

E&C

1992

●一九九二年 ●总期第86期

5

電子

ISSN 1000-1077

479

與 電腦



• **ELECTRONICS AND COMPUTERS** •

《LDQ光笔》简介

CEC—I 计算机是我国目前大量为社会拥有的一种普及型计算机，遗憾的是这种计算机没有光笔接口。

由武汉葛化工业集团公司职工大学教师刘待秋同志经过近一年时间的努力开发，使得CEC—I 计算机有了光笔。从此，CEC—I 计算机用户和计算机交流的手段就不再仅仅是键盘和游戏杆了。CEC—I 计算机的用户有了一支能直接在显示器屏幕上处理各种信息的“神笔”。

在91年度武汉地区发明展览会上这支“神笔”受到参观者和评委们的一致好评，并在第六届全国发明展览会上荣获铜牌奖。现在，这支“神笔”被定名为“LDQ光笔”。该光笔的性能详见下表：

项 目	指 标
定位精度	各显视页面的最小屏幕显示单位
跟踪速度	6.25屏幕显示单位/秒 (CEC—I 计算机、PAL制) 7.5屏幕显示单位/秒 (APPLE—II 计算机、NTSC制)
接口方式	游戏口接入主机 (不占用扩展槽，不接任何跳线)
固定方式	吸盘吸附
软件环境	LIU 1.0 LIGHT PEN SYSTEM (由DOS 3.3扩充形成，该系统除可使用DOS 3.3的全部操作命令外，还可在BASIC语言或汇编语言程序中使用光笔语句或调用光笔子程序)
最低硬件环境	64K主机，单软盘驱动器，单色或彩色CRT (也可用电视机)

这种光笔目前已经投入生产，我中心负责经销，批零兼营，欢迎来人来函洽谈。(含软件系统一盘和使用手册一本)

总经销单位：武汉葛化化工新技术产业开发中心

地址：武昌武珞路特一号 (陆军医院大门口东走50米)

邮编：430070

经销单位：湖北无线电厂门市部

地址：汉口胜利街130号

邮编：430014

电挂：2730

电话：237582



脉冲数字电路 (4.5元)

电工原理 (5.9元)

电子线路 (7.2元)

彩色电视机原理与维修 (7.0元)

黑白电视机原理与维修 (上) (6.0)

黑白电视机原理与维修 (下) (5.7元)

● 国内代号：2-888 定价：0.95元

让中华机发挥更大作用的 中华机教学网 (CENET)

(资料)

· CENET有二种使用方式:

1. 示教方式: 老师机处于主动地位, 可把自己屏幕显示送到学生机; 又可将学生机的内容取回, 随时检查其学习情况。
2. 多用户方式: 老师机作为网络服务器。每个学生机均可独立在DOS 3.3状态下运行, 并自由使用公共软盘机和打印机; 学生机又可共享老师机中的大量信息。

二种方式交叉使用, 非常便于组织课堂教学和练习。

· 信号传输距离为120M, 传输率0.1Mb/s, 网络节点可达256台。

· 网络由Apple机组成时, 在老师机上插入汉卡(CEH-1), 所有学生机不用插汉卡, 即可可直接运行中华机的汉化软件。

图1. CENET-I 典型配置 图2. CENET-I 典型配置之一 图3. CENET-II 典型配置之二

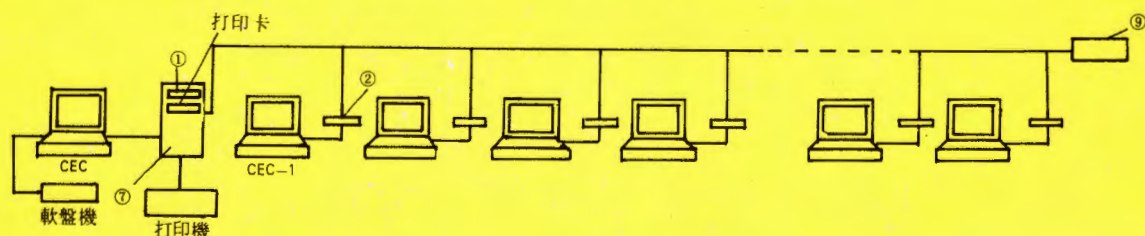


图1 · CENET-I 典型配置

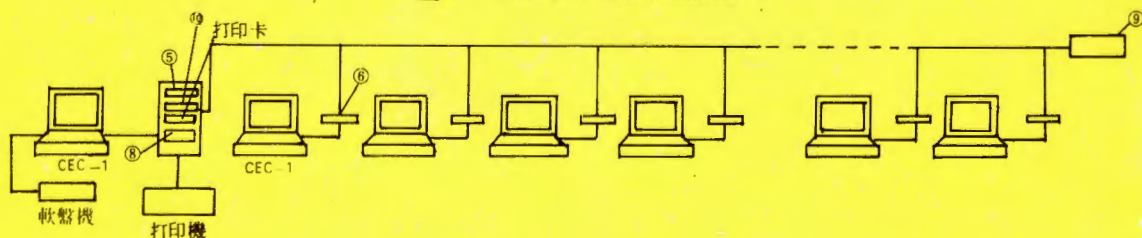


图2 · CENET-II 典型配置一

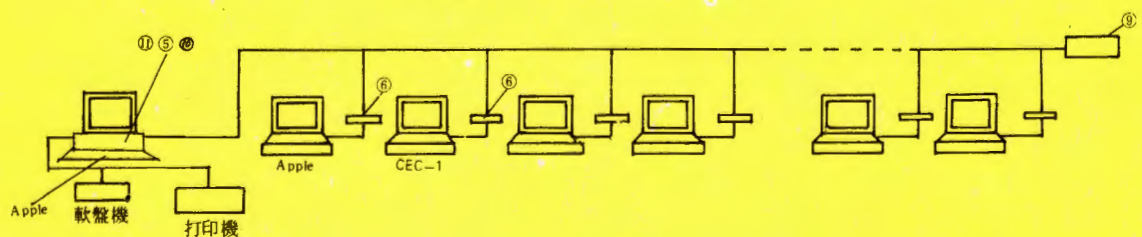


图3 · CENET-II 典型配置二

① 中华网络卡CENET-J-C-1 I型网用, 中华机做老师机用卡

② 中华网络卡CENET-X-C-1 I型网用, 中华机做学生机用卡

③ 中华网络卡CENET-J-A-1 I型网用, Apple机做老师机用卡

④ 中华网络卡CENET-X-A-1 I型网用, Apple机做学生机用卡

⑤ 中华网络卡CENET-J-2 II型网用, 老师机用卡

⑥ 中华网络卡CENET-X-2 II型网用, 学生机用卡

⑦ 中华机扩展箱CEK-1 I型网用, 二个扩展槽

⑧ 中华机扩展箱CEK-2 II型网用, 四个扩展槽, 带电源和缓冲

⑨ 中华网络匹配器CEP-1

⑩ 中华机内存扩展卡CEZ-512用于II型网扩展内存512KB

⑪ 中华汉卡CEH-1 II级汉字库, 用于Apple做老师机时

向读者提供“新一代Z80”资料

编者按：本刊第二期刊登了有关“新一代Z80”芯片介绍之后，许多读者致函本刊及中国电子器材深圳公司索取有关资料。为此，经与Zilog Asia Ltd中国代理—中国电子器材深圳公司协商，特向读者提供有关技术资料的影印件（英文）。目录如下：

Z80* Product Family	Page
Introduction.....	2
Product Migration Chart	3
Product Reference Chart	4
Z84C00 NMOS/CMOS Z80 CPU Product Specification	7
Z84C01 Z80 CPU with CGC Product Specification	43
Z8410/C10 NMOS/CMOS Z80 DMA Product Specification	75
Z8420/C20 NMOS/CMOS Z80 PIO Product Specification	99
Z8430/C30 NMOS/CMOS Z80 CTC Product Specification	115
Z8440/1/2/4/Z84C40/1/2/3/4 NMOS/CMOS Z80 SIO Product Specification	131
Z84C50 RAM80 Preliminary Product Specification	155
Z8470 Z80 DART Product Specification	193
Z84C90 CMOS Z80 KIO™ Product Specification	205
Z84011/C11 PIO Parallel I/O Controller Product Specification	225
Z84013/15, Z84C13/C15 CMOS IPC™ (Intelligent Peripheral Controller) Product Specification	287
Z80180 Z180™ MPU Product Specification	351
Z80181 SAC™ (Smart Access Controller) Product Specification	393
Z280™ MPU Preliminary Product Specification	465
 Z80 Application Notes and Technical Articles.....	 528
Z80 Family Interrupt Structure	529
Using the Z80 SIO in Async Communications	541
Using the Z80 SIO with SDLC	567
Binary Synchronous Comm Using the Z80 SIO	577
Serial Communication with the Z80A DART	581
Interfacing Z8500 Peripherals to the Z80	591
Serial Clock Generation using the Z8536 CIO	597
Timing in Interrupt-Based System with Z80 CTC	601
A Z80-Based System Using the DMA with the SIO	613
Interfacing Z80 CPUs to the Z8500 Peripheral Family	623
Z180/SCC™ Serial Communications Controller Interface at 10 MHz	647
Z80 Using the 84C11/C13/C15 in place of the 84011/013/015	682
Z80 Questions and Answers	687
Z180 Questions and Answers	703
 Additional Information	 715
Z80 Product Support	717
Zilog Quality and Reliability Report	733
Zilog Literature List	735
Package Information	741
Ordering Information	749

A 需要以上资料的读者，请写明起止页数，按0.50元/页计费，另加15%邮费。寄：“广州市1501信箱9分箱《电子质量》编辑部”（邮编：510610）”邮购。注明购“新一代Z80”资料。

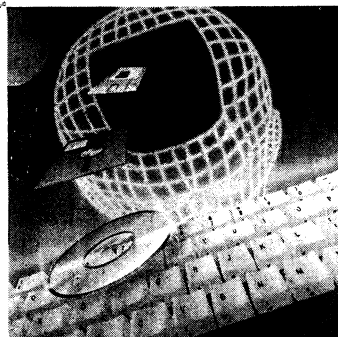
欲购器件可直接同“中国电子器材深圳公司”联系。地址：深圳市振华路（华发北路口）电子器材大厦。邮编：518031，电话：354214 354345 联系人：董伯明



電子

ISSN 1000-1077

與電腦



• ELECTRONICS AND COMPUTERS •

4791
5

一九九二年

总期第86期

電子與電腦

目 录

• 综述 •

91 年度 COMDEX 电脑博览会 吕问黎(2)

• PC 用户 •

获取 FOX 文件中密码的两种简单方法 周日初(3)

对全屏编辑命令 BROWSE 做一点补充 宋开胜(4)

感染 OMICROX 病毒文件的消毒方法 刘东祥(5)

COLOR400 卡屏彩色图形的硬拷贝 张辉 李东升(6)

利用格式化功能使磁盘“加锁” 张李文(9)

一种有效的软件解密方法 张彦洪(12)

三维图形显示的遮掩技术及实现 季军杰(13)

BASIC 超长随机文件的处理方法 朱建平(15)

• 学习机之友 •

打印机的定位和格式控制 栾傲发(16)

放大打印 APPLE II 高分辨率图形又一法 刘善平(18)

趣味速算练习器 赵 旭(19)

计算机解决数学问题 赵方明(19)

巧改系统 张 浩(20)

数字游戏 闫 浩(20)

ProDos 磁盘操作系统入门(续) 廖 凯(21)

• 6502 机器语言讲座 •

第五章 源程序的编辑、汇编和运行 朱国江(24)

• 初级程序员级软件水平考试辅导 •

数据库基础 阎 璐(29)

• 学用单片机 •

提高单片机最小系统抗干扰能力和自恢复方法

..... 张培仁、刘振安(32)

TP801 单板机与微型机数据双向并行传送一例

..... 刘浔和(33)

MCS-51 汇编指令外延的应用 陈亿善(34)

• 学装微电脑 •

制作能暂停显示的计数器 易齐干(35)

• 电脑巧开发 •

简易字幕灯的制作与控制 杨宪泽(38)

DOS 系统下多种软件在一台微机上的共存方法

..... 尤建忠(40)

• 电脑游戏机 •

第四章 F BASIC 语言的深入理解 于 春(41)

• 维修经验谈 •

TH3070 打印机常见故障分析与检修(上) 刘立华(44)

• 读者联谊 •

普及型 PC 个人用户软件交流联谊活动问题解答(五)

..... 王路敬(47)

封一:电脑键盘

封二:让中华机发挥更大作用的中华机教学网(CENET)

封三:向读者提供“新一代 Z80”资料

封四:《LDQ 光笔》简介

机械电子工业部电子工业出版社主办

编辑、出版:《电子与电脑》编辑部

(北京 173 信箱 邮政编码:100036)

印刷:北京三二〇九厂

国内总发行:北京报刊发行局

国内统一刊号:CN11-2199

邮发代号:2-888

国外代号:M924

出版日期:每月 23 日

主编:王惠民 副主编:王昌铭

责任编辑:施玉新

订购处:全国各地邮电局

国外总发行:中国国际图书贸易总公司

(北京 399 信箱 邮政编码 100044)

广告经营许可证:京海工商广字 147 号

定价:0.95 元



91 年度 COMDEX 电脑博览会

吕问黎

美国拉斯维加斯每年举行的电脑界盛会——COMDEX 博览会是北美规模最大的个人计算机产品展销会。1991 年的博览会共有来自 100 多个国家的 18000 名参加者,数百家公司和一些国家如美国、印度、比利时等的代表。

微型软件公司(Microsoft)的著名软件设计师比尔·盖茨首先作了基调性发言《指尖的信息》,指明了个人电脑今后发展的四个实用方向:1. 办公室系统中的声音、视觉与传真组成的综合信息传输网;2. 为农业工人服务的数据和信息处理方面的电子装置;3. 家用计算机进行室内规划和设计厨房设施;4. 在学校辅助教学。盖茨的发言不仅是微型软件公司的战略目标,也是美国电脑业今后的发展方向。

在“软件出版者协会”主办的一个研讨会上,与会者们着重探讨了“手写输入计算机”(Pen-based Computer)的发展前景,代表们注意到了首先进入市场的 Linus 手写系统。此外 Go 和 Slate 两家大公司在这方面的发展也引人注目,他们都在开发这种电脑的应用软件,最初的应用是帮助用户填写表格,然后就要研制便于携带的笔记本式电脑(Note book Computer)。

· 本届博览会上,各种尖新产品层出不穷,厂家之间竞争激烈。

便携式计算机 便携式电脑的市场这几年增长神速,带来激烈竞争,包括 Intel 公司在内的 40 家企业正在研制 386 膝上式电脑和笔记本式电脑,其产品颇为类似,内存都是 1—2 兆字节,硬盘都是 20~50 兆,都有鼠标和 VGA 显示屏。APPLE 和 Intel 两家大公司也在研制高度综合化的便携式集成电路,这将使明天的电脑更小、更轻、更便宜。

附加硬卡 近年来,靠外围设备来辅助主机以提高效率的方法颇为盛行,普拉斯发展公司的 IIXL 硬卡代表了这种趋势。这种硬卡可以提供大量存储空间;并且读写速度非常快,高过以前常用的硬盘,表 1 显示了一个实例;它的安装也极为方便,操作人员当场表演了五分钟装上主机的技术,令在场的观众叹为观止。

税单软件 杰普软件公司推出了“特普税单”软件。这套软件在 APPLE 公司的 Macintosh 系列机上运行,主要帮助用户处理纳税方面的事务。它包括多达 60 种纳税方式,为一些州的特种税也专门做了考虑,并且速度极快。对一般纳税人一年纳税情况,它只需一秒即可复查一遍。

轻型板状传真机 佛蒙特信息公司推出的这种轻型传真机为 9600 波特(每秒传输 9600 比特)。这是电脑在硬件方面的出色运用。这种小板似的传真机可以存储一个电话簿,可以发送、接收、定向传递文件,而且

接收到的文件既可以显示在屏幕上,也可打印在纸上。产品的主要特点是小型化,而且价格低廉,仅为 195 美元。

柯达的 PHOTO—CD 系统 柯达公司的这项新产品是它在图象数字化方面的最新进展。PHOTO—CD 系统可以将 35mm 的胶片或幻灯片上的彩色图像直接经过数字化以后输入电脑内存,或显示在屏幕上,或打印在纸上,或存入激光磁盘。这项产品成功地实现了自然图象与电脑的数字图像之间的方便的直接的转换,而这一直就是人们的梦想,因为这样可以在电脑屏幕上看到电视里那种活生生的图像了。此外,柯达在这个系统中运用的数字化方式非常先进,以致一张磁盘可容纳彩色图像多达一百幅左右。

今天的美国电脑业竞争非常激烈。在新上市的 486 型计算机上,AMP 和 Intel 两家大公司展开了激烈角逐。由于 AMD 在上代产品 386 型机上有相当优势,1990 年它还占有美国 286 型市场的 52%,所以估计 Intel 将吸取上次的教训,下大力推销产品。在操作系统这个广阔领域里,传统 DOS、扩展 DOS、UNIX 系统和 OS/2 系统展开了一场较量。迄今为止,最大的赢家是扩展 DOS,90 年秋其市场占有率达到了 40%;而 OS/2 则由 88 年的 32%下降到 12%,以上情况都是在博览会上调查而得。

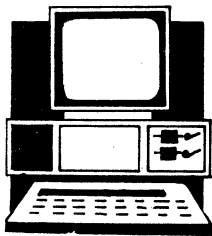
美国电脑业不仅内部竞争不断升级,还面临着外国公司的强大挑战。亚洲的南朝鲜以高士达为首的一批公司正决心把自己从 OEM 状态(原始产品生产者)中解脱出来,大力扩展海外销售,而且其产品极具竞争力。这次博览会上高士达公司的 386 型 16 位电脑及高分辨显示器都明显优于美国公司的产品。

COMDEX 电脑博览会是当今美国电脑业的一个缩影和窗口,它完整地显示了美国电脑业的特点,那就是:产品更替速度快、新颖独到的构思和你死我活的竞争。

表 1 IIXL 硬卡与 AST 硬盘读写速度比较

系统内存 缓冲区 (KB)	0	128	256
AST 硬盘	80.9 秒	77.1	53.9
IIXL 硬卡	62.8	52.1	42.2
提高速度百分比	22.3%	32.5%	21.6%

在内存缓冲区的三种不同情况下,在读写同一个程序的速度上,IIXL 硬卡都明显占有优势。



PC 用户

获取 FOX 文件中密码的两种简单方法

石油天然气总公司三机厂研究所(431734) 周日初

多数用户乐于使用 FoxBASE 来替代 dBASE 的原因之一就在于 FoxBASE 具有一定程度的保密性,可将 PRG 文件经 Foxpcomp. exe 编译后生成后缀为 FOX 的文件,它用系统命令 TYPE 是看不懂其含义的。因此在编写密码程序时,往往将其编译生 FOX 文件,为了防止别人从源程序中获得密码就将原 PRG 文件删除掉。但如果忘了密码或想破译别人的 FOX 文件中的密码怎么办呢? 下面我介绍两种简单易行的办法:

一、增加一中断处理程序

增加中断处理程序的目的就是改变 SET 命令的设置,将原来的 OFF 状态改为 ON,以便看到文件的内容及其执行过程,具体步骤如下:

1. 进入 FoxBASE,建一中断处理程序 ESCERR. PRG,内容见程序 1。

2. 敲入 ON ESCA DO ESCERR 命令

3. 装好打印纸,接通打印机

4. 执行密码程序,并按<ESC>键。

5. 分析打印纸上打出的程序,判断其密码。

举例如下:

设有一密码程序,其源程序见程序 2,将其中的第四句改为 SET ESCA ON(因为这种方法只适用于将 ESCAPE 设为 ON 的情况)。

按以上五步执行后,打印纸上便记下密码程序运行时的每一条指令,当读到 IF A#“1”时便知道 1 为密码,然后重新执行密码程序,当提示输出密码时,敲入 1,程序便正常结束,证明密码正确。密码错时会接连响三次铃,这时要继续分析打印结果,有的需要经过一定的转换才能获得真正的密码。这种增加中断处理程序获得密码的方法只适合于将 ESCAPE 置为 ON 的情况,当为 OFF 时则要采取第二种方法。

二、借助 DEBUG 修改 FOX 文件

要获得密码,就要了解源程序。这里所说的获得源程序的方法就是将 FOX 文件中有关 SET 命令置为 OFF 的地方改为 ON 状态,以便能看到执行的每一步,FOX 文件中这些 SET 命令是看不见的,但我们可用 DEBUG 得到它们编译后的目标码,详见表 1。例如:SET TALK OFF 编译后产生的目标码为 04 47 6A 51 FE,SET TALK ON 的目标码为:04 47 6A 52 FE,若要将 SET TALK OFF 改为 SET TALK ON,则用 DEBUG 的 S 命令先搜索一下代码 04 47 6A 51 FE,搜到后将对应的 51 改为 52 即可,再用 W 命令写回原文件,然后在 FoxBASE 状态下执行。现举例如下:

密码程序见程序 2,经编译后产生的 FOX 文件在 DEBUG 状态下显示的结果见表 2,查看表 1 可知:只要将 120~13F 间的 4 个 51 用 E 命令改为 52,便将程

表 1

设置项	设置为 ON/OFF 时的目标码	
	ON	OFF
ALTE	04 47 16 52 FE	04 47 16 51 FE
CARR	04 47 1F 52 FE	04 47 1F 51 FE
CLEA	04 47 79 52 FE	04 47 79 51 FE
CONF	04 47 27 52 FE	04 47 27 51 FE
DEBU	04 47 2C 52 FE	04 47 2C 51 FE
DELI	04 47 2F 52 FE	04 47 2F 51 FE
ECHO	04 47 33 52 FE	04 47 33 51 FE
EXAC	04 47 37 52 FE	04 47 37 51 FE
FIEL	04 47 3B 52 FE	04 47 3B 51 FE
HEAD	04 47 43 52 FE	04 47 43 51 FE
HIST	04 47 BD 52 FE	04 47 BD 51 FE
MENU	04 47 9B 52 FE	04 47 88 51 FE
SAFE	04 47 8B 52 FE	04 47 8B 51 FE
STAT	04 47 65 52 FE	04 47 65 51 FE
TALK	04 47 6A 52 FE	04 47 6A 51 FE
BELL	04 47 1B 52 FE	04 47 1B 51 FE
CENT	04 47 90 52 FE	04 47 90 51 FE
COLO	04 47 25 52 FE	04 47 25 51 FE
CONS	04 47 28 52 FE	04 47 28 51 FE
DELE	04 47 2E 52 FE	04 47 2E 51 FE
DOHI	04 47 C4 52 FE	04 47 C4 51 FE
ESCA	04 47 36 52 FE	04 47 36 51 FE
EXCL	04 47 CE 52 FE	04 47 CE 51 FE
FIXE	04 47 86 52 FE	04 47 86 51 FE
HELP	04 47 97 52 FE	04 47 87 51 FE
INTE	04 47 46 52 FE	04 47 46 51 FE
PRINT	04 47 57 52 FE	04 47 57 51 FE
SCOR	04 47 8C 52 FE	04 47 8C 51 FE
STEP	04 47 66 52 FE	04 47 66 51 FE
UNIQ	04 47 8D 52 FE	04 47 8D 51 FE

注:以上数值均为 16 进制数

序 2 中最前面的四条 SET 命令置为 ON 状态,然后用 DEBUG 的 W 命令写回到 FOX 文件中,重新执行,仿照第一种方法,把执行结果打在纸上(先执行 SET DEVI TO PRIN 和 SET PRIN ON),再分析密码。

用这两种方法,不仅可以方便地找到密码,而且还可以获得 FOX 文件对应的 PRG 程序。

表 2

```

-r
AX=0000 BS=0000 CX=00D9 DX=0000 SP=FFEE BP=0000 SI=0000 DI=0000
DS=3000 ES=3000 SS=3000 IP=0100      MV UP EI PL NZ NA PO NC
3000:0100 FB          STI
-d100 1d9
3000:0100 FB 2B 02 00 00 00 00 00-00 00 00 00 00 00 00 00 00  .+.
3000:0110 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 00  .....
3000:0120 00 00 9F 00 04 47 6A 51-FE 04 47 65 51 FE 04 47  ....GJQ..GEQ..G
3000:0130 33 51 FE 04 47 36 51 FE-0B 54 F7 00 00 10 FC F8  3Q..G61..T....
3000:0140 01 01 FD FE 1B 04 FC F8-02 0A FD 07 FC F8 02 14  .....
3000:0150 FD 5F FC FB 09 CA E4 C8-EB C3 DC C2 EB 3A FD FE  .-.....
3000:0160 10 18 73 FC F7 00 00 F8-01 04 0D FD F9 05 9A 00  ..s.....
3000:0170 FE 04 47 28 51 FE 06 05-6C F7 01 00 FE 04 47 28  ..G(Q..1....G(
3000:0180 52 FE 0F 25 FC F7 01 00-FB 01 31 0F FD F9 05 7D  R..%.1....}
3000:0190 00 FE 0E 02 FC 78 78 F8-01 07 22 F8 01 03 5C FD  ....XX..."..
3000:01A0 FE 06 1B F9 05 87 00 FE-02 21 FE 02 1E FE 0F 54  ....1....T
3000:01B0 F7 00 00 10 FC F7 00 00-F8 01 01 06 FD FE 02 1D  .....
3000:01C0 FE 01 55 02 00 4B 4B 00-00 00 00 00 00 00 41  ..U..KK.....A
3000:01D0 00 00 00 00 00 00 00 00-00 C4  .....

```

对全屏幕编辑命令 BROWSE 做一点补充

广州中山二路 48 号之二 1702 室 宋开胜

现已出版了多种 dBASE III 书籍,但纵观各种书籍,对全屏幕编辑命令 BROWSE 的介绍都是一种模式 BROWSE[FIELDS<字段名>]。在使用中按照这一模式进行全屏幕输入、修改时经常会遇到一些操作失误,尤其是那些不懂计算机操作的人,对于多个字段宽度超过 80 个字符的数据库,进行全屏幕输入、修改操作时,往往不知道利用 CTRL+HOME 来锁定字段,再利用 CTRL+(←或→)调出屏幕之外的字段。对于某一个字段的输入、修改操作,如不利用 CTRL+HOME 来选择字段,操作失误也时有发生。

本人在实际应用中发现它有多种模式,在以往各种书籍中都没有介绍过,现补充如下:

①BROWSE[FIELDS<字段名>]

②BROWSE [FIELDS<字段名>][LOCK #] # :是数字(1,2 ...)

③BROWSE [FIELDS<字段名>][FREEZE<字段名>]]

程序 1

```

* ESCERR. PRG
SET TALK ON
SET DEBUG ON
SET STAT ON
SET CONS ON
SET ECHO ON
SET PRIN ON
SET DEVI TO ON

```

程序 2

```

SET TALK OFF
SET STAT OFF
SET CEHO OFF
SET ESCA OFF
KK=1
@ 10,20 SAY' 输入密码:'
DO WHILE KK<4
SET CONS OFF
ACCE TO A
SET CONS ON
IF A<>'1'
? REPL(CHR(7),3)
ELSE
EXIT
ENDI
KK=KK+1
ENDD

```

④BROWSE [FIELDS<字段名>][LOCK #FREEZE<字段名>]]

利用以上几种模式,可以代替 CTRL+HOME 来锁定字段和字段选择。

功能:①是通用型书籍中已有说明。

②从某一字段开始锁定(#:1,2...)用 CTRL+←(或→)左右控制移动字段。

③固定对其中某个字段输入、修改并能保护其他字段误操作。

④同时使用②、③的功能。

附上使用范例

数据库:DATA.DBF

字段	字段名	类型	宽度	小数
1	A1	Character	8	
2	A2	Numeric	6	2
3	A3	Numeric	6	2
4	A4	Numeric	6	2
5	A5	Numeric	6	2
6	A6	Numeric	6	2

USE DATA

BROWSE FIELDS A1,A2,A3,A4,A5,A6 LOCK 2

字段 A1 和 A2 被锁住,用 Ctrl+←(或→)可以控制 A3,A4,A5,A6,左右移动

BROWSE FIELDS A1,A2,A3,A4,A5,A6FREEZE

A3

固定对字段 A3 输入、修改,保护其他字段误操作
BROWSE FIELDS A1, A2, A3, A4, A5, A6 LOCK 2
FREEZE A5

锁住字段 A1 和 A2,用 Ctrl+←(或→)可以控制 A3,
A4,A5,A6 左右移动,只对字段 A5 输入、修改,保护其
他字段误操作

感染 OMICRON 病毒文件的消毒方法

上海同济大学 8736(200092) 刘东祥

计算机病毒作为一类具有隐蔽性、传染性和破坏性的特殊程序,日益严重地干扰着用户的正常工作。以往的病毒按其寄生的部位,可分为引导区和执行文件病毒两大类。现在市场上流行的 ANTI、SCAN 或 TNTVIRUS 都可以消除这两类病毒的绝大多数。但是最近我国电脑软件中又发现了一种病毒,它区别于以往的单宿主病毒,可以同时寄生在硬盘主引导区和执行文件中,所以传染速度极为疯狂。经破译后,发现该病毒的显示信息有一段文字很奇特,为 OMICRON by PsychoBlast(OMICRON 为希腊语中的第十五个字母),所以常称之为 OMICRON 病毒。

感染这种病毒之后的文件将增加 2153 个字节。用 ANTI、SCAN 或 TNTVIRUS 都无法将它消除。分析其病毒程序,可见作者的编程技巧相当强,最大特点是它利用时钟产生的随机数作为密码钥,所以表面上看,不同时间感染所追加的病毒程序也不相同。

OMICRON 病毒发病时间是在每月 2 日上午 10 点,屏幕上的内容将上下左右相互对换。该病毒并没有多么深重恶劣的破坏功能,属良性病毒一类。

下面就详细介绍对 OMICRON 病毒的解毒方法。

首先用一张干净的系统盘启动机器,盘上要有 DEBUG.COM 文件。

COM 文件的解毒方法很简单,运行 DEBUG 调入染毒 COM 文件,程序的第一个语句为 JMP 语句,跳转地址记为 Addr,找出 Addr+9 处的一个字节的值,记为 KEY,KEY 即密码钥。原正常文件的长度为 Addr-821-100(HEX),然后再查出 Addr-821+BF(HEX)起始连续三个字节,记为 B1、B2、B3 将 B1、B2、B3。分别加上 KEY 之后依次放到 COM 文件开头,修改 CX、BX 为原正常 COM 文件的长度,记盘退出。

EXE 文件的消毒工作稍为麻烦一些,可分为以下四步:

(一)运行 DEBUG,调入 EXE 文件,用 G 功能运行到病毒程序中第一个 LOOP 语句后的 JMP 语句,再用 T 功能运行完随后的 CALL 语句、POP SI、SUB SI,0113 后止,查看 SI,用笔记录 CS:SI+C2 始的六个字节。注意,SI+C2、SI+C3 保存原 IP,SI+C4、SI+C5 与原 CS 有关,而 SI+C6、SI+C7 与原 SS 有关,退出 DEBUG。

(二)改 EXE 文件为以 BIN 为后缀的文件。

(三)再进入 DEBUG 调改名后的文件,将病毒文件长度减去 2153(DEC)个字节后,修改 CX、BX 为相应的值,同时修改 CS:102—CS:105 四个字节,如病毒文件长度为 27461,减去 2153 后为 25308,25308 个字节将占 50(HEX:32)个扇区,但最后的一个扇区只用了 220(HEX:DC)个字节,所以在 CS:102 处写 DC,CS:103 处写 00,CS:104 处写 32,CS:105 处写上 00,另外在 CS:10E 上填 [SI+C6]-10(HEX),CS:10F 上填 [SI+C7],CS:114 上写 [SI+C2],CS:115 上填 [SI+C3],CS:116 上写 [SI+C4]-10(HEX),CS:117 上填写 [SI+C5],之后记盘退出。

(四)将 BIN 文件恢复为 EXE 可执行文件。

值得注意的是,病毒文件可能是经过多次感染的,所以要经检查无毒后方可运行,若依然有毒,要做多次消毒工作。

如果机器带有硬盘,很有必要检查硬盘主引导区是否已经感染。如果在此之前运行过带毒程序,则主引导区肯定被感染。只要用硬盘主引导区的备份将之覆盖即可。

具体过程如下:

A:\>DEBUG C.INI.(C.INI 为硬盘主引导区备份)

```
-A400
3212:0400 PUSH CS
3212:0401 POP ES
3212:0402 MOV AX,0301
3212:0405 MOV BX,0100
3212:0408 MOV DX,0080
3212:040B MOV CX,0001
3212:040E INT 13
3212:0410 INT 20
3212:0412
-RIP
IP 0100
:400
-G
-Q
```

我们发现 OMICRON 病毒可以破坏硬盘分区表,因此不能视为良性病毒。美国 CENTRAL POIN 公司的抗病毒软件 CPAV 可以消除 OMICRON 病毒,关于 CPAV 的编者使用本刊将在近期另文介绍。

Color 400卡屏幕彩色图形的硬拷贝

东北工学院319信箱(110006) 张 辉 李东升

Color400彩色图形卡(又称 CGE400)是为 IBM PC 显示适配器的软硬件最大程度相兼容而设计的。它可提供640×400像素的具有16种色彩的高分辨率彩色图形,目前已在我国广泛应用。可令人遗憾的是,用Color400卡显示的漂亮图形,只能在屏幕上显示,而不能从打印机上打印出来。为了解决这一问题,我们以国内广泛使用的 M2024和 M1724打印机为例,介绍了Color400卡彩色图形的16种灰度硬拷贝的编程方法。

一、屏幕像素色彩的灰度转换

Color400卡的640×400高分辨率彩色图形模式可以显示16种颜色的像素。像素彩色值的获取可通过BIOS的INT10H的13号功能来实现。获取像素彩色值的8088汇编语言程序如下:

```
MOV AH,13
MOV DX,ROW
MOV CX,COL
INT 10H
```

程序中 ROW 和 COL 分别为像素的行和列,像素的彩色值在 INT 10H 返回时存放在 AL 寄存器中,其值为0~15。

由于打印机只能打印黑白两种颜色,所以获得像素的彩色值后,还需将彩色值转换为相应灰度值的打印机输出数据。由于 M2024/M1724打印机的最大横向打印列数为2716,所以我们设计了3×4点阵中点的排列,可获得16种灰度。为了数据的处理方便,我们用3个字节数据来表示一个屏幕像素的灰度值输出数据。如深灰像素的灰度图和输出数据如图1所示。像素彩色与打印机灰度输出数据的对照见表1。

表1 像素彩色与打印机灰度输出数据

彩色值	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
打印数据	0F060400	0E060602	0F0E020D	060400	0F0D0104	0S0F0909	0F0F0F0F	0F0F0F0F	0F0F0F0F	0F0F0F0F	0F0F0F0F	0F0F0F0F	0F0F0F0F	0F0F0F0F	0F0F0F0F	0F0F0F0F

图1

灰度打印数据

二、打印机的设置

M2024/M1724打印机为24针点阵式打印机,故每一纵列打印数据按3个字节传送。当送出的图形打印数据中的某一位为“1”时,对应的打印针动作,为“0”时打印针不动作。为了输出图形必须首先设置好相应的行距,然后将打印机置成图形模式。打印图形时可选择24针同时打印或只用其中的8针打印,选择24针

打印时,由于易使打印头产生高温影响打印速度。所以我们选用其中的8针进行打印,即送一个字节的图形数据后,紧接着送2个“0”字节。

确定了使用的打印针数后,就可以设置打印行距。使用8针时,打印机的行距设置为6/120英寸比较合适。这样设置后打印机打印出的图形既不断开也不重迭。其十六进制控制码为:1B 4A 06。24针打印机在图形模式下每行输出的一列有24个点,即每3个字节的图形打印数据对应某打印行的一列。图形打印数据与打印针的对应关系及控制码如图2所示。

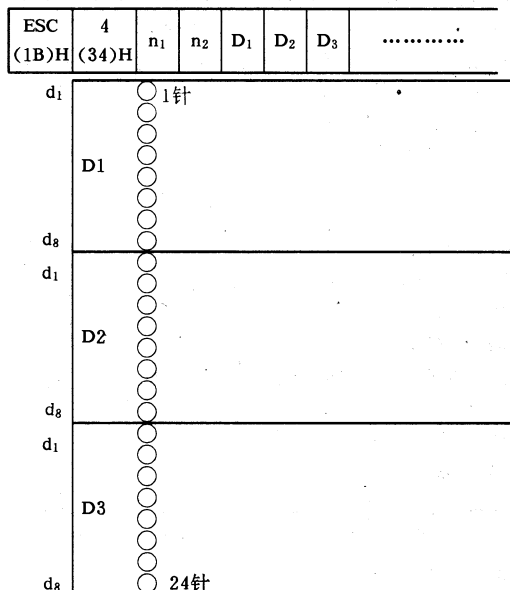


图2 图形打印数据与打印针对应关系

对于打印640×400的屏幕图形,如果用3×4点阵对应于屏幕上的一个像素,则打印机打印的列数为3×640=1920,对应的打印机控制码为:

1B 34 07 80 D1 D2.....D5760,其中 D1~D5760为打印图形数据。第二个代码34表示双向打印,若改为47则为单向打印。打印机收到以上5764个数据后仍然处于等待状态,还必须再向打印机送回车(0DH)和换行(0AH)的控制码,打印机才打印缓冲区中的数据。

三、程序实例

根据以上方法,我们用8088汇编语言编制了 Color400卡彩色屏幕图形拷贝程序 PRTSCR.COM,其源程序清单如附录所示。

在绘制图形前首先执行 PRTSCR.COM, 则该程序将常驻内存, 直至计算机重新启动。屏幕图形绘制完毕后, 按下拷屏幕键(prtsc 键)即可在打印机上拷贝出具有16种灰度的精美图形。拷贝的图形例子如图3所示。本方法也适于其它彩显卡(如 EGA,VGA)的彩色图形拷贝。



图3 640×400屏幕1/4图形拷贝

附录: 屏幕拷贝程序 PRTSCR.COM 源程序清单

```
CODE SEGMENT PARA 'CODE'
ORG 100H
START PROC FAR
ASSUME CS:CODE
ASSUME DS:CODE,ES:CODE
JMP CCCC
```

; 拷贝屏幕中断服务程序

```
PRTS PROC NEAR
STI
PUSH DS
PUSH ES
PUSH DI
PUSH SI
PUSH BX
PUSH CX
PUSH DX
PUSH AX
MOV AX,CS
MOV DS,AX
MOV ES,AX
```

; 查找打印机号

```
XOR DX,DX
Z: MOV AH,2
INT 17H
AND AH,10H
CMP AH,10H
JZ ZZ
```

```
INC DX
JMP Z
; 确定屏幕状态,如不是640x400图形模式则退出
ZZ: MOV SI,DX
MOV AH,15
INT 10H
CMP AL,42H
JNZ EXIT
; 置打印机行距
MOV AL,0AH
CALL PRI
MOV AL,0DH
CALL PRI
MOV AL,1BH
CALL PRI
MOV AL,4AH
CALL,PRI
MOV AL,6
CALL PRI
XOR DX,DX
XOR AX,AX
; 每次打印屏幕上的二行图形
P42: CALL PRCON
CALL PR640
CMP AX,200
JZ P41
ADD DX,2
INC AL
JMP P42
; 完成屏幕拷贝中断服务返回
P41: MOV AL,18H
CALL PRI
CALL CRLF
MOV AL,0FH
CALL PRI
EXIT: MOV AL,20H
MOV DX,20H
OUT DX,AL
POP AX
POP DX
POP CX
POP BX
POP SI
POP DI
POP ES
POP DS
IRET
PRTS ENDP
; 输出两行屏幕象素至打印机子程序
PR640 PROC NEAR
PUSH CX
PUSH AX
XOR CX,CX
P2: CALL READ2
CMP CX,639
```

```

        JZ P1
        INC CX
        JMP P2
P1:     CALL CRLF
        POP AX
        POP CX
        RET
PR640  ENDP
; 输出回车和换行命令子程序
CRLF  PROC NEAR
        PUSH AX
        MOV AL,0AH
        CALL PRI
        MOV AL,0DH
        CALL PRI
        POP AX
        RET
CRLF  ENDP
; 置打印机图形状态子程序
PRCON  PROC NEAR
        PUSH AX
        MOV AL,1BH
        CALL PRI
        MOV AL,47H
        CALL PRI
        MOV AL,7
        CALL PRI
        MOV AL,80H
        CALL PRI
        POP AX
        RET
PRCON  ENDP
; 向打印机输出 AL 中数据子程序
PRI  PROC NEAR
        PUSH AX
        PUSH DX
        MOV AH,0
        MOV DX,SI
        INT 17H
        POP DX
        POP AX
        RET
PRI  ENDP
; 象素彩色值灰度转换子程序
READ1  PROC NEAR
        PUSH CX
        PUSH DX
        PUSH SI
        PUSH BX
        MOV AH,13
        INT 10H
        MOV AH,0
        PUSH BX
        MOV BX,AX
        SHL BX,1

```

```

        ADD AX,BX
        POP BX
        LEA SI,GRAY
        ADD SI,AX
        MOV CX,3
        REP MOVSB
        POP BX
        POP SI
        POP DX
        POP CX
        RET
READ1  ENDP
; 灰度数据
GRAY  DB 0FH,0FH,0FH
        DB 06H,0DH,06H
        DB 04H,01H,04H
        DB 00H,04H,02H
        DB 0EH,05H,0BH
        DB 06H,0FH,06H
        DB 06H,09H,06H
        DB 02H,09H,06H
        DB 0FH,0FH,07H
        DB 0EH,0FH,07H
        DB 02H,0FH,06H
        DB 0DH,0BH,0DH
        DB 06H,0FH,07H
        DB 04H,01H,04H
        DB 00H,04H,00H
        DB 00H,00H,00H
READ2  PROC NEAR
        PUSH DX
        PUSH CX
        MOV BX,2
        LEA DI,BUF
OOP1:  CALL READ1
        INC DX
        DEC BX
        JNZ OOP1
        MOV CX,3
        MOV BX,OFFSET BUF
OOP2:  PUSH CX
        PUSH BX
        MOV AL,[BX]
        ADD BX,3
        MOV AH,[BX]
        SHL AH,1
        SHL AH,1
        SHL AH,1
        SHL AH,1
        PUSH DX
        MOV DL,AH
        MOV DH,0
        MOV AH,0
        ADD AX,DX
        CALL PRI

```


BX: * * * *

; 0000 此例因程序没有跨段, 所以为 0000H

—NCCCC.COM

—W * * * * 此处的 * * * * 为起始地址

不过, 这时复制出来的 CCCC.COM 还不能使用, 还必须做一些改动才行。主要有以下两点:

a. 原 CCCC.EXE 中的重定向语句

MOV DS, * * * *

这里可以改为:

PUSH CS

POP DS

b. 因 CCCC.COM 文件将是 100H 处加载而原 CCCC.EXE 是从 0H 处开始加载的, 故两个文件中的 DS 值相差 10H, 所以在 CCCC.COM 中的设置 INT 10H, 10H, 17H, 1DH 入口指令中的段值必须作加 10H 的修改。具体修改的步骤, 这里就不赘述了。

完成了以上工作后, 就算是得到了可以自由使用和拷贝的 CCCC.COM 了, 从而也就完成了对 CCCC.EXE 的解密工作。

2. 解密 STEP2.EXE

STEP2.EXE 实际上就是一个管理和传送汉字字模库的模块。要对 STEP2.EXE 进行解密, 首先必须知道压缩汉字库 CCBios 系统是如何进行汉字字模传送的。这一点可以在 INT 10H 的第 18 号功能块中找到, 请看 INT 10H 中第 18 号功能块的一段程序。

AND DX, 7F7F

PUSH AX

MOV AX, BX

INT 6EH

MOV DX, AX

POP AX

RET

由此可知, 取字模的工作是通过 INT 6EH 来完成的, 这与 CCCC.EXE 同理也就算找到了其解密的突破口了。

进入 DEBUG, 利用 DOS 的 35H 号系统功能调用找到 INT 6EH 的入口, 然后仿照解密 CCCC.EXE 的方法将 STEP2.EXE 复制出来并取名为 STEP2.COM 即完成了其解密工作。

三、关于这种解密方法的两点说明

1. 适用范围

了解了本解密方法就不难知道, 只要经过加密处理的软件在它加载后是常驻内存的, 一般都可很好地用这种方法解密, 只是寻找其在内存中的驻留位置和长度需要解密者经过分析后获得。这里介绍的只是众多寻找方法中的一种, 不同的软件因其功能不同, 用这种方法并不一定能找得到, 这时就需要解密者用别的方法去寻找, 这里笔者就不一一介绍了。对于那些不是常驻内存的软件, 只要它没有使用动态复盖技术的话, 这种方法也同样适用, 因为任何软件在它执行完后且没有在同一位置加载过别的软件时, 它在内存中总是会有其影子存在的, 可设法找到这个影子, 再用上面介绍的方法将其完整地复制出来。

2. 本解密方法的特点

本解密方法具有以下特点:

----不用了解软件是用何种方式进行加密的。

----不用象其它解密方法那样需费许多时间和精力去寻找加密的密钥, 并跟踪其加密过程并逐渐脱密。

----其原理和实现方法简单且易掌握。

----安全、可靠, 不会破坏原软件。

三维图形显示的遮掩技术及实现

武汉水利电力学院(430072) 季军杰

二元函数 $Z=f(x,y)$ 所呈现的图形一般是曲面形状, 有的在图形上表现出突出的几何特征。这些曲面在生活和生产实践中, 在数学、物理和工程技术中都是常见的, 熟悉它们的图形很重要。

为了获得较为真实的 $Z=f(x,y)$ 图象, 我们须避免描绘那些被隐藏了的曲面段, 这就要用到计算机图形学中的三维图形显示遮掩技术。

·遮掩技术:

遮掩技术的实现基于以下两个前提: (1) 前景线条

(Z 轴正向) 在背景线条之前描绘, (2) 若一线条或其一部分落入先前已描绘线条所在的区域内, 它就被遮掩 (或称为消隐)。消隐可通过生成一个可视最大函数和一个可视最小函数获得, 这样, 凡落入以上界限的线条部分即被遮掩。下面的隐藏线消隐程序可实现此技术。

·编程要点:

本程序以描绘函数 $Z=7e^{-0.1(x^2+y^2)}$ 为例, 产生一线状结构曲线映象以表示该曲面 (源程序附后)。

程序开头的初始化工作是建立视区和屏幕参数以

及待描绘的函数。定义两个数组 YMAX(I) 和 YMIN(I) 用于对每个水平屏幕坐标进行确认：(1) 所遇到的最大垂直屏幕坐标存入 YMAX(I)，(2) 所遇到的最小垂直屏幕坐标存入 YMIN(I)。

最初，每个 YMAX(I) 的值为 0，每个 YMIN(I) 的值为 199 (屏幕最大扫描线)，一旦屏幕坐标点 (x_s, y_s) 已描绘过，YMAX(x_s) 值就重置，一旦一个 (x_s, y_s) 被考虑描绘时， y_s 就与 YMAX(x_s) 和 YMIN(x_s) 比较，若 y_s 在 YMAX(x_s) 和 YMIN(x_s) 之间，点 (x_s, y_s) 就被消隐，否则就描出。

运行本程序，隐藏线消除后的三维图象更加美观。

该程序用 BASICA 写成，下面对程序中所用参数进行说明。

D: 视点 P 到物体所在坐标系原点 O 之距离。

(THETA, PHI): 视者观察物体的方位。其中 PHI 为 OP 与 Z 轴正向之夹角，THETA 为 X 轴正向与 OP 在 XY 平面上的投影的夹角。

VD: 视点和显示屏 (投影平面) 间的距离。

S: 屏幕中心大小值，图象在 $2S \times 2S$ 正方形区域内显示。

SCF: 屏幕比例调整因子，以反映图象真实纵横比。

TX, TY: 把视口坐标变换为屏幕坐标，并调整图象显示位置。

以上参数均可按照实际应用情况进行调节。

参数间关系如下：

① 改变 THETA 和 PHI 的值可从不同角度观察物体。

② 改变 D 值可从较远处或较近处观察。

③ 当视点 (D, THETA, PHI) 固定时，VD 的变化将影响显示屏上物体映象的大小。若显示屏与观察者较近 (VD 减小)，则投影映象将减小。

④ D 和 VD 同时控制着屏幕映象大小，增大 D 值可减小透视效果，但映象变小，放大映象可增大 VD，而不再引入透视效果。

⑤ VD/S 比率的大小类似于照相调焦时从取景器中看到的图象效果。若比率小，类似于用短焦照相，产生广角镜头映象画面；若比率大，类似于长焦效果，生成如“望远摄影”的一幅有限图象。

⑥ 当 $D = VD$ 且 $D \rightarrow \infty$ 时，将产生正投影映象。

```
110 CLS;KEY OFF
120 SCREEN 2
130 DIM YMAX(640),YMIN(640)
140 GOSUB 190
150 GOSUB 250
160 GOSUB 280
170 END
190 TX=120;TY=100;PI=3.141593;SCF=2.4
200 D=30;VD=420;THETA=30;PHI=80;S=2
210 THETA=THETA*PI/180;PHI=PHI*PI/180
220 SN1=SIN(THETA);SN2=SIN(PHI);CN1=
    COS(THETA);CN2=COS(PHI)
```

```
230 RETURN
250 FOR I=1 TO 640;YMAX(I)=0;YMIN(I)=199;NEXT
    I
260 RETURN
280 DEF FNZ(X,Y)=7*EXP(-.1*(X*X+Y*Y))
290 FOR X=8 TO -8 STEP -1
300 F=0
310 FOR Y=-8 TO 8 STEP 5
320 Z=FNZ(X,Y)
330 GOSUB 400
340 GOSUB 470
350 GOSUB 530
360 NEXT Y
370 NEXT X
380 RETURN
400 XE=-X*SN1+Y*CN1
410 YE=-C*SN1*CN2-Y*SN1*CN2+Z*SN2
420 ZE=-X*SN2*CN1-Y*SN2*SN1-Z*CN2+D
430 RETURN
470 XS=(VD/S)*(XE/ZE)
480 YS=(VD/S)*(YE/ZE)
490 XS=XS+TX;XS=SCF*XS
500 YS=-YS+TY
510 RETURN
530 IF F=0 THEN 540 ELSE 570
540 F=1;F2=0
550 OX=XS;OY=YS
560 RETURN
570 DX=OX-XS;IF DX=0 THEN DX=1
580 DY=OY-YS
590 SLOPE=DY/DX
600 T1Y=OY
610 NX=INT(OX)+1
620 FOR T1X=NX TO XS
630 F1=1
640 T1Y=T1Y+SLOPE
650 IF T1X<0 OR T1X>639 THEN F1=0;F2=0;GOTO 700
660 IF T1Y<0 OR T1Y>199 THEN F1=0;F2=0
670 IF T1Y<=YMIN(T1X) THEN 730
680 IF T1Y>=YMAX(T1X) THEN 780
690 F2=0
700 NEXT T1X
710 OX=XS;OY=YS
720 RETURN
730 YMIN(T1X)=T1Y
740 IF F1=0 THEN 770
750 IF F2=0 THEN PSET(T1X,T1Y);F2=1
760 LINE-(T1X,T1Y)
770 IF T1Y<YMAX(T1X) THEN 700
780 YMAX(T1X)=T1Y
790 IF F1=0 THEN 700
800 IF F2=0 THEN PSET(T1X,T1Y);F2=1
810 LINE-(T1X,T1Y)
820 GOTO 700
```


BASIC 超长随机文件的处理方法

上海前进实业公司中心电脑室(202179) 朱建平

在 BASIC 程序设计中,磁盘数据文件的输入和输出分为二类:顺序文件和随机文件。由于随机文件具有占用磁盘空间少,数据可以随机地存取等特点而被广泛采用。一般而言,随机文件的使用分为下面几步:

1. 以随机存取方式打开文件,并指定记录长度,缺省值为128个字节。
2. 对于要存取的随机文件的变量用 FIELD 语句在随机缓冲区中为其分配空间。
3. 用 LSET 和 RSET 将数据送入随机缓冲区。
4. 用 PUT、GET 语句将数据从缓冲区写入磁盘或从磁盘送入缓冲区。

由此可见,随机文件的建立与使用需要设定记录长度,并且要对记录缓冲区进行定义。现在的问题是,当一个随机文件的记录长度超过128个字节,且记录中的数据项较多,用一个 FIELD 语句定义不完时,如何用变通的方法实现随机文件的正确存取。笔者在软件开发中,就曾碰到过这个问题,一个随机文件的记录长度为230个字节。每10个字节分配一个区段,需要用二个 FIELD 语句对记录中的每个数据项进行定义。对这样一个超长记录的随机文件如何进行正确的存取呢?经查阅有关资料,笔者找到了解决这个问题的简单方法。

一、随机文件超长记录的长度设定

我们知道,IBM PC BASIC 命令的完整格式为:

```
BASIC[A][filespec][F:files][S:bsize][C:combuffer][M:max workspace]
```

其中:可选项/S:bsize 参数是设置随机文件的记录长度,OPEN 语句中表示记录长度的参数不得超过该值,其缺省值为128个字节,可指定的记录长度的最大值为32767。

由此可见,对于一个随机文件的长度超过128个字节时,可在 BASIC 启动时,设置可选项参数/S:bsize 来完成超长记录的定义。笔者碰到的问题就是采用下面的命令来解决的。

```
A>BASIC 文件名[.BAS]/S:230
```

二、随机文件的续行方法

BASIC 语言规定,一个语句行的最大长度为256个

字节。对于一个超长记录的随机文件,可能会出现用一个 FIELD 语句定义不完的现象,如何对每个数据项进行正确定义,即 FIELD 语句的续行定义呢?笔者发现:当一个 FIELD 语句定义不完时,可在后一个 FIELD 语句的开头设置一个虚拟变量,其长度为前一个 FIELD 语句定义项的总长度,继后再定义余下的数据项。采用这种方法,即可完成对 FIELD 语句的续行定义。假如在后一个 FIELD 语句中不设置虚拟变量,则在随机文件的存取中不保证数据的正确性。以笔者的例子为例,其程序段为:

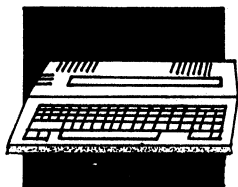
```
100 OPEN "JBADATA.DTA" AS #1 LEN=230
110 FIELD #1,10 AS CH$(1),10 AS CH$(2),.....,10 AS
    CH$(13)
120 FIELD #1,130 AS CH$(1),10 AS CH$(14),.....,10
    AS CH$(23)
```

程序中,120行语句中的130AS CH\$(1)是一个虚拟变量,其长度为上一个 FIELD 语句定义的数据项总长度。虚拟变量,在这里仅起到一个虚拟前一条 FIELD 语句定义的数据长度的作用,亦即仅起到几个 FIELD 语句之间的衔接作用。在这个例子中,笔者采用的虚拟变量名 CH\$(1)与上一个 FIELD 语句的缓冲数据项名同名,这并不影响随机文件的正确存取。当然,虚拟变量名亦可用其他变量名,如:A\$,B\$等,但是,虚拟变量的长度则必须等于上一条 FIELD 语句所定义的数据项的总长度。上面的例子中,假如后一个 FIELD 语句中不设置虚拟变量的话,则记录在随机文件中的数据为后一个 FIELD 语句所定义的数据项。

为了程序设计方便,也可采用循环语句来简化上述程序,以达到同样的目的。简化后的程序为:

```
100 OPEN "JBADATA.DTA" AS #1 LEN=230
110 FOR I=1 TO 23
120 FIELD #1,10*(I-1) AS CH$(0),10 AS CH$(I)
130 NEXT I
```

上述方法在 IBM PC/XT 机上通过。



打印机的定位和格式控制

西安石油仪器研究所(710054) 栾傲发

学习机之友

在数据输出时,我们往往要求在打印机上的格式整齐、美观。APPLE— I 和中华学习机的 BASIC 语言中没有提供很方便的格式输出。常规在 40 列范围内用 TAB 函数作为定位控制。超过 40 列则用 SPC 函数作为定位控制,但事先必须估计输出数据的域宽。尽管如此仍给编程时的数据输出带来了很多不便。下面介绍一种既方便、又灵活的打印机行宽和格式控制编程方法。

在 APPLE 和中华学习机中有一个可供用户调用的汇编子程序。调用地址为 49568(如果打印机驱动卡插在 1 号槽)。它能够像 FORTRAN 语言中的 I、F、E 输出格式那样来使用,但缺点是不能作为数据输入。在调用这个子程序之前先定义打印机的宽度。若输出宽度在 1—79 列内则用 POKE1657,80 语句定义,打印机输出的字符为 79 列标准字符。若输出宽度在 1—135 列内则用 POKE1657,135 语句定义,打印机输出的字符为 135 列缩小字符,再按如下介绍的两种调用形式来编程就可实现打印机的定位和格式输出。

一、一般形式

CALL SUB;M1,V1;F1,M2 V2;F2,……Mn,Vn;
Fn,CHR\$(表达式);

其中:

SUB——变量名或直接数 49568

M1~Mn——字符串,字符串变量,字符串数组元素。

V1~Vn——整型变量、浮点变量

F1~Fn——格式控制字、有如下几种形式

1. I_n 整型输出格式,n 为域宽。
2. Fw.d 浮点输出格式,w 为域宽,d 为小数位数。
3. E_n 指数输出格式,n 为小数位数。

CHR\$(表达式)——打印机控制函数,其中:

10 换行

11 垂直定位

12 跳页

13 回车

* 其它控制函数请参阅有关打印机说明书。

二、特殊形式

CALL SUB;CHR\$(表达式);[CHR\$(表达式);]
……[CHR\$(表达式);];

其中

SUB——同上

CHR\$(表达式)₁……CHR\$(表达式)_n——参阅

表一

这种调用形式定义打印机的各种功能。它往往与

一般形式共同使用以达到专门的输出格式。

一般形式调用实例

```
10 A%=254;B=3.141592;C=69.435
20 POKE 1657,80
30 WR=49568
40 PR#1
50 CALL WR;CHR$(13);
60 CALL WR;"A=",A%;I4,CHR$(13);
70 CALL WR;"B=",B;F8.6,CHR$(13);
80 CALL WR;"C=",C;E4,CHR$(13);
90 END
```

]RUN

A= 254

B= 3.141592

C= 6.943E+01

特殊形式调用实例

调用内部水平制表符

```
10 POKE 1657,80
20 WR=49568
30 PR#1
40 PRINT"0123456789012345678901234567890"
50 FOR I=0 TO 2
60 CALL WR;CHR$(9);"TAB";
70 NEXT
80 END
```

]RUN

]0123456789012345678901234567890

TAB TAB B TAB

调用垂直制表符

```
10 POKE 1657,80
20 WR=49568
30 PR#1
40 FOR I=1 TO 4;READ X%(I);NEXT
50 CALL WR;CHR$(27);"B";CHR$(1);CHR$(3);CHR$(6);CHR$(10);CHR$(0);
60 CALL WR;CHR$(11);"TAB";X%(1);I6;"stline";
70 CALL WR;CHR$(11);"TAB";X%(2);I6;"rdline";
80 CALL WR;CHR$(11);"TAB";X%(3);I6;"thline";
90 CALL WR;CHR$(11);"TAB";X%(4);I6;"thline";
100 END
110 DATA 1,3,6,10
]RUN
TAB 1stline
TAB 3rdline
TAB 6thline
TAB 10thline
```


建立右边列的起始位置(Right Margin)

```
10 POKE 1657,80
20 WR=49568
30 PR#1
40 A$="0123456789012345678"
50 CALL WR,CHR$(13);
60 CALL WR,CHR$(27);"Q";CHR$(12),A$,CHR
  $(13);
70 CALL WR,CHR$(27);"@";
80 END
]RUN
012345678901
2345678
```

建立左边列的起始位置(Left Margin)

```
10 POKE 1657,80
20 WR=49568
30 PR#1
40 A$="0123456789012345678"
50 CALL WR,CHR$(13);
60 CALL WR,A$,CHR$(13),CHR$(27);"1";CHR$(8);
  "left margin";
80 END
]RUN
0123456789012345678
left margin
```

建立7/72英寸的打印间隔

```
10 POKE 1657,80
20 WR=49568
30 PR#1
40 A$="7/72 inch line spacing"
50 CALL WR,CHR$(13),CHR$(27);"1";
60 FOR I=0 TO 6
70 CALL WR,A$,CHR$(13);
80 NEXT
90 CALL WR,CHR$(27);"@";
100 END
```

建立字符的下划线

```
10 POKE 1657,80
20 WR=49568
30 PR#1
40 A=3.1415926
50 CALL WR,CHR$(13),CHR$(27);"—";CHR$(1);"A
  =";
60 CALL WR,CHR$(27);"—";CHR$(0),A;F9.7;
70 END
]RUN
A=3.1415926
```

一般形式与特殊形式使用实例

```
4=A$="0123456789012345678901234567890123456
789012345678901234567890123456789012345
678901234567890123456789012345678901234
567890123456789012345"
```

表一

功 能	规 则
初始化	CHR\$(27);"@";@可用CHR\$(64)替代
1/8英寸间隔	CHR\$(27);"0";0可用CHR\$(0)替代
7/72英寸间隔	CHR\$(27);"1";1可用CHR\$(1)替代
1/6英寸间隔	CHR\$(27);"2";2可用CHR\$(2)替代
清除缓冲区	CHR\$(18);
加重打印	CHR\$(27);"E";E可用CHR\$(69)替代
取消加重打印	CHR\$(27);"F";F可用CHR\$(70)替代
压缩打印	CHR\$(15);
放大打印	CHR\$(14);
下划线	CHR\$(27);"—";CHR\$(1);—可用CHR\$(45)替代
取消下划线	CHR\$(27);"—";CHR\$(0);—可用CHR\$(45)替代
建立上标打印	CHR\$(27);"S";CHR\$(0);S可用CHR\$(83)替代
建立下标打印	CHR\$(27);"S";CHR\$(1);
取消上、下标打印	CHR\$(27);"T";T可用CHR\$(84)替代
建立左列起始位置	CHR\$(27);"1";CHR\$(n);1可用CHR\$(108)替代
建立右列起始位置	CHR\$(27);"Q";CHR\$(n);Q可用CHR\$(81)替代
回退一个字符	CHR\$(8);
响铃	CHR\$(7);
	上表只列出一般常用的几种功能,其它功能请参阅有关打印机说明书

注:本文中所列举的程序例子均将打印机接口卡插在1号槽口中。若打印卡插在其它槽口中,程序应按如下修改。

```
10 POKE 1656+N,80
20 WR=49312+256*N
30 PR# N
```

其中N为打印机接口卡所在的槽口。例如打印机接口卡插在3号槽口中,N为3。

```
5 B$="—"
10 POKE 1657,136
```

[illegible]

烟台师院数学系(264000) 刘善平

步骤:1. 运行程序 A,对高分辨率图形进行预处理。对第一页操作,预处理后的图形存放在第二页;对第二页操作,预处理后的图形存放在第一页。这样可以保留原图形。在运行程序 A 之前,需要设置4个参数(表1)。程序 A 的首地址是 \$6100。

2. 运行程序 B,按照要求对预处理后的图形打印输出。程序 B 将裁剪后图形的相应存储单元的内容按顺序调入 \$00FC 单元。用指令 LSR 顺序移位,每移一次判断一次标志位 C 的值及 \$00FB 单元的内容,进行不同的打印输出。程序 B 的首址是 \$6200,在运行前需要设置 6 个参数(表 2)。

举例:把程序 A 和 B 装入内存以后,运行程序 C,取 P=1、V=1 可将高分辨率图形第一页反相打印输出。

至于横向、纵向的放大、压缩以及加重等打印功能,只需编一块 BASIC 子程序,用 PRINT CHR\$(n)语句将打印机预置好。有关的几个 n 值见表3。

十进制地址	十六进制地址	参数内容及意义
6	\$ 0006	被操作页首址的低位
7	\$ 0007	被操作页首址的高位
252	\$ 00FC	预处理后图形存放页首址的低位
253	\$ 00FD	预处理后图形存放页首址的高位

十进制地址	十六进制地址	参数内容及意义
6	\$ 0006	矩形上界的首址的低位
7	\$ 0007	矩形上界的首址的高位
8	\$ 0008	矩形左界/7
9	\$ 0009	矩形右界/7
250	\$ 00FA	下界—上界+1
251	\$ 00FB	0—正相, 1—反相

• 18 •

趣味速算练习器

苏州景范中学(215005) 赵 旭

对孩子进行速算练习,除提高运算能力外,还能开发智力。只要输入以下程序,就可将中华学习机变成一台趣味速算练习器。它能帮助家长出色地完成辅导孩子速算练习的工作。练习器除自动出题和评分外,还能根据孩子的实际程度自动调节练习题的难度。孩子的速算能力随时都可通过练习器得到了解。

练习器具体使用方法是:运行程序后,屏幕上出现五个级别供选择。选定后,练习器自动出题,题目下面有四个答案供选择。同时,屏幕上出现一条不断伸长的光带,并限定在光带伸到某一长度的时间内答出此题。如果回答正确,屏幕上增添五颗星并成绩加五分。连续答对二十题得一百分并升一级,练习题难度也跟着提高。如果成绩不佳,练习器能自动将难度降低,直到孩子能适应为止。

```
1 A$(1)="特";A$(2)="高";A$(3)="中"
2 A$(4)="初";A$(5)="低";A$(6)="级"
3 PRINT CHR$(4);"PR#3";PRINT;HGR2;POKE
49168,0
4 L=1;FOR K=1 TO 9 STEP 2;VTAB K;HTAB 7;POKE 50,
63
5 PRINT L;POKE 50,255;PRINT";A$(L);A$(6);L=L
+1;NEXT
6 POKE 50,127;VTAB 5;HTAB 15
7 PRINT"请您选择(1-5)";POKE 50,225
8 A=PEEK(49152);IF A<127 THEN 8
9 P=A-176;N=10;FOR K=1 TO P:N=N*2;NEXT
HGR2
10 TEXT;HOME;T=0;VTAB 8;HTAB 30;PRINT"TIME";
FOR I=1 TO 19 STEP 2;FOR J=1 TO 11 STEP 10
20 A=INT(99*RND(1))+1;B=INT(99*RND(1))+1;IF
A+B>99 OR A+B<10 THEN 20
30 C=A+B;VTAB 21;HTAB 16;PRINT";VTAB 21;
HTAB
16;PRINT A;"+";B="
40 FOR K=1 TO 4;X=INT(90*RND(1))+10;IF X=
CTHEN 40
45 B(K)=X;NEXT
50 D=INT(4*RND(1))+1;B(D)=C
60 FOR K=1 TO 4;VTAB 23;HTAB 2+K*6;POKE 50,63;
PRINT K;POKE 50,255;PRINT";B(K);NEXT K
70 POKE 49168,0;FOR L=40 TO 1 STEP-1;POKE 50,63;
HTAB L;VTAB 10;PRINT"<";POKE 50,255;FOR M=1
TO N;NEXT
75 B=PEEK(49152);A=B-176
80 IF A=D THEN FOR Q=J TO J+9 STEP 2;VTAB Q;HTAB
```

```
1;PRINT"*";NEXT;T=T+5;VTAB 13;HTAB 30;
PRINT"DE:";T;FOR M=1+1 TO 40;HTAB M;VTAB 10;
PRINT";NEXT M,J,I
85 IF A<>D AND B>127 THEN 100
90 IF B<127 THEN NEXT L
100 PRINT CHR$(4);"PR#3";PRINT;HGR2
105 VTAB 4;HTAB 8;PRINT"您的成绩是:";POKE 50,63;
PRINT A$(P);A$(6);POKE 50,255;PRINT";T;
"分";VTAB 7;HTAB 6
110 IF T<60 THEN PRINT"别灰心,好好练习,一定成功!";
N=N*2;P=P+1;IF P>5 THEN P=5
120 IF T>=60 AND T<80 THEN PRINT"初步成功,继续努
力!"
130 IF T>=80 AND T<100 THEN PRINT"成绩优秀,争取
满分!"
140 IF T=100 THEN PRINT"祝贺您获得成功!";N=N/2;P
=P-1;IF P=0 THEN P=1
150 GET A$;GOTO 10
```

计算机解决数学问题

天津财经学院 216 信箱(300222) 赵方明

计算机无疑是计算能手,它的计算速度和计算范围都是令人惊叹的。但对于计算一些具体的数学问题,编写巧妙的程序是十分重要的,下面是我在平时为解决数学问题而编写的几则程序。

一、分解质因数程序

```
5 M=0;DIMB(100)
10 INPUT N
20 X=N
30 FOR I=2 TO SQR(X)
40 IF N/I <=1 THEN 80
50 IF N/I <> INT(N/I) THEN 70
60 M=M+1;B(M)=1;N=N/I;GOTO 40
70 NEXT I
80 PRINT X;"=";
90 IF M=0 THEN 130
100 FOR J=1 TO M
110 PRINT B(J);"*",
120 NEXT J
130 PRINT N
```

140 END

本程序利用数组存放各个因数,由 80—130 语句打出。而 30—70 语句则是巧妙利用判断语句求出各因数,此求解过程完全模仿人为计算方法,而这对计算机来讲便迅速多了,于是计算机的特长发挥出来了。

二、求两数最大公约数与最小公倍数程序

```

10 INPUT M,N
20 A=(M+N-ABS(M-N))/2
30 FOR I= A TO 2 STEP -1
40 IF INT(M/I) <> M/I THEN 60
50 IF INT(N/I)=N/I THEN 80
60 NEXT I
70 PRINT I,M*N,GOTO 10
80 PRINT I,M*N/I,GOTO 10

```

本程序通过计算机搜索最小公倍数 I,也就得到了最大公约数 $M*N/I$ 。20 语句实际是求得 M 与 N 中较小的一个。当然用短除法也可以编程解决此题,与程序一相仿,不如现程序简明。

三、求圆周率近似值程序

在一次计算机竞赛中出现过这样一道题:请编程计算出 π 的近似值,而且须用 355/113 求得,并保留 20 位有效数字。有的同学想方设法扩大计算机运算范围,以达到保留位数,但一无所成。其实灵活一点,很容易获得答案:

```

5 A=355,B=113,N=0
10 PRINT"355/113=";
20 N=N+1;PRINT INT(A/B);
30 IF N=1 THEN PRINT". ";
40 A=(A-INT(A/B)*B)*10
50 IF N<20 THEN 20
60 END

```

程序中 40 语句相当于笔算除法中的落零,50 语句判断是否达 20 位。这样程序一运行,计算机就象人一样算起除法,很快地完成任务。

以上三个程序看来并不高深,但却结构清晰。它们的共同特点都是考虑到我们自己的处理方法,结合具体过程,用灵巧的语句编制而成,这样就使计算机出色地完成了工作。对于数学问题和其他任何问题,我们都应通过合理简洁的程序来解决它。

巧改系统

北京三里河群智巷(100051) 张 浩

是否能用自己或朋友的名字作为系统命令呢?或把关键的命令更改,使别人不能使用?在 LASER310 机上 BASIC 命令是固化的,但通过对编译和执行阶段的拦截可以改造系统。我用 Z-80 机器语言编了一个小程序,用它可以把系统命令改成任何字符串。本程序以

改命令 CLS 为例,要改其它命令则要做相应改动。

使用方法:键入并 RUN 本程序后,出现问号,要求输入改后的命令字符串,然后回车即可。这样命令就被改了。如果,再键入 CLS 命令则会出现语法错误。本程序 RUN 后可 NEW 掉,不占 BASIC 区。程序附后。

```

10 INPUT B$;T=LEN(B$)
20 FOR I=1 TO T:B(I)=ASC(MID$(B$,I,1));NEXT
30 FOR I=31058 TO 31112:READ A;POKE I,A;NEXT
40 FOR I=31113 TO 31112+T;POKE I,B(I-31112);NEXT
50 POKE 31088,T;POKE 30862,82;POKE 30863,121;A=USR(0)
60 DATA 33,89,121,34,4,120,201,217,33,91,29,209,183,237,82
70 DATA 213,217,194,120,29,205,120,29,254,132,202,151,25
80 DATA 229,6,5,17,137,121,26,190,32,14,205,120,29,19,16,246
90 DATA 209,209,205,201,1,195,30,29,225,126,201

```

数字游戏

河北机电学院自动化电气 91.1 班 闫 浩

这是一个简单但很有趣的智力游戏。程序运行后,计算机先询问一下:“Player's Name:”,当你键入三个字如(代表你的姓名)后,游戏开始:屏幕上画出一个数盘,数盘里面的数字按一定的顺序排列,这时你应抓紧时间记住这个顺序和左上角显示的那个数字的位置。然后数盘被打乱,原先显示在左上角的那个数字变为闪烁字符,你能用控制键(I,M,J,K)来控制它的移动,分别使其和上,下,左,右的数字换位。如果你能在规定的步数(150 步)内将数盘复原,则算你胜,游戏结束(再玩,则按“S”键),若没能复原,则算失败,计算机将再让你试一次。

注:20 句中和 200 句中引号内均为 2 个空格,80 句中引号内为 14 个空格。

```

5 DIM A(5,4);A=150
10 HOME;B=-1;C=10;B$="";E=INT(RND(1)*20)+10
20 VTAB 2;HTAB 29;PRINT"1. Number";VTAB 6;HTAB 28;PRINT "HI.";A;" ";A$:VTAB 9;HTAB 28;PRINT "Player:";
30 FOR I=1 TO 3:GET C$;B$=B$+C$;PRINT C$;NEXT I
40 FOR I=0 TO 5;VTAB I*4+2;HTAB 4;FOR J=1 TO 21:PRINT "-";NEXT J,I;FOR I=0 TO 4:VTAB 2;FOR J=1 TO 21;HTAB I*5+4;PRINT"!";NEXT J,I
50 FOR I=1 TO 5;FOR J=1 TO 4:A(I,J)=C;IF C/E=1 THEN X=I,Y=J

```

(下转第 11 页)

ProDOS 磁盘操作系统入门(续)

北京铁路局中心卫生防疫站(邮编 100038) 廖 凯

第六章 外围设备

本章将介绍 ProDOS 如何用不同的外围设备及附件工作。涉及到的硬件有扩展 80 列卡, 扩展 80 列 Text/AppleColor 转换卡, ROM Upgrade Kit, Apple 鼠标器, ProFile 5M 硬盘, Thunderclock 和 Proclock 时钟/计时卡。

一、APPLE 扩展 80 列卡

扩展 80 列卡将在 ProDOS 下作为一个 RAM 磁盘, 这允许用户在 RAM 磁盘内快速地存取程序或文件。APPLE II c 具有 128K RAM, 64K 主存储器, 64K 辅助存储器, 和 APPLE II e 一样, 在 ProDOS 下这辅助存储器作为一个 RAM 磁盘。

APPLE II c 和 II e 具有双高分辨率图形功能。注意, APPLE II e REV A 不具备双高分辨率图形。

二、扩展 80 列 Text/AppleColor 转换卡

Color 卡是 APPLE 最初具有 RGB 输出方式的 80 列卡, 有扩充的 64K 容量, 可代替扩展 80 列卡。此卡用一个调制器连接 APPLE RGB 彩色显示器, 或经由标准视频输出连接复合彩色显示器或单色显示器。可以同时用 RGB 和单色两种方式显示。对于单显和复合彩显所使用的视频 I/O 是标准的连接器, 在主机后面可找到, RGB 彩显是用扩展/RGB 卡连接的。

此卡支持所有的标准 APPLE II 视频方式。另外, 扩展/RGB 卡支持文字颜色的转换和新的双高分辨率图形。借助于此卡可有 6 种图形方式:

- 方式 1 560×192 黑色和白色
- 方式 2 140×192 全 16 种颜色
- 方式 3 混合方式 方式 1 与方式 4 的组合
- 方式 4 160×192 全 16 种颜色
- 方式 5 280×192 16 种颜色
- 方式 6 混合方式和低分辨率一样, 80 列

此卡包含一套实用软件包。为了启动卡的图形方式, APPLE 提供了一个很有用的接口模块。B 型文件 Hires 程序将给用户提供一些惊人的功能。Hires 模块是由 19 个图形输出子程序组成, 这些程序执行一串 & 指令, 使用很简单。指令如下:

& GR# 设置指定的图形子程序。在执行此指令时, 当前笔的位置被设为 0,0(屏幕的左上角)。笔的颜色系统设定为白色, 背景为黑色, 若选择 0, 则关闭图形方式并清除存储器空间。

& TEXT 此指令转换图形方式为文本显示方式。

& COL=# 设置被 PLOT, DOT, DRAW 和 GPRNT 指令所使用的笔的颜色。此指令使用与低分辨率图形

相同的 16 种颜色。方式 1 只能用黑色和白色绘图。

& BCOL=# 设置背景颜色或经由 CLEAR 和 VFILL 指令来涂色, 使用与低分辨率图形相同的 16 种颜色。

& CLEAR 此指令用 BCOL 指令指定的颜色来清屏, 若未用 BCOL 指定颜色, 则系统设定为黑色。

& VFILL(X1,X2,Y1,Y2) 此指令在屏幕上设置四个座标点并在此范围内用 BCOL 指令最后设置的颜色进行涂色。X1 和 X2 是从左到右的水平座标, Y1 和 Y2 是从上到下的垂直座标。

& MOVE(X,Y) 将笔移到 X,Y 位置而不在屏幕上绘图。X 表示从左到右的水平座标, Y 表示从顶到下的垂直座标。

& DOT 用当前前景颜色(COL)放置一个点作为当前笔的位置。此指令不需要参数。

& SCRN(C) 送回当前笔位置的颜色(COL)值。

& SAVE/卷名/文件名 此指令是将高分辨率图形页 1 图形作为一个二进制文件存盘。

& LOAD/卷名/文件名 从磁盘上将上述保存的图形装到内存的高分辨率图形页 1。

& GPRNT 允许文字显示在图形屏幕上。在此指令有效时仍可以绘图, 所有随后的输出经一个 PRINT 语句使文字显示到图形屏幕上。使用字形与 APPLE II 一样。字符被显示在当前笔的位置的右边, 在字符之间有 7 个象素。除了回车字符(ASCII 13)和换行字符(ASCII 10), 控制字符将被忽略。

& CPRNT# 此指令在图形打印子程序中选择控制码。若 # 等于 1, 将选择控制码并打印; 若 # 等于 0, 则不选择控制码。例如, 这功能可接受附加的自定义图形字符。

& TPRNT 取消 & GPRNT 指令的设置。

& NCHARS/卷名/文件名 此指令用于改变在图形显示下的标准字形为用户自定义字形。自定义字形必须与 & DRAW 指令设置的参数一致。

& SCHARS 此指令恢复标准的系统字形。

& DRAW(M,N,N1,R,W,H) 在屏幕上绘制一个预先定义的造型。此造型被放置在当前笔位置的右边。M 参数是造型首字节的存储器地址; N 参数指定造型每行的字节数或长度; N1 参数指定每行之间的位数; R 参数指定在转移造型之前所跳过的行数; W 参数指定造型的宽度; H 参数指定被转移的造型的高度。

注意, 上述指令中, 括号内的参数必须与指令一起使用。

你可以用 PRINT 语句在任何图形方式中显示文字。下面的短程序将使用这种方法在方式 1 和双高分辨率图形中打印一行文字。

```
10 PRINT CHR$(4); "-HIRES"
20 &. GR1
25 &. CLEAR
30 &. PLOT (0,191); &. PLOT (559,191); &. PLOT (559,
    0); &. PLOT (0,0)
40 L$ = "You can output text to a GRAPHICS screen using
    PRINT"
50 Y=60; GOSUB 100
60 L$ = "ESC to back out"
65 Y=135; GOSUB 100
70 GET A$; IF A$ = CHR$(27) THEN TEXT; GO TO
    99
80 GOTO 70
99 &. TEXT; END
100 X = 280 - (LEN(L$) * 3.5); &. MOVE (X,Y)
110 &. GPRNT; PRINT L$; &. TPRNT; RETURN
```

程序先装入 HIRES 程序并设置图形方式为 1。行号 30 用四个 &. PLOT 指令从当前图形座标(0,0)沿屏幕四周画一边框。&. PLOT 指令将从当前位置到绘图位置画一条线。行号 50 设置 Y 座标并转到行号 100 执行。行号 100 将文字放在屏幕中央并移动指针到新 X, Y 位置。&. GPRNT 指令将使文字显示在图形屏幕上。随后的打印语句将输出指向图形屏幕, &. TPRNT 指令恢复屏幕为图形方式。

HIRES 模块也可用于扩屏 80 列卡,有三种双高分辨率图形方式。它们是:

560×192 黑色和白色
140×192 16 种颜色
混合方式 上面两种方式的组合

HIRES 模块在主机内占用极少空间,模块较大部分保留在辅助存储器内,其余部分在主存储器内。

三、APPLE 鼠标器

鼠标器经 9 芯连接器连接在 APPLE 机上。鼠标器可以取代箭头键在屏幕上移动光标。鼠标器必须用 PR # 命令来启动;要从鼠标器上接收信息就必须使用 IN # 命令,随后跟一个 INPUT 语句以存储资料到变量内。鼠标器的当前位置和按钮的状态随时都可以知道。

鼠标器用一个 BASIC 程序来沟通。下面的程序将演示用户怎样用鼠标器绘制一个高分辨率图形。

```
10 D$ = CHR$(4); PRINT CHR$(21)
20 HGR2
30 COLOR = 15; SCALE = 1; ROT = 0
40 HPLLOT 0,0 TO 0,191 TO 279,191 TO 279,0 TO 0,0
60 PRINT D$; "PR #4"; PRINT CHR$(1)
70 PRINT D$; "PR #0"
80 PRINT D$; "IN #4"
90 INPUT "X,Y,Z"
100 INPUT "X1,Y1,Z1"
110 IF Z=1 AND Z1=1 THEN 130
115 IF Z=4 THEN 90
```

```
120 IF Z<0 THEN 190
130 X=INT(X/3.66)
140 Y=INT(Y/5.33)
150 X1=INT(X1/3.66)
160 Y1=INT(Y1/5.33)
170 HPLLOT X,Y TO X1,Y1
180 GOTO 90
190 PRINT D$; "PR #4"; PRINT CHR$(0)
200 TEXT; PRINT D$; "PR #3"
210 END
```

行号 60 接通鼠标器;行号 70 设置屏幕输出。用 IN # 命令读取鼠标器的状态,随后跟两个 INPUT 语句用于从 A 点到 B 点绘图。鼠标器 X 和 Y 的取值范围在 0 和 1023 之间。当鼠标器向右移时, X 值将增加;当鼠标器向自己移动时, Y 值将增加。X 和 X1 不能超过 279, Y 和 Y1 不能超过 191。如果按住按钮并很快地移动鼠标器,那么将会引起跳行和线不相连。

行号 110 和 120 测试按钮的状态。如果 Z 是负数,则表示已从键盘上按过键。如果连续地按着按钮,将没有行被绘制。按钮可能存在的状态如下:

Z	当前读取	最后读取
1	按下	按下
2	按下	释放
3	释放	按下
4	释放	释放

如果键盘上的键被按过,行号 190 将关闭鼠标器,并返回文本显示状态。

四、ROM Upgrade Kit

ROM Upgrade Kit 可替换计算机内的三块 ROM。MouseText 字符发生器 ROM 替换 Video ROM 342-0133,此 ROM 将产生 32 个新字符并替换反相大写字符集。用两块监控 ROM 替换 ROM C66 和 C67,这些 ROM 与 MouseText 一起工作并提供一些新的功能。

MouseText 是为使用 APPLE 鼠标器而开发的,它可产生 32 个 ICONS 图形。标准字符集不受 Upgrade Kit 影响。如果你打算用 BASIC POKE 语句在屏幕上显示反相的大写字符,那么将会发生问题,因为新的字符集已被放在 ASCII 码 64 到 95 的位置。反相大写字符集已被放在 ASCII 码 0 到 31 的位置。

如果你使用的是 I c,这些 ROM 已被安装在主机上。其它机型可自行购买安装。

两块监控 ROM 在 80 列显示方式时会有些改进:

1. APPLESOFT 指令和 ProDOS 命令均可用小写字母输入。所有键盘指令名都已做在键盘上,与 shift 合用,不包含 DATA 指令、REM 和串指令。在 ProDOS 下所有指令在任何状态下都可用小写字母输入。

2. HTAB、TAB、SPC 和 PRINT 语名中的逗号均可在 80 列范围内使用。

3. BASIC 在 80 列状态下输出字符的速度将提高 30%。

4. 粗糙的屏幕卷印已被取消。

5. 增加了两个 ESC 指令;

* ESC CTRL-D 中止所有打印控制码,回车、换行、退格和警铃除外。

* ESC CTRL-E 恢复打印控制码为系统设定码。

6. 在列印程序时,整个程序都向右移动一列,这样在编辑程序时不必再向左移动光标扫描行号了。

下面的程序将显示 ROM Upgrade Kit 的 32 个新字符。用此程序可检查计算机是否装有新 ROM。

```
5 HOME
10 PRINT CHR$(4);"PR #3"
20 PRINT
30 INVERSE
40 FOR I=64 TO 95:PRINT CHR$(27);CHR$(I)CHR$(24);
50 NEXT
60 NORMAL:END
```

五、ProFile 硬盘

APPLE IIe 现在可以使用名为 ProFile 的 5M 或 10M 硬盘,此硬盘是苹果公司为 APPLE II 和 Lisa 计算机生产的。APPLE II 和 APPLE II 使用相同的磁盘格式,故两种计算机都可使用 ProFile 硬盘。

ProFile 是 5 英寸 Winchester 硬磁盘系统。ProFile 存取资料比 DISK II 快三倍。ProDOS 实用程序用 ProFile 操作并允许用户在不同的系统之间使用和拷贝文件。硬盘具有 5M 的容量,你可以用 ProFile 存取大量的资料,它相当于 35 个软磁盘。ProDOS 的分级文件结构使它很容易在列印的大量文件中找到一个文件。

有一个为硬盘而设计的名为 Catalyst 的程序,允许你从一个主菜单选择并运行硬盘里的程序。此程序可以在具有 64K 的 APPLE II 上运行。这程序对 ProFile 的用户来说是必要的。

六、时钟/计时卡

1. Thunderclock

Thunderclock 是一个与 ProDOS 系统兼容的时钟/

计时卡。ProDOS 可以自动识别此计时卡,并可在文件上记录文件的建立及修改日期和时间。在 MLI 内有一个支持 Thunderclock 的中断驱动程序。每当存取磁盘时,ProDOS 将当前的时间和日期存入内存地址 49040 至 49043 之间。此地址 BASIC 程序不会占用。

你可以用以下的语句显示日期和时间:

```
DAY=PEEK(49040)-INT(PEEK(49040)/32)*32
YEAR=INT(PEEK(49041)/2)
MONTH=(PEEK(49041)-YEAR*2)*8+INT(PEEK(49040)/33)
HOUR=PEEK(49042)
MINUTE=PEEK(49043)
```

这些指令只显示时间和日期。当前时间不能修改直到磁盘被存取为止。在 Users. Disk 的主菜单上选择 Display/Set Time 功能,你可以观察时间并过一会儿再观察时间,你会发现它没有变化。你在 CATALOG 磁盘后再观察时间,你会发现时间已改变。不足之处是你不能用 BASIC 写一个中断程序。

Thunderware 公司开发了一个名为 TUT (Time Utility) 的实用程序以在 ProDOS 下存取此卡。此程序支持所有用于此卡的指令,它占用内存 4K 字节。注意, HIMEM 将下移 4K 字节以保持正常工作。

2. Proclock

Proclock 是另一个具有存取 ProDOS 能力的兼容时钟/计时卡。象 Thunderclock 一样,此卡将在文件建立或修改时放置时间和日期,其用法与 Thunderclock 一样。 (全文完)

ProDOS 磁盘操作系统入门的介绍暂告一段落。关于 ProDOS 系统功能的调用,需要经由机器语言接口 MLI。这一部分资料我们将在准备就绪后陆续刊出,请读者留意,有关技术咨询请与廖凯同志接洽。 编者

(上接第 18 页)

表 3

n 值	功能
27,64	初始化设置
14	放大方式
15	压缩方式
27,65;n	行间距为 n/72 英寸
27,69	加重方式

程序 A:

```
6100- A5 06 85 08 85 FA A5 07
6108- 85 09 85 FB A9 08 8D 07
6110- 60 8D 08 60 A9 03 8D 06
6118- 60 A9 28 8D 00 60 A9 80
6120- 8D 02 60 A9 04 8D 05 60
6128- A9 00 8D 01 60 8D 03 60
```

```
6130- 8D 04 60 A2 00 A0 00 B1
6138- FA 91 FC C8 C0 28 90 F7
6140- D8 18 A9 28 65 FC 85 FC
6148- A9 00 65 FD 85 FD 18 AD
6150- 04 60 65 FA 85 FA AD 05
6158- 60 65 FB 85 FB CE 08 60
6160- D0 D3 A9 08 8D 08 60 18
6168- AD 02 60 65 08 85 08 85
6170- FA AD 03 60 65 09 85 09
6178- 85 FB CE 07 60 D0 B6 A9
6180- 08 8D 07 60 18 AD 00 60
6188- 65 06 85 06 85 08 85 FA
6190- AD 01 60 65 07 85 07 85
6198- 09 85 FB CE 06 60 D0 95
61A0-60
```

程序 B:

```
6200- A4 08 A2 07 B1 06 85 FC
6208- 46 FC 90 09 A9 00 C5 FB
6210- 90 09 4C 20 62 A9 00 C5
```

(下转第 31 页)

第五章 源程序的编辑、汇编和运行

南京大学 大气科学系(210008) 朱国江

用汇编符号语言编写的汇编语言源程序,必须翻译成用机器码表示的机器语言程序(即目标程序),计算机才能识别和执行。完成这种翻译工作的过程称为汇编。这可以用不同的方法来实现,最常用的是用系统提供的软件,如编辑汇编软件 EDASM、LISA 等;也可以用小汇编程序 Mini Assembler。在调试和执行机器语言时,也常用监控程序 MONITOR 中提供的监控命令。下面我们分别介绍这些软件和使用。

一、编辑汇编程序(EDASM)的使用

用汇编语言写好的源程序必须首先用编辑软件 EDITOR 将源程序送入计算机,形成源程序的 ASCII 码文件(源文件),存放在磁盘上,这个过程称为编辑。然后调用汇编程序 ASSEMBLER,对磁盘上的源文件进行汇编,形成目标文件送入磁盘,然后才能运行该目标程序。编辑程序和汇编程序已经合并成一个文件,称为 EDASM 文件(编辑汇编),下面介绍它的用法。

1. 编辑汇编程序的启动

启动 EDASM 有两种方法;它们是:

(1)如果系统已经引导了 DOS3.3,则应将 DOS3.3 软盘取出,换以 TOOL KIT(EDITOR 和 ASSEMBLER 均在此盘中)软盘插入驱动器,然后打入命令:

```
]BRUN EDASM. OBJ
```

即从磁盘调入并执行 EDASM 的目标码文件,使系统处于它的管理之下,屏幕上出现提示符:及游标。

(2)如果系统接通电源后,没有在驱动器中插入 DOS,而是插入 TOOL KIT 盘片,则可打入命令:

```
]RUN EDASM
```

从软盘调入并运行 EDASM 的 BASIC 文件,屏幕上出现:

```
APPLE I EDITOR-ASSEMBLER
CURRENT ASSEMBLER ID STAMP IS:
20 JAN 89 # 000001
```

字样,并有闪动游标,此时按一下 RETURN 键,待屏幕上出现 EDASM 提示符:和游标时,系统进入 EDASM 的编辑状态,等待用户输入命令,即可使用。

2. 源程序的编辑

首先用 EDASM 的编辑命令将源程序送入计算机,然后再使用有关命令对源程序进行必要的编辑和修改。

(1)A 命令(ADD)——键盘输入源程序命令

使用 A 命令可通过键盘输入汇编语言源程序。例如,将 16 个数据的数据块从内存单元 DATA~DATA

+15 移到 0 页的 \$F0~\$FF 中。源程序输入过程如下:

```
:A
1 ORG $7000
2 BUFFER EQU $F0
3 LDX #0
4 START LDA DATA,X
5 STA BUFFER,X
6 INX
7 CPX #16
8 BNE START
9 RTS
10 DATA DS 16
11 CTRL-D (按住 CTRL 键再按下 D)
```

上述程序输入过程中,每行的行号是机器本身自动产生的。输完一行后应输入回车符(用↵代表)。若某句中有标号,则紧跟行号后打入标号,行号和标号之间不留空格;若某句无标号,则应在行号后敲一空格键。全部汇编语言源程序输入完毕后,用 CTRL-D 敲入表示结束,并退出 A 命令返回“:”状态。

(2)L 命令(LIST)——列表显示源程序命令

此命令用于显示已送入内存的文本,其操作为:L 行号↵。

如:L 1-3↵ 显示已输入文本的 1-3 行;:L↵ 则显示文本的所有行。

(3)P 命令(PRINT)——打印源程序命令

该命令作用同 L 命令,但不显示行号,其操作为:P 行号↵。

(4)C 命令(CHANGE)——修改源程序命令

此命令可修改已送入内存中文本的某行,格式为:

:C 行号. 旧字符串. 新字符串↵

或:C 行号 \$ 旧字符串 \$ 新字符串↵

符号. 或 \$ 是分界符,不能省略,上述格式的含义是以新字符串代替旧字符串。并回答系统的询问:

ALL OR SOME? (A/S)? (全部或部分)

若敲 A 键回答,则表示该行中 C 命令指出所有旧字符串都要用新字符串代替;若敲 S 键回答,则表示仅修改部分旧字符串。对要修改的旧字符串,需要再敲一次 S 键,系统才能执行将新字符串替换该旧字符串;对于要保留的旧字符串,则敲 ESC 键跳过。例如,将前例中第 1 行的字符 7 改为字符 8,操作如下:

```
:L 1↵ 显示第 1 行 1 ORG $7000,以便下面
```


观察修改情况

:C 1.7.8/ ;将第1行的7改为8

ALL OR SOME? (A/S)?

若按 S 键回答,则表示只修改“7”字符,以后按 ESC 键,则不再作修改。此时用 L 命令

:L 1/

1 ORG \$8000

(5)D 命令(DELETE)——删除命令

此命令用来删除已用 A 命令送入机内的文本中的某行(或某连续段),如:

:D 1/ ;删除第1行

:D 1-5/ ;删除第1到第5行

执行 D 命令后,可用 L 命令检查被删除的情况。

(6)I 命令(INSERT)——插入命令

该命令可在文本的规定的行处插入一行语句。如在前面的例子中第8行插入 NOP,则操作如下:

:I 8/ ;机器立即空出第8行,等待插入

8 NOP/ ;在第8行插入 NOP

9- ;还可继续在第9行插入

如果不需再插入,则按 CTRL-D 退出 I 命令,用 L 命令检查插入后的结果,将发行后续行号已重新排列。

(7)R 命令(REPLACE)——替换命令

此命令对机内的文本中某行(或连续行)进行替换。例如,将上述程序第10行的 RTS 改为 SBC,则

:R 10/ ;修改第10行

10 BRK/

11- ;还可继续修改11行

如不需要继续修改,则按 CTRL-D 退出 R 命令,再用 L 命令检查修改后的情况。

(8)E 命令(EDIT)——编辑命令

此命令用来对已用 A 命令送入机器的文本中的某一行作局部编辑,执行:E 行号/,在显示出指示行号内容后,配合使用下述控制键对源程序进行修改:

→ 光标向右移动一个字符位置。

← 光标向左移动一个字符位置。

✓ 编辑结束并有效,退出 E 命令,回到 EDASM 状态。

^D 删去光标指出位置上的字符。

^I 在光标指出位置插入字符。

^T 删去光标指示位置以后的全部字符。

^R 改写光标指定位置的字符。

^F 在此命令后跟着打入要在该行寻找的字符,则光标会自动移动到所寻的字符位置上。

^X 取消正在进行的操作,回到 EDASM 状态。其中 ^ 代表 CTRL 键。

(9)F 命令(FIND)——查找字符命令

此命令用来对已用 A 命令送入机内的文本中找出某行,或某连续段,或某指定字符串所在行。如:

:F 1/ 查找上例中的第1行

:F. STA/ ;查找上例中含字符 STA 的所有行

4 START LDA DATA,X;第4行中含有 STA

5 STA BUFFER ;第5行中含有 STA

8 BNE START

;第8行中含有 STA

(10)SAVE——存盘命令

SAVE 文件名,可将用编辑命令编辑好的一个用户源文件存入磁盘。例如,我们将上面的汇编源程序起一个名字,如 EXE1,则

:SAVE EXE1/

存盘后可用 CATALOG 命令检查是否已存进盘中。

:CATALOG/

这时屏幕上显示的磁盘目录中已有一个名为 EXE1.EDASM 的文件存在了。

(11)LOAD——文件载入命令

LOAD 文件名将指定的文件从磁盘调入内存。

如:LOAD EXE1/,则将文件 EXE1 调入内存中。

以上介绍了如何将一个汇编语言源程序送入计算机,并用上述有关命令进行必要的修改和编辑,汇编语言源程序编辑完成后,可以存盘备用。

3. 源程序的汇编

汇编语言源程序经过编辑并存盘以后,就可以执行汇编命令 ASM 源文件名。此命令的作用是对磁盘上指定的源程序文件进行汇编,翻译成目标程序。

(1)如何汇编源程序

例如,有一汇编语言源程序,它将 \$2000 和 \$2001 两个单元的内容进行十进制相加后,存入 \$2002 单元:

ORG \$0300

START SED

CLC

LDA \$2000

ADC \$2001

STA \$2002

BRK

则采用下述前步对该程序进行编辑和汇编:

• 启动编辑汇编程序 BRUN EDASM.OBJ/

• 将汇编语言源程序送入计算机,并进行编辑

• 将编辑完成后的源程序取名存盘,如 SAVE

SHUX/

• 在编辑汇编状态“:”下对 SHUX 进行汇编,即:ASM SHUX/。此时,系统首先从 EDASM 文件中调出汇编程序,并在屏幕上显示如下提示:

PRESS ANY KEY TO CONTINUE

这时敲入任一键作回答,机器就开始对盘上的 SHUX 文件进行汇编,并在屏幕上显示源文件名 SHUX、汇编后的目标文件 SHUX.OBJ0 以及汇编后的程序清单:

SOURCE FILE,SHUX

---- NEXT OBJECT FILE NAME IS SHUX.OBJ0

0300 1 ORG \$0300

0300:F8 2 START SED

```

0301:18      3      CLC
0302:AD 00 20  4      LDA $ 2000
0305:6D 01 20  5      ADC $ 2001
0308:8D 02 20  6      STA $ 2002
030B:00      7      BRK

```

* * * SUCCESSFUL ASSEMBLY;NO ERRORS

? 0300 START ? 0300 START

汇编完成后敲任意键,系统退出汇编,回到 EDASM 状态,屏幕上出现冒号提示符。至此汇编完毕,若用 CATALOG 命令,可以看到盘上已有一个 SHUX.OBJ0 的目标码文件。

(2)如何打印汇编清单

要在汇编过程中打印清单,可按如下步骤操作:

- 打开打印机电源
- 键入以下命令
:PR #1,LXX PXX
:ASM 源文件名

其中 LXX 表示——页中要打印的行数,PXX 表示一页中的总行数。如 L 18 P20,表示这一页共 20 行,打印 18 行后空四行。如 LXX 中填入的数大于 PXX 中填的数,则打印将连续下去,不分页。

(3)汇编过程的控制

在汇编过程中,可用下面三个命令,对汇编过程进行控制:

- 停止汇编,回到编辑状态(CTRL-C)
- 暂停汇编(空格键),按任意键继续汇编
- 关闭显示(CTRL-N),CTRL-O 继续显示

4. 目标程序的运行

仍以 SHUX 为例,若要运行它的目标文件 SHUX.OBJ0,其操作如下:

```

:END;执行此命令,系统进入 BASIC 状态
]BRUN SHUX.OBJ;在 BASIC 状态下运行目标程序
*           ;运行完毕,进入监控

```

若要在监控下运行此程序时,可用 0300G,这时不用再调盘,而可直接运行已送入内存的目标程序。这里的 0300 是 SHUX.OBJ0 的首地址。

当遇到程序汇编后产生的目标程序不止一个时,例如某程序由三个 ORG 伪指令分成了三段,因而汇编后产生三个目标文件,运行时,首先将三个目标文件分别调入内存,如:

```

]BLOAD NUMBER.OBJ0
]BLOAD NUMBER.OBJ1
]BLOAD NUMBER.OBJ2

```

然后键入命令:

```

]CALL-151;进入监控状态
* 300G;从目标程序首地址开始运行

```

二、小汇编程序(Mini Assembler)的使用

固化在 CEC-I 中华学习机内的小汇编程序,是一个

简单实用的汇编工具软件,它可以将输入的汇编语言源程序的符号指令,逐条翻译成机器指令,存放在指定的内存单元中并显示出来,这在我们编写好汇编语言程序后常常采用,它会省去我们查指令表翻译机器码的麻烦。但要注意的是,小汇编程序不认识伪指令,所以使用小汇编时不能用符号地址和标号等,因而只能用来汇编一些简单的源程序。此外,小汇编程序在 APPLE I 机中是放在 DOS 系统主盘的 INTBASIC 文件中,它的启动与中华学习机 CEC-I 有些差别。

1. 进入和退出小汇编状态

(1)对中华学习机 CEC-I 进入小汇编

]CALL-151;在 BASIC 状态下进入小汇编

* D350G

! ;! 是进入小汇编的标志

在监控状态下进入小汇编可直接执行上述第二条命令。

(2)对 APPLE I 进入小汇编

• 插入带 INTBASIC 文件的系统主盘,键入 PR #6,待出现了提示符后键入:

]INT

> ;表示进入整数 BASIC 控制之下

>CALL-151

* ;表示进入 RAM 卡的监控程序控制之下

* F666G

!

(3)对 CEC-I 退出小汇编

! \$D360G;返回监控状态

或! \$FF69G;返回监控状态

或! \$35D0G;返回 BASIC 状态

(4)对 APPLE I 退出小汇编

! \$FF69G;返回监控状态

或! \$3D0G;返回整数 BASIC 状态

或! INT;返回整数 BASIC ">" 状态

或! FP;返回 Applesoft BASIC "]" 状态

2. 在规定内存地址送入符号指令

在小汇编提示符下,按照地址:符号指令格式操作,就可以将输入的一条符号指令翻译成机器码,并显示在指定的地址中,若输入多字节指令,则将机器码存放在以该地址为首地址的几个连续存储单元中。例如,敲入

! 1000;LDA \$ 4000

则输入一条符号指令,汇编一条指令,并显示结果

1000—AD 00 40 LDA \$ 4000

这里 LDA 的操作码是 AD,它放在 \$1000 单元中,操作数 00 40 存放在 \$1001 和 \$1002 单元中。

若想继续输入下一条指令助记符,则不用再输入地址,可在输入“地址”的位置先敲入一个空格,紧接着输入符号指令即可,这时小汇编会自动计算下一个地址。这样,只要设定初始地址后,就可以连续输入一系列的汇编语句。例如,有一完整的汇编源程序,按下述

操作送入:

```
! 300;SED/
0300-F8 SED
! CLC/
0301-18 CLC
! LDA # $ 40 /
0302-A9 40 LDA # $ 40
! ADC # ¥10/
0304-69 10
! BRK/
0306-00 BRK
!
```

当上述汇编语言输入结束后,汇编工作也就完成,为了显示上述工作的结果,可打入! \$ 300L/,这样,小汇编程序就把自 \$ 0300 单元开始存放的目标程序进行反汇编,并显示如下(一次显示 20 行):

```
! $ 300L/
0300-F8 SED
0301-18 CLC
0302-A9 40 LDA # $ 40
0304-69 10 ADC # $ 10
0306-00 BRK
0307-00 BRK
:
:
0313-00 BRK
```

3. 在小汇编状态下使用监控命令

在小汇编状态下,可以使用中华学习机或 APPLE II 机的各种监控命令,这是非常方便的。使用时应注意在监控命令前加上符号“\$”,这样,监控命令就变成了小汇编状态下的命令。例如:

! \$ 首地址 G/,运行指定首地址的目标程序。

! \$ 首地址 L/,从规定的首地址开始列出源程序及对应的机器码。

! \$ 0300/,检查地址为 \$ 0300 单元的内容。

! \$ 1000;00 00/,修改从 \$ 1000 单元开始的连续两个单元的内容,使皆为 00。

4. 使用小汇编的注意点:

• 小汇编不能识别伪指令。因而在使用小汇编时,不能输入伪指令,也不能用标号和符号地址,只能给出具体数据和绝对地址。

• 符号指令中的操作数或操作数地址,只能使用 16 进制数,不能用其它任一进制数,因为小汇编程序只识别 16 进制数。而且 16 进制表示符号 \$ 可以不写。

• 在使用转移指令时,其地址码部分,小汇编规定只能用转移目标的绝对地址,而不允许用符号地址,也禁止用转移步长。

三、监控(MONITOR)命令的使用

监控系统是与计算机结构密切联系的管理计算机的重要程序,它固化在只读存储器 ROM 中,是人们使用计算机的有用工具。监控命令提供对系统的基本操

作,比如机器语言程序的输入、修改、读出、检查、传送等。还为机器语言的使用、运行目标程序、跟踪程序执行、比较寄存器和内存内容,对程序进行反汇编提供了良好的手段。监控系统中许多结构严谨、短小精悍的子程序,便于用户直接调用,大大减少了编制其它程序的工作量。

(1) 进入和退出监控状态

• 在 BASIC 状态下进入监控状态,方法很简单,即 CALL-151/,出现的 * 号就是进入监控的标志符。

• 在监控状态下返回 BASIC 状态,可用 CTRL-B 或 CTRL-C 键,但前者将使 BASIC 程序及变量全部消失,而后者能保存原来的 BASIC 程序及变量。此外,用 * 3D0G/也可返回 BASIC。并保存内存中的 BASIC 程序。

(2) 显示存储器的内容

在调试程序时,有时要检查存储单元中的内容及变化情况,就可以使用监控程序提供的显示存储器内容的功能。它有四种格式:

格式 1: * 地址/,这个命令只能显示一个存储单元的内容,如 * 1000/,显示 1000-30,这里 30 就是 \$ 1000 单元中的内容,它是一个 16 进制数(下同)。

格式 2: * 首地址. 尾地址/,它显示首地址到尾地址的所有存储单元的内容。

格式 3: *. 地址 2/,显示从当前地址开始到地址 2 的各存储单元内容。

格式 4: * /,只敲回车键,显示从当前地址开始的八个单元内容。

(3) 修改(或输入)内存单元的内容

当需要直接修改某一存储单元的内容,或者在某些内存单元设置一些参数时,可以使用监控提供的这种功能。它有两种格式:

• 格式 1: * 地址:数据 数据 .../,这时敲入需要修改的内存单元地址和一个冒号“:”,然后送入要向该单元传送的数据,最后敲回车。若要连续存放一组数据,可在每个数据后敲一空格即可。

• 格式 2: *:数据 数据 .../,这是接前面的地址将数据继续送入内存。

如 * 1000;00 01 02 03/,这是从 \$ 1000 单元开始连续修改 4 个单元的内容,用这 4 个数据代替原来的数据。又如 *:04 05 06/执行后将 \$ 04, \$ 05, \$ 06 放入了 \$ 1004、\$ 1005、\$ 1006 单元。

(4) 传送一段内存区域的内容

有时需要将存储器中某一段程序或数据,从某一段内存区域搬到另一段内存区域,可用这个命令。其格式是: * 目的地址<源地址 1. 源地址 2M/,例如,将 \$ 2000 到 \$ 3FFF 单元中的内容传送到 \$ 4000 到 \$ 5FFF 中,则可用:

* 4000<2000. 3FFF M/

又如,将 \$ 1000 到 \$ 4000 单元都填上 00,则可

用:

* 1000;00✓

* 1001<1000.3FFF M✓

(5)比较两段内存区域的内容

对内存中两个长度相同的数据块内容进行比较,考察是否一致。当我们想比较一下两个相似程序是否完全一样时,就可以使用这个命令,其格式是:

* 地址 1<地址 2.地址 3V✓

命令执行的结果是将地址 2 到地址 3 之间的数据,与从地址 1 开始的长度相等的地址范围内的进行比较。如果一致,则退出该命令,并出现提示符*;否则就显示相应地址和数据。

(6)检查和修改寄存器暂存区内容

其格式是:* CTRL-E✓,命令执行后屏幕上顺次显示 A,X,Y,P,S 五个寄存器在零页的暂存单元(\$0045~\$0049)的内容。如:

* CTRL-E✓

A=43 X=27 Y=00 P=A0 S=F6

这时修改当前各寄存器暂存单元的内容:

*:A1 A2 A3 A4 A5

* CTRL-E✓

A=A1 X=A2 Y=A3 P=A4 S=A5

用 G 命令时,监控程序首先将这五个暂存单元的内容分别送入相应各寄存器,然后才进入 G 命令的首地址运行程序。

(7)反汇编命令

汇编语言程序是用指令助记符编写的程序,经过汇编后被转换成二进制的机器语言程序代码,使计算机可以认识和执行。但是,这种二进制代码不易记住,即使对指令代码相当熟悉的人,一一翻译它们也有困难。在调试这种二进制代码的机器语言程序时,人们常常希望看到它们的指令助记符,以便更好地了解程序在做什么。反汇编命令就提供了这种功能,它把内存中的二进制代码按照地址顺序一一列出来,而且还在旁边标出相应的助记符命令,这样一看就明白。例如,已知一段机器语言程序,它存在 \$300 到 \$30C 一段存储区域:

0300-A9 C1 20 ED FD 18 69 01

0308-C9 DB D0 F6 F0

执行 * 300L✓命令后,屏幕上显示如下:

```
0300 A9 C1 LDA # $C1
0302 20 ED FD JSR $FDED
0305 18 CLC
0306 69 01 ADC # $01
0308 C9 DB CMP # $DB
030A D0 F6 BNE $0302
030C 60 RTS
030D 00 BRK
:
0313-00 BRK
```

这个程序就十分明确了,首先将 A 的代码 C1 放进累加器,调用输出一个字符子程序,其入口地址为 \$FDED,然后清进位位,累加器内容加 1,判断是否为 \$DB(Z 的代码是 \$DA),不是跳转 \$0302 继续操作,是则返回调用程序。由此可知,本程序可以在屏幕上显示 26 个英文字母。

所以,* 地址 L✓的操作是把指定地址开始的 20 条机器语言指令翻译(反汇编)成符号指令,并显示在屏幕上。而 * L✓,则是将从当前地址开始的 20 条机器指令反汇编成符号指令并显示出来。

(8)运行机器语言程序命令

用汇编语言编好了程序,经汇编后将目标程序装入内存,你就可以用下述命令来执行程序。如:

* 1000 G✓即从 \$1000 单元开始执行程序。

* 1000 S✓,这是单步命令,执行从 \$1000 单元开始的一条指令,并显示寄存器值和指令助记符,例:

1000 A9 43 LDA # \$43

A=43 X=CE Y=D8 P=30 S=EC

* 1000T✓,这是一种跟踪命令,它把多个单步操作连起来做,直到执行完一条 BRK 指令为止。

此外,还有屏幕显示方式命令,选择输入、输出设备命令等,如:

* I✓,置屏幕为反相显示方式。

* N✓,置屏幕为正常显示方式。

* 1^P✓,启动打印机,显示打印结果。

* 0^P✓,停止打印,回到屏幕显示。

* 6^P✓,转入磁盘驱动器

* 3^P✓,进入中华学习机中文状态。

* 0^K✓,进入以键盘为输入设备的状态。

* 地址 1. 地址 2 R✓,将磁带上的数据读入指定的内存单元。

* 地址 1. 地址 2 W✓,将内存地址 1 到地址 2 中的信息写到磁带上。

* 300L 300G✓,是多重命令,反汇编并执行从 \$300 开始的机器语言程序。

四、编辑和汇编程序 LISA 简介

编辑和汇编程序 LISA,是 6502 汇编程序中比较适合初学者的一种工具软件,其特点是:能即时提示输入语法错误;汇编过程在内存中进行,不读写磁盘,速度快;结合 16K RAM 卡或语言卡,一次可汇编 4K 字节的目标代码。其不足之处是编辑功能没有 TOOLKIT 的 EDASM 强,目标码存盘时要用户算出实际代码长度,用 DOS 命令写入磁盘。LISA 有 15 条操作命令,33 条伪指令,操作码助记符除与 6502 指令一致外,还扩展了 5 条:BTR(相当于 BEQ),BFL(BNE),BGE(BCS),BLT(BCC)和 XOR(EOR),操作命令有 I(插入),D(删除),M(修改),L(列程序),LO(读入源程序),SA(源程序存盘),A(汇编),N(清源程序),BRK(进入监控),F(查找标号)等等。

数据库基础

北京电脑天地学校(100051) 阎 璐

习 题

一、从供选择的答案中,选出应填入下面_____

内的正确答案。

1. 内模式用_____语言定义。
A. 设备介质 B. 模型介质 C. 结构介质 D. 外型介质
2. 概念模式用_____语言定义。
A. 集合定义 B. 查询定义 C. 运算定义 D. 模式定义
3. 外模式用_____语言定义。
A. 关系模式定义 B. 层次 C. 子模式定义 D. 操纵
4. 下面_____不属于数据模式。
A. 查询模式 B. 外模式 C. 概念模式 D. 内模式
5. 用树型结构来表示实体之间联系的模型称为_____
A. 关系模型 B. 层次模型 C. 网络模型 D. 运算模型
6. 用表格数据来表示实体之间联系的模型称为_____
A. 关系模型 B. 层次模型 C. 网络模型 D. 运算模型
7. 关系数据库的任何检索操作的实现都是由三种基本检索运算组合而成的,这三种基本运算不包括_____
A. 选择 B. 投影 C. 关闭 D. 联结
8. DML 是_____语言,又是_____语言;DDL 是_____语言,又是_____语言。
A. 非过程化 B. 过程化 C. 系列化 D. 非系列化
E. 数据操纵 F. 数据选择 G. 数据运算 H. 数据定义
9. 关系模型的数据语言是_____的语言,其核心部分为查询,因此又称为查询语言。
A. 非过程化 B. 过程化 C. 系列化 D. 非系列化
10. 在数据库 dBASE III 中,一个过程文件中允许最多包含的过程为_____个。
A. 256 B. 28 C. 32 D. 128
11. 数据操作语言一般不具有的功能_____。
A. 从数据库中检索数据 B. 数据收集器和集中操作
C. 向数据库中添加数据 D. 用于并发访问控制的操作
12. 以下说法哪个不正确_____。
A. 数据库减少了数据冗余 B. 数据库避免了一切数

据重复 C. 数据库避免数据的不一致 D. 数据库中的数据可以共享

13. 数据库管理系统和操作系统之间的关系是_____。
A. 为实现数据保护,数据库管理系统通常提供了保证数据 B、C 及并发控制等方面的机制。数据库系统采用的数据模型有 D 为使用数据库方便,常常将数据库管理系统提供的数据库操作语言嵌入到某一高级语言中,称此高级语言为 E。

供选择的答案

- A ①数据库管理系统调用操作系统
②操作系统调用数据库管理系统
③两者互相调用
B、C: ①随机性 ②有效性 ③完整性 ④安全性
⑤相容性 ⑥顺序性
D: ①网状模型,链状模型和层次模型
②层次模型,分散模型和环状模型
③层次模型,网状模型和关系模型
E: ①虚拟 ②会话 ③宿主

二、从下列叙述中选出 5 条正确的叙述。

1. 用数据描述语言精确地定义数据模型的程序称为模式。
2. 数据库系统是一组软件的组合,可应用于各种程序。
3. 在 DBF 文件中存在记录的情况下,使用 GO BOTTOM 命令肯定能使记录指针指向记录号最大的记录。
4. 在数据库中,对数据(结构)的描述称为数据模式。
5. 用户使用的数据视图叫外模型。
6. 全局的逻辑数据视图叫概念模型,又叫数据模型。
7. 数据库管理系统可以用于各种应用程序。
8. 数据库系统即数据库管理系统,是由硬件、软件、用户等组成的。
9. 物理数据存储的模型叫内模型。
10. 在数据库系统中,数据一致性系指数据库中的数据类型一致。

三、现有一个数据库 ABC. DBF,库中有 10 条记录,请写出顺序执行下面各命令后库指针的位置。

- . USE ABC
. GO 5
. SKIP-3
. DISPLAY

. LIST NEXT 4

. GO BOTTOM

四、在下述程序中

IF a<b

LOOP

SKIP

ENDIF

语句 SKIP 是永远得不到执行的(是否正确)。

五、有一个已经打开的 DBF 文件,文件名为 BH-

SL. DBF,用 LIST 命令所显示的结果为:

RECORD #	编号	数量
1	A1	10
2	A0	85
3	A2	67
4	A10	50
5	A12	65

(注:“编号”字段为 C 型,“数量”字段为 N 型)阅读下列程序段:

SET TALK OFF

CLEAR

USE BHSL

a=数量

DO WHILE .NOT. EOF()

IF a<数量

a=数量

ENDIF

SKIP

ENDDO

? 编号,a

此程序执行后,屏幕中所显示的内容为()

供选择的答案:

(1)A0 85 (2)A1 10 (3)85 (4)10

习 题 分 析

1. 内模式(或存储模式)又称物理模式,是对数据库的存储结构等方面内容的描述,用物理模式定义语言来进行描述,有些系统,把描述存储模式的语言称为设备介质控制语言,或数据存储描述语言。

2. 概念模式简称为模式,是对数据库的全局逻辑结构的描述,用模式定义语言来进行描述。

3. 外模式又叫子模式,是对数据库的局部逻辑结构的描述,具有相同数据视图的用户共用一个子模式,用子模式定义语言来进行描述。

4. 在数据库中,对数据(结构)的描述称为数据模式。数据视图被划分为三个层次,那么对数据结构进行描述的数据模式也分为三个层次。与外部层,概念层和内部层相对应,数据模式分为外模式、概念模式和内模式。这是 ANSI(美国国家标准化协会)模型的划分方法。

5. 数据库的特征之一是数据库中的数据是有结构

的,即体现了数据之间的联系,因此数据库系统必须研究记录间的联系。所谓数据模型就是指具有这种联系的数据结构形式。

在格式化模型中,用树的数据结构形式来表示实体及其联系即为层次模型,这种树由结点和连线组成。结点表示实体集,而连线表示相连两个实体之间的关系。

6. 用表格数据来表示实体间联系的模型叫关系模型。关系模型是数学化的。它把数据看成二维表中的元素,而这个表就是关系。

7. 关系型数据库管理系统所必须具有的三种操作是:选择、投影和联结。所谓“选择”就是从 n 维空间内的所有点中选择出满足一定条件的那些点,而除出其它点。例如工资高于 100 元的职员表。通常包括指定范围选择和指定条件选择两种。“投影”操作是指只对记录类型中的某些项进行数据操作。而“联结”操作指的是将两个库文件按一定的条件联结成一个新的库文件。

8. DML(Data Manipulation Language)是数据操纵语言,它是数据库管理系统提供给应用程序员用以存储、检索、修改、删除数据库中数据的工具。DML 是过程化语言。

DDL(Date Description Language)是数据定义语言,它是数据库中用来描述数据及其关系的语言。是数据库管理软件的一部分。具体描述数据库管理系统中的数据结构和数据库的逻辑特征。描述的对象包括初等项、组项、记录等。DDL 是非过程化语言。

9. 关系数据库使用的语言一般具有定义、查询、更新和控制一体化的特点。它即可以嵌入到宿主语言,又可作为独立的交互语言使用;关系模型的数据语言是非过程化的语言,其核心部分为查询,因此又称为查询语言。

10. 在 dBASE III 中,一个命令文件可以包含过程也可以不包含过程,但过程总是包含在某个命令文件中的,包含过程的命令文件又称为过程文件,一般允许一个过程文件中最多含 32 个过程。

11. 数据操作语言适应于 DBMS 的数据模型,对于不同的数据模型,数据操作语言是不同的。数据操作语言一般应具有如下功能:

(1)从数据库中检索数据,这是最重要、最经常用的一类操作。

(2)向数据库中添加数据。

(3)删除数据库中某些已经过时,没有保留价值的原有数据。

(4)修改某些属性发生了变化的数据项的值,使之能确切反映变化后的情况。

(5)用于并发访问控制的操作。

12. 数据库管理系统技术的特点是:

(1)减少数据的冗余;

(2)避免数据的不一致;

(3)数据可以共享;

- (4)有数据安全和完整性保障及并发控制;
- (5)多用户操作的并行调度;
- (6)易于使用;
- (7)便于扩展

13. 操作系统是对计算机硬件的第一级扩充,它是系统软件,是其它软件的运行基础,因此数据库管理系统和操作系统之间的关系是操作系统调用数据库管理系统。根据 12 题解释的数据库管理系统技术的特点可以说明,为实现数据保护,数据库管理系统通常提供了保证数据完整性、安全性及并发控制等方面的机制。

数据库系统的一个核心问题就是研究如何表示和处理实体间的联系。人们把表示实体和实体之间联系的模型叫数据模型。

常用的数据模型有三种,层次模型、网络模型和关系模型。也就是通常人们所说的层次数据库、网状数据库和关系数据库。

为了克服自含系统在数据处理能力方面的不足,可以将数据库管理系统提供的数据库操作语言(DML)嵌入到某一个高级语言中去。这个高级语言称为宿主语言。整个系统就称为宿主系统,宿主型的数据操作语言(DML)是过程化语言。

答案: 1. A 2. D 3. C 4. A 5. B 6. A 7. C
8. E, B, H, A 9. A 10. C 11. B 12. B 13. A② B
③ C④ D③ E③

二

1. 正确
2. 错误。数据库系统是指计算机系统中引进数据库后的系统构成。构成数据库系统至少有四个成份,即数据、硬件、软件 and 用户。

3. 错误。此题没有明确指出用户打开了什么类型的文件,只是说数据库文件有记录,如果题目明确指出打开的是数据库文件而没有打开其它文件,可以说是正确的;但是如果此题还打开了索引文件,则题目说法是不对的,因为对数据库中不同的字段进行索引,记录号排列的顺序是不一样的。因此说本题目的答案是错误的。

4. 正确。
5. 正确。
6. 正确。

7. 错误。数据库系统主要不是用来作科学计算,它主要用来处理各种报表,是管理现代化和办公自动化的得力工具之一。

8. 错误。数据库系统和数据库管理系统之间的关系是数据库系统包含有数据库管理系统。

9. 正确。

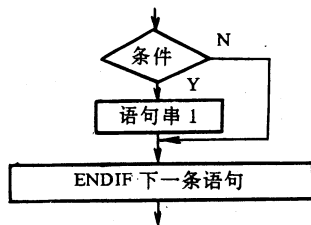
10. 错误。数据库系统中数据的一致性是指指在数据库系统管理数据(修改、删除或增加数据)时,数据的一致。

三、题目明确指出打开的是数据库文件,因此题目的答案为:1、5、2、2、5、10。

在做指针类题目时,应注意数据库文件记录指针、

排序后生成的数据库文件记录指针和索引文件记录指针指向的区别。数据库文件记录的排列顺序是物理排列,排序后生成的库文件记录的排列顺序是顺序排列,而索引文件记录的排列顺序是逻辑排列,因此它们的首记录和尾记录的记录号可能相同也有可能不同,这要看索引文件是按照什么关键字进行索引的。

四、本题是条件语句结构,执行条件语句的框图



当条件成立($a < b$)时,执行语句串 1,也就是本题目中的 LOOP 语句,而 LOOP 语句是回到循环头的命令;那么当条件不成立($a \geq b$)时,应当执行 ENDIF 的下一条语句。因此 SKIP 语句是永远得不到执行的。

五、程序首先关闭无用信息,清屏,开库,把数量送给 a 内存变量,此时 a 中存放的数字应为 10,进入循环结构,条件为在没有遇到库文件尾时循环,条件语句判断,将最大的数量送给内存变量 a 也就是挑选最大数,最后输出编号和 a。当库文件指针函数为真($EOF() = T.$),也就是退出循环时,指针指向最后一条记录之外,因此编号中没有数据。因此题的答案应为——(3) 85。

(上接第 23 页)

```

6218- FB 90 05 A9 A0 4C 22 62
6220- A9 CD 20 ED FD CA D0 E0
6228- C8 C4 09 90 D5 20 8E FD
6230- D8 18 A5 06 69 28 85 06
6238- A5 07 69 00 85 07 C6 FA
6240- D0 BE 60
  
```

程序 C:

```

10 INPUT "P=";P
15 INPUT "V=";V
20 POKE 6,0;POKE 7,32 * P
25 POKE 252,0;POKE 253,32 * (3-P)
30 CALL 24832
35 INPUT "X1,Y1=";X1,Y1
40 INPUT "X2,Y2=";X2,Y2
45 N1=(3-P) * 8192 + Y1 * 40
50 N3=INT(N1/256);N2=N1-N3 * 256
60 POKE 6,N2; POKE 7,N3
65 POKE 8,X1/7;POKE 9,X2/7
70 POKE 250,Y2-Y1+1;POKE 251,V
75 PR # 1
80 CALL 25088
85 PR # 0;END
  
```



提高单片机最小系统抗干扰能力和自恢复方法

合肥中国科技大学计算中心[230026] 张培仁 刘振安

单片机也象其他计算机一样遵循程序存储和程序控制的原理。最小系统的程序存储在 EPROM 中,单片机接通电源后从 0000H 地址开始执行一条一条指令。这时程序计数器(PC)不断指出目前程序执行到何处。一旦系统受到干扰后,CPU 取指令取的是随机错误拼写起来的指令,从而造成不是按原来程序进行工作,造成程序“跑飞”以至系统死锁。这时往往可冲乱 RAM (包括片内和片外)的数据。最终造成系统失控。这种现象在工业控制系统中特别突出。工业控制系统中,在强电、强磁场环境下工作的单片机系统易受干扰。那么我们如何以较低的成本,用硬软相结合的方法提高系统本身抗干扰能力呢?

在回答这个问题之前,首先要了解一般单片机系统的干扰源主要组成。干扰源由三大部分组成:信号线引入的干扰(电场、磁场、电磁波);电源引入的干扰;地线干扰。

要想提高抗干扰的能力,首先要尽可能防止干扰的产生,尽可能隔离单片机的输入输出信号和干扰信号的联系。可采用光电隔离,一般用 4N25, 4N36 等。见图 1。

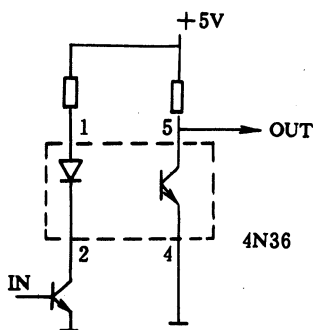


图 1

对于输入是模拟信号,主要采用屏蔽和浮地技术。严格区分模拟地、数字地和系统地,防止数字信号和模拟信号相互干扰。对 A/D 变换器来说,当干扰很大时,应适当增加信号的积分。当采样速度要求在 0.1 秒数量级时,可采用积分式 A/D 变换器。这样抗干扰能力明显提高。对于大功率开关、按键,电磁阀为了减少对单片机系统的干扰,可在强电回路中增加电阻,电容吸收回路,以便减少干扰。

以上这些都是系统注意防止干扰进入用户系统。一旦发生干扰又如何尽快排除干扰而恢复系统正常运行呢?一般是强迫系统复位或进入处理干扰的程序。但是如何知道已经有干扰呢?这里存在着一个如何测试,

分析、判断干扰已经发生的问题。只有判断的正确,才能通过处理,回到相应正常程序去执行。

系统受干扰后发生两种跑飞的现象。一种在用户程序内部转来转去,另一种现象是跳出用户程序而进入系统未使用的程序空间。对于这两种情况都应尽量减少 CPU 执行错误指令的时间。对于后一种情况,判断干扰已经发生是比较容易。例如 KDC-III 开发机中,138 译码器输出 $Y_0 \sim Y_7$,而系统只用 $Y_0 \sim Y_2, Y_3$ 至 Y_7 都未使用,可用硬件接成如图 2。

一旦 PC(程序计数器)脱轨而进入系统未使用区,就会产生中断请求,CPU 响应后进入中断处理,这时 CPU 先找出栈顶,将栈顶存储单元的下一个单元内容清零,此时将 0000H 装入堆栈,中断返回后,0000H 弹回 PC,系统复位。

如果跑飞的程序在用户程序内部跳转,这时将如何识别,判断呢?程序设计时常又是模块化设计,按照程序的要求一个模块、一个模块执行。在执行本模块时,为下一模块设立一个标志码。进入下一模块后再检测该标志码,如果正确就正常执行,如果不正确认为是非正常进入本模块的。这样就象接力棒一样,一级传一级,各模块有自己的标志码。这样程序很容易判断出是否是“跑飞”来的跳转,还是正常跳转过来。关键的是跳转时是否带有本模块的标志码。一旦发现非正常程序在执行,即转到错误处理程序中。

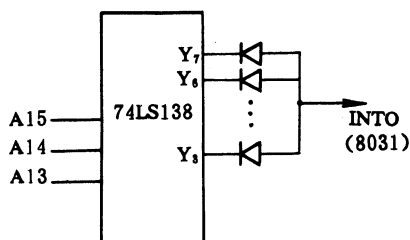


图 2

对这种“跑飞”程序还有另外一个办法解决。就是在用户程序中设置一些陷阱。在正常程序中不执行这些程序,也就是不会进入陷阱。而一旦“跑飞”就有可能遇到这些陷阱,只要遇到一个,马上进行错误处理程序,做出相应的处理。这些陷阱的指令组如下:

```
NOP
NOP
NOP
NOP
AJMP SLG
```

这样的指令组要有多组散放在用户程序各模块之间空余的单元里。这组指令前面是四个空操作,目的使

PC 跑飞后的指令首地址纳入正轨,以保证最后执行跳转指令。这个方法很有效,陷阱的多少一般依用户程序大小而定,一般每 1K 字节有几个陷阱就可以了。

还有一个方法可以很方便提高抗干扰的能力。一般用户 EPROM 常常是用不完的。可以把余下单元全部写成 LJMP 0000H 指令。一旦“跑飞”很快回到初始化程序,不会死锁。

不论哪一种“跑飞”,不论是在用户区还是在未使用空间区跳转时都有可能把 8031 片内和片外的 RAM

破坏掉。这时将如何尽快判断 RAM 已被破坏,并尽快恢复原来的数据呢? 这里主要的方法是:在系统初始化,就在一些用户不用 RAM 区域的几个单元中(最好散列在整个 RAM 空间)放入特定标志数,当 RAM 受破坏时,这些标志将发生变化,读这些标志数与原来设定值比较,即可判断 RAM 是否被破坏。一般在 1K 的 RAM 有几个字节就可以了。

以上介绍几种简单容易实现抗干扰的方法,这些方法将系统可靠性大大提高了。

TP801 单板机与微型机数据双向并行传送一例

江西九江国棉二厂(332006) 刘浔和

利用 TP801 单板机和微型机(笔者使用 BCM-3 型机)中的 PIO 芯片可以组成简单的双向并行传送装置,按图联接 9 根导线即可。

程序 1

```
org 2000h      ;
ld hl,800h     ;数据起始地址
ld c,06h       ;数据长
ld b,00        ;
ld a,4fh       ;置 PIO 为非中断输入
out (83h),a    ;B 口
ld a,07
out (83h),a
tina,(81h)
cp 13h         ;控制数
jr nz,t
tl in a,(81h)
cp 13h
jr z,tl
ld (hl),a
inc hl
djnz t
dec c
jr nz,t
```

程序 2

```
org 2000h
ld hl,800h
ld c,06
ld b,0
ld a,0fh
out (246),a    ;置 PIO 为非中断输出
ld a,07
out (246),a    ;A 口
tld a,13h      ;控制数
out (244),a
ld d,10        ;延时
tl dec d
```

```
out (244),a
jr nz,t1
ld a,(hl)
out (244),a
ld d,06        ;延时
t2dec d
out (244),a
jr nz,t2
inc hl
djnz t
dec c
jr nz,t
jp 0001        ;返回监控
```

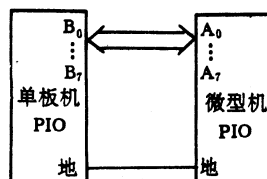


图 1

当微型机向单板机送数时,单板机先执行程序 1,微机再执行程序 2。在程序中,13H 为控制数,应是传送数据中没有出现的数,否则,在传送中会丢失 13H 这个数。另由于单板机与微机的频率不一样,且 PIO 输电平也要达到稳定。因此增加了延时部分,延时量由实际情况来定。

当单板机向微机送数时,应将两个程序中的输入、输出口交换,然后微机先执行程序 1,单板机再执行程序 2。

该方法简化了硬件部分,对于其它的单板机和微机,单板机和单板机及微机和微机,只要装有并行输入、输出芯片,都可按此法传送数据,但要改动程序中的输入、输出口和延时量。

MCS—51 汇编指令外延的应用

湖南怀化地区电子研究所(418000) 陈亿善

在使用 MCS—51 单片机汇编语言编写程序时,经常会遇到工作寄存器 $R_1 \sim R_7$ 之间进行数据传递。而在 MCS—51 的指令系统中没有工作寄存器 R 之间传递数据的指令。例如要把 R_3 的数据送到 R_5 中,如果写成:

```
MOV R5,R3
```

在用 MCS—51 汇编软件汇编时,就会指出这条指令是错的,因为 MCS—51 的指令系统中找不到与这条指令相对应的机器码。在一些介绍 MCS—51 指令和编程技巧的书中,都是采用累加器 A 或其他寄存器中转的办法来实现 R 之间的数据传递,如用 A 中转,上例便写成:

```
MOV A,R3
MOV R5,A
```

上面两条指令虽然可以达到传送数据的目的,但在编写时要多用一条指令,还动用了累加器 A ,如果 A 中原来的数据紧接着在下面程序中用到的话,还得先把 A 中的数据保护起来,中转完后又得调回 A 中,这样还需增加两条指令。笔者在编写程序中发现,MCS—51 指令有些还有很深的外延,如实现 R 之间的数据传递,应用 MCS—51 指令的外延,用一条指令完全可以达到目的,而且无需动用其他寄存器。编写时只要把指令中的两个 R 中的一个改写成该 R 的直接地址即可。如上例(假设 PSW 选择的 R 区在 0 区)可以写成:

```
MOVR5,03H
```

或:

```
MOV 05H,R3
```

这两条指令都可以达到把 R_3 中的数据送入 R_5 中的目的。这种编写方法在各种版本的 MCS—51 汇编软件下汇编,都能生成正确的机器码。编程实践表明,除了使用 R_0 或 R_1 进行间接寻址时, $@R_0$ 或 $@R_1$ 中的 R 不能用其直接地址代替外,其他任何场合 $R_0 \sim R_7$ 都可以用其直接地址来代替。比如在压栈和弹栈指令中,没有把 R 压入堆栈和弹出堆栈的指令,即 $PUSH R_n$ 和 $POP R_n$ (n 为 $0 \sim 7$) 指令在 MCS—51 指令系统中是找不到的,但象 $PUSH 01H$ (等同于 $PUSH R_1$) $POP 05H$ (等同于 $POP R_5$) 之写法在指令系统中是可以找到相对应的机器码的。 R 在不同工作区的直接地址见下表:

RS ₁	RS ₀	R 工作 区位	R ₀	R ₁	R ₂	R ₃	R ₄	R ₅	R ₆	R ₇
0	0	0	00H	01H	02H	03H	04H	05H	06H	07H
0	1	1	08H	09H	0AH	0BH	0CH	0DH	0EH	0FH
1	0	2	10H	11H	12H	13H	14H	15H	16H	17H
1	1	3	18H	19H	1AH	1BH	1CH	1DH	1EH	1FH

(上接 43 页)

STR \$ 语句的格式为:

```
STR $(X)      X 取值—32768~32767
```

例 10.

```
10 I. "A";A
20 I. "B";B
30 P. A+"B"="A+B
40 P. "STR$( "A")+STR$( "B")="";
   STR$(A)+STR$(B)
50 G. 10
```

Run 显示

```
A?      键入 125      回车
B?      键入 135      回车
125+135=260
STR$(125)+STR$(135)=125135
:
```

读者不难发现,通过 STR \$ 语句的处理,原来的十进制数 125、135 虽然外形没有改变,但已发生质的变化,这时已不再是数字而变为字符串。因此,当它们相加时,就按照字符串的运算规则相加了。



学装微电脑

制作能暂停显示的计数器

易齐干

计数器多数是计量输入的脉冲数,连续不断地显示。也有随用途不同,连续计数过程中,有要求暂停显示的情况。例如,检验一定时间的生产量时,计数器的显示有暂停功能,则会带来很大方便。这就是本文讨论的计数器。

1. 硬件安排

显示有暂停功能的计数器必须具备三部份:脉冲输入端;控制显示更新、暂停的开关;脉冲计数结果显示部份。照图1连接 $\mu P-80$ 教育套件。电源部件提供50Hz脉冲送入输入输出部件C口第0位;4位数字显示部件显示微电脑计数结果;C口高位相连的打码开关 S_0 开关(C口第4位)“通”暂停显示;“断”恢复显示。这就是本文讨论的计数器。

2. 软件开发方法

可按如下步骤进行:

- 1)按功能展开课题。
- 2)绘制简单流程图。
- 3)分配RAM区域。
- 4)绘制详细流程图。

图1. 带有显示暂停功能的计数器硬件。

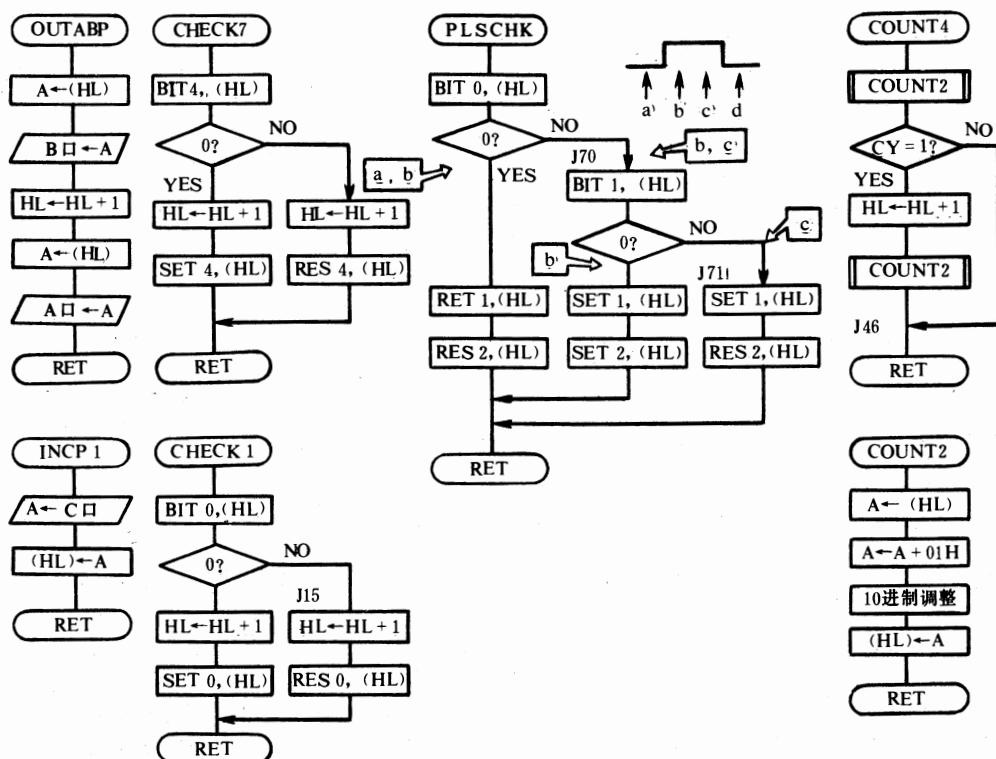
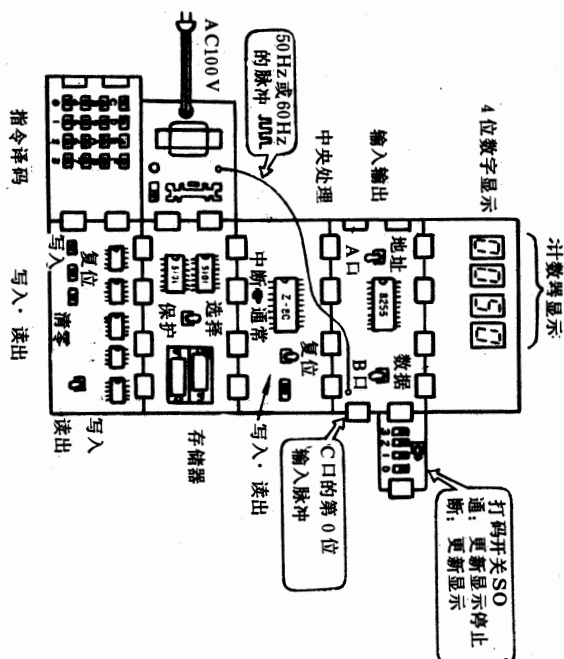


图 2(b)详细流程图

表 1 RAM 区域。

地 址	00E0H	00E1H	00E2H	00E3H
分 配	数据输入	检验结果	计数器低位	计数器高位
初始数据	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0	0 0

开关输入 → CHECK7 = 0:1
≠ 0:0

脉冲输入 → CHECK1 = 0:1
≠ 0:0

这次脉冲状态
前次检验时的状态
0:无效, 1:有效 } PLSCHK

表 2. (a) 程序清单(1)

标号	助记符	地址	机械语言	注 释
START	LD A, 89H OUT (03H), A LD SP, 0100H	0000 0002 0004	3E 89 D3 03 31 00 01	设置输入输出 设置 S
INIT	LD B, 03H LD HL, 00E1H	0007 0009	06 03 21 E1 00	设置初始值
J1	LD (HL), 00H INC HL DEC B	000C 000E 000F	36 00 23 05	
	JP NZ, J1	0010	C2 0C 00	
OUTPUT	LD HL, 00E1H BIT 4, (HL) JP NZ, INPUT INC HL	0013 0016 0018 001B	21 E1 00 CB 66 C2 1F 00 23	输出
	CALL OUTABP	001C	CD 48 001	
INPUT	LD HL, 00E0H CALL INCP1	001F 0022	21 E0 00 CD 50 00	输入
PROCESS	LD HL, 00E0H CALL CHECK7 LD HL, 00E0H CALL CHECK1 LD HL, 00E1H CALL PLSCCHK LD HL, 00E1H BIT 2, (HL) JP Z, OUTPUT LD HL, 00E2H CALL COUNT8 JP OUTPUT	0025 0028 002B 002E 0031 0034 0037 003A 003C 003F 0042 0045	21 E0 00 CD 54 00 21 E0 00 CD 61 00 21 E1 00 CD 6E 00 CB 56 CA 13 00 21 E2 00 CD 87 00 C3 13 00	处理
OUTABP	LDA, (HL) OUT (01H), A INC HL LD A, (HL) OUT (00H), A RET	0048 0049 004B 004C 004D 004F	7E D3 01 23 7E D3 00 C9	4 位显示输出
INCP 1	IN A, (02H) LD (HL), A RET	0050 0052 0053	DB 02 77 C9	数据输入

表 2(b)程序清单(2)

标号	助记符	地址	机器语	注释
CHECK7	BIT 4, (HL)	0054	CB 66	开关输入的检验
	JP NZ, J2	0056	C2 5D 00)	
	INC HL	0059	23)	
	SET 4, (HL)	005A	CB E6	
	RET	005C	C9	
J2	INC HL	005D	23	
	RES 4, (HL)	005E	CBA 6	脉冲输入 的检验
	RET	0060	C9	
CHECK1	BIT 0, (HL)	0061	CB 46	
	JP NZ, J15	0063)	C2 6A 00)	
	INC HL	0066:	23)	
	SET 0, (HL)	0067	CBC 6	
	RET	0069	C9	检测脉冲的一个峰
J15	INC HL	006A	23	
	RES 0, (HL)	006B	CB 86:	
	RET	006D	C9	
PLSCHK	BIT 0, (HL)	006E	CB 46:	
	JP NZ, J70	0070:	C2 78 00)	
	RES 1, (HL)	0073	CB 8E:	10进制 4 位 计数加
	RES 2, (HL)	0075	CB 96:	
	RET	0077	C9	
J70	BIT 1, (HL)	0078	CB 4E:	
	JP NZ, J71	007A	C2 82 00)	
	SET 1, (HL)	007D	CB CE	
	SET 2, (HL)	007F	CB D6	
	RET	0081	C9	
J71	SET 1, (HL)	0082	CB CE	
	RES 2, (HL)	0084	CB 96:	
	RET	0086	C9	
COUNT4	CALL COUNT2	00 87	CD 92 00	
	JP NC, J46	008A	D2 91 00)	
	INCHL	008D	23	
	CALL COUNT2	008E	CD92 00)	
J46	RET	0091	C9	
COUNT2	LD A, (HL)	0092	7E	
	ADD A, 01H	0093	C6 01	
	DAA	0095	27	
	LD (HL), A	0096	77	
	RET	0097	C9	

5) 编制程序清单。

以下逐项进行讨论：

1)本课题归纳为两大功能:脉冲计数;显示计数结果。

脉冲计数包括:脉冲输入;脉冲检验;脉冲计数。

显示计数结果包括:打码开关 ON、OFF 输入;ON

为暂停显示;OFF 为恢复显示。

2)简单流程图的顺序依次如下:设置输入输出;设置堆栈;设置初始值;输出;输入;处理;再返回到输出。输出程序在前目的要显示初始值。

3)由功能展开看出,本课题需要多个子程序,子程序之间数据交换使用 HL 寄存器对通过指定的地址来

市场游戏机十有八九都可

我们这些无线电爱好者，

工作者,很想知道电视游戏机



简易字幕灯的制作与控制

成都西南民族学院(610041) 杨宪泽

机场、车站、城市中心及体育馆的自控字幕,造价以数十万人民币计。本文介绍 TP801A 单板机控制的简易字幕灯,其方法若加以扩充,适宜于一般单位作为较固定的广告和小型娱乐场所使用。

一、灯点阵与控制电路

图 1 是普通白炽灯组成的点阵,涂黑部分代表灯亮时出现的字型“庆元旦”。

图 1 的点阵为 7×7 ,可组成一些简单字。阵列中的每一点译码(一个灯点亮)采用矩阵坐标方式,由二极管 D_1, D_2 构成与门,功率管 VT1 和微型直流继电器组成,电路如图 2 所示(交叉点线路连接一个灯)。

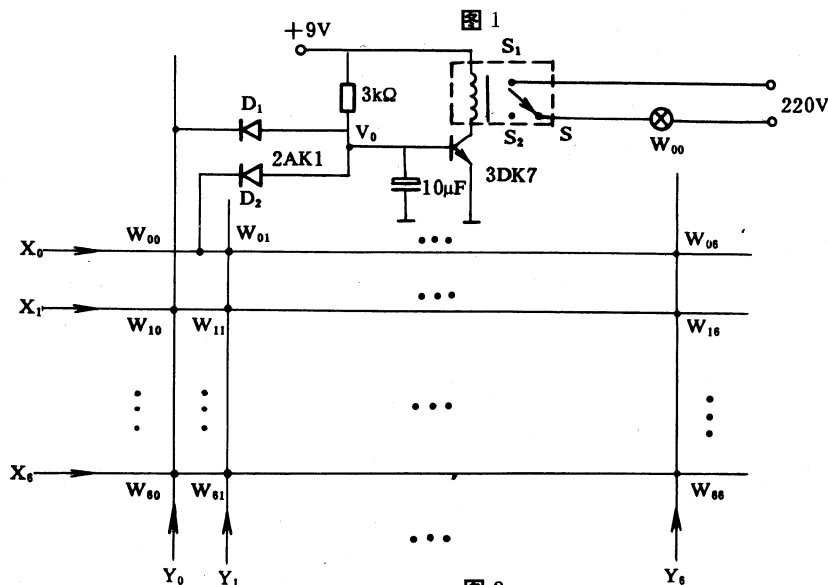
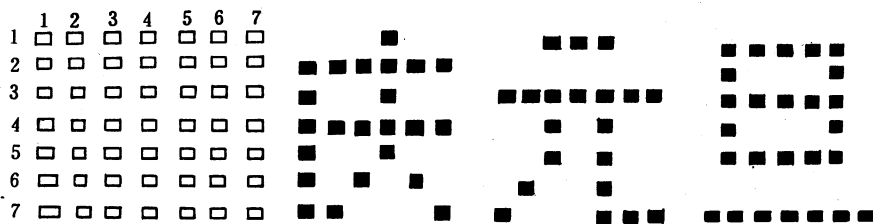


图 2

当需要点亮阵列中 W_{00} 灯时, X_0 和 Y_0 均为高电平约 5V, 即“1”, V_0 为高电压使 VT1 导通, 继电器线圈获足够的电流使触点 S 闭合在 S_1 处, 从而灯 W_{00} 亮; 反之, 只要 X_0 和 Y_0 任一或全为低电平, 即“0”, V_0 为低电压, VT1 截止, 直流继电器 S 回到 S_2 位置, 灯 W_{00} 熄灭。类似的所有点阵上灯由同样电路组成, 原理也相

同。

二、微机控制与程序

灯点阵由 TP801A 单板机控制, 采用单板机上现成的 PIO 接口的 PA 口和 PB 口实施, 如图 3 所示。

图 3 中, 为了增强 PA 口和 PB 口引脚驱动能力, 每一脚加了一级射极跟随器, 然后作为 $X_0 \dots X_6, Y_0 \dots$

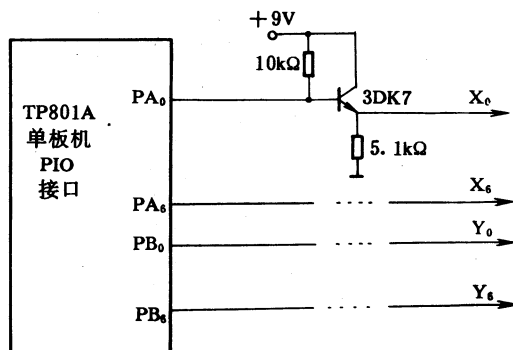


图 3

Y₆ 信号。

下面给出字显示程序,以“庆”字为例,其它编程参照这种方法:

```

ORG 2000H
LD A,00H
OUT (82H),A ;PA 口为输出方式
OUT (83H),A ;PB 口为输出方式
LD C,0FFH
DISPLY
LD A,01H
OUT (80H),A
LD A,08H
OUT (81H),A
CALL DELAY ;延时约 1ms
LD A,02H
OUT (80H),A
LD A,3FH
OUT (81H),A
CALL DELAY
LD A,04H
OUT (80H),A
LD A,09H
OUT (81H),A
CALL DELAY
LD A,08H
OUT (80H),A
LD A,3FH
OUT (81H),A
CALL DELAY
LD A,10H
OUT (80H),A
LD A,09H
OUT (81H),A

```

```

CALL DELAY
LD A,20H
OUT (80H),A
LD A,15H
OUT (81H),A
CALL DELAY
LD A,40H
OUT (80H),A
LD A,23H
OUT (81H),A
CALL DELAY
DEC C ;显示时间为 2 秒,未完继续
JR NZ,DISPLY
JP (显示“元”程序段)
DELAY LD B,0FFH
J1: DEC B
JR N2,J1
RET

```

上述显示程序每一字停留时间约 2 秒,利用分时原理,每一次接通点阵中一排相关点(灯)。由于每隔 7ms 程序再次返回,加之电容 C 的作用,继电器来不及断开电源,人们看到的是整个字 2 秒时间的显示。

字除了这种每两秒变换一个的显示方式外,还可以左、右、上、下移动显示。限于篇幅,本文仅介绍“庆”字向左移动显示的部分程序,其余的部分和其它方式读者不难补充和自行编程(见图 4)

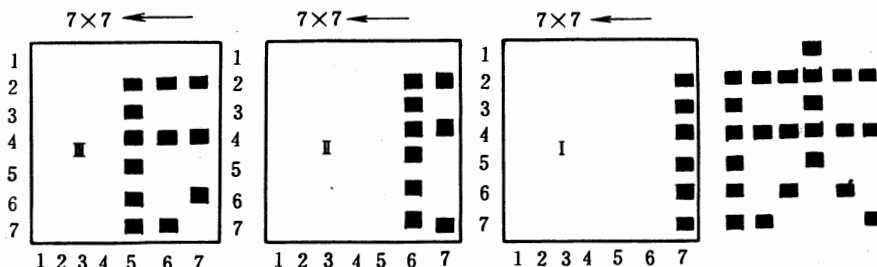


图 4

```

ORG 2000H
LD A,00H
OUT (82H),A
OUT (83H),A
LD C,0FFH
DISPLY1
LD A,40H
OUT (81H),A
LD A,7EH
OUT (80H),A
CALL DELAY
CALL DELAY
DEC
JR NZ,DISPLY1

```

```

LDC,0FFH
LD A,20H
OUT (81H),A
LD A,7EH
OUT (80H),A
CALL DELAY
LD A,40H
OUT (81H),A
LD A,4AH
OUT (80H),A
CALL DELAY
DEC C
JR NZ,DISPLY2

```

```

LD C, 54H
LD A, 10H
OUT (81H), A
LD A, 7EH
OUT (80H), A
CALL DELAY
LD A, 20H
OUT (81H), A
DISPLY3, LD A, 4AH }显示图 4 II 部分
OUT (80H), A
CALL DELAY
LD A, 40H
OUT (81H), A
LD A, 2AH
OUT (80H), A

```

```

CALL DELAY
DEC C
JR NZ, DISPLY3
:

```

这一自制 7×7 阵列字幕灯成本低于 300 元人民币(单板机除外)。若需显示任意汉字,可扩充字幕成 16×16 阵列,同时在单板机上扩充一片 PIO 芯片。作为广告使用的移动字幕,一般需两个或两个以上方阵更好,当然这要增加成本(一般 5 元人民币/点)。此外,选择灯泡瓦数注意继电器触点能承受的电流大小。

DOS 系统下多种软件在一台微机上的共存方法

杭州浙江大学出版社(310027) 尤建忠

随着计算机的应用普及和计算机的不同事务(作业)骤增,计算机所需配置的软件种数也不断增多,在同一台微机上装有两个或更多的大系统已不是什么新鲜事。从计算机的使用角度来讲,一机多用有利于提高计算机的使用效率。但如何在一台机器上有效地配置二种或更多种软件系统,则一是取决于微机硬盘的容量,二是取决于软件系统的安装方法。

举例,我社有一台 ALR Business VEISA 386/33—110 微机,硬盘内存达 106MB,其中硬盘 C、硬盘 D 和硬盘 E 均为 33MB。由于工作需要,在这台微机上要同时配置北大华光 IV 型排版系统和中国印研所的《科印》排版系统,外接 HP II 型激光打印机。对于这么大的两个软件系统(同时需装 64×64 点阵汉字字库),如全装在硬盘 C 上,则因 DOS 系统的限制,硬盘 C 无法装入,即硬盘 C 的容量不够大,如将两个系统分装在硬盘 C 与硬盘 D 上,则在硬盘 D 上的软件系统无法运行。

现用 DOS 命令 JOIN 来解决这个问题。首先,将华光 IV 型排版系统安装在硬盘 C 的子目录 C:\HG 中,而将《科印》排版系统安装在硬盘 D 的根目录中(硬盘 D 的子目录也可)。然后,在硬盘 C 中做一个空目录 C:\KY(必须是空的,即除“.”和“..”文件外,无任何其他文件)。这时,在硬盘 C 的根目录中做一个批命令文件:

```
C>COPY CON:KY.BAT<CR>
```

```

JOIN D: C:\KY
CD KY
ctrl—Z

```

即可。以后每次只要执行 KY.BAT 将硬盘 D 挂靠到硬盘 C 的子目录 C:\KY 上,就可运行《科印》排版系统,并且所有科印排版操作都是在硬盘 D 上进行,结果文件也存在硬盘 D 上。只是一执行过 KY.BAT 后,硬盘 D 就变为硬盘 C 上的一个子目录(C:\KY)。此时若访问硬盘 D,就会得到“非法驱动器号”。而重新启动系统,则会发现硬盘 D 已与硬盘 C 相分离,C:\KY 仍为空目录。

仿效这个例子,我们可编制以下批命令文件:

```

C>COPY CON:XXXX.BAT<CR>
JOIN E:C:\XX
CD XX
ctrl—Z
C>COPY CON:YYYY.BAT<CR>
JOIN F: C:\YY
CD YY
ctrl—Z
.....

```

即可将硬盘 E 上的系统和硬盘 F 上的系统……在 C:\XX、C:\YY……子目录上运行,从而实现多种软件在一台微机上共同存在与使用。



第四章 F BASIC 语言的深入理解

山东苍山机械电子化学工业局(277700) 于 春

前三章介绍了 61 条 F BASIC 的基本语句,其中有 8 条是随机手册中没有的。使用这些语句可以编制一些较复杂的运算程序和简单的游戏程序。但是,当进行较复杂的程序设计,尤其是游戏程序时,仅使用这 61 条语句往往不能达到预期目的。如:使卡通图案在规定的图形中运动,加配音乐使游戏更加生动逼真等都难以实现。可以这样说,前三章为 F BASIC 的入门篇——基础部分;第四章为 F BASIC 的深入篇——提高部分。

本章将重点介绍 F BASIC 的字符处理、出错处理和演奏语句,补充介绍完 F BASIC 的其它语句。在弄通本章语句后,你已经基本掌握了 F BASIC 语言的真谛。那么,再编制程序时,就会随心所欲,水到渠成了。

一、一般语句

1. 收录机读写语句:

收录机读写语句共有三条,分别为

写入语句:SAVE 简写 SA.

核对语句:LOAD? 简写 LO. ?

读出语句:LOAD 简写 LO.

具体操作过程如下:

(1)写入过程:

按说明书联接好信号线和完成录音机准备后,给欲写入的程序起一个名字,如:GAME1",然后键入 SAVE "GAME1"

启动录音机后,按回车键,这时闪烁光标消失,出现 WRITING GAME1

意思为正在写 GAME1。程序写完后,显示 OK 和光标。

(2)核对过程

程序写完后,一般要核对一次,检查程序是否准确地写入磁带。先将磁带倒回到开始处,键入核对语句 LOAD? "GAME1"

回车后,按下录音机 PLAY 键,光标消失,若显示 LOADING GAME1

表示正在核对 GAME1。核对完毕,出现 OK 及光标,表示程序写入正确。这时可 NEW 去 GAME1,再输入欲写入的第二个程序,重复上述操作。若出现 ? TP ERROR 表示无法从磁带中读出程序。这时,要重写一次,再核对,直到正确写入为止。

(3)读出过程:

若欲读出的程序名为 GAME2,而磁带中顺序写入了 GAME1、GAME2……,首先按说明书联接好信号线,键入

LOAD "GAME2"

回车后,按下录音机的放音键 PLAY,若读到程序,屏

幕显示:

SKIP GAME1

LOADING GAME2

当显示 OK 和出现光标时,表示读出完毕,可列出程序,检查读出的程序是否正确。若显示

? TP ERROR

表示无法从磁带中读出程序,必须再读,直到正确地读出程序。

2. 键盘输入语句(LINPUT)

第二章曾介绍了键盘输入语句 INPUT,它只能一次输入一个数据,而 LINPUT 语句可以输入一整行的数据,其中包括各种符号。

LINPUT 简写 LIN.

LINPUT 语句的功能是从键盘上接收整行字符的输入,并把它们当做一个字符串赋给一个串变量。语句格式为

LIN. [提示符];字符串变量

例 1

10 LIN. "A \$ = ";A \$

20 P. "Shu Ru De" A \$

30 G. 10

Run 显示

A \$ = 键入 ABC[]123FGE 回车后,显示:

Shu Ru De A \$ = ABC[]123FGE

注意——输入的字符个数,包括提示符不能超过 31 个。

3. 返回 BG 画面语句(SYSTEM)

SYSTEM 简写 S.

SYSTEM 语句的功能是在对话进入 BASIC 状态(BS 状态)时,返回到 BG GRAPHIC 画面以进行背景图案的设计。

注意——在直调 BASIC 状态时该语句的功能是打开驱动器系统。由于一般不配备磁盘驱动器,所以该语句在直调 BASIC 时无效。

4. 注释语句(REM)

REM 语句的功能是对程序进行说明或注释,以使程序容易理解,便于阅读。REM 语句可以编在程序中,注释的内容可以是任何文字或字符,甚至可以是 BASIC 的语句。程序运行时,并不执行 REM 语句。因此 REM 语句为非执行语句。

注意——REM 语句后的注释或叙述不能超过 255 个字符。

5. 功能键控制语句(KEY、KEYLIST)

(1)设定功能键语句(KEY)

F BASIC 系统提供了 8 个功能键 F1~F8,这 8 个功能键可由用户任意设定以代表某种意义。如在编制程序前,将本程序常用的语句定义给功能键,在以后的编程当中,就可以直接使用,以提高程序的输入速度。

KEY 简写 K.

语句格式为

KEY N,“字符串”

式中 N 的功能键 F1~F8 的号码。

注意——每个功能键所代表的字符数不能超过 15 个。

(2)显示功能键定义语句(KEYLIST)

KEYLIST 简写 K. L.

其功能是将功能键 F1~F8 的定义显示于屏幕上。

例 2.

10 P. “Yuan Ding Yi;”

20 K. L.

30 K. 1, “DEF SPRITE”

40 K. 2, “POSITION”

50 K. 3, “CLS;SPRITE ON”

60 K. 4, “DEF MOVE C”

70 P. “Xin Ding Yi;”

80 K. L.

90 E.

Run

Yuan Ding Yi;

F1 GAME 0 (M)

F2 GAME 1 (M)

F3 GAME 2,1(M)

F4 GAME 3(M)

F5 SPRITE

F6 PRINT

F7 LIST

F8 RUN(M)

Xing Ding Yi

F1 DEF SPRITE

F2 POSITION

F3 CLS;SPRITE ON

F4 DEF MOVE C

F5 SPRITE

F6 RPINT

F7 LIST

F8 RUN(M)

由于 F BASIC 系统已将 F1~F4 四个功能键定义为调用 GAME0~3 四个游戏程序,因而,在直调 BASIC 状态,可直接按 F1~F4 任一个功能键调入游戏程序并运行。但在进行程序设计时,这四个功能键毫无用处。因此,在编程前将 F1~F4 重新定义为编程中经常用到的语句或字符,以提高编程输入速度。

6. 测试空余单元语句(FRE)

FRE 语句的功能是显示内存单元的空余数量,以

防止输入的程序较长时,内存不能完全容下而丢失了数据。尤其在从磁带中读出程序时,若不了解内存空间。空余数量而盲目读出,就容易造成数据丢失。

语句格式为

PRINT FRE

7. 键盘扫描语句(INKEY \$)

INKEY \$ 简写 INK.

语句功能是对键盘进行扫描,等待读取键入的字符。如果在该语句执行的瞬间,用户按下某一键,则该键的字符就会赋给 INKEY \$。这样,程序的运行就可根据 INKEY \$ 的取值决定转向。语句格式为

INK. (0)

例 3.

10 A \$ =INK. (0)

20 P. A \$

30 IF A \$ = “Z” T. P. “OK”

40 IF A \$ = “A” T. P. “NOT”

50 IF A \$ = “X” T. E.

60 G. 10

Run

按 Z 键则打印 OK;按 A 键则打印 NOT;按 X 键则结束。

8. 进制转换语句

(1)十进制数转换十六进制语句(HEX \$)

HEX \$ 简写 H.

HEX \$ 语句的功能是将十进制数转换为十六进制数。语句格式为

H. (X) X 取值: -32768~32767

例 4.

10 F. I=100 TO 200

20 P. I. “D=&H” H. (I),

30 N.

(2)十六进制数转换十进制

十六进制数转换为十进制,随机手册中给出了 VAL 语句。实际上,使用 PRINT 语句就可以把输入的十六进制数转换为十进制数输出。(有关 VAL 语句的功能,下文介绍)

如,P.&HFF 回车后打印 255

P.&H7FFF 打印 32767

例 5.

10 F. A=&HAF TO &HFF

20 P. A,

30 N.

Run 则打印 175~255

二、画面控制语句

1. 测试卡通运动方向语句(VCT)

在程序运行中,VCT 语句可以测试出卡通图案的运动方向,根据卡通的运动方向决定程序运行的转向。VCT 语句在随机手册中没有介绍,请读者自己添上。

语句格式为

VCT(n) n 为被测试卡通的编号

例 6.

```

10 SP. O. : CLS
20 DE. M. (0)=SP. (0,RND(8),1,250,0,0)
30 POS. 0,120,120;M. 0
40 LOC. 0,0;P. VCT(0);PAU. 100
50 G. 20

```

Run

卡通图案的运动方向随机产生,在座标为(0,0)处不断打印出 SP 的运动方向 0~8。

2. 测试 BG 画面的数值语句(SCR \$)

SCR \$ 语句的功能是求出 BG GRAPHIC 画面中所显示的文字或图案的数值。语句格式为:

```
SCR $ (X,Y,n)
```

式中 X,Y 为测试画面的座标。n 为该处的配色号码 0 或 1,也可以省略。

注意——在随机手册中给出了 SCR \$ 的简写为 SC., 这只有在对话 BASIC 状态才有效,而在直调 BASIC 中,SC. 是 SCREEN 的简写。所以,在直调 BASIC 中,SCR \$ 没有简写,请读者更正,以免输入程序出错。

SCR \$ 语句可以求出指定点的 ASCII 码。有了这条语句,就可以判断卡通图案在任一位置时的背景图案是什么,以便于转向控制。

例 7.

```

10 F. I=0 TO 100
20 P. CH. (200); CH. (208);
   CH. (215);CH. (221);CH. (232);
30 N.
40 F. K=0 TO 20 ST. 2
50 P. ASC(SCR $ (K,K))
60 N.

```

Run

打印 11 个点的图案的 ASCII 码。

三、跟踪语句(TRON、TROFF)

TRON 简写 TRO.

TRON 语句的功能是对程序的运行位置进行跟踪。在跟踪状态下,程序在执行中会不断显示正在执行着的语句的行号。对程序的运行进行跟踪的目的是为了对程序进行调试和查错或了解程序执行的中间结果。

TROFF 简写 TROF.

TROFF 语句的功能是关闭跟踪。

TRON、TROFF 两个语句配合使用,可任意插在程序的任意位置,打开或关闭跟踪,以了解程序的运行情况。

例 8.

```

10 REM"TRON, TROFF"
20 TRON
30 P. "TRON";
40 F. I=0 TO 100
50 TROFF;P. "TROFF"

```

```

60 P. I" * " I" =" I * I,
70 N.

```

80 TRON

90 P. "Yan Shi Jie Shu"

100 E.

Run 显示

```
#30 TRON # 40 # 50 TROFF
```

```
0 * 0 = 0      TROFF
```

```
1 * 1 = 1      TROFF
```

```
⋮              ⋮
```

```
99 * 99 = 9801
```

```
100 * 100 = 10000 # 90
```

```
Yan Shi Jie Shu
```

```
#100
```

四、字符串处理语句

1. VAL

VAL 语句的主要功能是将字符串型的数据转换成十进制数。即 VAL 具有去掉字符、引号、\$ 符号的功能。经 VAL 语句处理后的数据可以参加数值运算。随机手册介绍 VAL 语句的功能是“将十六进制数转换成十进制数”,语句功能介绍不完全。本章在进制转换语句的介绍中已经讲了可用打印语句 PRINT 把十六进制数转换为十进制数。因此,VAL 语句的主要功能是将串型数据转换为十进制数。

语句格式为

```
VAL(字符串)
```

例 9.

```

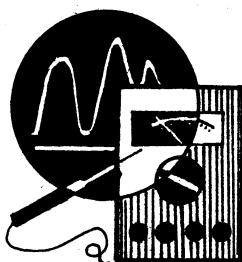
10 REM "VAL"
20 I. "Shu Ru Zi Fu Chuan";A $
30 A=VAL(A $)
40 P. "A $ ="A $,"VAL("A $")="A,
50 G. 20
Run 显示
Shu Ru Zi Fu Chuan?   键入 34ABC
A $ =34ABC      VAL(34ABC)=34
Shu Ru Zi Fu Chuan?   键入 7AB38
A $ =7AB38      VAL(7AB38)=7
.....          键入 GOOD
A $ =GOOD      VAL(GOOD)=0
.....          键入 AB3214
A $ =AB3214      VAL(AB3214)=0
⋮

```

通过演示,读者可以发现,VAL 语句就是保留字符串开头的十进制数,后面的字母等一概略去。若第一个字符不是数字则其值为 0。

2. STR \$

STR \$ 语句的功能是将十进制数转换成相应的字符串。随机手册中介绍 STR \$ 语句的功能是“将十进制数转换成十六进制数的字符串”的叙述是错误的;在该语句的说明中,说该指令用法与 VAL 相反”也是错误的。请读者更正。(下转 34 页)



维修经验谈

TH3070 打印机常见故障分析与检修(上)

国务院办公厅秘书局技术处(100017) 刘立华

TH3070 点阵式打印机是一种 24 针可打印汉字的打印机,目前被广泛采用,是一种优选机型,现针对此类型机的常见故障进行分析与实例检修。

表 1 功能代码表

代 码 符	十进制表示	十六进制表示	功 能
HT	09	09	横向制表命令
LF	10	0A	新行(换行+回车)
VT	11	0B	纵向制表命令
FF	12	0C	换页命令
CR	13	0D	回车命令
SO	14	0E	横向扩展一倍命令
SI	15	0F	解除横向扩展
DC1	17	11	选择使打印机联机
DC3	19	13	解除选择
CAN	24	18	清除当前行缓冲器中的数据
ESC1	2749	1B31	水平表位设定
ESC2	2750	1B32	水平表位清除
ESC5n ₁ n ₂ n ₃	2753 ...	1B35 ...	垂直表位设定
ESC6	2754	1B36	垂直表位清除
ESC A	27 65	1B 41	链式走纸模式设定
ESC C	27 67	1B 43	滚式走纸模式设定
ESC E	27 69	1B 45	ELITE 字型设定
ESC Hn ₁ n ₂ n ₃	27 72 ...	1B 48 ...	页长设定
ESC In ₁ n ₂	27 73 ...	1B 49 ...	图象打印
ESC N	27 78	1B 4E	PICK 字型设定
ESC Un ₁ n ₂	27 85 ...	1B 55 ...	换行单位设定
ESC Vn ₁ n ₂ n ₃ n ₄	27 86 ...	1B 56 ...	反向走纸
ESC Wn ₁ n ₂ n ₃ n ₄	27 87 ...	1B 57 ...	正向走纸
ESC Zn ₁ n ₂	27 90 ...	1B5A ...	空列设定
ESC <	27 60	1B3C	双向打印
ESC >	27 62	1B3E	单向打印

一、功能代码表

• 44 •

TH3070 打印机所能执行的控制功能简表如表 1 所示。

二、TH3070 点阵式打印机常见故障分析与检修

不论是计算机还是其外部设备,在机械结构方面发生故障的概率是很小的,这种故障即便发生亦能直观地看出,在条件允许的情况下,也比较好排除。而电路部分的故障,就没有机械方面那样简单直观,由于这种故障隐蔽在整个电路的某一部分,要想找到这方面的故障并排除掉它不仅要求维修人员具备一定的电路知识水平,而且还要求维修人员对其所维护的设备有所了解,如该设备的工作原理,所使用的器件及各器件的功能等。

下面,我们就 TH3070 的一些常见故障,做一些简单说明,并给出故障检查的流程图。

【故障实例一】走纸异常故障

在打印正常的情况下,打印机不走纸,致使打印字迹成一条黑印,这种故障,除了使用者在拆装机时,把打印辊装反,以及在安装时未能正确地插入 PJ12 插座外,一般都是出在电路部分的器件损坏上,这种损坏所导致的结果就是走纸电机缺相,走纸电机和字车电机一样,是四相步进电机。四相步进电机缺一相或几相,必然会使“步进”(换相)失去作用,从宏观上看,就是打印机不走纸。图 1 为这一故障的检修逻辑图。

在检修流程图中,检查某一器件是否正常,可以用示波器也可以用逻辑笔检查该器件的输入与输出状态,看是否与逻辑关系符合,即可判定其好坏。在走纸故障中,一般最容易坏的部分,除了机电本身外,就是 74LS06(IC44)。另外,在更换元器件时,如无吸锡工具,最好用剪刀先将该芯片的所有引脚剪断,再一个脚一个脚的焊下,否则容易损坏电路板。

【故障实例二】回车撞“墙”故障

这种现象表现为当打印机加电后字车回到左界,仍然不停止、继续向左墙板撞击,引起电机发出怪叫声,检查这种故障的流程图如图 2 所示:

在流程图中“检查 2764、8155”这一框,是要先检查 2764,因为打印机的字库型和普通型所用的控制程序是不一样的,二者不能通用,在有两台以上机器的单位,可通过交换 2764 芯片来检查 2764 的好坏,如果确实证明了 2764 芯片无问题,而 8155(IC27)的 33 脚有信号变化则必然是 8155 芯片损坏,换成好的,即可排除故障。不过这种故障多是发生在开关位置的左移动上,把左界开关适当换成好的,向右移动一下,这种故障即可排除了。

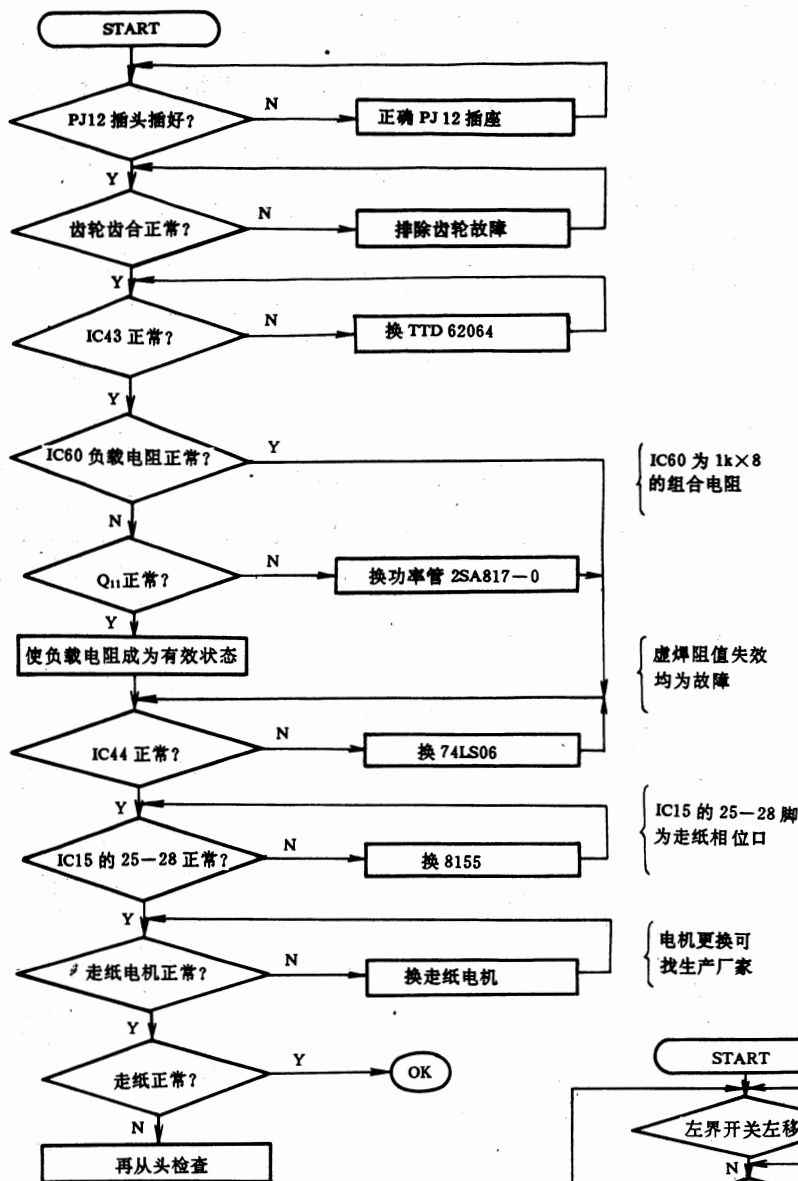


图 1

【故障实例三】回车后不自检故障

这种故障的表现是打印机加电后字车回到左界，但在按下“测试”(TEST)键时，打印机无打印动作，且在“联机”键按下时变无联机灯亮，这种故障，多是出在打印机控制程序 2764 装错或 2764 本身出了故障，若是后者可由厂家对三片 2764 重新写一次即可，这种故障的检修逻辑图如图 3 所示。

【故障实例四】印字漏点故障

打印漏点，是打印机最常发生的故障，其表现形式为打印的字形中缺点(一点或多点，它虽然不影响打印机的使用，但却使打印出的字形失真，影响美观，这种

故障，或者是由于电路内部的器件失效而造成，或者是由于打印头上线圈失效，针断、针短造成，或是由这两方面的原因造成的，下面是这种故障的检修逻辑图，如图 4 所示。

在图中，虚框中“确定不出针的位置”，实际上可由打印时漏点的位置确定下来，也可拆下打印头从针头上数出来。打印头上的打印针是这样排列的：打开打印头的后盖，从中线分两半，右边一半自上而下分别为第 2 针、第 4 针……第 24 针，左边一半分别为第 1 针至第 23 针，如图 5 所示。

打印针直径 0.2mm，

点中心距：纵 = 横 = 0.141mm

点覆盖率：20%

两列针跨距：8 列 = 1/20 寸

图 5 左图中的数字，代表的是衔铁排列的顺序(衔铁下面是线

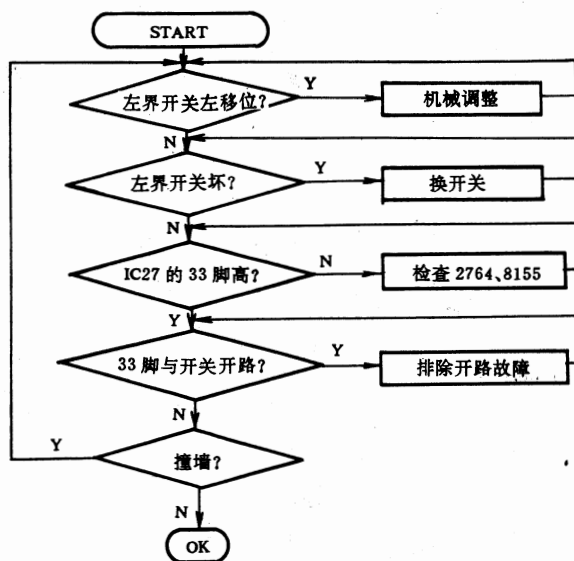


图 2

圈),它与右图的打印针的数字是一一对应的(打印针与衔铁线实际上是焊接在一起的),由此,也可以看出,打印机在打印时,偶数针先打,奇数针再到位补打,使字形成为完整的一列。

在确定打印头无问题,进行电路检查时,实际上就是检查 TD62064 芯片和 74LS374 芯片,如第 13 坏,检查 IC56 和 IC64 即可找出。这二个芯片必有一个已坏了。

若是打印机针变短或者已断,则需要换针,

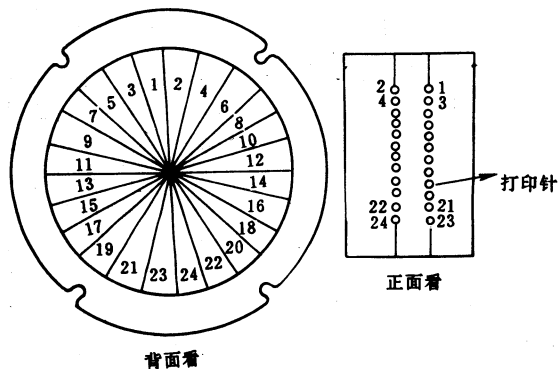


图 5

换针是一项有难度的工作,除了具备一定的技巧和熟练程度外,还需要有耐心,初换者往往由于开始换不好而失去耐心,乱捅硬压,从而使新换的针还未装进去就被再次弄断,这是换针时一定要注意避免的。

(下转 48 页)

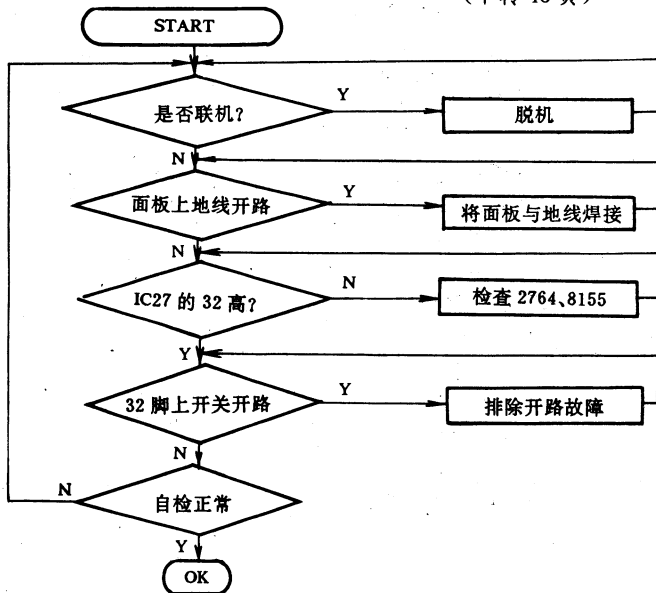


图 3

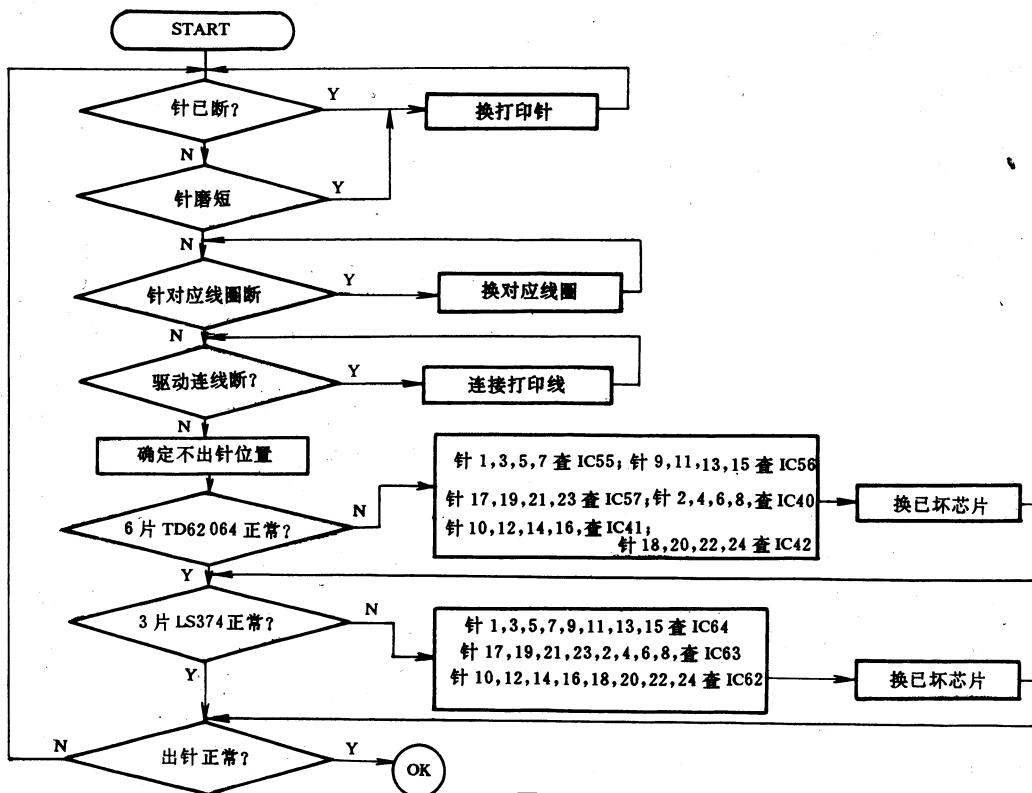
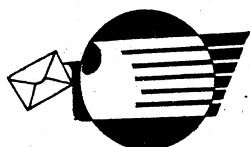


图 4



读者联谊

普及型 PC 个人用户软件交流 联谊活动问题解答(五)

北京中国农科院计算中心(100081) 王路敬

13. 怎样判别 PC 机键盘的好坏?

键盘是向微机提供指令和信息的必备工具之一,它是微机的一个重要输入部件。在 IBM PC 机上,键盘由一条永久性连接电缆,连到系统部件后部的 DIN 插头上,它是一条带屏蔽的四线电缆包括 +5 Vdc、地和两条双向信号线。

PC 机的标准键盘是 83 个键,现在不少 PC 兼容机又增加了一些键,如有 101 个键的等等,但它们的工作原理是一样的,操作方法也基本没有多大区别。键盘作为微机的输入设备,它与其他的外部设备如显示器、打印机、磁盘驱动器等一样,都需要硬件接口和软件接口的支持才能实现真正的功能操作。

PC 机的键盘可以分为三个区:功能键区、打字机键盘区、数字键盘区。键盘上键的功能和作用是由软件定义的,所以在不同的工作环境下,各键的作用就可能不同。键盘上的键好坏或工作正常与否可以利用随机的诊断程序进行检测。只要运行一次诊断程序,就可以对键盘上的每一个键进行检测,帮助查看键盘有没有问题,这种诊断比较方便、简单、易掌握。

使用键盘操作时,切记不要用力过猛,轻轻击打即可。保持键盘清洁、干净。防止灰尘落入键盘。

14. 微机的显示系统包括哪两部分?显示功能有哪两大类?

显示器是微机信息输出的重要设备,也是实现人机对话的主要工具,它既可以显示键盘输入的命令和数据,又可将结果数据变成字符或图形显示出来。它和键盘结合在一起可以方便地进行人机对话。

微机的显示系统包括 CRT 显示器和 CRT 显示适配器两部分。CRT 显示器的屏幕显示原理采用光栅扫描方式,即是在水平偏转和垂直偏转的同步信号控制下,电子束在屏面上从左到右,从上到下有规律地扫描,在屏面扫描期间,电子束强度受视频信号的影响而变化,在水平回扫和垂直回扫期间,由消隐电路抑制电子束,使得回扫线不会在屏幕上显示出来。CRT 显示适配器又称显示卡,是系统和显示器的接口,它的主要功能是控制光栅或显示器。

从显示功能上看可分为两大类:一类是字符显示,另一类是图形显示。虽然字符显示是用点图的方法来显示的,但在结构上是不同的。对字符显示方式,字

点图来自于字符发生器;对图形显示方式,显示缓存的数据位对应着显示屏幕的相应象点,象点的颜色取决于数据位的状态。因此,如果要求显示适配器能支持图形方式,就必须有足够大的显示缓存,以至能容纳屏幕上所有的显示象点。IBM PC 单色显示适配器,显示缓存容量小,只能支持字符显示方式,而 IBM 彩色显示适配器显示缓存达 16K,它不但支持 80×25 字符方式显示,还能支持 320×200 和 640×200 的图形方式。

15. 单色显示器主要性能指标应了解哪些?显示卡的主要部件有哪些?

从使用角度对单色显示器主要性能指标应了解:

- (1) 满屏可显示 25 行 \times 80 列字符。
- (2) 每个字符块大小为 9×14 点阵,字符大小由 7×9 点阵组成。
- (3) 可显示 256 种 ASCII 编码的字符点阵。
- (4) 每个输出字符均有各自的显示属性,如加亮,反视频(白底黑字),下横线等。

以 IBM 单色显示卡为例,主要部件包括以下几部分:

- (1) MC 6854 CRT 控制器芯片,它是整个控制逻辑的核心。
- (2) 字符缓存及属性缓存各 2K 字节。
- (3) 字符发生器。它是一个容量为 8KB 的只读存储器,保存 256 个不同的字符点阵模型。
- (4) 移位寄存器。
- (5) 时钟发生器。包括两个基本时钟,一个象点时钟 DOTCLK,另一个是字符时钟 CCLK。
- (6) CRT 控制器。
- (7) CRT 状态器,反映当前显示器的状态。
- (8) 视频信号合成电路。它是适配器中视频信号处理逻辑部件,也是与 CRT 显示器的接口界面。
- (9) 地址译码电路。

显示控制器的工作过程大致如下:CPU 8088 在显示器水平和垂直回扫期间,把欲显示的字符及其属性送入字符缓冲器和属性存储器;MC 6845 CRT 控制芯片以每秒 50 帧的频率,一面产生行同步、帧同步等信号送到视频信号处理逻辑,一面产生地址去读出缓存器中的字符码及其属性;以缓冲器中读出的字符代码和 MC 6845 CRT 控制芯片提供的扫描线序号为地

址,读出字符发生器中的字型信息,字符的属性则进行一定的译码处理;字型信息读出后送入移位寄存器,然后逐位移出到视频控制器逻辑中,与有关的属性光标等组合处理后,形成显示器所需要的视频输出信号和亮度信号,水平扫描信号直接驱动单色显示器。

16. PC 机系统的彩色图形显示器有哪两种基本工作模式?

PC 机的彩色图形显示器在系统中有两种使用方式,它既可以代替单色显示器作为控制台显示器使用,也可以与单色显示器同时使用。它具有两种基本工作模式:一种是字母数字模式也称 A/N 模式。在这种工作模式下,高分辨率:每帧 25 行,每行 80 个字符;低分辨率:每帧 25 行,每行 40 个字符。

上述两种分辨率既可支持单色字符显示方式,又可支持彩色字符显示方式,当选择单色(黑白)显示方式时,显示字符具有闪烁、加亮、反视频的属性,而选择彩色显示方式时,每个字符可以选用 16 种背景(底色)和前景(显示色)色彩,而且仍可以使用闪烁属性。字符方式下的工作过程:CPU 在显示器水平和垂直回扫期间,把欲显示的字符码及其属性码分别送入显示缓存的偶地址单元和奇地址单元;MC 6845 CRTC 片在字符字节时钟同步下,一面产生 15.75kHz 行同步和 50Hz 的帧同步等信号送到彩色显示器,一面产生刷新地址码去读显示缓存中的字符和属性;以显示缓存中读出的字符代码和 MC 6845 提供的扫描线序号为地址,读出字符发生器中的字符点阵信息,字符属性则送颜色编码器,字符点阵信息读出后送入字符像素并/串转换器,然后逐位移出到颜色编码器中,与有关的属性、光标等配合,形成显示器所需要的视频三元色输出

信号。

另一种是图形显示模式,也称 APA 模式。在这种工作模式下,显示器屏幕上的每个象点均可由程序控制其亮度或颜色,因而能显示出质量较好的图形,APA 模式又可分为三种不同的分辨率:高分辨率模式:每帧 200 线,每线 640 点,每点只能取黑白两种颜色;中分辨率模式:每帧 200 线,每线 320 点,每点可以有 4 种不同的颜色;低分辨率模式:每帧 100 线,每线 160 点,而每点则有 16 种不同的颜色。目前在 BIOS 基本输入/输出系统中的驱动程序不能支持低分辨率显示模式,需要时应由应用软件人员自行开发。

彩色图形显示控制器逻辑结构与单色显示控制器类似,它也采用 6845 CRTC 芯片作为控制器的核心,一方面产生显示缓冲器的地址码以读出其中的信息,另一方面同步地给出水平和垂直扫描信号去控制 CRT 显示器。图形方式下的工作过程:CPU 在显示器水平和垂直回扫期间,把欲显示的图形象素送入显示缓存,其中偶数扫描线像素放在显示缓存的前 8KB,奇数扫描线像素放在后 8KB;MC 6845 CRTC 芯片在像素字节时钟 CCLK 同步下,一面产生 15.75kHz 的行同步,50Hz 的帧同步送彩色显示器。另一方面产生刷新地址码,每次先后读出显示缓存中偶数和奇数地址单元的两个像素字节,送入图形锁存器;这两个像素字节同时送入并/串移位器,然后每次移出二位,送到颜色编码器中,若是 320×200 彩色图形方式,移出的二位与颜色选择寄存器的第 4、5 位一起经颜色编码器产生视频彩色信号,若是 640×200 图形方式,颜色编码器输出的视频三元色信号仅是黑色或白色。

(上接 46 页)

换针时,首先要准备好工具,主要是十字槽螺丝刀、镊子及小锉刀,换针开始,先拆下后盖,把打印头朝下,尾部朝上,用镊子取出所换的针,然后把要换的新针在对应位插下,打印针自尾部到探出头,共要经过四道孔槽,上面的两个孔,起着打印针的定位作用,接着,针顺势而下,穿过导向块,导向块是一个尼龙片,在它上面有两个弧形孔、奇数针(12)个和偶数针(12)个分别穿过其中的一个孔,针在穿过这个孔时,由于孔不象前两个孔是一一对应的,故可用镊子先把针捋几下,使该针基本上处于该处的位置,针进入最后一个孔,既从打印头冒出来,是最难的一步,可用镊子在第三、第四孔之间再捋几下,或者是边捋边插,一直到针穿出。

当针换过后,新针总是要比其它针长一点,可用锉刀把长出的部分锉掉,使该针和其它针平齐,然后再装上后盖,这样,打印针换针的工作就算是完成了。

在换针时,还有几点要注意的,一是切忌把拆盖后

的打印头朝上放置,以免其它针掉下来,若要头朝上时,一定用拇指把所有衔接铁按住。再就是若在换针时,其它的针已经掉下来,则可先将掉下来的针恢复上去,因为这样的针,再插上去比取下来的针撤上去要容易些,另外,如果需要把所有的针都拿下来,一定不要搞乱了它们的顺序。因为打印头上靠近上下两端的针比位置在中间的针要稍长些。把针的位置搞乱,这个打印头实际上也就没有什么用了。

打印针更换完毕再装上打印机时,也要注意一点,打印头和打印辊之间的距离是有规定的,在无屏蔽罩(MASK)的情况下是 0.35mm,在有屏蔽罩的情况下,距离是 0.25mm。在有屏蔽罩的情况下上头时,必须先要松开屏蔽罩的螺丝,使其紧贴在打印辊上,然后装头上头,用 0.5mm 的塞尺(厚薄规)塞在头与屏蔽罩之间,过紧或过松都不行,手觉得适度后,把打印头上紧,再把屏蔽罩上好。头与辊之间的距离是否准确,关系着打印头的寿命。