



● 一九九一年 ● 总期第79期

10

電子

ISSN 1000-1047

144
744

與 電腦



• ELECTRONICS AND COMPUTERS •

● 教师的好助手

● 学生的老师

● 游戏的好伙伴



■ 电子工业出版社向广大软件用户致敬!

- 本出版社软件部出版、发行的软件种类有：教学软件、系统工具软件、应用软件、幼儿智力开发软件及游戏软件等。
- 本部备有软件及《中华学习机、苹果机软件大全》一书。
- 教学软件由国家教委、机械电子部、中国科协的有关部门联合组成“教育软件评委会”评审。
- 电子工业出版社软件部
- 地址：北京万寿路173信箱
- 邮政编码：100036；电话815342
- 联系人：王爱英



■ The PHEI pays tribute to wide circle of software users!

- The software products published and distributed by Software Department of PHEI are classified to: educational software, system software tool, application software, intelligence exploit software for children, and game software, etc.
- We have in store the software catalogue and the book: "Software Encyclopedia of Zhonghua Learning Computer and Apple Computer."
- Educational software are evaluated by the "Educational Software Evaluation Committee", composed of such units, as National Educational Commission, Ministry of Machinery and Electronics Industry and China Association for Science and Technology.
- Software Department, PHEI
- address: P.O. Box 173, Wan Shou Lu, Beijing
- Post code: 100036
- Phone: 815342
- Person to Contact: Ms. Wang Aiying



耐心 ● 亲切 ● 生动 ● 直观

自学 ● 考试



欢迎
参加

为提高我国软件人员素质,鼓励广大软件人员为振兴中华而勤奋学习,并利于发现人才、培养人才,由中国软件行业协会考试指导中心、中国软件行业协会高档微机协会、北京市科学技术协会、电子工业出版社《电子与电脑》编辑部、AST 虹志(电脑)有限公司等单位联合举办:

一九九一年度“AST 杯” 第二届全国计算机软件人员竞赛

(初级程序员级和程序员级)

热心于社会教育事业的 AST 虹志(电脑)有限公司等对本次有奖竞赛积极赞助,在此我们谨向热情支持本竞赛的社会各界人士表示衷心感谢。

竞赛的第一轮比赛正在进行,复赛定于 11 月 19-20 日在北京清华大学举行。两个级别各设一等奖 1 名,二等奖 2 名,三等奖 5 名,四等奖若干。分别奖给获奖证书、AST 奖杯、奖牌及其它纪念品。

参赛对象:

1, 初级程序员级: 按规定时间完成刊登于《电子与电脑》杂志今年第八期上刊出的有奖竞赛全部试题, 成绩优秀者。

程序员级: 参加第三期全国计算机软件人员水平考试函授辅导班结业考试, 成绩优秀者。

2, 各省市软件水平考试实施办推荐参加今年 10 月 13 日全国计算机软件人员资格和水平考试的优胜者。

参赛条件:

(1) 初级程序员级, 未获得计算机专业本科学历或助理工程师职称者。

(2) 程序员级, 未获得计算机研究生学历或工程师职称者。

(3) 年龄在 35 周岁以下。

联系地址: 北京学院路 31 号中国软件行业协会考试指导中心

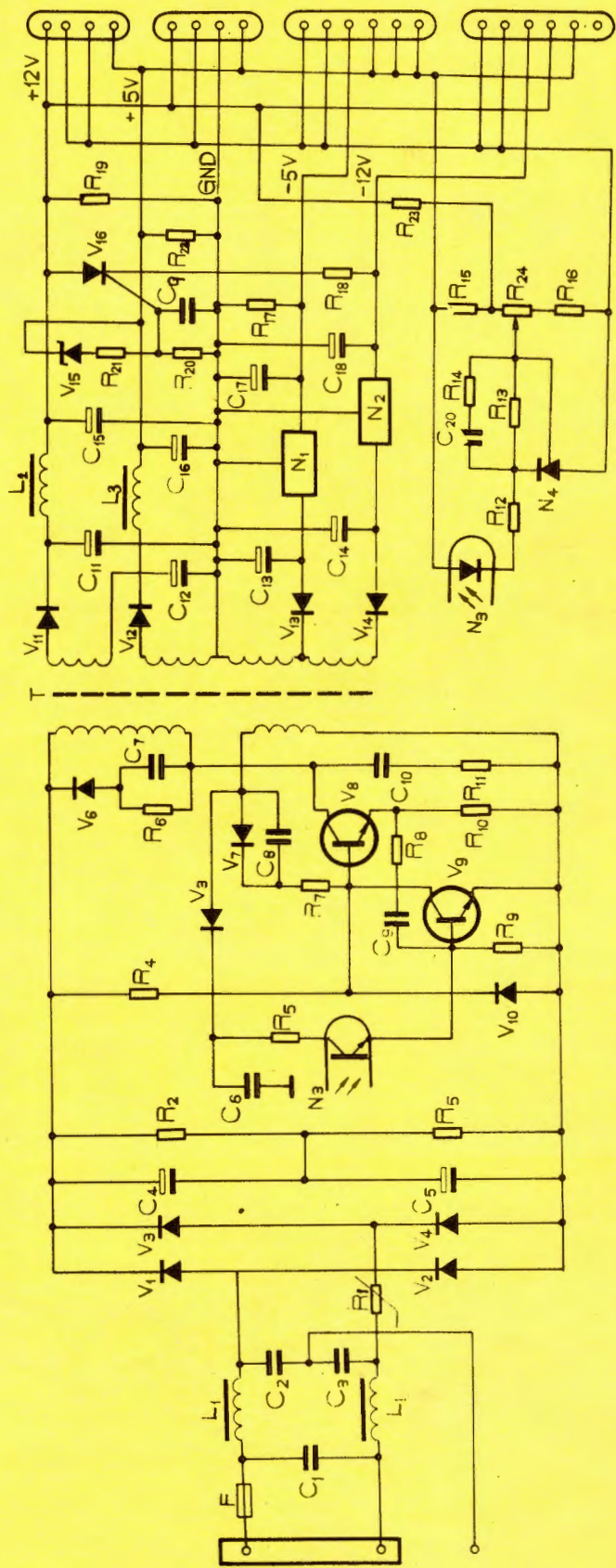
邮政编码: 100083

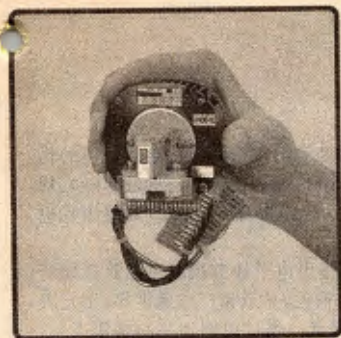
电 话: 2012233-322

传 真: 2024674



PS60-2 (60 W) 电源





• ELECTRONICS AND COMPUTERS •

一九九一年

总期第79期

電子與電腦

目 录

• 综述 •

论电脑写作 袁明光(2)

• PC 用户 •

BROWSE 命令中 FIELDS 参数的菜单选择 张述健(4)

给 dBASE 程序加入活用功能 贺映文(5)

在 dBASE III 中实现多维数组功能的另一方法 程林(6)

集成字处理软件 CCED 的使用技巧 许鹰(6)

如何自编 Park? 葛芝宾(7)

需 ROM BASIC 支持的软盘改造 曾礼伟(8)

最新病毒检测及清除软件 VIRUSCAN Ver7.2v77 梁伟杰(9)

Goldbach 奇数猜想的 BASIC 程序 陈君佐(10)

快速排序“快”在何处? 应宏(11)

两个高效率的排序程序 李军杰(11)

• 学习机之友 •

CEC-I 高级动画制作技术 黄一禾(13)

警察抓小偷 李明(15)

ProDOS 磁盘操作系统入门(续) 廖凯(16)

高性能的 CEC-I 自定义组合键 杨勇佳(18)

活泼的光标 殷瑞冬(19)

CEC-I 机汉字使用第十一行 任义(19)

三页图形竖向并打 陈钢(20)

电子打字机 王蔚(20)

• C 语言初阶讲座 •

第八讲 指针与数组 李文兵(21)

• 学用单片机 •

多功能电子钟 张培仁 刘振安(27)

• 电脑巧开发 •

简易精确的 APPLE II 时钟卡 姜仕仁 施青松(30)

用 EPPOM 制作多路报警器 王洪伟(32)

• 游戏机电脑 •

键盘电视游戏机的 FBASIC 语言及程序设计(上) 于春(33)

• 维修经验谈 •

AT(286)微机硬盘处理一例 刘文峰(35)

从屏幕显示异常判断苹果主机板故障 张艾美(36)

M2024 型打印机故障检修一例 王峰(37)

计算机设备故障检修表 管金省 宋靖涛(38)

IBM PC/XT 微型计算机检修一例 平凡(39)

MP-I 微电脑快速检修法 李红(40)

如何使九针打印机字迹更清晰 陆秀忠(41)

YOKO747 型教学语言学习机的修理 郭长福(41)

微机软盘驱动器“卡盘”的简易修复 乐冬青(42)

CEC-I 软驱故障维修一例 潘冬(42)

“坏”软盘的再生 张智(43)

GW286BH 系统板维修一例 董纯坚(43)

怎样使用清洗盘 瞿新国(43)

LASER 机 GA004 电路的代换和修复 张保田(44)

浪潮 0530-AT 死机故障恢复一例 董国钦(45)

• 专题 •

一九九一年计算机初级程序员有奖竞赛试题参考答案

..... (46)

• 读者联谊 •

关于召开北京地区“PC”机个人用户软件交流联谊会的

通知 (47)

• 信息与服务 •

天坛学习机教学网系统 SEN-I 型简易教学网络系统简

介 (48)

机械电子工业部电子工业出版社主办

编辑、出版:《电子与电脑》编辑部

(北京 173 信箱 邮政编码:100036)

印刷:北京三二〇九厂

国内总发行:北京报刊发行局

国内统一刊号:CN11-2199

邮发代号:2-888

国外代号:M924

出版日期:每月 23 日

主编:王惠民 副主编:王昌铭

责任编辑:施玉新

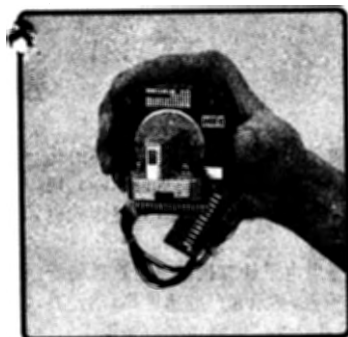
订购处:全国各地邮电局

国外总发行:中国国际图书贸易总公司

(北京 399 信箱 邮政编码 100044)

广告经营许可证:京海工商广字 147 号

定价:0.95 元



* ELECTRONICS AND COMPUTERS *

一九九一年

总期第79期

電子與電腦

目 录

• 综述 •

论电脑写作 袁明光(2)

• PC 用户 •

BROWSE 命令中 FIELDS 参数的菜单选择 张述健(4)

给 dBASE 程序加入活用功能 贺映文(5)

在 dBASE Ⅲ 中实现多维数组功能的另一方法 程林(6)

集成字处理软件 CCED 的使用技巧 许鹰(6)

如何自编 Park? 葛芝宾(7)

需 ROM BASIC 支持的软盘改造 曾礼伟(8)

最新病毒检测及清除软件 VIRUSCAN Ver7. 2v77 梁伟杰(9)

Goldbach 奇数猜想的 BASIC 程序 陈君佐(10)

快速排序“快”在何处? 应宏(11)

两个高效率的排序程序 李军杰(11)

• 学习机之友 •

CEC-I 高级动画制作技术 黄一禾(13)

警察抓小偷 李明(15)

ProDOS 磁盘操作系统入门(续) 廖凯(16)

高性能的 CEC-I 自定义组合键 杨勇佳(18)

活泼的光标 殷瑞冬(19)

CEC-I 机汉字使用第十一行 任义(19)

三页图形竖向并打 陈钢(20)

电子打字机 王蔚(20)

• C 语言初阶讲座 •

第八讲 指针与数组 李文兵(21)

• 学用单片机 •

多功能电子钟 张培仁 刘振安(27)

• 电脑巧开发 •

简易精确的 APPLE II 时钟卡 姜仕仁 施青松(30)

用 EPPOM 制作多路报警器 王洪伟(32)

• 游戏机电脑 •

健盘电视游戏机的 FBASIC 语言及程序设计(上) 于春(33)

• 维修经验谈 •

AT(286)微机硬盘处理一例 刘文峰(35)

从屏幕显示异常判断苹果主机板故障 张艾美(36)

M2024 型打印机故障检修一例 王峰(37)

计算机设备故障检修表 咎金省 宋靖涛(38)

IBM PC/XT 微型计算机检修一例 平凡(39)

MP-I 微电脑快速检修法 李红(40)

如何使九针打印机字迹更清晰 陆秀忠(41)

YOKO747 型教学语言学习机的修理 郭长福(41)

微机软盘驱动器“卡盘”的简易修复 乐冬青(42)

CEC-I 软驱故障维修一例 潘冬(42)

“坏”软盘的再生 张智(43)

GW286BH 系统板维修一例 董纯坚(43)

怎样使用清洗盘 瞿新国(43)

LASER 机 GA004 电路的代换和修复 张保田(44)

浪潮 0530-AT 死机故障恢复一例 董国钦(45)

• 专题 •

一九九一年计算机初级程序员有奖竞赛试题参考答案

(46)

• 读者联谊 •

关于召开北京地区“PC”机个人用户软件交流联谊会的

通知 (47)

• 信息与服务 •

天坛学习机教学网系统 SEN-I 型简易教学网络系统简

介 (48)

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

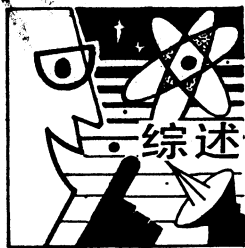
订购处:全国各地邮电局

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



论 电 脑 写 作

袁 明 光

目前,对电脑写作有各种不同的理解,有人以为,电脑写作就是电脑打字;也有人理解为,电脑写作是用电脑代替人脑进行写作构思,代替作者写成文章。这都是对电脑写作的性质不够明确。

大家知道,任何机器都是工具,不过,电子计算机是具有智能的工具,不仅可以进行数值计算,还可以进行自动控制,模拟人的感觉和思维过程等部分智能。正因如此,人们叫它做电脑是十分贴切的。虽然电脑已广泛应用于科学预测、事务管理、工程设计、医疗诊断、资料检索、文献翻译以及教学科研等各个领域。但是,这些功能都是辅助性的,其中有的被作为计算机技术的分支学科发展着,如计算机辅助测试(CAT)、计算机辅助设计(CAD)、计算机辅助教学(CAI)等。

电脑写作,不等于电脑打字,也不是用电脑代替人脑制作文章。电脑打字与电脑写作的关系,正如写字与写文章的关系一样,既有联系,又不能等而视之。电脑打字有“看打”、“听打”、“想打”之分别,着眼于单字;而电脑写作则重于“想打”,着眼于词语和语群输入。

用电脑代替人脑制作文章,这已大大超出了人工智能的范畴,也不是我们所探讨的电脑写作。因此,我认为,电脑写作是一种借助于电脑的人工智能,在电脑写作系统的支持下,运用写作软件进行的写作,简称为CAW。

二

用电脑进行写作的可行性如何?实现电脑写作需要具备怎样的条件呢?

电脑写作的可行性,是以其电脑技术为基础的。电脑写作的实现,必须具有电脑设备(硬件)和写作软件以及会操作使用电脑的作者。这是三个必备条件,缺一不可。

电脑写作现已不是停留在理论上的探讨,而是被写作实践所证实了。国外用电脑写诗歌、小说、论文已屡见不鲜。七十年代末,美国一家刊物Byte就出版过电脑写作专辑,用电脑创作的小说及其插图早已见于Omni杂志。1982年,加拿大作家伯克·坎贝尔用电脑花了61.5小时创作了一部共分19章的长篇小说《眼睛的法老》,经另一电脑校对后三个小时就从加拿大传输到美国的STC编辑部,并存入信息库中,供电脑读者索取。中文电脑写作起步稍晚,但海外华人作家陈若曦、聂华苓,台湾作家高阳、雷震谦等,都已开始用中文电脑写作。北京作家邵华,1988年底开始用电脑写作,深得效益,至1989年底陆续在电脑上写了30多万字的作品,1990年4—7月,用“纯”工作时间100多小时写完20多万字的长篇小说《三角——红黄白》。这些事实,难道还不足以证明电脑写作的可行性吗?当然,他们所进行的电脑写作还处于初始阶段,属于低层次的,有待完善与提高,使之向较高层次发展,但这毕竟是开了个头。

三

为什么要采用电脑写作?因为电脑写作比传统写作有许

多的优越性,能使写作实现自动化和现代化。电脑写作的优越性主要表现为快速性(Quickness)、灵活性(Adaptability)、规范性(Normality)等三大特性,合称为QAN特性。这是现代电子技术与写作结合的产物。

电脑写作的快速性(Q)是由电子计算机超运算性能所提供的技术基础。而中文的输入,显示和打印速度等,则是决定电脑写作快速性的重要因素。经过短期训练的操作人员,采用形码输入,每分钟可输入120多个汉字;采用“自然码”输入,每分钟可输入200个汉字;要是采用语群化输入或智能化输入,其速度还会更快。CCDOS的显示速度每秒钟9600位,一般的针式打印机中文打印速度每秒达100多字符。这些都是电脑写作快速性的技术保障。当然,快速性的实现,还要靠有好的写作软件和电脑写作者熟练地运用软件。

电脑写作的灵活性(A)表现在写作手段特别灵活。将文字信息输入电脑,这是电脑写作关键的一环。目前的输入手段有键盘编码输入、手稿扫描输入和语音合成输入等,而以键盘编码输入为最普遍。键盘编码输入方式种类繁多,已达500多种,但概括起来不外乎形码、音码、音形码、音义码和速写码等几类。这几类编码各有所长,具有互补性,电脑写作者可根据自己的喜好,自由选用一两码法。将文字信息输入电脑后,则须进行编排,使之形成文章。诸如标题居中、缩格、对齐、断行、接行、调行,设定字距、行距、行宽,选用字号等,都十分方便,一按某键便成。这在传统写作中是不可思议的。

传统写作的修改相当麻烦,常常将稿纸画得琳琅满目,过后甚至作者自己都看不清。而在电脑上对文章进行增补、删除、插入、更改,无论是词语、行段还是章节,甚至自动查找全文中的某个词语并加以更改等都十分方便。正是因为有如此丰富和特别灵活的写作手段,才使得电脑写作奇妙无比。

电脑写作的规范性(N)是指用电脑写成的文章,在款式、文字、标点和文面等各方面都达到标准化、规范化。在电脑上写文章,最易做到文体格式的统一,写错字的情况不会出现(别字就难说),繁简字相混杂的现象可消除,使字体标准,字迹清楚。而文面的整洁、清晰、美观,具有视觉美,这是传统写作无法相比的。

上述快速性(Q)、灵活性(A)和规范性(N)是电脑写作纵向方面的优越性,而横向方面的优越性则是写作编辑一体化(WEI)、写作阅读一体化(WRI)、写作存储一体化(WMI)、写作打印一体化(WPI)。总之,电脑写作的优越性是全方位的。电脑与写作结合,就给写作插上了现代化的翅膀,使写作理论与写作技能发生一次新的飞跃。

四

电脑写作的层次结构可分为:1. 字词处理层次(WHT); 2. 文体模式层次(LMT); 3. 智能辅助层次(IAT)。这三个层次是与电脑技术的发展进程相关联,与电脑技术的应用程度相适应的。字词处理层次(WHT),是电脑写作的初级层次。它最显著的特点是输入信息以字词或词组为单位。现在的四通、科理、文豪、光明、翰林等各种型号的文字处理机,汉化Wordstar和XE之类的软件,集中地体现了第一层次的优越性。将

文字信息输入电脑,在屏幕上进行编辑,把文稿存储软盘,又将文稿打印成文,这确实比传统的执笔写作来得方便而快速。但从实现写作现代化的角度去要求,这种进步还是低层次的。

文体模式层次(LMT),是电脑写作的中级层次。它虽然仍须文字信息输入,但它已与文体模式结合起来了,比字词处理更进了一步。写作各种文体都有一定的模式,诗与散文的模式不同,小说、散文与各类事务公文模式也不同。这种不同,是由其构成要素和构成方式相异所致。文体模式层次的特点,是将某种文体构成要素和构成方式输入电脑后,使其自动生成某种文体。国内这方面已有人作了有益的探索,上海学生梁建章,设计了一个诗歌写作程序,用来进行古典诗词和现代新诗创作。武汉的电脑写作研究专家研制了用于公文事务写作的ZL写作软件和新闻写作软件,使电脑写作在文体模式层次上迈出了可喜的一步。但总的说来,这第二层次的研究还相当薄弱,成果甚少,还不能象第一层次那样为写作者所广泛应用。之所以如此,除文体写作软件研制难度大、对此课题感兴趣的研制者少等原因外,而写作学界对各种文体模式的研究还很不够,未能达到科学化的程度也是个重要原因。

智能辅助层次(IAT),是电脑写作的高级层次。它借助于人工智能,即电脑模拟人类的感觉、思维等能力,以辅助写作构思。在写作过程中,除材料的搜集准备外,其成败的关键在于构思。写作构思包括确立主题、优选材料、设定框架以及调动各种语言手段等各个环节。这些环节都必须运用形象思维、逻辑思维和灵感思维。除灵感思维难以得到电脑的辅助之外,我认为形象思维和逻辑思维是可以借电脑的辅助而展开联想、进行判断和推理的。比如,在众多的材料中筛选与主题相关的材料,将优选出的材料按时间或空间进行序列,循某一线索展开各式联想等等,这些都是电脑写作智能辅助层次所要解决的问题。

我国已经研制了文字识别系统和语音识别系统,在人工智能的征途上,迈出了可喜的一步。这些研究成果表明,电脑写作智能化是能够实现的。

五

最重要的是开发电脑写作系统(CWS),电脑写作系统开发成功,乃是电脑写作进入实用阶段并走向成熟的标志。

电脑写作系统(CWS)是一个庞大的专家系统,它应由通用写作系统(GWS)和专业写作系统(SWS)构成。通用写作系统包括通用词语库、通用句型生成模块、通用词语联想模块、通用表达选择模块等。通用词语库是通用写作系统的基本构件,句型的生成、词语的联想、表达的选择,都须依靠它提供丰富的词语资料。词语库的建立,除了注意其丰富性和覆盖面之外,其排列及调用方式多样化也是不可忽视的。我主张,既要有音序、词形的排列,也要有词义的排列,既采用词首调用,也采用语尾调用,(如写诗要押韵,就需采用语尾调用)使其具有广泛的适应性。

通用句型生成模块,具有自动生成各种类型的句子功能,作者只要输入其句子的要素,便能自动生成所指定的那种句型。通用词语联想模块,具有通过词语帮助作者进行联想的功能,作者一旦输入想要表现的某一词语(事物),便能展现与此词语(事物)有关的各种词语(事物),以扩展作者的思路与视野。通用表达选择模块,具有选择叙述、描写、说明、议论等各种表达方式的功能,作者只需输入其表达要素,便可形成各种表达方式供作者选择。句型生成、词语联想和

表达选择等模块功能的实现,将使电脑写作在输入语群化、智能化的道路上向前迈进。

专业写作系统(SWS)是一个以适应各种专业写作需要而建立的大规模系统,它包含事务写作系统(RWS)、科技写作系统(S-TWS)、新闻写作系统(NWS)、理论写作系统(TWS)、文学写作系统(LWS)等众多的子系统。每个子系统都建立有各自的专业词语库、素材储备库和文体生成模块。

专业词语库中一部分是固定词语,一部分是自定词语,以适应各种不同语言风格的作者的需要。有人担心,用电脑写文章会千篇一律,失去个人风格。其实,这种担心是多余的。个人风格是由多种因素决定的,除了内容和思维方式的不同外,最主要的是语言风格的不同,自定词语,就是根据各自的独特的习惯用语和词语组合方式而建立的,它从词语方面保证了个人风格的实现。专业词语库是通用词语库的扩充。在进行专业写作时,可同时调用两个词语库的词语,也可调用通用写作系统中的句型生成、词语联想、表达选择模块。

素材储备库是一个内容独特的知识库,一般是由作者自己建立,或者是由写作系统研制者建立基本框架,然后再由作者自己扩充。它的特点是:既要容量大,又要便于存取。因此需要科学地划分好素材分类仓,给各种素材编上代号,随时将观察或阅读所获得的素材输入分类仓内。积累素材是写作的重要一环,无论传统写作还是电脑写作都是如此。传统写作将所获得的素材记入人脑或笔记本、卡片,而电脑写作是将素材输入电脑储存起来,以供调用。在写作实践中,许多作者都体验过,一旦写作灵感袭来,常常是来不及写下,所想好的东西就跑掉了,甚至事后很难追忆,感到十分遗憾。用电脑进行写作,在电脑中建立素材储备库,则可弥补这种遗憾。当灵感来临时,作者便可以迅速从素材储备库中找到与其相似的各种资料,很快形成文字,显示于屏幕上,然后再加以修改,使它更符合作者的构思。

专业写作系统中,有一些文体的模式已经定型,如事务写作系统中的通知、书信、总结、合同、起诉书、判决书,科技写作系统中的实验报告,新闻写作系统中的消息,理论写作系统中的短论,文学写作系统中的诗歌等,相对地说,实现文体自动生成容易一些,则可在系统中先实现。将某种文体的构成要素和构成方式输入电脑,使其按规定的模式自动生成某种文体,这是实现写作自动化、现代化关键的一环。

开发电脑写作系统有一个过程,在这个过程中,我认为进行双向交流与双向适应是十分重要的。双向交流,是指写作系统研制者与写作者之间所进行的正向和反向信息交流,一方面,写作系统研制者从写作者那儿取得写作动态信息,以促进系统研制;另一方面,写作者从系统研制者那儿取得开发进展信息,以利于将系统用于写作。双向适应,是指写作系统研制者与写作者之间的相互适应,一方面,系统研制者需要适应写作者的各种不同构思方式、词语组合方式、表达方式,以满足写作的不同要求;另一方面,写作者需适应系统研制者的新的构想与独特的技术处理,以调整自己的写作认识结构与技能结构。写作者的个人习性与风格并非一成不变的,当他采用电脑写作系统进行写作后,自己的认识结构和技能结构必然会起变化,最后将导致作品风格的改变。这种改变使得作品的风格变得多样化,并非坏事,写作者要有充分的思想准备,努力去适应它,从而充分发挥电脑写作系统的最大效益。

已经崛起的电脑写作将会按照它自己的规律向前发展,走向成熟。让我们迎接这一天的到来!



BROWSE 命令中 FIELDS 参数的菜单选择

陕西扶风胜利机械厂子弟学校 张述健

在 dBASE Ⅲ 中, BROWSE 命令以窗口显示方式给用户的数据修改、浏览以及追加提供了很大的方便。但由于屏幕所能显示的字段有限, 在数据库文件的字段比较多的情况下, 人们一般要对字段进行选择, 将需要的字段固定在屏幕上, 使用户界面清晰、直观。然而, BROWSE 命令格式中, FIELDS 可选参数项的选择不仅输入量大(敲键次数多, 尤其是字段使用汉字时), 而且要求操作人员对具体的库文件结构也得非常熟悉(起码需要记住各个字段的名称), 给操作人员造成很大的不便。笔者受“菜单”程序设计思想的启发, 用 dBASE Ⅲ 语言实现了 BROWSE 命令中 FIELDS 可选参数项选择的“菜单”化, 减轻了操作人员的负担, 提高了数据修改的速度和准确性。

一、首先利用 dBASE Ⅲ 中的 COPY TO (文件名) STRU EXTE 命令将源库文件的结构拷贝到一个结构描述文件“XU-1. DBF”中, 将库文件中的每一个字段转化为“XU-1. DBF”中的所有记录, 然后程序访问结构描述文件“XU-1. DBF”, 巧妙地应用记录号做为各字段选择的序号, 将 FIELD-NAME 的内容(即源库文件中的各个字段名称)以“菜单”的形式直观地显示在屏幕上(可分屏显示), 供操作人员选择。选择时, 只需简单地敲入字段的序号即可, 非常方便、灵活。若多项选择, 序号之间用逗号“,”隔开; 若直接回车则保留全部的字段, 接收的序号序列送入变量 JLH 中。

二、根据变量 JLH 中所接收到的序号序列, 程序分三种情况进行测试: ①利用 LEN(字符串)函数测试变量 JLH, 结果送 L 中, 若 L=0, 则表明用户将保留全部字段; ②若 L≠0, 再利用 AT(子串, 主串)函数测试变量 JLH 中是否有子串“,”, 若没有, 则表明用户仅选择了一个字段; ③若有子串“,”, 程序则进入一个双重循环结构, 对整个序号序列(JLH 变量中的值)进行测试分析, 做分离工作。内循环体将逐个确定每个序号在整个序号序列中的起始位置和字符数, 分别送入变量 I、J 中; 然后, 外循环体根据 I、J 的值, 使用 SUBS() 函数将它们逐个分离出来, 还原成所需要的记录号, 用 GO 命令在结构描述文件 XU-1. DBF 中进行定位, 将所选字段名称(即 FIELD-NAME 的内容)送入变量 ZDM 中, 利用变量 YY 排列组合所选择的所有字段名, 形成用户所期望的可选参数序列。最后用 dBASE Ⅲ 中的宏替换命令 & 完成 BROWSE 中 FIELDS 可选参数的替换赋值, 用户即可按照自己的意愿对所造各字段进行各种操作。

主要程序段如下:

“菜单”显示程序段

.....

```
ACCE “库文件名?” TO KNAME
USE &KNAME
```

```
COPY TO XU-1 STRU EXTE
```

```
X=1
```

```
Y=6
```

```
DY=17
```

```
YY=INT(80/DY)*DY
```

```
USE XU.1
```

```
CLEAR
```

```
@0,30 SAY “需修改、追加及浏览的字段”
```

```
DO WHILE .NOT. EOF()
```

```
IF X>8
```

```
WAIT “请记住各字段名序号! 按任一健继续……”
```

```
@1,0 CLEAR
```

```
X=1
```

```
Y=6
```

```
DY=17
```

```
YY=INT(80/DY)*DY
```

```
ENDIF
```

```
I=STR(RECN(),3)
```

```
@X,Y SAY I+“---”+FIELD-NAME
```

```
Y=Y+DY
```

```
IF Y>YY
```

```
X=X+1
```

```
Y=6
```

```
ENDIF
```

```
SKIP
```

```
ENDDO
```

```
@ROW()+1,5 SAY “多项选择时, 序号间用逗号隔开,  
直接回车则保留全部字段。”
```

```
ACCE “请输入所选字段序号:” TO JLH
```

.....

“测试、分离”程序段

.....

```
L=LEN(JLH)
```

```
YY=SPAC(50)
```

```
I=1
```

```
J=0
```

```
IF L=0
```

```
USE &KNAME
```

```
BROW
```

```
ELSE
```

```
IF AT(“,”,&JLH)=0
```

```
X0=SUBS(JLH,1,L)
```

```
GO VAL(&X0)
```

```
ZDM=TRIM(FIELD-NAME)
```

```
YY=YY+ZDM
```

```
ELSE
```

```
DO WHILE .T.
```

```
DO WHILE I<=L
```

```
XX=SUBS(JLH,I,1)
```

```

IF XX<>"",
I=I+1
J=J+1
ELSE
EXIT
ENDIF
ENDDO
X0=SUBS(JLH,I-J,J)
GO VAL("&X0")
ZDM=TRIM(FIELD-NAME)
IF I<L
YY=YY+ZDM+"",
FF="LOOP"
ELSE
YY=YY+ZDM
FF="EXIT"
ENDIF
J=0
I=I+1
&FF
ENDDO
ENDIF
YYY=TRIM(YY)
USE &KNAME
BROW FIELD &YYY
ENDIF
.....

```

给 dBASE 程序加入 活用功能

桂林市人民银行科技科 贺映文

我们在调试程序、运行管理系统时,经常会碰到一些特殊情况,不得不中止程序的运行,浪费了很多不必要的时间。例如,我们在编制会计电脑记帐程序时,稍不小心就碰到:

该文件不存在!

```

?
USE lszsj INDEX ndlszsj
调用处为-C:pxls. prg
.....

```

是否中止执行命令文件? (Y/N)

出现这种情况的原因有两个:一个是 C 盘上没有 lszsj. dbf 数据库文件;另一种可能是没有产生 ndlszsj. ndx 索引文件。通常的解决办法是按 Y 键,中止执行命令文件,返回圆点提示符或返回 DOS 系统,转去执行特殊处理,将 lszsj. dbf 拷贝进去,并对其进行索引。但处理完毕后还要重新启动,就象上面的例子,还要先进主控菜单,再进子菜单,子菜单再进子菜单,才能执行打印排序流水帐的程序,做起来很麻烦。

于是,我就想:如果在碰到上面的情况时,能够不

按 Y 键退回,而按 N 键在程序运行状态下进行相应处理后,再继续执行原程序就好了。经过一番努力,笔者利用 dBASE III 设计了两个活用于程序(见 hy1. prg 及 hy2. prg),加到程序适当处,巧妙地解决了上述问题。在上述例子中,加入两个活用程序后,不必按 Y 键退回,而是按 N 键转去进行活用处理(即拷贝 lszsj. dbf 及产生索引文件 ndlszsj. ndx 等),处理完毕后,自动转到原来执行状态,继续执行原程序。

加入活用功能后,系统不仅能处理类似上述的问题,而且还可以在程序运行过程中执行各种 DOS 内部命令和外部命令,同时也可执行 dBASE 命令及 Word-Star 等,活用处理后,继续执行原程序,大大地提高了程序的灵活性,优化了程序设计。

* * * 活用于程序 1 hy1. prg

```

* 在调用某文件的语句后插入本程序
if .not. file('文件名')
do hy2
* 在此重写调用某文件的语句
endif

```

* 活用于程序 2 hy2. prg

```

set talk on
set console on
do while .t.
clear
acce '请进行活处理,处理好后,按回车继续执行原程
序):' to ml
if ml>' '
&ml
? ' '
wait ' 按任意键继续!'
else
exit
endif
enddo
set console off
set talk off
use
return

```

(上接第 8 页)

即告成功。现在不需 ROM BASIC 支持即可运行。有些盘带有自启动,如不需要,可将 IBMBIO. COM、IBMDOS. COM、COMMAND. COM 三个文件删掉(在用步骤(4)时这三个文件已用 FILEnnnn. REC 的形式列出,步骤(6)更名),找出盘中的批处理文件名(扩展名为 BAT),以后在机器启动后,打入批处理文件名即可运行。

此法在 PC/XT 机上获得成功。

在 dBASE Ⅲ 中实现多维数组功能的另一方法

程 林

dBASE Ⅲ 没有数组语句。《电子与电脑》91 年第四期上介绍了一种方法,即巧妙应用 & 函数来实现多维数组功能。本人感到这种实现方法有其局限性,一是该方法所述的数组元素其实是内存变量,由于 dBASE Ⅲ 要求当前工作的内存变量总数小于 256 个,因此不能定义太大;二是其数组下标的实现要将循环变量由“N”型转换成“C”型,并且要删去前面的空格符再用 & 函数实现,故在调用数组元素时比较复杂。

这里介绍另一种方法——“取子串”方法。虽说实质还是采用内存变量代表其数组元素,故数组大小仍受限制,但实现方法比用 & 函数简便、直观,调用数组元素也较简单。对于各维元素小于 26 的数组,其实现程序如下(以实现一个 11×6 数组运算为例):

```
SET TALK OFF
CLEAR
```

A0101= 1	A0102= 2	A0103= 3	A0104= 4	A0105= 5	A0106= 6
A0201= 2	A0202= 4	A0203= 6	A0204= 8	A0205= 10	A0206= 12
A0301= 3	A0302= 6	A0303= 9	A0304= 12	A0305= 15	A0306= 18
A0401= 4	A0402= 8	A0403= 12	A0404= 16	A0405= 20	A0406= 24
A0501= 5	A0502= 10	A0503= 15	A0504= 20	A0505= 25	A0506= 30
A0601= 6	A0602= 12	A0603= 18	A0604= 24	A0605= 30	A0606= 36
A0701= 7	A0702= 14	A0703= 21	A0704= 28	A0705= 35	A0706= 42
A0801= 8	A0802= 16	A0803= 24	A0804= 32	A0805= 40	A0806= 48
A0901= 9	A0902= 18	A0903= 27	A0904= 36	A0905= 45	A0906= 54
A1001= 10	A1002= 20	A1003= 30	A1004= 40	A1005= 50	A1006= 60
A1101= 11	A1102= 22	A1103= 33	A1104= 44	A1105= 55	A1106= 66

```
N="01020304050607080910111213141516171819202122
232425"
```

```
I=1
DO WHILE I<=11
J=1
DO WHILE J<=6
V="A"+SUBSTR(N,I*2-1,2),+SUBSTR(N,J*2-1,2)
&V=I*J
?? "&V="+STR(&V,5),+SPACE(6)
J=J+1
ENDDO
I=I+1
ENDDO
?
```

运行结果:

集成字处理软件 CCED 的使用技巧

常州 102 医院 电脑室 许 鹰

CCED 是一种字处理、划线制表与数据加工融为一体的中文集成字处理软件。用户既可以方便地利用四个光标移动键任意画线,又可以利用程序本身的自动制表功能迅速地生成各种表格,对表中的每个格子都可以随时改变行宽列距。填表中的数据、文字时表线自动保护,不会破坏。对表中的数据可以方便地进行数据计算,还能自动打印 dBASE 的数据库表格。字处理时除可以进行各种字、词、块的拷贝、移动、复盖等多种功能外,还可做到中英文混排标点首尾禁则、首尾对齐。CCED 的使用环境可以适应各种显示器及显示行数。但是 CCED 有一个最大的不足,那就是不能进行子

目录操作,虽然在它的按装说明中提出了解决的办法,但是在实际操作中往往困难较大,不够方便,现介绍一种十分简单而方便的方法,使 CCED 可以在任何路径下随意调用。

首先,将 CCED 的几个文件拷入 C 盘根目录,然后在根目录中调用一次,这时 C 盘根目录中就会自动产生一个 CCED.OV1 的复盖文件,大小只有 256 字节,只要将这个文件拷入需要调用 CCED 的任何软盘或硬盘的子目录中,那么就可以在含有 CCED.OV1 的任何子目录中随时调用 C 盘根目录中 CCED 的各类文件了。如果在调用时还想使用在线帮助(按 Ctrl+J),那么只要把 CCED.OVL 拷入相应的子目录就可以了,使用起来十分方便。

在各个不同的子目录中,可以通过 Shift+F7 设定互不相同的屏幕参数,即可以在不同的子目录中产生不同的前后景颜色,不同的画线线型,不同显示行数的工作屏幕,使用起来更加生动形象。但如果使用的是单色显示器,则在显示方式中应输入 18,否则无法正确显示。

如何自编 Park

盐城师专 葛芝宾

硬盘是微机系统的重要外部设备,它随主机电源的通断而运停。由于读写磁头与盘片的接触方式,在开关机的瞬间,磁头与盘面的磨损加大,常会使硬盘因划伤而出现故障或报废。减少硬盘被划伤的方法,是在关机断电前运行 park.com 程序,把磁头移至盘片中心附近的非数据存储区(具有断电自动 park 功能的硬盘系统不在此例)。然而,我们在实际工作中发现,不少 park 程序都受硬盘型号或机型的局限,而有些 park (PC DOS 系统盘提供或随机 Utility 软件盘提供)存在错误,不能正常运行。例如在 PC DOS 3.20 下运行该系统提供的 park,会显示“Unknown System...”而退出,用 AST 286 Premium Utility Software 盘上的 park.com 锁定该机的 40MB 硬盘,在显示“Disk error”后不能继续。笔者初步分析认为属于程序本身的问题,PC DOS 3.20 版的 park 在运行之初先检查硬盘 BIOS 的 INT 13H 入口段值的高字节是否为 C8H 或 FCH,而事实上 DOS 3.X 的内核初始化后,INT13H 及其它一些 ROM BIOS 的中断向量均被改写,难怪其 park 不认识系统了。据笔者对不同的 AST 286 机检测,硬盘 BIOS 所在段也不同,有的位于 CE00 段,而有些则在 F000 段。至于 AST 286 机的 park,主要问题在于对硬盘柱面数的计算错误。再如 GW 0520CH、紫金 AT 等机器虽有 park 功能的工具程序,但均为己用。在此情况下,笔者着手自编共用的 park.com 程序,现奉献给读者共享。

我们知道,不同型号的硬盘的参数、操作命令码、硬盘 BIOS(INT 13H)程序不尽相同,但有二点是基本相同的:(一)硬盘 INT 13H 向 DOS 内核提供的功能服务应是一致的;(二)硬盘的柱面编号及分布顺序是一致的,即柱面号总是由外圈向盘心递增的。park 程序的任务就是获取当前系统配置的硬盘数目、柱面数,然后依次将各硬盘的磁头移至最末一道的内侧。这里有二点需说明:(1)不论硬盘的磁头有多少(4、5 或 7 等),移动某一磁头时,所有磁头随之一起沿同一方面移动相同距离(磁道数);(2)由于盘心区域很有限,磁头移出最末柱面不能多于 2—3 道。在介绍 park.com 编程之前,我们先将 INT 13H 的子功能列于表 1,对其各子功能的调用参数如下:

入口参数:AH 子功能号(见表 1)

AL 扇区数(读/写命令)或间隔数(格式化命令)

CH 柱面号低 8 位

CL 高 2 位为柱面号的高 2 位,低 6 位为扇区号

DH 磁头号

ES:BX 读/写的内存缓冲区始址

出口参数:成功 AH=0 CF=0

不成功 AH=错误状态字节 CF=1

对于返回驱动器参数子功能(AH=8H),还返回

DH 磁头数-1

DL 微机系统的硬盘台数

CH 柱面数的低 8 位

CL 高 2 位为柱面数高 2 位,低 6 位为扇区数/道

表 1 硬盘 INT 13H 的子功能

AH=00H	复位硬盘控制器,设置硬盘驱动器参数
AH=01H	读上一次中断 13H 执行后的状态到 AL
AH=02H	读
AH=03H	写
AH=04H	校核(读但不写到内存)
AH=05H	格式化指定磁道
AH=06H	格式化指定磁道并设置坏道标志
AH=07H	从指定道格式化整个硬盘
AH=08H	读驱动器参数
AH=09H	设置驱动器参数到控制器
AH=0AH	长读
AH=0BH	长写
AH=0CH	寻道
AH=0DH	复位控制器并设置驱动器参数
AH=0EH	读控制器缓冲区
AH=0FH	写控制器缓冲区
AH=10H	测试驱动器是否准备好
AH=11H	重回 000 道(校准)
AH=12H	控制器内存诊断
AH=13H	驱动器诊断
AH=14H	控制器内部诊断

从表 1 可见,可用来移动磁头的子功能有 02H、04H、0CH 等号子功能,但采用不需内存缓冲区的子功能较为方便。下面所附的即是在 DEBUG 下编写的简短而实用的 park.com,在此仅作二点说明:(1)由于软盘 INT13H 未提供 08H 号子功能,若系统未安装硬盘,可据 08 号子功能调用成功与否确定(不成功,CF=1)。(2)在移动磁头前,未采用惯常的 11H 号子功能让磁头回 000 道,而用 10H 号子功能测试驱动器状态,这样可使磁头(若位于较内圈时)移动完成得快些。在驱动器无故障时,重校准(11H 号)不是必须的,而减少磁头来回移动则是有益的。

C>debug

—a100

cs:0100 jmp 180

cs:0103

—a10a

cs:010a db 'PARK Hard Disk Heads Utility.',0d,0a,24

cs:012a db 'No Drive Installed!',0d,0a,24

cs:0140 db 'Disk Error!',0d,0a,'Drive(s) Was Not
Parked!',0d,0a,24

cs:0168 db 'Power System OFF',0d,0a,24

cs:017c

a180

cs:0180 mov dx,10a

cs:0183 mov ah,09

cs:0185 int 21

cs:0187 mov ah,08


```

cs:0189 call 1f0
cs:018c jnb 197
cs:018e mov dx,12a
cs:0191 mov ah,09
cs:0193 int 21
cs:0195 int 20
cs:0197 or dl,dl
cs:0199 jz 18e
cs:019b mov [0104],dl
cs:019f rol cl,1
cs:01a1 rol cl,1
cs:01a3 xchg ch,cl
cs:01a5 and cx,03ff
cs:01a9 mov ax,cx
cs:01ab inc ax
cs:01ac mov [108],ax
cs:01af mov [105],cx
cs:01b3 mov ah,10
cs:01b5 call 1f0
cs:01b8 jnb 1bf
cs:01ba mov dx,140
cs:01bd jmp 191
cs:01bf dec byte ptr [104]
cs:01c3 mov ax,0401
cs:01c6 call 1f0
cs:01c9 jb 1d8
cs:01cb inc word ptr [105]
cs:01cf mov ax,[105]
cs:01d2 cmp [108],ax
cs:01d6 jnb 1c3
cs:01d8 cmp byte ptr [104],00
cs:01dd jz 1e5
cs:01df inc byte ptr [107]
cs:01e3 jmp 1b3
cs:01e5 mov dx,168
cs:01e8 mov ah,09
cs:01ea int 21
cs:01ec int 20
cs:01ee mov dl,[107]
cs:01f2 or dl,80
cs:01f5 mov dh,00
cs:01f7 mov cx,[105]
cs:01fb xchg ch,cl
cs:01fd ror cl,1
cs:01ff ror cl,1
cs:0201 or cl,01
cs:0204 int 13
cs:0206 ret
cs:0207
-r cx
CX=0000
:0108
n park.com
-w

```

需 ROM BASIC 支持的软盘改造

四川涪陵地区林业局 曾礼伟

在 IBM-PC/XT 及其兼容机上运行的需 ROM BASIC 支持的图形软件,在不带 ROM BASIC 的机上往往不能运行或经过一定的加密处理而不能 COPY。本文据此提出一种改造这类软盘的方法,使之能在不带 ROM BASIC 的机上运行并且能 COPY 其中的各文件。

改造的基本思路是:一般这类图形或游戏软件运用的是 BASICA.COM,须在 ROM BASIC 的支持下才能运行,若能将 BASICA.COM 改换成 BASICA.EXE,问题即可解决。但往往这类软件经过了一些加密处理(如修改文件属性、改变簇号、隐含子目录等)使之不能 COPY。所以首先要解密。下面,依次给出改造的方法。

1. 用 DISKCOPY 命令备份软盘一张。
2. 用调试程序 DEBUG 的 D 命令将备份盘的文件目录项列出,并记下各文件名和各文件的 1A~1BH 字节和 1C~1FH 字节的内容(1A~1BH 字节为文件的起始簇号,有些加密盘将此改变;1C~1FH 字节为文件长度。记下这些数据的目的为以后确定文件名用)。
3. 用检查磁盘状态命令 CHKDSK [d:]/V 对备份盘进行检查,此时隐含文件和子目录都将显示出来。记下或打印所有文件名和子目录名。
4. 用恢复文件命令 RECOVER [d:]对备份盘进行恢复。使用此命令时,备份盘上的文件目录将被认为是坏的而被重新设置在新的系统目录下,所有文件将以 FILE nnnn.REC 的形式恢复到指定的盘片上(子目录也以文件的形式出现)。
5. 用调试程序 DEBUG 的 D 命令将刚恢复的各文件目录项列出,并记下各文件的序号(即 nnnn)和各文件的 1A~1BH 字节和 1C~1FH 字节的内容。
6. 将步骤(5)与步骤(2)记下的内容进行比较(即比较各文件的簇号和长度)可确定出原文件的真实文件名。各文件名一一确定出后,便可用 REN 命令将恢复盘中的 FILEnnnn 文件更名为真实文件名。若原加密时用的是怪字符文件名,可更名为可辨认的文件名;若加密盘改变了原簇号,可根据文件长度和建立文件的顺序,找出真实文件名;在原加密盘中的子目录经步骤(4)后恢复为长度为 1024 的 FILEnnnn.REC 文件,经与步骤(3)记录的文件名和子目录名进行比较,可找出子目录名。更名完后,解密工作完成。

备份盘经上述步骤解密后,即可对用 DIR 列出的各文件进行 COPY 了。对于经步骤(4)恢复的各 FILE nnnn.REC 文件,其代表属性的 0BH 字节令为 00,使用者可根据对文件属性的要求进行改变。

用 DIR 命令可找到 A 盘中的 BASICA.COM 文件,然后用 BASICA.EXE 文件替换它(下转第 5 页)

最新病毒检测及清除软件 VIRUSCAN Ver7. 2v77

梁伟杰

自从 1989 年美国 McAfee Associates 公司推出 VIRUSCAN 软件后, VIRUSCAN 就以其惊人的杀“病毒”能力而风靡全球电脑界。VIRUSCAN 更不断推陈出新, 以对抗形形色式的新病毒, 使得它们无所遁形。

今年 5 月份推出的 VIRUSCAN 最新版本 Version 7. 2v77 (简称 SCAN77), 更是强劲无比。它可以检测 243 种病毒及它们的若干变种, 总共 503 种, 并可清除其中的 98% 共 492 种病毒。这些病毒占目前个人电脑上流行病毒的 96%。SCAN77 可在任何 PC 及兼容机上运行, 只要求 256KB 内存及 DOS2. 0 以上版本。

SCAN77 比其前身 SCAN76 可以多检测 4 种病毒, 可以多清除 19 种病毒, 如 Black Monday, Print Screen—2, 3066 等病毒。时间上则比 SCAN76 快 5%。

SCAN77 中包括 SCAN. EXE, CLEAN. EXE 以及 VALIDATE. COM 三个主要文件。以下分别介绍其使用方法。

SCAN. EXE 是用于检测病毒的, 其命令格式如下:

```
SCAN d1: ... d10: /A/AV/CV/D/M/NOMEM/E.  
xxx/EXT d: filename/FR/MANY/NLI/NOBREAK/  
NOPAUSE/REPORT d: filename/RV
```

注: d1: ... d10: 为要检测的驱动器号, 驱动器号后可带有路径名, 文件名。“/”后的参数均可以缺省, 缺省时不执行该项相应的操作。

各命令参数的意义如下:

/A: 检测所有文件, 包括数据文件 (若不使用此操作, SCAN77 只检测. BIN, . COM, . EXE, . OV?, . PGM, . SYS, . XTP 以及用/E. xxx 参数指定的扩展名。

/E. xxx: 操作如上所述, 可以同时加入几个扩展名。

/AV: 对不同文件加入不同的 CRC (循环冗余校验码)

/CV: 检验文件的 CRC

/RV: 删去文件的 CRC, 不能与/AV 同时使用。

/FR: 用法文代替英文显示。

/D: 当检测到该文件有病毒时删除该文件。此参数一般只用于对付 CLEAN. EXE 不能清除的病毒。

/EXT d: filename: 使用记有病毒特征串的外部文件来检测用户自己发现的病毒。该文件可用任何编辑器来写, 如 PE, Turbo Basic 的编辑器等, 记录格式如下:

NEW virus in. EXE——此为说明

“5D FF 4E” NEW VIRUS [NVI]

特征串 名称 代码

注意, 每行结尾必须用<CR/LF> (回车/换行), 特征串可带有“?”, “*”等万用符。

/M: 对内存进行所有病毒的检测, 注意: 即使不用/M, SCAN77 也自动对内存检测 Disk killer, Taiwan—3, 1554, Brain 等危害性大的隐蔽病毒, 当检测到内存有这 28 种病毒之一存在时, 它会建议你重新启动, 因若在此时进行 SCAN, 会带来大面积的破坏和感染。此点在使用时应多加注意。

/NOMEM: 与上相反, 不对内存进行检测。只有当时肯定内存没有病毒时才可使用此参数。

/MANY: 对多个软盘进行检测。

/NLI: 不检测用 LI. EXE 压缩的文件。

/NOBREAK: 检测期间, 不允许 Ctrl—C/Ctrl—Break 中断。

/NOPAUSE: 检测期间, 不允许屏幕的暂停上卷。

/REPORT d: filename: 建立一个病毒感染报告文件, 存放检测结果。

当检测完毕, 会显示检测结果, 并返回一代码。其意义如下:

0: 未发现病毒。

1: 发现一个或多个病毒存在。

2: 程序有错误。

CLEAN. EXE 用于清除病毒, 其格式如下:

```
CLEAN. EXE d1: ... d10: [Virus ID] /A/E. xxx/FR/  
MANY/M/REPORT d: filename/ EXT d: filename/  
NOPAUSE
```

[Virus ID] 为病毒代码, 由 SCAN 时所得结果给出。注意: 病毒代码必须严格按格式书写。其余参数的意义同 SCAN. EXE。

当用 CLEAN 清硬盘分区表中的石头病毒时, 很可能会破坏分区表, 此时最好将硬盘的数据先备份, 再清病毒。

VALIDATE. COM 主要用于对文件的 CRC 校验。在 SCAN77 盘上所有文件均加有 CRC 码。如执行以下操作, 便可得到:

A> VALIDATE SCAN. EXE

FILENAME: SCAN. EXE

SIZE: 59, 123

DATA: 04—25—1991

FILE AUTHENTICATION

Check Method 1: 0DF0

Check Method 2: 0989

如果不是这样, 那么你的 SCAN 程序已被修改, 或已感染病毒了。

建议你吧 SCAN77 做成一张可启动软盘, 每次操作最好先启动, 这样就绝对安全了。若能配合上专清“国产”病毒的公安部 SCAN2. 0, 就真的能防百毒了。

Goldbach 奇数猜想的 BASIC 程序

汕头四中 陈君佐

从 1991 年第 1 期的《数学学报》，得知著名数学家陈景润仍在攻哥德巴赫猜想的关，并进而追阅他和王天泽同志发表在 1989 年第 5 和第 6 期《数学学报》和 1989 年第 11 期《中国科学》上的论文，通过对 L 函数零点分布及算术数列中的素数分布两问题的研究，证明奇数 $N \geq e^{11.503}$ 都能表为三个奇素数之和的结论。

为配合他们的研究，我把电算奇数猜想的 BASIC 程序介绍于下，这程序能完整的，并没有重复的把奇数 N，一个不缺，也一个不多的，把满足：

$$N = P_1 + P_2 + P_3$$

的解数 T(N)展现在读者面前。

如 33 这个奇数，可表为

- 1 33=3+7+23
- 2 33=3+11+19
- 3 33=3+13+17
- 4 33=5+5+23
- 5 33=5+11+17
- 6 33=7+7+19
- 7 33=7+13+13
- 8 33=11+11+11

用潘承洞教授《哥德巴赫猜想》专著中的符号，就是 $T(33)=8$ 。

今年是 1991 年，这是一个不大的奇数。LASER—310 微机，和 IBM PC/XT 微机都一致，并 5 次显示，顺序一个不变的给出 $N=1991$ 的 3285 组解，即

$$T(1991)=3285。$$

考虑到偶数猜想 BASIC 程序（见 1991 年本刊第 3 期）还不是最佳的，故再度修改，配以 GOSUB 300 语句，使奇数猜想比偶数猜想程序更优化，节约上机时间。

```
10 REM 哥德巴赫奇数猜想程序
20 INPUT "请送入奇数=";N
30 FOR X=3 TO N/3 STEP 2
40 FOR I=2 TO SQR(X)
50 IF X/I=INT(X/I) THEN 100
60 NEXT I
70 A=0;B=B+1;GOSUB 300
100 NEXT X
120 END
300 R=N-X
310 FOR Y=X TO R/2 STEP 2
320 FOR I=2 TO SQR(Y)
330 IF Y/I=INT(Y/I) THEN 430
340 NEXT I
350 Z=R-Y
360 FOR I=2 TO SQR(Z)
370 IF Z/I=INT(Z/I) THEN 430
380 NEXT I
```

```
390 A=A+1;M=M+1
```

```
410 PRINT M;" ";B;" ";;" ";N;"=";"X;"+";"Y;"
"+";"Z
```

```
430 NEXT Y
```

```
440 RETURN
```

当用 RUN 运行程序后送入 33 时，IBM PC/XT 机很快给出：

- 1 1 1 33=3+7+23
- 2 1 2 33=3+11+19
- 3 1 3 33=3+13+17
- 4 2 1 33=5+5+23
- 5 2 2 33=5+11+17
- 6 3 1 33=7+7+19
- 7 3 2 33=7+13+13
- 8 4 1 33=11+11+11

我编这个程序，自己感到十分满意，因它不仅给出 33 个这奇数，可表为 8 组不同的三素数之和的解答，而且给出 $X=3$ ，有 3 组不同的解， $X=5$ （第二个奇素数!!），有 2 组不同的 y 和 z 值的解， $X=7$ ，有 2 组不同的 y 和 z 值的解， $X=11$ ，只有一组满足 $N=P_1+P_2+P_3$ 的解，合共 $T(33)=8$ 。

由于没有打印机，故 $T(1991)=3285$ 又得手抄。下面是部分结果：

- 250 6 19 1991= 17+233+1741
- 300 6 69 1991= 17+953+1021
- 350 8 16 1991= 23+269+1699
- 400 9 6 1991= 29+ 89+1873
- 450 10 2 1991= 31+ 53+1907
- 500 11 6 1991= 37+131+1823
- 550 12 24 1991= 41+293+1657
- 600 12 74 1991= 41+911+1039
- 650 14 15 1991= 47+281+1663
- 700 15 12 1991= 53+197+1741
- 750 16 1 1991= 59+ 59+1873
- 800 16 51 1991= 59+739+1193
- 850 18 4 1991= 67+191+1733
- 900 19 25 1991= 71+353+1567
- 950 20 3 1991= 73+197+1721
- 1000 21 25 1991= 79+929+ 983
- 1050 22 49 1991= 83+937+ 971
- 1100 24 1 1991= 97+107+1787
- 1150 25 25 1991=101+311+1579
- 1200 25 75 1991=101+859+1031
- 1250 27 21 1991=107+433+1451
- 1300 28 23 1991=109+773+1109
- 1350 29 45 1991=113+839+1039
- 1400 31 15 1991=131+277+1583
- 1450 31 65 1991=131+919+ 941
- 1500 33 7 1991=139+353+1499
- 1550 34 33 1991=149+691+1151
- 1600 36 7 1991=157+281+1553

1650 38 6 1991=167+241+1583
 1700 39 10 1991=173+331+1487
 1750 40 18 1991=179+389+1423
 1800 41 26 1991=181+797+1013
 1850 42 45 1991=191+739+1061
 1900 44 18 1991=197+487+1307
 1950 45 22 1991=199+761+1031
 2000 47 17 1991=223+719+1049
 2050 48 44 1991=227+853+ 911
 2100 50 25 1991=233+641+1117
 2150 52 4 1991=241+269+1481
 2200 53 22 1991=251+461+1279
 2250 54 18 1991=257+571+1163
 2300 56 1 1991=269+269+1453
 2350 57 11 1991=271+461+1259
 2400 59 22 1991=281+601+1109
 2450 61 8 1991=293+419+1279
 2550 64 13 1991=313+659+1019
 2600 66 15 1991=331+647+1013
 2650 68 26 1991=347+787+ 857
 2700 71 3 1991=359+401+1231
 2750 73 9 1991=373+587+1031
 2800 76 4 1991=389+421+1181
 2850 78 16 1991=401+599+ 991
 2900 80 22 1991=419+733+ 839
 2950 83 2 1991=433+461+1097
 3000 86 10 1991=449+613+ 929
 3050 89 7 1991=463+617+ 911
 3100 92 10 1991=487+683+ 821
 3150 96 7 1991=509+643+ 839
 3200 101 5 1991=557+613+ 821
 3250 107 2 1991=593+641+ 757
 3260 108 8 1991=599+691+ 701
 3274 113 1 1991=619+653+ 719
 3281 116 1 1991=643+647+ 701
 3282 117 1 1991=647+653+ 691
 3283 117 2 1991=647+661+ 683
 3284 118 1 1991=653+661+ 677
 3285 119 1 1991=659+659+ 673

总之, $T(1991)=3285$, 是千真万确的, 不多也不少。

其它结果有: $Y(1993)=3182$, $T(1995)=2029$, $T(1997)=3309$. 含素数因子“3”的奇数, $(1+1+1)$ 的解数反而少。

华罗庚公式:

$$T(N) = \frac{N^2}{2L^3} C(N) + O(N^2 L^{-4} \log L)$$

也是偏大了。

按华罗庚公式, $N=1991$ 计算:

$$T(1991) = 10299.621$$

这 10299.621 是由主项算出的, 如果连“O 项”计入, 偏差就更大。如用(经验公式):

$$T(N) = \frac{[\Pi(N)]^2}{3L}$$

算

$$T(1991) = \frac{300^2}{3 \times 7.5963923} = 3949.2431$$

会误差小些!!

快速排序“快”在何处?

四川万县师专 应 宏

排序是计算机处理非数值问题的一类重要运算。据有人统计, 计算机整个运行时间的 25% 以上是花在数据的排序上。显然, 排序算法的效率高低是至关重要的, 人们为设计高效率的排序算法作了许多努力。1962 年, 由 Hoare 提出的快速排序(quick sort)算法, 对批量的数据元素进行排序时, 它比简单而直观的选择排序、插入排序、冒泡排序效率高、速度快。“快”的原因何在? 要回答这个问题, 我们先给出关于序列 $L = (a_1 a_2 a_3 \dots a_n)$ 逆序的概念。

序列 L 的一个逆序, 是指存在 $i, j (1 \leq i, j \leq n)$ 使得 $i < j$ 有 $a_i > a_j$ 。无疑, 序列 L 至多可以有 $n(n-1)/2$ 个逆序。

如下结论也是显然的:

假设有 $a_i, a_j \in L$, 若 (a_i, a_j) 构成逆序, 且在 a_i, a_j 之间有 k 个元素满足 $a_j < x < a_i$, 有 r 个元素满足 $x = a_i$, 有 s 个元素满足 $x = a_j$, 其余的元素满足 $x < a_j$ 或 $x > a_i$, 则将 a_i, a_j 对换后, 序列 L 的逆序数减少 $2k + r + s + 1$ 。

例如, 有如下序列:

25 8 59 7 20 45 18 ①

按非递减顺序(后同), 序列①中共存在 11 个逆序, 显然, $(25, 18)$ 构成逆序。若将 25, 18 两元素对换则得到序列

18 8 59 7 20 45 25 ②

②中还存在 8 个逆序, 减少的逆序数正是前面结论所指出的数目。若序列 L 中相邻两数据元素构成逆序, 则将此两元素对换后, 序列 L 就减少一个逆序。

要对 N 个数据元素进行排序(不妨设 N 个元素存储于向量 $A[1 \dots N]$ 中), 实质上就是要消除 $A[1], A[2], \dots, A[N]$ 中的所有逆序(相对某种顺序而言)。试图提高排序的效率, 就应着眼于如何提高消除逆序的效率。冒泡排序算法之所以不是高效率的排序算法, 原因在于每次对换只在相邻元素之间进行。下面是对序列①施行冒泡排序算法第一轮的过程:

25 8 59 7 20 45 18

8 25 59 7 20 45 18

8 25 7 59 20 45 18

8 25 7 20 59 45 18

8 25 7 20 45 59 18

8 25 7 20 45 18 59 ③

经过第一轮的五次对换, 得到的序列③中还有 6 个逆序, 显然每对换一次仅消除一个逆序。根据前面的结论, 如果允许对换不只是在相邻元素之间进行, 那么每对换一次就有可能消除多个逆序。由 Hoare 提出的

快速排序算法就允许一个元素可以跳越几个元素的位置而移动,即不相邻的两个元素之间进行对换,从而可以消除多个逆序,达到提高排序效率的目的。

下面是对序列①施行快速排序算法第一轮的过程:

```

25 8 59 7 20 45 18 25 送 x, 18 < x
↑i                      ↑j 移动 18, 修改 i
18 8 59 7 20 45 □ 8 < x, 修改 i
      ↑i                      ↑j
18 8 59 7 20 45 □ 59 > x, 移动 59
      ↑i                      ↑j 修改 j
18 8 □ 7 20 45 59 45 > x, 修改 j
      ↑i                      ↑j
18 8 □ 7 20 45 59 20 < x, 移动 20
      ↑i                      ↑j 修改 i
18 8 20 7 □ 45 59 7 < x, 修改 i
      ↑i ↑j
18 8 20 7 □ 45 59 i=j, x 送 i 位置
      i ↑ j
18 8 20 7 25 45 59 ..... ④

```

上面方框表示元素移动后的位置(空位)。经过第一轮的四次移动,得到的序列④中仅存在 4 个逆序。不难发现,算法中每次填补空位的方法实际上反复实现了两个非相邻元素(且构成逆序)之间的对换。从而使得一次对换消除了多个逆序。

除快速排序算法外,1959 年 Shell 提出的缩小增量法(即 Shell Sort),1964 年由 Williams 首先提出而经 Floyd 改进的堆排序(Heap Sort)算法,效率也都比冒泡排序高。仔细分析 Shell 排序算法,它不仅由于移动一个元素可以消除多个逆序,而且还由于每次采用新的增量排序,并不破坏老的增量已有的排序效果。至于堆排序,每次从堆顶删除时,空位的移动也是对构成逆序的非相邻元素进行对换,使得一次对换可能消除多个逆序。因此,在设计排序算法时,我们总是希望为消除逆序而进行对换的两个元素之间的距离能够拉开,以提高消除逆序的效率。快速排序、Shell 排序和堆排序都运用了这一思想,只不过 Shell 排序算法在运用这一思想时表现得典型、更直截了当。

两个高效率的排序程序

武汉水利电力学院教务处 李军杰

应用最广泛的快速排序法,是目前以比较为基础的排序中速度最快的方法,它的平均时间复杂度为 $O(n \log_2 n)$ 。该算法可以写成相当短的递归形式(绝大多数书籍都这样处理,采用类 PASCAL 或 PASCAL 描述其算法)。若把它写成非递归算法,程序会变得较复杂,而且对于那些不熟悉如何把递归化为非递归以及用不支持递归的高级语言(如: BASIC、FORTRAN

等)编程的用户,在工作中想要应用快速排序法就较困难。笔者在理解了快速排序法实质的基础上,经过上机实践和改进,编成了形式较为简洁的非递归 BASIC 排序程序(见程序 1)。该程序以独立的子程序形式出现,用户可随处调用。在应用中,首先定义数组 X 的大小,不妨设为 N,并为 X(N)赋初值;二维数组 S(栈)相应为 $S(2 * \log_2(N), 2)$ 。程序 1 中 110 句的 L、U 为数组 X 的下界、上界值,用户可加以修改。

堆排序法是一个既在平均情况下,又在最坏情况下,都是时间复杂度为 $O(n \log_2 n)$ 的有效算法。此法的优点在于不占用额外的存储空间。程序 2 是经过优化处理的堆排序程序。数组 B(N)为待排序数组,事先应赋值。语句 90—150 用于建堆,语句 160—250 用于排序和重建堆,直至 B 成为有序数组。

以上两个程序均在 PC 机上运行通过,对大量数据进行排序时,效果极佳。

程序 1

```

100 'Quicksort for BASIC
110 L=1;U=100;P=0
120 I=L;J=U;F=-1
140 IF X(I)<=X(J) THEN 170
150 SWAP X(I),X(J);F=-F
170 IF F=1 THEN 200
180 J=J-1;GOTO 210
200 I=I+1
210 IF I<J THEN 140
220 IF I+1>=U THEN 250
230 P=P+1;S(P,1)=I+1;S(P,2)=U
250 U=I-1
260 IF L<U THEN 120
270 IF P=0 THEN 310
280 L=S(P,1);U=S(P,2);P=P-1;GOTO 120
310 RETURN

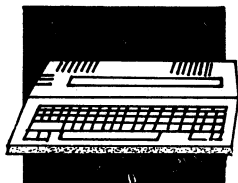
```

程序二

```

10 'Heapsort for BASIC
50 DEFINT F,I,J,K,N
90 FOR I=1 TO N
100 J=1
110 WHILE J>1
120 F=INT(J/2);IF B(F)>=B(J) THEN 150
130 SWAP B(F),B(J);J=F
140 WEND
150 NEXT I
160 I=N
170 WHILE I>1
180 SWAP B(I),B(1);I=I-1
190 J=1;K=J+J
200 WHILE K<=I
210 IF (K<I AND B(K)<B(K+1)) THEN K=K+1
220 IF B(J)>=B(K) THEN 250
230 SWAP B(J),B(K);J=K;K=J+J
240 WEND
250 WEND

```

CEC—I 高级动画制作技术

学习机之友

四川雅安教育学院微机室 黄一禾

在 CEC—I 或 APPLE II 上设计生动逼真的动画,常感到有几个困难,一是绘图麻烦,二是运动困难,三是内存开销大。虽然制作动画的方法很多,但不少常见的方法都或多或少存在上述某方面的不足。下面将结合实例介绍较先进实用的动画制作技术,其主要点是利用工具软件绘制运动分解图,并运用预移图序法原理剪辑成动画分解图,最后利用汇编语言程序显示这些分解图,形成动画,这种动画制作技术的主要优点是绘图方便、生成的动画动感强、逼真、速度很快,而且占用内存较少。缺点是运动显示程序编写较麻烦。

一、预移图序法原理

我们知道,动画电影是由一幅幅分解动作的图画连续高速放映后生成的。如果我们在计算机上将动画活动过程的主要瞬间变化的图象预先画好,然后按顺序快速显示出来,同样可产生动画效果。如图一(文后)有七幅长 28 个点、宽 10 个点的不同瞬间状态的直升飞机分解图,在每幅图中,仅主螺旋桨和尾螺旋桨的位置不同,若在高分辨显示页的相同位置上反复依次快速将这七幅图形显示出来时,就会出现一驾悬停在空中,其螺旋桨不停旋转的直升飞机,显然这仅仅是动画的一部分,即物体自身在不停的变化。

现在主要讨论在物体自身不断变化的同时,还要作上下左右运动的动画。物体的上下运动比较简单。根据 CEC—I 高分辨显示页的结构,只须在原位置擦去图形,随即在上或下一点的新位置上又将其显示出来,物体就上下运动了。注意:在每一显示位置上都需要将七幅分解图依次显示一遍,以产生自然平滑的动感。

物体的左右运动比较麻烦,主要原因是 CEC—I 屏幕上任一行的第 $7k$ 至第 $7k+6$ ($k=0,1,\dots,39$) 的 7 个点,用内存中的一个字节来表示,即高分辨显示页中每个字节的第 0—6 位分别代表屏幕上某行中从左到右的 7 个点。如果直接将物体分解图的显示位置左移或右移一个字节,那么显示出的图形实际上向左移或向右移了 7 个点。这样虽然是动起来了,但有较强的跳跃感和闪烁感,明显失真。为了解决这一问题,可以模仿动画电影画面安排物体位置的方法,把物体安排在分解图中的不同位置上,(注意:分解图均是大小完全相同的矩形图,如上例中为 28×10 点),即在第一幅分解图中将物体最左边的点安排在矩形左起第二列,(这里留出一列空白点以利显示),在第二幅分解图中将物体最左边的点安排在矩形左起第三列,……,在第七幅分解图中将物体最左边的点安排在左起第八列,也就是将每一幅图形中的物体安排在相差一个点的位置上,见图 2 中的七幅直升飞机分解图。这样在相同位置上,从 1 号图依次显示到 7 号图时,虽然显示位置没变,但直升飞机却前进了七个点,刚好一个字节,接着

将显示位置右移一个字节,又将这七个分解图依次显示一遍,……。于是就得到了一幅直升飞机在不断前进,其螺旋桨也在不停旋转的动画画面。这种动画没有闪烁感和跳跃感,平滑自然,非常逼真。同样,若从 7 号图形到 1 号图形显示一遍后,显示位置左移一个字节,再在新位置上反序显示一遍,……,就会得到直升飞机不断后退的动画画面。

概括起来,在同样大小的矩形内,绘制出形状基本相同的物体,并使这些物体的起始点依次相差一个点,形成若干动画分解图,调用时在同一位置上依次将分解图显示后,再在新位置上依次显示(即向左或向右移一个字节),不断重复下去,就会产生出物体本身在不断变化,同时又在不断向左或向右运动的动画效果,这种产生动画的方法一般称为预移图序法。

二、图形绘制和剪辑

图形绘制和剪辑历来比较麻烦。为了绘制出符合上述动画要求的分解图,一般是利用专用绘图软件,这使得绘图和剪辑变得十分方便。本文主要使用 A. C. E 软件的绘图系统(1 号盘)作为制作分解图形的工具。利用 A. C. E 软件绘图是十分方便的,这里不再加以介绍,下面主要谈谈有关剪辑和转换等问题。

将所需物体图形绘制完毕后,进入绘图系统的窗口剪裁,将物体图形剪裁下来,操作时实际上就是用一矩形方框将图形框起来,得到的矩形图块就是前面所说的分解图了。在确定矩形分解图左右两条边界线时,应按预移图序法的思想设取,即第一个分解图左边界线应与物体最左边的点相隔一个空白点;右边界线应与物体最右边点相隔 7 个空白点。第二个分解图左边界线与物体最左边点相隔 2 个点,右边界线与物体最右边点相隔 6 个空白点;……第七个分解图左边界线与物体最左边点相隔 7 个点,右边界线与物体最右边点相隔 1 个点。矩形分解图的上下边界线始终不变,这样所截取的各分解图完全一样大,便于快速显示,图 2 中的七幅直升飞机分解图就是按上述方法截取的。

在剪裁时还应注意,当剪裁好一个分解图后,应立即存盘,并给分解图文件编上号,以示区别,然后再剪裁下一个。在 A. C. E 系统中,这类文件是以窗口图形文件存取的,文件类型为 G,文件名加上前缀 W1。这类图形文件还不能直接在 DOS 3.3 系统下调用,还要进行转换,转换方法很简单,步骤如下:

①引导 DOS 3.3 系统,键入 `POKE 44272,4`。

②插入文件盘,键入 `LOCK W1.f` (f 是窗口文件名)。此命令执行后,磁盘中的 G 类文件 W1.f 已转换为可在 DOS 3.3 系统下调用的 B 类图形文件了。

每一个窗口图形文件经过转换后,其首地址和长度均未变,如果窗口图形的大小不同,则文件的首地址

和长度也将不同,但其末地址始终是一致的,均为 \$7FFF。应该指出,每个窗口文件的最末 8 个字节是文件的描述项,不是图形数据,其中:

\$7FF8 单元为图形纵向长度,以点为单位。

\$7FF9 单元为图形横向长度,以 7 个点为单位,例如若 (\$7FF9)=4,则图形的实际横向长度为 $4 \times 7 = 28$ 个点。

\$7FFB 单元为图形数据首地址(低位)。

\$7FFC 单元为图形数据首地址(高位)。

\$7FFD 单元为图形文件长度(低位)。

\$7FFE 单元为图形文件长度(高位)。

为了调用方便,一般应将这些分解图形文件合并起来,形成一个完整的图形文件。本文后面所附数据中 \$6000~\$6117 就是图一的七幅分解图合并而成的,其中每幅分解图的首地址位于 \$6334~\$6341。\$6118~\$622F 就是图二的七幅分解图合并而成的,其中每幅分解图的首地址位于 \$6326~\$6333。(低位在前,高位在后)。

至此,各分解图形的绘制、剪辑、转换和合并工作全部完成。当然,读者也可以而选用其它自己较熟悉的方法来得到物体的运动分解图,同样也可以不用移点剪裁分解图的方法得到预移分解图,而用程序将统一剪裁得到的数据编辑成预移数据。

三、动画的实现

有了动画预移分解图后,动画的实现就是将分解图依次快速地在屏幕上显示出来。由于客观物体的活动过程各不相同,因此,实现动画的程序也各不相同,本文仅给出一种较简单易行的实现动画程序。

1. \$627A~\$629F,直升机从左向右飞行程序。程序在初始位置依次显示图二中的七幅分解图后,再在新位置上重复显示。新位置的纵坐标保持不变,横坐标为旧位置加 1 个字节。

\$627E 单元为横向起始坐标值。

\$6283 单元为直升机横向飞行分解图形数据起始地址表地址低位(高位为 \$63)。

\$6292 单元为行数据个数,即每行有 4 个数据。

\$6307 此子程序将生成高分辨显示地址表并进入和清除高分辨第二页。

2. \$62A0~\$62B0,直升机悬停在空中程序。此程序将图 1 中七幅分解图反复显示。

\$62A4 单元为直升机悬停分解图形数据地址表地址低位(高位为 \$63)。

3. \$62B1~\$6306 直升机向上飞行程序。

\$62B5 单元为横向起始坐标初值。

\$62C7 单元为纵向起始坐标初值。

\$62D3 单元为列数据个数,即每列有 10 个数据。

\$62E7 子程序将清除刚才显示出的一个直升机分解图,为下一步在新位置显示作好准备。

上述 3 个程序都直接调用了程序 \$6230,此程序是动画显示的核心程序。

4. \$6230~\$6279 在相同位置依次显示图一或图二中的各分解图子程序。

此子程序由三层循环组成,最外层从 \$6234~\$6278,共循环执行 7 次,每次显示一个分解图。第二层从 \$6249~\$626B,共循环执行 10 次,每次将分解图的一行显示数据送显示地址显示。以层循环结束后,程序调用了延时子程序,以延长显示停留时间,使动画更清楚自然。第三层为 \$6255~\$6263,共循环 4 次,每次将分解图的一个数据送显示地址显示。在这个子程序中用到的一些初值,大都由程序 1,2,3 所设定。

由于本例中的各分解图形数据量较小,仅 10×4 个。在程序 4 中采用了 X 寄存器变址取得图形数据的方式设计。因此,本例程序只适用于单个分解图形数据量在 1 页(256 个数据)范围内的动画演示,但程序只要稍加修改,即可处理单个分解图形数据量大于 256 个的动画演示了。



图一



图二

6000— 00 60 7F 00 00 20 00 00
6008— 04 70 00 00 06 7C 1F 00
6010— 0C 78 30 00 78 7F 7F 00
6018— 78 7F 3F 00 00 7F 0F 00
6020— 00 10 42 00 00 7C 3F 00
6028— 70 3F 00 00 00 20 00 00
6030— 00 70 00 00 26 7C 1F 00
6038— 0C 78 30 00 79 7F 7F 00

6040— 70 7F 3F 00 00 7F 0F 00
6048— 00 10 42 00 00 7C 3F 00
6050— 00 60 7F 00 00 20 00 00
6058— 02 70 00 00 06 7C 1F 00
6060— 0C 78 30 00 78 7F 7F 00
6068— 70 7F 3F 00 00 7F 0F 00
6070— 00 10 42 00 00 7C 3F 00
6078— 70 3F 00 00 00 20 00 00

6080— 10 70 00 00 06 7C 1F 00
6088— 0C 78 30 00 78 7F 7F 00
6090— 72 7F 3F 00 00 7F 0F 00
6098— 00 10 42 00 00 7C 3F 00
60A0— 00 60 7F 00 00 20 00 00
60A8— 00 70 00 00 07 7C 1F 00
60B0— 0C 78 30 00 78 7F 7F 00
60B8— 70 7F 3F 00 00 7F 0F 00

60C0— 00 10 42 00 00 7C 3F 00
 60C8— 70 3F 00 00 00 20 00 00
 60D0— 08 70 00 00 06 7C 1F 00
 60D8— 0C 78 30 00 78 7F 7F 00
 60E0— 74 7F 3F 00 00 7F 0F 00
 60E8— 00 10 42 00 00 7C 3F 00
 60F0— 00 60 7F 00 00 20 00 00
 60F8— 00 70 00 00 06 7C 1F 00
 6100— 2D 78 30 00 78 7F 7F 00
 6108— 70 7F 3F 00 00 7F 0F 00
 6110— 00 10 42 00 00 7C 3F 00
 6118— 00 60 7F 00 00 20 00 00
 6120— 04 70 00 00 06 7C 1F 00
 6128— 0C 78 30 00 78 7F 7F 00
 6130— 78 7F 3F 00 00 7F 0F 00
 6138— 00 10 42 00 00 7C 3F 00
 6140— 60 7F 00 00 00 40 00 00
 6148— 00 60 01 00 4C 78 3F 00
 6150— 18 70 61 00 72 7F 7F 01
 6158— 60 7F 7F 00 00 7E 1F 00
 6160— 00 20 04 01 00 78 7F 00
 6168— 00 00 7F 03 00 00 01 00
 6170— 08 40 03 00 18 70 7F 00
 6178— 30 60 43 01 60 7F 7F 03
 6180— 40 7F 7F 01 00 7C 3F 00
 6188— 00 40 08 02 00 70 7F 01
 6190— 00 7F 03 00 00 00 02 00

6198— 00 01 07 00 30 60 7F 01
 61A0— 60 40 07 03 40 7F 7F 07
 61A8— 10 7F 7F 03 00 78 7F 00
 61B0— 00 00 11 04 00 60 7F 03
 61B8— 00 00 7C 0F 00 00 04 00
 61C0— 00 00 0E 00 70 40 7F 03
 61C8— 40 01 0F 06 00 7F 7F 0F
 61D0— 00 7E 7F 07 00 70 7F 01
 61D8— 00 00 22 08 00 40 7F 07
 61E0— 00 7C 0F 00 00 00 08 00
 61E8— 00 02 1C 00 41 01 7F 07
 61F0— 00 03 1E 0C 00 7E 7F 1F
 61F8— 00 7D 7F 0F 00 60 7F 03
 6200— 00 00 44 10 00 00 7F 0F
 6208— 00 00 70 3F 00 00 10 00
 6210— 00 00 38 00 00 03 7E 0F
 6218— 00 16 3C 18 00 7C 7F 3F
 6220— 00 78 7F 1F 00 40 7F 07
 6228— 00 00 08 21 00 00 7E 1F
 6230— A9 00 85 03 0A A8 B9 34
 6238— 63 8D 58 62 C8 B9 34 63
 6240— 8D 59 62 A2 00 A9 00 85
 6248— 02 A4 02 B9 08 64 85 05
 6250— B9 48 63 85 04 A0 10 BD
 6258— 78 60 EA EA 91 04 E8 C8
 6260— C0 14 90 F3 E6 02 A5 02
 6268— C9 0A 90 DD A9 40 20 A8

6270— FC E6 03 A5 03 C9 07 90
 6278— BB 60 20 07 63 A9 FF 8D
 6280— 56 62 A9 26 8D 37 62 8D
 6288— 3E 62 EE 56 62 AD 56 62
 6290— 18 69 04 8D 61 62 C9 28
 6298— B0 09 20 30 62 4C 8A 62
 62A0— 20 07 63 A9 34 8D 37 62
 62A8— 8D 3E 62 20 30 62 4C AB
 62B0— 62 20 07 63 A9 10 8D 56
 62B8— 62 A9 14 8D 61 62 A9 34
 62C0— 8D 37 62 8D 3E 62 A9 70
 62C8— 8D 46 62 CE 46 62 AD 46
 62D0— 62 18 69 0A 8D 69 62 C9
 62D8— 0B 90 09 20 30 62 20 E7
 62E0— 62 4C CB 62 4C A3 62 A9
 62E8— 51 8D 5A 62 A9 04 8D 5B
 62F0— 62 A9 01 8D 6D 62 20 43
 62F8— 62 A9 EA 8D 5A 62 8D 5B
 6300— 62 A9 40 8D 6D 62 60 A9
 6308— 40 85 E6 A0 00 98 48 20
 6310— 11 F4 68 A8 A5 26 99 48
 6318— 63 A5 27 99 08 64 C8 C0
 6320— C0 D0 EA 4C D8 F3 18 61
 6328— 40 61 68 61 90 61 B8 61
 6330— E0 61 08 62 00 60 28 60
 6338— 50 60 78 60 A0 60 C8 60
 6340— F0 60 00 00 00 00 00 00

警察抓小偷

宁夏 中卫县中学高一(2)班 李明

假如你是一名警察,现有一个小偷蹿上了一座六层楼,你必须在他跑到楼顶之前抓住他,不然他就会驾上准备好的飞机逃走。在楼道里,布满了障碍物,小偷碰到障碍物会一跃而过,而你遇到障碍物则必须绕道而行。若在小偷跑到楼顶之前你没有抓住他,那你就重新玩这一局;若你抓住了他,就会听到一阵动听的音乐,计算机也会对你作出一次评价,并询问你是否继续玩。若不玩了,则按“N”回车;若继续玩,只要回车就进入难度更大的下一局。玩过第三局后,计算机就会告诉你,你是一名合格的警察了。

游戏难度逐渐增大,可以照顾到各个层次的人,并伴有英文提示和逼真的音响,使人仿佛置身其中,极能锻炼人的反应能力,以及眼、手、脑的协调能力。

说明:↑、↓、←、→分别控制警察上、下、左、右移动,颜色暗的为小偷,颜色亮的为警察。当警察超过了每层楼的最左端时,就会跑到上一层楼的最右端。

```
5 A $ (1) = "GOOD"; A $ (2) = "BETTER"; A $ (3) =
"EXCELLENT"; FOR T = 1 TO 3
10 GR: COLOR = 3; FOR I = 2 TO 38 STEP 6; HLIN 0, 39
AT I; HLIN 0, 39 AT I + 1; NEXT
20 FOR I = 1 TO 85 + T * 25; PLOT RND(1) * 39, RND
(1) * 37 + 2; NEXT
25 HOME; VTAB 21; PRINT "LEVEL"; T, "TIMES = ";
S; GET A $
30 A = 39; B = 37; M = 5; N = 37
40 COLOR = X; PLOT M, N; M = M - .4 - T/10; IF M < 0
THEN N = N - 6; M = 39; IF N < 2 THEN S = S + 1; GOTO 10
60 X = SCRN(M, N); COLOR = 4; PLOT M, N
70 Q = A; W = B; P = PEEK(49152) - 128; A = A + (P =
21) - (P = 8); B = B + (P = 10) - (P = 11); IF A < 0 THEN A
= 39; B = B - 6
80 IF SCRN(A, B) = 4 THEN 110
90 IF SCRN(A, B) = 3 THEN A = Q; B = W; GOTO 40
100 COLOR = 7; PLOT A, B; COLOR = 0; PLOT Q, W; MU-
SIC 1, 2; GOTO 40
110 FOR I = 1 TO 90; MUSIC I, 10; NEXT; PRINT "YOU
ARE"; A $ (T); "I"; IF T = 3 THEN PRINT "YOU ARE A
QUALIFIED POLICEMAN"
120 INPUT "CONTINUE? (Y/N)"; A $; IF A $ <> "N"
THEN NEXT T; RUN
```


ProDOS 磁盘操作系统入门(续)

廖 凯

第四章 ProDOS 实用程序

为 APPLE II 机系列设计的 ProDOS 实用程序有四套: ProDOS USERS. DISK、ProDOS COPY II PLUS、APA 和 SYSTEM UTILITIES。前三套可在任何 APPLE 机上运行,而后者是为 APPLE IIc 设计的,只能在 APPLE IIe、IIc 和 IIGS 上运行。

在 ProDOS USERS. DISK 上有两个很有用的实用程序: FILER 和 CONVERT。这些程序将允许用户对磁盘和文件进行管理。

一、FILER

在启动 USERS. DISK 进入主菜单后,按 F 便进入 FILER 菜单。FILER 主要分为三部分,其中 FILE 和 VOLUME 是最常用的部分。FILE 部分是对文件进行管理, VOLUME 部分是对磁盘进行管理。FILER 的每一部分都有一个简要的说明(即 TUTOR,按?键)。

使用说明:在 FILER 中使用文件名时,必须以字母开头,最多 15 个字符(包括句号);路径名不能超过 64 个字符(包括斜线)。文件名必须用斜线分开,斜线需放在卷名后面,否则 FILER 将把文件名当作部分路径名。你可以用右和左箭头键对屏幕上的资料进行编辑。你并可随时按 Esc 键返回原位置或回到 FILER 主菜单。

1. VOLUME COMMANDS

这部分有七项功能。要想进入各项功能,只需按前面的字母即可。

FORMAT A VOLUME: 此功能是格式化一张 ProDOS 格式磁盘,用户只要依提示给出槽口号和驱动器号及卷名(磁盘名),或直接按回车接受系统设定的值。如果磁盘已格式化过,系统会问你是否再格式化,在格式化的过程中,磁盘上的数据将被清除。

COPY A VOLUME: 此功能是进行整盘拷贝。用户依提示输入源盘和目的盘的驱动器号和槽口号,也可直接按回车接受系统设定的值。系统会读取源盘的卷名作为目的盘的卷名,你可以按回车接受系统设定或输入自定的新卷名。如果目的盘已格式化过,系统会问你是否取消原卷名。在拷贝的过程中,如果磁盘有问题,系统将有提示。

LIST VOLUMES: 此功能将列印目前所有存在的磁盘的卷名。

RENAME A VOLUME: 此功能允许用户改变指定磁盘的卷名。在目前的卷名被显示后,你可输入/(斜线)和新的卷名,斜线必须放在卷名前面。

DETECT BAD BLOCKS: 此功能允许用户检查磁盘的情况。如果磁盘损坏或有软损伤,屏幕将显示坏块号,并将其删除。

BLOCK ALLOCATION: 此功能将显示在磁盘上块的使用情况。

COMPARE VOLUMES: 此功能允许用户对两个磁盘的资料进行比较,并显示不符的块号。

2. FILE COMMANDS

此项目允许用户在任何磁盘设备上对文件进行管理。在此项目中,可使用通配字符(WILDCARDS)即“=”和“?”来代替文件名。“?”字符将提示每个文件,用户可以回答 Y 或 N 来选择文件,“=”字符是表示所有的文件。在使用此项之前,用户应该对目录、路径名、文件名和部首有所了解。此部分有七项功能,要进入各功能,只要键入前面的字母即可。

LIST PRODOS DIRECTORY: 此功能将显示目录、文件名、文件类型、文件长度、修改日期和块分配的情况,相当于 CAT 命令。

COPY FILES: 此功能允许用户在任何磁盘设备之间拷贝文件。在拷贝过程中,如果在目的盘上有与源盘相同的文件名时,那么系统将提示用户是否需要重拷(复盖原文件)。另外,在拷贝单个文件时,用户可以更换文件名。

DELETE FILES: 此功能相当于 DELETE 命令。在删除一个完整的子目录时,使用通配字符是很方便的。在删除一个目录文件之前,必须先删除目录内的所有文件。

COMPARE FILES: 用于将两个文件的资料进行比较。在拷贝文件之后,可用此功能检验文件的准确性。

ALTER WRITE-PROTECTION: 此功能相当于 LOCK 和 UNLOCK 命令。用户可依提示对文件进行加锁或解锁操作。

RENAME FILES: 用于更换文件名和目录,相当于 RENAME 命令。

MAKE DIRECTORY: 用于在卷目录内建立副目录,相当于 CREATE 命令。注意在卷目录下只能存放 51 个文件。

SET PREFIX: 用于设置路径名。这只允许用户输入文件名或部分路径名以处理其它的文件命令。在文件被装入时,会自动地设置当前的部首。斜线必须放在卷名和目录的前面。如果要加一个副目录到现有的路径名时,可用右箭头键移动光标到行的尾端并开始输入。

3. CONFIGURATION DEFAULTS

此项目是用于设置磁盘驱动设备和选择信息输出的设备。

SELECT DEFAULTS: 此功能是设置当前你的系统。用户通过提示设置源盘和目的盘的驱动器号及槽口号以及显示器和打印机。系统设定打印机的槽口号为 1,

如果打印接口卡在其它的槽口上,那么就必须正确地设定槽口号。

RESTORE DEFAULTS: 此功能是恢复系统设定值。

二、CONVERT

这程序是用于将文件在 ProDOS 和 DOS 3.3 之间进行转换。具有保护的文件和随机存取文件不能被转换。由 ProDOS 主菜单按 C 进入此项目。屏幕前三行显示当前的状态:箭头指出转换的方向,日期可设定,部首(PREFIX)指出当前的路径名。

1. REVERSE DIRECTION OF TRANSFER

此功能是改变传输的方向,由 DOS 到 ProDOS 或由 ProDOS 到 DOS。

2. CHANGE DOS 3.3 SLOT AND DRIVE

允许用户改变 DOS 3.3 的槽口号的驱动器号。

3. SET PRODOS DATE

此功能是设置系统的日期。

4. SET PRODOS PREFIX

此功能是设置路径名。在任何文件被转换之前,必须设置正确的部首。你可通过路径名或驱动器号和槽口号来选择部首。

5. TRANSFER(OR LIST) FILES

此功能是转换或列印文件。经过前四项的设定之后,就可以进行转换或列印了。你可用左右箭头键来选择文件,按空格键即开始转换。你可用通配字符“?”来转换单个文件,或用“=”转换所有文件。

在转换的过程中,DOS 3.3 文件名可能包含有控制字符、空格和其它标点符号,它们不能作为 ProDOS 文件名,ProDOS 将自动地在无效字符的位置上设置一个句号。某些 DOS 3.3 磁盘有反相显示的文件名,ProDOS 将不能转换这些文件。

三、ProDOS COPY I PLUS

这是最新的多功能磁盘管理软件,它不仅管理 ProDOS 文件,而且也可以对 DOS 3.3 文件进行管理,并具有显示小写字母的功能。其版本有许多种,功能最多最强的版本是 ProDOS COPY I PLUS 9.0。其用法与 DOS 3.3 的 COPY I PLUS 最新版本相似,只不过将“SECTOR EDITOR”这一功能放在 BIT COPY 中了。用法这里不再赘述,请参考有关资料,如有需要者,请与本人联系。

四、APPLESOFT PROGRAMMER'S AID(APA)

这是用于 ProDOS 的程序设计辅助程序。它存放在 ProDOS EXAMPLES 磁盘上,它具有的功能是:自动产生行号、计算程序的长度、进行十进制与十六进制数的转换、删除 REM 语句、显示控制字符、列印变量、合并程序、重编行号、保护内存程序、中止自动行号、删除控制字符。APA 命令只能在立即执行方式下执行。

APA 命令集:

1. AUTO 和 MANUAL

格式:AUTO [开始行号],[增量]

MANUAL

功能:AUTO 命令是自动产生行号。行号范围为 0 至

63999。每当需要新行号时按空格即可。在输入程序时,如果想重新输入该行,只要按 CTRL-X,再按空格即可。MANUAL 命令是取消自动行号功能。

2. LENGTH

格式:LENGTH

功能:此命令是以字节数来显示内存程序的长度。长度以十进制和十六进制数表示。

3. CONVERT

格式:CONVERT [十进制或十六进制数]

功能:在执行该命令后跟一个十进制数后,将显示相对应的十六进制数,反之亦然。十六进制数用 \$ 表示。

4. COMPRESS

格式:COMPRESS

功能:此命令用于删除程序内的所有 REM 语句,以加快执行速度和节省内存。

5. SHOW 和 NOSHOW

格式:SHOW

NOSHOW

功能:SHOW 命令可将程序内所有隐含的控制字符显示出来。输入 SHOW 并列印程序,控制字符以反相方式显示。NOSHOW 命令是取消 SHOW 命令。

6. HOLD 和 MERGE

格式:HOLD

MERGE

功能:使用这两个命令可合并程序。先将第一个程序装入内存,用 HOLD 命令将程序放在 HIMEM 内,以防重写,再将第二个程序装入内存,用 MERGE 命令将第一个程序释放,这样两个程序已合并在一起。如果存在相同的行号,系统会把重复行号列出,供你选择。

7. RENUMBER

格式:RENUMBER [始行号],[增量],[首行号]
[末行号]

功能:此命令用于将整个程序或部分程序进行重编行号。

8. XREF

格式:XREF

功能:是用于将程序内所有的变量及所在的行号一起显示出来。此命令将分类显示每个变量,变量分为五种类型:

\$	串变量	%	整型变量
\$ (下标串变量	%(下标整型变量
(下标变量		

9. EXIT

格式:EXIT

功能:此命令是用于将 APA 从内存中删去。

四、SYSTEM UTILITIES

这是一个只适合于 APPLE IIe、IIc 和 IIGS 的系统实用程序,它具有很强的管理功能,但由于此软件使用的范围太窄,故在这里不予介绍,如有需要者,请与本人联系。

(未完待续)

高性能的 CEC-I 自定义组合键

无锡柴油机厂设计科 杨勇佳

CEC-I 缺乏自定义组合键功能,给操作者带来诸多不便。为了弥补这个不足,笔者设计出一种自定义组合键的方法,在几乎不影响原来键盘功能的情况下,一次最多可定义组合键 96 个,每个组合键可容纳 250 个字符,并能方便地更换组合键内容。

本方法首先修改了 DOS 的输入拦截指针(\$AA55 和 \$AA56 单元),当键盘上有键输入时,先转入程序一进行新扩展的键盘分析和外理,首先判断该键是否新设置的辅助键(Test 键),如不是,则把该键码交回原系统显示或操作。如确认是辅助键,则系统等待键盘上的再次键入,并根据这次的键值作“组合键”判读处理。如不是组合键就返回原系统处理,如是,则显示该组合键的内容,再返回原系统操作。

CEC-I 机系统决定了 F1~F5 键、Quit 键、光标键、SHIFT、Ctrl、Reset、ESC 诸键、中文键、西文键、Capslock 和 Return 这 18 个键不能用来定义组合键。除此以外,每次只按二次键即可获得组合键功能的键共有 48 个,而按三次键(其中一次为 SHIFT 或 Capslock 键)可获得功能的键亦可有 47 个键,再加上辅助键本身亦可被定义,故最多可定义 96 个组合键。

在 DOS 引导后,用 CALL-151 进入监控,无误地键入程序一,并用 BSAVE DYJ, A \$ 6000, L \$ 89 存盘,然后返回 BASIC,再进入中文状态,键入程序二。从 1000 行开始用 DATA 语句成对输入要定义的键和组合键的内容。在全部被定义键和定义内容输入后,再增加一个字符串“END”然后用 SAVE DYJ 1.0 存盘。

运行程序二,回答计算机提问的组合键定义文件简名。一般随用户的习惯输入一个短小的文件名“××”,然后由程序自动生成文件名的全名。接着显示该文件所包括的被定义键名和定义内容,并按扩展名把组合键文件记盘。这时计算机也具备了组合键功能。

如果用 CATALOG 检查磁盘,将会发现盘上除了先前存入的程序外,又新增了一个名为 DYJ-×× 的 B 型文件。

在本方法中,每一个被定义的键及其定义内容是按下列数据表格式放入内存的:

定义键码,定义内容长,定义内容

由于前二项各占一个字节,而各定义键的数据表是衔接的,因此内存利用率较高。

当您需设置不同的组合键定义时,只要修改程序二的 DATA 语句,然后运行程序,并重新起名。

每次开机后,只要 BRUN DYJ-×× 即可形成或更换组合键定义。也可以把最常用的定义文件用 PRINT CHR \$(4);“BRUN DYJ-××”加入 HELLO 程序中,以后只要一引导 DOS,随即可以得到一组常用的

组合键。

本方法的自定义组合键功能,可在中、西文 BASIC 状态下使用,也可在程序运行中使用。如果您想在监控和小汇编状态下使用,可输入 6000G 或 \$ 6000G。

当使用中进行了中、西文状态转换,或按过 Reset 键,组合键暂时失效,但只要有 & 或 6000G 命令输入,又重新激活组合键功能。

您可以根据使用需要灵活定义组合键内容。如定义“L”为 LIST,定义“C”为 CATALOG,定义“@”键为: 10 HOME;D\$=CHR\$(4);?D\$;“PR#3”;?:HGR2 等等。也可把一些常用的汉字词组定义成组合键,这等于扩充了一个最多 96 键的汉字词组输入。

作为对本系统功能的扩展,您可以在程序一中再增加一次对暂且称之为“代回车键”键值的判断和适当的程序转移,并根据需要在定义命令键时,把这个“代回车键”码放在命令字的最后。如设“代回车键”为字符“J”,定义“L”为:“LISTJ”,当以后使用组合键“L”时,即可进行 LIST 操作,而不必再按回车键。

如果有兴趣也可以进行不太大的改造,在定义功能键时,按 PC 机的 CCDOS 的汉字拼音码的拼音缩写规则进行定义,如定义“Y”为“ing”,则可以在 CEC-I 机上增加了仿 PC 机的汉字拼音输入功能。

最后指出一点:\$ 6087 和 \$ 6088 单元的值指示了定义码表的起存地址,因此在对程序一改造时必须把这两个单元的值移放在新的地址,并把这个地址在程序一的有关字节和程序二的某些变量反映出来,否则会严重影响本自定义组合键方法的实现。

程序一

```
6000- A9 0F 8D F6 03 A9 60 8D
6008- F7 03 A9 4C 8D F5 03 AD
6011- 55 AA 85 06 AD 56 AA 85
6018- 07 A9 27 8D 55 AA A9 60
6020- 8D 56 AA 60 6C 06 00 20
6028- 24 60 85 00 C9 96 D0 2F
6030- 20 24 60 85 00 AD 87 60
6038- 85 01 AD 88 60 85 02 A0
6040- 00 B1 01 F0 1A 85 03 C8
6048- B1 01 85 04 A5 03 C5 00
6050- F0 10 A5 04 20 7B 60 A9
6058- 02 20 7B 60 4C 3F 60 A5
6060- 00 60 C8 A5 04 F0 0E B1
6068- 01 9D 00 02 20 78 60 E8
6070- C6 04 4C 62 60 4C 0C FD
6078- 6C 53 AA 18 65 01 85 01
6080- A9 00 65 02 85 02 60 89
```


程序二

```

10 D$=CHR$(13)+CHR$(4);PRINTD$;"PR#
3";PRINT;HGR2;HOME
20 PRINT D$;"BLOAD DYJ"
25 PRINT"请输入组合键定义文件名:";INVERSE;
PRINT" ";NORMAL
30 VTAB 1;HTAB 24;INPUT NA$
35 NA$="DYJ-";NA$;PRINT"全名:";NA$
40 N=0;E=0;L=PEEK(24711);H=PEEK(24712);
BB=256 * H+L;RESTORE
50 READ N$;IF N$="END" THEN POKE BB+E,0;
PRINT;GOTO 140
55 PRINT"被定义的第";N+1;"键=";N$
60 PRINT"其内容是:";READ A$;PRINT A$
70 B=BB+E;LL=LEN(A$)
80 POKE B,ASC(N$)+128;POKE B+1,LL
90 FOR I=1 TO LL
100 POKE B+I+1,ASC(MID$(A$,I,1))+128
110 NEXT I
130 E=E+LL+2;N=N+1;X=FRE(0);GOTO 50
140 PRINT D$;"BSAVE";NA$;";A$ 6000,L";BB+
E+1-24576
150 CALL 24576;PRINT".....OK!";END
1000 DATA "1","LIST","2","RUN","3","CATALOG",
"END"

```

活泼的光标

吉林永吉县教师进修学校 殷瑞冬

一幅生动的屏幕画面将使人赏心悦目,倍感亲切和向往。这也是程序设计所追求的目标之一。能否一改APPLE I机呆板刺眼的方块光标使之具有一种活力,为屏幕画面增添生机?在这里介绍一段子程序,应用于程序中需要键盘输入的地方,将有一个形如往复似蹲下站起的问号光标,活泼可爱,你肯定会喜欢。

5000 REM 光标子程序

```

5002 FOR I=0 TO 1:PRINT W$(I)CHR$(8);:FOR J
=0 TO P:X=PEEK(-16384);IF X>127 THEN POKE -16368,
0:X$=CHR$(X-128);PRINT" "CHRS(8);:RETURN
5005 NEXT I:I=-(I=1);NEXT

```

在引入子程序时,需要在程序一开始的初始化部分加入下面的10语句,其中变量P是控制问号蹦跳的快慢,你可以随意调整:

```

10 DIM W$(1);W$(0)="?";W$(1)=CHR$(95);
P=14

```

光标子程序将把你按下的键赋值给X\$送回。每次"GOSUB 5000"相似于执行一次"GET X\$"。也正因

为如此,象

```
200 INPUT";A$
```

```
210 .....
```

的改写就需要多几条语句:

```
200 A$=""
```

```
205 GOSUB 5000;IF X$(<)CHR$(13) THEN
```

```
PRINT X$;A$=A$+X$;GOTO 205
```

```
210.....
```

CEC—I 机汉字 使用第十一行

北京一〇一中学高一 任义

在中华学习机汉字系统里,第十一行是作为系统提示用的,用户没法使用。我有一次在VTAB11定位打印时发现只在第十一行显示了一个汉字,其余的又回到前十行里显示去了。这给我一个启示,如果把汉字字符串分成若干个子字符串,一个子字符串存放一个汉字,然后分别用HTAB、VTAB逐个打印,这样就可以实现在第十一行显示汉字。笔者为此编了一段小程序,稍加修改就可加在用户的程序里作为子程序调用。

```
1000 A$="中华学习机汉字使用第十一行":A
```

```
=LEN(A$);H=5
```

```
1010 FOR I=1 TO A:B$=MID$(A$,I,1):B=
ASC(B$);IF B=127 THEN 1030
```

```
1020 HTAB H;VTAB 11;PRINT B$;:H=H+1;
NEXT;GOTO 1040
```

```
1030 C$=MID$(A$,I,3);HTAB H;VTAB 11;
PRINT C$;:H=H+2:I=I+2:NEXT
```

```
1040 END
```

如果想清除第十一行的汉字时,可以用下列方法:
CALL 62450(清除全屏幕)或PRINT CHR\$(18);
PRINT CHR\$(18)(只清除第十一行)。这里值得一提的是:打印一个子字符串后最好在程序中加一个分号,因为PRINT A\$后不加标点符号,机器便自动回车(屏幕上卷一行),有时会使我们在前十行中打印的字符上卷,所以大家在使用时要注意。还有一个应注意的问题是在第十一行显示汉字时,从第二十一列往后显示的汉字中,有时上半部分的字符阵乱七八糟的,所以在第十一行显示的汉字一不能太多,二最好靠前显示。

更正

本刊91年第7期“为无驱动器系统设置虚拟盘”一文,第15页程序代码中的BDC8单元内容应为“BD”,误排为“DD”,特此更正。

三页图形竖向并打

北京大学物理系一年级 陈钢

苹果机上高分辨率图形的分辨率只有 280×192 ，因此有时需要将一张较精细的图形分割制作，然后在打印时拼接起来。普通打印机在正常密度下只能实现两页图形并打，若在倍密度状态下打印又会引起比例失调。针对上述问题，我编了一段长度不到 120 个字节的机器语言小程序，将图形旋转 90° 打印，实现了三页并打，并有效地解决了多张图拼接问题。

将该程序输入后，把三页图形分别调至内存 $\$2000-\$3FFF$ ， $\$4000-\$5FFF$ ， $\$6000-\$7FFF$ ，然后打开打印机，在监控状态下键入 8000G 或在 BASIC 状态下 CALL32768 即可。需要注意的是，不必用诸如 PR #1 的命令接通打印机，这样就可以做到拼接打印，亦即打印完三幅图后再调入三幅图，再次运行程序就可以紧接着前三张图打印出来，中间不留空隙，因此本程序可以完成任意 $n \times 3$ 张图形的拼接打印。

此程序适用于象 PC-80, RX-80, PX-100 这些普通九针打印机，要求打印卡插在一号扩充槽上。

愿此程序帮助您打印出精美的图形！

```
8000- 20 6A 80 A9 41 20 6C 80
8008- A9 07 20 6C 80 A0 27 20
8010- 6A 80 A9 4B 20 6C 80 A9
8018- 41 20 6C 80 A9 02 20 6C
8020- 80 A9 20 85 08 A2 00 8A
8028- 0A 0A 29 1C 85 07 8A 6A
8030- 6A 6A 6A 29 03 05 07 05
8038- 08 85 07 8A 6A 29 E0 85
8040- 06 6A 6A 29 18 05 06 85
8048- 06 B1 06 29 7F 20 6C 80
```

电子打字机

长沙交通学院 4#616 王蔚

本文所介绍的程序，非常之短小，却非常实用。它可以使你的计算机变成一部电子打字机，把键入的字符显示在屏幕上。还有一个突出的优点，它的运行速度非常之快。读者要是手头有本打字训练的指导手册，很快便可以成为打字高手。程序清单如下：

```
300- 20 58 FC      JSR $FC58
303- 20 0C FD      JSR $FD0C
306- 20 ED FD      JSR $FDED
309- 4C 03 03      JMP $0303
```

读者可在监控或小汇编下键入程序。运行时可在监控状态下键入指令 300G 或 BASIC 下使用命令 CALL768，如要退出可按 Ctrl-Reset 键。

```
8050- E8 E0 C0 D0 D2 A5 08 18
8058- 69 20 10 C7 A9 00 20 6C
8060- 80 A9 0A 20 6C 80 88 10
8068- A6 60 A9 1B 2C C1 C1 30
8070- FB 8D 90 C0 60
```



程序的第一行调用监控的清屏幕子程序，清除屏幕，把光标移至左上角；第二行调用监控的读键子程序，把键入的键码存放在 6502 微处理器的累加器 A 中；第三行调用监控的显示子程序，把 A 中的键码转换成相应的字符显示在屏幕上；第四行无条件转移指令转回到第二行，继续读键。由于调用了监控子程序，因而程序十分短小。如果作以下的修改：

```
309- D0 F8 BNE      $0303
30B- F0 F6 BEQ      $0303
```

用条件转移指令模拟无条件转移指令的动作，则可以用监控的 M 命令把程序移至内存合适的地方。

不熟悉机器语言的读者，可以运行以下的 BASIC 程序：

```
10 FOR I=768 TO 779
20 READ A:POKE I,A:NEXT
50 DATA 32,88,252,32,12,253,32,237,253,
76,3,3
60 CALL 768
```

第八讲 指针与数组

李文兵

指针在 C 中占有重要位置。只有掌握了指针,才算把 C 学到手。这里,介绍指针结构及其各种用法。

1. 一维数组的指针

下面的说明语句定义一维数组 s 和指针变量 p。

```
char s[16], *p;
```

通过赋值语句:

```
p=s;
```

将在指针里赋上数组 s 的首地址,即指针 p 指向了数组 s。而且,数组元素 s[i] 可表为 *(p+i), 元素 s[5] 就可用 *(p+5) 表示。因此,数组元素的读写,完全可由指针实现,如练习 8.1 所示。

```
A>type exp8-1.c
```

```
#define MAX 10
```

```
main()
```

```
{ int array[MAX], *ptr, i;
```

```
ptr=array;
```

```
/* 指针 ptr 指向数组 array */
```

```
for(i=0; i<MAX; i++) array[i]=i;
```

```
/* 变化数组下标,给数组赋值 */
```

```
for(i=0; i<MAX; i++) printf("%3d", array[i]);
```

```
for(i=0; i<MAX; i++) *ptr++=i*i;
```

```
/* 用指针给数组赋值 */
```

```
printf("\n");
```

```
for(i=0; i<MAX; i++) printf("%3d", array[i]); }
```

```
A>exp8-1
```

```
0 1 2 3 4 5 6 7 8 9
```

```
0 1 4 9 16 25 36 49 64 81
```

2. 数组的动态分配

程序中设定的数组,到程序被编译时,其空间便被确定。但有时希望需要空间时再从称为堆的自由空间分配。为此,要使用存储管理库函数;诸如:

```
malloc(nbyte)
```

```
free(s)
```

来确定。这些函数能自由地占有或释放堆的存储空间, malloc() 的调用形式如下:

```
int *p;
```

```
unsigned int nbyte;
```

```
p=malloc(nbyte);
```

结果把分配的存储空间的首址赋给指针 p。练习 8.2 是确定能存 1000 个整型数据存储空间程序。

```
a>type exp8-2.c
```

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

```
#include "stdlib.h"
```

```
#define MAX 1000
```

```
main()
```

```
{ int *p, *s, i;
s=p=malloc(MAX * sizeof(int));
for (i=0; i<MAX; i++)
* p++=1000+i;
for(i=0; i<10; i++)
printf("%x", * (s+i));
free(s);
}
```

```
A>exp8-2
```

```
3e8 3e9 3ea 3eb 3ec 3ed 3ee 3ef 3f0 3f1
```

执行该程序,函数 malloc() 便确定了含有 1000 个整型元素的数组。当该数组不用时,该程序通过函数 free(), 把所确定的存储空间返回给系统。

这样动态地分配堆空间,会更为有效地利用存储空间,即该空间可多次用于不同目的,而且其数组的大小,可根据系统来设定,提高了程序的可移植性。这些函数,不仅可以动态分配数组,而且对于队列、链表、二叉树等,在动态分配存储空间时,也是必须的。

3. 指针和字符串

在字符串的操作中,使用指针是很方便的。关于字符串的长度计算、拷贝、连接等操作,如练习 8.3 所示。

```
A>type exp8-3.c
```

```
#define MAX 10
```

```
main()
```

```
{ char s[MAX], t[MAX], *p, *pp;
```

```
int len=0;
```

```
gets(s);
```

```
p=s;
```

```
/* 字符串长度 */
```

```
while(* p++ len++;
```

```
printf("length of string=%d\n", len);
```

```
/* 字符串拷贝 */
```

```
p=s; pp=t;
```

```
while(* pp++=* p++);
```

```
printf("string of t=%s\n", t);
```

```
/* 字符串连接 */
```

```
p=s; pp=t;
```

```
while(* pp++);
```

```
* pp--;
```

```
while(* pp++=* p++);
```

```
printf("string of t=%s\n", t);
```

```
}
```

```
A>exp8-3
```

```
qwer
```

```
length of string=4
```

```
string of t=qwer
```

```
string of t=qwerqwer
```


程序中,对字符型数组 s 输入字符串,是使用函数 gets()进行的;若使用指针,可按如下进行。

```
char *s;
s="string";
```

也可把这两个语句合并为:

```
char *s="string";
```

4. 指针的运算性质

适于指针的运算有:

(1) 指针加减一个整数 在 C 语言中,允许指针加减一个整数,其意义是当指针指向数组时,使指针相对原位置移动,以使其指向另一个数组元素。例如:

```
int a[10], *p;
p=a;
```

是使指针 p 指向了数组 a,即指向数组 a 的第一个(0 号)元素。这时,若执行:

```
p+=i;
```

便使 p 指向数组 a 的 i 号元素。如练习 8.4 所示。

```
A>type exp8-4.c
#define W 10
main()
{ int a[W]={0,1,2,3,4,5,6,7,8,9}, *p;
p=a;
printf("%d,%d,%d\n", *p+3, *(p+3), *++p);
printf("%d\n", *p++);
printf("%d,%d,%d\n", *p, *(p+3), *(p-2));
}
A>exp8-4
4,4,1
1
2,5,0
```

通过这个练习,应该对指针操作有如下了解。

① * (p+3) 与 * p+3 不同: * (p+3) 是取指针当前位置后面的第 3 个元素的值;而指针位置并无改变; * p+3 是先取 p 所指元素的值,再在该值上加 3。

② * p++ 与 * (p+1) 不同: * p++ 是先取 p 所指元素的值,再把指针向后调整一个元素,且指针便固定在此位置上;而 * (p+1) 则是取指针当前位置后面一个元素的值,指针位置不改变。

③ * ++p 的结合性是 * (++p)。因此,是先把指针后调整一个元素,再取所指元素的值。

指针的这一运算性质,被广泛地用于堆栈的操作上。所谓堆栈是按照“先进后出”进行存取操作的存储空间。程序中 push() 函数的功能是把一个数据存入到指针所指定的堆栈空间内,且把指针向高地址移动一个位置。pop() 函数则把指针位置向低地址移动一个位置,取出当前指针所指定的值。在堆栈操作中,使用指针变量是极为方便的,如练习 8.5 所示。

```
A>type exp8-5.c
#include <stdio.h>
#define MAX 10
int *p; /* 指向自由区 */
int *t; /* 指向栈顶 */
```

```
int *b; /* 指向栈底 */
void push();
main()
{ int x,y,i,var;
char s[80];
p=(int *)malloc(MAX*sizeof(int)); /* 建栈 */
if(!p)
{ printf("建栈失败");
exit(1);
}
t=p;b=p+MAX-1
printf("输入 10 个数:");
for(i=0;i<10;i++)
{ scanf("%d",&var);
push(var);
}
printf("弹出 5 个数:");
for(i=0;i<5;i++)
{ x=pop();
printf("%d ",x);
}
printf("\n");
}
void push(i) /* 压栈 */
int i;
{ if(p>b)
{ printf("栈满");
return;
}
*p=i;
p++;
}
pop() /* 退栈 */
{ p--;
if(p<t)
{ printf("栈空\n");
return 0;
}
return *p;
}
A>exp8-5
输入 10 个数:1 2 3 4 5 6 7 8 9 1
弹出 5 个数:1 9 8 7 6
```

对于不同类型的指针,同样移动一个元素的位置,实际移动的距离(字节数)是完全不同的,如下所示。

指针的数据类型	移动一个元素的实际距离
char	1
int	2
float	4
double	8

为了使指针移动时,能恰好指定相应元素,C 编译系统对指针所加减的整数乘上一个比例因子。这比例因子正是各数据类型所占字节数,例见练习 8.6。

```
A>type exp8-6.c
```

```

main()
{ char c[4] = {"1234"}, *pc;
  int i[4] = { 1,2,3,4 }, *pi;
  float f[4] = { 1,2,3,4 }, *pf;
  double d[4] = { 1,2,3,4 }, *pd;
  int n;
  pc=c;pi=i;pf=f;pd=d;
  for(n=0;n<4;n++)
    printf("c[%p]=%c\n",pc,*pc++);
    printf("\n");
  for(n=0;n<4;n++)
    printf("i[%p]=%i\n",pi,*pi++);
    printf("\n");
  for(n=0;n<4;n++)
    printf("f[%p]=%f\n",pf,*pf++);
    printf("\n");
  for(n=0;n<4;n++)
    printf("d[%p]=%f\n",pd,*pd++);
}

```

```

A>exp8-6
c[FF9F]=1
c[FFA0]=2
c[FFA1]=3
c[FFA2]=4

```

```

i[FFA4]=1
i[FFA6]=2
i[FFA8]=3
i[FFAA]=4

```

```

f[FFB0]=1.000000
f[FFB4]=2.000000
f[FFB8]=3.000000
f[FFBC]=4.000000

```

```

d[FFC6]=1.000000
d[FFCE]=2.000000
d[FFD6]=3.000000
d[FFDE]=4.000000

```

(2) 指针赋地址 这类运算有:

p=&var; 把变量 var 的地址赋给指针 p;
 p=&a[i]; 把数组元素 array(i) 的地址赋给指针 p;
 p=a; 把数组的首地址赋给指针 p;
 p=px; 把同类指针 px 的值赋给指针 p。

关于这类运算, 请看练习 8.7。

```

A>type exp8-7.c
main()
{ int x=15,*p,*q,*r;
  int a[10]={1,2,3,4,5,6,7,8,9,10};
  q=a;p=q;
  printf("%p,%p,%p\n",a,q,p);
}

```

```

p=&a[3];q=&a[5];r=&x;
printf("%p,%p,%p\n",p,q,r);
printf("%4d,%4d,%4d\n",*p,*q,*r);
}

```

A>exp8-7

```

FFCA,      FFCA,      FFCA
FFD0,      FFD4,      FFC8
4,         6,         15

```

(3) 两个指针比较运算 当两个指针指向同一数组时, 可用关系运算符对它们进行比较。实际上, 就是比较两个指针在该数组的相对位置。设 px 和 py 指向同一数组, 那么,

px<py; 当 px 所指位置在 py 的之前时为真;
 px>py; 当 px 所指位置在 py 的之后时为真;
 px<=py; 当 px 所指位置在 py 的之前或为同一位置时为真;
 px>=py; 当 px 所指位置在 py 的之后或为同一位置时为真;
 px==py; 当 px 所指位置与 py 的相同时为真;
 px!=py; 当 px 所指位置与 py 的不相同时为真。

[练习 8.8]

A>type exp8-8.c

```
#define TRUE 1
```

```
#define FALSE 0
```

```
main()
```

```
{ if(isitpal("FOUR SCORE IN SEVEN YEARS")) == TRUE)
```

```
printf("string 1 is a pal indrome\n");
```

```
if(isitpal("ABLE WAS I ERE I SAW ELBA")) == TRUE)
```

```
printf("string 2 is a pal indrome\n");
```

```
}
```

```
isitpal(str)
```

```
char *str;
```

```
{char *str1=str;
```

```
while(*str1 != '\0') ++str1;
```

```
---str1;
```

```
while((str<str1)
```

```
if(*str++ != *str--)
```

```
return(FALSE);
```

```
return(TRUE);
```

```
}
```

A>exp8-8

```
string 2 is a pal indrome
```

(4) 两个指针相减 当两个指针指向同一数组时, 其差值是两指针相对移动的元素个数。利用这一运算性质, 可以计算字符串的长度, 如练习 8.9 所示。

A>type exp8-9.c

```
main()
```

```
{char *s="abcdef ghijkl. \n";
```

```
char *p=s;
```

```
while(*p)
```

```
p++;
```

```
printf("length of string s=%d\n",p-s);
```

```

}
A>exp8-9
length of string s=15

```

5. 指针与多维数组

多维数组可用指针调用,这是因为多维数组在存储空间是按行存储的,用一般指针变量调用多维数组的程序,如练习 8.10 所示

```

A>type exp8-10.c
#include "time.h"
#include <stdio.h>
#include "stdlib.h"
#define ROW 4
#define col 4
main()
{int i,j,s[ROW][col];
    long n;
    srand(time(&n));
    for(i=0;i<ROW;i++)
    for(j=0;j<col;j++)
    s[i][j]=rand()/100;
    print_mat(" mat s[][]=",s);
    trans(s);
    print_mat(" trans mat=",s);
}
trans(a)
int a;
{ int i,j;
for(i=0;i<ROW;i++)
for(j=i+1;j<col;j++)
swap(a+i*ROW+j,a+j*ROW+i);
}
swap(a,b)
int *a,*b;
{int t;
t=*a;*a=*b;*b=t;
}
print_mat(s,mat)
char *s;
int mat[][col];
{int i,j;
printf(" ---%s---\n",s);
for(i=0;i<ROW;i++)
{for(j=0;j<col;j++)
printf("%4d ",mat[i][j]);
printf("\n");
}
}

```

```

A>exp8-10
---mat s[][]=---
199    177    263    84
242    119    115    262

```

```

239    165    44    288
73     68    29    60
---trans mat=---
1991    29682    -20224    0
21510    256    30465    0
239    165    44    288
73     68    29    60

```

6. 指针数组

用户还可以定义指针数组,例如:

```
float *x[100];
```

就定义了一个指向浮点数的指针数组。可按如下所示,对该数组赋初值。

```

x[0]=a;
x[1]=b;
:

```

为了较清楚地理解指针数组的概念,这里,有必要再多说几句。假如,我们定义了一个指针数组:

```
float *x[2];
```

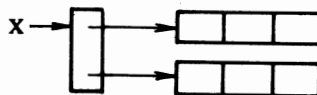
注意,变量 x 不是二维数组;而是由两个指向一维 float 数组的指针组成的数组。但对它适当地初始化,可用作二维数组。要想使它能表示 2×3 的 float 数组,只要按如下初始化即可

```

x[0]=(float *)malloc(3*sizeof(float));
x[1]=(float *)malloc(3*sizeof(float));

```

现在 $x[0]$ 和 $x[1]$ 都指向能保存三个 float 数据的存储空间,其图示如下:



其实例如练习 8.11 所示。

```

A>type exp8-11.c
char *err[]={ "EZER0 0 /* ERROR 0 */",
               "EIVLFNC 1/* invalid function number */",
               "ENOFILE 2/* File not found */",
               "ENOPATH 3/* Path not found */",};
main()
{ int i;
  for(i=0;i<4;i++)
    printf("errno[%d]=%s\n",i,err[i]);
}
A>exp8-11
errno[0]=EZER0 0 /* ERROR 0 */
errno[1]=EIVLFNC 1 /* invalid function number */
errno[2]=ENOFILE 2 /* File not found */
errno[3]=ENOPATH 3 /* path not found */

```

引用命令行参数是使用指针数组的典型例子。即如下所示的命令:

```
A>程序名 参数1 参数2 .....
```

为引用命令行参数,主函数 $\text{main}()$ 应按如下设置:

```

main(argc,argv)
int argc;

```

```
char *argv[];
{ ... }
```

其中, `argc` 的值等于参数总数+1; `argv` 是指针数组, 其内容如下:

`argv[0]` 的内容为程序名的地址
`argv[1]` 的内容为参数 1 的地址
`argv[2]` 的内容为参数 2 的地址
 作为例子, 请看练习 8.12。

```
A>type exp8-12.c
main(argc,argv)
int argc;
char *argv[];
{ while(argc-->0)
  printf("argv[%d]=%s\n",argc,argv[argc]);
}
```

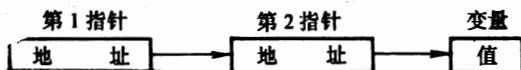
```
A>exp8-12 arg1 arg2 arg3 arg4
argv[4]=arg4
argv[3]=arg3
argv[2]=arg2
argv[1]=arg1
argv[0]=A:\EXP8-12.EXE
```

7. 指向指针的指针

应该指出的是, 指针数组是指向指针的指针, 即形成一个指针链。前面我们所介绍的指针其值是某变量的地址, 如下所示:



而在指向指针的指针里存放的是第二个指针的地址, 其实际值是由第二指针指出的, 即



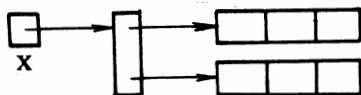
在 C 语言里, 用 `**x`, 表示 `x` 为指向指针的指针, 其作用相当于 `*x[]`。例如, 在程序中, 我们定义了:

```
float **x;
```

则意味着 `*x` 的类型是指向 `float` 的指针, 而 `**x` 的类型是 `float`, 即用 `**x`, 可引用值。这样的结构按如下初始化后, 也可作二维数组。

```
x=(float **)malloc(2*sizeof(float *));
x[0]=(float *)malloc(3*sizeof(float));
x[1]=(float *)malloc(3*sizeof(float));
```

其图示如下:



关于指向指针的指针的用法, 请看练习 8.13。

```
A>type exp8-13.c
main(argc,argv)
int argc;
char *argv[];
{ printf("argc=%d\n",argc);
  while(*argv++>0,argc-->0)
    printf("argv %0x %s\n",argv,*argv);
}
A>exp8-13 aa bb cc
argc=4
argv ffdc aa
argv ffde bb
argv ffe0 cc
argv ffe2 (null)
```

8. 数组指针

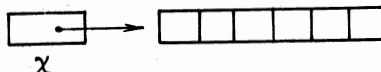
下面的说明语句便定义了一个指向三个 `float` 数组组成的数组指针。

```
float(*x)[3];
```

它的值最初是不确定的, 必须使用 `malloc()` 函数来确定。按如下所示初始化, 便可得到 `2*3` 数组。

```
x=(float(*)[3])malloc(2*3*sizeof(float));
```

这样, `x` 所指向的存储空间, 不是保存一个三元素 `float` 数组, 而是连续存放这样的两个数组。 `x` 可图示为:



练习 8.14 是求某年某月某日是这一年第几天的程序, 该程序就使用了数组指针。

```
A>type exp8-14.c
int dtab[][13]={
  {0,31,28,31,30,31,30,31,31,30,
   31,30,31},
  {0,31,29,31,30,31,30,31,31,30,
   31,30,31},
};
```

```
main()
{ int y,m,d,yd;
  scanf("%d,%d,%d",&y,&m,&d);
  yd=dofy(dtab,y,m,d);
  printf("%d:%d:%d=%d\n",y,m,d,yd);
}
```

```
dofy(x,ty,tm,td)
int (*x)[13],ty,tm,td;
{ int i,j;
  j=ty%4==0&&ty%100!=0||ty%400==0;
  for(i=1;i<tm;i++)
    td+=*(*x+i);
  return(td);
}
```

```
A>exp8-14
1991,2,11
1991:2:11=42
```

我们把本节所介绍的几种指针结构, 就二维数组的情况, 小结一下, 如表 8.2 所示, 供参考。

表 8.2 指针结构用于二维数组小结表

指针结构	名称	行、列情况	左值	效率
float xa[2][3]	二维数组	静态数组(固定行列)	xa, xa[0]均不是左值	高
float * xb[2]	指针数组	固定行、动态列	xb 不是左值, xb[0]是左值	低, 节省空间
float (* xc)[3]	数组指针	固定列、动态行	xc 是左值, xc[0]不是左值	低, 节省空间
float ** xd	指针的指针	行、列皆为动态	xd, xd[0]皆为左值	低, 节省空间

邮购广告:

▲中外合资温州福灵达仪器仪表有限公司邮购部, 长期优惠价供应进口组装 $3\frac{1}{2}$ 位数安万用表。批价从优。

型号	主要特点	邮购价(含运费)
53188	袖珍, 可放置上衣口袋。多功能、全保护, 可测于电池 9V、1.5V 好坏。最大电阻量限 200M。	170.00
M3800	显示器带放大镜、全保护功能, 误测不致烧坏仪表。微电流测量分辨率为 0.01mA	180.00
DT8908	显示器带放大镜, 量程全保护 5 档电容测量。不需调零	260.00
DT940C	显示器带放大镜, 量程全保护; 测电容 5 档, 不需调零。内置温度传感器, 可直测室温。	390.00
5115	全量程保护, 配有专用稳压管, 改善电池降压物性和温度特性。	190.00
PF72	台式、便携两用。3 $\frac{1}{2}$ 位绿色显示, 适用车间、实验室、车、船、野外及光线暗弱环境或远距离观测	620.00

注: 备有详细说明, 函索即寄, 款到一周发货。

电话: 225222 直拨区号: 0577 电挂: 2752

帐号: 中行温州分行营业部 12825069—3

地址: 浙江温州黄府巷 40 号

邮编: 325000

联系人: 王文瑞

(上接第 40 页)

因而地址 LED 全亮。用万用表的一根表笔取下当短路线, 一端接地, 另一端分别触碰 6502 的 9~18 脚, 则相应 LED 灭。如不受控制, 则说明此线路有故障, 故障原因可能有: U_{13} 、 U_{14} 三态门的控制端开路, 连接线开路或短路等。

此外, 6502 的 2、4、6、40 脚如有断路, 则 6502 也不执行程序。

五、接口电路故障的检修

在前面检查显示器时已检查了输入接口电路 U_{20} , 下面只介绍输出接口电路 U_{21} (74LS374) 的检修方法。

用键盘发出地址 \$201 (或 \$203、\$205……等奇数地址), 选中 U_{21} , 将 8 位数据开关全置 1, 按住 \overline{WR} 键, 则 U_{21} 的 8 个数据输出端应全为 1。然后将数据为全 0, 数据输出端也应为全 0。如有异常, 可检查相应接线及 U_{21} 各引脚的连接情况, 即可排除故障。



多功能电子钟

(KDC-Ⅲ 最小系统应用实例)

张培仁 刘振安

以单片机最小系统为核心,增加显示器和有关功能键,即可组成多功能电子钟。

电子钟用定时器 0 产生了定时中断,控制电子钟走时,电子钟的初值(时、分、秒)由键盘输入,而后运行。增减定时常数值,即可细调电子钟走时精度。

定时器 0 的定时溢出脉冲周期 $t = t_c \times T_0 \times 10^{-6}$ 秒。其中 t_c 当单片机主振频率是 6MHz 时,等于 $2\mu s$ 。 t 为每次溢出中断的间隔时间,我们取 0.1s,也就是每 0.1s 中断服务一次。这样求出 T_0 值 = 50000 即等于 16 进制数是 3CBOH。因为每次中断都执行几条中断服务程序,所以 T_0 的修正值为 3CBFH。这样确定了 T_0 的时间常数值为: $TH_0 = 3CH$
 $TL_0 = BFH$ 。

显示器和键盘和 KDC-Ⅲ 开发机的显示和键盘完全一样,地址口也一样,只是用 2 个或 4 个键就可以。键的用法和电子表对时使用方法相似。键分析和处理程序用了个智能仪器中常用的多次复用算法,这里不再讲了。图一是电子钟程序框图。

电子钟程序清单如下:

```

MOV     DPTR, # 4101H
MOVX    @DPTR, A    送字形
MOV     A, R2
INC     DPTR
INC     DPTR
MOVX    @DPTR, A    开显示
DSUP2: DJNZ   R3, DSUP2
INC     RO          指向下一缓冲区
CLR     C
MOV     A, R2
RRC     A          指向下一位显示
MOV     R2, A
JNZ     DSUP1      未完继续
MOV     R0, # DS6  向显示缓冲区放数
MOV     A, SBF
ACALL   PTDS
MOV     A, MBF
ACALL   PTDS
MOV     A, HBF
ACALL   PTDS
POP     DPL
POP     DPH
SJMP    DSUP
STOP:   CLR     TR0    停止定时器
        LJMP    2000H  关中断,返回
GTDS1:  MOV     A, @R0  从显示缓冲区取数字程序

```

```

CJNE    A, # 0AH, GTDS2
JNC     STOP
GTDS2:  ANL     A, # 0FH
        MOV     R1, A
        DEC     R0
        MOV     A, @R0
        SWAP    A
        ANL     A, # 0F0H
        ORL     A, R1
        RET
PTDS:   MOV     R1, A  向显示缓冲区放数字程序
        ACALL   PTDS1
        MOV     A, R1
        SWAP    A
PTDS1:  ANL     A, # 0FH
        MOV     @R0, A
        DEC     R0
        RET
CLOCK:  MOV     TL0, 0B7H  中断服务程序
        MOV     TH0, # 3CH
        ORG     2000H
        BUF     EQU 23H1/10 秒单元缓冲区
        SBF     EQU 22H 秒单元缓冲区
        MBF     EQU 21H 分单元缓冲区
        HBF     EQU 20H 时单元缓冲区
TIMER:  SJMP    GTDS
        AJMP    CLOCK  转中断服务程序
        ORG     2030H
GTDS:   MOV     R0, # DS2
        ACALL   GTDS1  向显示缓冲区取数
        CJNE    A, # 24H, CHK  测试时间初值合法性
        JNC     STOP
CHK:    MOV     R0, # DS4
        ACALL   GTDS1
        CJNE    A, # 60H, CHK1
        JNC     STOP
CHK1:   MOV     R0, # DS6
        ACALL   GTDS1
        CJNE    A, # 60H, CHK2
        JNC     STOP
CHK2:   ANL     TMOD, # 0F0H  设定定时器工作方式 1
        ORL     TMOD, # 01H
        ORL     IE, # 82H  允许中断
        ORL     TL0, # 0B0H  送时间常数初值
        MOV     TH0, # 30H
        MOV     BUF, # 00H1/10 秒单元清 0

```

```

SETB      TR0 开定时器
DSUP: MOV   R0, #DS1 显示程序
      MOV   R2, #20H
      MOV   R3, #00H
      MOV   DPTR, #4100H 指向字形表
      MOV   A, #0DH
      MOVX  @DPTR, A
      MOV   DPTR, #SGTB
      PUSH  DPH
      PUSH  DPL
DSUP1: MOV   A, #0FFH
      MOV   DPTR, #4101H
      MOVX  @DPTR, A 关显示
      MOV   A, @R0 取字形
      POP   DPL
      POP   DPH
      MOVC  A, @A+DPTR
      PUSH  DPH
      PUSH  DPL
      PUSH  PSW
      PUSH  ACC
      SETB  RS0
      INC   BUF
      MOV   A, BUF
      CJNE  A, #0AH, DONE
      MOV   BUF, #00H
      MOV   A, SBF
      INC   A 秒增 1
      INC   A
      DA    A
      MOV   SBF, A
      CJNE  A, #60H, DONE
      MOV   SBF, #00H
      MOV   A, MBF
      INC   A 分增 1
      DA    A
      MOV   MBF, A
      CJNE  A, #60H, DONE
      MOV   MBF, #00H
      MOV   A, HBF
      INC   A 时增 1
      DA    A
      MOV   HBF, A
      CJNE  A, #24H, DONE
      MOV   HBF, #00H
DONE: POP   ACC 恢复
      POP   PSW
      RETI  中断返回
DS1:  DATA  5AH
DS2:  DATA  5BH
DS3:  DATA  5CH
DS4:  DATA  5DH
DS5:  DATA  5EH
DS6:  DATA  5FH
SGTB: DB   0C0H, 0F9H, 0A4H, 0B0H, 099H,
        092H, 082H, 0F8H 字形代表码

```

```

DB      080H, 090H, 088H, 083H, 0C6H,
        0A1H, 086H, 08CH, 0FFH
END

```

我们在这个电子钟基础上又增加了一些功能。增加了三十二年夏时制自动转换, 闰年判断和处理, 三十二年农历的年、月、日的计算和显示。

三十二年夏时制自动转换程序:

我国夏时制是每年的四月中旬的第一个星期天开始至 9 月中旬的第一个星期天结束。一周为七天, 故星期天只能在四月或九月的 10 日—17 日之间。所以程序中只判断四月或九月的 10 日—17 日。再查夏时制转换日期表。夏时制的实现过程在相应的日期将时钟拨快或拨慢 1 小时。程序框图如图 2:

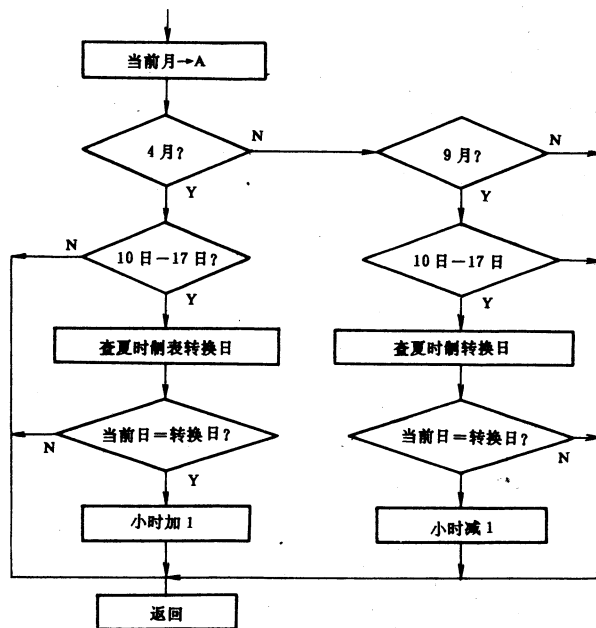


图 2 夏时制处理程序框图

闰年判断程序中, 用寄存器 B 来确定是否为闰年。闰年的判断是根据建立的闰年来查询的。闰年表是根据天文常识编制的。一般来讲, 公历年数能够被 4 整除时为闰年, 否则为平年。但有特例除外, 能被 100 整除, 而不能被 400 整除的不是闰年。在 1989 年至 2021 年无此特例。当闰年时 B 置 1, 否则 B 为 0。

农历时转换程序:

农历是具有我国民族特色的历法体制。它是根据月亮的运转周期确定的。由于它的天文计算过程很繁琐, 并且农历日期还有个别特例地方很难处理。所以我们依旧使用建表的方法来实现日历变换。但表又不能太长, 不可能把 32 年每天都建一个农历表。如每天建表需占用 12K ROM。因此我们采用建表和计算相结合的方法。即要转换农历时, 先查本月公历 2 日的农历日期, 然后进行数据分离和计算, 再得相应农历年、月、日。农历日表共占 $12 \times 32 \times 1$ 字节 = 384 字节。这样大大节省了 ROM 的容量。

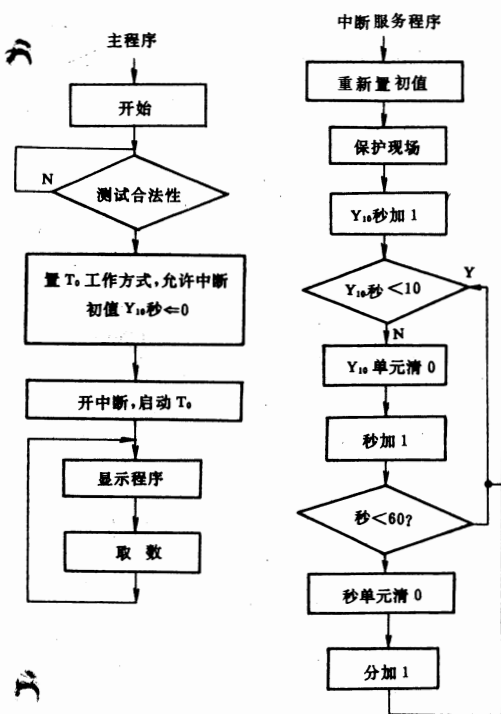


图1 电子钟主程序和中断服务程序框图

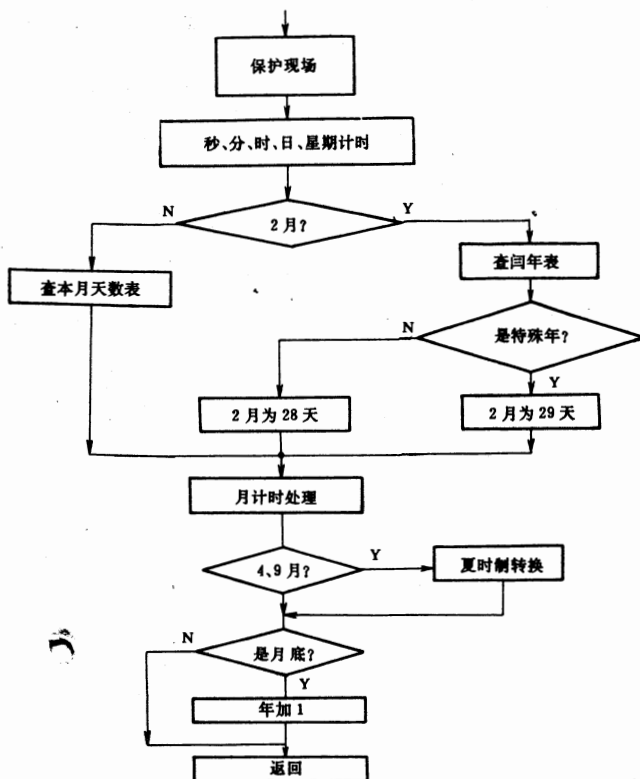


图3 闰年处理程序框图

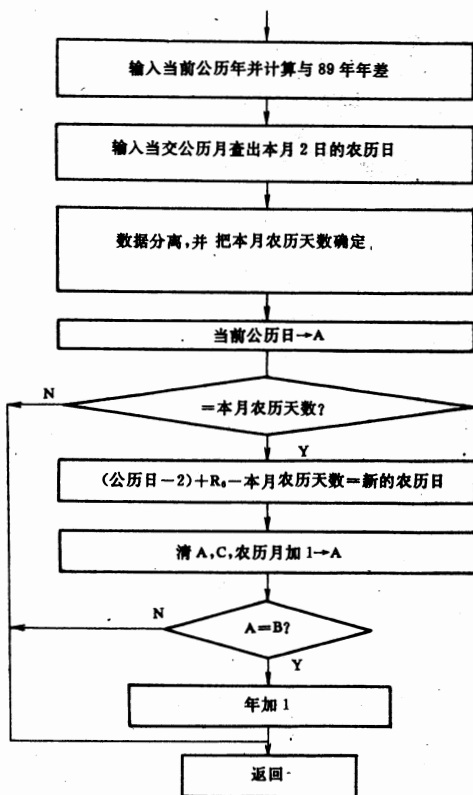


图4 农历转换流程框图

农历日计算方法如下：

(公历日-2)+公历2日的农历日期。若此数大于当月农历总天数,则减掉总天数,剩余数为农历日。同时农历月,年进行相应的变化。若当前公历日为1日时,则2日的农历日减1。但在这三十二年中,只有一特例。只有2000年6月1日特殊。2000年的6月1日农历日则应为4月30日(按以上算法计算出的),但2000年农历4月为小月,只有29天,所以公历6月1日的农历日实际应为4月29日。因此要对2000年6月1日这一天进行特殊处理。

2000年部分日期农历、公历对应表

农历	公历
4月26日	5月29日
4月27日	5月30日
4月28日	5月31日
4月29日	6月1日
5月1日	6月2日

农历转换程序对农历日表中数据进行数据分离。农历日表按顺序列了自1989年1月至2020年12月每月2日的农历日期。每一数据一个字节(8位),数据位定义如下:

(1)数据低五位是该农历日期的16进制数。

(2)数据的第六位表示闰月信息。置1时为农历月为闰月,0为平月。(下转第35页)



简易精确的 APPLE II 时钟卡

杭州大学 姜仕仁 施青松

APPLE II 和中华学习机被广泛应用于过程控制和自动采样系统,但是当它们被用于与时间有关的控制场合和定时采样过程中时,用户往往为该机没有时间输出指令(如许多计算机都可通过 PRINT TIME 指令得到时间值)而烦恼。如笔者在做小型动物活动频率自动记录实验中,需要记录动物每次出入洞口的时间,在此,由于程序跳转等原因,很难用软件来准确计时。于是,我们利用电子钟机芯制作了一块计时精确的时钟卡。

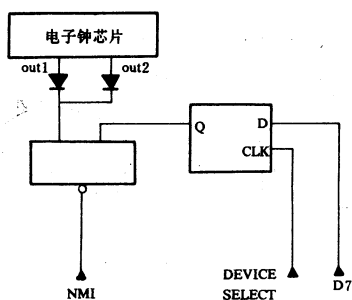


图 1

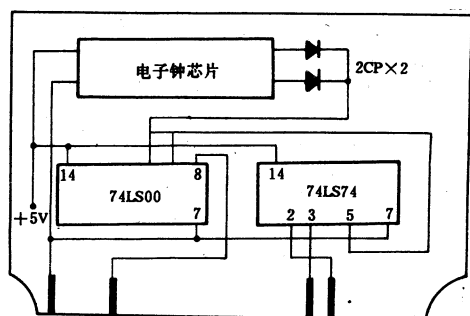


图 2

该卡是利用电子钟产生的秒脉冲。APPLE II 扩展槽口上的第 29 脚为非屏蔽中断请求接脚(NMI),当该脚出现高电平向低电平的负跳变时能被 6502 所识别,6502 在执行完正在执行的本条指令之后就暂停主程序而进入 NMI 中断响应周期。故只要在 \$FFFA 和 \$FFFB 两单元指向的地址(\$3FB)开始的三个单元中置一跳转指令,将非屏蔽中断响应引导到一个计时子程序。这样,每当出现一个负脉冲,中断响应计时子程序便加 1,并且用户程序和计时子程序分时执行,互

不影响。当用户程序中需要当前时间,可方便地从有关地址中读取。

本时钟卡的线路图如图 1 所示,图 2 为实际接线图。

电子钟机芯上有两个 0.5Hz 的脉冲输出端 out₁ 和 out₂,两者交替推动秒针每秒走动一次。在 out₁ 和 out₂ 两输出端分别接上一个二极管,两个二极管构成电路,在输出端产生 1Hz 的脉冲送入与或门电路中的一个输入端,另一个输入端受 D 触发器控制。当 D 触发器输出为 0 时,不管电子钟机芯的脉冲是 0 还是 1,与或门的输出不变。而当 D 触发器输出为 1 时,与或门输出为与输入反相的脉冲,从而引起非屏蔽中断响应。D 触发器受设备选择线 DEVICE SELECT 和 D₇ 控制,当 D₇ 为 0 时输出为 0, D₇ 为 1 时输出为 1。

按图 2 的接线联结各元件,将它们安装在一块实验线路板上,检查无误后插入 1~7 任一槽口中,然后给 \$3FB~\$3FC 三个地址中置入跳转指令,可在监控状态下输入:

3FB:4C 00 03

然后再以 \$300 开始输入 CLOCK.SUB 中断响应计时子程序,最后通过 POKE AD,X 打开时钟卡,(AD = 49280 + SLOT × 16; X ≥ 128)。由于 CLOCK.SUB 子程序是自动显示的,时钟卡一打开,先是显示出分别存放时、分、秒的 \$7、\$8、\$9 三个地址中的随机数,用户可通过 POKE 指令来设置这三个地址的当前时间值,以后便开始自动记时。用户在程序的任何地方的任何时刻都可向 \$7、\$8、\$9 三个单元得到精确的时间。程序 1 是用该时钟卡记录老鼠每次出入洞口时间的实例(洞口安装一光电传感器)。同样有了这块时钟卡,可方便地应用于任何定时采样及过程控制。

当数据或程序存储之前,必须关掉时钟卡,以免损坏盘中的程序。关时钟卡的指令与开时相仿,但 X 值 < 128 即可。

程序 1

```
5 POKE 1019,76:PLKE 1020,0:POKE 1921,3
10 PRINT CHR$(4);"BLOAD CLOCK.SUB"
20 INPUT "PRESENT TIME: (Hour, Minute, Second)";
   HO,MI,SE;HOME
25 IF HO>23 OR MI>59 OR SE>59 THEN 20
30 POKE 7,HO:POKE 8,MI:POKE 9,SE
35 POKE 49344,255
40 ST = 4096
50 A=PDL(1)
60 IF A = 255 THEN 80
70 IF PEEK(7) = HO-(MI=0) AND PEEK(8) = (MI-1)
```

```

+60 * (MI=0) THEN 100
75 N=0;GOTO 50
80 N=N+1;IF N=1 THEN POKE ST,PEEK(7);POKE
    ST+1,PEEK(8);POKE ST +2,PEEK(9)
85 N=2;ST=ST+3
90 GOTO 50
100 END

```

CLOCK.SUB 子程序

```

0300- 48      PHA
0301- 8A      TXA
0302- 48      PHA
0303- 98      TYA
0304- 48      PHA
0305- E6 09   INC  $ 09
0307- A5 09   LDA  $ 09
0309- C9 3C   CMP  # $ 3C
030B- 30 1C   BMI  $ 0329
030D- A9 00   LDA  # $ 00
030F- 85 09   STA  $ 09
0311- E6 08   INC  $ 08
0313- A5 08   LDA  $ 08
0315- C9 3C   CMP  # $ 3C
0317- D0 10   BNE  $ 0329
0319- A9 00   LDA  # $ 00
031B- 85 08   STA  $ 08
031D- E6 07   INC  $ 07
031F- A5 07   LDA  $ 07
0321- C9 18   CMP  # $ 18
0323- D0 04   BNE  $ 0329
0325- A9 00   LDA  # $ 00
0327- 85 07   STA  $ 07
0329- A9 BA   LDA  # $ BA
032B- 8D 12 04 STA  $ 0412
032E- BD 15 04 STA  $ 0415
0331- A9 01   LDA  # $ 01

```

```

0333- 85 22   STA  $ 22
0335- A9 00   LDA  # $ 00
0337- 85 06   STA  $ 06
0339- A2 00   LDX  # $ 00
033B- A0 00   LDY  # $ 00
033D- 86 1A   STX  $ 1A
033F- B5 07   LDA  $ 07,X
0341- C9 0A   CMP  # $ 0A
0343- 30 08   BMI  $ 034D
0345- 38      SEC
0346- C8      INY
0347- E9 0A   SBC  # $ 0A
0349- C9 0A   CMP  # $ 0A
034B- 10 F9   BPL  $ 0346
034D- 48      PHA
034E- 98      TYA
034F- 69 B0   ADC  # $ B0
0351- A6 06   LDX  $ 06
0353- 9D 10 04 STA  $ 0410,X
0356- 68      PLA
0357- 69 B0   ADC  # $ B0
0359- E8      INX
035A- 9D 10 04 STA  $ 0410,X
035D- E6 06   INC  $ 06
035F- E6 06   INC  $ 06
0361- E6 06   INC  $ 06
0363- A6 1A   LDX  $ 1A
0365- E8      INX
0366- E0 03   CPX  # $ 03
0368- D0 D1   BNE  $ 033B
036A- 68      PLA
036B- A8      TAY
036C- 68      PLA
036D- AA      TAX
036E- 68      PLA
036F- 40      RTI

```

邮 购 服 务

电子工业出版社软件部向广大用户提供
LASER310/200 计算机磁带软件,质量可靠,欢迎选
购,定价如下:

- | | |
|-------------------|-----------------|
| 1. 天体运行 10 元 | 8. 英语同义词 10 元 |
| 2. BASIC 入门① 20 元 | 9. 键盘绘图 15 元 |
| 3. BASIC 入门② 20 元 | 10. 一元二次方程 10 元 |
| 4. 简汉字库 10 元 | 11. 二元一次方程 10 元 |
| 5. 监 控 15 元 | 12. 测试图 5 元 |
| 6. 标准带 10 元 | 13. 打字编辑 20 元 |
| 7. 机械波 10 元 | |

- | | |
|-----------------|-----------------|
| 14. 电脑唱歌 8.50 元 | 20. 打飞机 8.50 元 |
| 15. 快速反应 8.50 元 | 21. 宝龄球 8.50 元 |
| 16. 抓特务 8.50 元 | 22. 吃豆子 8.50 元 |
| 17. 溶洞探险 8.50 元 | 23. 陷井 8.50 元 |
| 18. 高炮射击 8.50 元 | 24. 冲出迷宫 8.50 元 |
| 19. 超级大盗 8.50 元 | 25. 取宝 8.50 元 |

另加邮费 15%。

联系地址:北京 173 信箱软件部 邮编 100036

联系人:谈众安

用 EPROM 制作多路报警器

王洪伟

用 2K 的 EPROM2716 制作的 11 路报警器非常简单,如图 1 所示。

地址线 $A_0 \sim A_{10}$ 作报警器输入端,数据线 $D_0 \sim D_6$ 用于驱动七段 LED 数码管显示, D_7 用作触发报警声电路。 $BG_0 \sim BG_6$ 为驱动管。

正常情况下,通过前级的报警探头使 $A_0 \sim A_{10}$ 维持低电平, $D_0 \sim D_7$ 输出 000H 单元数据,LED 显示为零,报警声不触发。当有报警输入使 $A_0 \sim A_{10}$ 中某一位出现高电平时,不妨设 A_2 变为高电平,其余皆为低电平, $D_0 \sim D_7$ 输出 002H 单元数据,数码管显示为 2,同时 D_7 的高电平触发报警声电路推动喇叭发声,这种状态一直维持到报警解除, A_2 变为低电平为止。由于有可能产生某几路报警同时输入的情况,因此通过对 EPROM 编程,使 $A_0 \sim A_{10}$ 进行优先排队,优先顺序是 A_{10}, A_9, \dots, A_0 。

EPROM 的编程见图②。可将一片已擦除干净的 2716 插入单板机 TP801A 的编程插座中,分段输入数据。例如:100H~1FFH 地址应输入 EF 数据。数码管用共阳的 LED 数码管。报警声电路可用 KD-9561 等音乐片。

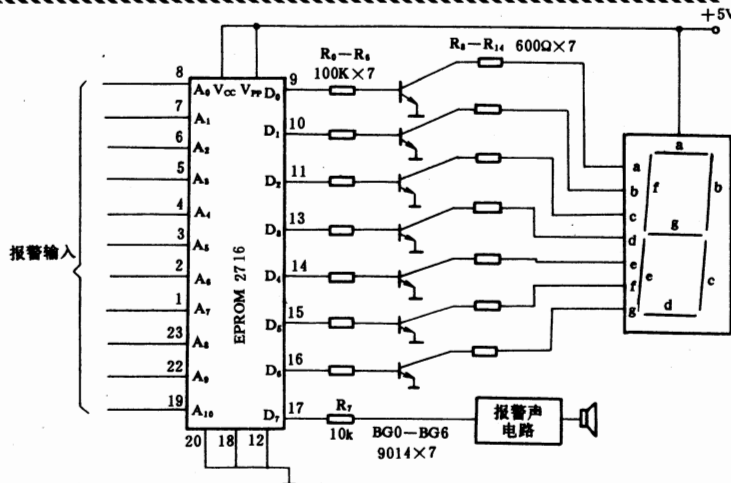


图 1

报警输入 $A_{10} A_9 A_8 A_7 A_6 A_5 A_4 A_3 A_2 A_1 A_0$	对应 EPROM 首址	对应地址数据		数码显示
		二进制 $D_7 D_6 D_5 D_4 D_3 D_2 D_1 D_0$	十六进制	
0 0 0 0 0 0 0 0 0 0 0	000H	0 0 1 1 1 1 1 1	3F	0
0 0 0 0 0 0 0 0 0 0 0	001H	1 0 0 0 0 1 1 0	86	1
0 0 0 0 0 0 0 0 0 1 x	002H	1 1 0 1 1 0 1 1	DB	2
0 0 0 0 0 0 0 0 1 x x	004H	1 1 0 0 1 1 1 1	CF	3
0 0 0 0 0 0 0 1 x x x	008H	1 1 1 0 0 1 1 0	E6	4
0 0 0 0 0 0 1 x x x x	010H	1 1 1 0 1 1 0 1	ED	5
0 0 0 0 0 1 x x x x x	020H	1 1 1 1 1 1 0 1	FD	6
0 0 0 0 1 x x x x x x	040H	1 0 1 0 0 1 1 1	A7	7
0 0 0 1 x x x x x x x	080H	1 1 1 1 1 1 1 1	FF	8
0 0 1 x x x x x x x x	100H	1 1 1 0 1 1 1 1	EF	9
0 1 x x x x x x x x x	200H	1 1 1 1 0 1 1 1	F7	A
1 x x x x x x x x x x	400H	1 1 1 1 1 1 0 0	FC	b

(x 表示任意电平,即 0、1 皆可)

图 2

北京安华科技市场西城经营部

主要经营:电子计算机、电子元器件

各种微机:

• 中华学习机:CEC-I ¥900.00 小蜜蜂-I ¥920.00

CEC-E:驱动器 2 个 1150 元/台,扩展器 1 个。

• 家庭 PC 机:¥2950

主频:4.77/12 兆 RAM:640KB 软驱:360KB×2,101

键盘 14" 双频单显

Super286:¥6000 元

主频:10/21 兆 RAM:1MB 可扩 5MB^{51/4} 软驱:1.2MB.

360KB 各 1 硬盘 3.5"40MB. 101 键盘 14"双频单显 数显机箱 选配:14"CGA 彩显加 1500 元 14"1024×768VGA 彩显加 2300 元

各种:APPLE II、CEC-I、PC 机接口板;九针、24 针打印机,还承接各种微机的修理业务,对教育部门、个人购机实行优惠。

地址:北京西四北大街 120 号 电话:65.6342

邮编:100034 联系人:孙义强 宋涵华 BB 机:126 呼 63882

键盘电视游戏机的 FBASIC 语言及程序设计(上)

于 春

目前,市场上销售的家用电视游戏机(如小天才、任天堂等)都开始增加与游戏机配套的键盘,不但能进行电视游戏,还具有算术运算、谱曲演奏、英文打字、背景绘图和使用 FBASIC 语言进行各种运算程序、游戏程序设计等功能。使孩子们从游戏中学习电脑知识和基本编程技巧。将会有越来越多的“游戏机迷”,带着游戏机画面是怎样画出来的疑惑进入电脑之门。

第一讲 FBASIC 程序浅述

一、FBASIC 语言的概念

电脑进行数值计算、数据处理、自动控制等工作,是通过执行一系列的命令而完成的。这些命令叫指令。一条指令进行一种操作,它告诉电脑进行什么样的运算,有哪些数参加运算,结果如何处理等。所以指令可以看作是人和电脑进行交谈的语言,称为电脑语言。目前电脑语言种类也很多,初学者一般都常用 BASIC 语言。

BASIC 是 BEGINNER'S ALL-PURPOSE SYMBOLIC INSTRUCTION CODE(初学者通用符号指令代码)一词的缩写。它的特点是语法结构简单、学习掌握容易,是一种会话式语言。它还可以不编程序,利用键盘象计算器那样直接进行运算。如欲求 35 与 25 两数之和,可用 BASIC 语言编一段程序进行运算:

```
10 A=35
20 B=25
30 C=A+B
40 PRINT C
```

也可以直接利用键盘进行立即运算:
PRINT 35+25

BASIC 语言有许多版本。不同型号机器的 BASIC 语言也略有不同。键盘电视游戏机所用的语言是日本 MSX BASIC 语言的扩展,称为 FAMILY BASIC 语言,简称 FBASIC 语言。FBASIC 语言除符合 BASIC 语言的指令、语法、逻辑运算、程序设计等一般的通用规则及标准外,其显著特点是在简化指令的前提下,利用最简单的程序,最少的容量完成绘图。

上例程序是进行两数求和运算,为加深认识,下面再列一个完整的 FBASIC 程序。在电子电路中计算流过并联电阻的总电流 I,计算公式为 $I = U/R = U(\frac{1}{R_1} + \frac{1}{R_2} + \frac{1}{R_3})$ 若 U、R₁、R₂、R₃ 已知,则求 I 的程序可写为

```
10 R1=10
```

```
20 R2=20
30 R3=30
40 U=12
50 I=(1/R1+1/R2+1/R3)*U
60 PRINT I
70 END
```

这个程序的头四行是提供原始数据,第五行是计算 I,第六行是将计算结果显示打印出来,最后一行是说明程序到此结束(当然该程序还可以简化)。由此所见,程序是由一些英语词汇或缩写和一些数学表达式来表达的。我们学习电脑语言就是要学会这些表达方式,知道那些符号是合法的,数据如何书写,函数怎样命名,有些什么句型及句型的含义和书写格式,一个程序如何构成以及编程技巧等等。游戏机外接教学键盘,必须插上 FBASIC 语言卡。这些卡上有编制好的专用程序,如卡通图案,电子音乐,计算器等,有关这些专用程序的用法,可参阅有关厂家的说明书,本讲座只介绍 FBASIC 的常用知识及编程方法。

二、FBASIC 语言的构成

1、FBASIC 所用的字符

FBASIC 语言中规定使用如下字符:

- 26 个英文字母,有大、小写之分,一般使用大写字母。
- 0~9 十个阿拉伯数字。
- 算术运算符:+(加)、-(减)、*(乘)、/(除)、MOD(求余数)。
- 关系运算符:=(等于)、>(大于)、<(小于)、>=(大于等于)、<=(小于等于)、<>(不等于)。
- 逻辑运算符:AND(与)、OR(或)、NOT(非)、XOR(异或)。
- 分隔符:,(逗号)、:(冒号)、“(双引号)、’(单引号)、;(分号)、-(连号)、.(点)。
- 其它符号:\$ (字符串变量)、#(半音)、&(十六进制符号)、?(代替 PRINT)。

2、数据

FBASIC 使用的数据有两类,一类是数值数据;一类是字符串数据。

a. 数值数据为整型数。整型数可以是正整数或负整数。不含正负号的数认为是正整数。最小的整数是-32768,最大的数是+32767。FBASIC 不识别实型数,即没有小数(在进行算术运算时可以有小数)。遇到小数,计算机自动舍去小数点后的数字,且不遵循四舍五

入法则。

b. 字符串数据。用双引号括起来的一串字符叫字符串。它也可以作为数据参与运算和处理。字符串中的字符可以是字符集里除引号外的所有字符。(由于引号要作为字符串的标志,所以字符串中不能再包含引号。)字符串最长不超过 31 个,如下一些是字符串数据:

“FBASIC YU YAN HE CHENG XU”

“FCS-90 IS AMICROCOMPUTER”

“X,Y,Z=”

“(50X-3A)/6A=”

3、变量

变量有简单变量、下标变量两种。

简单变量又分整型变量和字符串变量两种。字符串变量也称串型变量。

简单变量的名称可以用一个、二个、三个字符表示,表示规则为:

×××

最后一个字符表示变量类型。\$ 表示串型变量,没有这个字符表示整型变量。

可以是字母或数字,也可以没有。

第一个字符必须是字母。

除了变量类型符号 \$ 之外,变量名也可以超过两个字符,变量名称最长可以到 255 个字符(但只能是字母和数字)。计算机将按照你命名的变量名称存贮和打印,但运算时它只识别头两个字符,因而 ABC1 和 ABC2 代表同一个变量 AB,EOOP 和 EOCC 是同一个变量,当然 EOOP 与 EOOP\$ 不同。

在 FBASIC 语句中使用了一些特定的词,如 GO-TO、FOR、THEN 等,这些词在 FBASIC 中都有确定的语法意义,它们叫保留字,我们设定的变量名不能是保留字,也不能包含保留字,否则计算机机会把它当做保留字

对待,并不认为是变量。

变量的类型应与数据类型一致。整型变量只能取整型数,字符串变量的值只能是字符串常数,即某个具体的字符串。以后还会看到,下标变量所能取的值也是数值或字符串。下标变量留待第二章介绍。

4、数学运算的运算符,运算规则与表达式

a. 运算符:

FBASIC 中的数学运算符与数学中的运算符基本相同。

加号+,减号-,乘号*,除号/,求余数 MOD。

注意:表达式中“*”号不能省略,也不能用“.”代替。如 $5 * Y$ 不能写成 $5Y$ 或 $5 \cdot Y$, $A * B$ 不能写成 AB 或 $A \cdot B$ 。

b. 运算规则与普通代数中运算规则一样。如果表达式中有括号,则先算括号的内容,然后是函数,乘除、求余数、加减、关系运算,逻辑运算。它们的优先顺序是 $() \rightarrow \text{函数} \rightarrow * , / \rightarrow \text{MOD} \rightarrow + , - \rightarrow = , < > , > , < , > = , < = \rightarrow \text{NOT} \rightarrow \text{AND} \rightarrow \text{OR} \rightarrow \text{XOR}$ FBASIC 中的括号无大、中、小之分,一律用圆括号 $()$,可以在 $()$ 再套 $()$ 。

c. FBASIC 表达式

数学中,用运算符、括号把常数、变量连接起来的式子叫代数式。FBASIC 语言中用符合 FBASIC 规定的运算符和括号把常数、变量连接起来的式子叫 FBASIC 表达式。下面列表举例说明表达式的写法:

代数式	FBASIC 表达式	错误表达式
$\frac{A}{B} \div \frac{C}{D}$	$(A/B)/(C/D)$	$A/B/C/D$
$a[x+b(x+c)](A*(X+B*(X+C))A(X+B(X+C)))$		

下期刊登三、FBASIC 语句和程序。

《全国中小学计算机教育资料汇编》

简介

为总结我国“七五”期间中小学计算机教育方面的经验,向广大教师、中小学校及有关教育管理部门提供计算机教育方面的信息,以促进计算机教育事业的发展,“全国中小学计算机教研中心”和电子工业出版社编辑出版了《全国中小学计算机教育资料汇编》,内容包括:

1. 有关部委及省、市领导部门关于计算机教育方面的文件;
2. 中央领导同志的题词与讲话;
3. 国内外计算机教育工作会议纪要及学术活动;
4. 教师的优秀论文及研究报告;
5. 中学计算机课教学大纲及有关规定;
6. 全国中学计算机配置及教师队伍情况的调查;
7. 全国及各省市青少年计算机竞赛与竞赛题,国

际信息学奥林匹克竞赛题;

8. 全国教育软件评审机构、评审标准、优秀软件简介;

9. 开展计算机教学的中小学名单,科技馆、青少年宫名单;

10. 有关计算机普及教育的管理部门及学会、协会等机构通讯录;

11. 全国出版的青少年计算机类图书、报刊目录;
本《汇编》是我国计算机教育方面的一本较全面的资料,可供中学师生、计算机教研工作人员、中小学校领导及各级教育管理部门参考。

本《汇编》16 开本,75 万字,定价 13.50 元,1991 年 8 月出版。



AT(286)微机硬盘处理一例

刘文锋

AT DIAGNOSTICS 是专门用于 PC—AT(286)机,而无法用于 PC—XT 机的诊断测试软件,常用于帮助 AT(286)机使用者判断计算机的系统板、键盘、磁盘及驱动器等装置是否在正常操作状态和故障的过程。还有对硬盘进行低级格式化处理等功能,由于计算机内装有大量的数据信息,这一功能不敢尝试使用,其使用方法也不得知。为了同行在使用 AT 机的过程中,遇到硬盘出现问题,或者被染上病毒,需要对硬盘进行处理时,不至于走弯路,现将操作步骤介绍于后。由于篇幅限制,仅给出有关的屏幕显示,供参考。

1. 装置 AT DIAGNOSTICS 的盘片入 A 驱动器,进行冷启动或热启动,屏幕出现菜单,有 6 种选择项:

- 0—SYSTEM CHECKOUT (系统诊断)
- 1—FORMAT DISKETTE (格式化软盘)
- 2—COPY DISKETT (复制软盘)
- 3—PREPARE SYSTEM FOR MOVING (锁系统)
- 4—SETUP (参数设置)
- 9—END DIAGNOSTICS (结束)

2. 选择系统诊断“0”并回车,等待约一分钟,给出配置情况。各机器的配置不一样,显示也不同!

- 1—SYSTEM BOARD (系统板)
- 6—2 DISKETTE DRIVE (S) AND ADAPTER (2 个软盘驱动器及接口)
- 17—1 FIXED DISK DRIVE (S) AND ADAPTER (1 个硬盘驱动器及接口)
- IS THE LIST CORRECT (Y/N)? (正确否?)

3. 如显示与实际配置不相符,可按“N”并回车,选择补充正确后再继续。一般是相符的,选择“Y”并回车,进入系统诊断菜单。

- 0—RUN TESTS ONE TIME (诊断一次)
- 1—RUN TESTS MULTIPLE TIMES (诊断多次)
- 9—END SYSTEM CHECKOUT (结束)
- SELECT THE ACTION DESIRED? (选择操作号)

4. 选择测试一次“0”并回车;其显示的选择项与步骤“2”显示的相同,其要求是:选择要操作的号或按回车全部诊断。

(上接第 29 页)

(3)数据第七位表示农历月比公历月滞后信息。置 1 时农历月份比公历月滞后 2 个月数,置 0 为滞后 1 个月数。

(4)数据的第八位表示大小月信息。置 1 时该农历月为大月,置 0 农历月为小月。

例如查表得 89 年 6 月 2 日的数据为 0DDH。即 2

5. 选择硬盘驱动器“17”并回车,对硬盘进行操作。

进入

- FIXED DISK DIAGNOSTIC MENU (硬盘诊断菜单)
- 1—WRITE, READ, COMPARE (ON TEST CYLINDER) 读写,比较(测试柱面))
- 7—FORMAT MENU (格式化菜单)
- 9—RETURN TO CONTROL PROGRAM FOR OTHER OPTIONS (返回主控程序)(其它操作,键入操作数,驱动器标识(1,C)并回车)

TYPE THE OPTION NUMBER, DRIVE ID (1,C), AND PRESS“ENTER”

6. 选择格式化菜单“7”并回车,对硬盘进行格式化处理。进入

- FORMAT SELECTION MENU (格式化选择单)
- 1—CONDITIONAL FORMAT (条件格式化)
- 2—UNCONDITIONAL FORMAT (非条件格式化)
- 9—RETURN TO FIXED DISK MENU FOR OTHER OPTIONS (返回硬盘菜单)(其它操作)
- TYPE THE OPTION NUMBER, DRIVE ID(1,C), AND PRESS“ENTER” (键入操作数,驱动器标识(1,C)并回车)

7. 键入“1,C”并回车,即条件格式化。“1,C”为硬盘的标识。

8. 询问继续否,回答“Y”,开始对硬盘进行低级格式化,需要时间约十分钟。

9. 格式化完后,根据提示进行选择,退出诊断系统。

10. 用 AT(286)机的系统盘从新启动机器后,用 FDISK 建立分区,大于 30 兆的硬盘,还要建立扩展 DOS 分区,并建立逻辑 D 盘。活动分区只能为硬盘的主 DOS 分区 C 盘,逻辑 D 盘不能作为活动的 DOS 系统盘。

11. 用 FORMAT C:/S/V 和 FORMAT D:高级格式化后,用 COPY 命令或 RESTORE 命令装入 DOS 文件及实用程序与数据。

进制数 11011101。分离开来即 89 年 6 月 2 日时,农历为 89 年 4 月 29 日。4 月为大月有 30 天。农历转换流程图如图 4 所示。

多功能钟还可在此基础扩展有功能秒表、节日的显示、农历 24 节气的显示转换、临时加闹、数字脉搏计、数字温度计等。从而使单片机更多发挥本身功能多功能时钟整个占 2K 字节 ROM。

从屏幕显示异常判断苹果主机板故障

扬州工学院计算中心 张艾美

苹果微机及其兼容机在我国推广使用已有多年,目前不少机器正处于故障率较高时期。由于苹果微机及其兼容机的原理和结构相对于其它功能更强的微机来说要简单些,尤其是对那些具有两台以上同类型号苹果机的用户来说,在维修过程中使用替代法或比较法,即使没有多少维修经验,也能将一般的故障排除。问题是对仅有一台苹果机或兼容机的用户来说,怎样在经济代价不大的原则下,使用普通的万用表和高频示波器,自己动手解决检修问题。笔者根据多年微机维修实践,在此介绍一些从苹果机及其兼容机屏幕所显示的异常现象入手,分析解决检修问题的方法,供同行参考。

故障现象	故障可能出现的位置 及需检测或更换的芯片
通电后电源指示灯亮,屏幕上只有光栅显示	①时钟产生区,检查时钟发生器的 A ₂ 、B ₁ 、B ₂ 、C ₁ 、C ₂ 。 ②视频地址发生器,检查 D ₁₁ 、D ₁₂ 、D ₁₃ 。 ③影响产生区,检查多路数据选择电路部件 A ₉ 、A ₁₀ 。
通电后屏幕上只出现绿黑色边框,没有任何图形或文字	检查存储器 RAM 的外围电路即或门 H ₁ ,与非门 E ₂ ,多路选择器 E ₁₂ 、E ₁₃ 、J ₁ ,地址缓冲器 H ₃ 、H ₄ 、H ₅ ,地址译码器 F ₂ 和数据寄存器 B ₅ 、B ₆ 、B ₇ 、B ₈ 等。

故障现象	故障可能出现的位置 及需检测或更换的芯片
通电后屏幕上出现无规则的图形(线条、字母、符号或方块等)	①时钟产生区,检查与非门 A ₂ ,数据选择器 C ₁ 及计数器 D ₁₄ 。 ②影响产生区,检查反相器 C ₁₁ ,或门 C ₁₄ ,高分辨率数据选择器 A ₈ ,视频发生器部分的 A ₁₂ 、B ₁₁ 以及脉冲控制器 B ₁₃ 。 ③程序控制区,首先检查 CPU 的外围电路 H ₄ 地址缓冲器及 H ₁₀ 、H ₁₁ 数据缓冲器,再检查 ROM 部分的 3—8 译码器 F ₁₂ ,I/O 部分的 3—8 译码器 H ₁₂ ,RAM 部分的选择电路 E ₁₄ ,数据选择器 C ₁₂ ,以及与非门 D ₂ 等。

由于苹果机主机板上的芯片都是插在插上的,更换芯片非常方便,关键是找准故障的部位或器件。一台

故障机通电后,必然在 CRT 屏幕上显示异常现象,几种常见故障及其对应部位和需检测或更换的芯片,列于下表中:

根据上述屏幕显示异常,进行分析、检修的若干实例:

例 1、开机后屏幕上出现绿黑色边框,框内无图形或文字。

屏幕上的图形或文字来源于 RAM 中被选中的地址单元信息的输出,现在没有输出信息,应先检查 RAM 部分数据寄存器的芯片 B₅、B₈,如果其中有一个芯片损坏,则信息送不到视频发生器。经检查 B₈ (74LS174) 六 D 触发器的 6 脚、7 脚逻辑关系不对,6 脚输入脉冲器,但 7 脚却输出低电平,即 B₈ 芯片损坏,换片后正常。

例 2、同例 1。

首先按例 1 查 B₅、B₈ 均正常。再检查地址缓冲器 H₃、H₄、H₅ (74LS367),发现 H₄ 中第 2 脚输入端有波形,而对应的第 3 脚输出端无波形,即 H₄ 芯片损坏,换片后工作正常。

例 3、开机后屏幕上只有光栅,而无文字或图形。

查时基脉冲区的有关芯片,发现 B₂ (四异或门 74LS86) 的输入端 10 脚、9 脚与输出端 8 脚逻辑关系不正确,即 8 脚无脉冲信号输出,致使视频发生器部分接收不到 14MHZ 的脉冲信号,而使影像产生区没有同步信号,无法正常工作,换片后正常。

例 4、同例 3。

查影像产生区 A₁₀ 芯片 (74LS194 移位寄存器) 的 3、4、5、6 脚有输入波形,而 12、13、14、15 输出端无电平,即 A₁₀ 损坏,换片后,屏幕显示正常。

例 5、开机后屏幕上出现规则宽边框和中间有无规则线条图案。

用示波器按顺序先检查时钟产生器的输入、输出逻辑关系都正常,再在影像区查 C₁₄ 芯片 (二输入四或门),发现 9 脚上有 R/W 脉冲波形,10 脚上有 ϕ_1 脉冲波形,但 8 脚输出不符合二输入或门的逻辑关系,即该组或门损坏了,换芯片后,屏幕画面正常。

例 6、开机后屏幕上呈现中间垂直深黑宽线条,左右两旁呈对称的淡黑色横线条。

首先检查时钟产生区,发现 C₁ 芯片 (74LS153) 的 11 脚、13 脚、14 脚有脉冲波形,而输出端 9 脚无脉冲波形,致使时序电路同步脉冲输出不正常,整机同步信号不正确,因此视频电路输出不正常而造成屏幕画面混乱,即 C₁ 损坏,换片后工作正常。

例 7、开机后屏幕呈现无规则的粗细不同的横线

条,且线条之间有稀疏的字符。

检查时钟产生区和影象区均无损坏芯片,检查程序控制区,在RAM地址部分, E_{11} 芯片(74LS153)2脚无脉冲选择信号,而芯片插座对应点却有脉冲信号,可见是芯片与插座接触不良造成,切断电源,用无水酒精清洗该插座及芯片腿脚后,再将芯片插入,开机后画面正常。

例8、开机后,屏幕出现规则或不规则排列的小黑方块图案。

检查时钟产生区及影象区未发现异常现象,再检查程序控制区,RAM中的 C_5 损坏或 C_7 损坏都会使屏幕呈现此现象,换片后工作正常。

例9、开机后屏幕呈现不规则的黑绿色边框及不规则的竖线条图案。

经检查发现在程序控制区中 C_{11} 的13、14脚一组反相器坏,致使 H_3 、 H_4 、 H_5 三态门不能正常工作,使视

频部分输出波形混乱,换片后问题解决。

例10、开机后系统不能进入BASIC工作状态,屏幕上显示:

3—PEAKS

F12A⁻ A=C0 X=00 Y=03 P=35 S=F1
* |

或

3—PEAKS

F150⁻ A=C0 X=00 Y=00 P=37 S=EB
* |

且喇叭发出咕咕响声。

经检查并非主机板芯片损坏,而是某些芯片接触不良所造成,只要将主机板上每个芯片用手指往下按,使之接触良好,机器就恢复正常,若主机板积尘较多,可用无水酒精清洗后,再将芯片重新插入,机器即可正常工作。

M2024 型打印机故障检修一例

烟台师范学院 王 峰

有一台打印机(M2024型)用不到半年出现下列故障:连通电源打开电源开关,机器自校正常,但不能联机,并且打出不规则的变形码。

故障分析与检修:我们看该机控制信号部分的接口电路,如图所示:

Centronics 并行接口:

- | | |
|-----------------------|--------------------|
| 1; strobe, 2~9; DATA. | 10; ACK |
| 11; BUSY; | 12 PE(PAPER ERROR) |
| 13; SELECT; | 14; AUTOFEED |
| 15; FAULT; | 16; INIT |
| 17; SELECTIN; | 18~25; GND. |

信号说明:

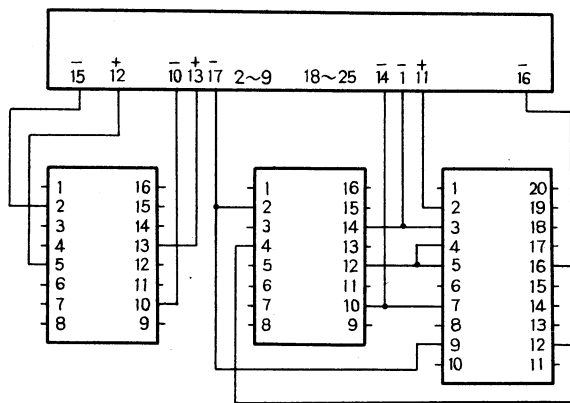
(1) $\overline{\text{DATA}} \text{ TROBE} (\text{CPU} \rightarrow \text{PR})$, 读数据选通脉冲。脉冲宽度最低需要 $1\mu\text{s}$, 正常状态下为高, 变低时读出。

(2) $\overline{\text{ACKNLG}} (\text{CPU} \leftarrow \text{PR})$, 这是 $4 \pm 1\mu\text{s}$ 的应答脉冲, 连文字输入完毕或者打印动作结束时产生, 但是输入 DC3 信号后变成 BUSY 状态, 不能输出 ACKNLG 信号。

(3) BUSY (CPR \leftarrow PR) 表示打印机能否动作的 DC 电平信号:

(4) DATA (CPR \rightarrow PR) 数据信号, 脉冲宽度最低需 $3\mu\text{s}$ 。

(5) PE (CPU \leftarrow PR) 打印纸不正常时为高电平。



(6) SELECT (CPU \leftarrow PR) 打印机选择状态时为高电平。

经我们检查发现是 U_{23} (74LS240) 烧坏。更换后仍不能打印。进一步检查打印机控制接口电路, 发现上述信号均能正常传送, 但 7406 无取反作用, 更换 7406 后, 故障排除。之所以导致此类故障是因为电网电压波动太大所致。

计算机设备故障检修表

咎金省 宋靖涛

笔者从事计算机维修工作,就近年来所接触的部分故障作以下小结,供同仁们参考和借鉴,如遇相同类型的设备发生故障,不妨测试如下部位,一般尚可解决。

故障设备	现象	部位	可代元件
614 型交流稳压器	2A 交流保险烧断	控制整流晶体二极管 2ZD14M 击穿	
	电压不稳	大功率偏置电阻烧断或电子管坏	
王安电源	无电压输出,保险烧断	压敏电阻 RV250 短路,开关管 2N6675 烧坏短路。	C1942 或 BUX48
王安 12 寸显示器	无显示	行输出管 BU407 坏或行输出变压器高压包局部短路	
王安硬盘控制卡	起动失败	查专用芯片 L4WD1100-05 L23 WD1100—01	
王安 512K 内存扩充卡	I/O 系统错	内存芯片 4164 坏,具体位置可用王安诊断测试代码。	4864
M2024 打印机	自检正常,联机后死机	接口板 7414 和 74LS06 易坏	74LS14
	无 +36 伏电源电压	调整管 TR22 D1026 坏	
AR3240 打印机	无交流输出	变压器坏(内部保护电阻烧断,急用可短接此电阻)	
AR2463 打印机	字车步进电机不能被驱动	B825, 3A 保险电阻坏	B601
GW100 显示器	无光	行输出管 V403 2SD1365	BU407
	水平一条亮线	TDA1170 坏	
GW300 显示器	无光	行输出管 D1426 坏	
GW286 主机	实时时钟 CMOS 信息丢失	换锂电池(6 伏)	
IBM PC/XT	PARITY CHECK1	奇偶校验 RAM 芯片 4164 或奇偶校验芯片 74LS280 损坏	
GW286EX	机器加电自检 RAM 时死机,无错码显示。	查数据收发器 U6 (74F646) U12 (74F245) U9 (74F245)	
360K 软盘驱动器	读错	磁头脏,用清洗盘清洗;或磁头小车卡死,上润滑油。	
PULSE 2000R 型 UPS	逆变器不工作	末级驱动晶体管 QM50H—H 坏	
苹果机电源 MEQ—063P	无输出电压	调整管 C2502 坏 TL494CN 坏(无锯齿波形)	BU508 BU407

IBM PC/XT 微型计算机 检修一例

平凡

故障现象:

开机时,出现一长一短“嘟”声后,系统不自检,不启动。

故障分析与检修:

IBM PC/XT 机一旦系统板上 8088CPU 出了故障,或系统时钟工作不正常,再就是数据总线有损坏元件,都将导致以上故障现象的出现。

首先,开机用示波器测如图所示 8284 时钟发生器引脚⑧(CLK 时钟),有脉冲波形信号,再测 CPU8088 引脚⑩(CLK 时钟),也有脉冲波形信号,且波形很正常。再次开机时,测 8088 引脚②(RESET 复位)有一个明显的正脉冲(该脉冲是先从低电平变为高电平,进而再恢复为低电平)。因此,基本说明系统板上的 CPU8088,时钟发生器 8284 电路没问题,通过测 8088 引脚②(READY 准备好)为高电平,推测故障可能出在数据总线电路上。

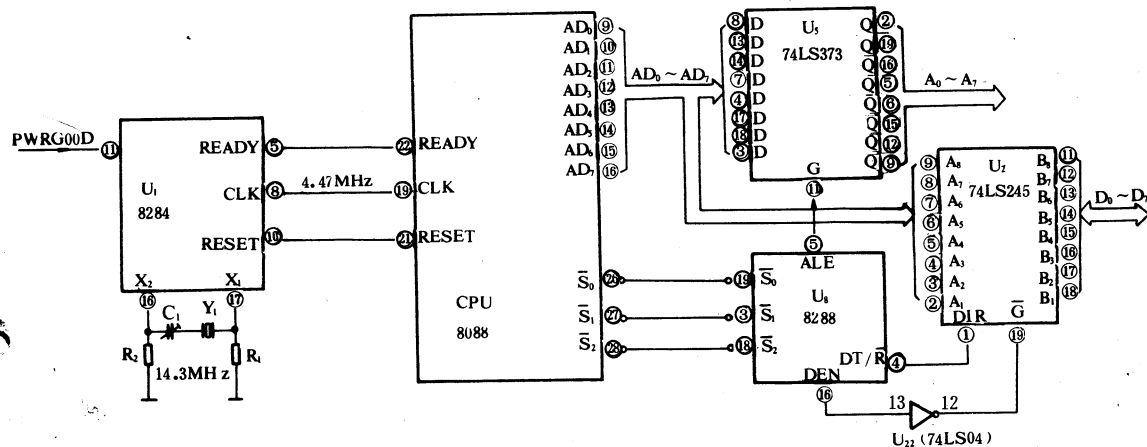
这里,先了解一下数据总线的工作过程,如图为系统板数据传送情况。

在时钟发生器 8284 引脚①内部,由斯密特触发输入电路和同步触发器组成的复位逻辑电路,与 8284 引脚①外部 RC 网络构成加电复位电路。在开机时,加电复位电路便产生 PWRGOOD(电源好)脉冲信号,使复位逻辑电路进入工作状态,从 8284 的引脚⑩输出一个先从低电平升为高电平,进而再恢复为低电平的正脉冲复位信号。该复位信号进入 8088 引脚②使 8088CPU 复位,复位后的 CPU 便从 \$FFFF0 单元开始执行指令。如图可见,数据总线是通过 V₂(74LS245)数据缓冲

器与 8088CPU 的数据线连接的。由于 8088CPU 的数据线引脚(AD₀~AD₇)又兼地址低 8 位(A₀~A₇)的引脚,所以,8088CPU 引脚⑨~⑯采用分时传送方式将地址与数据信息分别传送,即在不同时间里,既传送数据信息,又传送地址信息。8088CPU 的总线控制器 V₈(8288)为 20 脚双极型芯片,它能根据 8088 执行指令时提供的状态信号 S₀、S₁、S₂建立控制时序,输出相应的地址锁存允许信号 ALE 和读写控制命令。图中的 CPU 地址锁存器 V₅(74LS373)用来锁存地址信息。当 8288 引脚⑤输出的地址锁存允许信号 ALE 为高电平时,地址锁存器 V₅的锁存控制端 G(引脚⑪)接受 8288 输出的 ALE 信号开始工作,把 CPU8088AD₀~AD₇ 送来的地址信息锁存选通到 V₅中,然后送往地址总线。另一方面,8288 总线控制器根据 8088 的状态信号 S₀、S₁、S₂,从 8288⑯脚发出的数据总线允许信号 DEN,经过 V₂₂(LSO4)反相器反相后,至双向数据收发器 V₂(74LS245)引脚⑩,控制 V₂工作,使 8088 的数据总线和系统数据总线相接。同时,8288 引脚④发出数据发送/接收信号 DT/R 至 V₂引脚①,以控制数据传输方向。当 DT/R=1 时,数据从 8088CPU 经 V₂传送到数据总线;当 DT/R=0 时,数据从总线经 V₂传送到 8088CPU。

如果机器正常的话,其数据总线上的数据信息将处在不断变化之中,用示波器观察将显示出变化无常的断续波形。测试结果,数据总线上无波形信号。于是,测试双向数据收发器 V₂的输入和输出端,均为低电平,进而再测试 V₂引脚⑩收/发控制端,无论开机瞬间或开机之后,总在高电平上。在这种状况下,数据收发器 V₂是不能正常的。造成 V₂引脚⑩总为高电平的原因有三个:一是 V₂本身出现故障;二是反相器 V₂₂不正常;三是总线控制器 V₈损坏当用示波器测反相器 V₂₂输入引脚⑬时,示波器显示有波形,而 V₂₂输出引脚⑫无波形,断定故障出在反相器 V₂₂芯片上。因为 V₂₂损坏,使 V₂数据信息的收发功能失控,导致 8088CPU 指令和数据信息道中断,引起机器不能启动。

更换 V₂₂(LSO4)芯片后,微机恢复正常工作



MP-I 微电脑 快速检修法

青岛大学 李 红

本方法只用万用表就可在 20 分钟内检查出 MP-I 机 90% 以上的常见故障。

检查前,先用万用表测量一下印刷电路板上各芯片的 V_{CC} 和 GND 两脚, $V_{CC}=5V$, $GND=0V$ 。如若不是,说明电源引线开路或断路。

一、显示器故障的检修

1、显示电路的检查

将 U_7 (74LS00) 从电路板上卸下,则 DMA 开关停止工作。由于键盘上的地址三态门和数据三态门都处于高阻态,因而键盘与地址总线、数据总线断开。又因 $DMA=1$, MPU 也与总线断开。

再将 U_{18} (74LS02) 卸下,则存储器 2114 的片选端 $\overline{CS}=1$, 因此存储器也与总线断开。

由地址译码电路可知,此时输入接口电路 U_{20} (三态门 74LS244) 被选中,但其 8 个数据输入端悬空(高电平),因而输出端为 1。输出接口 U_{21} (8D 触发器 74LS374) 未被选中,对总线无影响。

此时,10 根地址线悬空(高电平),8 根数据线为 1,因此,18 个 LED 应全亮,哪个灭灯,说明故障在哪条线路上。

然后,取下万用表上的一根表笔用作短路线,一端接地,另一端分别触碰 10 根地址线(电路板上左上角处的 10 根地址线端口(沉铜孔))和 8 根数据线(板上右上角 8 根地址线端口)。碰到哪根,相应 LED 就应灭。否则,说明这条线路上存在故障。

2、数据总线的检查

将 8 位数码开关拨下,改插到输入接口插座,此时,由于输入接口导通,因此 8 个开关可以直接控制数据总线。

① 令 8 位数据开关分别置 0,相应的 LED 灭,这说明数据线 $D_7 \sim D_0$ 之间彼此无短路现象。

② 令 8 位开关全部置 0,用万用表分别检查 U_{12} (6502), U_{16} (2114), U_{17} (2114)、 U_6 (74LS244) 各片上的数据端引脚也应为 0;再将数据开关全置 1,各相应引脚也应为 1。否则说明某数据线上存在故障。

显示器故障的原因有:LED 损坏或两个引脚焊反, U_1 、 U_2 、 U_3 (反相器 74LS04) 芯片个别引脚断路,其插座焊点开路或断路,线路板上引线断路或短路,沉铜孔开路等。

显示器检修完毕,将 U_7 、 U_{18} 插回原处。

二、键盘故障的检修

1、DMA 键或 OFF 键不正常

先查 DMA 开关电路。按过 DMA 键后, U_7 的 6 脚(即 DMA 信号线)应为 1,3 脚(\overline{DMA} 信号线)为 0。按过 OFF 键,则 6 脚、3 脚电平翻转。DMA 开关电路出现故障的可能原因:按键本身短路或开路, U_7 引脚开路或短路、外围阻容元件开路或损坏等。如果 DMA 开关电路一切正常,则继续检查 U_5 的 8、9、10 脚处的三态门是否正常。

2、 A_0 、 A_1 、 A_n 键不正常

A_0 键抬起时, U_9 、 U_{10} 、 U_{11} 三片的 2、3 脚都为 0;按下 A_0 键,2、3 脚为 1。

A_1 键抬起时, U_8 的 1、2 脚为 1,3 脚为 0;按下时,1 脚为 0,3 脚为 1,而 2 脚仍为 1。而 3 脚输出的信号应能送到 U_9 计数器的时钟端 \overline{CP}_0 (14 脚)。而 U_9 的 11 脚输出进位信号应能送到 U_{10} 的 \overline{CP}_0 端,同理, U_{10} 11 脚应送到 U_{11} \overline{CP}_0 端。

A_n 键抬起时, U_8 的 2 脚为 1;按下时,由于接通了振荡电压(约为 1.2~2.7V),因而 U_8 的 2 脚电压为 1.2~2.7V,这时,地址应能连续加 1。

上述三键出现故障的可能原因有:按键开路或断路, $U_7 \sim U_{11}$ 引脚接触不良,外围阻容元件开路或损坏等。

三、存储器的故障检修

1、检查存储器片选端 \overline{CS}

由地址译码电路可知,当地址 $A_9A_8=00$ 、01、11 时,选中存储器 \$00、\$01、\$11 页,此时两片 2114 的 \overline{CS} 端为 0, U_{20} 的选通端(1 脚及 19 脚)应为 1, U_{21} 的 CP 端为 1。当 $A_9A_8=10$ 时,选中 I/O 接口,这时,两片 2114 的 \overline{CS} 端为 1, U_{20} 的 1 脚及 19 脚当 $A_0=0$ 时为 0, $A_0=1$ 时为 1, U_{21} 的 CP 端为 1。

如果出现异常,说明地址译码电路有故障,可检查 U_{20} 、 U_{21} 各引脚与总线连接情况。

2、检查存储器与读写线 R/\overline{W} 的连接情况。

按一下 DMA 键后,则 U_8 的 8 脚(\overline{DMA} 信号)为 0,此时 \overline{WR} 键抬起,两片 2114 的 R/\overline{W} 端应为 1;按下,则 2114 的 R/\overline{W} 端为 0,如果异常,说明 2114 与读写线的连接有故障。

3、检查存储器与地址总线 $A_0 \sim A_9$ 的连接情况。

将 8 位数据开关全置 0,利用 A_n 键及 RW 键,将存储器 \$00、\$01、\$11 页全清零(\$01 页为 I/O 接口地址)。

然后在 \$000 单元写入 \$FF。如果存储器与地址总线连接良好,则用键盘重新读数据,除 \$000 单元为 \$FF,其余均为 \$00。如果存储器与某一根地址总线断路,例如 2114 的 7 脚(A_2)与地址线 A_2 断路,则此脚电平将处于不稳定状态;可能为 0,也可能为 1。若为 1,则当用键盘给 \$000 单元写进数据 \$FF 时,存储器的实际地址将为 $A_9A_8A_7A_6A_5A_4A_3A_2A_1A_0=00\ 0000\ 010\ 0$,因而 \$FF 实际写入 \$003 单元。这样,读数据时,会在 \$000 和 \$003 两个单元都读出 \$FF。

当 2114 的 7 脚处于不稳定状态 0 时,用同样方法可以查出错误。

4、检查存储器与数据总线 $D_0 \sim D_7$ 的连接情况,这在前面一、2 中已进行。

四、微处理器故障的检修

按一下 OFF 键,地址 LED 全亮,按住 RES 键,MPU 将发出一个随机地址,地址 LED 显示这个随机数。当 RES 键抬起,LED 又全亮。如有异常,用万用表检查 6502 的 40 脚(\overline{RES})的电平变化:按下 RES 键为 0,抬起为 1。如果不是这样,应检查 RES 键与 U_8 的 4、5、6 脚以及 RES 信号总线之间的连接情况是否良好,以及 U_8 右边阻容元件是否断路或损坏。其次应检查 U_{15} 构成的时钟发生器是否振荡,振荡时各引脚的电压大约为 1.2~2.7V(12 脚为 0),停振时各脚电平为 0 或 1。如停振,可在 U_{15} 的 11、13 脚之间并联一个 1K Ω 的电阻。如仍不起振,说明 U_{15} 引脚断路、或外围阻容元件断路、损坏。

经以上检修后,如果 MPU 仍不执行程序,则说明 6502 与地址总线的连接存在问题。可将 6502 从电路板上卸下,按一下 OFF 键,此时地址总线受 6502 地址端 9~18 脚控制,由于 6502 已卸下,使 U_{13} 、 U_{14} 的输入端悬空,(下转 26 页)

如何使九针打印机 字迹更清晰

陆秀忠

目前,中华学习机,APPLE 机配备的打印机基本上是九针打印机。其打印针一般为片状针。在这些打印机使用日久后,打印字的笔划明显变粗。当打印小号等中文字时,字迹便不清晰。这对于使用打印纸打字时,还问题不大。若是使用打印蜡纸,则油印出来后,笔划多的字就变得模糊一团了。当然,此时可调换打印针。可是,一来打印针并非每个地方都能买得到,二来还有一个经费上的问题。如果刚好你的打印机正是如此,请别心焦。按下面的方法,便可使你的打印机针头恢复正常或者变得更细一些。下面以 CPA-80 为例。

一、工具准备:中、小号十字头螺丝刀各一把,300 号以上水砂纸一片或者粒度相近的油石一块,软毛刷一个,放大镜一个,小镊子一把,可架在打印机上的平板一块,其长、宽为 $15 \times 5\text{cm}$ 。

二、拆卸打印头:不要把机器从机箱拆出来。仅用螺丝刀把打印头和电缆的固定螺丝拆下就行了,打印头便可拿出到打印机外。此时注意,由于有电缆连结,不能用力拉扯,以免损坏电缆,接着,把可见到一列 9 支打印针头的那一面盖板从打印头上拆下,便可见 9 支片状打印针上下整齐迭成一叠,建议拆到此为止。若把每一支打印针都拆下来,要打磨得一致是很不容易的。

三、打磨针头:这一部分工作最要细致。每支打印针的前半部如图。粗一些的针头大小约为 $0.4 \times 0.4\text{mm}$ 的矩形。现要将其磨成 $0.2 \times 0.2\text{mm}$ 的矩形。打磨共分三步。

第一步,打磨九支并排针头的侧面。把平板架在打印机上,用镊子夹紧九支打印针,把针头部分压在平板上,使九支针头整齐一致。然后用油石或者一小片水砂纸包在钢锯片的外面打磨两侧,每侧磨去 0.1mm 。

第二步,打磨每支针头的上下面,也是每面磨去 0.1mm 。此时要分支打磨。把要打磨的那支针用镊子夹住,稍向一边移出,然后用水砂纸包在钢锯片处仔细打磨。建议磨完每支针头同一个面后,把打印头翻转,再磨每支针头的另一面。这样以免记不清那一支磨过。然后检查各针头磨得是否一致,进行一些修整。

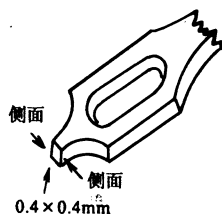
第三步,用水砂纸轻擦每支针头的边、角,使之不致太锐利。

打磨注意事项:(1)为防止磨过头,要经常用放大镜检查针头的形状。(2)在下方用纸垫好,以免铁屑掉进打印机。(3)要经常用软毛刷把针头上的铁屑轻轻刷去,注意刷时针头朝下,以免铁屑掉进打印头内。(4)对针头不可用力扳动,特别是磨打印针头上下两面时更要注意,以免打印针变形。

四、试机:这时要特别小心,把打印头重新装好,便可开机试

磨邊鑿磨
磨邊鑿磨

磨邊鑿磨
磨邊鑿磨



打印针头图

打,由于针头变小,其冲击力便增大,再加上新磨的针头边角锐利,所以很容易打破色带。建议先不装色带,用一张金相砂纸作为打印纸,其砂面要面向打印头,利用自检,把砂纸打几遍,以磨钝针头的边角。并可从纸上字迹的深浅来了解打印力度的大小,以便调整打印头的位置使力度恰到好处。万一力度调不小,可把在打印头上用来固定打印头的螺丝孔靠胶轮那边用圆锉稍稍锉去一些,使打印头的位置可退后一点,打印力度便会小一些。

下面,给出打磨前与打磨后两种字迹的比较如下。有兴趣的朋友不妨动手一试。

YOKO747 型教学 语言学习机的修理

郭长福

该教学语言学习机使用灵便,实用性强,特别是放音卡的走带速度可以根据需要而连续可调,很适合学校外语教学的需要。但在实际使用中,也发现该型机存在不少问题,在修理 40 多台该型机中,总结几点共性问题如下:

1、该型机中的一些插接件,特别是 2 卡的磁头插座极易发生接触不良现象,具体表现为用 2 卡放音时无声,扬声器发出“嘟嘟”的叫声。解决的办法是把该插接件去掉。引线直接焊在印刷电路板上即可。

2、该机如果使用小型稳压电源供电时要求所用电源的输出电流必须足够大,否则,当 1 卡和 2 卡同时工作时,稳压输出电压降低过多,带速变慢,声音产生变调现象。笔者所修理多台此型机中,都是因为稳压电源带负载能力差,放音变调,而一些使用者用提高输出电压的方法来加以弥补(该机额定直流电压为 6V ,有的使用者把电压调到 9V)。造成的后果是烧坏该机的功放集成电路 TBA820M。分析其原因,系稳压电源带负载能力差,当电机工作时消耗电流较大,电源输出电压降低,在电机工作时,把稳压直流输出调到 6V 左右,当电机停止工作时,电压马上回升到 9V 左右,这样,电压瞬间的升高,很容易将功放块烧毁。多台该型机功放集成电路的损坏,都是该原因所造成的。

3、为使修理人员对故障的集成功放块的检测有据可查,笔者在维修实践中总结了功放集成电路的电压电阻数据:

功放集成块 TBA820M 正常电压:

单位:伏

引脚	1	2	3	4	5	6	7	8
IC ₁	0.7	0.55	0	0	3	6	5.6	4.5
IC ₂	0.66	0.6	0	0	3.3	6	5.6	4.5

功放集成块 TBA820M 正常电阻:

单位:KΩ

引脚	1	2	3	4	5	6	7	8
IC ₁	1.1	1	3	0	0.85	0.8	0.9	5
IC ₂	1.1	1	0	0	0.85	0.8	0.9	5

MF10 型万用表, RX100 档黑表笔接地时测得。

微机软盘驱动器“卡盘”的简易修复

乐冬青

“卡盘”(盘片抽不出)是软盘驱动器常见的故障之一。关于它的修复方法一般都是采用松开软盘驱动器浮动压紧轮螺丝,调整压紧轮架的位置,或是拆下压紧轮架,用小刀将浮动压紧轮外圈刮去一层塑料。但用这两种方法修复的软驱,使用效果都不甚好。因为“卡盘”故障不是因压紧轮架位置不正确造成的,而是因浮动压紧轮变形造成的;由于浮动压紧轮是伞片状塑料轮,很不容易刮得均匀,一不小心,就会使压紧轮失圆,造成软磁盘定位不准。笔者对“卡盘”故障进行了较细的分析,发现“卡盘”故障从现象上看,好像是浮动压紧轮变形造成,实际上是浮动压紧轮与内压紧轮卡位不能复位造成。其工作过程如下:当软盘驱动器小门关上时(读、写磁盘信息,浮动压紧轮支架被压杆压下,内压紧轮又压下浮动压紧轮,这时伞片状浮动压紧轮被撑开,同时将软盘片卡压在软驱主轴凹轮内。软驱操作结束,驱动器小门开关打开,浮

动压紧轮支架压杆抬起,内压紧轮借助复位弹簧,(内压紧轮与浮动压紧轮之内的弹簧)弹起,被撑开的浮动压紧轮恢复原状。这时如内压紧轮不能弹起,浮动轮就一直处于撑开状态,于是就出现了“卡盘”,对此,我们采用了一种简便的修复方法。

首先打开机壳,用老虎钳拆下浮动压紧轮轴上的卡片,取出浮动压紧轮,将轴上的宝塔弹簧拿下,再用仪表小起子拆下内压紧轮上的卡片,取出内压紧轮,用锉刀将内压紧轮的外圈锉去一层,然后装回轴上,再将宝塔弹簧、浮动压紧轮装回轴上,卡好卡片就行了。装好后,可检查一下。用手指将内压紧轮压到底,然后松开手,如内压紧轮顺利弹起,说明复位良好,否则,将内压紧轮再锉一层,直至内压紧轮复位自如。

CEC-I 软驱故障维修一例

潘冬

CEC-I 型中华学习机,由于价格适宜、功能较强,已逐步进入家庭。但是由于它的驱动器接口不够完善,如使用不慎易使驱动器出现故障。下面谈谈一例维修经验

一、故障原因及现象

由于安装驱动器(建南 JN-510A 型)时不慎,只插了上面一排插针,造成了驱动器故障。驱动器读软盘操作时正常,但不能进行写操作及格式化磁盘。驱动器在工作时指示灯不亮,当停止工作时指示灯反而发亮。

二、故障的排除

检修时,首先让 CEC-I 主机自检,确定其完好。因了解到驱动器损坏是由于插错位造成的,所以把检修重点放在驱

动器接口周围的电路。打开驱动器的外壳,此电路板设在驱动器的底盘上,板上有几块集成电路和其它元件。驱动器的工作指示灯由 74LS05 控制,用万用表检查 74LS02 各脚的逻辑关系,确认 74LS05 损坏,更换后驱动器工作指示灯故障消除。但仍不能进行写操作,认为是驱动器写保护电路的毛病。检查红外线光电耦合器是否失效。用有写保护的磁盘和无写保护的磁盘分别插入驱动器,测得 74LS05 的输入端 11 脚有变化,而输出端 13 脚无变化,确认 74LS02 损坏。更换后用驱动器测试盘检查一切正常,故障消除。

三、总结

上述故障是由于接口与电缆线的插头错了位,只插了上面一排插针造成的。为此,同行们务必引以为戒。同时在更换集成块时,最好使用集成电路插座,以防不测。

浙江温州市精密电子仪器经营部

地址:温州市飞霞南路龟湖路

邮编:325004 电挂:5072 电话:336072

开户:工商五马办 帐号:1567892

供应(单位) EPROM2716/32, 12 元; 2764/128, 10 元; 27256, 15 元; RAM6116, 15 元; 6264, 25 元; 6117, 10 元; 8031/8085/8155, 15 元; 80C31, 24 元; 8748, 35 元; 8255, 10.5 元; 8035/8039, 11.5 元; D780(Z80CPU), 7.2 元; Z84C00(低功耗型 Z80CPU), 10.8 元; 8243, 12.5 元; 82C43, 16.8 元; 8251, 16.5 元; 82C51, 23 元; 8212/8253, 14 元; 8257, 18 元; 8279, 22.5 元; 光隔 TLP521-4, 7.5 元; TLP521-1/541-1, 2 元; TLP541-2, 3.5 元; 晶振 3.579M, 2 元; 4M, 3.2 元; 6M/8M/10M, 3.6 元; 集成插座 40P, 0.65 元; 以上邮资 2 元/次。进口开关电源成品板, 输入 AC220V/110V 转换; 输出 +5V/7A, -5V/0.11A, +12V/1.7A, -12V/0.4A, +115V/0.4A。外形尺寸(毫米)232×178×35, 价 188 元, 含邮资。另有系列 74LS, CD4000 电路及数百种其它集成电路, 晶体管, 继电器等均为进口器件, 函索价目表者, 请用贴上肆角邮票信封, 写清回邮地址寄我部。

求 援

美国产 TMS9900A 型 16 位家用计算机, 音频信号和视频信号经 RF 调制器后, 电视机(全制式, 要求用 NTSC 制式)能收到载波, 但无图象无声音。希能提供 TMS 产品的有关资料或线索。

通讯地址:长春市东北师大电教系

联系人:刘杰

邮编:130024

中华学习机小蜜蜂 1(XMF-1)内一片门阵列坏, 型号: HKE 610 605 8820 何处有售。

联系地址:河北保定建设路化工部第一胶片厂设计室
孙长令 邮编:071054

“坏”软盘的再生

张 智

有的新软盘,由于厂家装盒不当,造成盘套变形无法使用,只好弃之,但这类软盘可通过以下方法再生。

检查:

将中指与食指插入软盘的驱动圆中转动一周,同时通过读写槽观察盘片有无明显的痕迹,若有痕迹,此盘就无法再生。

复原:

取打印纸两张,把软盘放于上下两纸之中,平整地放在

光滑的桌面上,找一平底瓷盘(底面积要大于软盘面积),内装沸水,将瓷盘表面的水擦净,压在纸上面,两分钟后将软盘翻面,再将瓷盘压在纸上,两分钟后取出软盘,马上放置于16开的厚书之中,二十分钟后取出,此盘已经复原。

将该盘进行格式化,若格式化不成功,可找一张同容量的软盘格式化后,用DISKCOPY命令(格式化的好盘作源盘,再生盘作目标盘)进行盘备份后,再将再生盘格式化,通过以上工作,使手中积累多年的“坏”盘得到了再生。

GW286BH 系统板 维修一例

董纯坚

1、故障现象:自检正常,且能进硬盘,但不认软盘,硬盘软盘时显示“驱动器读错”。

2、分析与排除,该故障属于引导和自举软盘操作系统时失败,这与DMA电路、软盘适配器锁相式数据分离器有关,也与软盘驱动器方面的数据读出电路,磁头定位偏差及磁盘片的好坏有关。首先取已确认是好的磁盘片,插入软盘驱动器中,故障未消除,然后依次检查并替换了软盘驱动器、信号电缆线、软盘适配器,故障仍未消除,看来故障肯定出在系统板的DMA电路上,因为DMA电路有故障,就会导致引导块装入到不正确的RAM区域,引导块程序就不能正确执行,DOS不能自举,致使系统无法进入DOS系统。用示波器检查DMA电路的有关信号,发现CHIPS门阵芯片82C206的66脚ADSTB信号(地址选通信号,高电平有效)虽为高电平,但它所连接的8D锁存器74F573的输入端却为低电平,且复位时,仍无变化,这说明82C206的66脚与74F573的输入端有开路现象,卸下82C206发现该芯片第66脚已断,连接两点,故障消除。

怎样使用清洗盘

瞿新国

清洗盘是用来清洗磁盘驱动器磁头的专用工具。当磁盘驱动器工作时间长了之后或用了质量低下的软盘后,磁头上就会附着磁粉,影响软盘信息的正确存取,甚至根本不能读取。这时,可使用清洗盘清洗,但使用不当反而会缩短磁盘机的寿命。

怎样正确使用清洗盘呢?将清洗液涂在清洗盘上,要尽量均匀,将清洗盘放入驱动器,关上仓门,此时对那些关上仓门磁盘就转动的机器(如一些PC系列机)让它们转30秒即可,对关上仓门磁盘不转的机器(如APPLE II系列机),用启动或调盘命令使驱动器转30秒。一般情况下,磁头业已清洗干净。

注意:不要用不涂清洗液的清洗盘清洗磁头。因为清洗盘较粗糙,不湿润时对磁头磨损较大;另外,磁头上的磁粉附着较强,被清洗液浸润后容易脱落,也易被清洗盘吸附。还有,清洗时间不要过大,否则,同样会磨损磁头。

《计算机辅助设计与图形学学报》正式公开发刊

由中国计算机学会主办的《计算机辅助设计与图形学学报》已被国家科委批准,将于1991年第四季度开始正式向国内外公开发刊。该刊已获得国内统一刊号CN11-2925/TP,国际标准刊号ISSN 1003-9767。这是我国CAD和计算机图形学领域第一个公开出版的学术性刊物。该刊面向全国,刊登CAD与图形学领域里的学术论文,CAD与图形学的科研、开发与应用的最新成果和学术动态,及时反

映该领域学科的发展水平与发展方向。该刊于1989年8月创刊,暂为季刊,已内部出版了9期。此次被批准公开发刊,将会进一步促进CAD和计算机图形学的发展,更好地为我国国民经济建设及计算机事业服务。该刊编辑部设在中国科学院计算技术研究所。

LASER 机 GA004 电路的代换和修复

张保田

近年来我们修复了一些 LASER 机,其中大规模集成电路 GA004 损坏率较高,约占总故障率的一半。但这种芯片是原整机生产厂家专为 LASER 定做的,目前市场上根本无货供应,所以在维修时采用了整块电路代换和局部修复两种方式。

一、用 LS 系列代换 GA004

GA004 是 40 线封装的大规模集成电路,其引线标号及在主机中的连接见本刊九一年一期封三。引线定义和功能如下:

\overline{RD} 由系统总线来的读信号,低电平有效表示 CPU 从数据总线上读 I/O 口或存储器数据。

\overline{WR} 由系统总线来的写信号,低电平有效表示 CPU 向 I/O 口或存储器写数据。

\overline{IO} 实际是一个存储器空间译码信号,当 CPU 读写 6800H 到 6FFFH 空间时为低,此空间在 LASER 上作 IO 用。

\overline{VR} CPU 读写显示存储器时的译码信号,低电平有效,地址是 7000H 到 77FFH。

\overline{VRWR} 其准确逻辑式为 $\overline{VR} + \overline{WR}$,简称为 \overline{VRWR} 为低时 CPU 写 6116, \overline{VR} 低而 \overline{WR} 高时 CPU 读 6116。

D0 到 D7 来自系统总线,三态双向数据线。

KD0 到 KD7 接键盘矩阵,单向数据线,当 \overline{RD} 和 \overline{IO} 同时为低时 CPU 读键盘矩阵,检测用户按键情况。

D'0 到 D'7 与 6116 相连的双向数据线,方向见 \overline{VRWR} 的说明。

Q0 到 Q7 GA004 内部 8D 锁存器的输出端,当 \overline{WR} 和 \overline{IO} 同时为低时 CPU 写 8D 锁存器,各端口作用:

Q0 和 Q5 接蜂鸣器,由软件控制 Q0 和 Q5 总是反相,在两端电平连续翻转时蜂鸣器发声。若使 Q0 和 Q5 同相蜂鸣器就不会发声了。

Q2 磁带记录电平输出端。

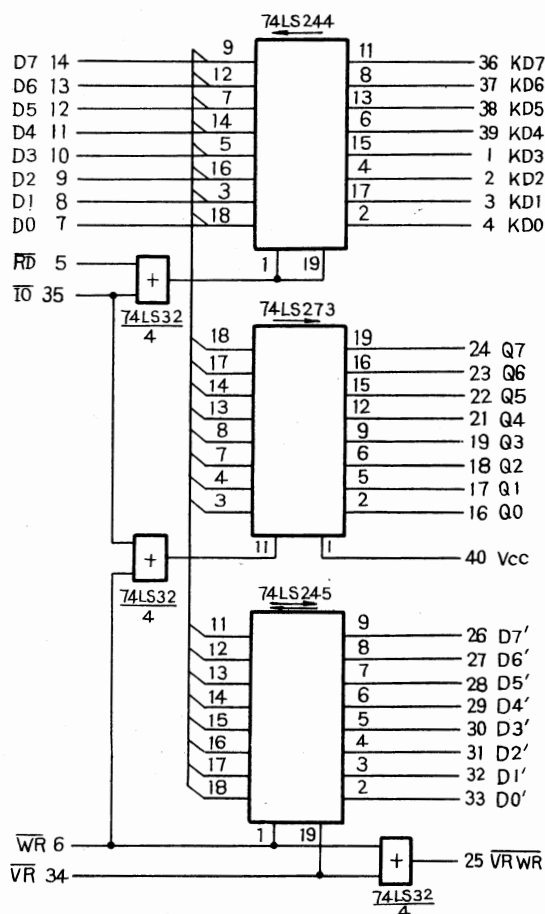
Q4 屏幕底色控制端,为低时绿色屏幕,为高时橙色屏幕。

Q3 屏幕显示状态控制,为低是文本状态,为高时图形状态。

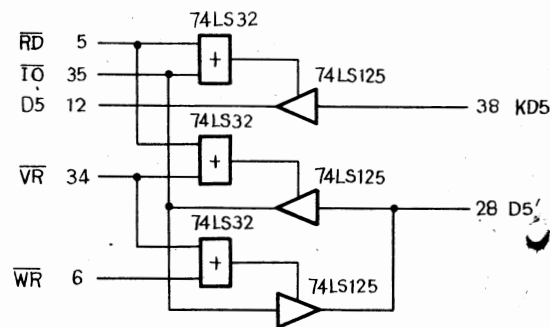
还有三个锁存端主机未用,所以参考电路中没有画出,它们对扩展功能是有用的:

Q1, 17 脚主机硬件电路未接, Q2 系统正是利用它锁定高显状态。

Q6, 23 脚和 Q7, 24 脚,如果扩展系统仅为 305 和 310 机用,此两端可以利用;但如同时要在 200 上用则不宜用 Q6 和 Q7,因为大部分 200 机内部只有 6D 锁存器。



图一 GA004 代换电路



图二 GA004 局部修复电路

根据以上分析可以用四片 74LS 系列中小规模芯片组成 GA004 的等效代换电路,电路见图一,图中大矩形框内是 LS 电路本身的引线编号;而框外是与 GA004 对应的 40 线引线编号。四片 LS 电路装在一块小印刷板上,由印板焊出 40 条引线插入原 GA004 位置使用。

二、局部修复

有时 GA004 只是局部损坏,此时可设法采用较少的器件有针对性地只修复损坏部分。维修中曾遇到一实例:数据线 D5-D'5 和 D5-KD5 都损坏不通,但锁

存器 D5 位可写入。这种情况下如用图一代换电路则成本高功耗大且自制印板并不容易。所以用了 74LS32 和 74LS125 各一片修好 D5-D'5 和 D5-KD5 通道。LS 芯片用万能胶粘在 GA004 上省去制印板的麻烦,电路如图二。74LS125 是四路缓冲驱动器,注意由于其中有两路是输入输出端相互并接的,所以电路一定要反复检查确认连线无误后再开机,否则容易烧毁电路芯片。据用户反映按图二修复的微机已正常工作了半年。

浪潮 0530-AT 死机故障恢复一例

董国钦

故障现象:一台 LC0530 机,在用机过程中,突然掉电,事后开机可自检,但进入记忆联想汉字显示后死机,热启动无效(软盘可启动,使用硬盘)。

故障分析:低于 40MB 的硬盘微机在系统突然掉电后,磁头不能自动退至主保护道内,这时如果搬动机器或磁头正在读写磁盘,磁头很容易划伤磁盘,而造成硬盘的操作系统不能使用,甚至于不能进入、使用硬盘,初步估计为磁盘文件或磁盘损坏,

故障排除:刚开始笔者尝试着利用以下方法来恢复故障,但均无效:

1. 利用 DOS 盘 A>SYS C:传送系统,然后重新装入联想记忆汉字系统,启动无效。

2. 利用 FDISK 软件,删除 DOS 分区,然后重新建立 DOS 分区(注意:如果此时建立的 DOS 分区与原来的 DOS 分区不一致,将导致硬盘信息的丢失)重新启动,故障仍未排除,进行操作系统格式化 A>FORMAT C:/S 后,硬盘可进入使用,但当调用记忆联想汉字系统时,故障又出现。

确定为硬磁盘磁道损坏,需要对硬盘进行低级格式化,利用浪潮机随机测试盘:

A>LC0530 显示菜单:

- 0—SYSTEM CHECKOUT
- 1—FORMAT DISKETTE
- 2—COPY DISKETTE
- 3—PREPARE SYSTEM FOR MOVING
- 4—SETUP
- 9—END DIAGNOSTICS

选择 0 进行系统测试,按下按 Y,再按 0 进行单次测试运行,显示菜单:

1—SYSTEM BOARD

2—1024KB MEMORY

3—KEYBOARD

6—2DISKETTE DRIVE(S) AND ADAPTER

.....

17—1FIXED DISK DRIVE(S) AND ADAPTER

.....

选择 17 进行硬盘处理,显示菜单:

FIXED DISK DIAGNOSTIC MENU

.....

1 — WRITE, READ, COMPARE (ON TEST CYLINDER)

.....

7—FORMAT MENU

9—RETURN TO CONTROL PROGRAM

选择 7,C 进行硬盘格式化,显示菜单:

FORMAT SELECTION MENU

.....

1—CONDITIONAL FORMAT

2—UNCONDITIONAL FORMAT

.....

选择 2,C 对硬盘进行无条件低级格式化,最后利用 FDISK 建立 DOS 分区,进行操作系统高级格式化,至此故障排除。

微机硬盘系统出故障时,不要急于对硬盘进行低级格式化,而要从其它途径想办法恢复,如果确诊是硬盘磁道或扇区损坏,且高级格式化失败,则要进行低级格式化,剔除盘面损坏的块,我们建议用户在搬动机器或关机前,进行磁头复位工作。

一九九一年计算机初级程序员有奖竞赛试题参考答案

(共 150 分)

题 号		小 题 号 与 答 案										得 分		
一	1	AHCIE	2	BD	3	C	4	AC	5	DA	12	40		
	6	D	7	B	8	A	9	C	10	D	5			
	11	B	12	CB	13	B	14	AC	15	AC	8			
	16	AFFID	17	D	18	CE	19	D	20	C	10			
	21	C	22	B	23	A	24	B	25	D	5			
二	1	运算器 控制器		2	匈牙利 存储 存储 串行		3	下一条 当前			8	30		
	4	指令译码存放结果		5	立即 直接 间接		6	定义 通讯			7			
	7	385 3080		8	输出 非击打式 串行式		9	阴极射线管显示器			6			
	10	256 1FF3H		11	纯音 顺序 纯形 多		12	收集 存储 加工			9			
	13	软磁盘									1			
三	1	DIR A: *.PRG									2	12		
	2	MD A:\BAS\BA1									2			
	3	COPY CON A:BA1\AA.TXT									2			
	4	COPY A:\BAS\BA1\AA.TXT+A:\DB3\AA.BAK A:BAS\BA2\ALL.BAK									2			
	5	REN A:BA2\ALL.BAK BA2\ALL.TXT									2			
	6	DEL A:BA1\AA.TXT 或 ERASE A:BA1\AA.TXT									2			
四		COPY CON B:PC.BAT TREE B:/F CLS TYPE B:PC.BAT PAUSE 姓名 单位 TYPE B:PC.BAT COPY A:TREE.COM B: ^ Z									8			
五	1	① REPLACE ALL 平均分 WITH(语文+数学)/2									1	20		
		② LOCATE FOR 平均分>90									1			
		③ COPY TO BBB FOR SUBSTR(姓名,1,2)= '王'									1			
		④ INDEX ON 性别+STR(平均分,6,2) TO CCC									1			
		⑤ LIST FOR 民族<> '汉' FIELDS 姓名 OFF									1			
		⑥ COPY STRUCTURE TO DDD FIELDS 姓名,性别,平均分									1			
		⑦ LIST FOR 出生日期 ≤CTOD('12/31/65')									1			
		⑧ COPY TO EEE SDF									1			
	2	1 2 1 2 4 1 8 9									4			
六	②	2 212 21012 212 2		③	21 21		④	5 8 11 14 17			9	12		
	①	1 3 6 10 15 21 28 36 45 55									3			
七		INT ((I-2)/6) INT((I-3)/7) S+I N+1									6			
八	①	A(I,2)*A(I,3) I+1 A(I,4)>A(J,4) J,K 6 T									6	12		
	②	流程图(略)									6			
九	1	D	2	H	3	I	4	E	5	J	5	10		
	6	G	7	C	8	B	9	F	10	A	5			

关于召开北京地区“PC 机个人用户软件 交流联谊会”的通知

兹定于一九九一年十二月二十七~二十八日两天在京召开北京地区“普及型 PC 机个人用户软件交流联谊会”。凡北京地区参加《普及型 PC 机个人用户软件交流联谊会》的用户均可报名参加。外地用户待这次活动取得经验后再举办。联谊会的目的是总结一年来联谊活动情况,交流应用经验,交流软件与资料,推动软件交流联谊活动进一步开展。

报到日期:1991 年 12 月 26 日上午 8 时至下午 4 时

报到地点:中国农科院计算中心培训部

会议地点:中国农科院计算中心

报名费(含资料费):20 元

凡愿参加会议的北京地区用户,普及型 PC 机销售单位,请于 1991 年 12 月 10 日前务必将回执寄到北京市海淀区白石桥路 30 号中国农科院计算中心培训部王路敬收。

会议联系地点:中国农科院计算中心培训部

联系人及电话:王路敬 896531-241

回 执

姓 名	参加交流资料	希望从交流中得到软件

中国农科院计算中心培训部
《电子与电脑》编辑部

事业无穷年 奋力谱新篇 ——北京市单片机应用技术协会动态

协会通讯员

北京市单片机应用技术协会成立以来,在推广应用单片机方面开展了一系列的技术交流活动。如介绍国内外技术发展趋势;推广新产品;组织汽车电子、医疗电子等专业应用交流;举办单片机应用成果交易市场;开办从基础到 BASIC—S2,智能仪表,工业控制、8098 单片机,PL/M 语言等不同层次培训班;以及开发各类单片机仿真器、应用板等等,由于它活动经常、内容丰富、已成为广大专业技术人员的活动中心。

今年“协会”根据进一步推广普及单片机应用的需要,除继续做好上述各项工作外,还将重点抓好以下工作:

一、坚持经常性的技术咨询服务,诚心诚意地为各行各业应用单片机出谋划策,排忧解难。除“走出去”为各行各业服务外,六月份起每周五上午定为免费义务咨询接待时间,以协助解决开发应用中遇到的各种问题。

二、积极组织专题研讨活动,当前已计划列入专题的有:网络、总线方面如 BITBUS 的应用;单片机中断及抗干扰能力;单片机的数字实验;以及采用 DL/M 语言编辑后的仿真等问题。

三、进一步开展应用成果的交流。随着各类 8098 开发器的相继推出。8098 的应用自然应作为工作重点。三季度“协会”将在广泛征集的基础上,组织一定规模的应用成果交流活动,热忱欢迎各在京单位的应用成果参加,共同努力把

单片机的开发应用推向新阶段。

〔注:凡对上述活动有兴趣者,请直接与北京虎坊路 13 号“协会”联系。〕

电子工业出版社新书介绍

《电子测量仪器选购指南》定价 35.00 元

《最新简明中外集成电路互换型号手册》定价 18.00 元

《家用电器元器件手册》(精装) 定价 56.00 元

《传感器敏感元器件大全》(精装) 定价 64.00 元

《标准集成电路数据手册——TTL 集成电路》(精装)定价 31.50 元

《标准集成电路数据手册——运算放大器》(精装)定价 32.00 元

《国内外功率晶体管实用手册》上(精装)定价 21.00 元

《国内外功率晶体管实用手册》下(精装)定价 45.00 元

《dBASE 应用解疑》定价 5.00 元

《IBM PC 编程指南》定价 7.00 元

《DOS/BIOS 使用详解》定价 8.00 元

《MS-DOS 高水平程序设计》定价 5.50 元

邮购办法:邮局汇款 北京万寿路电子工业出版社发行部邮购科

银行汇款 开户行北京市工商银行翠微路分
理处帐号 661036-40

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

本部电话:81.3693 电挂:3101 邮政编码:100036

浙江省宁海县源源电子屋

地址：浙江省宁海县城关水角凌路 93 号

邮编 315600

邮购以下产品

(一) YD—2 小型紫外线治疗消毒仪：在开发单片、单板机等使用 EPROM 芯片的应用场合，紫外线擦除器是不可缺少的设备，目前市场上的擦除器存在功能单一，擦除时间长（一般为 15 分钟）等缺点。为此，本屋隆重推出手持式小型紫外线治疗消毒仪，它集杀菌消毒，皮肤病治疗，EPROM 擦除等功能为一体。经上海医科大学杀菌试验，临床应用，对近二十种病菌在 10 秒之内，杀菌力为 100%。并对各种皮肤创伤、溃烂、外科感染等有明显治疗效果。EPROM 可在 3 分钟内完全擦除。该仪器体积小，使用方便，欢迎广大用户选购。推广期间优惠供应，每台价 245 元，免收邮费，保修 1 年。

(二) 电视录像放大器：该机体积小、外形美观，使用时只要将电视天线、录像或游戏机输出线插入该机，即可通过本机开关选择所需节目。可省去收看电视、录像、游戏节目时经常插拔插头的麻烦。本机内设二级低噪声放大器，使用二节五号电池对改善节目中的重影、雪花等有明显效果。邮购价 18.20 元，邮费 1.50 元。

(三) 各种电子元器件：二极管 1N4001 0.1 元/台，1N4007 0.13 元/只，1N4148 0.09 元/只，三极管 9012, 9013, 9014, 9015 0.22 元/只，8050 5551 5401 0.25 元/只，3DG130B(绿) 1.30 元/只，3A×31B(绿点) 0.30 元/只，发光管， $\phi 3, 2 \times 5$ 红 0.23 元/只， $2 \times 5, \phi 3$ 绿，0.24 元/只， $\phi 5$ 红 0.26 元/只， $\phi 5$ 绿 0.27 元/只，高亮度 $\phi 3$ 红绿 0.29 元/只， $\phi 5$ 红绿 0.31 元/只，0.5" 数码管共阴，共阳(请注明) 4.20 元/只，集成电路，LA4160 3.90 元/块，TA7668 3.20 元/只，TA7240 9.20 元/只，TA7640 2.80 元/块，LB1405(电平指示) 1.90 元/块，74L500 国 1.40 元/块，273, 373, 244, 2.50 元/块，以上每块邮费 0.80 元，另有电阻 RTX1/8W 系列，RJ1/4 系列，电容 CD11 系列，CL 系列，CC 系列等，不能尽录，备有清单，欢迎索取，清付邮资 0.30 元。

本屋经营计算机应用系统，软件及单片、单板工业控制系统，意同有此专业特长和志趣者结为朋友，欢迎联系。本屋为您发挥技术特长和创造才能提供广阔天地，对提供技术、货源、信息等有效益者，则给予优厚报酬。欢迎合作，欢迎指导。联系人：尤思胜。

天坛学习机教学网系统、SEN—I 型 简易教学网络系统简介

本刊通信员

由北京计算机五厂研制开发的天坛 TEC—NET 教学网络系统和 SEN—1 型计算机简易教学网络近日通过市电子办鉴定。

TEC—NET 是专为天坛学习机而研制的教学网络系统，该网络系统采用 CSMA/CD 访问控制方式和总线拓扑结构，是一种分布式串行网络。网络传输距离 > 200 米；传输速度 64Kbps；网络工作站最多可联接 64 个；可实现资源共享：即各站共享教师机配备的软盘驱动器和打印机；文件可按点对点 and 广播两种传输，可实现网络各站信息的汉字传输。

SEN—1 型计算机教学网络可由教师机向学生机统一或单机传递屏幕、内存或磁盘中的程度、数据等信息；该网没有排异性，能在该网络系统的设备上建立其他网络，充分使用现有设备，可实现资源共享；教师机可强行中断学生机的工作而统一显示教师机的内容，但并不破坏各学习机内存中的数据；任何一个终端可向其它终端统一或单机发送内存中的程序、数据或屏幕信息。APPLE 机、中华学习机、天坛学习机等同时进网。

以上两种网络系统结构简单合理、功能完善、操作简便、易于维护、且充分考虑中小学教学的特点。教学网络的研制成功可有效的组织教学，对普及计算机教育提供了可靠的设施。

遗 补

本刊第 9 期《91 年第三届国际信息学奥林匹克 (IOI'91) 试题》中，由于工作疏忽，拼版时将原图遗漏，谨向读者致歉。

	1	2	3	4	5
1					
2		1			*
3					
4				*	
5		*			