(X) 将光标从右向左移到 X 位置  
POS 送回光标当前所在的水平位置  
SPC(X) 将光标从上一次光标位置向右移动 X 空格  
SPEED 设置字符输出的速度(0 到 255)  
INT(X) 送回一个近似 X 又不大于 X 的最大整数

在屏幕上显示输出有许多方法。下表列出了控制文本显示窗口的指令。HTAB 只能用于 40 列显示, 当用 80 列显示时, 就要用 POKE 1430, H(H 是水平位置) 代替 HTAB。对于 APPLE IIc 和具有 ROM Upgrade Kit 的 APPLE IIe, HTAB 已被扩展, 可以用于 80 列显示。

指 令	控 制 范 围
POKE 32, L	L 是左界, 在 0~39 之间
POKE 33, W	W 是宽度, 在 1~40 之间
POKE 34, T	T 是上界, 在 0~23 之间
POKE 35, B	B 是高度, 在 1~24 之间
POKE 1403, H	H 是宽度, 在 1~80 之间

#### 2. 文字居中

下面的程序是将文字显示在 80 列屏幕的中央。

```
5 HOME
10 L$ = "An example of text centering"
20 GOSUB 1250
999 END
1000 POKE 1403, H: VTAB 12
1100 RETURN
1250 H = INT(40 - (LEN(L$) / 2))
1260 GOSUB 1000: PRINT L$
1270 RETURN
```

如果没有 80 列卡, 就将行号 1250 改为:

```
1250 H = INT(20 - (LEN(L$) / 2))
```

#### 3. 列印文字

下面的程序用 READ/DATA 语句将文字输出到屏幕上。

```
10 HOME: VTAB 24
20 FOR I=1 TO 3
30 READ A$: PRINT A$
40 POKE 69, 255: CALL -856
50 IF I=2 THEN PRINT CHR$(13)
55 NEXT I
60 FOR I=1 TO 23: CALL -912
70 IF I=21 THEN FOR Z=1 TO 500: NEXT Z
80 FOR J=1 TO 100: NEXT J
90 NEXT I
99 END
100 DATA HELLO, This demonstration will show you a simple
    scrolling technique
110 DATA GOOD BYE
```

行号 10 将光标放在屏幕的底部。行号 40 产生一个延迟, POKE 69, X 设置延迟的长度, 从 0 到 255, CALL -856 执行延迟。行号 50 在 I=2 时加入一个回车字符。行号 60 开始从底部向上列印, CALL -912 将屏幕上的所有文字向上移动一行。行号 80 只作为一个延迟, 减缓列印速度。 (未完待续)

# 第十讲 文件管理

李文兵

本讲介绍文件管理概念、标准输入输出、文件输入/输出、低级文件 I/O 函数、随机文件的存取。

## 1. 文件的概念

(1) 文件系统及其功能 文件是指具有名字(文件名)的一组信息序列,系统和用户都可以将具有一定功能的程序模块、一组数据或文字命名为一个文件。一般的操作系统中都有专门负责文件管理的文件管理系统(filing system)。其基本功能有:

- ①能够建立新文件或删除旧文件;
- ②能对文件进行读写操作;
- ③能为文件自动分配物理空间,并建立逻辑结构与物理位置的映象关系;
- ④能对文件实施维护。

(2) 文件结构 文件结构分逻辑结构和物理结构。

①文件的逻辑结构 它有两种形式:一是记录式文件,亦称标准文件;另一种是非结构的流式文件。

记录式文件是一种结构文件。例如,DOS 系统的文件结构是由文件、记录块、记录三要素组成的。其结构如图 10.1 所示:

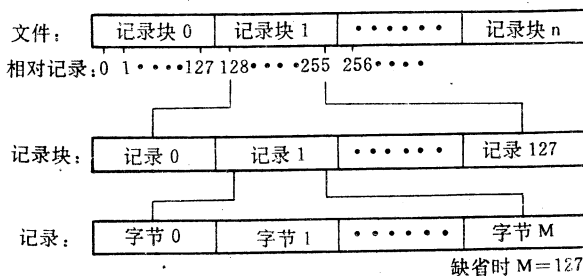


图 10.1 DOS 文件的逻辑结构

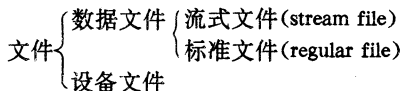
由该图可以看出:

- 一个 DOS 文件由若干个记录块组成。
- 一个记录块由 128 条记录组成。
- 一个记录由若干字节(缺省时为 128)组成。所有记录的长度都相等的文件,叫定长记录文件,其文件长度取决于记录数;记录长度不相等的文件,叫变长记录文件,其文件长度为各记录长度的累加和。

非结构流式文件是字符的有序集合。Turbo C 提供了这种文件逻辑结构形式。流式文件的长度是该文件所包含的字符数。这种文件,很便于操作系统进行管理;也适于用户进行文本处理,可不受任何约束地灵活组织其内部逻辑结构。

②文件的物理结构 它是指文件在外存上的存储结构,可分为连续、链接和索引三种组织方式。这种文件的读写方法有 2 种,即顺序读写和随机读写。

(3)C 语言的文件种类 C 语言中文件的分类如下:



在 C 语言中,I/O 设备可以当作文件来处理,这样的文件叫设备文件。设备文件既可按流式文件处理,又可按标准文件处理。常用的设备文件其名称、文件指针如表 10.1 所示。

表 10.1 常用设备文件

设备	文件名	文件指针
键盘	CON: /KYBD;	stdin
显示器	CON: /SCRN;	stdout/stderr
打印机	PRN: /LPT1;	stdout
串行	AUX: /COM1;	

## 2. 标准 I/O

前面我们介绍了数据的按格式输入输出。这里以文件处理为中心,介绍 I/O 处理。

(1)标准 I/O 所谓标准输入输出是指在操作系统所认可的所谓标准 I/O 设备上进行的输入输出。把标准输入文件拷贝到标准输出上的程序,如练习 10.1 所示。该程序是利用标准输入函数 getchar()和标准输出函数 putchar()实现的。程序中的 EOF 是文件结尾符。

```
C>type exp10-1.c
#include <stdio.h>
main()
{ char c;
  while((c=getchar())!=EOF)
    putchar(c);
}
C>exp10-1
学习 Turbo C,使用 Turbo C.
学习 Turbo C,使用 Turbo C.
^Z
C>
```

C 语言除了支持标准输入、标准输出外,还支持输入输出重定向,以及文件追加、管道功能。

(2)输入输出重定向 输入重定向的符号为<,其用法如下:

```
outf <inf
```



功能是 outf 文件使用函数 `getchar()`、`scanf()`、`gets(char *s)`、`cgets(char *s)`，从文件 `inf` 中读取数据，且实现 `outf` 的功能。

输出重定向的符号为 `>`，其用法如下：

```
inf >outf
```

功能是文件 `inf` 使用函数 `putchar()`、`printf()`、`puts(char *s)`、`cputs(char *s)`，把数据输出到文件 `outf` 中。

此外，`<` 和 `>` 还可以同时使用，如：

```
prog <newin >newout
```

功能是文件 `prog` 从文件 `newin` 中读取数据，把文件 `prog` 的处理结果输出到文件 `newout` 中。

因为文件 `expl0-1.C` 中含有 `putchar()` 函数，故执行 `expl0-1 >a.txt` 输出重定向命令，便可把从标准输入设备（键盘）上输入的数据，通过程序中的函数 `putchar()`，输出到 `a.txt` 文件中；即改变了 `putchar()` 函数的输出方向。如练习 10.2 所示。

```
C>expl0-1 >a.txt
学习 Turbo C,
使用 Turbo C.
C> type a.txt
学习 Turbo C,
使用 Turbo C.
^Z
```

同样，命令：

```
A>expl0-1 <a.txt >b.txt
```

改变了函数 `getchar()` 的输入方向，从文件 `a.txt` 输入数据；同时亦改变了 `putchar()` 的输出方向，把 `expl0-1` 要输出的数据输出到文件 `b.txt` 中。如练习 10.3 所示。

```
C>expl0-1 <a.txt >b.txt
C>type b.txt
学习 Turbo C,
使用 Turbo C.
文件追加 其符号为 >>，用法如下：
```

```
filed >>output
```

功能是把文件 `filed` 的内容附加到文件 `output` 的原有内容后。例如：

```
A>expl0-1 <a.txt >>b.txt
```

执行结果是把文件 `a.txt` 中的数据，追加到文件 `b.txt` 的原有内容后，如练习 10.4 所示。

```
C>expl0-1 <a.txt >>b.txt
c>type b.txt
学习 Turbo C,
使用 Turbo C.
学习 Turbo C.
使用 Turbo C.
```

(4) 管道 其符号为 `|`，用法如下：

```
inpr |outpr
```

功能是执行文件 `inpr` 和 `outpr`，且使 `inpr` 的标准输出作为 `outpr` 的标准输入，等效于：

```
inpr > temp
outpr < temp
```

```
del temp
```

文件 `expl0-5.c` 是对字符串计数的程序，执行命令：

`C>expl0-1 <expl0-1.c |expl0-5` 可以实现对文件 `expl0-1.c` 计数，如练习 10.5 所示。

```
C>type expl0-5.c
#include <stdio.h>
main()
{ int i=0;
  char c;
  while ((c=getchar()) != EOF)
    if (c != '\n') i++;
  printf("total character = %d\n", i);
}
```

```
C>expl0-1 <expl0-1.c |expl0-5
total character = 87
```

C 语言之所以能够实现 I/O 重定向、文件追加、管道等功能，就在于其输入输出函数能够重新分配标准输入输出设备，如练习 10.6 所示。

```
C>type expl0-6.c
#include <stdio.h>
#include "ctype.h"
main()
{ char c;
  while ((ch=getchar()) != EOF)
  { if (ch == '&')
    putc(ch, stderr);
    else
    putchar(ch)
  }
}
```

```
C>expl0-6 >x.txt
abcdef
a&b&c&d
&&&
&&&
&&&
a b c d
^Z
c>type x.txt
abcdef
abcd
```

```
a b c d
```

由文件 `expl0-6.c` 和执行 `type` 命令的结果来看，标准输出 (`stdout`) 分配给文件 `x.txt`，标准诊断输出分配给显示终端，可见，根据程序的功能，可以把从键盘输入的一部分字符（这里为 `&`）分离出来，输出到显示终端上（兼作标准诊断输出）。注意，当把标准输入 (`stdin`) 分配给其他文件时，便不能从键盘进行输入操作；要想从键盘上输入，就必须把标准输入重新分配给键盘。

### 3. 缓冲型文件的输入输出

经常用到的使用缓冲区的文件输入输出函数如表 10.2 所示。

表 10.2 使用缓冲区的文件 I/O 函数

函数名	功 能
fopen()	stream 的打开
fclose()	stream 的关闭
putc()	向 stream 输出一个字符
getc()	从 stream 输入一个字符
fseek()	查找 stream 的指定字节
fprintf()	输出到 stream
fscanf()	从 stream 输入
feof()	文件结尾(EOF)时为真
ferror()	发生错误时返回真
rewind()	把文件位置定位在文件的开头
remove()	删除文件

用如下命令行:

```
exp10_7 file1 file2
```

实现程序拷贝的程序如练习 10.7 所示。

```
C>type exp10_7.c
#include <stdio.h>
main(argc,argv)
int argc;
char *argv[];
{ FILE *from, *to;
  char ch;
  if(argc != 3)
    { printf("usage,exp10_7 from-file to-file\n");
      exit(1);
    }
  if((from=fopen(argv[1],"rb"))==NULL)
    { printf("can't open from-file\n");
      exit(1);
    }
  if((to=fopen(argv[2],"wb"))==NULL)
    { printf("can't open to-file\n");
      exit(1);
    }
  while(! feof(from))
    { ch=getc(from);
      putc(ch,to);
    }
  fclose(from);
  fclose(to);
}
C>type a.txt
学习 Turbo C,
使用 Turbo C.
C>exp10_7 a.txt exp.txt
C>type exp.txt
学习 Turbo C,
使用 Turbo C.
```

该程序的功能是把文件 file1 的内容拷贝到文件 file2 中。变量 from、to 是指向结构 FILE 的指针。因此，命令行第 1、第 2 参数是文件名，分别用 fopen() 函数打开。若能打开，则返回指向结构的指针值；若打开失败，

返回 NULL 的值。若两个命令行参数所指定的文件都打开成功，则用 feof() 函数检查 EOF，便可进行文件之间内容的拷贝了。

函数 getw() 和 putw() 可用于整型数的输入输出，即输出用 putw(n, file)，功能是把变量 n 的值输出到文件指针 file 所指定的文件中。输入用 getw(file)，功能是从文件指针 file 所指出的文件中读取下一个数据。利用这两个函数进行整数输入输出的程序，如练习 10.8 所示。

```
C>type exp10_8.c
#include <stdio.h>
#define NMAX 20
main()
{ FILE *from, *to;
  char *fname="fint";
  int i;
  if((to=fopen(fname,"wb"))==NULL)
    { printf("can't open to-file\n");
      exit(1);
    }
  for(i=0; i<NMAX; i++)
    putw(i,to);
  fclose(to);
  if((from=fopen(fname,"rb"))==NULL)
    { printf("can't open from-file\n");
      exit(1);
    }
  for(i=0; i<NMAX; i++)
    printf("%d ",getw(from));
  printf("\n");
}
C>exp10_8
```

```
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19
```

要随机存储文件，可使用能指定文件内任意位置的函数 fseek()。该函数的用法如下所示：

```
int fseek(FILE *fp, long int n-byte, int org)
```

其中，fp 为文件指针，打开文件时使用。n-byte 为 org 所设定场所的字节数。org 可使用如下宏：

```
SEEK-SET 文件的开头位置
SEEK-CUR 文件的当前位置
SEEK-END 文件的最末位置
```

使用函数 fseek() 进行文件随机存储的程序如练习 10.9 所示。设文件名从命令行引进。读文件使用函数 fread()，其用法如下：

```
int fread(void *buf, int num-bytes, int count, FILE *fp)
功能是从文件指针 fp 所指定的文件中读取每字段为 num-bytes 个字节，一共 count 个字段的数据到指针 buf 所指定的缓冲区中。
```

```
C>TYPE EXP10_9.C
```

```
/* example for fseek(), fread() */
#include <stdio.h>
#include "ctype.h"
#define SIZE 128
unsigned char buf[SIZE];
```

```

main(argc,argv)
int argc;
char * argv[ ];
{ FILE * fp;
  int sect,read_num;
  if(argc!=2)
  { printf("usage:exp10_9 filename\n");
    exit(1);
  }
  if((fp=fopen(argv[1],"rb"))==NULL)
  { printf("can't open file\n");
    exit(1);
  }
  do { printf("enter sector: ");
      scanf("%ld",&sect);
      if(fseek(fp,sect * SIZE,SEEK_SET))
      { printf("fseek error\n");
        exit(1);
      }
      if ((read_num=fread(buf,1,SIZE,fp))!=SIZE)
      { printf("EOF reached\n");

```

```

        exit(1);
      }
      display(read_num);
    } while(sect>=0);
  }
  display(n)
  int n;
  { int i,j;
    for(i=0;i<(n/16);i++)
      {for(j=0;j<16;j++)
        printf("%3x",buf[i * 16+j]);
        printf(" ")
        for (j=0;j<16;j++)
          { if(isprint (buf[i * 16+j]))
            printf("%c",buf[i * 16+j]);
            else printf(". ");
          }
        printf("\n");
      }
  }
}

```

A>exp10\_9 exp10\_9.obj

```

enter sector: 0
80 b 0 9 45 58 50 31 30 5f 39 2e 43 15 88 1d ...EXP10_9.C...
0 0 0 19 54 43 38 36 20 42 6f 72 6c 61 6e 64 ...TC06 Borland
20 54 75 72 62 6f 20 43 20 32 2e 30 20 fc 88 11 Turbo C 2.0...
0 0 0 e9 fc 31 18 17 9 45 58 50 31 30 5f 39 2e ...1...EXP10_9.
43 c2 88 f 0 0 e9 1 52 dc 16 7 53 54 44 49 C.....R...STDI
4f 2e 48 3b 88 10 0 0 e9 a0 8 8a f 8 53 54 O.H;.....ST
44 41 52 47 2e 48 fb 88 f 0 0 e9 a0 8 8a f DARG.H.....
7 43 54 59 50 45 2e 48 3d 88 6 0 0 e5 1 0 .CTYPE.H=.....
enter sector: 1
0 8c 88 a 0 0 e3 19 0 2 0 15 8 4 4f 88 .....0.
a 0 0 e3 18 0 2 0 15 19 4 3f 88 15 0 0 .....?....
e6 4 61 72 67 76 18 a 6 0 4 61 72 67 63 4 ..argv....argc.
a 4 0 ee 88 6 0 0 e5 1 6 0 86 88 48 0 .....H.
0 e2 0 5 6c 65 76 65 6c 4 0 5 66 6c 61 67 ...level...flag
73 a 0 2 66 64 8 0 4 68 6f 6c 64 8 0 5 s...fd...hold...
~62 73 69 7a 65 4 0 6 62 75 66 66 65 72 19 0 bsize...buffer..
4 63 75 72 70 19 0 6 69 73 74 65 6d 70 a 80 .curp...istemp..
enter sector: ^C

```

在使用文件找开函数 fopen()时,用参数“rb”指定为只读打开二进制文件模式。

删除文件,使用如下函数:

```
int remove(char * filename);
```

其应用程序如练习10.10所示。

```

C>dir *.bak
EXP10-1 BAK 97 1-01-80 12:05a
EXP6-19 BAK 549 1-01-80 12:02a
EXP10-6 BAK 182 1-01-80 12:15a
3 FILE(S) 12830720 bytes free
C> type exp10_10.c
#include <stdio.h>
#include "dos.h"

```

```

main()
{ char fp[80];
  printf("enter remove filename:");
  gets(fp);
  remove(fp);
}

```

```

C>exp10-10
enter remove filename:exp10-6.bak
C>dir *.bak
EXP10-1 BAK 97 1-01-80 12:05a
EXP6-19 BAK 549 1-01-80 12:02a
2 File(s) 12832768 bytes free

```

#### 4. 非缓冲型文件的输入输出

3中所介绍的文件输入输出函数,使用系统自动设

定的缓冲区。这里介绍向用户指定的缓冲区直接进行输入输出的函数。这样的输入输出叫无缓冲区的文件输入输出。这些函数叫低级输入输出函数,也叫操作系统连接的文件子程序。这些函数有:creat()、close()、open()、read()、write()、setmode()、rename()、unlink()。关于这些函数的功能、用法和返回值,请参考《IBM PC C语言例题 习题 库函数》。

这些低级输入输出函数,使用如下标题文件:

```
io.h
fcntl.h
\sys\stat.h
```

文件打开的一般形式如下:

```
int open (char * filename,int access, [int permis]);
```

其中 filename 为文件名;access 指定文件存取模式,取文件 fcntl.h 中所定义的任何一种,如图10.2所示:

```
/* The first three can only be set by open */
#define O_RDONLY 1
#define O_WRONLY 2
#define O_RDWR 4
/* Flag values for open only */
#define O_CREAT 0x0100 /* create and open file */
#define O_TRUNC 0x0200 /* open with truncation */
#define O_EXCL 0x0400 /* exclusive open */
/* a file in append mode may be written to only at its end. */
#define O_APPEND 0x0800 /* to end of file */
/* MSDOS special bits */

#define O_CHANGED 0x1000 /* user may read these bits,
but */
#define O_DEVICE 0x2000 /* only RTL\io functions may
touch */
#define O_TEXT 0x4000 /* CR-LF translation */
#define O_BINARY 0x8000 /* no translation */
/* DOS 3.x options */
#define O_NOINHERIT 0x80
#define O_DENYALL 0x10
#define O_DENYWRITE 0x20
#define O_DENYREAD 0x30
#define O_DENYNONE 0x40
```

图10.2 文件存取模式

另外,当指定为 O\_CREAT 时,permis 可指定为如下存取模式当中一个,如图10.3所示。

```
#define S_IREAD 0x0100 /* owner may read */
#define S_IWRITE 0x0080 /* owner may write */
```

图10.3 文件的存取模式

或者指定为:

```
S_IREAD, IWRITE
```

open()函数的返回值为文件句柄,读、写或关闭时使用。关闭文件时,使用 close()函数:

```
int close(int fd);
```

文件读写时,分别使用 read()和 write():

```
int read(int fd, void * buf, int size);
int write(int fd, void * buf, int size);
```

低级输入输出函数的例子如练习10.11所示。该程序的流程为:文件的生成、数据的写进、文件的关闭、文件以读方式打开、数据的读出、文件的关闭。

```
C>type exp10-11.c
#include <stdio.h>
#include "io.h"
#include "fcntl.h"
#include "sys\stat.h"
#include "string.h"
#define SIZE 80
#define ERR -1
main()
{ int i,fp;
char * fname="test.f";
char buff[SIZE];
if ((fp=creat(fname,S_IWRITE | S_IREAD))==ERR)
{ printf("can't create %s",fname);
exit(1);
}
/* write to file */
printf("enter string -->\n");
do{ gets(buff);
for(i=strlen(buff);i<SIZE;i++)
buff[i]='0';
if(write(fp,buff,SIZE)!=SIZE)
{ printf("error on write()\n");
exit(1);
}
}while (strcmp(buff,"end"));
close(fp);
/* read from file */
if ((fp=open(fname,O_RDONLY))==ERR)
{ printf("error on read()\n");
exit(1);
}
while(1)
{ if (read(fp,buff,SIZE)==0) break;
printf("%s\n",buff);
}
close(fp);
}
C>exp10-11
enter string -->
11111111111111
1111111111
1111111
111111
111
11
1
end
11111111111111
111111111
11111111
111
11
11
```

```
1
end
```

### 5、文件的随机存取

要想实现文件的随机存取,就得使用能使文件指针定位在任何位置的函数 `lseek()`,其用法如下:

```
long lseek(int handle,long offset,int fromwhere);
```

该函数把对应句柄值的文件指针从 `fromwhere` 位置移动 `offset` 个字节。其中, `fromwhere` 按如下取值:

```
SEEK_SET(0)   文件的开头
SEEK_CUR(1)   当前文件指针位置
SEEK_END(2)   文件的末尾
```

函数的返回值为文件指针的新位置距文件开头的偏移量。要知道文件指针的当前位置,使用函数 `tell()`:

```
long tell(int handle);
```

使用这些函数的程序,如练习10.12所示。该程序根据输入的所要打开文件的扇区号,便可显示其内容。若 `lseek` 的返回值为 `-1`,则意味着发生错误。该程序既能读出文本文件,也可以读出二进制文件。

```
C>type exp10-12.c
```

```
#include "io.h"
#include "fcntl.h"
#define SIZE 80
#define ERR -1

main()
{ int i,fh,n;
  char buf[SIZE],s[80];
  long offset;
  printf("Enter filename:");
  gets(s);
  if ((fh = open(s,O_RDONLY)) == ERR)
  { printf("can't open %s\n",s);
    exit(1);
  }
  while(1)
  { printf("Enter the sector number:");
    offset = (long)(atoi(gets(s)) * SIZE);
    if (lseek(fh,offset,SEEK_SET) == -1)
    { printf("lseek() error\n");
      exit(1);
    }
    if ((n = read(fh,buf,SIZE)) == ERR)
    printf("read error\n");
    printf("current file pointer = %ld\n",tell(fh));
    for (i = offset; i < offset + SIZE; i++)
    { printf("%3x",buf[i]);
```

```
if ((i%10) == 9)
printf("\n");
}
if (eof(fh) == 1) break;
}
close(fh);
}
```

```
C >exp10-12
Enter filename:exp10-12.obj
Enter the sector number,0
```

```
current file pointer=2
 a a 0 0 0 0 0 0 0 0
ff86 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0ff87 0 0 0
 0 0 0 0 0 0 0 0 0 0
 0 0ff88 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0ff89 0
 0 0 0 0 0 0 0 0 0 0
 0 0 0 0ff8a 0 0 0 0 0
Enter the sector number,1
```

```
current file pointer=82
31 0 70 31 30 5f 31 32 2e 6f
62 6a 0 0 0 0 0 0 0 0
 0 0 0 0 0 0ff8c 0 0 0
 0 0 0 0 0 0 0 0 0 0
 0 0ff8d 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0ff8e 0
 0 0 0 0ffd4ffff 6d 5 10 0
 0 0 0 0ffdeffff 36 6 10 0
Enter the sector number;
```

从执行结果来看,文件指针的位置为80的整数倍。但是,实际输出的字节数,0扇区和2扇区是不同的,并不都是80个字节。原因是0AH前面的0DH给删除掉了。若把 `open()` 说明为:

```
if (fh = open(s,o_RDONLY|O_BINARY) == ERR)
{ printf("can't open %s\n",s);
  exit(1);
}
```

则每个扇区的目标码便是80个字节,且上例中出现的0AH,也变成了0DH、0AH了。

到本期止,C语言初阶讲座已按计划刊出十讲。自讲座发表后,编辑部及作者都陆续收到许多读者的良好反映,为满足学习C语言的需要,本刊将于1992年内出版专辑,并请李文兵老师作重要补充,有兴趣的读者留意。

本编辑部



# SCB—1型单板单片机的存贮器改进

江 琪

SCB—1型单板单片机以它的实用、简便、经济等优点受到广大用户的欢迎,更好地完善它,又是广大用户的共同心愿。笔者在分析了SCB—1型单板单片机的硬件以后,改进了它的存贮器方面的设计,使之能够更加灵活地为用户服务。

## 一、问题的提出

我们知道,SCB—1型单板单片机具有8K EPROM和16K RAM,而对于有些用户来说,可能嫌8K ROM小,而16K RAM大。实际上,只要在加工该单板单片机时,稍加改动,就可解决这一问题,变8K EPROM、16K RAM为16K EPROM和8K RAM。这样一方面解决了上面所提出的问题,另一方面也使得用户可以在不改动原来监控程序的基础上,自己把执行程序固化在另一块 EPROM上运行。

## 二、改进原理

下面将与改进有关的原SCB—1型单片单板机原理图摘抄如图1所示:

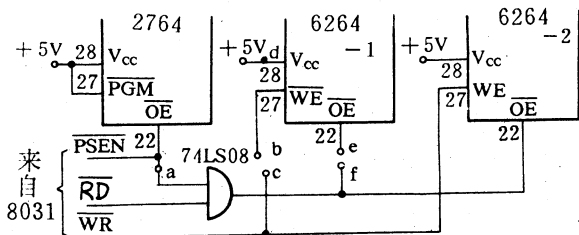


图1

由图1可知,2764、6264与8031的连接有以下不同:

1. 2764的第22引脚  $\overline{OE}$  为数据输出选通脚直接引至8031的  $\overline{PSEN}$  引脚;而6264的第22引脚  $\overline{OE}$  为读允许引脚是由8031的  $\overline{PSEN}$  引脚和  $\overline{RD}$  引脚经与门后引至的,这样做的目的是使6264既可以视作外部程序存贮器,亦可视为外部数据存贮器。

2. 2764的第27引脚  $\overline{PGM}$  需接到+5V,而6264的第27引脚为  $\overline{WE}$  需接至8031的  $\overline{WR}$  引脚。

显然,欲改变6264<sub>-1</sub>、为2764只要将原第27引脚接至+5V,第22引脚接至8031的  $\overline{PSEN}$  即可。这样,SCB—1型单板单片机的外部存贮器容量即为16K、外部数据存贮器容量为8K了。

## 三、改进方法

根据上面的分析可知,只需制板时在适当的位置留出几个插针,适当的插接插头就可以解决问题。如图2所示,可设置六个插针 a、b、c、d、e 和 f。当系统为8K EPROM和16K RAM时,b、c短接,e、f短接。当系统为16K EPROM和8K RAM时,b、d短接,e、a短接。在线路

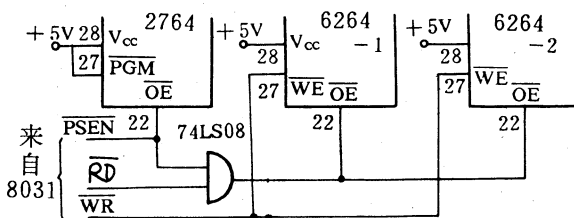
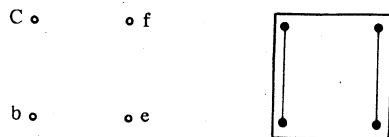


图2

板上可如图3所示按排插针和焊好插头。这样,当插头



(a)插针位置  
(b)插头接法

图3

插在上面位置时,系统为8K EPROM、16KRAM。如拆去6264<sub>-1</sub>,插上2764,然后将插头插在下面位置时,系统就为16K EPROM和8K RAM了。

### 全国软件资格水平考试函授辅导班 (第四期)招生

为配合人事部,国务院电子办每年进行的全国计算机软件资格和水平考试,由中国软件行业协会考试指导中心主办,清华大学计算机系,北京大学计算机系等校专家,教授主讲。

主要通过《全国软件人员水平考试丛书》、《软件函授通讯》等多种教材、资料,指导学员围绕考试大纲进行自学,优惠供应配套的教学录相带。在学员集中省市设三十余个面授辅导站,9月初结业模拟考试,推荐优秀者参加全国软件人员(初级程序员级和程序员级)竞赛。

报名费5元,学费初级程序员级110元、程序员级120元、高级程序员级150元、系统分析员级160元(暑期集中一月面授另交费140元)。现在开始报名。简章函索。

邮汇:北京学院路31号,中国软件行业协会考试指导中心 邮编:100083 传真:2024674 电话:2012233-322

银行汇款:北京工商银行海淀区东升分理处62学信,账号:07187-28

# MCS-51 单片机端口结构图示中的一个普遍错误

上海师大计算机科学系 王修才

(编者按:本文根据 P3 口的逻辑图的分析指出其逻辑图上的错误,希望有使用经验的读者就此发表一些看法。)

MCS-51 单片微机的应用适合于我国国情、对双增双节、产品的升级换代、机电一体化具有重大意义。但是,在我国已出版的有关 MCS-51 单片微机的教科书、资料中,在对 P3 端口的结构图示中有一个隐蔽性错误一直未被发现和纠正。

图 1 是国内出版的有关 MCS-51 单片微机资料对 P3 端口结构普遍采用的示意图。

按照 MCS-51 单片微机的规定,P3 口除了具有与 P1 口、P2 口同样的准双向功能(即作一般 I/O 口用)接口之外,还具有特殊的变异功能(有人称其为第二功能),P3 口各位的变异功能如下表所示。

按照 MCS-51 单片微机的规定,P3 口用作一般 I/

O 口使用的条件是变异功能输出端自动置“1”;P3 口

P3 口变异功能表

I/O 引脚	变异功能
P3.0	RXD——半行数据接收端
P3.1	TXD——串行数据发送端
P3.2	$\overline{\text{INT}}_0$ ——外部中断 0 请求
P3.3	$\overline{\text{INT}}_1$ ——外部中断 1 请求
P3.4	T <sub>0</sub> ——定时器/计数器 0 的外部输入端
P3.5	T <sub>1</sub> ——定时器/计数器 1 的外部输入端
P3.6	$\overline{\text{WR}}$ ——外部数据存储器写选通
P3.7	$\overline{\text{RD}}$ ——外部数据存储器读选通

用作变异功能使用的条件是位锁存器置“1”。但从图 1 所给出的端口结构来看,它只能实现用作一般 I/O 口使用,而无法实现全部变异功能的输入,况且要求变异功能的输入项(例如 RXD—P3.1,  $\overline{\text{INT}}_0$ —P3.2, T<sub>0</sub>—P3.4 等)从“读引脚”三态门的左边引入是错误的。按照常识,这些变异功能的输入项应该从它们各自的外引脚处输入;还有,图 1 中外引脚 P3.X 与“读引脚”三态门之间的一个门电路 G 在这里又有何用呢?为此,我查遍了国内出版的有关 MCS-51 单片微机的教科书和资料,均存在同样问题,如出一辙,最后追查到美国 Intel 公司出版的 MCS-51 单片微机用户手册——MCS-51 Family of Single Chip Micromputer user's manual,它也是像图 1 所示那样画出的。我估计这是原文出错,因此就一错百错了。

我认为 MCS-51 单片微机的 P3 端口的端口结构示意图应当像图 2 所示那样才是正确的,能自圆其说的。

按照图 2,门电路 G 的作用也就清楚了。按照图 2,P3 口作为一般 I/O 口的使用条件是:变异功能的内部输入、输出端自动置“1”,把与非门 H 和三态门 G 打开;P3 口用作变异功能使用条件是:相应位锁存器置“1”,变异功能中的输出项从内部变异输出端送出,经与非门 H 和驱动器至外引脚处输出;变异功能项的输入项,在对三态门 G 控制端自动置“1”的条件下从相应外引脚处输入。

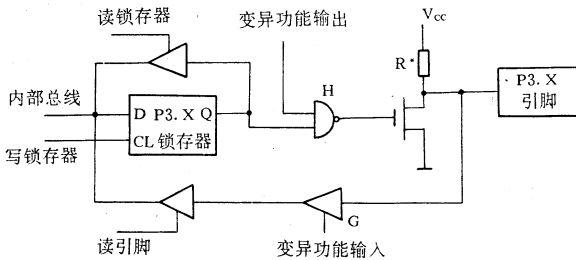


图 1

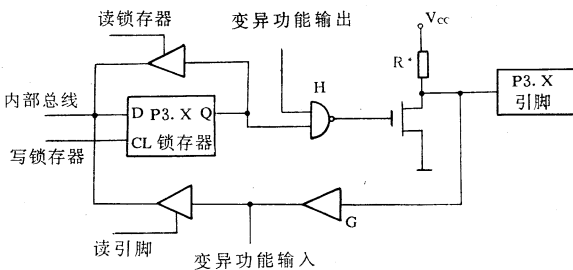
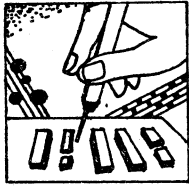


图 2



## 学装微电脑

# 串行 I/O 部件的制作与实验

易齐干

关于并行 I/O 8255A 的基本用法,在  $\mu p80$  套件中已经介绍了。

这里介绍串行 I/O 芯片 8251A 的基本使用方法。8251A 也称 USART (Universal Synchronous/Asynchronous Receiver/Transmitter) 通用同步/异步接收器/发送器。

串行传送的示意图如图 1 所示。即使用一条数据线可将 8 位数据由 A 传到 B。

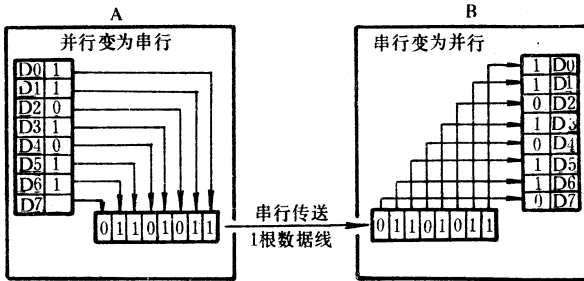


图 1 串行传送方式

和资料。

### 1. 非同步(异步)通讯

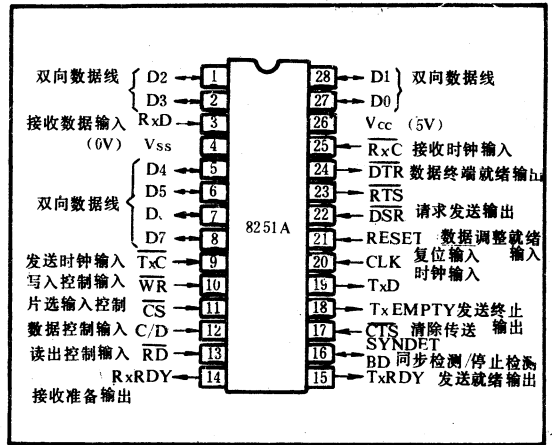


图 2 8251A 引脚

下面结合 8251A 讨论如何实现非同步(异步)通讯的基本方法。图 2 给出 8251A 管脚说明。由于篇幅有限,各个管脚的作用不再叙述,读者可参阅有关书籍

非同步(异步)通讯的数据有起始位、数据位、奇偶校验位和停止位所表示的字符所组成。是一个字符挨着一个字符地传送。有如下主要特征:

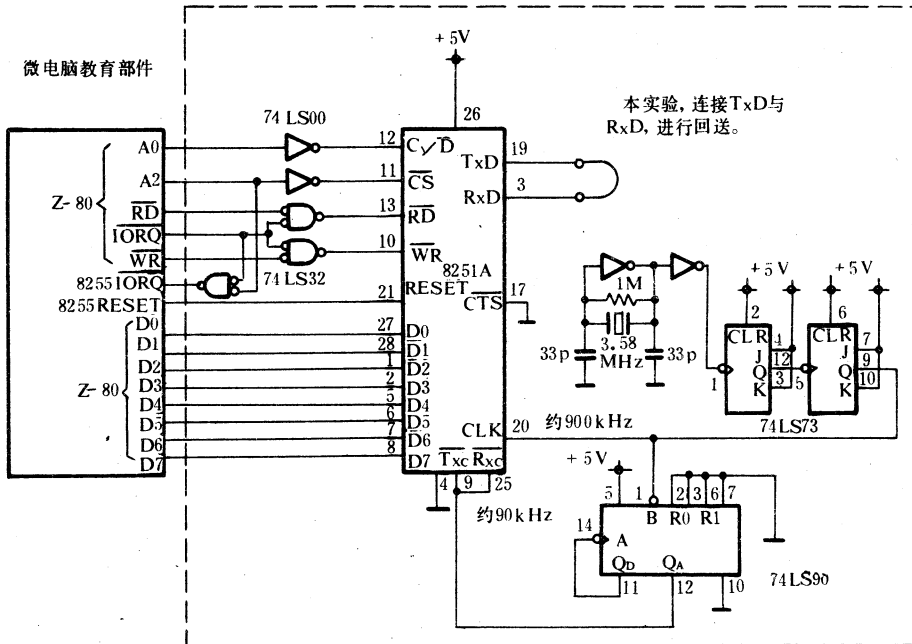


图 3



(1)根据方式指令码的设定传送5~8位数据。  
 (2)时钟速率为通讯波特率的1、16、64倍。波特率是数据传送速度。

- (3)有1、1.5倍2个停止位。
- (4)非同步方式波特率为0~9.6K波特。

2. 串行输入输出部件的制作。

串行输入输出部件的电路原理图如图3所示。时钟脉冲信号由3.58MHz石英振子和CMOS4069产生。经过74LS73的1/4分频变为900KHz输入给8251A的CLK出脚。再经过74LS90的1/10分频得到90KHz的脉冲输入给 $\overline{R_{xc}}$ 、 $\overline{T_{xc}}$ 出脚。

$\mu p-80$  套件的输入输出部件(8255A)A、B、C各端口和CW寄存器的初始化地址分配为00H-03H,则8251A的C/ $\overline{D}$ (控制/数据)分配为04H、05H。使用事例如下:

IN A, (04H) 表示8251A的工作状态的状态码送入CPU的累加器A。

OUT (04H), A 表示CPU向8251A输出8251A的控制码。

IN A, (05H) 表示8251A接受的数据送入CPU的累加器A。

OUT (05H), A 表示CPU的累加器A的数据由8251A向外部传送。

地址决定表如表1所示。

表1 地址决定表

		地址 (2进制)								地址 (16进制)
		A7	A6	A5	A4	A3	A2	A1	A0	
输入输出部件 (8255A)	A口	x	x	x	x	x	0	0	0	00H
	B口	x	x	x	x	x	0	0	1	01H
	C口	x	x	x	x	x	0	1	0	02H
	CW寄存器	x	x	x	x	x	0	1	1	03H
8251A	控制	x	x	x	x	x	1	x	0	04H
	数据	x	x	x	x	x	1	x	1	05H

3. 串行传送实验。

设计一个封闭反馈实验,开关的ON/OFF状态送入 $\mu p-80$ 套件的输入输出部件A口,输入输出部件的C口连接H/L电平显示部件,8251A串行传送、接收开关状态信号,如图4所示。

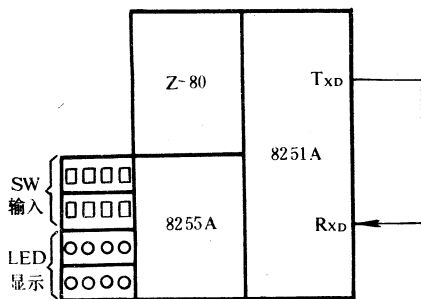


图4 封闭反馈实验示意图

8251A是可编程LSI集成电路。使用时首先要进行初始化,决定异步方式码和指令字格式。分别如图5、图6所示。它们均为控制码。如前所述,应该送入I/O地址04H中。

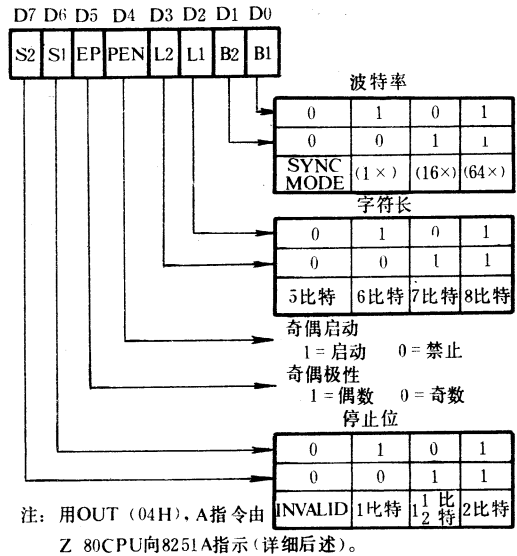
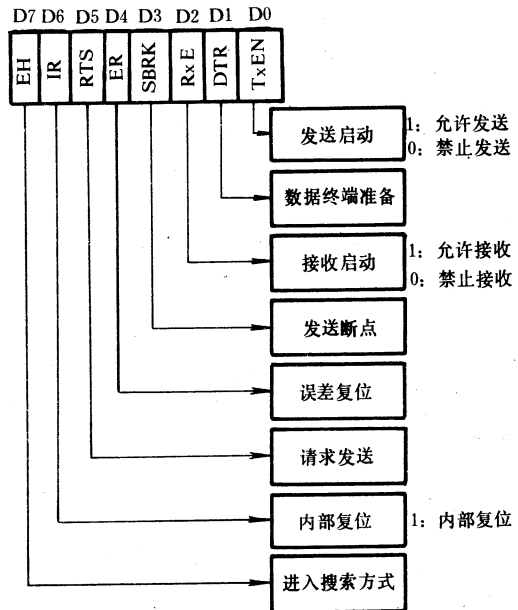


图5 方式码



注: (1)本实验中,仅使用D6, D2, D0。其他详细说明请参照文献(4)-(6)。

(2)用OUT (04H), A指令由Z-80CPU向8251A指示

图6 指令字格式

图6中,取D<sub>0</sub>=1为内部复位,其它位均为0,这样得到指令字格式为

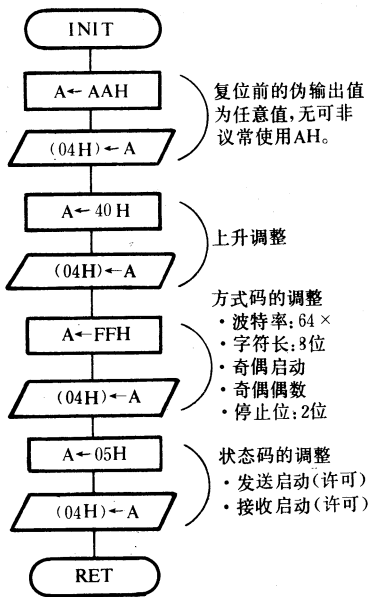
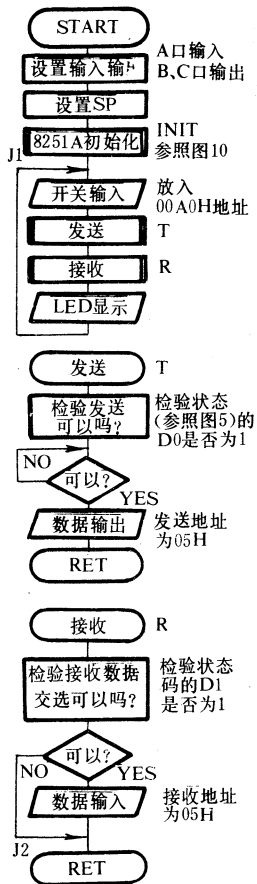


图7 8251A 初始化流程图

40H。异步方式码为FFH。为使8251A能传送和接收数据,指令字格式D<sub>0</sub>=1, D<sub>2</sub>=1,其它位均为0,指令字格式为05H,应放入I/O的04H地址中。

封闭反馈实验流程图和程序清单如图7、图8所示。它包括8251A的初始化、开关输入、传送数据、接收数据和H/L电平显示子程序。

执行这个程序,与开关ON/OFF状态相对应,H/L电平显示中的发光二极管应该亮。



● RAM区域	
00A0H	00A1H
输入开关的“通”“断”状态	存放接收数据

标记	助记符	地址	机械语	注释
START	LDA, 90HH	0000	3E 90	)设置输入输出
	OUT(03H), A	0002	D3 03	
	LD SP, 0100H	0004	31 00 01	设置SP
	CALL INIT	0007	CD 48 00	8251A初始化
J1	LD HL, 00A0H	000A	21 A0 00	)开关输入
	IN A, (02H)	000D	DB 00	
	LD (HL), A	000F	77	)发送
	CALL T	0010	CD 60 00	
	INC HL	0013	23	)接收
	CALL R	0014	CD 70 00	
	LD A, (HL)	0017	7E	)LED显示
	OUT(02H), A	0018	D3 02	
	JP J1	001A	C3 0A 00	
INIT	LD A, AAH	0048	3E AA	
	OUT(04H), A	004A	D3 04	
	LD A, (04H)	004C	3E 40	
	OUT(04H), A	004E	D3 04	
	LD A, FFH	0050	3E FF	
	OUT(04H), A	0052	D3 04	
	LD A, (05H)	0054	3E 05	
	OUT(04H), A	0056	D3 04	
	RET	0058	C9	
T	IN A, (04H)	0060	DB 04	
	RRC A	0062	CB 0F	
	JP NC, T	0064	D2 60 00	
	LD A, (HL)	0067	7E	
	OUT(05H), A	0068	D3 65	
	RET	006A	C9	
R	IN A, (04H)	0070	DB 04	
	RRC A	0072	CB 0F	
	RRC A	0074	CB 0F	
	JP NC, J2	0076	D2 7C 00	
	IN A, (05H)	0079	DB 05	
	LD (HL), A	007B	77	
J2	RET	007C	C9	

图8 8251 封闭反馈实验程序

## “微电脑控制技术”函授学习班总结

易齐干

本刊与天津纺织工学院共同开办结合μp80微电脑的“微电脑控制技术”函授学习班第一期圆满结束了。全国各地学员81人。其中多数学员结合本职工作带着实践中的课题参加学习。

这期以电子工业出版社发行的《微电脑图解教程》为实验教材。接着本刊90年9期刊登的教学大纲向学员发寄讲义,讲义密切配合“教程”一书所讲述的硬件,重点讲述汇编语言程序的编制技巧;常用的子程序;子程序的组合;特殊程序的编制以及应用实例。

学员按照计划,认真阅读教材,边实验,边思考,由浅入深循序渐进,将疑难问题及时与函授教员通信联系,均得到满意解答。

最后,进行了开卷考试。学员成绩均为优良,天津纺院成人教育处颁发了单科结业证。

这期函授班学员普遍反映效果很好。讲授内容通俗易懂,结合实际,它作为一座桥梁将微电脑爱好者由入门引进到应用阶段。

(转35页)

# 将 MP-1 电脑的时基信号改为晶体振荡

邓文超

MP-1 电脑简单易装,可广泛应用于各种自动控制场合之中。但若用于定时控制,则显得时间精度稍差一些,原因是 MP-1 电脑的时基信号是由简单的 RC 电路产生,(见图 1)其振荡频率随环境的温度、湿度而变化。笔者取 MP-1 的一位输出作为数字电子钟的 50Hz 时基信号,日误差大于 40 秒。

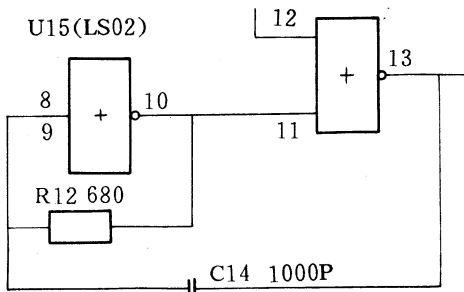


图 1

为了提高精度,笔者将 MP-1 改为晶体振荡,使 MP-1 的频率稳定性与普通计算机相仿。(见图 2)。

改装步骤如下:

1. 按照图 2,准备一只 1MHz 的石英晶体(MP-1 原设计工作频率为 500kHz,因 500kHz 的晶体价昂难觅,故改用 1MHz,查资料知 6502 CPU 的工作频率可为 1MHz); 0.01 $\mu$  涤纶电容一只; 1k $\Omega$  金属膜电阻二

只。

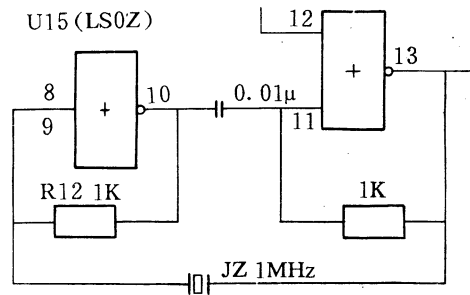


图 2

2. 用刀片割断连接 U15(LS02)第 10 脚与第 11 脚之间的印刷线路(在插件面)。
3. 焊下 C<sub>14</sub>,并在该位置焊上 1MHz 晶体。
4. 将 R<sub>12</sub>的值换为 1k。
5. 在电路版的焊接面直接将一个 0.01 $\mu$  的电容跨焊在 U15(LS02)的第 10 脚与第 11 脚之间。
6. 以同样方式焊接一只 1k 电阻在第 11 脚与第 13 脚之间。

只要元件完好,焊接无误,完成以上六点工作之后,MP-1 就变成工作频率非常之精确稳定的单板电脑了(无须调式)。

## 邮售中华学习机通用 EPROM 读写卡

南京华山电子技术服务部向读者提供本利 1990 年 8 期介绍的通用 EPROM 读写卡,该卡有以下特点:

- 功能齐全:可读、写或复写 27 系列 2K 至 32K 字节的各种型号 EPROM 芯片;扩展其它系列芯片也极方便。

- 操作简便:有详尽的汉字提示和过程显示;可不关主机更换操作芯片,便于连续读写大规模软件;可从任意地址开始读写任意字节数据,便于分段固化大容量芯片。大批量固化 EPROM 芯片只需一次建立工作参数后即可连续运行。

- 功能扩展:在上述基本型读写卡的基础上已可作三项扩展(详见下表),读者可任选  $n$  项扩展功能。

邮售价格表:

序号	品种	主要功能说明	套件价 (元)	成品价 (元)
1	基本型	见 90 年 8 期文章介绍	110	120
2	扩展 I	增加自带监控系统,开机用 PR * n 命令联机运行。	35	
3	扩展 II	增加自备编程电源、减少外接电源的麻烦。	18	
4	扩展 III	增加专用 EPROM 读写插座(活络带锁紧手柄)	65	

南京华山电子服务部 地址:南京市小营路 2 号  
 邮码:210016 电话:403872 开户行:南京市工行后宰门分  
 理处 帐号:1686007 联系人:陈万伟

## 第二章 FBASIC 的基本语句

于 春

上一章介绍了几个初步的 FBASIC 语句,从本章开始,陆续介绍 FBASIC 中用的较多的,最基本的语句。

### 一、循环语句

FOR~TO~STEP——NEXT

#### 1. 循环语句的结构

循环语句由 FOR、NEXT 两句组成。FOR 语句是循环的起点,是循环的说明语句,由它决定循环次数。NEXT 语句是循环的终端语句。FOR 与 NEXT 语句必须成对出现。

FOR 语句的形式为

FOR i=L TO m STEP S

i:循环控制变量,可用任一简单变量表示。

L:循环控制变量初值。

m:循环控制变量终值。

S:循环变量每一次循环的增量,也叫做步长。步长为 1,可以省略 STEP S。

例 1. 打印 1~30 整数的平方、立方表。

```
10 FOR A=1 TO 30 STEP 1
20 PRINT A, A * A, A * A * A
30 NEXT
```

上程序中 10 句也可写为 FOR A=0 TO 30。

步长也可以为负值。当步长为负值时,终值应小于初值。

例 2.

```
10 FOR X=20 TO 1 STEP -2
20 PRINT X,
30 NEXT
```

例 3. 求 5 的阶乘 5!

```
10 A=1
20 FOR B=1 TO 5
30 A=A * B
40 NEXT
50 PRINT A
```

RUN 后 显示 120

循环变量的初值、终值、步长必须控制在 -32768 ~32767 之间取值。否则,程序不能输入,且屏幕显示 OV ERROR 溢出提示。

FBASIC 允许输入简写语句,简写输入后,计算机自动翻译成原语句。如 FOR—NEXT 语句可简写为 F. ~TO~ST. ——N.。注意,简写中字母后的点不能

丢失。为节约篇幅,以后介绍的语句和举例均用简写。读者练习时,在弄通语句的结构后,建议使用简写输入程序,这样可以提高程序的编写、输入速度。同样 LIST 的简写为 L. PRINT 的为 P. 或?;END 的为 E. 等。

例 4. 求前 n 项和  $\sum_{n=1}^{100} n$ 。

```
10 C=0
20 F. D=1 TO 100
30 C=C+D
40 N.
50 P. C
RUN 后,显示 5050
```

#### 2. 多重循环

以上介绍的是单循环,FBASIC 允许两重以上的多重循环嵌套。

例 5. 打印九九乘法表。

```
10 F. A=1 TO 9
20 F. B=1 TO A
30 C=A * B
40 P. B " * " A " = " C,
50 N.
60 N.
```

例 5 也可以写为

```
10 F. A=1 TO 9;F. B=1 TO A
20 P. B " * " A " = " B * A
30 N. : N.
```

### 二、转移语句

计算机在执行程序时,一般是按行号由小到大依次进行,但由于解决问题的需要,往往要求改变执行顺序,转到其它行号执行,这一现象叫转移或叫转向。实现这一功能的语句叫转移语句。转移语句有三种:无条件转移语句(GOTO);条件转移语句(IF—THEN);开关转移语句(ON—GOTO)。下面分别介绍。

#### 1. 无条件转移语句

GOTO 简写 G.

GOTO 语句的格式为

G. <行号>

当程序执行到这一语句时,把程序的运行无条件的转移到 GOTO 语句中规定的行号去,并从这一行号起继续执行。

#### 2. 条件转移语句

IF—THEN 简写 IF—T.

条件转移语句也叫 IF 语句,使用该语句进行条件判断,决定某种功能。

IF 语句的格式为

IF<关系式>THEN<行号>

IF<关系式>GOTO<行号>

IF<关系式>THEN GOTO<行号>

IF<关系式>THEN<语句>

值得说明的是:关系式后可以用 THEN,也可以用 GOTO,也可以用 THEN GOTO,效果是一样的。头三种格式可以任用。

IF 语句的作用是:若关系式成立,就转到 THEN、GOTO 后的行号继续执行或执行 THEN 后面的语句;若关系式不成立,就跳过 IF 语句,继续执行后面的语句。

例 6. 用转移语句求前 n 项和  $\sum_{n=1}^{100} n$

```
10 A=B=0
20 A=A+1 : B=B+A
30 IF A<100 T. 20
40 P. "B=" B
50 E.
```

RUN 后,显示 5050

程序中 10 行对 A,B 赋以初值,20 行进行连加运算,30 行为条件判断。当执行到 30 句时,计算机进行判断 A 是否小于 100,若小于 100 则转到 20 行继续进行加法运算。若 A 等于 100,关系式 A<100 不成立,则跳过 30 行,执行 40 行。30 行关系式后若换用 GOTO 20 或 THEN GOTO 20 效果一样,读者可自己验证。

例 6 还可以改为如下程序

```
10 A=B=0
20 A=A+1 : B=B+A
30 IF A=100 T. P. "B="B : E.
40 G. 20
```

30 行中的条件关系式改为 A=100,它表示若 A=100 时,则执行 THEN 后的语句;若 A≠100 则跳过 30 行,执行 40 行。

通过例 6 可以发现,程序的编制是很灵活的,解决同一个问题,可以编制不同的程序来解决。

由于循环语句、转移语句是程序设计中常用,最有用的语句,也是最难接受,最易出错的语句,因此,本文通过较多的示例介绍这两语句的配合使用。

例 7. 有一算式,式中每一种动物代表一个阿拉伯数字,试求出各代表的数字。

算式如下:

马蛇×马马+牛牛=1991

设 A=马, B=蛇, C=牛, 程序如下:

```
10 F. A=0 TO 9 : F. B=0 TO 9 : F. C=0 TO 9
20 IF(A * 10+B) * (A * 11)+C * 11=1991
```

T. P. "Ma="A, "She="B, "Niu="C : E.

30 N. : N. : N.

RUN 后显示

Ma=4 she=3 Niu=9

THEN 后可以跟多个语句。程序执行完第一个语句再执行第二个语句……直到执行完所有语句再执行下一行。20 行中,THEN 后的第二语句是 END,所以程序执行到此就停机。

上程序只求出了例 7 的一组解,可能还有其它解,这只要将 20 行的 END 语句移到最后一行(添 40 行),则可求出全部解。结果如下:

```
Ma=4 she=3 Niu=9
Ma=4 she=4 Niu=5
Ma=4 she=5 Niu=1
```

例 8. 民间趣题

用 100 匹马驮 100 块瓦,大马驮 3 块,中马驮两块,两个小马驮一块,求大、中、小马各多少?

这是一个多解问题,先列出求解方程

设大马为 X,中马为 Y,小马为 Z 匹,则可列出方程组:

$$\begin{cases} X+Y+Z=100 \\ 3X+2Y+\frac{1}{2}Z=100 \end{cases}$$

整理得

$$\begin{cases} X+Y+Z=100 \\ 6X+4Y+Z=200 \end{cases}$$

这道题用代数方法求解,需用试商法,比较麻烦。若用计算机求解,则比较简单。程序设计思路是:先设定单用一种马驮瓦最多用多少匹,决定循环变量终值,则大马最多用 34 匹;中马最多用 50 匹;小马用 100 匹(因出题限定 100 匹)。程序如下:

```
10 F. X=1 TO 33 : F. Y=1 TO 50 : F. Z=1 TO 100
20 IF X+Y+Z=100 T. 40
30 G. 70
40 IF 6 * X+4 * Y+Z=200 T. 60
50 G. 70
60 P. "X="X, "Y="Y, "Z="Z
70 N. : N. : N.
80 E.
RUN
X=2 Y=30 Z=68
X=5 Y=25 Z=70
X=8 Y=20 Z=72 .....
```

本题共有 6 组解。

由于程序中有三重循环嵌套,程序执行速度较慢。

若把 Z 的循环变量初、终值改为

F. Z=60 TO 80

速度要快一倍。(由于小马 60 匹才驮 30 块,下余 40 匹马,70 块瓦,若全部用中马驮,也仅用 35 匹,所以小马最少不能低于 60 匹。同理,80 匹‘小马驮’40 驮 40 块瓦,下余 20 匹马 60 块瓦,全部用大马刚好驮完,所以

小马最多不超过 80 匹。)

例 9. 山上一群羊,山下一群羊,若山上拿一只羊给山下,上、下羊数相等;若山下拿一只羊给山上,则山上羊数是山下的两倍,山上山下各有多少只羊?

由于山上、山下羊数有一只波动,就产生倍级的量变,所以山上山下羊数各不会超过 10 只。设山上有 A 只,山下有 B 只。

```
10 F. A=1 TO 10 : F. B=1 TO 10
20 IF A-1=B+1 AND A+1=(B-1)*2 T. P.
"Shan Shang="A,"Shan Xia="B
30 N. : N.
40 E.
RUN
Shan Shang=7   Shan Xia=5
```

20 行使用了逻辑判断式 AND,它表示两关系式相与。即若两关系式都成立,则执行 THEN 后的语句;若有一个不成立,则跳过 20 行而执行 30 行。使用逻辑判断式,可使程序简捷。大家可以发现例 8 的程序可以改编如下:

```
10 F. X=1 TO 20 : F. Y=1 TO 30 : F. Z=60 TO
80
20 IF X+Y+Z=100 AND 6*X+4*Y+Z=200
T. P. "X="X,"Y="Y,"Z="Z
30 N. : N. : N.
40 E.
```

例 10 《电子与电脑》1989 年 9 期曾刊登了美国物理学家爱因斯坦做过的数学趣题,有一阶梯(1)每步跨 2 阶,剩 1 阶;(2)每步跨 3 阶,剩 2 阶;(3)每步跨 5 阶,剩 4 阶;(4)每步跨 6 阶,剩 5 阶;(5)每步跨 7 阶,剩 0 阶;问阶梯至少应有多少阶?

原文程序用了 18 行,本文编程如下:

```
10 F. N=0 TO 500 ST. 7
20 IF N MOD 2=1 AND N MOD 3=2 AND N
MOD 5=4 AND N MOD 6=5 T. P. N : E.
30 N.
RUN 119
```

程序中使用了 MOD 运算,即求余运算,关系式中只判断余数如何而不管商是多少。

例 11. 民间趣题

兄弟三人打鱼,老大把打的鱼分成三等分,最后剩了一条,由于不好分,老大把这条鱼扔进了河里,拿了一份回家。老二不知道老大已分了鱼,就把剩下的两份鱼合在一起,又三等分,结果也剩了一条,他也扔进了河里,拿走了一份。老三也不知道已分鱼,他又把剩下的两份合在一起又三等分,又剩了一条,又扔进了河里。问最少有多少条鱼才能满足以上分法。

该题比较复杂,得数必须满足除以 3 余 1 商乘以 2 之积除以 3 余 1……。该题虽有无数个解,但用代数求解也比较麻烦。设总数为 A,程序如下:(设定循环终值为 100)

```
10 F. A=1 TO 100
```

```
20 B=A/3 : C=B/3
```

```
30 IF A MOD 3=1 AND 2*B MOD 3=1 AND 2*
C MOD 3=1 T. P. "A="A : E.
```

```
40 N.
```

```
RUN A=25
```

若求 100 内的解,只须把 30 行末的 END 删掉即可求得 25,52,79。若求 1000 内的解,只须把循环终值改为 1000。

在 IF 语句中,THEN 后面的语句也可以是 IF 语句,这称为 IF 语句的嵌套,因此上例 30 行也可以写为如下形式:

```
30 IF A MOD 3=1 T. IF 2*B MOD 3=1 T. IF 2*
C MOD 3=1 T. P. "A="A,
```

两种形式,效果一样,读者自己验证。在使用 IF 语句嵌套时,IF,THEN 必须成对出现。

例 12. 民间趣题,留给读者自己编制程序。

龟、蟾、蟹共 19 只,100 条腿进行分配,求各有多少只?(注:龟四条腿,蟾四条腿,蟹十条腿。)

3. 开关转移语句

ON—GOTO 简写 O.—G.

开关转移语句起分支转移的作用,它控制转向,随变量的值不同而分别执行不同的功能。开关语句的格式为

```
ON<表达式>GOTO<行号 1>、<行号 2>……
```

```
ON<表达式>GOSUB<行号 1>、<行号 2>
```

……

```
ON<表达式>RETURN<行号 1>、<行号 2>
```

……

```
ON<表达式>RESTORE<行号 1>、<行号 2>
```

……

在一般程序设计中,开关,语句用的不多,本文简要介绍其应用的方便之处。

在举例前,先介绍一个常用的语句——键盘输入语句 INPUT 简简写 I.

使用 INPUT 语句,可以不把数据编在程序里,而在程序执行时,临时从键盘输入。当程序运行时,计算机执行到 INPUT 语句,自动停止运行,打出“?”等待用户输入数据。输入数据后,再继续运行程序。下面结合开关语句示例介绍 INPUT 语句的使用。

例 13 分段统计学生成绩,分不及格、60~69、70~79、80~89、90~99、100 六档。

使用以上所学语句,编程程序如下:

A. 回车(自动提供行号)

```
10 A=B=C=0 : D=E=F=0
```

```
20 I. X
```

```
30 IF X>100 T. 100
```

```
40 IF X<60 T. A=A+1 : G. 20
```

```
50 IF X>=60 AND X<70 T. B=B+1 : G. 20
```

```
60 IF X>=70 AND X<80 T. C=C+1 : G. 20
```

```
70 IF X>=80 AND X<90 T. D=D+1 : G. 20
```

```
80 IF X>=90 AND X<100 T. E=E+1 : G. 20
```

```

90 IF X=100 T. F=F+1 : G. 20
100 P. "<60="A, "60~69="B, "70~79="C,
"80~89="D, "90~99="E, "100="F
110 E.

```

按 Ctrl+C 键(解除自动提供行号)

程序开始前 A. 为自动提供行号语句 AUTO 的简写。当程序较长时,每输入一行要键入一次行号很麻烦,为提高编程速度,先键入 AUTO 语句,回车后则显示 10 行号,该行编完回车后,自动另起一行,显示 20 行号……。程序编写完毕,同时按 ctrl+c 键可解除自编行号功能。

上程序中,20 行为键盘输入语句,程序运行到此自动停止运行,等待输入数据。30 行为键盘输入结束语句,若键入大于 100 的任何数,程序运行结束,打印结果。

上程序 RUN 后,显示?号后跟闪烁光标,输入一个得分回车后,则另起一行出现?等待再次输入,直到输入大于 100 的数,立即打印各档成绩人数。

若使用开关转移语句,例 13 的程序要简捷的多。

A.

```
10 A=B=C=0 : D=E=F=0
```

```
20 I. X
30 IF X>100 T. 110
```

```
40 O. X/10+1 G. 50,50,50,50,50,50,60,70,
80,90,100
```

```
50 A=A+1 : G. 20
```

```
60 B=B+1 : G. 20
```

```
70 C=C+1 : G. 20
```

```
80 D=D+1 : G. 20
```

```
90 E=E+1 : G. 20
```

```
100 F=F+1 : G. 20
```

```
110 P. "<60="A, "60~69="B, "70~79="C,
"80~89="D, "90~99="E, "100="F
```

```
120 E.
```

程序中 40 行为开关语句,X/10+1 是为了使得分从 0~59 变化时,关系式值为 1~6,均转 50 行;当得分为 60 时,关系式值为 7,则转 60 行,余者依此类推。

至于开关语句的其它形式,留待以后示例中附带介绍。

(未完待续)

(接 30 页)

现将函授过程中学员来信摘录如下:

广西柳州赵克明学员讲:“…… $\mu$ p80 已组装完毕,其原理也已搞清。而且,我还给它扩充了内存。学习之余有个强烈的念头——给它装备一个 CRT 控制器,让它在电视机或在显示器上输出信息。”

苏州某乡办企业赵斌学员讲“……我们乡办厂搞微电脑应用多为外面引进人材,而本地没有很好的继承人,作为本乡人,我想自己具有这方面特长会有远大前景的。通过函授学习,增加了一门才干,不仅对个人,而且对自己家乡的经济、科技都有益。……”

很多学员共同反映,在学习过程中,对照实验教材参照讲义,一块一块焊接  $\mu$ p80 微电脑套件;一条线一个焊点地检查;发现问题及时修复, $\mu$ p80 微电脑不知不觉地工作起来时,那种高兴劲儿无法形容。

现在第二期函授学习班正在进行。我们也预祝第二期学员每人都能取得丰硕果实。将微电脑控制技术得心应手地用于本职工作上。在祖国的四化建设中大显身手。

读者询问有关“微电脑控制技术”函授学习与  $\mu$ p80 微电脑套件的事宜请与天津纺织工学院机械系高殿斌同志联系。邮政编码 300160 电话 41.2833 转 279 电报挂号:天津 3251。

本刊“学装微电脑”栏目介绍  $\mu$ p-80 微电脑教育套件及其应用。颇受广大非电专业工程技术人员、微电脑爱好者的欢迎。纷纷来信询问  $\mu$ p-80 套件有关情况与教材。现给予答复:

教材为我社发行的“初级微电脑图解教程”售价:2.9 元/册,邮资 0.40 元。邮购者请与“北京 173 信箱“电子与电脑”编室联系。编码 100036。

$\mu$ p-80 教育套件是“教程”一书配套实验器材。分为三大部份。

电源部件售价 35 元

基础篇部件:21 块,是学习数字电路的理想器材,售价 130 元。

实践篇部件:6 块,组装成 Z80 单板机,使用汇编语言结合基础篇部件可进行简单的应用研究。

应用篇部件:11 块,包括光、声、温度传感器部件和继电器、DC 电机、压电蜂鸣器、扬声器和 7 段数码管等驱动元件,与实践篇组合进行应用控制研究。售价 140 元。

上述套件可选购、可只购印刷电路板、8 线插接件等,方式灵活、服务周到。

邮购事宜请与天津纺织工学院机械系高殿斌同志联系,编码 300160。



## 维修经验谈

# 计算机电源修理

黄培鑫

计算机电源常采用效率高、稳定性好、重量轻、体积小的开关稳压电源,由于计算机不带图纸,常给维修带来极大不便。图1是根据某一种计算机开关电源实物整理出的电路图。图2是开关电源中排风扇的电路图。下面就图1图2的维修作一介绍,供读者参考。

### 第一部分 开关电源的修理

图1是一种隔离式并联型开关电源。其工作原理简单叙述如下:220V交流电经0.5A保险丝BX、保险电阻 $R_1$ 、扼流圈 $T_1$ 送到 $D_1$ 桥整流块的交流输入端, $D_1$ 整流后输出直流经 $L_2$ 、 $L_3$ 扼流圈再经 $C_6$ 、 $C_7$ 滤波,产生约300V的高压。 $R_3$ 对约300V的电压降压后提供给 $Q_2$ 基极正偏, $Q_2$ 开始导通,集电极回路形成电流。随着 $Q_2$ 导通程度的加深, $I_{C2}$ 不断增大。当达到一定值时, $T_2$ ②—①绕组中感应出脉冲电压使 $Q_2$ 基极反偏, $Q_2$ 由导通变成截止。从而使反馈绕组脉冲极性发生颠倒, $Q_2$ 重新获得 $R_3$ 提供的正偏再次开始导通,重复节一过程,形成间歇振荡。在 $Q_2$ 导通期间,次级绕组感应出的脉冲电压,经 $D_6$ ~10半波整流,电感 $L_6$ ~7、电解 $C_{19}$ ~21的滤波后,输出 $\pm 12V$ 、 $\pm 5V$ 四组低压直流电,作为计算机主机、接口电路、及驱动器的工作电源。 $+5V$ 和 $+12V$ 还通过 $R_{22}$ ~26组成的取样电路,取出电压的变化量送到 $IC_1$ 输入端。 $IC_1$ 将输入的变化量进行放大,并产生一脉冲信号,经 $Q_3$ 放大,通过隔离变压器 $T_3$ 推动 $Q_1$ ,来改变 $Q_{1E-C}$ 之间的内阻,进而改变 $Q_2$ 基极负偏压的大小,来改变 $Q_2$ 导通、截止的周期。如负载减轻,次级电压升高, $IC_1$ 输入端取样电压升高, $IC_1$ 输出脉宽变窄, $Q_3$ 集电极电流增大, $Q_1$ 基极正偏时间增长,使 $Q_2$ 导通时间缩短, $T_2$ 中储能减少,次级电压降低。一般开关电源的振荡频率为20KHZ。(可用示波器测量获得)。 $R_9$ ~12、 $D_4$ 组成过流保护电路。一旦 $Q_2$ 集电极电流异常增大时, $V_{e2}$ 也增大; $R_{11}$ 、12分压值也增大,使 $Q_{2E-C}$ 内阻迅速减小,从而迫使 $Q_1$ 退出导通区,保护了开关调整管 $Q_2$ 。 $D_{14}$ 、 $R_{27}$ 、 $CR_1$ 组成次级低压保护电路,一旦 $+5V$ 由于故障升高至5.6V时, $CR_1$ 单向可控硅导通, $+12V$ 对地短路,迫使 $Q_2$ 停振,保护了计算机的安全。

### 故障及修理

如发现彩色显示器不显示(显示器工作正常)和键盘指示灯不亮时,其大多是开关电源发生了故障,应立即关机。将计算机上盖打开,取出电源盒。然后将电源盒盖螺钉拧下,将电源盒盖拆下。先采用直观法和电阻测量法检查,然后运用电压测量法检查。修理仪表为一只500型万用表和一只示波器(没有条件可以省去)。

### 一、直观、电阻测量检查法

电源盖拆下后,首先检查BX(0.5A)保险丝管是否损坏、检查有无异常元器件。如BX损坏应更换同规格熔丝管。从而可以判断故障可能在高频振荡部分。用万用表X1K挡测量交流电源输入两端的正反向电阻应均为350K左右,说明 $D_1$ 之前元件完好,无短路现象。再测量 $C_6$ 正极 $C_7$ 负极两端正向电阻为5K左右,反向电阻为500K左右,说明 $D_1$ 整流、 $C_6$ 、7滤波等元件正常。在检查中如发现异常元件,应拆下更换同规格元件,并要根据元件的位置,找出故障原因。证实高频振荡部分基本正常后,方可通电进行电压测量法的检查。

### 二、电压测量检查法

下面主要介绍三种故障现象的修理方法。

#### 1)无电压

接上电源后,测量次级四组低压全无。用万用表直流500V挡,正极红棒接 $C_6$ 正极,负极黑棒接 $C_7$ 负极,应有300V左右高压,说明 $C_6$ 、7滤波之前电路正常。黑棒位置不变,红棒改接 $Q_2$ 集电极。测 $Q_2$ -c极如无300V左右电压则是 $R_2$ 开路损坏而造成。如有300V左右电压,则测量 $Q_2$ 基极和发射极电压。如均无电压,则是 $R_3$ 损坏。断开200V交流电,用一导线同时接触 $C_6$ 正极和 $C_7$ 负极两端,将高压短路放电,以免修理时被高压击伤。用烙铁拆下 $R_3$ 一端,万用表 $\times 10K$ 挡测量 $R_3$ , $R_3$ 阻值为 $\infty$ ,从而造成 $Q_2$ 无起始偏置,使电路无法工作。更换电阻,电路可恢复工作。当测量基极有0.9V左右电压,而发射极无电压。这种现象一般是 $Q_{2c-E}$ 极开路损坏或 $R_{11}$ 损坏或 $R_{12}$ 、 $D_4$ 同时开路损坏造成的。更换 $D_4$ 如手头无同型号二极管,可选用 $I_0=1A$ , $V_R \geq 700V$ 的高频二极管。 $Q_2$ 可选用 $B_{V_{ce0}} \geq 1500V$ , $P_{CM}=50W$ , $I_{CM}=5A$ , $H_{FE}=5(I_C=2.5A$ 时), $T_{O-3}$ 金属封装管代替。造成 $Q$ 损坏的原因一是 $Q_1$ 损坏,二是 $C_6$ 无容量而失效,使 $Q_2$ 一直导通, $I_C$ 越来越大最后损坏。在更换 $Q_2$ 后通电之前,应对 $Q_1$ 和 $C_6$ 进行检查,以免 $Q_2$ 再次损坏。

在通电后,能听到“嗒嗒……声”,测量 $Q_2$ -E极电压,会发现电压随着“嗒嗒”声而来回摆动。这种现象通常是 $Q_3$ 损坏或是 $Q_3$ 发射极回路元件损坏,如 $L_8$ 导。

#### 2)通电几秒钟后次级无电压

用万用表测量四组低压。接通电流时电压正常,接着电压逐渐上升。 $+5V$ 升高到5.6V时,电压突然跌落。从现象看,是次级低压保护电路已工作所造成电压跌落。保护电路工作很正常,关键是要找出导致 $+5V$ 升高的原因。接通电源瞬间,测量 $I_{C1}$ ①脚电压值正常。



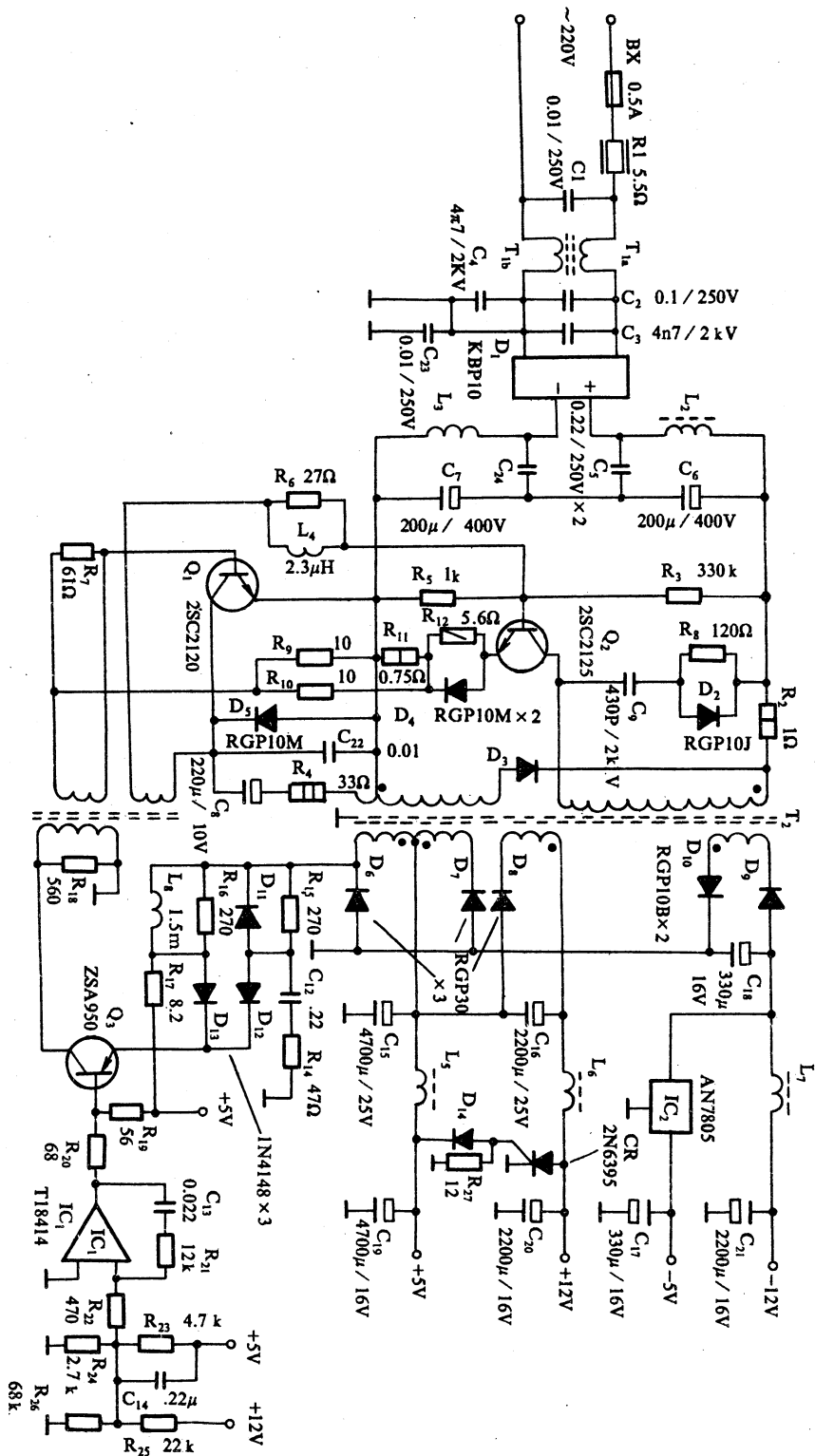


图 1

用同样方法测  $I_{c1}$  ③脚电压,发现电压比正常值 2.9V 升高了 1.3V,变成了 5V。从而造成  $Q_3$  基极反偏, $Q_3$  截止,高频振荡脉冲变宽, $Q_2$  导通时间变长,次级电压上升。当 +5V 升至 5.6V 时,CR 导通,+12V 短路,迫使电路停止工作。分析判断是  $I_{c1}$  有问题。拆下  $I_{c1}$ ,用  $\Omega \times 1K$  挡测量其电流阻值应和表 1 相似。 $I_c$  是一只多功能触发器,它能将输入电压的变化变成输出脉冲宽度的变化,来改变  $Q_2$  的导通时间,起到使电压稳定的作用。在更换新  $I_{c1}$  之前,也应对其进行测量。如买不到,笔者曾试用代替法,使用效果良好。选用两只优质 CS9014 三极管, $\beta$  应  $\geq 250$ , $BV_{ceo} \geq 25V$ , $BV_{ces} \leq 0.4V$ 。将它们接成达林顿接法,见图 3。接入电路后,应对 R26 进行调整,使  $Q_3$  基极电压值为 4V 左右。

在通电后瞬间,测  $I_{c1}$  ③脚电压正常。用同样方法测  $Q_3$ -C 极为 0.006V,测  $Q_1$ -B 极为 0.5V,测  $Q_2$ -E 极为 0.55V。说明电源通电瞬间电路工作正常。再测量 +5V 也正常。在电路工作正常的情况下,造成次极电压跌落只能是保护电路误动作所致。关掉电源,拆下  $D_{14}$  一端,再通电检查开关电源工作是否正常。如正常则是  $D_{14}$  变值或损坏。如仍不正常则是 CR 损坏。用万用表  $\times 1K$  挡测量  $D_{14}$  正反向电阻值,如正确应在 JT-1 图示仪上再次对其测量。笔者多次发现  $D_{14}$  稳压值变小的特殊情况。再换 CR 只要  $IF \geq 3A$  就可以使用。如读者无条件使用 JT-1 仪,可采用简单方法测量  $D_{14}$  的稳压值,按图 4 电路连接,万用表指示值即为被测管的稳压值。对待换管也要进行测量,符合要求后才能上机使用,否则起不到保护作用。 $D_{14}$  选用功耗为 1W, $V_z$  值为 5V 的密封稳压二极管。

### 3) 负压偏高,无正压

接通交流电,测量 +5V、+12V、-5V、-12V 电压分别是 0V、0V、-5V、-12V。根据现象判断是整流二极管  $D_6$ 、 $D_7$  损坏,造成 +12V 也无输出。由于无 +5V,使取样电压下跌, $Q_2$  导通时间增长。由于保护电路接在正电压网络,所以次级中的负电压就升高(-5V 已被  $I_{c2}$  稳压,故不受影响)。图 1 电路中  $D_6$ 、 $D_7$  采用两只高频二极管,读者可用国产型号 RGP-3C 高频整流二极管代替,其  $I_0 = 3A$ , $V_R = 300V$ 。PC 机电源中采用高频全波整流块,也可用 RGP-3C 管代替。如果手头只有小电流二极管,可以用两只或叁只并联使用,效果基本一样。

开关电源修复后不要急于上机使用,应对开关电源接上假负载进行老化试验。老化“中华学习”机电源时,用 2.5 $\Omega$ 、15W 线绕电阻接在 +5V 与地端,用 12 $\Omega$ 、15W 线绕电阻接在 +12V 与地端,-5V 和 -12V 端各接 50 $\Omega$ 、1W 和 120 $\Omega$ 、2W 电阻。接通交流电,用万用表观察次级各组低压值的变化情况,有载时电压下降幅度和空载时相比应不大于 3%,重点是 +5V 和 +12V 两组电压。如四组电压均下降较多,应对 R24 或 R26 重新调整。(R<sub>24</sub> 为粗调、R<sub>26</sub> 为细调)。如 +5V、+12V 两组电压下降较多,应对 R25 进行调整。以上老化时间不应少于 2 小时。老化 PC 机电源时,用 0.83 $\Omega$ 、30W 线绕

电阻接在 +5V 端,用 6 $\Omega$ 、30W 线绕电阻接在 +12V 端,-5V、-12V 用负载电阻规格同上,调整也相同。由于假负载上呈受的功率很大,表面温度很高,通电老化时,不要用手接触负载电阻,以免被烫伤。

## 第二部分 排风扇的修理

开关电源中的排风扇由于长时间工作,很容易损坏,以致使开关调整管、整流二极管元件工作温度升高而损坏。目前,众多的微机电源中,排风扇结构基本相同,其工作过程是不断改变送给定子线圈电流,使定子铁芯产生磁场作用于转子(有许多 NS 极性磁体组成的磁环)的某异性磁场而形成旋转力矩。用来根据转子的磁性来判断送入某组定子线圈电流的任务是由定子供电电路担任。见图 2,其工作原理是这样的。+12V 电源送入电源后,经  $D_1$  对电源进行极性识别后,首先加在开关管  $Q_1$  上。 $Q_1$  基极偏置受  $I_c$  控制。 $I_c$  是一只霍尔磁敏器件。当转子中 S 磁场靠近  $I_c$  时, $I_c$  ③脚输出一高电位,触发  $Q_1$  导通,发射极呈高电位, $Q_3$  工作。 $L_2$  经过 +12V  $\rightarrow L_2 \rightarrow V_{ce2} \rightarrow$  地形成电流回路,使  $L_2$  的两个成对角同向绕组的铁芯上产生 N 磁场,推动与 N 极定子铁芯临近的转子中的 S 工作面,促使转子旋转。而  $L_1$  的两个对角绕组无电流而不发生作用。待转子旋转 90°,转子工作区脱离了定子的磁力,同时  $I_c$  也脱离 S 区。由于转子旋转而来的 N 区,使  $I_c$  关闭,③脚恢复低电位, $Q_1$  截止,集电极呈高电位。 $Q_1$ -c 极高电位经隔离二极管  $D_2$  使  $Q_3$  由截止变成导通。 $L_1$  经 +12V  $\rightarrow L_1 \rightarrow V_{cs3} \rightarrow$  地形成电流回路, $L_1$  绕组中的两铁芯产生 N 磁场,作用于旋转到来的转子 S 极工作区上,形成第二次转矩。接着又重复第一工作过程,如此周而复始。转子越转越快, $L_1$ 、 $L_2$  轮流工作的周期就越短,直至达到  $I_c$  最高开关频率时,转子保持稳定转速。

如计算机在使用中突然发现排风扇不转,应立即停机。拆下开关电源盒上盖,拧下固定排风扇的四只螺钉,取下风扇,并从电路板上焊下风扇电源线。用独立的 12V、关于 200mA 直流源作为临时电源。排风扇常见故障现象有两种。接上 12V 电源,手帮助旋转转子,转子旋转后一种是可贯性旋转,另一种是不可贯性旋转。下面就以上现象的修理介绍如下。

小心地取下风叶,拆下转子轴夹片,取下转子。先直观检查有无异常元件,并加以处理。然后按上 12V 电源。测量  $Q_1$ -c 极电压应为 1.4V, $Q_1$ -e 极应为 0V。因为这时  $Q_1$  处于截止状态,而  $Q_3$  则工作在导通状态。测得该管 B 极为 0.6V,集电极应为 8V 左右。如果  $Q_3$ -c 极电压为 0V,则可能是  $L_1$  开路。关闭电源,用  $\Omega \times 10$  挡测量  $L_1$  绕组直流阻值应为  $\infty$  (正常值为 140 $\Omega$  左右)。说明  $L_1$  确实短路。如无法找出线头,则需重新绕制。在拆下定子之前,应先用尖头烙铁将剩余线头焊下,并做好定子位置标记。然后轻轻来回转动定子并同时向外拉。取下定子后,将  $L_1$  绕组从铁芯上轻轻绕出,并记住绕向和圈数,最后用同号线径的高强度漆包线绕制,有条件的话在线包上涂一层绝缘漆以防潮。在

安装定子之前,还要对  $Q_{2,3}$  进行检查。如果  $L_1$  是因通过电流过大而烧短,即  $Q_3$  早已先击穿损坏了。在更换  $Q_{2,3}$  时,除采用同型号管子外,还可选用  $BV_{ceo} \geq 25V$ ,  $P_{cm} \geq 0.9W$ ,  $I_{cm} \geq 1A$ ,  $\beta \geq 200$ , TO-92 塑封管代替。如 2SD471、2SD773 等。

当出现转子不可贯性旋转的现象,这说明  $Q_3$  已工作,  $L_1$  有电流流过,产生了 N 磁场吸住了转子中的 N 极磁区。遇到这种情况先将直流 12V 降至 10V,防止  $L_1$ 、 $Q_3$  长时间而损坏。取下转子,通电测量  $Q_3$ -c 极应为 6V 左右,再测量  $Q_2$ -c 极电压应等于电源电压,因  $Q_1$  截止,  $Q_2$  也处于截止状态。用一只 1K 电阻和  $R_3$  并联,再次测量  $Q_2$ -c 极电压应为 6V,  $Q_3$ -c 极应为 10V,说明开

关管  $Q_1$  和放大管  $Q_2$  正常。去掉并联电阻,取一小磁铁将 S 极靠近  $I_c$ ,测量  $Q_{2,3}$ -c 极电压和刚才相同,说明  $I_c$  工作正常。如果  $I_c$  损坏最好选用同型号器件,如果买不到可选用其它型号,如 UGN3019、UGN3020 等。判断  $I_c$  的好坏可参考表 2 数据,用  $\times 1K$  挡测量。在安装 IC 时可将原来卧式装置改成立式装置,这样可以增大磁敏作用面,提高霍尔器件的开关效果,见图 5。另外还应注意  $I_c$  的平面应和定子铁芯面垂直,防止  $I_c$  被转入磁铁磨擦而损坏。最后装上转子进行通电试验。要求转子通电后应能迅速起转,转速均匀平稳。正常后的排风扇就可以投入使用。

表 1

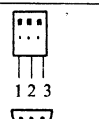
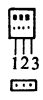
序号 表棒 阻值	序号			 ①—1N 端 ②—COM ③—OUT 端
	①—②	①—③	②—③	
红棒-黑棒	35 k	$\infty$	$\infty$	
黑棒-红棒	27 k	7.6 k	5 k	

表 2

序号 表棒 阻值	序号			
	①—②	①—③	②—③	
红棒-黑棒	6.2 k	$\infty$	$\infty$	
黑棒-红棒	13 k	50 k	6.2 k	

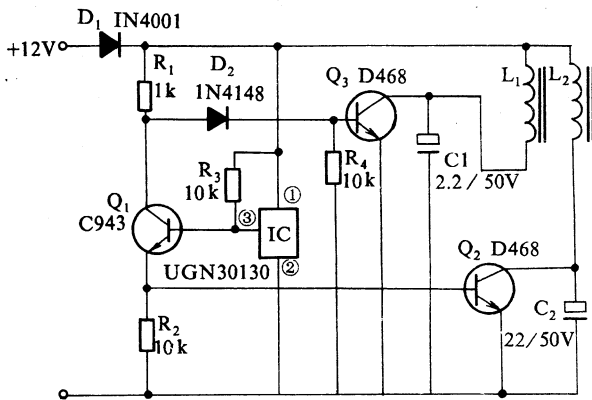


图 2

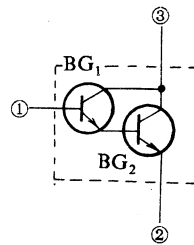


图 3

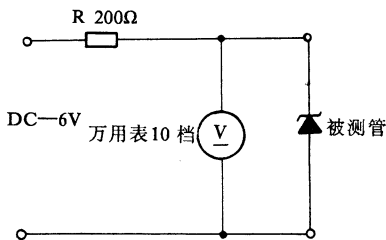


图 4

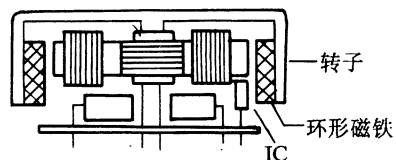


图 5

# 四通 MS-2401 打字机使用维修经验

江苏盐城师专 蔡长安

四通 MS-2401 中外文高级打字机是集显示、键盘、打印、主机于一体的专用于文字处理的计算机系统,它有体积小、重量轻、结构紧凑、打印的字体、字号多变等特点,进入了各个办公室,以取代传统的文印设备。但如何使用维护好这类设备,除了按使用说明书上的规定要求进行操作外,我们还总结出了下面几点,以便使用者参考。

(1)蜡纸的打印方法。蜡纸的正确打印方法是:把档次调节杆调整到合适的位置,取下色带,将蜡纸下面的校对棉纸拿出放在蜡纸的上面。用这种方法打印蜡纸的好处是:既防止了蜡纸上的油墨等污物被打印针带入打印头中,阻塞打印针出口孔,使出针和回针阻力增大,打印针被色带拉弯或折断;又可防止了由于色带的原因(由于色带上有孔洞等)而刮弯打印针。

(2)打印色带的更换。打字机经过一段时间后,需要更换色带。色带实际上是一个墨比乌斯带,由于色带采用了这种形式,从而使色带的实际使用长度增加了一倍,提高了利用率。根据扭转方向的不同,墨比乌斯带分为左旋和右旋,左旋和右旋墨比乌斯带是不能互换的,这就由色带盒的结构所决定,四通 MS-2401 电脑打字机必须使用右旋的色带。更换色带时,只要打开盒盖,取出旧带装进右旋新色带即可。由于市售色带无标记,购买时必须认准旋转的方向。如果买回的色带是左旋的,可以用剪刀将色带剪开个斜口。把色带回转为右旋后,再把剪开处搭接,用结实的细线缝好即可。

(3)字形上下出现扭曲的处理。四通 MS-2401 打字机使用一段时间后,会出现上下扭曲。这是由于打印

头经常向左边回转而碰撞造成的。处理这个现象的方法是:按住键盘上的“<控制>+<面>”键,出现“请输入误差修正值(-4-3)”,可输入误差修正值“-2”,即可得到修正。

(4)“划痕”的处理。有时我们在打印蜡纸时,会在蜡纸上出现一条一条的“划痕”。这可能是档次调节杆调节的位置不适当造成的,这时我们只要把档次调节杆调到合适的位置即可消除“划痕”。如果调节档次调节杆后,还出现“划痕”,那么这可能是打印头的一根或几根打印针坏了,这时我们就要拆开打印头上的四颗小螺钉,更换新的打印针了。

(5)个别按键失灵的处理。由于打字时经常敲击键盘。会造成部分按键敲入时,屏幕上无显示的故障。现将修理方法简单介绍如下:首先确定是否是键盘故障。这可断开电源,旋下外壳上的 4 颗螺钉,取下外壳上盖,小心取下各接线卡,露出键盘。用万用表进行检测。按动键盘,测两焊点间的电阻,应为零值。如不为零值,说明两焊点间存在接触不良现象。

其次,取下键帽。方法是:去掉键体两脚上的焊锡。用烙铁进行加温,再用气球吹去焊锡。然后,在键盘表面上用两把中等型号的改刀放在键体两旁,适当用力向上挑,就能取下键体。拆开键体后,会看到两块铜片,一片像“π”,一片像“山”型。可把“山”型铜片的两边向外拨动,目的使两铜片接触良好。然后把两铜片装入到内部的两个小槽中,再用万用表测试是否已修理好。

对于上述方法不起作用的键,还可采用不常用的键体进行对换。

(接 41 页)

表 1

74LS123 功能表

CLR	A	B	Q	Q
L	X	X	L	H
X	H	X	L	H
X	X	L	L	H
H	L	↑	□	□
H	↓	H	□	□
↑	L	H	□	□

脚号	3,11	1,9	2,10	5,13	4,12
----	------	-----	------	------	------

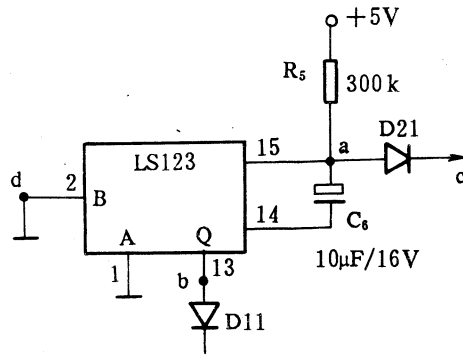


图 2

脉宽缩短到

$$tw = 0.45 \cdot R_s \cdot C_s = 0.45 \times 300k\Omega \times 10\mu F = 1.35S$$

经上述处理后,开机使用表明:前述的两个毛病均已治愈。读写操作也都正确无误。这样修理工作到此结束。修改后的局部电路图如图 2 所示。74LS123 的真值表如表 1 所示。

结论:

此毛病是因电路设计不周而引起的,并非使用不当,应引起生产厂的注意。

# 修理 CEC-I 型中华学习机软盘驱动器一例

赵升

有 CEC-I 型中华学习机一套，其软盘驱动器有如下二个毛病：

1. 将软磁盘插入或取出时，驱动器中的驱动电机起动，运行约 30 秒钟后才停止。软驱上的 LED 灯不亮。

2. 对磁盘进行读写操作后（即软驱上的 LED 灯灭后）驱动电机仍要运行约 30 秒钟才停止。不影响对磁盘读写的正确性。

虽说以上两点只是“毛病”而非故障，也不影响对驱动器的使用，但毕竟对用户来说是件很不舒服的事。正常情况应该是：插入、取出磁盘时驱动器电机不起动，对磁盘进行读写操作后（即软驱上 LED 灯灭后）约 1~2 秒，驱动电机即应停机。

分析：

1. 根据毛病 1 判断：电机的起动一定与驱动器中的防误写红外光电开关信号有关。

2. 根据毛病 2 及毛病 1 判断：驱动器内的驱动电

机控制电路中有时间电路，并且这部分电路可能有问题。

检查：

此驱动器是杭州磁记录设备厂生产的 HS-550 型。因为无图纸，所以测绘其电路是十分必要的。但其控制电路较复杂，全面测绘即费时，也没有必要。基于以上分析，测绘电路时可以从以下三个方面入手进行。

a) 驱动器上的防误写红灯开关。

b) 从 CEC-I 主机来的磁驱控制信号。经分析应该是 J4-14 脚上的磁驱允许工作信号 ENBLI。

c) 通向驱动电机的控制信号（驱动器内的 PIN1 插头）。

将与这三个方面都密切相关的那一部分电路先测绘出来，然后再测绘与此部分电路相关联的其它部分电路。直到能看清电路原理就可以了。有了这样一份局部电路图就可以对毛病进行定量分析了。测绘出的局部电路如图 1 所示。

这部分电路的核心是一可重触发单稳态电路，由  $R_5$ 、 $C_6$ 、 $\frac{1}{2}$  片 74LS123 构成。查 74LS123 真值表可知：当 2# 脚上有一个个跳沿出现时，在 Q 端（13# 脚）输出一个正向单稳脉冲，其脉宽为：

$$t_w = 0.45 \cdot R_5 \cdot C_6 \\ = 0.45 \times 300k\Omega \times 220\mu F = 29.7S$$

与实际情况相符（见毛病 1）。

从主机来的允许工作信号 ENBLI 是一低电平有效信号。异或门的接法为同相驱动方式。故当 ENBLI 信号为低时， $C_6$  通过  $D_{21}$ 、74LS86 回路放电，使 a 点电位为“0”，Q 端输出“1”电平，驱动电机启动运行。当完成读写操作后，ENBLI 信号则上跳为“1”。此时电源经  $R_5$  向  $C_6$  充电，a 点电压逐步升高。经过约 30S 后，Q 端输出“0”电平，驱动电机停转（见毛病 2）。

根据以上分析，对电路中的 a、b、c、d 各点电压进行了测试，测试结果证实了以上分析是正确的。这样解决问题的办法也就有了。

修理：

1. 在 d 点将印刷线断开，认切断红外开关与单稳电路的联系。并将 LS123 的 2# 脚接地，以防有干扰脉冲将其误触发。

2. 电容  $C_6$  用  $10\mu F/16V$  替换原  $220\mu F/10V$  规格，使单稳（下转 40 页）

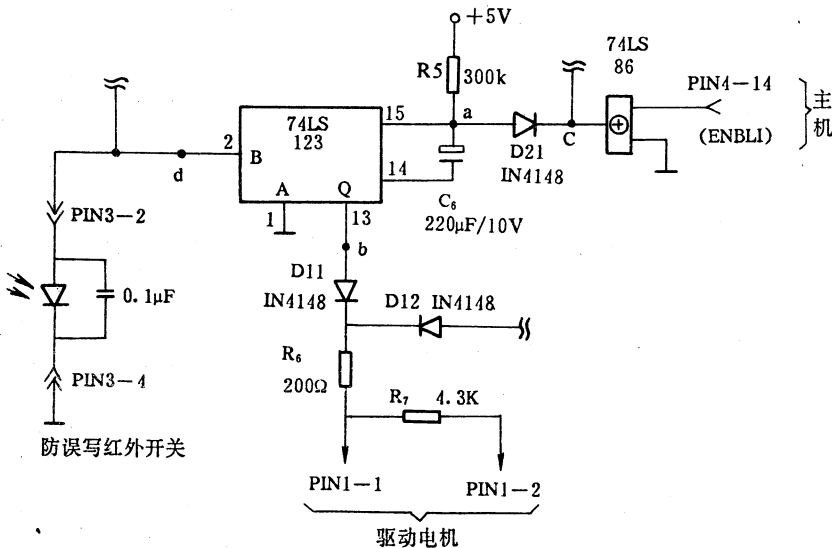


图 1

# 谈谈尼龙打印色带的使用技巧

福建省建阳一中 林斌

合式环形尼龙色带,由于具有不易折断,打印字符清晰,而且可以反复使用,比较经济等特点,所以深受广大用户的欢迎。下面介绍一些尼龙打印色带的使用技巧,供参考。

1. 为什么从市面上买来的尼龙打印色带,有的刚打印时颜色就很淡,而且字符不清楚?

答:这是因为各个厂家生产的尼龙打印色带,尽管出厂前已用油墨浸渍过了,但有的由于存放时间太久,油墨已干了,有的只是用很稀的油墨浸渍过,所以它只能直接放在装有墨盆的打印机上使用,如放在没有墨盆的打印机上使用时,则会出现上述现象。遇到这种情况时,一般只须将买来的尼龙打印色带先上好色带油后,再放入打印机中使用就行了。

2. 尼龙打印色带必须使用哪一种油墨?

答:必须是打印机色带专用的油墨,也就是通常有墨盆装置的打印机所采用的油墨,市面上称为高级再生打印色带油。这种油墨在一般计算机商店均有出售,只是价格较贵,但决不能用普通油印机中所用的油墨来代替。

3. 怎样给尼龙打印色带上油?

答:先在桌面上放一块塑料薄膜,并将色带盒内的色带全部拉出,然后将色带一段一段地铺平在薄膜上上油。上油时,可先用毛笔蘸少许色带油,然后在包带与打印头相背的那一面上的中间部分均匀地抹上一层薄薄的色带油,中间上油宽度约为带基的三分之一即可。但千万要注意,色带油不能抹得太多,而且色带朝向打印头的那一面不要上油,即使上油的那一面,也只能在中间部分,上下两边也不要上油。整条色带上油完毕后,最好不要立即卷入色带盒内,可以先露在外面让油墨凉干二、三天后再卷入色带盒内,这样以后就很好用了。

4. 尼龙打印色带使用时,打印字符油墨太浓,字迹模糊不清,且打印纸上有许多横条的“干扰线”,怎么办?

答:这种情况多是属于色带上油太多的缘故,必须采取以下几项措施:

(1) 将色带全部拉出,用毛笔在色带上过油的部分再重新刷过一遍,将过多的油墨弄去;

(2) 将上述处理过的色带先露在外面凉干几天后再卷入色带盒内使用;

(3) 用棉花蘸上酒精将打印头前端部及色带护罩上粘有的油墨全部擦拭干净。因为色带上油太多,使用时必然会造成打印头前端部及色带护罩上粘有油墨,而影响了打印效果;

(4) 检查一下打印头调整手柄的位置,看看打印头与打印辊之间的间隙是否太小,若间隙太小,也容易产

生上述现象,特别是使用刚上过油的色带。这时可将打印头调整杆后移,将间隙调大即可。

5. 尼龙打印色带使用时产生“停带”现象,是什么原因? 怎么解决?

答:这里所讲的“停带”,是指色带盒内的驱动轮照常转动,而色带却停滞不前,这样由于色带的打印位置不能变换,所打的字符就会越变越淡,甚至还会损坏色带。产生这种现象的原因可能有以下几种:

(1) 色带盒内的驱动轮与被动轮上粘有较多的油墨。由于色带是夹在驱动轮和被动轮之间,靠摩擦传动而前进的,所以如果它们粘有较多的油墨时,就会使摩擦力减少,甚至产生打滑,而带不动色带。对此只好拆开色带盒的盖子,将两轮上的油墨全部擦拭干净。从这里进一步启示我们,在给色带上油时,驱动轮及被动轮与色带的接触部位(即色带的上下两边部分)千万不能上油,否则既浪费了色带油,又容易使驱动轮及被动轮粘上了油,而不能带动色带前进。

(2) 色带盒内被动轮上的压紧弹簧脱落或漏装。由于该弹簧能使被动轮、色带、驱动轮紧密地接触在一起,以增大它们之间的摩擦力,而带动色带前进,所以如果该弹簧脱落或者漏装上去,就会造成摩擦力不足,而无法带动色带前进,这时应拆出色带盒的盖子,检查一下该弹簧是否装上或装好,弹力够不够等。

(3) 色带在色带盒的出带端被挂住或出带阻力太大。遇到这种情况,可先用手或镊子将色带拉拉看有无被挂住或阻力太大等情况,通常多是尼龙色带接头处连接不好或色带某处有破损或出带端支架有裂缝等现象都会造成色带被挂住而不能前进,此外色带出带路线安装不妥或出带压紧簧片没有装好也极易使色带绞合而拉不出。对此应仔细检查,并采取相应措施排除,甚至要更换新的色带。

(4) 色带卷入色带盒内后,有时由于某种原因(如色带安装不好、久置不用等),会使色带在带盒内的前后排列顺序紊乱,甚至有的与驱动轮或被动轮缠绕在一起,而造成色带不能前进,这种情况尽管很少发生,但也不能忽视,解决方法就是将色带全部拉出,再重新卷过一遍即可。

6. 尼龙打印色带使用时,产生“拖带”现象,是什么原因? 如何解决?

答:这里所讲的“拖带”,是指色带在使用中,不能始终保持张紧状态,而被打印头拖出,又来不及卷入色带盒内,致使色带脱落下垂,这样轻者使打印机打不出字符来,重者会使色带与打印机架里的齿轮带或齿轮绞合在一起,使打印机不能正常工作,甚至会损坏打印机。产生这种故障的原因可能是:

(1) 尼龙打印色带质量不好,如色带接头处连接不

圆滑,或色带使用久后某处有产生破损等,这样色带极易被打印头挂住,而产生“拖带”,对此只好更换一条新的色带。

(2)打印头及色带护罩在移动过程中与色带之间的滑动摩擦阻力太大。当色带受到的该阻力大于拉出端所受的阻力时,也会产生这种现象。对此可以从两方面来解决:一方面要尽量减少打印头及色带护罩与色带之间的摩擦力,即看看打印头与色带护罩上有没有毛刺,它们之间的间隙是否太小;另一方面应想法增大色带拉出时阻力,即看看色带在出带端是否有被弹簧片压好,弹簧片的弹力足不足等。

7. 尼龙打印色带使用时经常碰到这种情况(特别是冬季);在刚打印头几张时,字符不清楚,甚至还会发生“漏点”,只有当工作了一段时间后,字符才逐渐变为正常,这是什么原因?

答:这种现象开头往往被人误为色带上的油已用光,其实不然,它的原因多数是属于打印头上粘有较多油墨的缘故。

我们知道,上油的尼龙打印色带极易使打印头粘有油墨,而且使用久了,油墨还会慢慢渗入到打印针之间去,特别是充塞了打印针的中孔部分。这样在刚开机打印时,由于打印头的温度还较低(特别是冬季),打印针之间被油污粘住了,致使打印针运动阻力增加,打印幅度减小,所打印的字符就不清楚。但工作一段时间,打印头的温度逐渐升高,油污溶解,阻力减小,打印针又可运动自如了,打出的字符就恢复正常清晰了。

为此,使用尼龙打印色带的打印机,要定期清洗打印头,方法如下:将打印头从托架上拆下(注意电缆线不要拆),先将打印头前端部的油污用酒精擦拭干净,

再将打印针面的圆形导杆拆下,露出打印针的上半部分,然后用镊子夹一小团蘸有酒精或清洁剂的棉花对打印针进行清洗。清洗时应着重在针尖附近,特别是针尖中孔部分,并可将针尖左右分开以便逐根擦拭,但不要上下扳动针,以免针尖弯曲变形,也不要将打印针上下位置顺序弄错,同时要避免清洗后的油污流到打印头里面去,最好将打印头前端部朝下清洗,或事先用棉花塞在打印针下端以汲取清洗后的油污,洗完再取出。

8. 尼龙打印色带怎样应急使用?

答:尼龙打印色带使用一段时间后,打印出的字符就会逐渐变淡,这时应及时给色带上油才可继续使用,但给色带上油要花费一定时间,而且最好还要凉干几天后才好用,这样如果遇到要急需马上打印某些内容时,怎么办?下面我们介绍一种应急办法:即在色带盒底部前面两端的定位销与打印机架上的定位槽之间垫上2~3毫米厚的物体(例如硬纸片等),将色带盒的前端部抬高,实则是为了改变色带的打印位置,这样打印时,由于色带的打印位置全部往下移一些,就会继续打出清晰的字符来。不过采取这种办法,不能将色带盒的前端部抬得过高,否则容易发生色带脱落现象。

为了方便使用,我们建议有用尼龙色带的打印机,最好都配备二条色带(包括色带盒),以便交替上油,轮流使用,这样就不会出现上述问题了。

9. 尼龙打印色带使用到什么程度,需要更新?

答:尽管是可以反复上油使用,但这并不意味着它可以一直使用到坏为止。在一般情况下,尼龙打印色带在使用适当程度时,通常是色带表面开始起毛了,就不宜再继续使用了,这样既可减少打印针的磨损,又可避免发生刮断打印针现象,还能提高打印的质量和效率。

## 1991年度“AST”杯第二届全国软件人员竞赛在京揭晓

为了提高我国软件人员的素质,鼓励广大青年计算机软件人员为振兴中华而勤奋学习,以利于发现人才、培养人才,由中国软件行业协会、北京市科学技术协会、电子工业出版社联合举办,并由AST虹志(电脑)有限公司赞助的“1991年度《AST杯》全国计算机软件人员竞赛”已于11月20日在京揭晓,并在清华大学隆重举行了授奖大会。国内许多知名的计算机专家、教授出席了授奖大会。胡昆山(中国软件行业协会副秘书长)、吴文虎(清华大学计算机系教授,初级程序员级评奖委员会主任委员)、方裕(北京大学计算机系教授,程序员级评奖委员会主任委员)、张大力(北京市科学技术协会副主席)、胡貽志(中国软件行业协会副会长,物资部信息中心主任)、吴金生(电子工业出版社副社长)、郑人杰(清华大学软件开发中心副主任)、徐国平(中国软件行业协会高档微机协会副理事长、秘书长、中国软件行业协会考试指导中心副主任)、贾月珍(北京电脑天地学校副校长)。最后由胡貽志、张大力、吴金生及AST虹志(电脑)公司北京首席代表邓平贵等先生分别将奖品授予优胜选手。

### 程序员级获奖名单

等级	姓名	性别	年龄	学历	单位
一等奖	马力	男	18	大学	吉林大学少年班学生
二等奖	郭东伟	男	19	大学	吉林大学学生
	王英超	男	19	大学	鞍山钢铁学院学生
三等奖	徐建东	男	20	大学	大连理工大学学生
	李志宏	男	23	大学	南昌市工商银行人民广场储蓄所
	夏伟文	男	23	大学	北京税务局崇文区分局
	王革	男	22	大学	大连船用柴油机厂
	陈晖	女	23	大学	北京市丰台区区长辛店二七机车厂

### 初级程序员级获奖名单

等级	姓名	性别	年龄	学历	单位
一等奖	陈兆前	男	20	大学	杭州浙江大学学生
二等奖	潘杰	男	17	高中	北京一六六中学生
	李万冬	男	19	大学	天津大学学生
三等奖	吴志军	男	23	大专	大连纺织品批发公司
	楼激扬	男	25	大学	浙江广播电视学校教师
	冯爱民	男	19	中专	南京工业学校教师
	李中政	男	19	中专	青岛碱厂
	陶坚	男	21	高中	浙江省嘉兴市第一中学

# 初级程序员级水平考试辅导讲座

## 计算机病毒防范问答

常晓冀

从我国目前情况看,计算机病毒的防范已进入综合治理阶段。国内流行的七种常见病毒已经得到基本控制;但是,新的计算机病毒的数目仍在增加。今后随着我国计算应用技术的不断提高及工作站、计算机网络的普及,对计算机病毒的防范工作将变得更为迫切。

问题1 何谓计算机病毒?

答:作为一种能够通过某种途径潜伏在计算机存储介质,如磁盘、磁带、光盘、网络中,条件成熟时,即被激活的具有对计算机资源进行破坏作用的“程序”或“指令集合”,称为计算机病毒。

问题2 计算机病毒的实质及共性是什么?

答:计算机病毒是通过正常的操作运行的非授权入侵。所谓“非授权入侵”指的是病毒非法入侵系统,对其进行破坏的行径。

病毒的共性是:

- (1)程序设计精巧;
- (2)有较强的再生机制;
- (3)潜伏深,难以发现;
- (4)可激发性;
- (5)危害性大。

问题3 计算机病毒模块划分与工作流程情况如何?

答:病毒入侵计算机系统及在系统内传染循环过程如图一所示:

- (2)计算机病毒模块划分如(图二)所示;
- (3)病毒工作流程(以小球病毒为例)如图三所示

问题4 计算机病毒的分类原则是什么?

答:(1)根据病毒入侵途径大体可分为以下四种

\*. 源码型病毒:病毒在源程序被编译之前插入到诸如 C、Pascal、COBOL、FORTRAN 等高级语言所编写

的源程序中。该类病毒难以编制,较少见。

\*. 入侵型病毒:病毒侵入主程序中,不破坏主程序,难以清除病毒。

\*. 操作系统型病毒:嵌入或部分取代操作系统工作。病毒通常在操作系统启动时(开机时)自动装入。具有极强的传染力和破坏力。如常见的小球和大麻即为该类病毒。(包括它们的变种)

\*. 外壳型病毒:病毒将自身置于主程序的外围(或头或尾),不修改原程序主体,以感染.COM或.EXE文件为多见。如常见的以色列(黑色星期五)、1575病毒及其变种均属这类病毒。

(2)根据病毒的表现形式,可分为良性和恶性二种。后者是破坏系统文件、数据文件和造成死机的罪魁祸首。

(3)根据病毒攻击的对象可分为三类:

\*. 攻击微型机(我国目前以这种病毒所占比例为最大)

\*. 攻击小型机

\*. 攻击工作站、网络

其中以第三种病毒所造成的危害最大,它的存在,构成对重要信息系统和金融信息系统的潜在危险!是今后计算机病毒防范的重点。

问题5 如何理解计算机病毒的复杂性?

答:病毒的复杂性表现在以下几方面

(1)病毒种类多,国外发现的病毒已达上百种,国

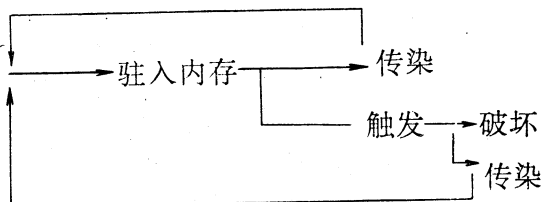


图1

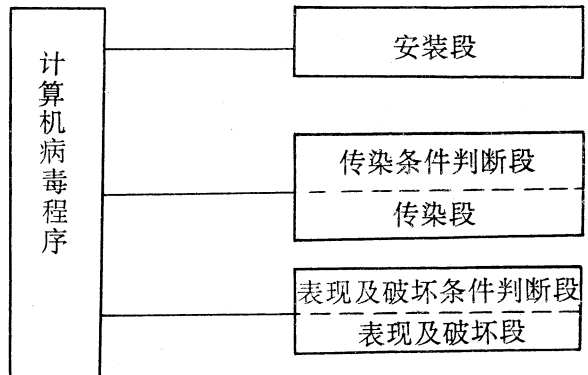


图2



内也有数十种之多。

(2)病毒变种多,仅小球病毒变种就有十多种。

(3)来源复杂,国外传入,如小球、大麻、以色列、杨基、1701/1704、1575 及其变种等。国产病毒数目也在增多,如“中国炸弹”、“6.4”等。

(4)病毒正在升级;正在由单一型向智能型方向演变。如“4096”病毒即是智能型病毒,该病毒已突破通用解毒软件的防护系统,通过篡改 DOS 系统内部数据,对 DOS 进行大量修改,而不改变其中断向量。主要感染.COM 或.EXE 文件,有些病毒甚至带上了某些政治色彩。

问题 6 计算机病毒将对计算机系统造成什么样的危害?

答:计算机病毒所造成的危害按其侵入系统的次数和它的病毒性质可分为以下几方面:

(1)单一型病毒的危害

\*.一次侵入,以大麻为例,多次入侵系统导致计算机系统引导程序被病毒程序完全复盖造成死机,无法启动。

(2)两种以上病毒入侵计算机系统所造成的交叉感染,所导致程序和数据文件遭破坏及引导系统故障的严重程度将大大超过单一型病毒所造成的危害。以“6.4”与“大麻”病毒同时入侵计算机系统为例来说明。若以一般解毒软件去解会出现硬盘无法启动的严重后果。

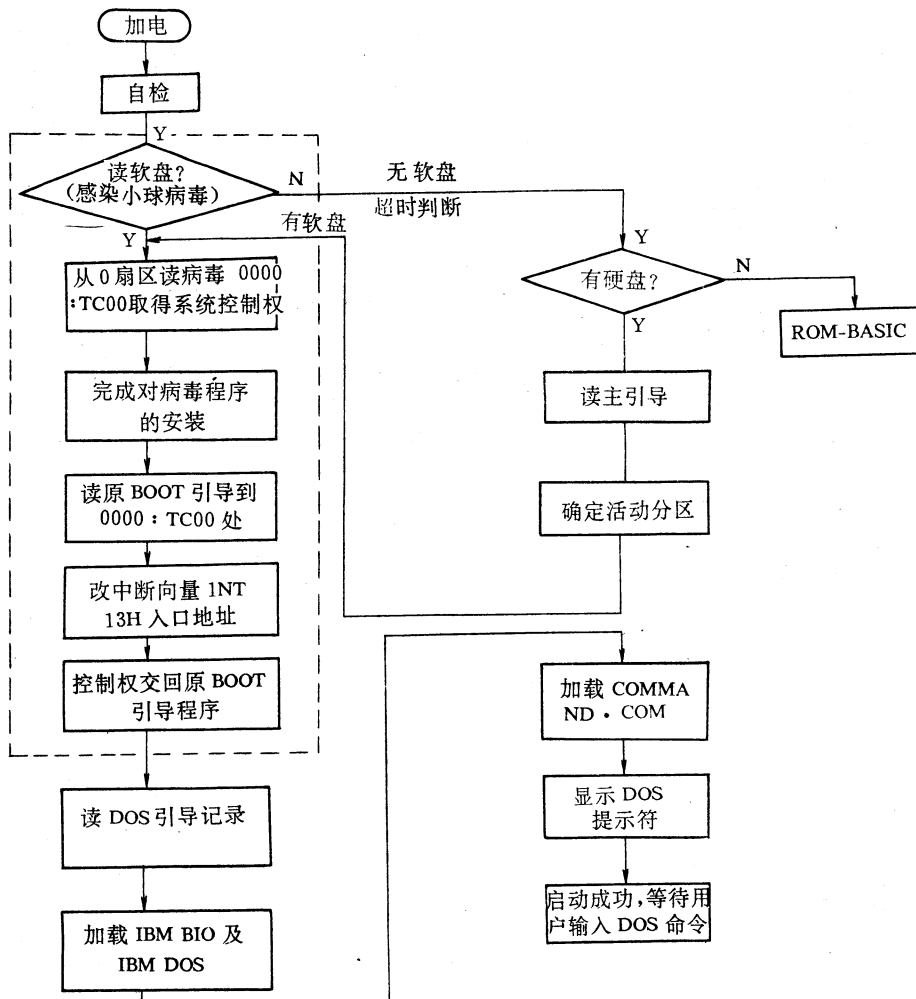
(3)智能型病毒的危害则难以预料。

问题 7 如何实现对计算机病毒的检测?

答:检测计算机病毒主要通过以下工具和方法

(1)DEBUG 调试程序,提供一个检测手段使用户能监督和控制欲调试的程序执行过程。

(2)PCTOOLS 实用软件,完成对系统的检测及对系统数据的修改和恢复工作。根据用户需要可采用不



说明:图中虚线框内为小球病毒程序部分。

图 3

同版本。目前, PCTOOLS 最高版本为 6.0V(英文版)

(3)重要系统信息的掌握

- \*. DOS 系统磁盘参数表
- \*. DOS 系统文件分配表
- \*. DOS 系统文件目录表
- \*. DOS 系统中断向量表
- \*. DOS 所使用的寄存器

(4)病毒检测软件

问题 8 如何对常见病毒进行处理?

答:对常见病毒可使用以下简易方法加以清除。

(1)清除小球病毒:系统复盖法,将未感染病毒的系统盘插入 A 驱动器,冷、热启动均可(该系统版本不得低于原硬盘操作系统版本)。

(2)清除大麻病毒:软盘清除法

```
A>DEBUG↵
-L 100 1 0b 1↵
-D 100 2FF↵
-W 100 101↵
```

免疫法:将未感染病毒的非系统盘引导区的前 4 个字节置成 EA,05,00,C0。

(3)以色列(黑色星期五)病毒症状与免疫

\*. 症状

(a). COM 或. EXE 文件字节长度增加 1.8K 字节。

(b)每运行一次感染病毒的文件,硬盘空间减少 1.8K,直至死机。

\*. 免疫方法:将病毒开始的第 5 个字节处标志“MSDOS”存入文件。具体作法是最后把 CX 的值为

文件长度加 10,再存盘。

(4)1575 病毒症状与免疫:

\*. 症状:

(a). COM 或. EXE 文件长度增加 1575~1591B 病毒文件日期为感染日期。

(b)每 DIR 后,磁盘空间减少 1575~1591B 字节。

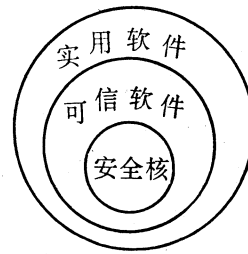
(c)屏幕出现一条“蜈蚣状虫子”。

\*. 免疫:在文件末尾加上 0A0C,并把 COM 或 EXE 文件属性改为只读。

问题 9 目前我国使用哪些办法对计算机病毒进行防范?

答:对计算机病毒的防范,可采用软件,硬件防范及对开放的 DOS 操作系统进行修改等方法。

1. 软件防范



安全操作系统图示

注:图中“安全核”是由计算机硬件设备和软件构成,它的功能是完成对数据的存取控制。

图 4

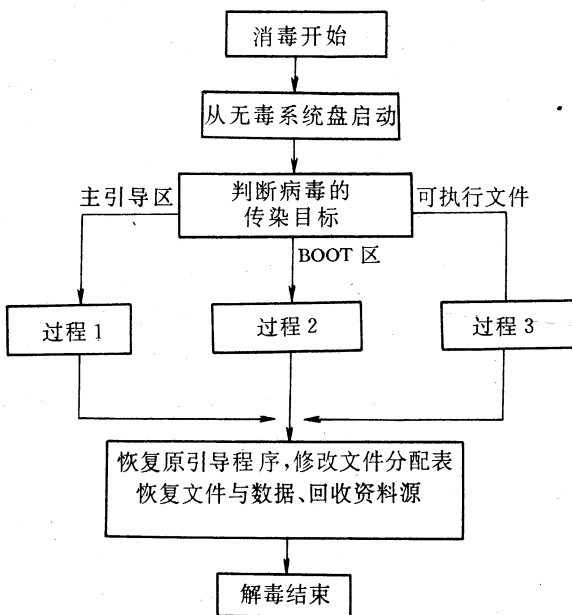


图 5(1)

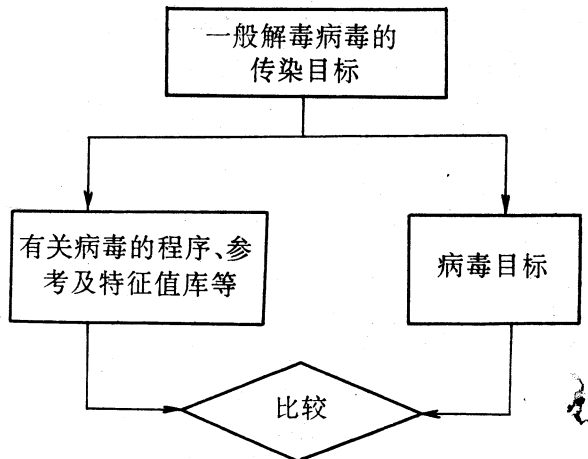


图 5(2)

(1)为用户设置数据保护区[ADM 软件]

(2)对常见病毒的免疫预防

(3)研制主动防御病毒程序,使其常驻内存密切注意系统内引导扇区及中断向量的变化,如有变化,立即报警,并中断其执行。

(4)解毒软件

\*.一般解毒软件:

采用传统编程法编制的解毒软件,称为一般解毒软件,只能对付单一型病毒。

\*.智能型解毒软件:

采用人工智能算法和知识库编制而成,可用以对付病毒变种和多种病毒交叉感染在同一计算机系统的复杂情况,是有发展前途的人工智能型软件。

2.硬件防护:

(1)病毒硬件防护卡,可直接插入系统扩充槽内。用以监督系统中中断向量等重要参数的变化。优点是速度快,缺点是造价高。因此,硬防护卡与解毒软件联合使用效果可能好些。

(2)硬盘保护电路,是对高档微机的硬盘,采用配置信息变换技术加以限制。开机时,配置为无硬盘驱动器,从而实现隐藏并保护硬盘的功能。

3.修改 DOS 操作系统等措施

(1)设计安全操作系统(如图 4 所示)

(2)固化 DOS"病毒防护卡

以固化 DOS 操作系统为基础,将引导扇区 DOS 系统文件写入芯片,用以避免引导区和外壳型病毒的入侵。

问题 10 一般解毒软件的工作原理是什么?

答:我们以图五为例,说明解毒软件的工作原理。

(图五)说明:图中“判断病毒传染目标”,这一步骤是以该软件所建立的病毒特征值库等内容为依据完成对病毒目标的判别。

与病毒有关的程序文档、各类库内容如下:

(1)正常程序与 BOOT 区内容档案

(2)病毒特征值库

(3)内存中断向量表

(4)内存容量库

实践证明,病毒特征值库的建立,无论对病毒的检测还是消除都是必不可少的。

问题 11 如何正确使用解毒软件?

答:解毒软件的正确使用与否,直接关系到用户的切身利益,切不可等闲视之。以下从几方面来说明。

(1)解毒软件(尤其是一般解毒软件)是针对某些病毒(单一型病毒,并不包括其变种)。没有万能的解毒软件,如同没有万能的包治百病的药品一样。

(2)解毒软件一般不主张拷入硬盘运行,因它本身不具有对病毒的免疫能力。在装有 DOS 操作系统的硬

盘环境下,只能助长病毒的进一步扩散。

(3)最好不要使用不同解毒软件,去解同一系统内的病毒。其后果可能很糟。

\*.其一,可能发生用户文件被删除,硬盘启动故障等严重问题。原因是不同的解毒软件,解毒原理不同,所使用的汇编语言、高级语言及算法均有差异。

\*.其二,出现上述问题后,难以查明是哪个解毒软件的错误及错误所在?

(4)学会正确使用解毒软件的方法,严格按操作说明进行解毒。解毒后,一定要重新启动(热)机器。

问题 12 如何预防计算机病毒?

答:最根本的预防病毒措施,是加强对设备(微机、外设)的管理工作。作好对系统的后备和用户文件后备。养成良好的设备文档记录习惯尤其是对设备故障(包括病毒、硬件、软件等故障)的处理记录;此外,还要注意以下几点:

1. 尽量避免往本系统内装入外来的、未经严格检测的应用软件。电子游戏盘是传染病毒的最有效的媒介之一。目前的检测软件,其检测范围有限,难以保证外来软件所携带的病毒不侵入本系统。

(2)对本系统中重要文件进行简单加密,来实现对文件和系统的保护作用。

(3)外单位人员上机,也是病毒入侵途径之一。

(4)机器故障,送出维修,有可能带回病毒。

(5)新购机器是传播病毒的又一途径。

(6)向外出租或外借机器,难以保证不传染上病毒。

为维护微机系统的安全运行,现提出以下建议:

1. 微机系统中硬件、软件、外设故障与病毒感染交织情况发生,应尽快使用系统自身的检测软件、解毒软件分层次加以排除,避免发生因误判对系统所造成的不必要损失。

例如,硬盘系统不能正常启动。此时可采用以下步骤逐层进行排除:

(1)用系统检测软件对硬盘进行检测,用以判断硬盘故障。

(2)用解毒软件检测系统中存在的病毒。

(3)以上二个因素均排除后,再用其他软件检测工具(或其他工具)排除系统软件中存在的软故障,以求尽快恢复硬盘的正常启动功能。

2. 病毒感染后的善后恢复工作很重要。有些系统反复被某种病毒感染。其结果造成文件分配表紊乱,严重影响系统正常工作。

(3)配备一名技术较强的系统管理员,负责微机系统的日常维护,对安全运行十分重要。

# SCB-1 单片单板机简介

随着科学技术和生产的发展,单片机技术已广泛应用于工业、农业、国防、科研、教学及日常生活各个领域。为进一步推广应用单片机技术,逐步取代日渐落后的以 Z80 作 CPU 的 TP801 单板机市场,我所 1989 年在国内首先研制成功一种性能优异的 SCB-1 型单片单板机,随即在《电子与电脑》杂志上发表,该机硬件原理及软件监控程序清单全部刊出。产品批量投放市场后,在全国引起广泛影响,曾几度供不应求。

SCB-1 何以有如此旺盛的生命力呢?

首先,SCB-1 的硬件配置全面、典型、规范。它以 8031 为 CPU,系统时钟最高可达 12MHz,扩有 8K 外部 EPROM,带掉电保护的 16K RAM,6 位 LED 数据显示,5×5 的键盘及专用于抗干扰的“看门狗”电路,另外还配有一般开发机上没有而又经常用的 8 路 8 位 A/D 和 1 路 D/A 电路,并配有一标准 RS232 口和由 8255 扩充的 3 个 8 位并行或位控的 I/O 口。另外还可以直接选配 E<sup>2</sup>PROM 片子。

SCB-1 不仅硬件配置合理,而且还有愈 4K 的监控程序,它为用户提供了大量丰富实用的软件,并且都以子程序形式提供,它包括单步执行程序及现场保护与恢复程序、键扫描程序、显示程序、存储器读写程序、存储器内容数据移动与 E<sup>2</sup>PROM 编程程序、串行通讯的接收程序及 IBM-PC 兼容机上的发送程序和 SCB-1 的发送和 IBM-PC 兼容机的接收程序,还有很实用的 A/D 转换子程序、D/A 程序、打印机驱动程序以及检空、比较、定时时钟等程序,在用户要求下还可以配上各种运算符程序库。

监控工作主程序可以使用户通过键盘操作进行搬家、检查、改写内外 RAM 并可以进行连续或单步执行程序,并可以直接与 PC 机进行通讯操作。除了监控中提供的软件外,SCB-1 系统还提供一在 PC 机上运行的软盘文件,它有正反汇编程序,正反通讯程序及监控程序清单。

优良的软、硬件基础 SCB-1 具有广泛的应用价值。今年 9 月武汉大学出版社将介绍 SCB-1 单片机的《单片机原理及制作》一书选列为大中专教材,正式出版发行,该书不仅对 SCB-1 的软、硬件进行了详尽

的介绍,还从教学实际出发,配以实验操作及应用实例等内容,使 SCB-1 系统在教学领域内大受欢迎。现已有几十所各类学校用此系统进行单片机教学。

SCB-1 的带掉电保护的 16K RAM 为用户提供了足够大的程序调试空间,当单步执行程序,可由用户选择观察某个寄存器的内容变化。利用软盘提供的汇编模块,直接用汇编语言编程后传送到 SCB-1 上去,使程序可以即调即改,省却了强记、手工输入机器码等繁重劳动,使开发周期大大缩短。因而 SCB-1 还被广泛用作开发工具。

SCB-1 外形有两种结构,即整体式和分体式。分体式结构特别适用于组配各种仪器仪表,配上面板即可成产品,又由于其有专门的抗干扰电路,即使在系统遇到强干扰死机后,它也可以自动地恢复并继续运行下去,再加上易于开发的特点及 SCB-1 优越的性能价格比,使其在工业控制及仪表配套领域里的应用日益广泛。

在全国同仁的大力支持之下,我所生产的 SCB-1 单板机销售量不断上升,生产成本不断下降。使其现在零售价仅为 495 元;又由于我所毫不保留地将软、硬件资料全面公开,以用户需要为第一原则,并有各种灵活的价格和销售方式努力做好销售服务工作,深受广大用户欢迎。为更好地开拓 SCB-1 的应用市场,本所将在全国各地征求经销代理单位,并可给予优惠的价格及技术支持。对于配产品用的大批量用户,本所除可以进行专门技术指导服务外,还可以按指定要求进行生产并另行定价。

本所另有详尽的 SCB-1 的销售及服务说明,函索即寄。

单 位:武汉市尚吉电子研究所

联系地址:武汉武昌珞瑜路 37-3 号

邮政编码:430070

电 话:716138

电报挂号:4561

开户银行:工商银行洪山区办

帐号:2604-67-016072001

## 转 让

中华学习机 CEC-I 型主机一台,价格 750.00 元。负责邮寄费用并免费赠送若干软件及资料。

联系人:陈长健 地址:西藏拉萨市电信局长话科 邮编:850000

## 更 正

本刊 91 年第 6 期第 10 页《女子健美体型电脑咨询程序》一文的作者地址应为四川泸州医学院。

编辑部

一九九一年“AST”杯  
 全国计算机软件人员竞赛（第二届）  
 初级程序员笔试题（120分钟）

准考证号 \_\_\_\_\_ 地区 \_\_\_\_\_ 姓名 \_\_\_\_\_

题号	一	二	三	四	五	六	七	八	总分
分数	11	15	4	6	9	25	15	15	100
得分									

一、从供选择的答案中选出正确答案的编号填写在横线上。（每空1分，共11分）

1. 下列数中最小的数为 \_\_\_\_\_。

供选择的答案：

- A.  $(101001)_2$       B.  $(52)_{10}$       C.  $(101001)_{BCD}$       D.  $(23)_{16}$

2. 采用 \_\_\_\_\_ 表示数字时，零的表示是唯一的。

供选择的答案：

- A. 原码      B. 补码      C. 原码和补码      D. 原码和反码

3. 已知二进制数  $X = 0.11011$ ， $Y = -0.10010$ ，则  $X - Y$  的补码形式的运算结果是 \_\_\_\_\_。

供选择的答案：

- A. 1.01101      B. 1.10111      C. 0.11001      D. 0.01001

4. 指令周期是指 \_\_\_\_\_。

供选择的答案：

- A. CPU 从主存取出一条指令和执行这条指令的时间  
 B. CPU 执行一条指令的时间  
 C. CPU 从主存取出一条指令的时间  
 D. 时钟周期时间

5. 广泛用于 IBM PC 中的 Intel8088CPU 是 \_\_\_\_\_。80386 是 \_\_\_\_\_。

供选择的答案：

- A. 8 位微处理器      B. 16 位微处理器  
 C. 准 16 位微处理器      D. 32 位微处理器

6. 对逻辑表达式  $AB + \overline{A}C$  化简的结果是 \_\_\_\_\_。

供选择的答案：

- A.  $AB + BC$       B.  $AB + \overline{A}C$       C.  $\overline{A}B + \overline{A}C$       D.  $\overline{A}B + \overline{A}C$

7. 80286 采用了两种不同的操作方式，即 \_\_\_\_\_ 方式和 \_\_\_\_\_ 方式。在后一种方式下，提供分段式虚存系统所需的环境，是真正发挥硬件性能的操作方式。

供选择的答案:

- A. 保护方式      B. 批处理方式      C. 假脱机方式  
D. 虚拟方式      E. 实方式      F. 实时处理方式

8. 为用户提供诸多 DOS 功能, 具有 PCShell, DiskFix, PCBackup 和 DeskTop 功能模块的 \_\_\_\_\_ 是高级 PC 工具软件之一。

供选择的答案:

- A. Windows      B. Pctools      C. X-Window      D. Open Desktop

9. 在 IBM PC 机上使用的 CCDOS 是一个 \_\_\_\_\_ 操作系统。

供选择的答案:

- A. 中西文兼容      B. 全中文      C. 联想汉字      D. 汉字排版印刷

二、从供选择的答案中选出与下列叙述关系最密切的字句, 把编号写在题后答案对应的横线上。(每空 1 分, 共 15 分)

1. 在操作系统的存储中, 覆盖和交换是 A, 用以进行存储扩充; 设备管理中的缓冲技术是 B, 用以合理地利用外部设备。对磁盘的管理属于 C; 对磁盘中信息的管理属于 D。

供选择的答案:

- A, B. ①以空间换取时间    ②以时间换取空间    ③充分利用时间    ④充分利用空间  
C, D. ①设备管理    ②文件管理    ③驱动器管理    ④作业管理

答案: A \_\_\_\_\_ B \_\_\_\_\_ C \_\_\_\_\_ D \_\_\_\_\_

2. 排序是数据处理中的重要运算, 冒泡排序和快速排序属于 A; 直接选择排序属于 B; 二分法插入排序属于 C。

供选择的答案:

- A, B, C. ①插入排序    ②选择排序    ③交换排序  
④表插入排序    ⑤归并排序    ⑥希尔排序

答案: A \_\_\_\_\_ B \_\_\_\_\_ C \_\_\_\_\_

3. 要进行顺序查找, 则线性表 A。要进行二分法查找, 则线性表 B。若表中元素个数为  $n$ , 则顺序查找的平均比较次数为 C。

A, B. ①必须以顺序方式存储

②必须以链接方式存储

③必须以顺序方式存储, 且数据元素已按值递增或递减顺序排好

④必须以链接方式存储, 且数据元素已按值递增或递减顺序排好

⑤既可以以顺序方式存储, 也可以以链接方式存储

C. ①  $n$       ②  $n/2$       ③  $n^2$

答案: A \_\_\_\_\_ B \_\_\_\_\_ C \_\_\_\_\_

4. 栈和队列都是 A。若进栈的序列为 1, 2, 3, 4, 则 B 不可能是一个出栈序列。若进队列的序列为 1, 2, 3, 4, 则 C 是一个出队列的序列。

供选择的答案:

A. ①顺序存储的线性结构

②链式存储的线性结构

③限制存取点的线性结构

④限制存取点的非线性结构

- B. ①3, 2, 1, 4    ②3, 2, 4, 1    ③4, 2, 3, 1  
 C. ①4, 3, 2, 1    ②1, 2, 3, 4    ③1, 3, 2, 4

答案: A \_\_\_\_\_ B \_\_\_\_\_ C \_\_\_\_\_

5. 奇偶校验是一种常用的代码校验方法。这种校验方法能发现 A 个错误, B 纠正错误。

供选择的答案:

- A. ①奇数    ②偶数    ③奇数或偶数    ④奇数和偶数  
 B. ①不能    ②能    ③特定条件下能    ④某些条件下不能

答案: A \_\_\_\_\_ B \_\_\_\_\_

三、判断下列叙述是否正确,请在题前的括号内画√(正确)或×(错误)。(共4分)

- ( ) 1. 浮点运算指令对于科学计算的计算机是必要的,能提高机器的运算速度。  
 ( ) 2. I/O接口电路是一种输入/输出设备。  
 ( ) 3. 一般的1.2MB的软盘驱动器既能对1.2MB软盘进行操作,也能对360KB软盘进行操作。  
 ( ) 4. 兼容机之间指令系统基本上是相同的,但硬件实现方法可以不同。  
 ( ) 5. 数据库系统最基本的特点是管理的数据量很大。  
 ( ) 6. 数据的完整性是指保护数据以防止不合法的使用。  
 ( ) 7. 计算机病毒的破坏能力主要取决于病毒程序的长短。  
 ( ) 8. 数据库管理系统是数据库中程序和数据的总称。

四、将下列英文的中文译文写在对应横线上。(每题1分,共6分)

1. Program must be written in a language the computer will understand.

\_\_\_\_\_

2. Many microcomputer systems are single-user systems, only one person can use the systems at a time.

\_\_\_\_\_

3. The screen display tells you the program is ready to run.

\_\_\_\_\_

4. File allocation table bad, drive A Abort, Retry, Ignore?

\_\_\_\_\_

5. Warning! All data on non-removable disk drive C will be lost.

\_\_\_\_\_

6. Insufficient memory for system transfer.

\_\_\_\_\_

五、阅读下面的 BASIC 程序,写出程序运行时显示的结果:(每题3分,共9分)

```

⑩ DEF FNA(X)=1+2 * X
20 X=VAL(CHR$(65))-2
25 PRINT "X=";X
30 Y=FNA(FNA(FNA(X)/2))

```

运行结果:

```
40 PRINT "Y=";Y
```

```
50 END
```

```
②10 INPUT "输入";X
```

输入 2 显示结果:

```
20 IF X <= 0 THEN PRINT "x=";X:END
```

```
30 GOSUB 60
```

```
40 PRINT "X+X=";X+X
```

```
50 END
```

输入-4.6 显示结果:

```
60 X=X*3
```

```
70 IF ABS(INT(X))=ABS(X) THEN 90
```

```
80 PRINT"x=";X
```

输入 0 显示结果:

```
90 PRINT "&&&"
```

```
100 RETURN
```

```
③10 A$ = STR$(VAL(RIGHT$("123450",1))) 运行结果:
```

```
20 FOR X=5 TO 1 STEP -1
```

```
30 PRINT TAB(2*X+5);
```

```
40 FOR Y=1 TO 7-X
```

```
50 PRINT A$;
```

```
60 NEXT Y
```

```
70 PRINT
```

```
80 NEXT X
```

```
90 END
```

## 六、填空 (共 25 分)

①填充下面的 BASIC 程序,使其功能与 dBASE III 的 FILE( )函数类似:如果磁盘上存在指定的文件,就显示-1;否则显示 0。(4分)

提示: BASIC 程序运行时如果在磁盘上找不到指定的文件,返回的错误号是 53。

```
10 ON ERROR GOTO 100
```

```
20 CLS
```

```
30 X=-1
```

```
40 INPUT "Input a Filename:";F$
```

```
50 _____
```

```
60 _____
```

```
70 PRINT X
```

```
80 END
```

```
100 IF ERR=53 THEN X=0
```

```
110 RESUME 70
```

②设四个连续自然数中的最小数为 A,且四个数之积为 1680,填充下面的 BASIC 程序,使其能从小到大依次输出这四个数。(4分)

(待 续)



# 《电子与电脑》1991年总目录

## · 综述 ·

期	页	利用屏幕缓冲区实现语言的三重调用	张建新	3 11
为《电子与电脑》而作——中国计算机学会向广大计算机爱好者拜年	1 2	谈谈子目录加密与解密的一种方法	李连德	3 12
今天我们相识——本刊 91 年致读者	1 3	PC 机显示氢原子轨道波函数	陈可中 曾庆涛 郭冰莹	3 12
PAL 和 GAL	朱世鸿 2 2	BASIC 程序 P 加密的简易解密法	李志刚	3 13
磁盘和磁盘机	林兼 3 2	用随机文件建立数据新探	纪京松	4 5
90 年代的电脑技术和采购策略(上)	殷广济 4 2	英文 Wordstar 文件的标准 ASCII 化	赵毅	4 7
90 年代的电脑技术和采购策略(下)	殷广济 5 2	BASIC 程序 P 保护在文件加密中的应用	何骏	4 7
CGA, EGA, VGA	余红 5 3	在 dBASE III 中实现多维数组功能	石供奉	4 8
计算机技术的产物——条形码技术	于江苇 5 4	为 DOS 增加定义功能键外部命令	张红	4 9
80 年代微机发展回顾及 90 年代展望(上)	陈志雄 6 2	设置 MS-DOS 的功能键	何管略	4 11
80 年代微机发展回顾及 90 年代展望(下)	陈志雄 7 2	编译 BASIC 运行文件 CCDOS 不相容的解决方法	刘亚兵	4 12
水不过时的电脑软件——自生成软件系统	郑佑 8 2	硬盘的加锁和解锁	张文	4 12
参加 91' 第三届国际信息学奥林匹克(IOI'91)		键控图形处理程序	徐维科	4 13
竞赛的体会	吴文虎 薛宏熙 9 2	彩色的高分辨图形	潘麟元	4 14
众说纷云:8 位学习机是否已过时?		回溯法及其在编程中的应用	陈磊	4 15
专家认定:君在教育界仍是主力军!	9 3	通用打印机状态检测程序的改进	夏伟文	5 5
论电脑写作	彭明光 10 2	通用制表程序	赵欣	5 6
在中华机上扩充中文输入码的探索		也谈 Cdbase III 的纵向送数方式	王传信	5 9
	胡志群 胡志广 胡沛云 11 2	PCTOOLS 的使用技巧	刘建明	5 11
新型信息录入技术-条码技术及其应用	王枫等 12 2	程序常驻内存(TSR)的实现方法	李华	5 12
		防止非法使用 DBASE 程序的一种方法	黄世刚	5 14
· PC 用户 ·		也谈 IBM 机图形存取	焦小友	5 14
单色卡仿真 CGA 视频标准的图形功能	朱大公 1 4	侦销核对程序	张永辉	5 14
西文 Wordstar 文件在中文状态下的操作	尤建忠 1 6	汉字同音字查询的实现	李凤明	6 4
试论磁盘文件的最佳存储方法	陈宏明 1 8	如何提高 CdBASE III 数据库汇总速度	刘昌斌	6 6
断电保护打印程序	郭河伟 1 9	文件目录修复一例	唐银红	6 6
将三角函数引进 DBASE 系统	何管略 1 10	多功能卡日历/时钟功能的开发	张翠玲	6 8
经验与技巧两则	吴仲文 1 11	Turbo C 问答	施悦华	6 8
中国计算机学会 1990 年度全国青少年信息学(计算机)通信赛试题	1 12	就 Turbo C 谈充分利用系统内存	宋卫东	6 9
通用打印机状态检测程序	夏伟文 2 3	女子健美休型电脑咨询程序	黎彦才等	6 9
dBASE III 与 BASICA 之间的数据传送问题	严桂兰 2 5	使清屏画面变美妙	王存	6 11
dBASE 运用经验与技巧	孙鹏 2 7	技巧二则	汉神	6 12
如何解决在运行 FOXBASE 内存不够的问题	周日初 2 9	利用 DEBUG 进行反汇编存盘的方法	金林樵	6 13
COMPRESS 在硬盘集中存储中的应用	许鹰 2 9	90 年全国计算机通讯赛题分析	汪五一	6 14
怎样在 IBM-PC 机上实现汉字的动态显示	李晓华 2 10	用未公布的 dBASE III 命令开发图形功能	金林樵	7 4
DOS I/O 重定向、管道功能及其使用	徐国辉 2 11	谈 dBASE III “模糊”查询	班艳玲	7 5
在低牌本 DOS 下安装虚拟盘	周健勇 2 12	FOXBASE 编程技巧两则	贺文华	7 5
五笔 CCDOS 中一字节修正	吴帮忠 2 12	恢复误删除文件的原理和方法	朱大公	7 6
一个快速实用的字符串替换程序	金林樵 2 12	微机绘制平面曲线	魏新民	7 9
PC 机磁盘文件的加密与解密	刘忠达 2 13	若干子目录下共用一个 dBASE III (或 WS) 软件包	龚文彪	7 11
硬盘正确格式化一例	袁士彬 2 14	小议 PC-DOS 的拷贝命令	刘杰	7 11
数据库通用维护模块设计	徐维祥 3 4	能读多位数的计算机程序	周礼森	7 12
字典库的编辑和使用	黄焕如 3 7	模拟钢琴演奏	冯剑秘	7 12
用 DOS 命令设计菜单	陈凯 3 8	如何对中文 DOS 的汉字库进行直接读写	刘旭华	7 13
在 IBM PC/AT 机异步通信适配器的自检方法	杨沂 杨耘华 3 10	NC—新型文件管理工具软件	杨进	7 14
产生真正的随机数	应海涛 3 11	通用下拉式菜单的实现方法	张志远	8 3

如何改变数据记录在库中的绝对位置 .....	李海飞	8	4	巧用 FID 解密 .....	杨克	1	22
dBASE III 过程文件加密的发现 .....	刘人伟	8	5	CALL 新用法 .....	姜路	1	23
对 SET CARRY ON 的一点补充 .....	李志刚	8	5	APPLE II 的口令设置 .....	成鹰	1	23
IBM PC/XT 磁盘扇区读取方法分析 .....	吴兆熊	8	6	性格类型测试程序 .....	叶平	1	23
使用设备驱动程序 ANSI · SYS 扩充 屏幕和键盘 .....	曹长英 李杰	8	8	程序出错处理与 RESUME 的功能扩充 .....	丁志伟	2	15
用 PC TOOLS 修改软件的汉字提示 .....	李开新	8	10	特殊 6502 指令 .....	李铁	2	17
巧用批命令定义功能键 .....	邢观锋	8	10	四舍六入法程序 .....	韦肖鸿	2	17
CCDOS 下屏幕死锁的解决 .....	成刚 杨明秋	8	11	“怪”指令在软件加密中的应用 .....	傅剑	2	18
VDISK 的妙用 .....	叶舸	8	11	双显示电子钟 .....	万斌	2	19
IBM PC/XT 定时中断服务程序设计 .....				苹果机汉字造型表自动生成程序 .....	刘士明	2	20
..... 张辉 李东升 李奎贤		9	5	高效率的排序方法 .....	陈庆祥	2	21
通用数据库修改程序 .....	周日初	9	7	演示抛物面的光学效应 .....	王晓林	2	23
PC-DOS 命令 WHEREIS 的改正 .....	李敏	9	8	LASER QZT/C 应用经验 .....	张海翔	2	24
巧算 1000! .....	李帆	9	9	BASIC 语言实现递归 .....	苏亚华	3	14
又发现一种新病毒 .....	王家春	9	10	数据处理原理和技巧 .....	朱国江	3	15
杨基病毒快速解毒法 .....	李凤明	9	10	哥德巴赫偶数猜想的 BASIC 程序 .....	陈君佐	3	18
设备驱动程序 ANSI.SYS 的使用 .....	王立勇	9	11	梵塔移动的动态模拟 .....	刘闯	3	19
巧用虚拟盘 .....	李焕民	9	12	查找内存中的代码 .....	陈宇	3	20
用递归解九连环 .....	钟武	9	12	奇妙的图案 .....	李明	3	20
BROWSE 命令中 FIELDS 参数的菜单选择 .....	张述健	10	4	最简数据盘生成法 .....	江东	3	20
给 dBASE 程序加入活用功能 .....	贺映文	10	5	超级 8—40 单元打印 .....	王壮	3	21
在 dBASE III 中实现多维数组功能的另一方法 .....	程林	10	6	将 16 进制代码翻译成 DATA 语句的数据 .....	张振堂	3	21
集成字处理软件 CCED 的使用技巧 .....	许鹰	10	6	关于多目录磁道问题 .....	张剑平	3	22
如何自编 Park? .....	葛芝宾	10	7	电子表键盘输入方式的改进 .....	田洪瑜	3	22
需 ROM BASIC 支持的软盘改造 .....	曾礼伟	10	8	LASER 301 兼做数字石英钟 .....	郑嘉琦	3	23
最新病毒检测及清除软件 VIRUSCAN Ver7.2V77 ..... 梁伟杰		10	9	中华学习机使用技巧 .....	任义	4	16
Goldbach 奇数猜想的 BASIC 程序 .....	陈君佐	10	10	实用改良的 NEWCODE-POKES 程序 .....	黄影昌	4	17
快速排序“快”在何处? .....	应宏	10	11	巧解“八皇后”问题 .....	柳杨	4	17
两个高效率的排序程序 .....	李军杰	10	11	对《APPLE 和 CEC—I 设计者的疏忽》一文看法 .....	刘峰	4	18
利用 PASCAL 扩充 dBASE III 的函数功能 .....				APPLE 缺陷的解决办法 .....	陈盛施	4	18
..... 王越 张秀邦		11	4	数值计算中存在问题的解决办法 .....	朱洪波	4	18
如何动态生成有记忆功能的菜单 .....	苏士俊	11	6	无数据磁盘格式化程序 .....	姜宏岩	4	19
在 dBASE III 下实现菜单的光带键选 .....	阮自力	11	7	跟踪机器语言程序的运行 .....	田俊成	4	20
双列分页打印程序 .....	陈锋	11	8	局部清屏程序 .....	程华	4	20
广州大学一号病毒解除与免疫 .....	张文锋	11	10	扩展 BLOAD 命令的功能 .....	蒋阳	4	21
谈谈 2.13 汉字系统在软盘上的使用 .....	廖红旭	11	11	从“重赏”谈起 .....	卢保堂 卢玉堂	4	21
PC/XT 机病毒报警程序 .....	王传智 刘秉瀚	11	12	在 APPLE SOFT 下使用小汇编 .....	稽克威	4	22
考生姓名编码自动生成程序 .....	任绥海	11	12	工资票面张数计算程序的简化 .....	孙承灿	4	22
dBASE III 下实现 Foxbase 中 prompt 的菜单功能 .....				让 CEC-BASEC 也有自动行号功能 .....	沈东山	5	17
..... 毛维平		12	4	图形屏幕坐标的对应存储单元 .....	葛海东	5	18
信息系统中模糊查找的设计 .....	戚靖山 朱玉环	12	5	神奇的 \$ COBO 单元 .....	李峰	5	18
高密软盘的单驱动器拷贝 .....	李毅	12	6	怎样解读 CEC-I 学习机的汉字系统 .....	刘文勤	5	18
用汇编程序显示硬盘主引导记录 .....	李日林	12	7	用机器语言解八皇后问题及其推广 .....			
PC DOS 内存管理 .....	陈宏	12	8	..... 户祖兴 刘超宏		5	19
CC BIOS 2.13F 的小补丁 .....	严仲武	12	9	也谈汉字平移显示 .....	李茂学	5	19
中文弹出式窗口设计 .....	姜咏江	12	10	前台显示后台绘图的秘密 .....	谢吉华	5	20
一种实用新颖的排序方法 .....	朱国江	12	12	演示抛体运动 .....	白梅兰	5	20
<b>· 学习机之友 ·</b>				多单元电视接收无线设计程序 .....	史文忠	5	22
APPLE I 高分辨率绘图与文字编辑技术 .....	邹培国	1	15	一个实用的造型表编辑程序 .....	左震宇	5	23
向量图形表自动生成程序 .....	万维侠	1	18	CEC-I 全功能造字 .....	傅叔平	6	16
学生成绩管理程序 .....	朱江	1	20	梵塔问题和奇数的幻方的一行程序 .....	陈纲	6	18
CEC 控制键的调整及保护 .....	张保田	1	21	APPLE-I 机打印格式的控制 .....	张冠玉	6	19
磁带小汇编系统自制简法 .....	石勇 张优方	1	22	将 BASIC 程序变成 B 型文件并以 BRUN 运行 .....			
				..... 杨峰		6	20

APPLE DOS 的一处错误 .....	尹进渝	6	20	ProDOS 磁盘操作系统入门(续) .....	廖凯	12	18
ProDOS 系统盘的修改 .....	廖凯	6	21	· C 语言初阶 ·			
制作 40 磁道磁盘的最简方法 .....	瞿波	6	21	《C 语言讲座》第一讲 TurboC .....	李文兵	1	25
PC-1500 机新监控程序 .....	沈玉波	6	22	第二讲 Turbo C 基本数据 .....	李文兵	2	25
关于四舍五入程序的讨论 .....	梁放 叶德祥	6	25	第三讲 命令行 Turbo C 和实用程序 .....	李文兵	3	24
为无驱动器系统设置虚拟磁盘 .....	刘庆丰	7	15	第四讲 C 运算符 .....	李文兵	4	23
巧用 LOGO 解难题 .....	廖大勇	7	16	第五讲 程序控制结构(一) .....			
修改 CEC 汉字系统程序 .....	周易	7	16	..... 李文兵 王玉华 蓝智斌	5	25	
浅谈计算机在中专数学中的应用 .....	王志平 顾正刚	7	17	第五讲 程序控制结构(二) .....			
数据块的后移 .....	冯端品	7	18	..... 李文兵 王玉华 蓝智斌	6	23	
从 QXC 卡读取汉字字模 .....	牛海峰	7	18	第六讲 宏指令的用法 .....	李文兵	7	23
ProDOS 磁盘操作系统入门 .....	廖凯	7	19	第七讲 结构数据(一) .....	李文兵	8	22
在 APPLESOFT 系统下实现 HSCRN 功能 .....	李齐	7	21	第七讲 结构数据(二) .....	李文兵	9	24
BASIC 语言中几个命令功能的扩展 .....	那履弘	7	21	第八讲 指针与数组 .....	李文兵	10	21
谈谈 FORTH .....	丁志伟	8	12	第九讲 指针与函数、结构 .....	李文兵	11	23
再谈特殊 6502 指令 .....	李永前	8	13	第十讲 文件管理 .....	李文兵	12	20
扩展监控系统命令 .....	蒋建一	8	14	· 学用单片机 ·			
MZ-731 机绘制的财务计划执行分析报告 .....	黄浩珍	8	14	单片机化产品中的键盘和显示器的设计 .....	吕锡庆	1	29
机器语言程序计算阶乘 .....	易鹤龄	8	17	EPROM 及 8751 安全智能编程法 .....	翁甲辉 李柯	1	31
ProDOS 磁盘操作系统入门(续) .....	廖凯	8	18	普及型单片机开发工具的硬件设计 .....			
用堆栈思想处理任意 N 个元素的排列组合问题 .....				..... 张培仁 刘振安	2	29	
..... 索素文	8	20		普及型单片机开发工具的软件设计 .....			
简单有趣的解密办法 .....	张亨	8	22	..... 张培仁 刘振安	3	28	
将 CEC-I 中华学习机改制成专用机 .....	姜良夫	9	14	智力竞赛抢答装置 .....	宋济华	4	27
用栈结构优化程序 .....	陈合力	9	17	单片机电子音乐编程实践 .....	张培仁 张新宇	4	30
寻找无理数 e 的约率和密率 .....	施为钢	9	17	简易单片机电子琴 .....	张培仁 张新宇	5	30
怎样节省 CEC-I 存储空间 .....	尹向阳 梁曦	9	18	MCS-51 单片机在听力播放、发射系统中			
ProDOS 磁盘操作系统入门(续) .....	廖凯	9	19	的应用 .....	沈乃才 宋振环	5	32
数制转换简法 .....	李峰	9	20	单片机最小应用系统 .....	张培仁	6	26
也谈扩展 COPY I PLUS4.0 .....	关炳坤	9	21	电脑坦克-KDC 最小系统应用实例之一 .....			
从 CP/M 转移文件到 APPLE DOS 下 .....	方福兴	9	21	..... 张培仁 严琦 阮永光	6	27	
多功能 CEC-I 高速排序 .....	王庆顺	9	22	单片机旅馆客房门卫系统 .....	傅士澄	7	27
小学生速算训练 .....	李永升	9	23	单片机汉字打印驱动程序的设计思想 .....	尹征	8	25
CEC-I 高级动画制作技术 .....	黄一禾	10	13	MCS-51 单片机应用中二种 LED 显示方法 .....	傅篱	9	27
警察抓小偷 .....	李明	10	15	多功能电子钟 .....	张培仁 刘振安	10	27
ProDOS 磁盘操作系统入门(续) .....	廖凯	10	16	汽车转弯信号灯控制 .....	张培仁 刘振安	11	26
高性能的 CEC-I 自定义组合键 .....	杨勇佳	10	18	SCB-I 型单板单片机的存贮器改进 .....	江琪	12	26
活泼的光标 .....	殷瑞冬	10	19	MCS-51 单片机端口结构图示中的一个普遍错误 .....			
CEC-I 机汉字使用第十一行 .....	任义	10	19	..... 王修才	12	27	
三页图形竖向并打 .....	陈钢	10	20	· 学装微电脑 ·			
电子打字机 .....	王蔚	10	20	学装 $\mu$ P-80 手动 EPROM 写入器 .....	易齐千	1	33
把 APPLE DOS 移入 RAM 卡 .....	傅剑	11	14	40 支开关输入部件的制作 .....	易齐千	2	34
将程序 CHAIN 升级为 APPLE DOS 命令 .....				8 位 A/D 变换部件的制作 .....	易齐千	3	31
..... 赵江龙 孙杰亮	11	15		数字游标卡尺的制作 .....	易齐千	4	34
向天坛机移植中华机汉字软件的一些体会与经验 .....				用微电脑检查印刷电路板 .....	易齐千	5	37
..... 张斌	11	17		微电脑控制电梯模型的实验与制作 .....	易齐千	6	33
FORTH 问答堆栈和逆波兰表示法 .....	丁志伟	11	18	微电脑控制跨接弯制机 .....	易齐千	8	29
中华学习机图书管理程序 .....	王二珠	11	19	相位控制部件的制作 .....	易齐千	9	29
ProDOS 磁盘操作系统入门(续) .....	廖凯	11	21	Z80 CPU 与单片微处理器 8031 的交换 .....	高殿斌	11	28
多层次全功能阴影菜单 .....	黄文忠	12	13	串行 I/O 部件的制作与实验 .....	易齐千	12	28
虚线 .....	刘裕	12	14	“微电脑控制技术”函授班总结 .....	易齐千	12	30
技巧两则 .....	宋正荣	12	15	· 电脑巧开发 ·			
改进 MUSIC 的发音 .....	熊波	12	15	微电脑控制电饭锅 .....	万国泰	1	35
中华学习机 RAM 监控及其应用 .....	朴一兵	12	16	MP-I 汉字及图形显示装置 .....	陈名则	2	37
CEC-I 键控光标作图程序 .....	陈国良	12	17	APPLE-I 彩灯自控电路 .....	张孝玖	2	38

MP-I 微电脑加装编码键盘 .....	李志平	3	36	微机软盘驱动器“卡盘”的简易修复 .....	乐冬青	10	42
扫描式红外线十路遥控器 .....	王刚勤	3	37	CEC-I 软驱故障维修一例 .....	潘冬	10	42
语言报时钟的原理与实现 .....	蔡莲红	5	28	“坏”软盘的再生 .....	张智	10	43
8031 与 PP40 描绘器接口技术 .....	皇甫文忠	6	29	GW286BH 系统板维修一例 .....	董纯坚	10	43
解决 TEC-B1 不能使用 AP51-I 卡 问题的方法 .....	刘希栋	6	32	怎样使用清洗盘 .....	瞿新国	10	43
MSX-EPROM 读写/控除器 .....	刘园	7	32	驱动器修理一例 .....	张红平	11	35
如何用 EPROM 读写器对 EEPROM 编程 .....	闫怀兵 陈立山	8	35	硬磁盘驱动器专用接口电路 501262 的测试 .....	卜建辉	11	36
LASER-310 机加接数字音响发生器 .....	王建锋	8	36	IBM-PC/XT 一种系统错误的分析 .....	姜金友	11	41
怎样用电脑检修录像机 .....	李玉全	9	32	用非格式化方法修复硬盘一例 .....	周建辉	11	42
用娃娃机开发单片机程序 .....	张涛	9	32	检修 IBM AS/400B35 故障二例 .....	王志远	11	42
简易精确的 APPLE I 时钟卡 .....	姜仕仁 施青松	10	30	自己学修计算机 .....	陈勇	11	43
用 EPROM 制作多路报警器 .....	王洪伟	10	32	TEC-I 机磁盘驱动器 I/O 接口板故障检修一例 .....	崔时珍 玄勇活	11	45
在 APPLE 机接口卡中固化 BASIC 应用程序 .....	解武杰	11	30	3070 打印机断针定位方法及 8088 汇编程序 .....	龙兵生	11	46
设计微电脑控制电路的小常识 .....	王林	11	31	打印机故障维修一以 STAR 或 GX-15 打印机为例 .....	王威 王候	11	47
型号不同的绘图仪绘制 TANGO 软件图形的方法 .....	黄健 陈军	11	32	• 初级程序员级水平考试辅导问答 • 1991 年初级程序员(有奖学习)水平考试辅导讲座			
将 MP-I 电脑的时基信号为晶体振荡 .....	邓文超	12	31	第一讲 计算机基础知识(上) .....	王路敬	1	39
• 维修经验谈 • DM-2214 高分辨显示器无字符的检修及 提高字符清晰度的研究 .....	杨敦行	1	37	计算机基础知识(下) .....	王路敬	2	43
王安机 RAM 故障简易检修法 .....	董纯坚	2	40	微型计算机操作系统使用问答 .....	王路敬	3	40
IBM286 维修一例 .....	叶志斌	2	41	微型数据录入操作回答 .....	刘英 张茂芝	4	40
中华学习机典型故障一例的分析 .....	杨瑞华	2	42	BASIC 语言程序设计回答 .....	李志刚 吕品	5	44
APPLE II 主机常见故障的检修 .....	陈鸿	2	42	汉字 dBASE III 使用技巧(上) .....	王路敬	6	37
软盘驱动器机械故障的分析与维修 .....	魏学则	4	44	汉字 dBASE III 使用技巧(中) .....	王路敬	7	39
更换集成块一法 .....	吴广元	4	44	汉字 dBASE III 使用技巧(下) .....	王路敬	9	39
APPLE-I 机内存故障维修方法 .....	腾达	4	45	91 年第三届国际信息学奥林匹克(IOI'91)试题 .....	吴文虎	9	46
巧修 CEC-I 受损键盘 .....	黄文贵	4	45	Z80 的“新”指令 .....	章永得	9	47
彩色显示器亮度失控的探讨 .....	胡野红	6	45	• 新书与软件 • ORACLE 关系数据库管理系统 .....	王学民 魏国	1	44
微计算机的维修工具-8088 系统板故障诊断卡 .....	董珊	6	46	Turbo CAD 软件简介 .....		2	46
任天堂系列游戏机检修实例 .....	林华春	7	34	VIRUSCAN 病毒处理软件包功能简介 .....	朱禹	4	46
让 PC 机接上家用彩电——CGA-PAL 卡即将问世 .....		7	35	Microsoft Windows 3.0 窗口软件简介 .....		4	46
微计算机维修工具-8088 系统板故障诊断卡(续) .....	董珊	7	36	最新版本 PC TOOLS V6.0 .....		4	47
磁盘驱动器修复一例 .....	雷勇	7	38	TANGO 印刷线路板自动布线软件包 .....		7	45
任天堂系列游戏机检修实例(续) .....	林华春	8	38	电子工业出版社新书介绍 .....		7	45
“8088 系统板测试卡”对系统板总线部分的诊断测试 .....	梁杰熙	8	39	电子工业出版社软件部发行 CCED V3.0 中文字 表编辑软件 .....		7	46
一九九一年计算机初级程序员有奖竞赛试题 .....		8	41	给 TANGO 软件增加汉字元件库 .....	黄健 陈军	8	47
一九九一年计算机初级程序员有奖竞赛试题答卷 .....		8	48	• 读者联谊 • 普及型 PC 机个人用户软件交流联谊活动 .....		1	46
TH3070 打印机接口电路工作原理与故障分析 .....	李燕华	9	35	湖北地区中华学习机联谊活动情况 .....		2	48
长城 0520CH 机打印机适配器故障维修 .....	邱槟	9	37	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	7	47
AT(286)微机硬盘处理一例 .....	刘文峰	10	35	• 游戏机电脑 • 键盘电视游戏机的 FBASIC 语言及程序设计(上) .....			
从屏幕显示异常判断苹果主板故障 .....	张艾美	10	36	..... 于春		10	33
M2024 型打印机故障检修一例 .....	王峰	10	37	键盘电视游戏机的 FBASIC 语言及程序设计(下) .....			
计算机设备故障检修表 .....	管金省 宋靖涛	10	38	..... 于春		11	33
IBM PC/XT 微型计算机检修一例 .....	平凡	10	39	第二章 FBASIC 的基本语句 .....			
MP-I 微电脑快速检修法 .....	李红	10	40	巧用电脑作幻方 .....	郭高惠	2	47
如何使九针打印机字迹更清晰 .....	陆秀忠	10	41				
YOKO747 型教学语言学习机的修理 .....	郭长福	10	41				