

計算機應用研究

1990

3

APPLICATION RESEARCH OF COMPUTERS 《計算機應用研究》雜誌社





北京四通新技术产业股份有限公司

计算机事业部

隆重推出

● 万能编程测试卡

无所不能
开发必备
十位一体

● MICE-II 微机在线仿真器

硬件侦错
系统调试
逻辑分析



MICE-II

1. EPROM

2. EEPROM

3. 8751 系列

4. 8748 系列

5. Z86 系列

6. BPROM(PROM)

7. 逻辑电路测试

8. RAM 测试

9. PALS GAL 编程

10. PEEL/ PLD 编程

仿真CPU: 8085 8086

8088 Z80 等

是开发调试微电脑的必要
工具。

四川省电子计算机应用研究中心简介

四川省电子计算机应用研究中心(四川省电子技术研究所)是在四川省科委直接领导下,面向全省各行业,从事计算机及电子技术应用研究的开发型科研机构。

该中心位于成都市人民南路四段,现有职工 254 人,其中技术人员 180 人,含高级工程师 22 人,工程师 92 人,拥有从日本引进的日立 M-240D 大型计算机系统一套,微型计算机数十台及多种科研设备。中心坚持技术立所,在计算机过程控制、文字图形处理、CAD、管理信息系统、医用电子设备、智能化仪器、电子乐器、工程力学计算等方面进行了大量卓有成效的研究工作,获省科技进步二等奖一项,三等奖七项,其中微机运动系统、铁路大站信号检测系统、电视形象创作系统、脑电波地图仪、电能综合测试仪、电脑流量计、数字电子琴等项成果均在全国范围内推广,赢得广大用户的高度赞誉和一致好评。

中心自一九八五年起实行所长负责制,全部取消了财政统定事业费,实现了经济自立。一九八五年至一九八九年,横向技术性收入 1226.2 万元,逐步形成自我约束、自我发展、自我完善的运行机制。一九八八年度荣获四川省先进科研单位一等奖。

中心将继续立足信息技术领域,坚持技术立所,搞好高科技产品的科研、开发、生产和经营服务,坚定不移地走发展高技术产业的道路,逐步形成以市场为导向,以高科技产品为龙头,集科研、开发、生产和经营服务于一体的经济实体。

现将该中心的业务范围列述如下:

- * M-240D 大型机算题服务、项目开发服务以及微型机机时服务。
- * 承接国民经济各行各业各种类型微机应用项目的研制开发及其课题任务。
- * 研制开发各种高科技智能化、自动化仪器仪表。
- * 承接电子技术应用项目的开发及设备研制。
- * 电子、电脑整机及之器件销售。
- * 研制开发并转让各类计算机系统软件及应用软件。
- * 开展电子、电脑技术咨询、培训、展示服务。
- * 开展有关电子、电脑技术的信息交流及其学术活动。
- * 开展有关电子、电脑的科、工、贸方面的国内、国际横向联合。
- * 与兄弟单位联合出版《计算机应用研究》公开发行的科技双月刊,并开展广告、技术资料、科普等有关服务。

通讯地址:成都市人民南路 4 段 11 号附 1 号

电 话:581946 554969 554849

电报挂号:成都 5938

电 传:60101CDTSO CN SRCCA

开户银行:成都工商行跳伞塔分理处

账 号:89501299

《计算机应用研究》杂志办刊单位

四川省电子计算机应用研究中心

贵州省科学技术电子计算机中心

安徽省计算中心

新疆电子计算中心

吉林省计算中心

青海省测试计算中心

甘肃省计算中心

四川省电子学会

《计算机应用研究》杂志编辑委员会

主任委员：张执谦

副主任委员：李泽民

委员：余凯 张国栋 贾洪钧

曾光初 王小华 朱景生

《计算机应用研究》杂志社董事会

董事长：周赛渝

董事：唐珍 郑国基 陆慰椿

秦小竹 龚宇清 黎瑰常

计算机应用研究（双月刊）

（公开发行）

一九九〇年

第七卷 第三期（总第35期）

主 编：张执谦

副 主 编：李泽民

本期责任编辑：齐墨之

编 辑 出 版：《计算机应用研究》杂志社

内 文 印 刷：西南冶金地质印刷厂

封 面 印 刷：四川省印刷制版中心

出 版 日 期：1990年5月30日

本刊通讯地址：成都市人民南路4段11号附1号

本刊邮政编码：610015

订 阅 处：全国各地邮局

总 发 行：成都市邮政局

每册定价：1.30元

本刊邮发代号：62—68

国内统一刊号：CN51—1196

广告经营许可证：川蓉工商广字 005 号

一个简单的标准文稿格式打印程序

西北工业大学 周 明

一、问题的提出

在一些文书工作中,有时候需要将文章誊抄在方格稿纸中,其中每个汉字、英文字母、标点符号等均应占一格。手工抄写容易出错,并且速度很慢,甚至还由于字迹潦草而使得其他人错误地理解文章的意思。AR3240打印机以及四通公司的MS-2401打字机具有按标准文稿格式打印文章的功能,但一般普通的打印机却大都无此功能。

这里,我们利用软件来实现这个功能,即在打印机上输出标准的“20×20”的方格稿纸,并同时在其中按指定的行文格式打印出文章的内容。这个程序使得我们不用再在硬件上投资,就可又增加打印机的一个功能。

二、解决问题的方法

解决这个问题的方案是:先利用中文文字编辑程序WORDSTAR输入所要打印文章的文稿,然后再使用由BASICA语言所编写的文稿打印程序来打印出这个文稿。WORDSTAR在对文稿进行自动编辑和排版时,在磁盘文件中插入了一些不可显示的控制字符;BASICA程序在打印该文稿时则需要对这些控制字符进行检测和处理。故在该程序的实现中需考虑如下几个方面的问题。

1. 压缩打印行距

在标准的方格稿纸中,每两行文字之间有一狭长的细行。我们可以使用对应的两行制表符来

造出这个细行。这时若不压缩打印行距的话,则这样打印出的一页文稿的长度将超过一页标准稿纸的长度。故我们在打印两行文字之间的细行时,需压缩打印行距;而在打印下一行文字之前,要恢复为以前的打印行距。

2. 对继续行控制符的处理

用中文WORDSTAR所产生的文件,若一行文字有继续行的话,则它自动在继续行处增加一个二字节的控制符“8D 0A H”。程序读进这两个字节后,应丢掉它,然后继续去读后面的两个字节,并接着往下打印。

3. 对回车控制符的处理

回车是文本文件中一行结束的标志,其控制符为“0D 0A H”。程序读进这两个字节后,应在稿纸中补满当前打印行所剩余的空格子,然后将打印头换到稿纸的下一行。

4. 对文件结束符的处理

打印时遇到文件结束符之后,应先在稿纸中补满当前打印行所剩余的空格子,然后再补满该页稿纸中所剩余的行及其空格子。

三、程序清单及其说明

程序清单如图1。

要注意的是,上面给出的这个程序每次从文本文件中读取的是2个字节(即一个汉字)。但它同样也能处理数字、字母、空格等其他一些符号。这里的要求是,它们的长度必须是偶数;若不是偶数的话,可以通过在文本文件中增补一个空格来使其成为偶数。

```

100 WIDTH "LPT1:",90
110 X1$=""
120 X2$=""
130 X3$=""
140 X4$=""
150 P=0
160 INPUT "请输入要打印的文件名称: ",NM$
170 OPEN NM$ FOR INPUT AS #1
180 WHILE NOT EOF(1)
190 LPRINT TAB(26),"中国设备管理培训中心论文专用稿纸"

```

灵活的文件打印程序

兰州大学计算机系 刘建福

一、引言

近年来,人们在汉字的输入/出系统的开发中作了大量的具体工作,特别是使打印机的输出基本上可以完成手工的大部分编排工作。但是每种开发都程度不同地解决了当时所需的打印要求,一般地不具有普适性,如WS的打印功能就是一例。

为了设计一个好的打印程序我们先来看看文本打印程序应当具有什么样的要求。1)简明地将打印文本的模式以廉价的方式供给用户便于用户以适当的方式对文件进行调整。2)按使用者的要求等待用户装入各种稿纸。3)使用者可以随时选择任何一种所需打印的文件。4)按使用者的要求控制打印格式如:行数、行距以及是否要求分两部分进行。5)自由地选取打印的起始页并用一定的特征标志相区别。6)文本应能打印成多种字体的形式。7)在汉字和图形共存的文本中应在输出的行距有差别。8)一个文本文件打印完后其空白部分是否需打印下一个文件。9)科技文献中的角

标输出。10)特殊情况下要求竖排汉字等。

这里给出的程序基本上具有上述大部分功能,它是在1724打印机上得到实现。其程序还可以进行适当扩充。

二、程序设计思想

本程序将屏幕做为一种特殊的打印机,用户想要看自己所设计格式是否满意可通过先要在终端显示合乎要求时再正式打印输出,否则调整后再输出。在打印完一页后打印机等待某个继续打印的命令以便于用户控制装纸过程。另外,通过对话方式让用户选择每页行数以及是否需要页号、起始页号可以是任一页为防止选择失误特此设置特征字以示区别。打印中的字体控制可以在文本文件中用 `ctr-p ESC] <` 控制字 附在相应段前。打印时可以以这个字开始按控制字型号打印直到下一个控制字出现为止。在汉字与图形共存的文本中,图行要求行距几乎为零而汉字较宽为此在图形的首尾附加.x和.y。最后,可以通过简单的调用下一个文本文件的内容就可以将两

```

200 LPRINT X1$+" "&16"
210 LPRINT X2$+" "&19"
220 FOR I=1 TO 20
230 LPRINT " | ";
240 FOR J=1 TO 20
250 IF EOF(1) THEN FOR K=J TO 19:LPRINT " | ";NEXT K: LPRINT " | ":GOTO 300
260 A$=INPUT$(2,1): IF ASC(A$)=141 THEN GOTO 260
270 IF ASC(A$)=13 THEN FOR K=J TO 19:LPRINT " | ";NEXT K: LPRINT " | ":GOTO 300
280 IF J<>20 THEN LPRINT A$+" | "; ELSE LPRINT A$+" | "
290 NEXT J
300 IF I=20 THEN 340
310 LPRINT X3$+" "&16"
320 LPRINT X2$+" "&19"
330 NEXT I
340 LPRINT X4$
350 P=P+1
360 LPRINT TAB(7);"20×20=400"; TAB(63);"第 "+STR$(P);TAB(75);"页"
370 FOR I=1 TO 10: LPRINT: NEXT I
380 WEND
390 CLOSE #1
400 END

```

图1 程序清单

个文件连接起来打印到同一页。整个程序采用对话方式实现用户对打印格式控制。

三、程序清单

```

L30H0210 REM 一个多功能打印程序
20 REM 程序的说明部分
30 PRINT TAB(3); "----使用指南----"
40 PRINT
50 PRINT TAB(2); "1. 选择输出方式"
60 PRINT TAB(2); "2. 设置打印格式"
70 PRINT TAB(2); "3. 选择起始页号"
80 PRINT TAB(2); "4. 多字体打印(CTR-P ESC I(控制字))"
90 PRINT TAB(2); "5. 紧凑打印(X(紧凑体).Y"
100 PRINT TAB(2); "6. 续页打印和分半打印"
110 CLS
120 INPUT "向打印机输出[Y]:", YP$: IF YP$="N" OR YP$="n" THEN FS$="SCRN:"
    ELSE FS$="LPT1:" : YV$="1"
130 IF FS$="SCRN:" THEN XX$="屏幕": TSXX$="按回车键继续..."
    ELSE XX$="打印机": TSXX$="调整并装纸, 按回车键继续..."
140 LOCATE 2,20: PRINT XX$
150 OPEN FS$ FOR OUTPUT AS #2
160 WIDTH "LPT1:", 255
170 JL=30
180 WIDTH "LPT1:", 254
190 ES="T"
200 LOCATE 2,1
210 INPUT "要打印的文件名:", FILE$
220 INPUT "每页打印的行数[37]:", P: IF P=0 THEN P=37
230 INPUT "打印页号[Y]:", YN$: IF YN$="N" OR YN$="n" THEN YN$="N" ELSE YN$="Y"
240 INPUT "是否分两半打印[Y]:", PART$: IF PART$="N" OR PART$="n" THEN PART$="N"
    ELSE PART$="Y"
250 IF PART$="Y" THEN GOTO 260 ELSE INPUT "左边留多少空格[0]:", SP: SP=SPACES(SP):
GOTO 290
260 INPUT "当前打印的是左半部[Y]:", LH$: IF LH$="N" OR LH$="n" THEN LH$="N" ELSE LH$="Y"
270 IF LH$="Y" THEN INPUT "左边留多少空格[0]:", SP: SP=SPACES(SP)
    ELSE INPUT "左边留多少空格[37]:", SP: IF P=0 THEN SP=37: SP=SPACES(S
P)
290 INPUT "从第几页开始[1]:", F: IF F=0 THEN F=1
300 OPEN FILE$ FOR INPUT AS #1
310 PA=F-1
320 IF F=1 THEN 490
330 INPUT "本页特征词:", CHAC$
340 IF LEFT$(CHAC$,1)="$" THEN CHAC$=CHR$(27)+MID$(CHAC$,2)
350 PRINT "打印字型(A_I):";
360 ZX$=INKEY$: IF ZX$="" THEN 360
370 TY=ASC(ZX$) AND &HDF
380 IF ZX(&H41 OR TY)ASC("T") THEN BEEP: GOTO 360

```

```

390 PRINT ZX$
400 PRINT #2, CHR$(27); "I"; CHR$(ZX)
410 FOR I=1 TO (F-2)*P-1: LINE INPUT #1, AS: NEXT
420 LOCATE 1, 76-LEN(CHACS): PRINT "I";
430 LOCATE 2, 76-LEN(CHACS): PRINT "L";: FOR I0=1 TO INT(LEN(CHACS)/2)+2: PRINT "-";:
NEXT
440 LINE INPUT #1, AS
450 TZCS=LEFT$(AS, LEN(CHACS))
460 LOCATE 1, 79-LEN(CHACS): PRINT TZCS
470 IF TZCS=CHACS THEN CLS: GOTO 650
480 GOTO 440
490 P0=0

500 INPUT "是否要继续(Y):", XUS: IF XUS="N" OR XUS="n" THEN XUS="N" ELSE XUS="Y"
510 IF EOF(1) THEN IF XUS="N" THEN FOR I=P0 TO (P-1)*30 STEP 30: PRINT #2, :PRINT
#2,
CHR$(27)+@L30;: NEXT: IF YNS="Y" THEN PRINT #2, TAB(35); SS; "--": PA+1, "--": CL
OSE: END
520 IF XUS="N" THEN GOTO 560
530 INPUT "续页文件名:", FILENS
540 OPEN FILENS FOR INPUT AS #1
550 GOTO 410
560 P=P0+JL
570 IF P0<P*30 THEN 630

580 IF PARTS="Y" AND LHS="N" THEN P0=0: PA=PA+1 ELSE IF PARTS="N" THEN P0=0: PA=PA+1

590 IF PARTS="Y" AND LHS="N" OR PARTS="N" THEN PRINT #2, :PRINT #2, IF YNS="Y" THEN
PRINT #2, TAB(20); SS; "="; PA; "="; Z0="Y"

600 IF FSS="LPT1:" AND YYS="1" THEN LOCATE 9, 1
610 PRINT CHR$(8); CHR$(7); CHR$(7); TSX$: INPUT "", X$: CLS
620 IF YYS="1" THEN LOCATE 9, 1: PRINT SPACES(60)
630 IF Z$="T" THEN AS; "": GOTO 480
640 LINE INPUT #1, AS
650 D$=LEFT$(AS, 2)

660 IF D$="."X" THEN E$="F": PRINT #2; CHR$(27)+@L20; CHR$(27); "01";: JL=20 GOTO 630
670 IF D$="."Y" OR D$="."y" THEN E$="T": PRINT #2; CHR$(27)+@L0; CHR$(27)+@H02"
CHR$(27); "02";: JL=30: GOTO 630

680 PRINT #2, SS; AS
690 WS=INKEY$: IF LEN(WS)=1 THEN IF ASC(WS)=27 THEN END
700 GOTO 510
710 IF P0<90 THEN Z$="F" ELSE GOTO 510
720 IF ERR=62 THEN CLS: LOCATE 5, 10: PRINT "本号证号!"
73 CLOSE: END

```


一个实用的方格纸格式文稿打印程序

安徽省芜湖市纺织器材厂微机室 徐雪斌

目前文稿的格式打印几乎成了电脑打字机的“专利”，微机上还不享有该软件产品，实在是令人遗憾。笔者曾使用过四通电脑打字机，并对其格式打印产生了兴趣，因此想到了编写一个程序在微机上实现格式打印。（程序附后）。该程序的主要功能是对经EDLIN、WordStar或HW等软件编辑的ASCII码文稿（尤其是信件和稿件）进行格式打印，既可以设置每行的字数即行宽，也可以选择页长即每页的行数，在灵活性方面比四通等电脑打字机的格式打印要强得多，若微机上配置了多种字体、多种字号的汉字处理系统（如：CC-DOS2.13D），则格式打印的效果不亚于一般的电脑打字机。

程序是用Turbo Pascal语言编写的，是在CC-DOS系统下用西文Turbo Pascal 4.0版本编辑、调试和运行的，本程序已在长城0520CH、PC/XT、286等微机上运行通过，使用效果良好。编译方法很简单，只要键入TPC文件名←（可不打入.PAS的后缀）即可，若源程序无误，则生成一个带.EXE后缀的可执行文件。

程序虽是用Turbo Pascal 4.0编写，但可很方便在其它版本或非Turbo系列的Pascal上运行通过，因为程序很少涉及Turbo Pascal 4.0环境下所特有的功能。

下面对程序的设计思想和算法作一简单的介绍，以便读者为其增加更多的功能或改写成其它的语言程序。

本程序在设计时，主要考虑到要解决两个方面的问题：一是程序对任一文本文件应能自动将文件中每个半角字符作为全角汉字处理，二是程序对处理后的串变量应能进行自动格式输出。第一个问题的关键在于全角汉字和半角字符（包括空格）的识别和处理。解决这个问题的主要语句是IF Line[i] < chr(160) THEN... ELSE..。因为任一汉字都可由 chr(160+n)+chr(160

+m)组成（其中 $0 < n \leq 86$, $1 < m \leq 94$ ），即两个ASCII组成一个汉字，而常用字符的ASCII都小于 chr(160)，因此程序可将每个半角字符作为全角汉字处理。第二个问题的关键是一些算法。

程序采用的是结构化设计方法，将各主要功能分成不同的模块，分别编写成过程子程序，再由主程序或子程序调用完成不同功能的过程子程序，以达到设计要求。

程序的算法如下：

程序第2行语句 uses printer 主要是为将输出送打印机服务的，程序中所出现的 write(lst, ...) 和 writeln(lst, ...) 都与第2行语句有关，若Lst用其它变量来替换并增加相应的一些语句，则程序运行结果既可以送打印机，也可以作为文件输出到磁盘上，这样程序的灵活性将更大，读者不妨试一试。

程序从第20行开始，一直到第136行都是一些能解决不同问题的功能模块，主程序是137行到148行，虽与程序首部的长度相差无几，但起的是统管全局的作用。

过程PrLeftBlanks(20~24行)的功能是打印格式文稿左边的空格（包括装订线）；

过程Pr-col(25~28行)的功能是打印每行字符的上封线或下封线；

过程PrUpOrDn(29~32行)是打印每页的上封线或下封线；

过程PrUpAndDn(33~46行)是页末处理，有打印页尺寸（即每页的字数）和页号；

过程JoinTable(47~55行)其功能是表格线的连接，它也是一个重要部件，当一行字符打印完毕以后，由它担任继续画线的任务，然后打印下一行的表格线；

过程ReadFileAndPretreatment(56~78行)正像程序名那样，其功能是读文件和预处理。

理,即程序初始化;

过程TextTreatment(79 \approx 100行)是对已编辑的文稿进行处理,其功能是将每次由串变量Line读到的半角字符经处理后送字符数组变量str1中,在每两个相邻变量里依次存放的是以二个半角为计算单位(非汉字字符后面增加了空格)的全角字符,然后再对空格进行折半处理,以保证与原稿件的空格数相符;

过程PrintTreatment(101 \approx 126行)是打印的核心部分,它包括了所有打印功能的调用,换句话说,格式打印是由它来控制的,尤其是111 \approx 125行,可以说没有这几行语句的“通力合作”,打印将告失败,读者若有兴趣的话可认真读一读该段程序;

最后一个过程EndTreatment主要是解决当源文件结束后,若格式未完则将继续打印,直到该页印完为止。主程序(137 \approx 148行),它是将各个功能模块有机地组合起来去完成设计要求。

本程序是在纯西文的Turbo Pascal 4.0下编辑的,那么其中的汉字字符串是如何编辑的呢?最简单的方法是借助于其它中西文编辑软件,比如在CC-DOS下用EDLIN来编辑程序的汉字或汉字字符。

程序使用说明:

程序运行时,屏幕将提示:

--**--本程序为您提供格式文稿

打印功能--**--

请输入需打印的文件名(例: c: file1.txt):

这时只要打入文本文件名(可包括盘符、路径和扩展名),程序若找到该文件,则进一步提示:

请输入行宽及页长(字数和行数)(例: 20 20):

此时可输入两个相应的参数(两参数间用空格隔开),然后准备好打印机再按一次回车键即可打印。每次换页时,程序将处于等待状态,换页完毕再按回车键继续打印,直至印完文稿。

在使用该程序前,最好先对源文稿作一次必要的清理,比如删除源稿中的CC-DOS2.13A或D等的一些控制码或HW、WS自动排版的控制码等,以保证打印效果,对于CC-DOS2.13A或D等的一些控制码,程序除“吃掉”所有的“/”字符(因它有可能导致程序运行失败)外,其它控制码(即字符)将原样照印。另外程序对换行是这样处理的:当文稿的第二行以后的行未若是一个句号或第二行以后的某行前面有空格,则在格式打印时引起一个换行,否则将接在上一行继续打印。若读者认为行未若是一个分号或冒号也应换行的话,则只要在程序109 \approx 110语句中增加分号或冒号的判断(注意应将分号或冒号折成两个部分并用AND连接,请参考语句中的句号折分)并在本程序的第20行前对相应的变量作一说明即可。

下面谈一点程序的不足和修改建议:

考虑到程序的通用性,因此对格式文稿打印时的行距、列距及字体、字号都未作设定,这就要求使用者在打印前设定,对于CC-DOS2.13A或D等,使用<Ctrl>+<F10>键即可设置字体、字号和行距,但对于有的系统则必须用其它方式来设定(比如,对于无多种字体的CC-DOS 2.0 \approx 4.0系统,可在BASIC语言或dBASE II下设定后再打印),否则打印的表格线可能是虚线。当然,读者也可为本程序增加可选字体、字号和行距等功能;其次,由于笔者对由HW、WordStar编辑,又享受了其自动排版“好处”的这类文稿,本程序对其格式打印的效果不太理想,主要问题是在调整空格的控制码出现的地方会打印出怪字符(非经自动排版的文稿不会有此现象),对于这类文稿,可在EDLIN等方式下将控制码删除后再使用本程序,但最好是在本程序中解决这一问题;受稿件长度的限制,本程序未对显示屏作必要的处理,也未对读文件的出错进行处理。建议读者在本程序的基础上根据自己的实际需要增加一些实用功能。

```

1:*  PROGRAM TextFileOfFormPrint;
2:  uses printer;
3:  const
4:    blank=' ';chr0='.';chr1='|';chr2='┐';chr3='—';chr4='└';
5:    chr5='┌';chr6='└';chr7='┐';chr8='┌';chr9='+';chr10='x';
6:  type
7:    str=array [1..500] of char;blanks=array [1..4] of string[2];
8:    wid1=array [1..128] of string[2]; wid2=array [1..132] of string[2];
9:  var
10:    FilVar : Text;
11:    FileName : string[14];
12:    Line : string[255];
13:    blanks1,blanks2 : blanks;
14:    ch_width1, ch_width2 : wid1;
15:    ch_width3, ch_width4 : wid2;
16:    I,J,K,K1,k2,k3,k4,k5,k6 : Integer;
17:    Width,Length,Page,Pg,LeftB : Integer;
18:    str1 : str;
19:    Bool1 : boolean; xx,yy : char; fullstop : string[2];
20:  Procedure PrLeftBlanks(page:Integer);
21:    var i: Integer;
22:    begin if ((page=LeftB) or (page=length-LeftB-1)) then
23:      for i:=1 to 4 do write(lst,blanks2[i])
24:      else for i:=1 to 4 do write(lst,blanks1[i]); end;
25:  Procedure pr_col(ch:wid1);
26:    var i: Integer;
27:    begin writeIn(lst);prleftblanks(page);write(lst,chr1);
28:      for i:=1 to width do write(lst,ch[i]); write(lst,chr1); end;
29:  Procedure PrUpOrDn(ch:wid2);
30:    var i: Integer;
31:    begin WriteIn(lst);prleftblanks(page);
32:      for i:=1 to width+2 do write(lst,ch[i]); end;
33:  Procedure prUpAndDn(eof_end:boolean;var page,pg:Integer);
34:    var i: Integer; j1,j2: real;
35:    begin pg:=pg+1; PrUpOrDn(ch_width4);writeIn(lst);
36:      for i:=1 to 10 do write(lst,blank);
37:      j1:=(width-1)/2;j2:=j1*length;write(lst,j1:2:0,chr10,length,'-');
38:      if j2<100 then write(lst,j2:2:0)
39:      else if j2<1000 then write(lst,j2:3:0) else write(lst,j2:4:0);
40:      for i:=1 to width*2-26 do write(lst,blank);
41:      write(lst,'第 ');write(lst,pg);write(lst,'页 共 页');
42:      for i:=1 to 4 do writeIn(lst);
43:      if eof_end=False then begin
44:        write('按回车键继续打印...');readln;PrUpOrDn(ch_width3);end;
45:      page:=0;
46:    end;
47:  Procedure JoinTable;
48:    begin write(lst,chr1);k3:=k4;
49:      while k3<=width-1 do
50:        begin k3:=k3+2; write(lst,chr1);end;write(lst,chr1);

```

```

51:
52:         pr_col(ch_width2);page:=page+1;
53:         if page=length then PrUpAndDn(bool1,page,pg);
54:         pr_col(ch_width1);writein(lst);
55:         prleftblanks(page);write(lst,chr1);k4:=1;
56:     end;
57: Procedure ReadFileAndPretreatment;
58:     begin writeIn('--**-- 本程序为您提供格式文稿打印功能 --**--');
59:         Write('请输入需打印的文件名(例: c:\file1.txt):');
60:         Readln(fileName); Assign(FilVar, fileName); Reset(FilVar);
61:         Write('请输入行宽及页长(字数和行数)(例: 20 20):');
62:         Readln(width,length);
63:         if width<4 then width:=4; if length<5 then length:=5;
64:         for i:=1 to 4 do begin blanks1[i]:=' '; blanks2[i]:=' ';end;
65:         blanks2[2]:=chr9; width:=width*2+1;fullstop:=chr0;
66:         if length>15 then LeftB:=5 else LeftB:=2;
67:         i:=2; k4:=1; k6:=0; page:=0; pg:=0; bool1:=False;
68:         while i<=width-1 do
69:         begin ch_width1[i]:=chr3; ch_width1[i+1]:=chr4;
70:             ch_width2[i]:=chr3; ch_width2[i+1]:=chr7; i:=i+2;
71:         end;
72:         ch_width1[1]:=chr2;ch_width1[width]:=chr5;
73:         ch_width2[1]:=chr6;ch_width2[width]:=chr8; i:=1;
74:         while i<=width+1 do
75:         begin ch_width3[i]:=chr3;ch_width4[i]:=chr3; i:=i+1;
76:         end;
77:         ch_width3[1]:=chr2;ch_width3[width+2]:=chr5;
78:         ch_width4[1]:=chr6;ch_width4[width+2]:=chr8;
79:     end;
80: Procedure TextTreatment;
81:     begin
82:         i:=0; j:=0;
83:         while not eoln(FilVar) do
84:         begin i:=i+1; j:=j+1;
85:             read(FilVar, line[i]); str1[j]:=line[i];
86:             if line[i]<chr(160) then
87:                 begin j:=j+1; str1[j]:=blank; end
88:             ELSE
89:                 begin i:=i+1; j:=j+1;
90:                     read(FilVar, line[i]); str1[j]:=line[i];
91:                 end;
92:         end;
93:         readln(FilVar); k:=0; k1:=0; k3:=0; k6:=k6+1;
94:         while k<=j do
95:         begin k:=k+1;
96:             if str1[k]=blank then k1:=k1+1 else k1:=0;
97:             if k1=4 then begin
98:                 for k2:=k to j do str1[k2-2]:=str1[k2]; k1:=2; k3:=k3+1;
99:                 end;
100:         end;
end;

```



```

101:
102: Procedure PrintTreatment;
103: begin
104:     k1:=j-2*k3;k:=0;k2:=1;
105:     if k6=1 then
106:         begin write('准备好打印机后按回车键...');readln;
107:             PrLeftBlanks(page); PrUpOrDn(ch_width3);
108:             pr_col(ch_width1); writeIn(1st);
109:             prLeftBlanks(page);write(1st,chr1);end;
110:     if (((str1[1]=blank) and (k6<>1)) or ((yy=fullstop[1])
111:         and (xx=fullstop[2]))) then JoinTable;
112:     while k<k1 do
113:         begin k:=k+1;
114:             if k4=width then
115:                 begin k4:=1;k:=k-1;write(1st,chr1,chr1);
116:                     pr_col(ch_width2);page:=page+1;
117:                     if page=length then PrUpAndDn(bool1,page,pg);
118:                     pr_col(ch_width1);writeIn(1st);
119:                     prLeftBlanks(page);write(1st,chr1);
120:                 end
121:             else
122:                 begin k4:=k4+1;k2:=k2+1;
123:                     if k2=2 then begin write(1st,chr1);k2:=0;end;
124:                     write(1st,str1[k]);
125:                     end; xx:=str1[k];yy:=str1[k-1];
126:                 end;(while k<k1 do)
127:     end;
128: Procedure EndTreatment;
129: begin
130:     while page<length do
131:         begin i:=2;page:=page+1;
132:             while i<=width-1 do begin i:=i+2;write(1st,chr1,' ');end;
133:             write(1st,chr1,chr1);pr_col(ch_width2);
134:             if page<>length then begin pr_col(ch_width1);
135:                 writeIn(1st);prLeftBlanks(page); write(1st,chr1);end;
136:             end;
137:         end;
138: Begin(Mainprogram)
139:     ReadFileAndPretreatment;
140:     while not Eof(Filvar) do
141:         begin
142:             TextTreatment;
143:             PrintTreatment;
144:         end;
145:     JoinTable;
146:     EndTreatment;
147:     bool1:=True;PrUpAndDn(bool1,page,pg);
148:     close(Filvar);
149: end.(Mainprogram)

```

硬盘分区与DOS版本

四川省涪陵地区卫生局 彭 禾

读《计算机应用研究》1989年第3期《硬盘不同DOS版本的使用方法》受益良好,今就个人实践,补充如下:

一、分区与版本

1984年以来,DOS 3.00系列版本陆续发表,随之出现从低版本,主要是DOS2.00系列的2.00/2.10软盘启动后,不能访问硬盘,使部分用户深感不便,对DOS升级无余兴,甚至286型机降级安装DOS 2.0/2.1,导致1.2MB软驱等不能发挥正常作用,还影响速度。国内专业报

刊曾发表多篇文章论及此题,各具特色。

DOS 3.00系列手册中均在硬盘准备篇内提示:若DOS分区大于10MB及以DOS 3.30(3.20、3.10、3.00)格式化,将不能用3.00以前的版本访问硬盘。(原文——Note: If your DOS partition is larger than 10M bytes and you format it using DOS Version 3.30, you cannot use versions of DOS prior to DOS 3.00 to access your fixed disk.)但事实并非全如手册所述。实验数据记录如下:

DOS两系列版本对10-20MB硬盘DOS分区空间分配实例

	硬盘容量	文件分配表(FAT)				目录区(ROOT)				串(CLUSTER)				接受DOS 2.0/2.1访问
		(MB)	起始扇区	扇区数	表项值(B)	扇区号	扇区数	项数	扇区数	项数	扇区数	项数	KB	
DOS 2.00 FDISK.COM	10	1H	9H	10H	1.5	11H-30H	20H	512	8	4	—	—	—	—
FORMAT.COM	20	1H	9H	10H	1.5	11H-50H	40H	1024	16	8	—	—	—	—
DOS 2.10 FDISK.COM	10	1H	9H	10H	1.5	11H-30H	20H	512	8	4	✓	✓	✓	✓
DOS 3.30 FORMAT.COM	20	1H	9H	10H	1.5	11H-30H	40H	1024	16	8	✓	✓	✓	✓
DOS 3.00	4	1H	4H	6H	1.5	7H-26H	20H	512	8	4	✓	✓	✓	✓
	10	1H	9H	10H	1.5	11H-30H	20H	512	8	4	✓	✓	✓	✓
	16*	1H	DH	18H	1.5	19H-38H	20H	512	8	4	✓	✓	✓	✓
FDISK.COM FORMAT.COM	18	1H	25H	48H	2.0	49H-68H	20H	512	4	2	×	×	×	×
	20	1H	2AH	52H	2.0	53H-72H	20H	512	4	2	×	×	×	×

注: DOS 2.10 适用于PPC及PCjr,在PC/XT上,与DOS2.00等效

*实测15,974,400 Bytes, DOS分区数据依次为0, 1, 25(19H), 57(39H)

上表展示:

一、拒访分界线在文件分配表的项值1.5与2.0之间。凡字节数1.5B者,均容访问;2B即发生拒访。

二、DOS 2.0系列FDISK.COM分区者,不论容量10MB或20MB,执行DOS 2.00或DOS 3.00系列版本FORMAT.COM,其硬盘项值均为1.5B。

三、DOS 3.00系列(例如DOS 3.30)FDISK.COM随硬盘容量设置上述两种项值,超过16MB为2B;否则取1.5B。

四、未发现FORMAT.COM对'拒访'有决定性影响力。

DOS由FDISK.COM向硬盘第一扇区主引导程序之后公用分区信息表写入DOS系统标识符。DOS 2.00系列是01H,设FAT项值为1.5B,即12位,致串号上限4096,取每串扇区数8,长度4KB,对10MB硬盘属优选;DOS 3.00系列常规设置系统标识符为04H,设FAT项值2B,16位,串号上限达35536,大于16MB硬盘每串含扇区4,2KB,也属优选,若硬盘容量不超过16MB,此符取01H;DOS 3.30的

DOS扩展区为05H, 或04H (大于16MB)。

据此, 或可认为: DOS 3.00 系列手册中的提示宜修改为“若DOS分区大于16MB及以DOS 3.00(3.10、3.20、3.30)分区, 将不能用DOS 3.00以前的版本访问硬盘。(译文: If your DOS partition is larger than 16M bytes and your fdisk it using DOS Version 3.30, you cannot use versions of DOS prior to DOS 3.00 to access your fixed disk.)

上述发现对需要接受低版本DOS访问的用户似有益:

1. DOS 3.30 用户可将其大容量硬盘的DOS基本区(PRIMARY PARTITION)容量由10MB增加至16MB, 提高60%之多。

2. 采用硬盘管理软件(DISK MANAGER之类)的DOS 3.00—3.20用户分设硬盘时, C盘上限也达16MB, 基本上满足需要。

3. 拥有20MB硬盘的用户不妨先以DOS 2.00系列FDISK.COM设置DOS分区, 换DOS 3.00系列软盘, 热启动后执行同系列FORMAT.COM, 即能兼顾。

但是, 20MB硬盘安装DOS 2.00/2.10确有不不足之处, 每扇区数高达16, 即8KB之多, DOS分配文件容量以扇区为基本单元, 在实践中, 不少文件的长度小于4KB, 最突出是DBASE III命令文件, 造成的硬盘空间浪费往往颇为可观, 不过另一面, 却有利于提高大容量文件读出速度, 且目录区文件数增加一倍。此外, DOS 2.00系列全面支持硬盘容量上限为16MB, 一旦16MB以上区域磁媒出现缺陷需要处理时, DOS 2.0/2.1就无能为力, 其FORMAT.COM的出错信息“借用”TRACK 0 BAD—DISK UNUSABLE, 程序中断, 信息不全, 硬盘无法启动, 宜升级为DOS 3.00系列(DOS 3.00全面支持20MB硬盘; 3.10版增至30MB; 3.30支持30—720MB, 实测32—770MB; DOS 4.00高达2GB)。若必须用DOS 2.00/2.10, 可仅分配给DOS分区16MB, 再次格式化, 或在格式化中断后, 试行A) SYS C: 及拷贝COMMAND

.COM; 亦可先以DOS 3.20/3.30分区并格式化, 当显示柱面值大于480且读数缓慢, 乃坏柱面起始值, 予以中断, 删除DOS分区后, 再用DOS 2.00/2.10启动, FDISK.COM设置DOS分区柱面总数(>480), 使弃用量恰如其份。

使用DOS 3.30经验之一是: 将应用软件拷入硬盘子目录, 正常操作结束, 及时备份。为保密, 可删除子目录内容, 用其它信息覆盖。深受用户青睐的CC-BIOS 2.13F移装PC/XT、286及386机硬盘子目录也为有力佐证。

二、PC-DOS与CC-BIOS

CC-DOS之称谓沿用近七年, 实为CC-BIOS之误, 开发者之一在《计算机世界》1987年12月号《CCBIOS的发展》中已作充分阐述。它的版本自成体系, 如CC-BIOS 4.0在1987年推出, 较PC-DOS 4.00问世早余年, 其运行环境广泛, PC-DOS 2.00及以上版本均可; 另一面, CC-BIOS 2.0在PC-DOS 3.30环境下运行也正常。PC-DOS与CC-BIOS两者本不致混淆, 但文中例3似能反映部分实况, 从安装PC-DOS 3.10的硬盘引导, 再执行A: AUTO-EXEC([.BAT])—运行FILE1.COM及CCCC.EXE装载CCLIB等, 如斯启用CC-BIOS并未也不可能将“CCDOS 2.1”软盘上PC-DOS 2.1的IBMBIO.COM、IBMC.COM及COMMAND.COM替换内存中PC-DOS 3.10三个基本软件, 以A) VER显示现版本号—DOS 3.10, 证实微机仍处DOS 3.10控制下, 硬盘当调用如常。因此认为“这样系统进入CCDOS 2.1中, 就能访问硬盘进行任何操作, 系理解不全面所致。将CC-BIOS拷入硬盘子目录, 在根目录建启用批文件, 如CC-BIOS 2.13F, 堪称方便。

DEBUG对硬盘分区的浅析

福建省三明化工厂合成氨厂计量科 陈一侃

一、引言

在IBM和长城系列微机中的硬盘,为用户提供了1~4个的用户分区,通常情况下人们都是用系列应用程序FDISK来对硬盘进行分区的大小设置、活性分区选择等。但由于FDISK进行重新分区时往往造成边界数据的丢失。特别是在硬盘中已进行过分区的情况下,再建立和追加一个分区是极为不方便,必须将原有数据拷出,操作后再拷入的办法解决,为此我对硬盘的信息分布进行剖析,寻找方便可靠的方法来实现填补FDISK应用中的不足之处。

二、硬盘分析

硬盘的支持系统允许多个操作系统使用,逻辑上可使用1~4个操作系统,即要求对硬盘进行分区,一个用户从硬盘上启动所选择的分区,在一指定区内的空间是连续的,并专门供一指定的操作系统使用,每个操作系统只能有它自己的一个分区,并且任何操作系统一定要把它的分区看成是一个完整的磁盘,同时还要确保它的功能和应用程序不要去取这个磁盘的其它分区,由于上述的要求,所有硬盘的第一扇区(0道、0面(头))上,写上一个主引导记录(见分区信息表)。

该引导记录包含有:1.装入引导记录的代码。2.在主引导记录的末尾有一“分区表”。在表中每项是16个字节长,存放四个可能分区中每个分区的开始和结束柱面,扇区和磁头号,以及在这个分区以前的扇区数和这个分区占用的扇区数。该表是与主引导记录一起,在微机加电自检并自举成功后,由系统功能调用INT19读入内存单元并以一定的偏置(见分区信息表)存放。

其中BOOTind字节是“引导指示符”,决定一个分区是否包含有一个能装入的操作系统。FDISK初始化应用程序,把用户选择的分区(活性分区)标为“引导表”,就是通过把“BOOTind”字节数值设为80H,同时把其它分区的“BOOTind”字节置零来实现。这个80H通过标准的引

导程序去装入扇区,这个扇区的位置包含在随后的3个字节中,即分区开始的H, S, CYL三字节为开始,以分区结束中的H, S, CYL三个字节为终点的扇区。“CYL”字节存放柱号的低位,而它的高2位在“S”字节的高2位。它和ROM BIOS中断13H一致,柱面编号允许10位。

“sys tind”字节存放操作系统的指示符,其方式为:

00H—不识别的(即没指定的)

01H—DOS

分区的分配是按柱面的倍数,并以扇区1,磁头0开始。分配盘开始部的分区的扇区2开始,因为盘中主引导记录和分区信息表占了第一扇区(见硬盘信息表),对于长城0520机,提供20M字节硬盘,出厂前已经以指定间隔因子进行格式化。盘为611柱,4磁盘,17扇区/每磁头,512字节/每扇区。标有四个字节信息的“该分区前扇区数”字节放有当前分区前面的扇区数。这个数值是这样得来的即从磁盘的0柱、1扇区,0头开始计数,增加扇区,磁头和磁道的数值,一直到这个分区的开始位置。例如:第二个分区在柱面1,扇区1,磁头0开始,那么这个值就是68(十进制)。因为它前面已分配了1个磁道,每磁头17扇区。先用低位字节存放在这个信息区中。“该分区扇区数”存放的是当前分区占有扇区数,先从低位字节存起。引导记录把最后两个字节用FDISK设置为(55H, AAH)。当做“符号差”去识别一个有效记录。

三、DEBUG的使用

从以上的分析,根据信息在硬盘的分布情况,用DEBUG进行测试状态后,用L0201读分区表到内存,同时用D命令进行显示,根据需要用E命令修改分区信息表中相应单元内容,即可更改分区的属性、大小。并有不妨碍已经存在的分区的情况下,增加一个分区。最后将修改的分区信息表用W0201写盘。该方法简便可靠,也

是较为灵活的手段。正确使用不会造成数据丢失。只要了解微机硬盘有关资料,用类似方法均能得到满意的效果。

附表:

<1>硬盘信息表

0道	主引导记录	PC DOS分区	文件分配表	文件根	
0面	分区信息表	引导程序	FAT	目录	文件内容
(头)					

—> | 第一扇区 | | <—————PC DOS 第一分区—————>....

<2>硬盘分区信息表

偏置的用途

磁头 扇区 柱面

1BE 1#分区开始	BOOT IND	H	S	CYL
1C2 1#分区结束	SYS IND	H	S	CYL
1C6 1#分区前扇区数	low word		high word	
1CA 1#分区扇区数	low word		high word	
1CE 2#分区开始	BOOT IND	H	S	CYL
1D2 2#分区结束	SYS IND	H	S	CYL
1D6 2#分区前扇区数	low word		high word	
1DA 2#分区扇区数	low word		high word	
1DE 3#分区开始	BOOT IND	H	S	CYL
1E2 3#分区结束	SYS IND	H	S	CYL
1E6 3#分区前扇区数	low word		high word	
1EA 3#分区扇区数	low word		high word	
1EE 4#分区开始	BOOT IND	H	S	CYL
1F2 4#分区结束	SYS IND	H	S	CYL
1F6 4#分区前扇区数	low word		high word	
1FA 4#分区扇区数	low word		high word	
1FE 符号差	hex 55	AA		

保护硬盘的DOS外部命令

新疆军区后勤部自动化站 王志中

在众多的微机用户中,存于硬盘的数据、文件有时会被他人误删除或非法窃取,造成无可挽回的局面。为杜绝这种现象的发生,最为有效的方法是用软件方法给硬盘加以保护,下面就如何实现对硬盘的有效保护,介绍一种简单实用的方法。

一、实现方法

通过分析硬盘自举程序(INT 19)可知,它的自举过程为:首先从软盘读取自举程序并装入自举单元(0:7C00),如读软盘不成功再试读硬盘,硬盘上有有效的自举块存放于0号柱面1扇区,且最后两个字节为055H、0AAH,自举程序通过判断最后两个字节,而决定是继续还是转到驻留ROM BASIC(INT 18)。据此,只要将硬盘有效标志更改后,再次使用硬盘就会出现系统中无硬盘的现象,使其它用户无法使用硬盘。如用户欲从硬盘启动便进入ROM BASIC,如用户从软盘启动欲使用硬盘,则系统会提示:Invalid drive specification。文中所介绍的程序可经过编译,链接形成EXE文件,并通过EXECOM.COM命令转换成.COM文件,作为DOS外部命令使用。

二、程序清单

1. 保护硬盘程序

```
sseg segment para stack
sseg ends
cseg segment para public
    assume cs: cseg, ds: cseg, es: cseg,
        ss: sseg
buffer DB 512 DUP(?)
```

```
ORG 100H
START: cli
    mov dx, 0080h
    sub ax, ax
    int 13h; 硬盘复位
    mov ax, 0201h
    mov bx, offset buffer
    mov dx, 0080h
    mov cx, 1
    int 13h; 读出硬盘0道一扇区512字节
    mov ax, 0000
    mov word ptr buffer+510d, ax;
    改写标志
    mov ax, 0301h
    mov bx, offset buffer
    mov cx, 0080h
    mov cx, 1
    int 13h; 将改写后内容写入硬盘0道一
        扇区
    int 18h; 进入 Rom basic
cseg ends
end START
```

2. 去掉硬盘保护程序

```
sseg segment para stack
sseg ends
cseg segment para public
    assume cs: cseg, ds: cseg, es: cseg,
        ss: sseg
buffer db 512 dup(?); 存放硬盘0道一扇区的
        内容
ORG 100H
START: cli
    mov dx, 0080h
    sub ax, ax
    int 13h; 硬盘复位
    mov ax, 0201h
    mov bx, offset buffer
    mov dx, 0080h
    mov cx, 1
    int 13h; 读入硬盘0道1扇区512字节
    mov ax, 0aa55h
    mov word ptr buffer+510d, ax;
    还原标志
    mov ax, 0301h
    int 13h; 将正确的硬盘标志写回0道1扇区
    int 19h; 硬盘引导
cseg ends
end START
```

逻辑硬盘‘丢失’的预防及处理

四川省涪陵地区卫生局 彭 禾

近年286档次微机配置40MB或更大容量硬盘日众,且即使原装DOS3.30,仍有以DISK-MANAGER(DM)之类硬盘管理软件添设逻辑盘,常见为C盘32MB, D逻辑盘8MB……。

DM管理逻辑盘必须在C盘根目录建立以下两项条件:

1.COMFIG.SYS中添DEVICES=DMD-RVR.SYS

再谈多种操作系统共享硬盘的方法

安徽省芜湖市纺织器材厂微机室 徐雪斌

多种汉字操作系统共享硬盘确实有不少优点,笔者深有体会。成都电缆厂的杨厚生同志在89年第6期上已谈了两种操作系统共享硬盘的问题,该文对初学者在CW—0520系列微机上安装中、高分辨率两种操作系统提供了一个可行的方法。笔者认为:对于已掌握了DOS操作基础知识的微机工作人员来说,只要解决如下几个问题,就可达到多系统共享硬盘的目的:

一是汉字DOS与原硬盘里的西文DOS的版本应相匹配,但不一定要完全一致。如长城0502C—H微机里的西文DOS版本是2.10,则汉字DOS的实际版本可以是2.11或2.13等等,但不可使用3.X的汉字DOS版本。

二是各系统所使用的汉字字库,若不相同,则文件名也不能相同,否则要用DEBUG去进行有难度地修改。因为字库只能处在根目录下方可

使用。

三是要分别将各系统放入不同的子区(即将除必须放在目录下的几个文件外,如:SYS.COM、长城机上的GRAFTABLE.COM和GRAPHICS.COM等,其它均拷贝到不同的子目录里),使其互不干扰。

最后可在根目录下建立一个自启动批处理文件,使开机后直接进入可选汉字系统。

多种操作系统存于硬盘,给使用者提供了极大地方便。笔者曾在本单位的一台长城0520C—H微机上将CW—DOS3.0(其实际版本是DOS2.10)、长城五笔字型 and GW—DOS2.13A三种汉字操作系统共存于硬盘之中,但各系统都有自己的“小天地”,即分别放入了三个不同的子目录里,并在根目录下编写了一个自启动批处理文件的程序(程序清单附后),当机器冷、热启动

2. 拷入DMDRVR.SYS

两者少一不可,缺前项屏幕无提示,而后者则显该文件不存在(或属无效)。

实践中见过多台微机D盘‘丢失’(显示:无效驱动器符),不明真相者以为用30MB硬盘冒充。查C盘,每见CONFIG.SYS缺支持逻辑盘语句,而根目录有支持逻辑盘文件,原由一目了然。

实践告示,上述‘故障’每与拷贝不当有关,大都是‘盲动’地将整个软盘文件拷入硬盘根目录,CONFIG.SYS遭复盖,而DMDRVR.SYS的文件名及扩展名均特殊,不致误删,偶见于格式化硬盘前忽视根目录文件备份,又缺乏使用DM经验者。

预防的关键在于严格执行常规——整个软盘文件拷入硬盘根目录之前,必须查看源盘及目标盘目录中有无同名文件,如:CONFIG.SYS及COMMAND.COM、AUTOEXEC.BAT等。若有,应予正确修改补充,COMMAND.

COM,则保持主机版本,AUTOEXEC.BAT大都需改名。并将新的CONFIG.SYS,以及DMDRVR.SYS备份。行之有效的简明办法是将源盘装入子目录,调用不难。

1987年推出的DOS3.30在理论上支持768MB硬盘,除C盘外,可将DOS扩展区划分为D—Z逻辑盘,硬盘符总数达24之多,其容量上限均为32MB,且分盘的灵活性优于DM,逻辑盘设置在执行FDISK中一气呵成,又无需配置、设备文件,迄今尚未遇见DOS3.30的逻辑盘‘丢失’。

DM弥补DOS早期版本的缺陷,发挥出应有效应。现有DOS3.30,在一定范围内,似无必要外加软件,上述隐患不复存在。由于DM分置的D盘非DOS分区,DOS无删除之力,需调用DM,或AT诊断盘(2.03版)‘PREFORMAT’,再以DOS3.30FDISK.COM分设DOS基本区及扩展区——逻辑盘区。

抢占硬盘空间的有效方法

上 海 茹建平

在IBM—PC系统中,硬盘是开放性的。任何用户都能使用硬盘,占用硬盘空间。即使硬盘容量再大,也会被占尽,后来上机的用户又不知应删去哪些文件,以便装入一些必须在硬盘中执行的文件。为防止这类情况的发生,笔者编写了两个汇编程序,一个用来保存硬盘空间,名为SAVE.ASM,另一个用来释放这些空间,以供程序员使用,名为RELEASE.ASM。

本方法是根据系统对文件分配表(FAT)中标记为坏簇的扇区不予分配的原理,把空闲可用的簇号标记(0000H),改为坏簇记号(FFF

7H),待到使用时,再将坏标记(FFF7H),改回成0000H,此时就可使用这些空间了。二段程序基本相同,只在标记判断及修改处有区别。下面以SAVE.ASM为例。

笔者是在30兆硬盘的微机上编写程序的。30兆硬盘有五个磁头,每个柱面有85个扇区,文件分配表从相对扇区1号开始存放,共为61个扇区(十六进制值为3DH)。按序将存放文件分配表的扇区一个个读入,判断处理,最后写回硬盘。读入扇区前,需把将读入的扇区的相对号转换为绝对扇区号,然后,才可调用BIOS功能。程序如下

后,可键入数字1、2或回车来选择不同的系统。 14: GWINT16

由于各汉字操作系统都有其自身的特点,如GW 15: 3

—DOS有原装的“味道”,五笔字型有高速输入的效果,DOS2.13A有四种体的打印,因此, 16: goto c

多系统共享硬盘虽然牺牲了一点存贮空间,但它 17: :A

相当于增强了微机的系统功能,为不同的软件选择 18: cd\0520c-h\wb

最佳的运行环境奠定了良好的基础。 19: kbd

20: wb

21: db

22: lch

23: cd\off

24: 3

25: :c

26: cd\qdcx

27: REALTIME/C

28: PATH\;\Dbase3;\0520c-h;\hw

29: VERIFY ON

30: KEY

31: cd\

32: CLS

33: ECHO ON

1: ECHO OFF

2: cd\qdcx

3: MENUH

4: IF ERRORLEVEL 50 GOTO A

5: IF ERRORLEVEL 49 COTO B

6: GWINT16A

7: INT10C

8: FILE24 SFHK

9: lx1

10: 9999c4 4

11: goto c

12: :B

13: cd\off


```

; IBM-PC HARDDISK UTILITY.SAVE.ASM
STACK SEGMENT STACK
DB 1000 DUP ('/STACK/')
STACK ENDS
DATA SEGMENT
EIGHTS DB 85
SECTNUM DW 0
COUNT DB 3DH, FAT 占的扇区长度
FATBUF DB 512 DUP (4 2); 工作缓冲区
DRIVER DB 80; 硬盘驱动器号
SECTOR DB 2
HEADER DB 1
TRACK DB 0
WRITEEER DB 0DH, 0AH, 'WRITE WRONG $'
MESS AGE DB 0DH, 0AH, 'BEGINNING $'
DATA ENDS
PROGRAM SEGMENT
MAIN PROC FAR
ASSUME CS: PROGRAM, DS: DATA
ASSUME ES: DATA, SS: STACK
START:
    PUSH DS
    XOR AX, AX
    PUSH AX
    MOV AX, DATA
    MOV DS, AX
    MOV ES, AX
; DISPLAY WORKING INFORMATION
    MOV DX, OFFSET MESS AGE
    MOV AH, 9
    INT 21 H
; READ FIRST SECT OR OF FAT
    MOV SECT NUM, 2
    MOV COUNT, 3DH
    JMP READ
CIR,
JUDGE WHET HER FINISH
    MOV AH, COUNT
    SUB AH, 1
    MOV COUNT, AH
    CMP AH, 0
    JNZ CC1
    JMP SUCC; 完成
CC1,
    MOV AX, SECTNUM; 处理下一个FAT扇区
    ADD AX, 1
    MOV SECT NUM, AX
    DIV EIGHTS; 绝对扇区变换
    MOV TRACK, AL
    CMP AH, 18
    JB C1
    CMP AH, 35
    JB C2
    CMP AH, 52
    JB C3
    CMP AH, 69
    JB C4
    JMP C5
C1,
    MOV SECT OR, AH
    MOV, HEAD ER, 1
    JMP CC2
C2,
    MOV HEAD ER, 2
    SUB AH, 17
    MOV SECTOR, AH
    JMP CC2
C3,
    MOV HEAD ER, 3
    SUB AH, 34
    MOV SECTOR, AH
    JMP CC2
C4,
    MOV HEAD ER, 4
    SUB AH, 51
    MOV SECTOR, AH

```

```

C5:      JMP      CC2
        MOV      HEADER, 5
        SUB      AH, 68
        MOV      SECTOR, AH
; READ SECT OR
READ:    MOV      BX, OFFSET FATBUF
        MOV      AL, 1
        MOV      CH, TRACK
        MOV      CL, SECT OR
        MOV      DH, HEADER
        MOV      DL, DRIVER
        MOV      AH, 2
        INT      13H
; HANDLE WITH FAT
        MOV      CX, 0
        MOV      BX, 0; 变址寄存器
        PUSH     BX
LOP:     ADD      CX, 1
        POP      BX
        MOV      AL, [FATBUF+BX]
        ADD      BX, 1
        MOV      AH, [FATBUF+BX]
        ADD      BX, 1
        PUSH     BX
        CMP      AX, 00
        JZ       SPACE; 是空闲扇区
        JMP      L3
SPACE:   MOV      [FATBUF+BX-2], 0F7H; 置坏标志
        MOV      [FATBUF+BX-1], 0FFH
L3:      MOV      AX, CX
        CMP      AX, CX; 每扇区可存放的簇标记个数
        JZ       WRITE; 一个扇区已被处理, 回写
WRITE:   JMP      LOP
        MOV      BX, OFFSET FATBUF
        MOV      AL, 1
        MOV      AL, SECT OR
        MOV      CH, TRACK
        MOV      DH, HEADER
        MOV      DL, DRIVER
        MOV      AH, 3
        INT      13H
        CMP      AH, 00
        JNR      ERROR
        JMP      CIR; 处理下个扇区
SUCC:    RET;     全部处理完
ERROR:   MOV      DX, OFFSET WRITEER
        MOV      AH, 9
        INT      21H
        RET
MAIN ENDP
PROGRAM ENDS
        END      MAIN

```

长城0520C-H 高中分辨率汉字系统同从 硬盘自举的新方法

山东大学晶体材料研究所 刘立强

长城0520C-H 微机是国内普遍使用的微机系列,它具有高中分辨率两种显示模式,并可分别运行长城微机公司提供的GWBIOS和IBMP/C/XT系列及其兼容机流行使用的CCBIOS (CC-DOS) 两种汉字系统,但由于这两种汉字系统

不能同从硬盘自举,给微机应用带来许多不便。为解决上述问题,有人采用了修改COMMAND.COM文件来改变自举过程的方法^[1],此方法虽切实可行,然而由于用户各自使用的DOS操作系统版本不同,其文件长度及修改位置也不一

样,给修改工作带来许多困难。笔者通过在 AUTOEXEC.BAT 文件中增加一小屏幕状态判别程序,使在不同分辨率下开机时自动进入不同的汉字系统。这样就无需修改 COMMAND.COM,使问题得到解决。

众所周知,长城0520C-H汉字系统是在微机启动时,通过 AUTOEXC.BAT 来建立的,笔者是在执行汉字系统文件之前,增加一个名为 SJ.COM 的程序,检查判断屏幕状态并设置不同的退出代码,然后利用DOS提供的IF ERRORLEVEL 功能、测试退出代码并执行不同的汉字系统文件。在[程序1]中给出了SJ.ASM 汇编语言程序。

```
CSEG SEGMENT PARA PUBLIC 'CODE'
    ORG 100H
    ASSUME CS:CSEG,DS:CSEG,ES:CSEG,SS:CSEG
JUDGE PROC NEAR
    MOV DX,OFFSET MESSAGE
    MOV AH,09
    INT 21H
    MOV AH,0FH
    INT 10H
    CMP AL,03
    JZ A03
    MOV AX,4C00H
    INT 21H
A03: MOV AX,4C01H
    INT 21H
JUDGE ENDP
MESSAGE DB 'GW 0520C-H SCREEN JUDGE',0AH,0DH
        DB 'WAIT PLEASE ! ','$'
CSEG ENDS
    END JUDGE
```

程序 1: 屏幕判别程序 SJ.ASM

此程序首先是利用 BIOS 中断 INT10H 的 0FH 号功能,检查屏幕状态,然后根据 AL 中的返回值,用 DOS 中断 INT21H 的 4CH 号功能设置不同的退出代码,供 IF ERRORLEVEL 测试。程序经过 MASM 汇编、LINK 连接后,再用 EXE2BIN 转变成 COM 文件。而后将 SJ.COM 和 GWBIOS 及 CCBIOS 的全部文件都考入硬盘,并建立合适的 AUTOEXEC.BAT 批文件[程序2],即完成所需要的全部工作。

```
ECHO OFF
CLS
SJ
IF ERRORLEVEL 1 GOTO MOD
FILE1
CCCC
GOTO END
:MOD
GWINT16
:END
```

程序 2: AUTOEXEC.BAT

现在就可可在不同分辨率下开机或热启动时进入不同的汉字系统。当然,利用这一方法,只要对 AUTOEXEC.BAT 稍加改动,就可可在不同的分辨率下开机时进入不同的应用系统。

参考文献:

- [1] 邓京明,《计算机应用研究》第2期, 29页, 1989
- [2] 张怀莲编,“宏汇编语言程序设计”, 电子工业出版社, 1987
- [3] “DOS 使用手册”, 长城微机开发公司, 1987

提高IBM PC/XT 硬盘空间利用效率的一种有效方法

武汉汽车配件厂科研所 许传国

PC DOS 在给磁盘文件分配空间时是以簇(Cluster)为单位进行分配的。如果文件长度不足一簇, DOS 也给它分配完整的一簇;如果文件长度大于一簇但少于二簇, DOS 则给它分配完整的二簇;以此类推。因此,在磁盘上,磁盘文件之间可能是充满着许多无法利用的“碎片”空间的。如果磁盘上文件的数量相当多,则这种“碎片”空间的浪费也就更加惊人。我厂在

PC/XT 的 20 M 硬盘上开发了一个经营信息管理系统,各类磁盘文件接近 1200 个,查看硬盘容量只剩下几 KB 字节空间,要在上面再开发新东西已是很困难了。

簇的大小在不同的 DOS 版本中是不同的。使用较小的簇可以使文件在磁盘中的放得比较紧密些。也就是说,使用较小的簇,每个文件可能浪费的“碎片”空间要小一些,因此磁盘空间的使

用会更有效率。

查阅DOS手册可以知道, DOS 3.X在文件分配表上(FAT)做了重大的修改。文件分配表的每一项(对应到磁盘中的一簇)以16位元来表示,而不是以DOS 2.x的12位元来表示。所以,一个磁盘中最多可以有65536簇,而不是以前的4096簇。簇的数量增加意味着在同样大小的磁盘中每一个簇可以有较小的空间。例如,一台20MB的硬盘,用DOS 3.x去格式化时每簇的大小是2KB字节,而用DOS 2.x格式化时每簇大小是8KB字节。

因此,如果用DOS 3.x代替DOS 2.X去对IBM PC/XT硬盘进行格式化,那么PC/XT硬盘空间利用效率肯定会得到提高。

我们用AST 286机的DOS 3.3版本升级代替DOS 2.1对PC/XT机的硬盘重新进行格式化,收到了较好的效果。具体做法如下:

1.用BACKUP程序(DOS 2.1版本)把PC/XT机20MB硬盘上全部文件备份到软盘进行备份保存。

2.制作360K软盘的DOS 3.3系统盘,(因为我们用的AST 286机DOS 3.3系统盘是1.2MB的高密盘,在PC机上无法使用):

在AST 286机A驱动器放入1.2MDOS3.3系统盘,启动之;在286机B驱动器中放入360K软盘,键入

```
A>FORMAT B: /S
```

待B盘格式化完成后,把DOS 3.3的FORMAT.COM, FDISK.COM文件拷入B盘。

```
A>COPY FORMAT.COM B: 
```

```
A>COPY FDISK.COM B: 
```

3.把PC/XT的20MB硬盘进行初级格式化:(我们的机器如果不首先进行初级格式化,直接用DOS 3.3的FORMAT进行格式化时,虽然格式化工作可以做完,但不能在C盘上直接启动,也列不出目录,无法对硬盘进行操作)

有二种方法。一种是用硬盘管理程序(HARD DISK MANAGEMENT PROGRAM),先用360K的DOS 3.3在PC/XT机A盘进行启动,然后将DISK MANAGER Version3.62盘放入A盘。键入A>DM

程序装载后提问: only 1 drive detected, is correct? (y/n) (只检测到一个硬盘,对吗?) 键入: y

然后又问: ENTER MODEL CODE FOR DRIVE 1 (RETURN IF STANDARD) (键入硬盘型号,如果是标准的,按回车键): 因为是标准硬盘,按回车键。

然后又问: Do you wish to INITIALIZE this drive? (y/n): 回答y

然后程序将自动对硬盘进行初级格式化、硬盘分区及格式化(FORMAT)。

第二种方法,用AST 286机所提供的实用软件(AST premium/286 UTiLity software)中硬盘管理程序。此软件也在1片1.2M高密盘上。为此要把有关部分转移到360K软盘上来。具体操作如下:

先把DOS 3.3系统放入286机A驱动器,360K软盘放入B驱动器,启动286机,然后

```
A>FORMAT B: /S
```

B盘格式化完成后,把应用软件盘放入A驱动器,键入:

```
A>COPY AUTOEXEC.BAT B: 
```

```
A>CD /ASTUTIL
```

```
A>B: 
```

```
B>MD /ASTUTIL
```

```
B>CD /ASTUTIL
```

```
B>A: 
```

```
A>COPY *.* B: 
```

A盘ASTUTIL子目录下的所有文件拷贝到B盘的ASTUTIL子目录后,一个360K的实用程序盘就完成了。

将此软盘放入PC/XT机A驱动器,启动,屏幕上出现此实用程序菜单,选择4(Fixed Disk Options),然后屏幕上出现Fixed Disk Options部分菜单,选择1(Perform Low Level Format),以后各部分选择可参阅DOS 3.3操作手册进行操作,直到把硬盘初级格式化做完。

然后,调用前面所叙制成的360K DOS 3.3系统盘的FDISK程序对硬盘进行DOS分区。

4.利用DOS 3.3对PC/XT硬盘进行格式化:

```
A>FORMAT C: /S
```


硬盘保护的一种方法

河北省经济信息中心 宋俊山

如何保护个人计算机的硬盘,使在硬盘上的数据或程序不被别人使用或拷贝,许多拥有个人计算机的用户通过在 AUTOEXEC.BAT 中加口令通行字 (PASSWORD) 或者通过改造 COMMAND.COM 文件设置口令通行字来保护硬盘。但这些方法都不能有效地防止非法用户从软盘启动系统来进入硬盘。本文给出一种方法,使得非法用户能在软盘上启动系统但不能进入硬盘,并且你还可以看到,即使非法用户剥夺了你的解除保护程序,仍然不能进入你个人计算机的硬盘。

我们知道,IBM PC/XT 的 10M 字节硬盘的 0 磁道 0 面第 1 扇区上驻留有硬盘主引导记录(硬盘分区表嵌在硬盘主引导记录之中)。该扇区是一个独立的扇区,它不属于任何一个分区,既使在硬盘上只划分一个分区也是这样。因为 DOS 的 FDISK.COM 程序在分区时把十六进制 80 放入分区表相应的分区引导指示栏来标明该分区是可引导的,而且该扇区以最后两字节的值是否为 55AA 来标识该扇区为有效或无效的引导记录(同时也是分区表的结束标志)。因此若硬盘主引导记录的关键信息丢失或有错,例如:分区表中引导指示栏标识为不可引导(非 80H)或硬盘主引导记录标识为无效(扇区最后两字节不为 55AA),硬盘便不能引导和启动操作系统并相应给出错误提示或进入 ROM BASIC 状态。

据此,我们只要将引导扇区某一关键字节内容作相应改动,如上所述将分区表中引导指示栏标识为不可引导(非 80H)或将硬盘主引导记录标识为无效(非 55AA)。这样,在系统初始化时便不能引导和启动操作系统,并且用软盘启动后试图进入硬盘,系统提示: InvaLid drive,

Specification (无效驱动器定义)。而当你自己想使用硬盘时,只需将正确的硬盘主引导记录写入硬盘的 0 磁道 0 面第 1 扇区即可。

下面是用汇编语言写的两个程序,用 DEBUG 输入。

程序一: P.COM (保护硬盘程序)。

程序二: NP.COM (解除保护程序)。

程序 P.COM 的作用: 首先将硬盘 0 面 0 磁道 1 扇区的主引导记录读到内存磁盘缓冲区,并将缓冲区的内容写到 A 驱动器中软盘的逻辑 200H 扇区上,然后移动缓冲区指针将 55AA 修改为 00AA,使硬盘主引导记录标识为无效。将已修改了的扇区主引导记录写入硬盘的 0 面 0 磁道 1 扇区,此程序对硬盘进行保护。

程序 NP.COM 的作用: 将 A 驱动器中的软盘逻辑 200H 扇区上的内容读入内存缓冲区,然后将缓冲区的内容写到硬盘 0 磁道 0 面 1 扇区。此程序对硬盘解除保护。

可以看到: 运行程序一 (P.COM) 将产生一个保护盘,此盘逻辑 200H 扇区上的内容为正确的硬盘主引导记录。而解除保护程序 NP.COM 的运行依赖于此盘逻辑 200H 扇区上的内容。因此,既使非法用户将 NP.COM 拷贝,由于他的软盘逻辑 200H 扇区的内容不是硬盘主引导记录,运行后仍然不能进入你的硬盘。

需要注意的是: 保护软盘最好是一个格式化了空盘,否则可能会使你盘上逻辑 200H 扇区的内容受到破坏。有兴趣的读者可以利用 INT 13 在对磁道作格式化时将所选择的磁道记为“坏”磁道,然后将硬盘主引导记录写在此磁道某扇区上。这样,你就不必担心因为存入别的程序而破坏你指定扇区的内容了。

5. 利用 RESTORE 程序 (DOS 2.1 版本)
把先前拷贝的硬盘文件备份恢复到硬盘上。拷贝恢复完毕后,我们查看硬盘目录,发现现在的硬盘的自由空间有 5MB 多,即利用 DOS 3.3 对 PC

/XT 的 20M 硬盘格式化后,硬盘空间的有效利用率多出了 5MB 空间(总的硬盘容量仍同用 DOS 2.1 格式化后一样不变)。利用这 5MB 空间又可以新开发一些应用项目了。

用BASIC语言调用汇编 子程序的方法修复硬盘初始引导扇区

广州南方医院信息科 肖浩延、郭松亮

摘要: 本文介绍了一个 BASIC 程序, 借助于它可以方便地修复遭破坏的硬盘初始引导扇区。

硬盘初始引导程序在硬盘的0面, 0柱面, 第一个扇区上, 启动的时候由INT 19H 引导到内存的0000:7e00。它的任务是根据分区信息表上提供的活动标志和各个分区的定位信息, 找到当前活动分区在硬盘上所占据的空间的起始位置, 即找到活动分区逻辑上的第一扇区所对应的物理上的磁头号、柱面号、扇区号, 然后再把这一扇区的内容读到内存的0000:7C00。

假如硬盘初始引导程序被破坏, 就无法用硬盘来启动操作系统, 只能用系统盘启动操作系统, 即使这样, 我们也不能进入硬盘, 当我们企图对硬盘进行读写操作时将出现 "Invalid drive specification" (非法的驱动器标志)。重新格式化硬盘虽然能够恢复直接从硬盘启动, 但是这样作会导致硬盘所有信息的丢失, 十分可惜。下面介绍一个 BASIC 程序, 可以恢复硬盘

```
DEBUG
-A
5BD8:0100 MOV DX,0143
5BD8:0103 MOV AH,09
5BD8:0105 INT 21
5BD8:0107 MOV AH,08
5BD8:0109 INT 21
5BD8:010B MOV BX,7C00
5BD8:010E MOV DL,80
5BD8:0110 MOV DH,00
5BD8:0112 MOV CH,00
5BD8:0114 MOV CL,01
5BD8:0116 MOV AL,01
5BD8:0118 MOV AH,02
5BD8:011A INT 13
5BD8:011C MOV BX,7C00
5BD8:011F MOV DX,0200
5BD8:0122 MOV CX,0001
5BD8:0125 MOV AL,00
5BD8:0127 INT 26
5BD8:0129 MOV BX,7DFE
5BD8:012C MOV AH,00
5BD8:012E MOV BX,1AH
5BD8:0130 MOV BX,7C00
5BD8:0133 MOV DL,80
5BD8:0135 MOV DH,00
5BD8:0137 MOV CH,00
5BD8:0139 MOV CL,01
5BD8:013B MOV AL,01
5BD8:013D MOV AH,03
5BD8:013F INT 13
5BD8:0141 INT 20
5BD8:0143 DB 0,0,0,0D,0A
5BD8:0148 DB "Insert p-np diskette in driver A:",0D,0A
5BD8:016B DB "and strike any key when ready","$"
5BD8:0189
-R CX
CX 0000
:89
-N A:P.COM
-W
Writing 0089 bytes
-Q
```

程序一, P.COM

```
C>DEBUG
-A
5BD8:0100 MOV DX,012B
5BD8:0103 MOV AH,9
5BD8:0105 INT 21
5BD8:0107 MOV AH,08
5BD8:0109 INT 21
5BD8:010B MOV BX,7C00
5BD8:010E MOV CX,0001
5BD8:0111 MOV DX,0200
5BD8:0114 MOV AL,00
5BD8:0116 INT 25
5BD8:0118 MOV BX,7C00
5BD8:011B MOV DL,80
5BD8:011D MOV DH,00
5BD8:011F MOV CH,00
5BD8:0121 MOV CL,01
5BD8:0123 MOV AL,01
5BD8:0125 MOV AH,03
5BD8:0127 INT 13
5BD8:0129 INT 20
5BD8:012B DB 0,0,0,0D,0A
5BD8:0130 DB "WARNING: IF YOU ARE A ILLEGAL USER,PLEASE
DON'T USE THIS PROGRAM!",0D,0A
5BD8:0174 DB 0,0,0,0D,0A
5BD8:0179 DB "Insert p-np diskette for driver A:",0D,0A
5BD8:019D DB "and strike any key when ready ","$"
5BD8:01BC
-R CX
CX 0000
:BC
-N A:NP.COM
-W
Writing 00BC bytes
-Q
```

程序二, NP.COM

初始引导程序,使硬盘能直接启动,又不破坏硬盘上其它的信息。

本程序的思路是先找一块已格式化的软盘,运行功能1,把一台完好的硬盘初始引导程序读到此软盘的0面,0磁道、第1扇区。然后当碰到硬盘主引导程序损坏时,再用此软盘运行功能2,将软盘上的初始引导程序写回硬盘的0面、0柱

面、第1扇区。这就完成了修复过程。

程序附后,请参阅。顺便指出,软盘0面、0磁道、第1扇区留驻有DOS的引导程序。它是格式化时候写入的,与硬盘的初始引导程序不是一回事。因此,运行功能1所用的软盘不应是系统盘,否则往后该系统盘将无法启动。

```

10 DEFINT A-Z
20 DEF SEG=&H1700
30 Z=0
40 CLS
50 LOCATE 3,10:PRINT "1 -- 读硬盘初始引导程序到 A 盘"
60 LOCATE 4,10:PRINT "2 -- 恢复硬盘初始引导程序"
70 LOCATE 5,10:PRINT "3 -- 退出"
80 LOCATE 6,10:PRINT "请选择:"
90 LOCATE 6,18:INPUT C
100 ON C GOTO 1000,2000,3000
1000 DIM X(512)
1010 FOR I=0 TO 33
1020     READ J
1030     POKE I,J
1040 NEXT I
1050 Y=VARPTR(X(0))
1060 CALL Z(Y)
1070 GOTO 40
1500 DATA &H55,&H8B,&HEC
1505 DATA &H8B,&H76,&H06,&H8B,&H1C
1510 DATA &HB8,&H01,&H02,&HB9,&H01,&H00,&HBA,&H80,&H00,&HCD,&H13
1520 DATA &HB8,&H01,&H03,&HB9,&H01,&H00,&HBA,&H00,&H00
1530 DATA &HCD,&H13,&H5D,&HCA,&H02,&H00
2000 DIM X(512)
2005 RESTORE 2500
2010 FOR I=0 TO 33
2020     READ J
2030     POKE I,J
2040 NEXT I
2050 Y=VARPTR(X(0))

```

程序动态加密

山东济宁市信托投资公司微机室 刘士军

在各种管理系统的设计中,为禁止非法操作者调用程序和修改数据,常常在一些重要模块转换处设置操作口令。以前研制的程序中,操作保密都由程序设计者在程序中预先设计好,然后教会操作者,这样,应由操作者对输入数据负责的问题就变得职责难分。理想的作法是:程序设计者在将程序交付使用时,应在各操作模块转换处留有加密空间,然后教会操作者加密的方法。至于控制参数则由操作者自行更换,且使保密口令无法破译,即使程序设计者也无能为力。

设计思想是:首先随时间产生十个十位随机数 $n_1, n_2, n_3, \dots, n_{10}$ (该数随产生时刻不同而不同)作为源密码,然后由操作者自行定义口令字,口令字4~10位。设计的关键问题是如何将口令字溶于密码中,使得不知道口令字就无法查算。我们利用口令字按质数法产生一位校验码 W ,并用 W 和口令字的各位值逐步相加,每步的和除以“模”10,余数再加1,作为被替换的源密码序号 W_i ,以该位口令字替换第 W_i 个源密码的相应位,构成新密码保存。在破译时,按操作者键入的口令字,附上相应校验码,去破译新密码。只有当键入口令字等于定义的口令时,才能破译新密码。这样对于不知口令的任何人,均很难通过口令核对这一关。因程序设计者无法测算哪些源密码中哪些位被替换,故亦不能破译。

例如:在时刻15分30秒,以分秒之积450作

为随机数的根,产生第450,第451,第452,...,第459个随机数 2089967131, 4649457931, 1802480369, 2476731240, 5669011473, 7662285566, 6138937473, 6644393205, 1629523932, 1297459751。我们定义四位口令字为2468,各数码分别乘以质数23,19,17,13等,然后乘积相加,再除以“模”10,余数加1即为校验码。具体计算如下:

$$W = 2 \times 23 + 4 \times 19 + 6 \times 17 + 8 \times 13 = 328$$

$$W = (328 \bmod 10) + 1 = 9$$

由此计算被替换的随机数序号 W_i 。

$$W_1 = ((W + 2) \bmod 10) + 1 = ((9 + 2) \bmod 10) + 1 = 2$$

$$W_2 = ((W_1 + 4) \bmod 10) + 1 = ((2 + 4) \bmod 10) + 1 = 7$$

$$W_3 = ((W_2 + 6) \bmod 10) + 1 = ((7 + 6) \bmod 10) + 1 = 4$$

$$W_4 = ((W_3 + 8) \bmod 10) + 1 = ((4 + 8) \bmod 10) + 1 = 3$$

我们用口令字的第一位数字(左边数起)2替换第2个源密码的第一位,用口令字的第二位4替换第7个源密码的第二位,用口令字的第三位6替换第4个源密码的第三位,用口令字的第四位8替换第3个源密码的第四位,源密码变为新密码: 2089967131, 2649457931, 1808480369, 2466731240, 5669011473, 7662285566, 6438937473, 6644393205, 1629523932, 1297459751。破译时,按键盘键入口令字产生校验码,与键入口令字逐步相加,且按上述原理取

2060 CALL Z(Y)

2070 GOTO 40

2500 DATA &H55,&H8B,&HEC

2505 DATA &H8B,&H76,&H06,&H8B,&H1C

2510 DATA &HB8,&H01,&H02,&HB9,&H01,&H00,&HBA,&H00,&H00,&HCD,&H13

2520 DATA &HB8,&H01,&H03,&HB9,&H01,&H00,&HBA,&H80,&H00

2530 DATA &HCD,&H13,&H5D,&HCA,&H02,&H00

3000 END

出口令字,若取出口令字等于键入口令字,则口令正确,否则,口令不正确。根据上述思想,设计了下面三个程序:程序1(MA1.BAS)为密码产生程序,编译形成EXE文件,调用时产生密码文件;程序2(MA2.BAS)和程序3(MA3.PR)为破译程序,可分别插入用BASIC和FOXBASE语言设计的软件中,验证口令关键字,禁止非法操作。

破译难度分析:因口令定义范围为4~10位,各位级值0~9,共有:

$$10^4 + 10^5 + 10^6 + 10^7 + 10^8 + 10^9 + 10^{10} \\ = 11111110000$$

种口令。假设测试一种口令需要一秒,则全部测试一遍共需352年零4个月。即使口令被别人破译,我们可以利用密码产生程序更改口令参数。另外,为叙述方便,在程序1(MA1.BAS)中,直接将新密码存盘。如果将新密码加密存储,则破译难度更大。

密码产生程序应由操作者严格控制保管,以免他人更改控制参数。万一有人利用密码产生程序,非法改动了控制参数,但是,因非法改动者不知原操作者定义的控制参数,无法使控制参数恢复原样,所以,操作者可通过检验控制参数确知是否有人进行非法操作。笔者将上述方法应用于软件设计中,保密效果非常明显。

程序1

```
100 ' Create MiMa ----MA1.bas
110 DIM Z(10), A$(10), KLW$(10)
120 Z(1)=23:Z(2)=19:Z(3)=17:Z(4)=13
130 Z(5)=11:Z(6)=7:Z(7)=5:Z(8)=3
135 Z(9)=2:Z(10)=1
140 '产生随机密码
150 T1=VAL(MID$(TIME$,4,2))
155 T2=VAL(RIGHT$(TIME$,2))
160 RANDOMIZE T1*T2
170 FOR I=1 TO 10
180 R#=RND(1)*RND(1):IF R#<.1 THEN 180
190 KLW$(I)=MID$(STR$(R#),3,10)
200 NEXT I
210 PRINT "随机密码产生程序"
220 INPUT "定义口令密码(4--10位):", KL$
230 JMA=0 '计算校验码-->W
```

```
240 FOR N=1 TO LEN(KL$)
250 A$(N)=MID$(KL$,N,1)
255 JMA=JMA+VAL(A$(N))*Z(N)
260 NEXT N:W=(JMA MOD 10)+1
270 FOR N=1 TO LEN(KL$)
280 W=((W+VAL(A$(N))) MOD 10)+1
290 KLW$(W)=LEFT$(KLW$(W),N-1)+A$(N)+
MID$(KLW$(W),N+1)
300 NEXT N
310 '保存新密码
320 OPEN "klfile.ma" FOR OUTPUT AS #1
330 FOR I=1 TO 10:WRITE #1, KLW$(I):NEXT I
340 CLOSE #1
350 PRINT "切记口令密码,否则,任何人无能为力!"
360 A$=INKEY$:IF A$="" THEN 360 ELSE END
```

程序2

```
100 ' Pass MiMa ----MA2.bas
110 DIM Z(10), A$(10), KLW$(10)
120 Z(1)=23:Z(2)=19:Z(3)=17:Z(4)=13
130 Z(5)=11:Z(6)=7:Z(7)=5:Z(8)=3
135 Z(9)=2:Z(10)=1
140 '读密码
150 OPEN "klfile.ma" FOR INPUT AS #1
160 FOR I=1 TO 10:INPUT #1, KLW$(I):NEXT I
170 CLOSE #1:CLS
180 '接收键盘口令字
190 LOCATE 2,10:PRINT "验证口令密码程序"
200 LOCATE 4,10:PRINT "输入口令密码",SPC(8)
210 LOCATE 4,22:COLOR 0,0,1
215 INPUT KL$:COLOR 7,0,1
220 JMA=0 '计算校验码-->W
230 FOR N=1 TO LEN(KL$)
240 A$(N)=MID$(KL$,N,1)
245 JMA=JMA+VAL(A$(N))*Z(N)
250 NEXT N:W=(JMA MOD 10)+1
260 KL1$="" '计算控制码
270 FOR N=1 TO LEN(KL$)
280 W=((W+VAL(A$(N))) MOD 10)+1
290 KL1$=KL1$+MID$(KLW$(W),N,1)
300 NEXT N
310 IF KL1$=KL$ AND LEN(KL$)>=4 THEN END
320 GOTO 200
```

在绘图机上绘写汉字的新方法

陕西机械学院 胡培民

摘要: 本文系统介绍了一种直接调用 CC DOS 汉字操作系统的点阵型汉字驱动绘图机在图件上绘写汉字的新方法, 并将利用此法编制的程序附之于后, 并且给以说明, 以供参考。

目前, 各种汉字操作系统的广泛使用已使汉字的屏幕显示和打印易如反掌, 然而对于绘图机来说, 要想绘写汉字就没有那么方便了。汉字的屏幕显示和打印采用的是点阵型汉字, 目前社会上流行的各种汉字操作系统都具有这种点阵型汉字库, 而绘图机采用的是矢量型汉字, 虽然目前在汉化AUTOCAD中已开发出数种矢量型汉字

库供用户选择, 但该字库和系统占用内存太多, 这对一些微机用户来说就难免产生一些问题, 而对于有些用户, 手头可能没有该系统, 有些可能主要工作量是绘制图样, 只需要标注少量的汉字, 如果为此专门设立一个汉字库, 势必带来不必要的麻烦。为此, 本文介绍一种直接调用 CCDOS的点阵型汉字驱动绘图机绘写汉字的新

程序3

```
* Pass MiMa----ma3.prg
clea
z1=23
z2=19
z3=17
z4=13
z5=11
z6=7
z7=5
z8=3
z9=2
z10=1
use klfile
zap
appe from 'klfile.ma' deli
do while .t.
  @ 12.30 say '输入口令:'
  set cons off
  acce to kl
  set cons on
  jna=0
```

```
n=1
do while n<=len(kl)
  n=iif(n<10, str(n, 1), str(n, 2))
```

```
    jna=jna+val(subs(kl, n, 1))*z1
    n=n+1
endd
w=mod(jna, 10)+1
kl1=''
n=1
do while n<=len(kl)
  w=mod(w+val(subs(kl, n, 1)), 10)+1
  goto w
  kl1=kl1+subs(klword, n, 1)
  n=n+1
endd
if kl1>kl.or.len(kl)<4
  ?? chr(7)
else
  exit
endi
endd
use
retu
```

注: 程序 3 (MA3.PRG) 中数据库文件 KLFILE.DBF 的数据结构为:

Field	Field Name	Type	Width	Dec
1	KLWORD	Character	10	
**	Total	**	11	

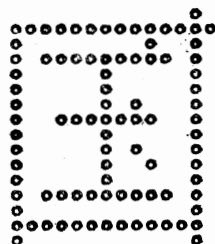
方法,并给出驱动程序,以供参考。

一、原理和方法

CCDOS汉字操作系统均有一个 16×16 的点阵型汉字字库,在系统自举时,它被一次调入内存以供使用。在该操作系统下,当我们输入某汉字的输入码时,则系统自动将输入码转换成机内码,再把机内码转换成相应的汉字字模,最后把字模信息送到屏幕上显示,这一过程我们已经非常熟悉了。为了能使输入的汉字也能由绘图机绘制出来,我们可采取下面的方法:首先想法获取欲绘写汉字的点阵信息,并将点阵信息转换为数字信息,然后用计算机进行判别和计算找出欲绘写汉字的竖、横、斜笔划的长度和相对位置,经过数值变换后送往绘图机将其绘制出来。其详尽方法和步骤如下:

1) 汉字点阵信息的获取

当屏幕设置为中分辨率彩色图形方式时,屏幕点阵为 320×200 ,每个汉字的点阵为 16×16 ,每个汉字的笔划都是由亮的象元一点一点组合而成的。例如:图一所示的就是“国”字的点阵图

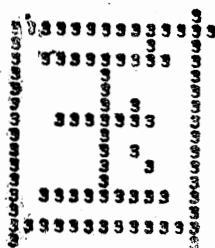


图一

为了获取汉字的点阵信息我们可以采用下面的方法:首先将欲绘写的汉字显示在屏幕上一特定位置,然后对该位置进行扫描,利用BASICA提供的POINT函数对显示的汉字进行采样,对应于屏幕上的灰点,POINT函数值为0,对应于亮点,值为3,将结果存入一个数组中,经采集后得到的数据就如实反映了原来的汉字信息。例如,“国”字经采集后得到的数据排列起来如图二所示,为了以后阐述方便,我们姑且将其称为数值点阵。

2) 绘写汉字

绘写汉字分三部进行,首先绘出汉字的所有

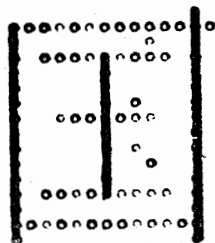


图二

竖笔划,其次绘出所有的横笔划,最后绘出剩下的斜笔划,这样一个汉字的绘写就完成了。下面以“国”字的绘写为例做具体的说明。

① 绘写汉字的所有竖笔划

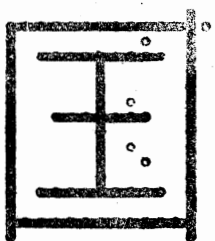
首先,利用循环语句和条件语句扫描汉字的数值点阵,找出所有竖笔划的位置和长度,经一定的数据变换后驱动绘图机将其绘制出来,绘完后,将数值点阵中竖笔划所对应位置的点阵值改为零。图三为第一部绘制的结果。



图三

② 绘写汉字的横笔划

经第一步工作后,字型的数值点阵已不再包含汉字竖笔划的信息,仅仅含横、斜笔划的信息,然后,可仿照第一步根据数值点阵找出汉字所有横笔划的位置和长度,经数字转换后,送往绘图机中绘出,最后再将数值点阵中相应位置点的点阵值改为零,得到新的数值点阵。图四为第二步工作后已绘出的笔划。

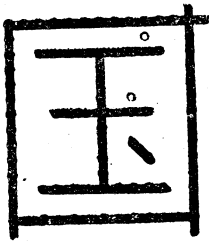


图四

③ 绘写汉字的斜笔划

经过以上工作后,已绘出汉字的所有直线笔划,剩下的不直笔划采用扫描加游动的方法绘出。

扫描自上而下,从左到右进行,当扫描到某一点,其数值点阵值不为零时,则表明该点为斜笔划上的点,此时停止扫描,记下该点的位置,把它作为开始游动的起点。游动的方向根据与该点相邻点的数值点阵值来判定。若找到某一相邻点其点阵值不为零时,则在绘图机上将此两点连接起来,然后,将前一点的数值点阵值改为零,将后一点做为游动的新出发点,重复上面的过程,直到游动到某一点其相邻点的点阵值均为“0”,则游动结束。返回到刚才扫描停止点的下一点继续扫描,若扫描到某处又碰到点阵值不为零的点,则停止扫描,又开始游动,重复上面的过程,直到绘出汉字的全部笔划,图五所示的就是最后结果。



图五

通过以上步骤,一个完整的汉字就绘制出来了。

二、程序说明

根据以上方法,用 BASIC 语言编写的子程序如程序一所示。它可在程序中直接调用,程序二就是调用的一个例子。如果要多次调用该子程序的话,程序一中第 810 句应移到主程序中去。

程序一是针对 SPL—400 型绘图机编写的,用于不同型号的绘图机其绘图命令可能有差异,所以当将此子程序用于其它型号的绘图机时,应将程序中 SPL—400 型绘图机的命令改为该绘图机的命令,而这一工作量是微不足道的。

调用该子程序时,应首先给如下的变量数值:

- ① HX: 要绘写汉字的大小。
- ② HY: 要绘写汉字的纵横比。

③ K: 要绘写的汉字数。

④ XX0,YY0: 要绘写的位置坐标(mm)。

⑤ P\$: 要绘写的汉字。

要绘写的汉字可以用 INPUT 语句从键盘直接输入,也可用赋值语句赋给,也可从数据文件中读取,这可根据使用情况,灵活选取。但不论采用什么方法输入汉字时,汉字间最好不要留空格,要么留偶数子空格,否则绘出的汉字是不完整的。

最后还需要说明的是,使用该程序时,尽量采用汉字点阵比较简单的 CCDOS 系统,如 CCDOS4, 0 等。汉字点阵比较复杂的系统如 CCDOS2·1, 字型虽然美观,但点阵复杂,每个汉字的竖笔划都是加粗的,这样绘制起来比较慢,且清晰度不如前者。

三、结束语

本文主要讨论的是汉字的绘写,实际上,采用此方照样可以绘写西文字符,而且还可以绘写标点符号,制表符,日法假名,俄文字母等各种符号。这一点读者是不难理解的,只要利用区位码象输入汉字一样将它们输入,其处理和绘制方法和汉字是没有两样的。

程序一:

```

800 SCREEN 1:CLS
810 DIM H(16,16,40)
830 FOR KK=1 TO K:LOCATE 1,1,0:PRINT MID$(P$,KK*2-1,2)
840 FOR X=0 TO 16:FOR Y=0 TO 16
850 H(X,Y,KK)=POINT(X,Y)
860 NEXT Y,X,KK
865 HY=HY+HX:XX0=XX0+10:YY0=YY0+10:SCREEN 2
870 FOR KK=1 TO K
880 X0=XX0+(KK*18+HX):Y0=YY0
890 LPRINT "M*X0,Y0
900 FOR X=0 TO 16
910 EY=0
920 Y=EY
930 IF H(X,Y,KK)=0 AND Y<16 THEN Y=Y+1,GOTO 930
940 IF Y=16 THEN 1070
950 SY=Y:EY=SY
960 EY=EY+1
970 IF H(X,EY,KK)<>0 THEN 860
980 EY=EY-1
990 IF EY-SY<3 THEN EY=EY+1,GOTO 920
1000 HSY=Y0-SY*HY:HEY=Y0-EY*HY
1010 HSX=X0+X*HX
1020 LPRINT "M*HSX,HSY
1030 LPRINT "D*HSX,"",HSY HSX,"",HEY
1040 FOR YY=SY TO EY
1050 H(X,YY,KK)=0:NEXT YY
1060 IF EY<16 THEN EY=EY+1,GOTO 920
1070 NEXT X
1080 FOR Y=0 TO 16
1090 EX=0

```

道路选择辅助决策系统的功能与设计

甘肃省计算中心 王雄基 曾光初

该系统是解放军某部“器材发放辅助指挥决策系统”的一个子系统。选择合理的道路，是为了保障器材发放任务的彻底实现，也是器材发放任务的最后完成。这里简要地介绍该系统的主要

功能和设计中的几点技术措施。

该系统首先在屏幕上显示出道路示意图（如图1所示），及道路的结点编号，主要路段的名称等。该部从库房通往发货火车站共有五条公

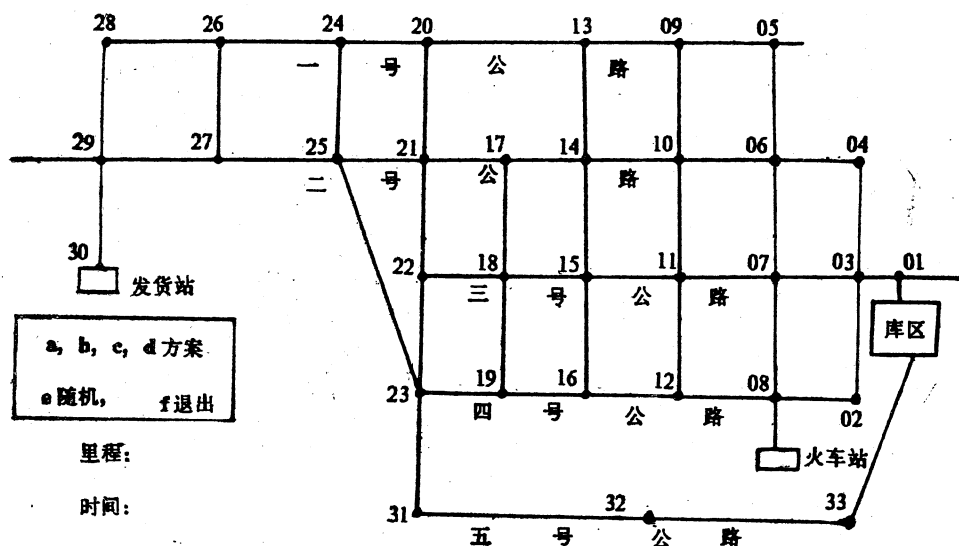


图1 行车路线示意图

```

1100 X=EX:N=J
1110 IF H(X,Y,KK)=0 AND X<16 THEN X=X+1:GOTO 1110
1120 IF X=16 THEN 1260
1130 SX=X:EX=SX
1140 EX=EX+1
1150 IF H(EX,Y,KK)<>0 THEN 1140
1160 EX=EX-1
1170 IF EX-SX<2 THEN EX=EX+1:GOTO 1100
1180 IF SX<4 THEN N=0
1190 HSX=X0+(SX-N)*HX:HEX=X0+(EX+1)*HX
1200 HSY=Y0-Y*HY
1210 LPRINT "M",HSX,HSY
1220 LPRINT "D",HSX," ",HSY,HEX," ",HSY
1230 FOR XX=SX TO EX
1240 H(XX,Y,KK)=0:NEXT XX
1250 IF ZX<16 THEN EX=EX+1:GOTO 1100
1260 NEXT Y:N=J
1270 FOR Y=0 TO 16
1280 EX=0
1290 X=EX:J=1
1300 IF H(X,Y,KK)=0 AND X<16 THEN X=X+1:GOTO 1300
1310 IF X=16 THEN 1468
1320 SX=X:SY=Y:EX=SX
1325 HSX=X0+SX*HX,HSY=Y0-SY*HY
1330 FOR I=-1 TO 1
1340 FOR J=-1 TO 1
1350 DX=SX+I:DY=SY+J
1355 IF I=0 AND J=0 THEN 1380

```

```

1360 IF DX<0 OR DY<0 THEN 1380
1370 IF H(DX,DY,KK)<>0 THEN 1410
1380 NEXT J,I
1385 IF I>1 THEN 1460
1402 LPRINT "W",HSX,HSY,1,1,0,500:GOTO 1460
1410 HDX=X0+DX*HX:HDY=Y0-DY*HY:I=I+1
1420 LPRINT "M",HSX," ",HSY
1430 LPRINT "D",HSX," ",HSY,HDX," ",HDY
1440 H(SX,SY,KK)=0:SX=DX:SY=DY
1450 IF SX<16 THEN 1325
1460 H(SX,SY,KK)=0
1465 IF EX<16 THEN EX=EX+1:GOTO 1290
1468 NEXT Y,KK
1530 LPRINT "H"
1540 RETURN

```

```

200 SCREEN 2
210 PRINT "输入汉字字号 (3, 4, 5, ...) 纵、横比"
220 INPUT HX, HY
230 PRINT "输入要绘写的汉字数"
240 INPUT K
244 PRINT "输入绘写的位置座标 (MM)"
245 INPUT XX0 YY0
250 PRINT "输入要绘写的汉字!"
260 INPUT P$
290 GOSUB 800
300 END

```

路。在正常情况下,系统将五条道路按照交通管制和路况、距离、人与车的流量等情况,综合评价,从优到劣分别确定为A、B、C、D四个等级,作为四个不同的方案,供指挥人员选择。在战争情况下,各种外界因素难以确定,正常通路遭到破坏或临时受阻,这时指挥人员应根据当时掌握的情况,灵活机动地确定出一条道路,可能是迂回曲折地到达目的地。当一条行车路线确定之后,为了进行核实,屏幕上立即采用加色和闪烁方式重现选定路线,以及行车里程、时间等。如果认为不满意,可以进行修改,重新确定,直到满意为止。选定满意的行车路线之后,通过打印机,打印出行车命令,包括路线上所经各结点的汉语名称,行车里程和时间。

该系统是在IBM PC机上实现的。由于需要和其它管理软件联结,为了不因本系统的需要而增添新的系统软件,以及便于操作人员使用和维护,所以系统设计是在CCDOS和BASICA和DBASE II的支持下编写的,因此在设计过程中,对几处技术问题进行了处理。

首先是图形、数据和计算之间的关系。由于所用的硬件配置是采用CGA (ColorGraphics Adapter) 彩色/图形适配器和标准彩色显示器,在高分辨情况下可以产生 640×200 个象元。由于大量的数据是DBASE的库文件,为了直观,示意图必须始终显示在屏幕上,而数值计算则以BASICA语言较为方便。所以整个系统在DBASE状态下运行,在进入BASICA之前应建立有关的外部文件,然后进行作图和计算,直到道路选择完成之后,返回DBASE状态。

其次是图形与字符的相互覆盖问题。由于屏幕上的平面坐标系是笛卡尔坐标顺时针旋转 90° 度,荧光屏垂直向下为y轴,可以显示200个象元,水平方向右为x轴,可以显示640个象元。在道路示意图中,字符(包括汉字)是用PRINT语句产生的,受该语句的影响,字符的位置是在屏幕的行(1—10)与列(0—79)上考虑的;而表示路径的线段是由LINE语句产生,它是在屏幕上的640—200个象元上考虑的。经过精确地计算就可以将字符与线段分别显示在理想的位置上,而避免相互覆盖。

第三是随机选择行车路线时,各路段距离的查询。随机选择行车路线,指挥人员是根据当时掌握的情况,直接在示意图图上依次指定各相邻结点的编号,然后由程序查询相邻两结点之间的距离,所有路段之和就是行车路线的距离。为了查询各路段的距离,可以采用邻接矩阵的办法。但由于结点太多,矩阵过大,大量的零元素浪费掉不少的存储空间,而且会降低处理速度。为了克服这些缺点,首先设计两个数组:一个是结点与路径数组,为二维数组S(),存放相邻两结点的编号和它们之间的距离(如图2所示);二是指针数组,为一维数组XD(),存放键盘依次输入的各结点编号。其次,从键盘输入结点编号时,应是顺序相连,构成一条不重复的道路。因此只要执行下面这段程序,就可以查询各路段的距离,并计算出行车里程。

0	01	5
0	33	15
01	03	3
03	04	6
03	07	7
03	02	5
⋮	⋮	⋮

图2 数组S()

```

行号1  FOR I=1 TO K-1
行号2  FOR J=1 TO N
行号3  IFS(J,1) <> XD(I)OR S(J,2)
        <> XD(I+1) THEN 行号5
行号4  SS=SS+S(J,3):GOTO行号6
行号5  NEXT J
行号6  NEXT I

```

其中,K为结点个数;N为二维数组S()的行数;SS为行车路线的里程。

经过上述处理,程序简捷,结构清晰,满足了用户的要求。在试运行中,基本做到了图文并茂,直观明瞭,处理迅速。

参考文献(略)

一个自动生成表格的FoxBASE程序

石家庄铁道学院 樊金生

在用dBASE III或FoxBASE编制各种应用软件时,经常要打印各种表格。而编制和调试这种打印程序十分费时,编制好的程序不能适应表格的变化。目前虽有一些自动生成表格或自动生成程序的软件,但这些软件一般都是单独的系统,在应用软件中不能灵活地使用,生成后的表格或程序不易于修改。本文提供的FoxBASE程序即可根据用户提供的数据库文件名和一些参数显示或打印出表格,产生表格参数文件,也可根据用户提供的数据库文件名和表格参数文件名自动显示或打印出表格。在打印表格比较多的应用系统中,可将此程序作为一个通用打印模块,对每个要打印的表格只要用文字编辑软件建立一个参数文件,在应用系统的打印程序中只要设计一

个菜单,对每个表格提供要使用的数据库文件和参数文件即可,当某个表格的格式需要修改时,只要用编辑软件修改对应的参数文件,十分容易和方便。此程序简便实用,并可根据需要扩充新的功能。下面说明此程序的使用方法:

1. 参数文件的格式

参数文件(扩展名必须为.txt)的第一行存放要使用的数据库文件,第二行存放各个参数,其他各行存放表头、表中、各字段说明和表尾,型式如下:

olymp.dbf

/o=0/b=1/pl=8/pm=1/lm=0/ym=1/y
=1989/m=12/d=12/cd=.t./cd=国别
="中国".and.名次<4

第24届奥运会中国运动员获奖名单 # head

y.年# m.月# d.日

第# ym.页 # head

项目	分项	姓名	名次	成绩
*1	*2	*3	*5	*6

head
head
head# inter
field
bottom

各参数说明:

/o= 报表输出方式(0表示输出到屏幕,1表示输出到打印机)
/b= 打印的起始记录(隐含为1)
/e= 打印的终止记录(隐含为最后一记录)
/pl= 每页打印记录数(隐含为20)
/pm= 页间空白行数(隐含为1,此值为1时表示页间暂停)
/lm= 左边空白列数(隐含为0)
/ym= 起用页码(隐含为1)
/y/m/d 打表日期(隐含为打表时的系统时间)
/cd= 打印条件(要满足FoxBASE的表达式格式)

每个参数的值都可以是表达式,打印条件应为逻辑表达式,其他参数的值都应为数字型。

表头中关键字解释:

head -----表头部分(可含标题行)
inter -----每两个记录之间分界行(可不要)

field -----字段位置定义和格式定义
bottom-----表尾部分

以上这些关键字必须放在行尾, # head 和 # inter可用于同一行中,但 # head 必须在 # inter之前。

表头部分、表尾部分均可数行,但每行都应标记清楚。在这些行中还可出现表达式(最常用的是变量名),每个表达式前加"#",表达式

后加“.”，且要求表达式为字符型(参数y、m、b、ym除外)。使用形式有如下三种：

- (1) 参数名或变量名：如示例中的#y、#m、#b和#ym；
- (2) 打印控制符：如#chr(27)+“IA”等；
- (3) 对变量进行宏代换，即在“#”之后加一宏代换符“&”：如#&dyzxa。

各字段说明的格式为：

- *0-----记录号(可不要)的打印位置
- *n-----数据库中第n个字段的打印位置
- *n-----数据库中第n个字段的打印位置，当此字段为字数型字段时且字段内容为零时打印空格

2. 程序的使用

此程序的文件名为dybb.prg，执行此程序后，提示输入参数文件名，如果此文件存在，则按此文件提供的参数打印或显示出表格；如果此文件不存在，则提示输入使用的数据库文件名、各个参数、要显示或打印的各字段及长度，然后自动形成参数文件，并显示或打印出表格。例如

有一数据库文件olymp.dbf存放第24届奥运会各项目的成绩，要打印中国运动员获奖牌名单，执行dybb.prg时输入参数文件名及有关参数就可形成上面给出的参数文件示例并显示或打印出表格。

要形成更美观的表格，可用编辑软件建立打印时所用的参数文件，在此文件中设置各个参数和表头、表尾及各字段的格式。在应用系统中将此程序作为通用打印模块时，一种表格用一个参数文件和一个数据库文件，此时可将dybb.prg中的提示输入参数文件及取出数据库文件名等语句去掉，参数文件中不存放数据库文件名，而在程序中根据要打印的表格选定参数文件和数据库文件。例如在财务报表管理系统中要打印多个报表，数据库文件名要根据变量nf(年份)、jd(季度)、mc(单位号)和表名而定；成本表(表名为cb)又分为工程成本表(库中的第1~11个记录)、产品成本表(第12~19个记录)、作业成本表(第20~29个记录)，则可建立一参数文件dycby.txt如下：

```
/o=1/b=bb/e=ee/lm=5
#&dyzxd.      #bg.#&dyzj0.
#&dyhj16.#&dyzxa.  #head
```

会施年汇：03表#head

编制性位：#单位名称。(#xsdwmc.)

19#nf.年#jd.季度

单位：

元#head

成本项目	行次	预算成本	实际成本	降低额	降低率(%)
		1	2	3	4
*2	*3	*4 9,999,999.99	*5 9,999,999.99	*6 999,999.99	*7 99,999.99

```
#head
#head
#head
#head
#head#inter
#field
#field
#bottom
```

在此参数文件中，开始记录号、结束记录号均为变量，打印前应根据所打的表赋所需的值。另外变量bg中存放表的标题，单位名称中存放本级单位名称，xsdwmc存放此表所属单位名称，dyzxd、dyzxa、dyzj0、dyhj16分别存放选D型字、A型字，调整字距和行距的打印控制符(这些变量可存入一内存变量文件中，打印前调入内存，当更换打印驱动程序时只需修改此文件中这些变量的内容)。字段说明的第二行给出了数字型字段的打印格式。当金额单位为千元

时所用的参数文件dycbqy.txt的内容与dycby.txt基本相同，只数字型字段的打印格式和表头稍有区别。打印成本表的程序如下：

.....

***变量NF、JD中分别存放报表的年份和月份，MC中存放单位代号，

***“CB”为成本表的表名，变量fdb中存放要打印的数据库文件名

```
fdb="CB"+NF+"-"+JD+mc
```

***变量jedw为表中金额单位的标志，

jedw= "1" 表示单位为千元, 参数文件为
 ***dycbqy.txt; 否则表示单位为元,
 参数文件为dycby.txt.
 dycswj=iif(jedw= "1", "dycbqy",
 "dycby")
 bgmm=1
 do while bgmm <=3
 *** 参数文件中要用到变量bg,bb,ee,
 分别作为表名、开始记录号、结束记录号。

```
DO CASE
CASE bgmm=1
  bg= "工程成本表"
  bb=1
  ee=11
CASE bgmm=2
  bg= "产品成本表"
  bb=12
```

```
ee=19
CASE bgmm=3
  bg= "作业成本表"
  bb=20
  ee=27
ENDCASE
wait "打印" +bg+"吗(Y/N):" to DY
IF UPPER(DY)="Y"
  DO DYBB
ENDIF
bgmm=bgmm+1
enddo
```

因程序清单太长而篇幅有限, 此处略去。需此程序者, 请寄10元工本费至本刊编辑部索取。

(在本文完成过程中参考了国家科委信息研究所朱崇君同志的字表编辑软件 cced 中打印表格的设计思想, 在此表示感谢)

介绍一例简易实用的键控绘图程序

湖北省荆州师专计算机室 瞿 波

下面是笔者编写的一例键控绘图程序。由于巧妙地进行了一些分支、转向语句的替换, 使得用一程序即可完成通常需大块头多行程序才能完成的功能。

该程序将高分辨率第二页作为绘图页, 程序运行后, 在 (0, 0) 坐标处出现一个光点, 通过 I、J、K、M 键, 分别控制光点上、左、右、下的移动。L 键是画与不画的“开关”, 即按一次 L 键, 处于不画的状态, 再按一次 L 键, 又回到画的状态。在不画状态, 可移动光点擦除已画的点。若想结束绘图, 按 N 键即可。

程序设有错误处理功能, 当某一按键使 X 或 Y 的值超出范围时, 程序中会自动予以修正, 不致使绘画中断。当按了 I—N 六个键以外的其它键, 程序将不予理睬, 而等待键入有效的键后,

程序才会作出反应。所以, 你尽可放心的操作键盘, 去进行绘图和电脑画的创作。

程序在 APPLE II PLUS 及兼容机上通过, 清单如下:

```
ILIST
HGR2 :X(2) = - 1:X(3) = 1:Y(1)
) = - 1:Y(5) = 1:D = 3: FOR
I = 1 TO 2 STEP 0: HCOLOR= D
: HPLLOT X,Y:X = X + X(A):Y =
Y + Y(A):X = X + (X < 0) - (
X > 279):Y = Y + (Y < 0) - (
Y > 191): HCOLOR= 3: HPLLOT X
,Y: GET A$:A = ASC (A$) - 7
2:A = (A > 0) * (A < 7) * A:
D = ABS (D - (4 = A) * 3): IF
A < 6 THEN NEXT
```

IBM PC/XT 及其兼容机故障诊断仪问世

东南大学研制成功 IBM-PC/XT 及其兼容机故障诊断仪, 该机应用特征分析、特征激励、在线和离线测试等一系列新技术: 电路采用大规

模集成电路, 体积小、成本低、功能多, 它能自动测试、分析、诊断微机多种故障。可称为修理微机的“万用表”。 (李相彬)

单缝衍射实验的计算机模拟分析

空军第一航空技术专科学校计算机教研室 李志伟

摘要: 本文介绍了光的单缝衍射实验的一种计算机模拟软件, 该软件不但能对光的单缝衍射进行模拟实验, 而且能对衍射效果进行分析。具体分析采用菲涅耳半波带法, 程序设计采用 Turbo Pascal语言, 在Turbo Pascal4.0系统上实现。

一、引言

自然界中, 光不但有反射和折射现象, 而且还有干涉和衍射现象。在光学教学中, 光的反射和折射实验容易做, 而且也比较容易解释, 而干涉和衍射实验要求条件就比较高, 而且对实验产生的现象也不易直接分析, 为这部分的教学带来了一定的困难, 为此我们编制了这套软件, 用计算机模拟实验并进行分析, 不但节省人力财力, 而且生动逼真, 提高了实验者的兴趣, 起到了一般实验达不到的效果。由于时间仓促, 我们仅编制了光的单缝衍射实验部分的内容, 其它部分正在逐步完善。

二、开发工具

我们要编制的软件, 其中绝大部分是图形演示, 为了使演示更加生动, 要在软件中实现一些特殊效果, 为此我们选用 Turbo Pascal 语言作为软件开发工具。

Turbo Pascal 是这些年来国际上流行最广泛的高级程序设计语言之一, 它具有编辑快、编译快、目标代码执行速度快等特点, 是目前PC机上通用的一种Pascal语言。它是一种高度集成化的系列软件, 它集编辑、编译、连接、排错和窗口环境于一体(3.0以前的版本无窗口功能), 提供了友善的多窗口软件开发环境, 用户可以在该系统下做所有的事情——编辑、编译、连接、排错、执行, 并且可产生.EXE(或.COM)文件, 所有这些将大大提高程序的开发效率。另外, Turbo Pascal 与其它语言交互的能力极强, 尤其是它可以和 Turbo Prolog 语言灵活交接, 为将来编制高级智能软件提供了良好的开发环境。

从Turbo Pascal 1.0到5.0, 每版都有改

进, 从4.0开始, 其源程序便可突破64K的限制, 这为我们编制大程序提供了可能。另外, 4.0版还提供的多窗口下拉菜单环境和帮助环境, 以及更加丰富的内部子程序集, 都为我们快速、高效地开发软件提供了可能, 为此, 我们选用 Turbo Pascal4.0 作为软件工具。

三、软件概况

(一) 软件功能: 本软件能对单色光的单缝衍射实验进行模拟, 并可用菲涅耳半波带法对衍射效果进行分析。

(二) 运行环境: 硬件——IBM PC/XT
及其兼容机

软件——DOS 2.0或更高
的版本

(三) 程序: 源程序采用全屏幕编辑, 共500余行, 占内存14K, 编译文件53K。

四、设计原理

如附图, 设从单缝AB以一定入射角传播的子波射线经透镜会聚于屏幕上的Q点, 在Q点应该出现明条纹或暗条纹, 我们用菲涅耳半波带法进行分析, 即把单缝的波面分割成若干面积相等的波带的方法, 来讨论衍射条纹。过B作一平面BC垂直于AC, 则由平面BC上各点到达Q点的光程都相等。这样, 从AB面发出的各射线在Q点的相位差就等于它们在BC面上的相位差, 即其相位差只发生在AB面转向BC面的路径之间。设AC恰好等于单色光波长的整数倍, 即

$$AC = K \cdot \lambda / 2 \quad K = 1, 2, 3, \dots$$

作彼此相距为 $\lambda/2$ 的平行于BC的平面, 把AC分成K等分, 同时这些平面单缝上波面AB切割成K个波带。图例中划分的波带数 $K=3$ (The number of wave band is 3), 则波面AB

被分成3个面积相等的波带,从每个波带发出的子波的强度可以认为是相等的,每个波带上对应点所发出的子波光线到达Q点处的光程差均为 $\lambda/2$ (这种波带叫半波带),从整个波带来说,前两个波带上各对应点发出的子波到达Q点处后相互干涉抵消,而最后半波带上的子波还有被抵消,因而在衍射屏幕上将出现明条纹,光强度曲线将出现峰值。同理,当K为其它奇数时效果一样,只是明条纹的宽度和强度将发生变化。当K为偶数时,波面AB上各子波到达Q点后相互干涉抵消,衍射屏幕上将出现暗条纹,光强度曲线上对应点处于波谷。

特殊地,当 $AC=0$ 时,Q点处呈现中央明条纹,对应于光强度曲线上的最大峰值点。

五、软件特点

(一) 使用方便

为了使广大用户能方便地使用本软件,源程序已在Turbo Pascal 4.0系统上编译成了编译文件OPTICS.EXE,用户不需要了解Turbo Pascal语言的知识,在DOS提示符下直接键入文件名即可运行本软件。

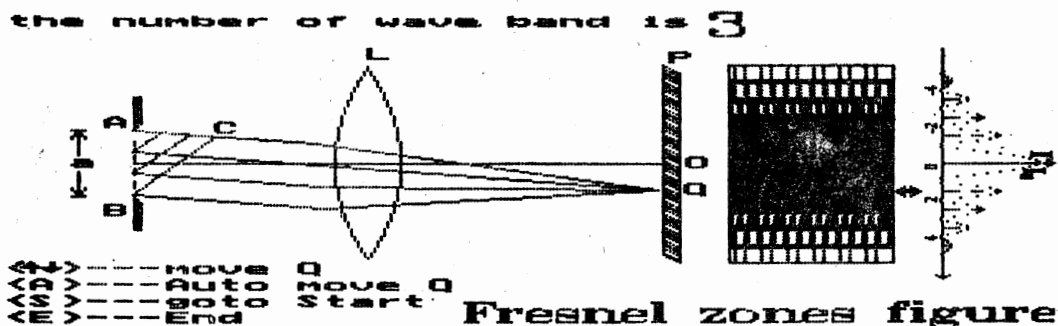


图1 菲涅耳波带图

(三) 附加效果

色、声、图、字相互配合,从心理上赢得了用户。

该系统以图形演示为主,各种颜色相互配合,起到了一般实验所达不到的效果。另外,对系统中的文字说明及提示信息采用了多种字体(矢量字体、小字体和一般的点阵字体等)的相互配合,并根据具体情况对相应的字体进行一定

另外,在实验的每一步都有相应的提示信息,用户一旦进入系统,不需指导手册便可在系统提示下完成本实验。

(二) 生动逼真

在软件中我们采用多幅画面对单缝衍射实验的内容进行说明和模拟,从衍射装置、衍射效果到结果分析分别演示,实现了实验、分析的全过程。

对单缝衍射实验的结果分析我们采用菲涅耳半波带法,在菲涅耳分析图(如附图1)中实现了如下效果:

1. 动画效果。用户可以在屏幕提示下按上下箭头键($\uparrow \downarrow$)移动经单缝AB射来的一束光线,分析它会聚到屏幕上一点Q应该出现的明暗条纹和光强度,也可按A键由系统自动移动光线束进行分析。

2. 各种关系的协调对应。分析经单缝AB射来的一束光线会聚到屏幕上某点Q的情况,则在光路图中将显示出相应的波带划分情况,并将所划分的波带数显示于屏顶,与此同时,屏幕右部的左右箭头即时地指出此刻Q处出现的明暗条纹及光强度分布曲线的相应点。

的放大(这在其它系统上不易实现),此外,系统中还加入了音乐效果。一旦进入系统,就会使人感到自己仿佛不是在做一项高难度的光学实验,而是在欣赏一场迷人的书法大赛,陶醉一曲动人的交响乐章,领略五彩缤纷的自然风光。寓教育于艺术之中,陶冶了实验者的情操,提高了实验者的兴趣,起到了特殊的效果,这正是我们软件设计的目的之一。

APPLE II 机上超长程序动态调试的实现

国防科技大学电子技术系 李 勇

摘要 本文介绍了一个简单实用的方法,使长达32K以上的程序能在APPLE—II 微机上进行动态仿真调试。

APPLE—II 主机上有 48K RAM 和 12K ROM, 其中 RAM 区分配如下:

系统占用缓存区 (\$0000~\$03FF);

字符及低分辨率图形显示缓存区 (\$0400~\$0BFF);

高分辨率图形显示缓存区(\$2000~\$5FFF)

用户程序占用区 (\$0C00~\$1FFF, \$600~\$BFFF)。

APPLE—II 微机上存在彼此不兼容的 DOS 和 CP/M 两种操作系统。DEBUG 仅对 CP/M 文件有效,而固化于 EPROM 中的二进制文件只能在 DOS 状态下读入内存。再加上用

户区被分为长度为 5K 和 24K 的两段,使得当用户程序大于 24K 时的动态调试遇到麻烦。

在实际软件开发中,我们遇到 32K 程序及 12K 数据区的超长程序的动态调试仿真问题。利用图形显示缓存区及两种操作系统下文件的转换,我们摸索出了能在 APPLE—II 上对上述程序进行动态调试的方法。

具体实现步骤如下:(划线者表示键入)

1. 先将分别存于 4 片 2764 的 32K 文件读入内存指定位置,并联成一片。

即:插 EPROM 读写卡于扩展槽,开机启动 DOS 3.3, 进入 DOS 状态。

六、技术处理

(一) 动画设计

我们分析衍射屏上不同位置的衍射效果,就要移动相应的一束光线,同时单缝 AB 上的波带划分情况也要变化,如果简单地在屏幕上移动光线束,则在光线束扫过的区域将出现一片涂色区(光线颜色),从而分不清当前的光线束,为此在每移动一次光线时,先对原来的光线束进行一些技术处理,将其“擦去”后再画下一束光线,从而实现了简单的动画效果。

(二) 画面处理

在实验的不同阶段,屏幕上出现了相同的画面(如衍射图象),为减少重复性工作,我们采用了 Turbo Pascal 的存图和取图技术,大大提高了软件的开发效率。

(三) .EXE 文件的产生

由于我们的软件中用到了 Turbo Pascal 4.0 系统的图形功能及特殊字体的显示功能,所以运行该软件,即使编译成 EXE 文件也要用到

Turbo Pascal 4.0 的图形驱动程序 (BGI) 及字体文件 (.CHR),而这些程序又分别放在磁盘上的不同文件里,如果用户使用我们的软件,则必须带上相应的这些文件,这与我们软件设计要高度集成化的要求格格不入,为此我们采用了特殊的连接技术,在源程序生成 .EXE 文件的同时将相应的图形驱动程序和字体文件连接到 .EXE 文件中,这样,最后产生的 .EXE 文件便可单独在 DOS 下运行,方便了用户。

主要参考文献

1. 《物理学》(下册) 南京工学院等编 马文蔚等改编 高等教育出版社
2. 《大学物理》(振动、波动与光学) 杨仲者等编 人民教育出版社
3. 《Turbo Pascal 4.0 使用手册》 中国软件技术有限公司 软件交易中心
4. 《Turbo Pascal 3.0 指南、实例和经验》 唐常杰等编

J PR井2 (或相应扩展槽号)

将出现 COMPUTER EPROM PROGRAMMER

- (1) 2708
- (2) 2716
- (3) 2532
- (4) 2732
- (5) 68764
- (6) 2564
- (7) 2764

ROM SIZE? 7 选2764

- (P) PROGRAM
- (R) READ
- (C) CHECK
- (V) VERIFY

(M) MONITOR? R 读入内存
START ADDRESS? \$1000 第一片

2764指定首址\$1000.重复上述步骤,第二、三、四片2764读入首地址分别为\$3000, \$5000, \$7000.

↑ C 回DOS状态

2. 以二进制码形式存盘, 文件名为fn.

] BSAVE fu, A\$1000, L32767

执行后, \$1000~\$BFFF 共32K 长度存入文件.

3. 转为CP/M文件.

启动CP/M系统

A> APPOS

B: fn=A: fn 将A盘中的DOS

文件转存于B盘中的CP/M文件.

↑ C 退出APDOS, 返回CP/M 状态.

4. 此时虽已是CP/M文件, 但不能直接运用DEBUG, 需重新编辑文件, 产生能够用DEBUG调试的文件fn. BAK.

A> ED fn

* #A

* E 此时将产生 fn. BAK 文件, 长度为33k.

* ↑ C 退出ED状态, 返回CP/M.

5. 用DEBUG进行动态调试.

A> DEBUG fn. BAK

再利用DEBUG状态二级命令, 便可对原文件进行动态仿真调试, 其中内存\$9000~\$BFFF作为12K RAM数据存放区.

对于长于32K的文件, 此法仍然有效. 但由于BSAVE fn. A首addr, L1命令中l最长为32K, 对长于32K的程序应用两次BSAVE命令.

参考文献:

1. 戴述祖, 都志杰: 《APPLE—II 微型计算机系统实用操作 100 例》, 上海科教出版社, 1986年9月
2. 刘诚方: 《APPLE—II 操作与应用》, 天津出版社, 1986年9月

软件汉化及一个辅助工具

浙江师范大学计算机中心 杨传斌

汉化一个软件一般要做二项工作, 一是使它在汉字系统下能显示和打印汉字, 二是把原来的英文提示改成中文.

汉化一个软件的提示信息一般可采用程序调试工具DEBUG来完成, 但这种方法费时、麻烦, 更重要的是查找提示信息所在的段区不容易, 为此笔者参考有关资料, 用TURBO Pascal 语言写成一个提示信息汉化的辅助程序,

使用方便, 汉化速度快, 特别适合不熟悉DEBUG程序使用的人员, 为了使用普及, 下面把该程序的BASIC版本介绍给广大读者.

程序中10—150行是初始化部分, 其中字符串数组Y存放原提示信息, X存放新提示信息, D数组存放提示信息在软件中的位置, 输入要汉化的文件名, 从第几字节开始, 到第几字节为止, 并需考虑一个大型软件可以分几次汉化, 也

考虑到提示信息一般在程序的尾部, 可以先从后面开始查找提示并汉化; 160—340行是原来提示信息的提取, 它是根据软件中提示一般以“\$”, CHR\$(0), 回车(CHR\$(13))等特殊字符结尾, 并假定提示信息到至少有一个单词的长度在4个字符以上, 以减少误操作, 其中310—340行对每一提示进行回找, 这是因为如提示以两个字符或字符组成的单词开头, 程序提取将它忽略掉了, 通过回找可形成完整的信息提示行, 例EDLIN.COM中有这样一条提示, End of

INPUT file., 其中“End”和“of”两个词的提取就是由这部分来完成的。370—670行是对提示的编辑、修改、打印, 用法可见源程序中的370、380两行; 680—730行是对新的提示存盘, 退出BASIC就可使用新的提示信息的软件, 如果对提示还有不满之处, 还可使用本程序修改, 因为本程序对中文提示也可处理(230—250), 直到使用满意为止。

若把本程序在编译后运行, 速度与用TURBO PASCAL写的程序相差无几。

```

10 CLS:KEY OFF
20 DIM Y$(500),X$(500),D(500)
30 AS="ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz"
40 BS=AS+CHR$(34)+"!@%^&*()_+-={}[]\|'<.>?/, "
50 WS=CHR$(13)+"$"+CHR$(0)+CHR$(10)
60 INPUT "请输入文件名: ",FF$
70 OPEN "R",1,FF$,1:FIELD 1,1 AS DS
80 IF LOF(1)=0 THEN CLOSE:KILL FF$:GOTO 60
90 INPUT "请输入从第几字节开始": HH
100 IF HH=0 THEN HH=1
110 INPUT "到第几字节为止": HH1
120 GET 1,HH
130 FL=LOF(1)
140 IF FL>HH1 AND HH1<>0 THEN FL=HH1
150 L=0:N=0
160 WHILE N<FL-HH
170   S1=0:S2=0:SD=HH+N+1:TS=""
180   N=N+1:GET 1:C=ASC(D$)
190   LOCATE 5,1:PRINT USING "####":N;
200   IF INSTR(AS,D$)=0 THEN 230
210   TS=TS+D$:S1=S1+1:S2=0
220   IF S1>3 THEN 260 ELSE 180
230   IF C<161 THEN 300
240   TS=TS+D$:S2=S2+1:S1=0
250   IF S2>3 THEN 260 ELSE 180
260   WHILE (C>31 AND C<128 OR C>161) AND LEN(TS)<255
270     N=N+1:GET 1:C=ASC(D$):IF INSTR(WS,D$)=0 THEN TS=TS+D$ ELSE 290
280   WEND
290   IF LEN(TS)>5 THEN L=L+1:D(L)=SD:Y$(L)=TS
300 WEND
310 FOR I=1 TO L
320   GET 1,D(1)-1:IF INSTR(BS,D$)<>0 THEN D(1)=D(1)-1:Y$(1)=D$+Y$(1):GOTO 320
330   IF INSTR(AS,MID$(Y$(1),1,1))=0 AND ASC(MID$(Y$(1),1,1))<161 THEN D(1)=D(1)+1:Y$(1)=RIGHT$(Y$(1),LEN(Y$(1))-1):GOTO 330
340 NEXT I
350 FOR I=1 TO L:X$(I)=Y$(I):NEXT I
360 PRINT
370 PRINT "回车修改本行,↑上行,↓下行,←取左一个,→取右一个,<Del>删一个"
380 PRINT "<End>结束,<PgUp>定位前10行,<PgDn>定位下10行,<P>打印":PRINT
390 LL=L:L=1
400 PRINT:PRINT I:PRINT "原提示: ";Y$(L)
410 PRINT "新提示: ";X$(L)
420 AS=INKEY$:IF AS="" THEN 420
430 K=ASC(AS):IF LEN(AS)=2 THEN 580
440 IF K=13 THEN 520
450 IF AS<>"P" AND AS<>"p" THEN 400
460 PRINT "打印(1_原提示; 2_新提示; 3_对原打印; 0_退出)":AS=INPUT$(1)
470 LPRINT
480 IF AS="1" THEN FOR I=1 TO LL:LPRINT "原提示(";I;"): ";Y$(I):NEXT I
490 IF AS="2" THEN FOR I=1 TO LL:LPRINT "新提示(";I;"): ";X$(I):NEXT I
500 IF AS<>"3" THEN 400
510 FOR I=1 TO LL:LPRINT "原提示(";I;"): ";Y$(I):LPRINT "新提示(";I;"): ";X$(I):NEXT I:GOTO 400
520 PRINT "修改成: ";:LINE INPUT TS
530 IF TS="" THEN 570
540 IF LEN(TS)>LEN(X$(L)) THEN TS=LEFT$(TS,LEN(X$(L)))
550 IF LEN(TS)<LEN(X$(L)) THEN TS=TS+SPACES(-LEN(TS)+LEN(X$(L)))

```


如何用2716EPROM代替16FDC板中的74918ROM

浙江水产学院 叶富乐

摘要: 本文叙述了如何根据16FDC板的工作原理列出损坏的原74918ROM真值表,并根据真值表把数据写入2716EPROM,实现用2716代替74918的功能。

16FDC板是CROMEMCO微机的一块软磁盘接口板。在我们维修中曾碰到此板中的74918ROM损坏,造成机器无法起动。由于是早期产品,目前市场上难以买到此片子。为了使机器能正常运行,我们就设法用手头上有的2716来代替74918,其结果是令人满意的。

要用2716代替74918,关键是要设法了解74918芯片中的原始数据。根据16FDC板的工作原理,我们知道:系统按端口地址 $30H \sim 34H$ 执行输入或输出指令的时候,74918的 DO_8 端将会输出低电平 $30H \sim 34H$ 信号。在系统按端口地址 $30H \sim 33H$ 执行输入指令的时候,这个74918芯片的 DO_2 端将输出低电平 \overline{RE} 信号。在系统按端口地址 $30H \sim 33H$ 执行输出指令的时候,这个74918芯片的 DO_1 端将输出低电平 \overline{WE} 信号。在系统按端口地址 $34H$ 执行输入或输出指令的时候,这个74918芯片的 DO_1 端或 DO_7 端将分别输出低电平 $\overline{IN34H}$ 或 $\overline{OUT34H}$ 信号。在系统按端口地址 $40H$ 执行体选输出指令的时候,这个74918芯片的 DO_8 端将输出低电平 \overline{BANK} 信号。在系统按端口地址 $00H \sim 09H$ 对这块16FDC板软磁盘控制板上的TMS—5501芯片执行输入或输出指令的

时候,这个74918芯片的 DO_4 端将会输出低电平 \overline{UART} 信号。原电路如图1所示。

当我们知道了它的工作原理后,我们就可以根据它的工作原理列出它的真值表,见表1。这儿需要说明的是我们将2716的 A_7 脚分配给 \overline{SINP} , A_8 脚分配给 \overline{SOUT} , $A_9 \sim A_{10}$ 2716的输出 D_0 脚分别还是分配给对应系统地址 $A_9 \sim A_{10}$ 分配给 \overline{WE} , D_1 分配给 \overline{RE} , D_2 分配给 \overline{BANK} , D_3 分配给 \overline{UART} , D_4 分配给 $\overline{IN34H}$, D_5 分配给 $\overline{OUT34H}$, D_7 分配给 $30H \sim 34H$ 。当然读者可以有自己的另外分配。现在我们可以把表1的左边16进制表示的各数作为2716的地址,表1右边16进制表示的各数作为2716的输入数据,通过EPROM写入器把这些数据写入2716EPROM。

完成以上工作后,现在2716中已存放了原74918ROM中的原始数据,我们可以如图2所示把2716各脚与74918各脚一一对应接好。注意:我们写入2716时把它的地址线 A_0 与 A_{10} 作为“0”,因此我们这儿把它接地,反之要接+5V电源。

以上所谈的代换方法,读者可以举一反三,应用到别的机器类似的故障ROM代换中去,关键就是设法知道原ROM中的信息。

```
560 XS(L)=TS
570 L=L+1:GOTO 400
580 K=ASC(MID$(AS,2))
590 IF K=79 THEN 680
600 IF K=72 AND L>1 THEN L=L-1:GOTO 400
610 IF K=89 AND L<LL THEN L=L+1:GOTO 400
620 IF K=75 AND LEN(XS(L))<254 THEN GET 1,D(L)-1:XS(L)=DS+XS(L):D(L)=D(L)-1:GOTO 400
630 IF K=77 AND LEN(XS(L))<254 THEN GET 1,S(L)+LEN(XS(L))+1:XS(L)=XS(L)+DS:GOTO 400
640 IF K=83 AND LEN(XS(L))>1 THEN XS(L)=MID$(XS(L),2,LEN(XS(L))):D(L)=D(L)+1:GOTO 400
650 IF K=73 THEN L=L-10:IF L<1 THEN L=1
660 IF K=81 THEN L=L+10:IF L>LL THEN L=LL
670 GOTO 400
680 PRINT "全部正确 (Y/N)";QS=INPUT$(1):IF QS<>"y" AND QS<>"Y" THEN 400
690 PRINT "请稍等"
700 FOR I=1 TO LL:FOR J=1 TO LEN(XS(I))
710 LSET DS=MID$(XS(I),J,1):PUT 1,D(I)+J-1
720 NEXT J:NEXT I
730 CLOSE #1:END
```

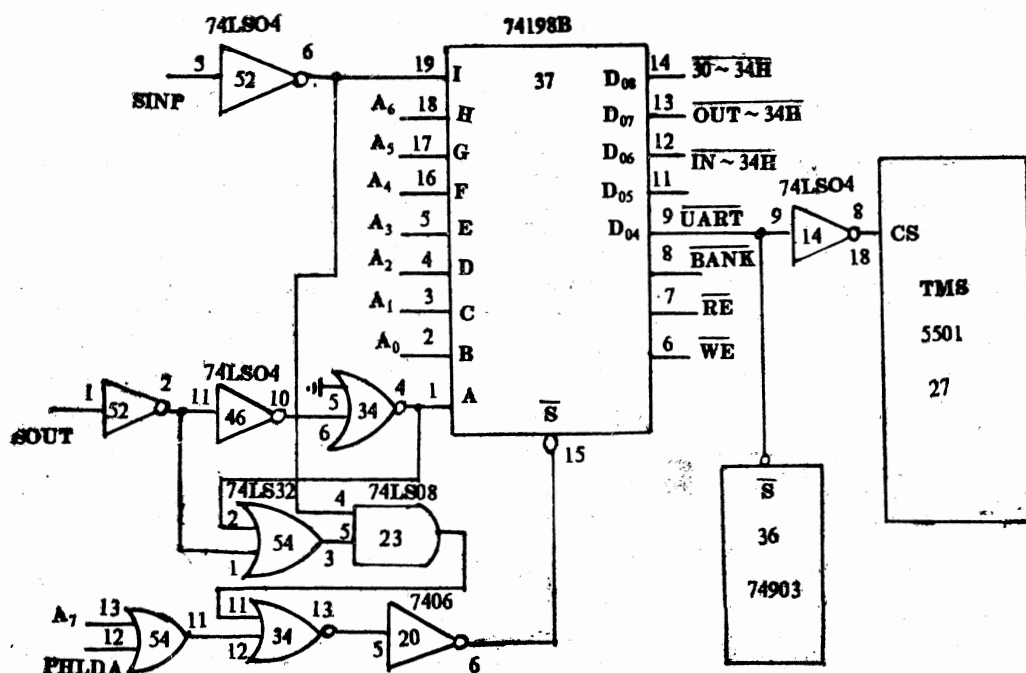


图1

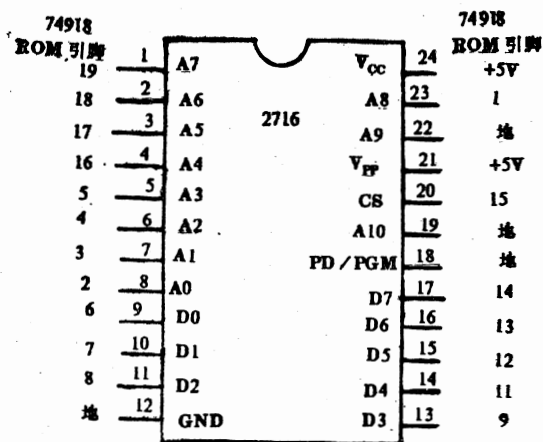


图2 2716与74918各对应引脚

电力建设工程信息管理系统研制成功

机电部28所和华能国际电力开发公司南京分公司共同研制成功电力建设工程信息管理系统并已投入运行。该系统采用长城0520 DH微机为工作站, 程序语言采用汉字DBASE-III PLUS, 全系统由计划、工程、供应、财务、人事等10个子系统、9个工作站组成。该系统功能齐全, 能完成设备合同、设备备件、台帐、仓库、材料供应、运输等项目的管理, 还能对国内外设备合同执行情况进行统计, 打印出表报。它的网络功能多, 包括有邮件业务、共享打印、文件共享、磁带机共享、磁盘空间共享等。

(李相彬)

表一 真 值 表

十六进	SOUT SINP A ₈ A ₇ A ₆ A ₅ A ₄ A ₃ A ₂ A ₁ A ₀										WE RE BANK UART				IN _{34H} OUT ₃₄ 30H				十六进
制表示	A ₈	A ₇	A ₆	A ₅	A ₄	A ₃	A ₂	A ₁	A ₀	D ₀	D ₁	D ₂	D ₃	D ₄	D ₅	D ₆	D ₇	制表示	
130	1	0	0	1	1	0	0	0	0	1	0	1	1	1	1	1	0	0BE	
131	1	0	0	1	1	0	0	0	1	1	0	1	1	1	1	1	0	0BE	
132	1	0	0	1	1	0	0	1	0	1	0	1	1	1	1	1	0	0BE	
133	1	0	0	1	1	0	0	1	1	1	0	1	1	1	1	1	0	0BE	
134	1	0	0	1	1	0	1	0	1	1	0	1	1	1	0	1	0	0FA	
0B0	0	1	0	1	1	0	0	0	0	0	1	1	1	1	1	1	0	07E	
0B1	0	1	0	1	1	0	0	0	1	0	1	1	1	1	1	1	0	07E	
0B2	0	1	0	1	1	0	0	1	0	0	1	1	1	1	1	1	0	07E	
0B3	0	1	0	1	1	0	0	1	1	0	1	1	1	1	1	1	0	07E	
0B4	0	1	0	1	1	0	1	0	0	1	1	1	1	1	1	1	0	0FC	
0C0	0	1	1	0	0	0	0	0	0	1	1	0	1	1	1	1	1	0DF	
100	1	0	0	0	0	0	0	0	0	1	1	1	0	1	1	1	1	0EF	
101	1	0	0	0	0	0	0	0	1	1	1	1	0	1	1	1	1	0EF	
102	1	0	0	0	0	0	0	1	0	1	1	1	0	1	1	1	1	0EF	
103	1	0	0	0	0	0	0	1	1	1	1	1	0	1	1	1	1	0EF	
104	1	0	0	0	0	0	1	0	0	1	1	1	0	1	1	1	1	0EF	
105	1	0	0	0	0	0	1	0	1	1	1	1	0	1	1	1	1	0EF	
106	1	0	0	0	0	0	1	1	0	1	1	1	0	1	1	1	1	0EF	
107	1	0	0	0	0	0	1	1	1	1	1	1	0	1	1	1	1	0EF	
108	1	0	0	0	0	1	0	0	0	1	1	1	0	1	1	1	1	0EF	
109	1	0	0	0	0	1	0	0	1	1	1	1	0	1	1	1	1	0EF	
080	0	1	0	0	0	0	0	0	0	1	1	1	0	1	1	1	1	0EF	
081	0	1	0	0	0	0	0	0	1	1	1	1	0	1	1	1	1	0EF	
082	0	1	0	0	0	0	1	0	0	1	1	1	0	1	1	1	1	0EF	
083	0	1	0	0	0	0	0	1	1	1	1	1	0	1	1	1	1	0EF	
084	0	1	0	0	0	0	1	0	0	1	1	1	0	1	1	1	1	0EF	
085	0	1	0	0	0	0	1	0	1	1	1	1	0	1	1	1	1	0EF	
086	0	1	0	0	0	0	1	1	0	1	1	1	0	1	1	1	1	0EF	
087	0	1	0	0	0	0	1	1	1	1	1	1	0	1	1	1	1	0EF	
088	0	1	0	0	0	1	0	0	0	1	1	1	0	1	1	1	1	0EF	
089	0	1	0	0	0	1	0	0	1	1	1	1	0	1	1	1	1	0EF	

计算机决策支持系统在铁路站场中的应用

西南交通大学计算机系 朱怀芳 尹治本 楼新远

引 言

在现代化管理中,决策是其中最为重要的一个部分。本世纪40年代开始,决策理论得到了重大发展,决策科学更趋于数学化、模型化和程序化;与此同时,计算机技术也产生了巨大进步。

对于一个复杂的社会,经济、生产等大型系统来说,决策是一个十分复杂的过程,面对大量的经济、政治、科技信息以及各种不确定因素,要作出科学、合理的决策是非常困难的。决策支持系统就是辅助决策者进行决策的应用软件,它是数

学化、模型化、信息化与计算机化的产物。

决策支持系统是一门较新的计算机应用技术,从理论到实践都还处于研究和发展阶段。本文介绍了决策支持系统的概念,讨论了决策支持系统的结构及设计和研制方法。同时运用决策支持系统、软件工程、程序设计技术、数据采集技术、管理决策理论、运筹学、图论、计算理论等技术和理论,针对一个实际的系统——“铁路站场调度计算机辅助决策系统”进行了分析、设计和研制,并在 IBM PC/XT 机上实现了该系统。

该系统主要用于站场机车调度辅助决策及站场信息采集处理。系统自动采集和通过人工输入方式采集站场机车、车辆、股道等信息,根据站场状况和生产要求作出优化的机车作业阶段计划,并可为调度人员提供站场状况查询和信息管理。该系统以提高运输设备有效利用率,保障安全生产及加速运输管理现代化而带来较大的社会效益。这类系统国内尚未见到。

一、决策支持系统概念

1. 决策概念

决策就是对未来的行动方向、目标、方案、原则和方法所作的决定,它是一个过程,又是一种选择。

(1) 决策的步骤和内容

决策目标的确定,在整个决策过程中至关重要,它直接关系到全局的成败。确定目标应注意以下几个方面:

- 确定目标的根据
- 目标的需要与实现的可能
- 目标必须具体明确
- 确定目标的约束条件
- 多目标问题

(2) 拟定各种可能方案

(3) 选定方案的标准

选择方案应综合方案的作用、效益、意义确定其价值,用于衡量方案的优劣程度,选择方案的标准当然是越接近原定目标越好,但是实际中往往不易达到。决策目标达到最优化至少满足如下条件:

- 决策目标可以数量化
- 所有可能方案都必须全部找到
- 每个方案的执行结果都能预先知道
- 有绝对的择优标准
- 决策不受时间限制

H·A·Simon 提出以“满意”的原则来代替“最优化”原则,称之为“有限合理性原则”。

(4) 方案的比较和选择

选择方案有经验判断法、数学分析法、试验法三种,这些方法各有优缺点,有赖于决策者根据实际情况灵活地进行综合运用。

2. 几种决策模型

- 理想模型
- 行为模型
- 直觉模型

另外,可以编出程序用计算机解决的问题称为程序化决策,反之称为非程序化决策。事实上,大部分问题介于二者之间。随着计算机技术的发展,人们正在努力将非程序化问题尽量程序化,以便于使用计算机进行处理。

3. 决策支持系统

(1) 决策支持系统的定义和作用

由于考虑问题观点不同,对决策支持系统的定义也不同,比较狭义的定义是:决策支持系统是一种人机交互系统,它能帮助决策者用数据和模型去解决非程序化决策问题。目前更为广义的定义是:任何具有帮助决策者进行决策功能的系统都称为决策支持系统。

(2) 计算机在信息管理中的运用

①电子数据处理系统 (Electronic Data Processing System简称EDPS)

其功能主要是进行数据存储、处理、通信、简单的统计等。

②管理信息系统 (Management Information System简称MIS)

该系统是对管理的信息进行采集、处理、加工、存储的人—机交互系统。

③决策支持系统 (Decision Support System简称DSS)

决策支持系统是辅助管理人员进行决策的人-机交互系统。其主要作用为:

- 预估, 根据当前的数据对未来进行估算和规划
- 预见, 对不确定因素进行估计
- 洞察, 研究影响事物发展的各种因素及其作用。

二、决策支持系统的结构和设计

1. 决策支持系统的结构

DSS是由收集信息、问题识别、模型化以及分析与计算等多种功能模块协调配合构成的支持整个决策过程的集成化人机交互系统。从国内外的的发展来看, 一般分为二库、三库、四库和五库系统:

(1) 二库系统

二库系统包括对话部件、数据库和模型库。其主要功能在信息和模型计算方面实现支持, 包括信息查询、预测、优化等。二库系统逻辑结构如图1所示。

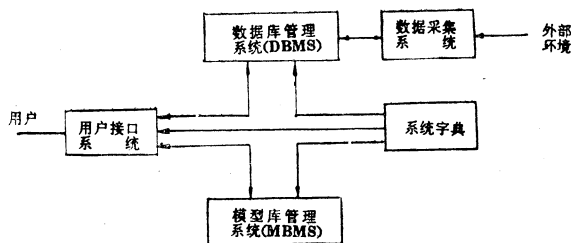


图1 二库系统逻辑结构

(2) 三库系统

三库系统是在二库上加入知识库, 实现对生产规律、决策规律以及数据、模型等方面知识的存贮和管理, 使DSS智能化。

(3) 四库和五库系统

四库是在三库的基础上又加入方法库, 五库则是在三库的基础上加入文字库和方法库。

2. 决策支持系统的设计原则

(1) 总体可行性原则

系统应实用、可靠、可扩充、先进, 并保证DSS的应用能带来社会和经济效益。

(2) 整体最优原则

DSS是一个庞大且复杂的系统, 应将其科

学地划分为若干子系统, 并协调各子系统, 以求全局最优。

(3) 开放性原则

在设计时应兼顾当前实际和今后的发展。

(4) 有限合理性原则

应按行为模型设计, 不能按理想模型设计。

3. 决策支持系统的设计方法

和传统的软件研制一样, DSS的研制也要经历: 系统分析→系统设计→系统实现→系统评价等阶段。由于决策条件千变万化, 很多情况不能事先准确确定。因此设计必须压缩其生命周期, 同时使之具有较强的灵活性。

①用规范化模型的方法分析决策过程, 将“非程序化”化为“程序化”或部分“程序化”。

②将系统可扩充和易修改性作为系统设计的基本要求, 使DSS足够灵活, 以适应决策环境的变化。

③进一步地吸收用户参与分析和设计全过程, 缩短系统一用户的反馈周期。

④利用维型和模块化设计方法缩短DSS研制的周期, 提高效益和适应性。

⑤依据决策问题的发生率(是否重复)、作用(是否关键决策)、时帧(短期或长期)选择制定开发研制DSS的策略。

三、决策支持系统在铁路站场中应用

在铁路运输生产中, 列车编组、行车组织、调度指挥以及沿线众多单位的协调作业中, 都产生着大量的信息, 需要及时、正确地收集和处理。而运输管理则需要根据各种信息及时作出正确的决策。因此, 铁路管理现代化必须依赖于计算机这个强有力的现代化信息处理工具。决策支持系统在铁路站场中的应用可划分为数据采集、传输、存储; 信息处理、加工、通信和辅助决策三个层次。

1. 站场数据采集

数据采集是指从外界获取信息并把它们转换成编码送入计算机的过程。它是决策支持系统的一个十分重要的组成部分, 分终端采集、实时采集和批量采集。

(1) 终端采集

操作员通过显示终端的键盘把数据按一定格式送入计算机,一般采用人机会话方式。主要用于采集目前不易自动采集的信息以及作为自动采集的补充。例如上级下达的任务,尚未建立计算机通信网的中间站的到达、出发列车预确报信息等,这种方式的缺点是输入速度慢,出错率高,但目前仍有60%—70%的数据采用这种方式输入。尤其是在计算机应用的初期阶段更是如此。因此,要在程序设计上多下功夫,使之得到更好的应用。

(2) 实时采集

这种方式可将站场有关信息定时或随机地输入计算机,以利于及时分析处理,作出决策。主要用于采集信息网传输来的列车预确报信息,站场现车状态等。对于现车状态自动采集,国外研制了多种车辆自动识别系统,但是由于铁路运用环境恶劣,可靠性要求高,以及设备投资过高等原因,目前尚未达到实际运用水平。这里我们提出一种投资少、见效快的采集方案。

①尽量利用站场现有电气集中等设备,采集站场现车流动动态,采集站场股道、进路占用状态。这些信息可通过光电耦合方式,从电气集中设备直接采集。这就减少了信息传输、干扰等问题,同时大大降低了设备投资。

②附加部分传感装置,通过在线路两端设置计轴器,采集进出的车辆台数。通过测重设备或其它设备采集车辆重量,以区分机车、重车、空车等情况。

③利用计算机进行逻辑判断和推理,确定站场现车所在线路位置及其排列次序。计算机通过信息传输网或人工输入方式确定到达列车数量及排列次序,由于铁路站场每条线路都类似于队列或堆栈。因此,只要确定了站场内现车数量和次序,并能确定进出每条线的车数,无论车辆如何流动,都可以确定其位置以及排列次序。这样就利用计算机软件弥补了硬设备的不足。

④辅以部分人工输入信息,作为自动采集的校正和补充。

(3) 批量采集

由数据设备或仪器设备把收集到的有关数据

放到磁带、磁盘等介质上,然后再成批地送入计算机。主要用于采集各种原始数据等。

2. 管理信息

(1) 接收输入信息,并进行合法性验证和数据格式转换。主要接收人机对话数据、自动采集数据、信息网输入数据等。

(2) 维护系统数据库。数据库分为动态和静态两部分。动态的数据经常要进行读写操作,主要供决策和随机查询使用。这类数据有:现车状态、线路状态、任务计划等。静态数据是一些不常变更的备查数据,主要有列车运行图、编组计划、站场设备和技术作业过程等。动态数据应在内存开辟公用数据区存储,由数据库管理软件进行管理,并定时存于硬盘。静态数据仅在使用时才调入内存。

(3) 对采入信息进行计算、逻辑判断等预处理工作。主要有:根据采入信息,推定现车状态,线路状态;按确定的车列解体钩计划,推测未来一段时间车流集结情况;列车到发预确报;编制列车编组次序表及各种统计等。

(4) 输出信息。通过信息网输出列车到发信息;通过人机会话方式输出随机查询信息;打印输出作业计划;传送或打印各种统计表;利用绘图机绘制机车运行图。

3. 辅助决策

(1) 将列车运行图、计划任务、专家知识等存储入计算机,作为决策的依据。

(2) 根据给定任务、车流变化情况、站场现车、线路等情况编制出优化的调车作业计划,并用人机会话的方式对调车作业计划进行修改和调整。

(3) 当作业过程中出现意外时,可调出预先存贮的处理方案实时调整或通过人机会话实时调整。

4. 系统结构

由于决策支持系统在站场中的应用尚属初步,且决策支持系统正处于发展阶段,主要功能是采集、预测、优化、查询等问题,因此,拟采用二库或三库系统。系统宜以使用率最高的人机交互系统为主控模块。系统运行之后,一般不停机,由人机交互模块常驻运行,不断地从数据接

口自动采入站场信息,通过人机会话接收操作人员指令和辅助信息;并按操作人员的指令提供查询数据,作出决策方案,提供各种输出方式;对决策方案可由操作人员通过人机交互进行局部调整。

5. 研制中的一些问题

(1) 数据采集的主要问题是:

- 站场气候恶劣,污染严重
- 站场分散,采集信息传输困难
- 站场各种电气信号之间干扰严重
- 采集设备投资、安装、维护费用过高
- 对计算机系统的内存容量和处理速度有较高要求

目前的科学技术水平已完全能够解决这类问题,但要寻求经济、实用、符合我国铁路现状的采集系统还是一项艰巨的任务。

(2) 决策模型和算法问题

铁路站场调度决策是一个十分复杂的过程。目前调度人员使用的多为行为模型,既有程序化问题也有非程序化问题。建立这类决策支持系统模型较为困难,涉及决策理论、数学、计算机技术以及调度专家知识,因此需要多方人员的密切合作。

对决策目标而言,大多数调度问题都是多目标问题,如何评价最优也很难有一个统一的标准。即使有标准,由于大多数调度问题属于 N_P 一完全问题,目前没有,将来也未必会有有效算法。因此寻求实用的近似算法也是理论方面的艰巨任务。

(3) 在建立决策支持系统时,切忌盲目性。要科学地进行系统分析,要有全局和大系统观念,并结合我国铁路实际情况,宜先开发一些子系统,取得经验,逐步完善健全。

四、应用实例

1. 系统来源及目标

炼钢站是钢铁生产过程的一个重要环节,其调度工作的优劣将直接影响到模铸钢的质量和产量。攀钢二期工程之后,要增加模铸钢产量。虽然要对炼钢站进行扩建改造,但由于受客观条件限制,运输设备能力也受到相应的限制,尤其当

作业过程中出现意外情况时更甚。因而,加强炼钢站的运输管理工作,引入先进的科学技术就非常必要。为此,攀钢运输部与我校签定了科技项目《炼钢站计算机调度辅助决策系统》。该系统将成为调度决策的强有力的辅助工具,也即决策支持系统。

炼钢站的任务是按照钢铁生产的要求完成所需的运输任务,调度的任务就是合理地安排调机、进路等运输设备,保证完成生产任务;并尽可能减少咽喉道岔占用时间,以最少的行程保证运输生产的需要。经调研分析,拟定用户要求如下:

- 通过数据采集接口对现场的有关信息进行自动采集与处理
- 通过人一机会话方式采入现场不易采到的信息或现阶段尚不能采到的信息
- 编制调度指挥作业计划并输出
- 当作业中出现意外情况时,实时编制出调整的调度指挥作业计划
- 通过人一机会话方式对站内各线路占用情况,调机动态情况进行随机查询
- 用绘图机绘制调度指挥作业计划图
- 输出方式:屏幕显示,打印机打印,绘图机绘图
- 操作方式:采用简便易行的汉字菜单提示方式进行人一机会话

2. 系统原理及主要模块功能

(1) 炼钢站作业过程

炼钢站是特种作业与普通作业的混合站,其作业过程与钢铁生产工艺流程密切相关。炼钢厂每24分钟出一炉钢,注入特种车辆上的铸模中,然后从“铸锭车场”运到“缓冷车场”进行冷却,冷却后的钢锭运到“脱模车场”进行脱模,脱去模子后的铸锭送入“初轧厂”轧为钢锭,空车和空模分别送回“整模车场”重新修复后再次送入“铸锭车场”待注,如此循环不已。车场、作业关系如图2所示。

(2) 系统各主要模块及功能

① 数据采集子系统

- 实时采集电气集中,计数器等有关现车、线路等有关信息

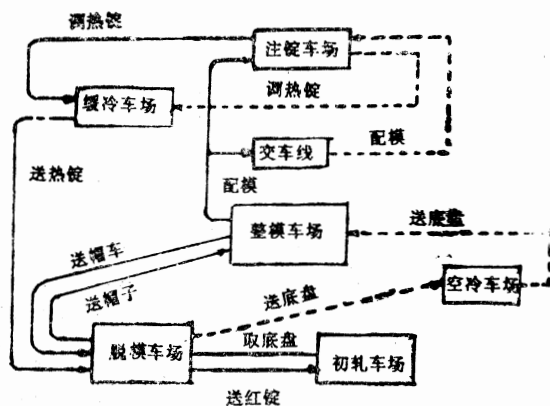


图2 车场、作业关系图

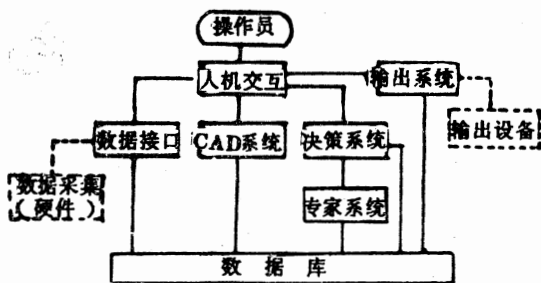


图3 系统联结图

· 将采入的信息进行放大、转换、保持，并输入计算机

②人机会话子系统

- 系统初启，设置时钟计时
- 接收操作员指令并调度相应模块付诸实施
- 调度和管理其它子系统
- 接收人工输入信息
- 对采入信息进行分析、加工处理
- 维护系统正常运行

③决策子系统

- 按要求作出调车作业阶段计划
- 当作业过程中出现意外时，作出实时的调整方案

④其它模块

a. 专家系统：这是一个小型专家知识库，主要用于处理一些异常情况。

b. 数据库系统：存储和维护系统公用数据。

c. 简易CAD：以人机会话方式对决策方案进行人工调整。

d. 输出模块：以屏幕显示、绘图机绘图、打印机打印等方式输出结果。

(3) 调度模型及算法

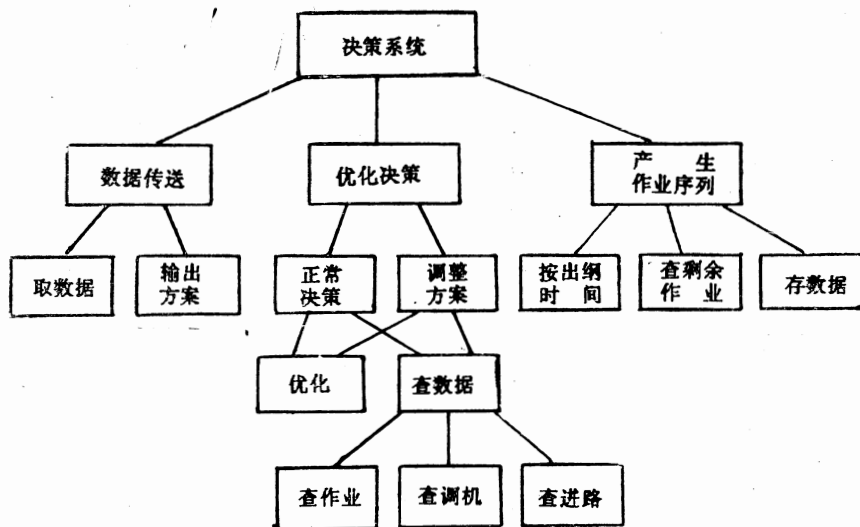


图4 决策功能模块图

①调度模型

炼钢站由5台机车分东2台、西3台进行作业。

目前炼钢站调度采用调机原则上分工，根据作业

忙闲情况灵活调度的方法。人工调度的方法和步骤是：

a. 根据各车场(车间)电话掌握当前站场上

机车和作业的时空分布状态(时间和地点)。

b. 确定当前站场上哪些作业需要完成, 并区别其轻重缓急。

c. 按作业的分工及优先次序安排调机, 并同时考虑到作业省时省约。

d. 安排调机进路。

e. 下达调度指令

f. 收集作业完成情况, 并作出实绩运行图。

根据计算机的特点和人工调度的模型, 用计算机进行调度决策的步骤大致与人工相同。根据人、机调度决策模型的功能和DSS的设计原则, 确定决策模块的子功能模块如前图4。

②产生作业序列子模块

调车作业计划是在未来一段时间内, 对机车进行作业计划安排。由于产生的作业序列必须与站场实际情况大致相同, 因此, 产生作业序列的过程也是一个站场作业过程的模拟, 但此时比实际过程更理想化。其主要算法设计思想是:

按照给定一段时间内的出钢计划, 站场现有的作业情况以及零星作业(与钢铁生产无直接关系)计划, 以时间(分钟)为序, 模拟站场各车间的工作过程, 并按时间次序同时扫描各车场, 若有作业, 则按规定的运行时间使之完成, 并对车场作相应修改。这些按规定时间完成的作业就是一段时间内在站场上产生的作业序列。

③机车调度决策算法

调度决策问题可分为四种情况来讨论。对作业而言, 可分为严格按照作业时间先后次序调度, 和打乱时间次序(局部)进行调度。按时间次序调度的优点是算法简单、可靠, 缺点是考虑问题片面, 有时离优化目标较远。打乱时间次序进行调度, 可进行局部优化, 但算法复杂。对机车而言, 可分为机车作业分工和机车作业不分工两种。调度机车作业的方式可分为: 机车分工按时间次序、机车分工不按时间次序、机车不分工按时间次序、机车不分工不按时间次序四种。限于篇幅, 这里只粗略讨论两种。

a. 机车不分工按作业先后次序调度方式

这种调度模式是: 从当前作业序列中取出优先权最高的一个作业, 然后从机车序列中找出离该作业距离最近、前一个作业结束最早, 最近一

段时间作业时间最少的机车分派之。

b. 作业分片调度方式

这里提出一种分片优化调度决策的算法的基本思想是: 设作业集为TASK, 将TASK按时间划分为k个子集 $T_1, T_2 \dots T_k$, 然后按时间先后次序对每一个子集 T_i 的作业进行调机作业优化调度。在每个子集中可粗略地认为作业没有时间先后次序, 且作业之间几乎不存在“前驱”或“后续”关系。这就使问题得到了简化, 可以在 T_k 中进行局部优化。

现考虑对某一子集 T_k 的优化调度决策问题。设 T_k 有n个作业 $t_1, t_2 \dots t_n$, r_i 是作业 t_i 的工作(运行)时间。

定义有向图 $G = \langle T_k, TD_k \rangle$,

$$td[i, j] \in TD_k$$

$$td[i, j] = \begin{cases} 0 & i=j \\ \begin{matrix} ti \text{ 结束位置到 } tj \text{ 开始} \\ \text{位置之距离} \end{matrix} & i \neq j \end{cases}$$

于是m台机车完成n个作业可看成m台机车周游n个结点的有向图问题。在不考虑每个作业 t_i 的截止时间时, 问题的目标为m台机车的总行程最少。

程序中最为困难和关键的部分是寻找较优Hamilton路径和在Hamilton路径上分派机车。对于寻找较优的Hamilton路径, 这里采用的是基于分支交换法的改进方法, 称为局部交换法。其基本思想是: 在图G中依次选取 $td[i, j]$ 最小, 且满足限制条件者, 并使它们构成H路径。最先入选者, 由于挑选余地大, $td[i, j]$ 较小, 称为优边; 后入选者余地小, 迫不得已, $td[i, j]$ 较大, 称为劣边。为使H路径较优, 而计算复杂性又不至于过高, 最后对 r ($0 \leq r \leq n$) 条后入选的劣边进行局部分支交换调整。

介绍一个围棋软件

笔者有幸获得89年最新版本的人机对话围棋软件, 它是在IBM-PC和其兼容微机上运行的。名为GO89.EXE, 其实力相当于业余初段。它与人既可下分先棋, 又可下让先棋和让子棋, 它能在中途较准确地判断形势和在棋结束后准确点目, 并能自动记谱和打印。感兴趣者, 请与上海科技大学顾建林联系 邮政编码 201300

一种简单易行的多机互连技术

西北工业大学 龙卫红

摘要: 随着微机应用的广泛深入, 常常需要将多个微机连接起来构成一个多机系统。本文选用先进的VLSI芯片Z8038作为微机之间的硬件接口。Z8038最突出的特点它可为两个微处理器之间直接提供双向异步并行传输标准接口, 使得多机系统的互连硬件十分简单。本文介绍的通信软件也非常简单, 容易实现。

我们在设计一个数据采集系统时, 采用了一种简单易行的办法, 把专用单板机和通用微机系统连接起来。在研制采集器阶段, 把微机系统作为开发工具, 调试接口程序; 在实际应用中, 微机系统被用来存贮采集到的大量数据, 并对数据进行事后分析。

一、硬件连接

我们的数据采集系统选用了十六位的Z8000单板机和八位的Z80微处理机系统。从硬件物理层的角度来看, 这二者之间的连接, 实质上就是Z8000CPU同Z80CPU之间的连接, 如图1所示。从Z80微机系统引出数据总线和控制总线, 就象从单板机引出一样容易。它的Z80CPU的所有信号都经过驱动器缓冲, 并引到插件板的端口。从图1可以看出, 整个接口线路就是一个超大规模集成电路芯片——Z8038。

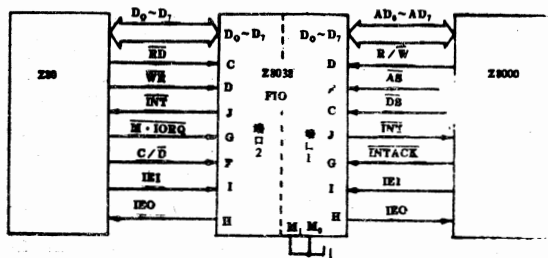


图 1

Z8038—FIO是Zilog公司的新产品, 是Z8000系列外围芯片。FIO最突出的特点是它可为CPU和CPU之间提供双向异步并行传输标准接口。它的两个端口, 均可以连接多种类型的CPU, 如Z8000, Z80或Intel 8086等。接口信号通过引脚M₀和M₁以及程序控制来定义。例如数据总线D₀~D₇在与Z8000 CPU连接时,

可定义为地址/数据总线AD₀~AD₇。又如控制信号C在与Z80连接时, 定义为读信号RD, 而与Z8000连接时, 则定义为数据选通信号DS。

FIO两个端口的工作方式分别由其内部的两套控制寄存器来控制。这些控制寄存器可分别由两边的CPU直接寻址。CPU通过对这些寄存器的编程来指定FIO的工作方式。FIO具有一个128字节的先进先出(FIFO)缓冲区, 使得CPU/CPU之间能够进行异步数据传输和高速DMA传输。图2是FIO的内部结构方框图。

二、接口编程

端口1的引线定义是通过引线M₀和M₁来定义的。当M₀和M₁都接地时, 端口1定义为“Z8000低字节”方式。Z8000 CPU可以直接对其进行读写。

```
LDB RL1, # %01 ; 向控制寄存器0写“1”, 复位FIO
OUTB CR0, RL1
LDB RL1, # 0 ; 复位后必须清除“复位”位
OUTB CR0, RL1
LDB RL1, # %94 ; 置位端口1的“中断允许”位
OUTB CR0, RL1 ; 并定义端口2的引线与Z80兼容
LDB RL1, # %01
OUTB CR2, RL1 ; 使能端口2
LDB RL1, # %40 ; 规定数据传输方向为从端口1
OUTB CR3, RL1 ; 到端口2
LDB RL1, # %C0 ; 当缓冲区满时, 端口1将发出
OUTB ISR3, RL1 ; 中断信号
LDB RL1, # %40 ; 送中断向量
OUTB IVR, RL1
```

往数据缓冲区送数据是通过直接寻址数据缓冲寄存器(设其地址为DBR)来实现的, 例如

```
OUTB DBR, RL2
OUTB DBR, RH2
OUTB DBR, RL3
OUTB DBR, RH3
```

这样就把四个字节送到FIFO缓冲区中。

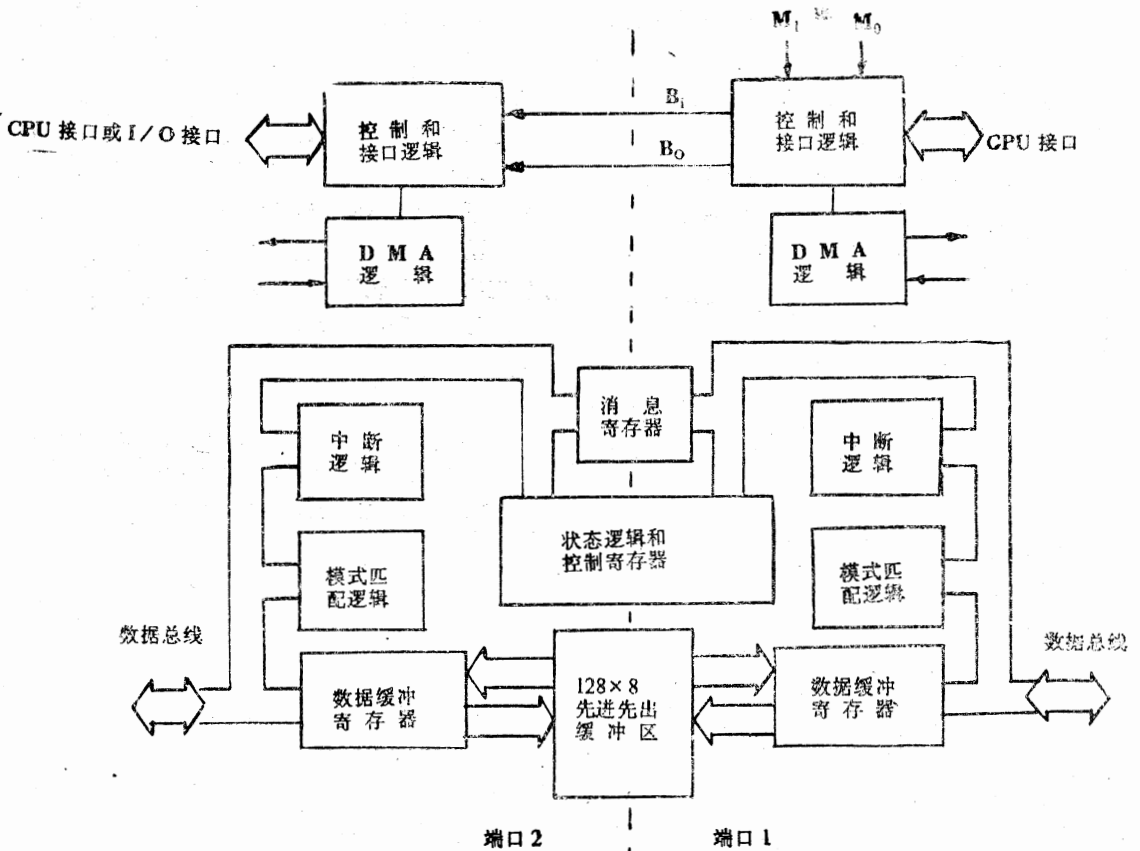


图 2

端口2必须被端口1定义和使能后,才能进行编程。Z80CPU可以通过读端口2控制寄存器0的内容得知端口2是否被使能。FIO只辨识Z80的两个I/O地址,当FIO片选信号 \overline{CE} 为低时,若 C/\overline{D} 输入信号为高,则访问FIO的控制寄存器;若 C/\overline{D} 为低,则访问FIO的数据缓冲寄存器。因此,Z80对FIO所有控制寄存器的访问必须分为两步:第一步先送出要访问的控制寄存器地址;第二步才能对该寄存器进行读或写。下面介绍Z80对FIO编程的一例。

```

WT: IN A, (FIOC) ; 查看端口2 是否被使能
    CP 01H
    JR NZ, WT      ; 若未被使能则等待
    LD A, 0        ; 清除“复位”位
    OUT (FIOC), A
    LD A, 00H      ; 送出控制寄存器0的地址
    OUT (FIOC), A
    LD A, 90H      ; 置位端口2“中断允许”位
  
```

```

    OUT (FIOC), A
    LD A, 08H      ; “字节比较寄存器”地址
    OUT (FIOC), A
    LD A, 30        ; 该寄存器赋值30, 每当缓冲区中的字节数达到30字节时, 端口2产生中断请求
    OUT (FIOC), A
    LD A, 04H      ; “中断状态寄存器2”地址
    OUT (FIOC), A
    LD A, 0CH      ; 置位“字节比较中断允许”位
    OUT (FIOC), A
    LD A, 0CH      ; “中断向量寄存器”地址
    OUT (FIOC), A
    LD A, 20H      ; 中断向量
    OUT (FIOC), A
  
```

Z80 CPU对数据缓冲区FIFO的读写,只需直接访问数据缓冲寄存器,即 $C/\overline{D}=0$ 。例如连续读两个字节:

```

IN A, (FIOD)
LD B, A
IN A, (FIOD)
  
```

FIO还有许多特殊功能,如“邮箱”传送,模式匹配,DMA传送等。这些功能的灵活运用,将大大方便接口软件,并增强系统的性能。

三、通信软件

Z8000单板机和Z80微机系统之间的通信規約采用Tektronix标准。其大致内容如下:

1. 将待传输的八位二进制数据转换成对应的ASCII码。

2. 每三十个字节构成一帧。

3. 每一帧有一个“检查和”。

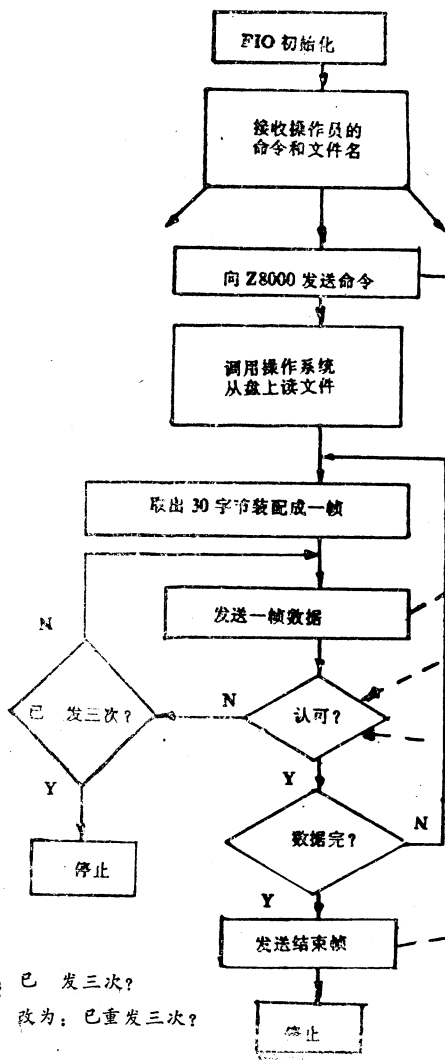
为了保证数据传输的可靠性,采用了“检查和”技术。在传输中,发方将数据分为多个数据

块(称为帧),每个数据块中的数据进行累加,得到一个“检查和”作为校验字。收方接收一个数据块后,也把数据块内的数据累加,然后把累加结果和“检查和”比较。若相同,则传输正确,即向发方回答一个“认可”消息,让其继续发送下一个数据块;若不相同,则回答“不认可”消息。发方将重新发送同一数据块。如果重发三次仍“不认可”,则通知操作员干涉。程序流程如图3所示。

参考文献

- [1] R. Mateosian et al., “采用 Z8000 系列的分布式处理”, 1982年全美计算机会议论文集。
- [2] Zilog, “Z8000系列芯片说明”

Z80 微机系统



注: 已发三次?
改为: 已重发三次?

Z8000 单板机

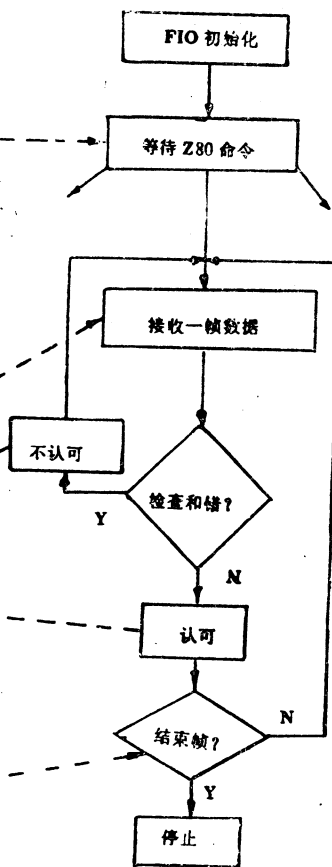


图 3

微机控制X光测厚仪

合肥工业大学微机所

胡士甫

前言

随着国民经济的发展,各行各业对冷轧带钢的质量要求越来越高,为了实现这一目标,在冷轧机上采用了液压压下、液压弯棍、厚度自动控制、板型控制和电子计算机控制等新技术。所有这些新技术都脱离不了厚度的自动检测。目前,世界各先进工业国都在此下了很大功夫。日本东芝511型x射线测厚仪,测量精度误差 $<1\mu\text{m}$ 。西德接触式Volmer测厚仪,测量精度误差 $<1\mu\text{m}$ 。为了达到世界先进水平,现提出一种连续的非接触式x光测厚仪。该仪器采用x射线,对厚度为 $0.2\sim 5\text{mm}$ 的带钢进行双x光束对称式测量。由于采用了计算机技术,能自动补偿误差,能对非线性测量进行自动校正,所以测量精度的相对误差 $<0.1\%$ 。该仪器还有计算机标准RS422接口,应用该接口能与上位机进行双向通讯,将测得的厚度直接传送到轧机的系统控制计算机。

该仪器是实现轧钢自动化必不可少的工具。

一、原理

微机控制x光测厚仪是利用物质对x射线的吸收现象制成的非接触式连续测厚仪。x射线通过不同厚度材料时,被吸收的强度与材料的厚度有关。其吸收特性可用 $I=I_0e^{-\mu x}$ 来表示。其中x为材料厚度,I为透过该材料时剩余射线的强度。 I_0 为当 $x=0$ 时的射线强度, μ 为吸收系数,它与被测材料的原子序数及射线的波长有关。穿透后的x光射线由探测器接收。并将它转换成电信号,微机对此电信号进行快速测量,经数据处理后就可以得到材料的厚度。

二、设计方案及特点

1. 双x光束对称式测量系统

随着时间、温度和电源电压的变化,测量系统会产生漂移。这漂移将严重影响着测量的精度和稳定性。为此,设计了一个双x光束对称式测量系

统。如图1当x光管①发出的x光经过分光板②,分成强度相等的两束光线。其中一束透射过轧制后的带材③由探测器I④接收。另一束透射过基准斜楔⑧由探测器II⑤接收。检测箱⑥的计算机将分别对两路检测到的信号进行放大、模/数转换、数据处理。如果带材的厚度与基准楔板的厚度不相等,计算机发出指令使步进电机⑦转动。步进电机带动斜楔移动,直至计算机所获得的两路检测相等。此时,斜楔与带材厚度相同。计算机从步进电机移动的距离就可以计算出带材的厚度。

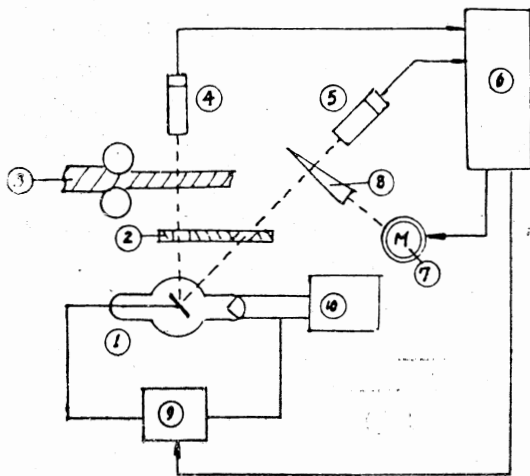


图1. 双x光束测厚仪结构图

①x光管②分光板③被测带材④探测器I⑤探测器II⑥微机检测箱⑦步进电机⑧基准斜楔⑨x光管阳极电压调节单元⑩x光管阳极电流调节单元

当产生下列情况:

(1)x光源强度、波长发生变化时,(2)探测器内光电管、晶体的特性随温度变化时,(3)探测器内光电管电源电压变化时,(4)探测器内电子线路随温度、电压的变化而变化时,两检测器路产生相同的偏差,就可以利用对称的原理,加以抵消,这样有利于克服系统误差。

为了保持测量系统的稳定性,除了采用稳压措施外,还设计了专用的阳极电压调节单元⑨和阳极电流调节单元⑩。

2. 自动校正、补偿系统

虽然我们尽量挑选性能相同的探测器,但仍然以做到完全一致,两测量回路还是存在着差异。如果我们能在测厚仪开始测量之前用标准平板代替被测带材先测量一次,计算机就能记下两回路的初始偏差值,以便在正式检测时进行自动校正及补偿。为此,设计了一种专用探测器,如图2。探测器由晶体①(NaI(Tl)),光导②、光电倍增管③等组成。x光照射到晶体后产生荧光,经光导传输到光电倍增管经光电倍增管的多级放大,产生负脉冲。该脉冲经放大、滤波、跟随器④转换成低阻抗,通过电缆输入到检测箱。此电平的幅度正比于输入x光的强度。在探测器的前部装有一块标准平板⑥,它受电磁铁⑦及弹簧⑤的作用,可以在导槽⑨中移动,阻挡或打开x光接收孔。在带材被轧制前,或者一卷带材轧制完毕后的空隙,可以使电磁铁失电,标准平板挡住x光接收孔。x光就射透标准平板到达晶体。计算机此时对两回路进行检测,并通过自动调节使两回路的强度相等。记下此时的数据,用它作为正式检测时自动校正及补偿的偏差。经过自动校正、补偿后检测系统的精度可以提高一个数量级。

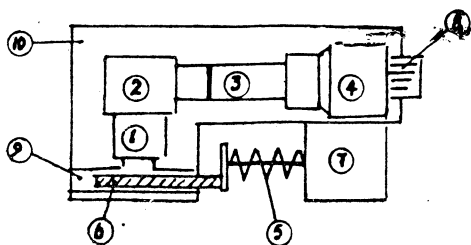


图2. 探测器结构

- ①晶体 ②光导 ③光电倍增管 ④放大、滤波、跟随器
⑤弹簧 ⑥标准平板 ⑦电磁铁 ⑧接插座 ⑨平板导槽
⑩外壳

3. 最佳测量范围的优选机构

上面提到物质对x射线的吸收特性用: $I = I_0 e^{-\mu x}$ 表示。从上式可知,在 $\mu x = 1$ 处,仪器对

被测物厚度的变化具有最大的相对灵敏度。而在 $Mx = 2$ 处,有最小的统计误差。也就是说 Mx 值有一个最优范围。在这个范围内测器具有最高的精度。式中的吸收系数 μ 与被测物质原子序数以及射线波长有关。本仪器用一个x光管自动阳极电压调节单元来实现对x射前波长的调节。x光调阳极电压 U_a 与x射线波长 λ 存在下列关系:

$$\lambda = K \frac{12.2}{\sqrt{U_a}} \text{ \AA}, \text{ 其中 } K \text{ 是系数。当管压 } U_a$$

增加时,波长减小,x光的穿透能力增强。但由于x光管本身特性的限制,在同一电压下,不同波长的射线,它的辐射强度不同,如图3所示,所以我们既要考虑增加管压,减短波长,又要考虑x光管本身能承受的电压范围。计算机可以根据上述原则计算出管电压调节量,使探测器接收的强度为最佳状态。这样,x光管产生射线的波长随被测物厚度x而变化,使与x射线波长相关的 μ 与被测物厚度x的乘积 μx 始终选在最高灵敏度区,达到了最佳测量范围。

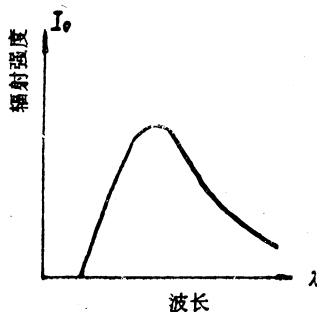


图3. x光管的波长与辐射强度

管电压自动调节单元组成如图4。计算机算出x光管的电压(数字)量,由数据总线送8D触发器74LS273锁存,经光电隔离管PC817送到模/数转换器DAC0808。DAC0808输出的电流 I_0 经运算放大器 A_2 , 转变成电压量。该电压量经运放 A_1 的放大直接驱动达林顿管 BG_1 。二极管 $D_1 \sim D_4$ 组成桥路串接在管压变压器的初级回路上,而 BG_1 作桥路的直流负载。调节 BG_1 的导通量就可以达到调节 T_3 、 T_4 变压器的初级电压。 T_3 、 T_4 是x光管的升压变压器。

x光管管的提高,不仅使x射线波长发生

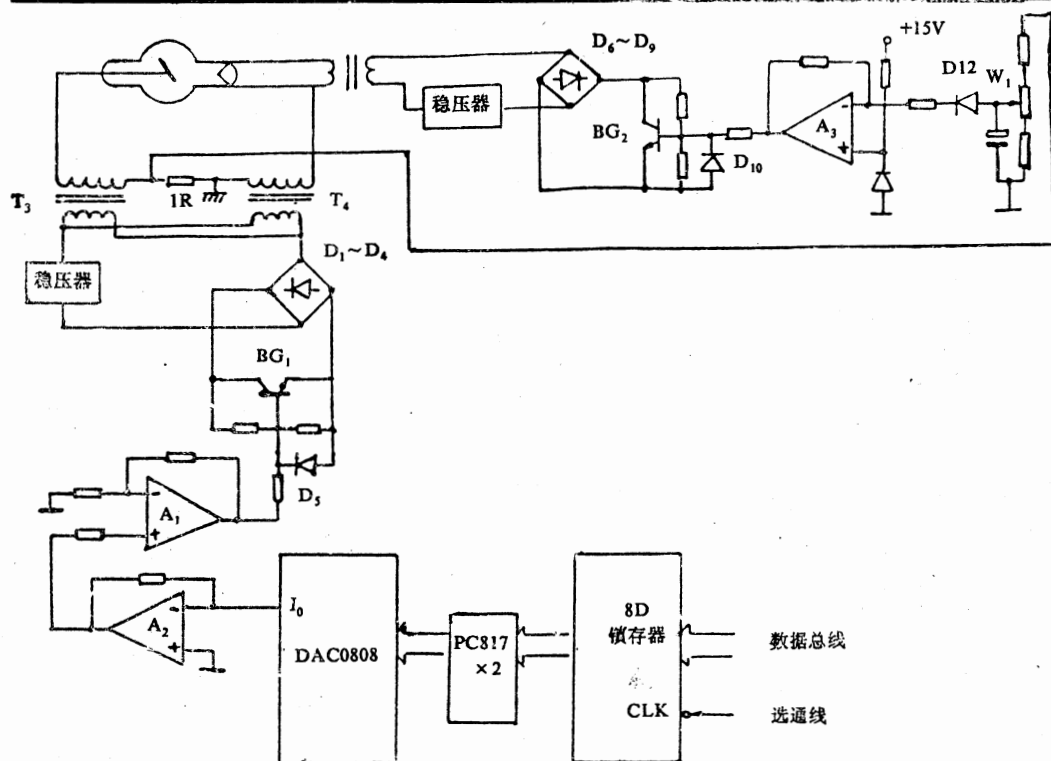


图4. x光管电压、电流调节单元原理图

变化，而且使x光管的阴极电流也随之提高。为此，设计了一个x光管管流稳定单元，见图4。管流的反馈信号取自管压变压器 T_2 、 T_3 中点接地电阻 $1R$ 上。经运放 A_1 比较、放大后驱动达林顿管 BG_2 。其原理同管压调节单元。从而实现x光管的恒流。

4. 高分辨率、高精度的可编程模入通道

高分辨率、高精度的模入通道是本仪器设计的一个关键。它的原理如图5所示。从探测器I、II来的信号，由多路转换开关MUX进行分时切换。选其中的一路送到可编程数据放大器3606。放大后的模拟量经12位的AD574转换成

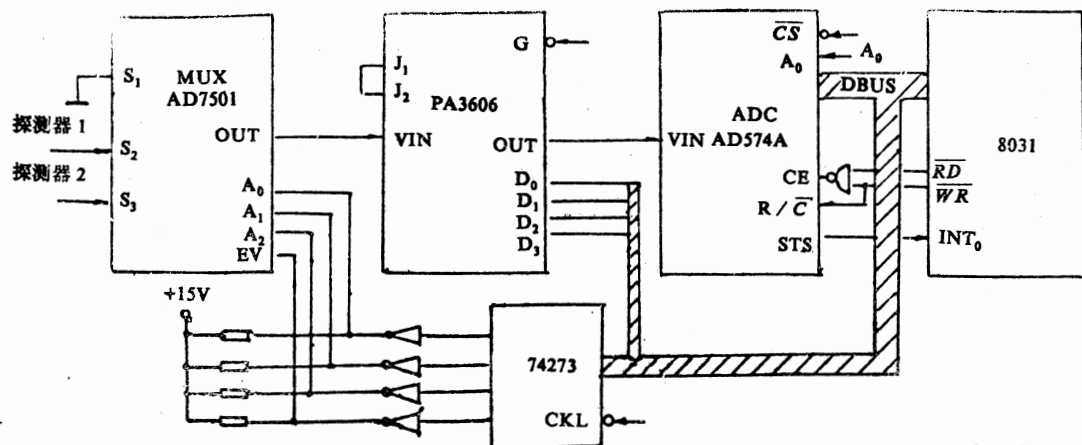


图5. 可编程模入通道原理图

数字量。A/D转换结束, AD574向8031发中断信号。8031响应中断, 分两次接收A/D转换数据。一次是低四位 $D_3 \approx D_0$, 另一次是高八位 $D_{11} \approx D_4$ 。

多路转换器开关的 S_1 输入端接模拟地。这是由于多路转换开关、放大器的校零

3606可编程数据放大器具有输入阻抗高、共模抑制比高、增益线性好($<0.005\%$)和温度、时间漂移低等优点。其增益范围 $1 \approx 1024$ 倍, 通过软件用 $D_3 \approx D_0$ 四根线来设置增益。使用它就能将模拟电压始终放大到 $+5+10V$ 范围内, 这样进入A/D转换器的电压值最大, 从而保证了A/D转换器的分辨率和精度, 也保证了整个模入通道的分辨率和精度。

5. 标准的RSS-422串芯口

仪器还设置了计算机标准的RS-422串芯口, 利用8031的RXD、TXD两根线, 经光电隔离后送给差分驱动器75175、75174。此RS-422串芯口能与上位机进行双向通讯, 既能接收上位机的命令, 又能将测厚仪测得的数据送到上位机。它传输的距离为1500m。它是一种以差动方式输出, 抗干扰能力强的通讯口。

6. 软件功能及流程图

本仪器的软件均采用模块结构, 具有较强的通用性和扩充性。共有以下七部分:

- (1) 监控模块
- (2) 系统自动校正、补偿偏差模块
- (3) 模拟信号采集模块

(4) 数据处理模块

(5) 双x光束平衡及基准斜楔调节模块

(6) 最佳测试条件优化模块

(7) 通讯模块

这些软件用51系列单片机的汇编语言编制而成。

监控模块包括键盘、显示器的管理以及人机对话软件。

数据处理模块包括对采集的数据进行数字滤波、线性校正等功能。经处理后的数据能达到较高的精确性、可靠性和稳定性。

最佳测试条件优化模块包括x光管管压自动调节、模入通道增益自动调节。

结束语

本仪器采用双x光束对称测量, 有利于克服系统误差。采用多种自动校正、补偿方法、数据处理以及最佳测试点优选等措施, 故仪器的精度较高。测试1mm带钢时, 仪器的分辨率达到 $\pm 0.122 \mu m$, 已满足国内外新型轧机对测量仪器的要求。这种仪器改换相应的基准斜楔后, 能应用于铜、铝等带材的测厚, 具有推广应用价值。

参考文献

1. 《无损检测的原理和方法》 李新作主编
2. 《冷带钢轧机厚控(AGC)技术简介》 轧钢88-4
3. 《轧钢工业中非接触式检测》 轧钢87-3
4. 《塑性加工〈二〉》 计为为非接触计测《塑性加工》 Vol. 26, No. 296, (1985, 8)

微机计算计数器大批投产

南京电讯仪器厂研制成功的微机计算计数器大批投产, 该仪器分别获机电部科技一等奖和部、省优质产品称号。它是微计算机技术和测量技术相结合的智能测试仪器。该机计算计数器是微计算机和多周期同步, 模拟内插扩展技术相结合。机内采用MC6800微处理器组件, 构成微计算机。仪器的所有功能在微机的控制下产生, 其程序可使仪器自动实现时间间隔、周期、频率和阿伦方差等功能的测量。可广泛应用于航空、航天、导弹和核工程等多种领域, 精密测时, 测距和进行数字通讯, 频率稳定度的测试、分析等。测试是否准确, 可自动检验, 测量精度比普通仪器高一千倍, 普通仪器测试数据, 人工计算要三天, 而该仪器只需三秒。用户编程容量为 $4K \times 8Bit$, 并配有一合盒式录音机, 每合磁带可记录20K左右的应用程序。还配有一台22倍点阵式打印机, 自动记录测量结果, 其打印速度为3行/秒。

(李相彬)

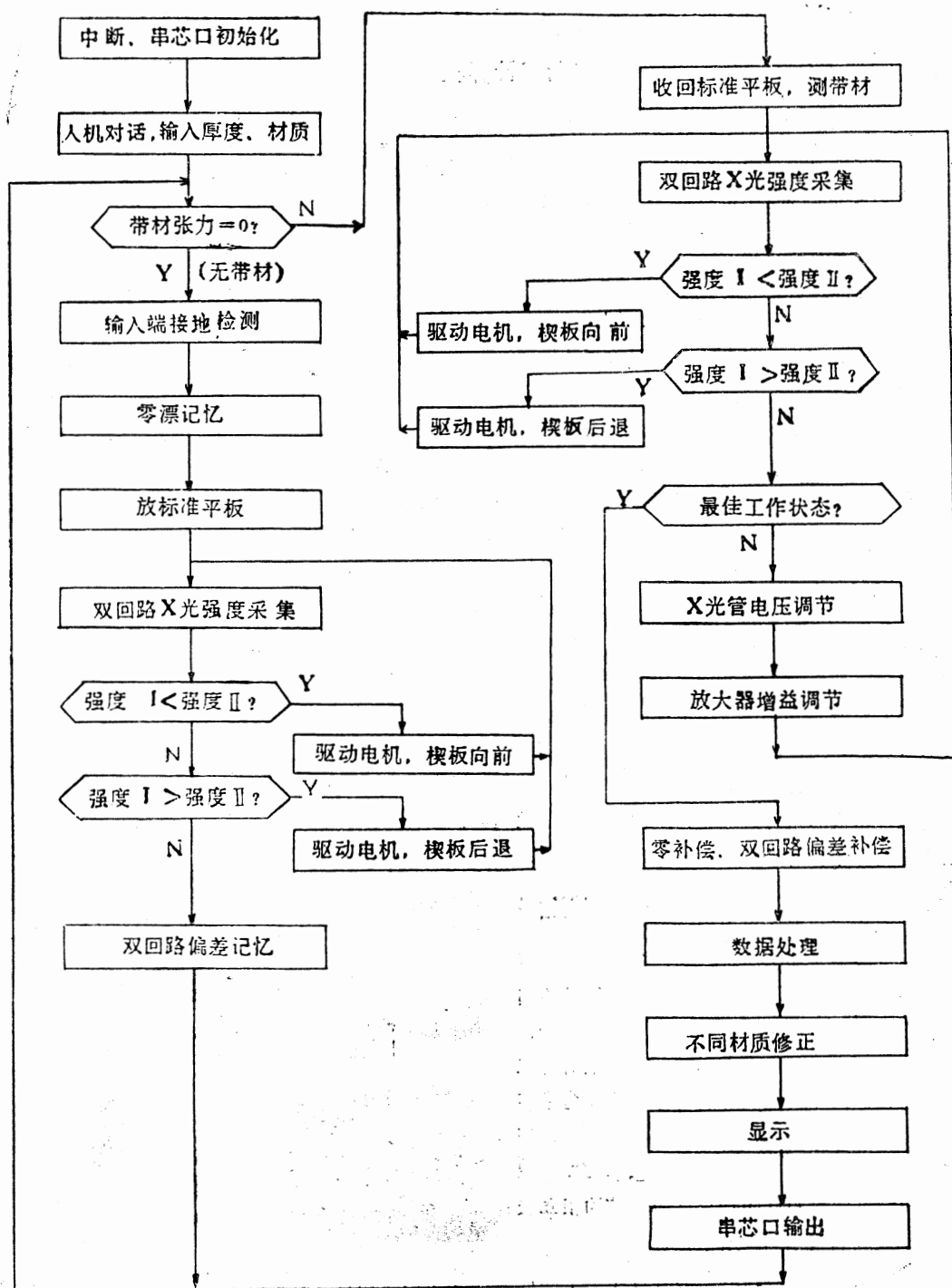


图4. 程序框图

3070打印机断针定位方法及8088汇编程序

重庆制药七厂计算机室 龙兵生

摘要 本文针对紫金 3070 24 针汉字打印机断针问题, 介绍一种用来测试断针位置的简易方法, 并给出一个使用 8088 汇编语言编制的测试软件。该软件只要修改一个字节, 即可用来测试其它型号的打印机断针定位问题, 故具有较强的通用性。

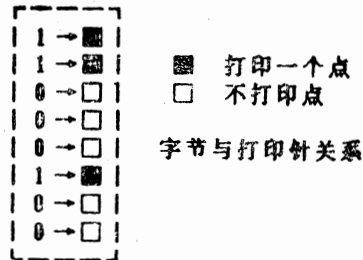
一、引言

计算机外部设备之一的打印机, 由于打印针具有一定的使用寿命, 用户使用一段时间以后, 或多或少都会遇到打印机发生断针现象(从打印输出的图文信息即可确认), 影响打印输出字符的美观。为迅速进行断针的更换, 需确定断针的准确位置之所在, 这虽然可通过人的肉眼对打印信息和打印头部位进行观察得出答案, 但准确性较差且需反复校验, 费时费力。为此, 笔者根据自己在工作中碰到的这类问题及解决方法, 以紫金 3070 24 针汉字打印机断针为例, 介绍一个简易的测试方法和程序, 断针测试的准确率达 100% 且一次完成。其它类型的打印机断针定位问题, 只需将本文提供的程序作一字节改动, 即可用来进行断针测试。

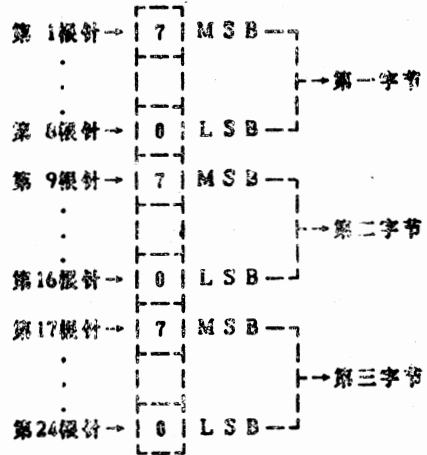
二、检测原理及方法

测试打印机的断针之位置, 基本原理就是设法直接控制打印机的某一根针来打印一条直线, 据此观察所测试的针是否可以打印出直线来。若能够打印出直线, 那么可以确认该针完好无损, 打印不出直线来, 那么该针已经打断。记下该断针的编号, 然后用一根好针去替换(打印针的编号顺序我们规定为: 最上面的一根针为一号针, 其次为二号针、三号针... 最下面的一根针为二十四号针), 恢复该位置针的打印功能。为了实现直接控制打印机一次只出一根针打印信息, 需要了解打印机 24 根打印针与打印信息字节的对应关系。3070 打印机的打印针与打印字节的相互关系是这样对应的(如图一所示):

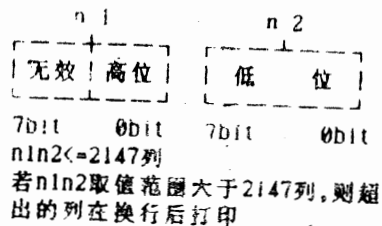
图象数据为“1”时, 对应的针打印一个点, 为“0”时对应的针不打印。与打印头对应的数据分三字节传送(如图二所示), 第一字节的最高位对应第一根针, 最低位对应第八根针; 第二



图一



图二



图三

字节的最高位对应第九根针, 最低位对应第十六根针; 第三字节的最高位对应第十七根针, 最低位对应第二十四根针。图象数据的传送列数用 $n1$, $n2$ 表示, 如图三所示, $n1$ 和 $n2$ 不能同时为

零。打印点列数 N 与 n_1 和 n_2 的关系如下： $N = 256Xn_1 + n_2$ 。

了解了打印信息与打印机24根针的对应关系以后，我们可以向打印机发送控制码序列 $ESC + 49H + n_1 + n_2$ ，将打印机设置成图象打印方式，从程序中对打印信息字节的内容，根据所选择的针号进行有目的的控制，一次只出一根针来打印768个点组成的直线（在程序中我们取 $n_1 = 3$ ， $n_2 = 0$ ，得到打印点列为 $N = 256Xn_1 + n_2 = 3 \times 256 + 0 = 768$ ）。例如：将3字节数128, 0, 0传送给打印机后，第一根针出针打印一个点，其余23根针不动作，连续打印768次，即可打印出由第一根针打印出来的一条768点组成的直线。直线的点阵长度用户可以取其它的值，以打印出来的直线便于观察为准。附录一是我们使用DAJC.EXE检测3070打印机24根针的打印情况。

三、程序使用说明

用户将本文提供的8088汇编语言源程序DAJC.ASM用字处理软件键入计算机，经编译(MASM DAJC, [回车])，连接(LINK DAJC, [回车])以后，生成可执行文件DAJC.EXE。此后，用户需要检测3070打印机断针位置时，只要在操作系统提示符下键入：

A>DAJC [回车]

```
type dajc.asm
; DAJC.ASM ---- 3070 打印机断针定位程序 (1989.11)
STACKSG SEGMENT PARA STACK 'STACK'
    DW 50 DUP(?)
STACKSG ENDS
DATASG SEGMENT PARA 'DATA'
LBU1 DB 0DH,0AH
    DB '检测第几号针 (1-24, 按其余键程序结束)? '
LBU2 DB '第 根针打印'
LBU3 DB 3 DUP(?)
LBU4 DB 1BH,49H,3H,0H
LBU5 DB 0DH,0AH
LBU6 DB 3
LBUFF1 DB 4 DUP(?)
DATASG ENDS
CSEG SEGMENT PARA PUBLIC 'CODE'
LBPRG PROC FAR
    ASSUME CS:CSEG,SS:STACKSG,DS:DATASG
    PUSH DS
    XOR AX,AX
    PUSH AX
    MOV AX,DATASG
    MOV DS,AX
    MOV AX,6
```

程序运行后，屏幕提示“检测第几号针(1—24)，按其余键程序结束)?”用户可输入1—24中的任一数字来检测相应的打印针打印一条由768个点组成的实线条，并在直线打印完后自动打印出第XX根针打印字样，以便于用户记下断针位置。若用户键入1—24以外的其它键，则程序自动识别用户意图而结束测试，返回DOS。

四、结束语

使用本文提出的方法及程序测试打印机断针问题有一定的局限性，即若某一根针不打印是由其它故障而非打印机断针引起，则测试无效。由断针引起的则测试是非常直观和方便的。本文提供的程序只需作一个字节的替换（即点图方式功能码序列 $ESC + 49H + n_1 + n_2$ 中的特征码49H换成相应的打印机的特征码即可。例如M2024打印机可将49H换成34H），就可用来测试其它型号的打印机（如M1724, 2024, 9400等）的断针定位问题。

参考文献：

- 1 《高级程序员编程指南》Robert Jourdain著
中国科学院希望电脑公司译
- 2 《微型计算机原理及应用》周明德著
清华大学出版社出版
- 3 《3070-24针汉字打印机使用手册》
南京紫金信息工业公司编著

```
INT 10H
MOV DX,3D9H
MOV AL,6
OUT DX,AL
IXKU: CALL QLIX
    CALL DIS
    CALL JPUR
    CMP AL,0
    JE IXEND
    CMP AL,1
    JE P10
    CALL PPL
    CALL JSW2
    CMP AL,10D
    JL IXEND
    CMP AL,24D
    JG IXEND
    CMP AL,17D
    JL P9
    CALL JSW4
    JMP KUDY
P9: CALL JSW3
    JMP KUDY
P10: CALL PPL
```

	INC BX				
	MOV AL, (BX)	UJDYMK	ENDP		MOV AL, (BX)
	CMP AL, 31H	DIS	PROC NEAR		SUB AL, 30H
	JL IXEND		MOV DX, OFFSET LBU1		MOV DL, 10D
	CMP AL, 39H		MOV AH, 9		MUL DL
	JG IXEND		INT 21H		INC BX
	CALL JSW1		RET		MOV AH, (BX)
KUDY:	CALL UJDYMK	DIS	ENDP		SUB AH, 30H
	JMP IXKU	DYMK	PROC NEAR		ADD AL, AH
IXEND:	RET		XOR AH, AH		RET
LBPRG	ENDP		PUSH DX	JSW2	ENDP
JPUR	PROC NEAR		XOR DX, DX	JSW3	PROC NEAR
	MOV DX, OFFSET LBU6		INT 17H		MOV CL, 16D
	MOV AH, 0AH		POP DX		SUB CL, AL
	INT 21H		RET		MOV CH, 0
	MOV BX, OFFSET 1.BUFF1	DYMK	ENDP		CALL JSIX
	MOV AL, (BX)	JSIX	PROC NEAR		MOV LBU3+1, AL
	RET		MOV AL, 1		RET
JPUR	ENDP		CMP CX, 1	JSW3	ENDP
UJDYMK	PROC NEAR		JL JS10	JSW4	PROC NEAR
	MOV CX, 0EH		MOV DL, 2		MOV CL, 24D
	MOV BX, OFFSET LBU2	JSP:	MUL DL		SUB CL, AL
P000:	MOV AL, (BX)		LOOP JSP		MOV CH, 0
	CALL DYMK	JS10:	RET		CALL JSIX
	INC BX	JSIX	ENDP		MOV LBU3+2, AL
	LOOP P000	QLIX	PROC NEAR		RET
	MOV BX, OFFSET LBU4		MOV AL, 0	JSW4	ENDP
	MOV CX, 4		MOV LBU3, AL	PPL	PROC NEAR
PP1:	MOV AL, (BX)		MOV LBU3+1, AL		PUSH BX
	CALL DYMK		MOV LBU3+2, AL		INC BX
	INC BX		RET		MOV CH, (BX)
	LOOP PP1	QLIX	ENDP		INC BX
	MOV CX, 300H	JSW1	PROC NEAR		MOV CL, (BX)
P00:	MOV BX, OFFSET LBU3		CMP AL, 38H		CHP AL, 1
	MOV DX, 3		JG P11		JE PPL0
PP2:	MOV AL, (BX)		SUB AL, 30H		MOV LBU2+2, CH
	CALL DYMK		MOV CL, 8		MOV LBU2+3, CL
	INC BX		SUB CL, AL		JMP PPL1
	DEC DX		MOV CH, 0		MOV AL, 30H
	JNZ PP2		CALL JSIX	PPL0:	MOV LBU2+2, AL
	LOOP P00		MOV LBU3, AL		MOV LBU2+3, CH
	MOV BX, OFFSET LBU5		JMP P13	PPL1:	POP BX
	MOV AL, (BX)	P11:	MOV AL, 00H		RET
	CALL DYMK		MOV LBU3+1, AL	PPL	ENDP
	INC BX	P13:	RET	CSBG	ENDS
	MOV AL, (BX)	JSW1	ENDP		END LBP00
	CALL DYMK	JSW2	PROC NEAR		
	RET		INC BX		

IBM PC/XT 硬盘维修及00磁道的调整

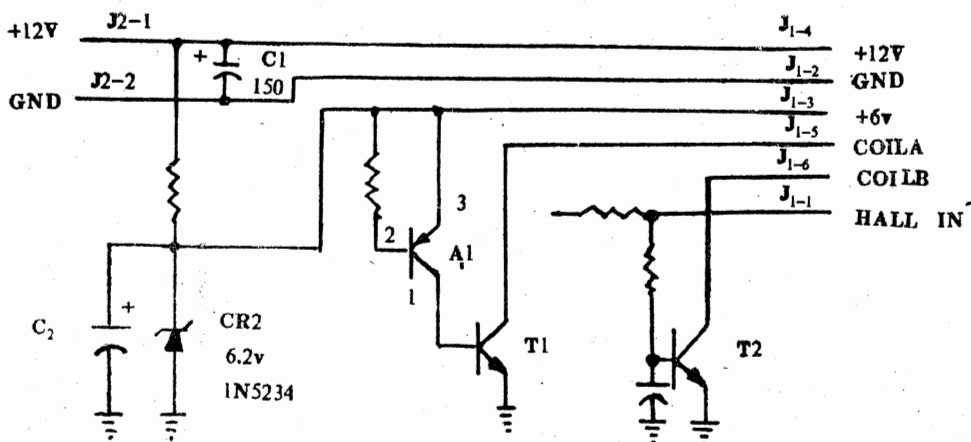
河南洛阳矿山机器厂信息中心 陈永青

1. 故障现象: IBM PC/XT 读写硬盘时, 显示非法的驱动器说明 (Invalid Drive Specification).

分析与排除: 首先用低级格式化程序对硬盘进行格式化无效, 后用交换对比法确认硬盘故障。经检查硬盘加电后马达就未旋转, 硬盘型号

为ST412型, 其马达转速控制部份, 采用一独立的电路, 由+12V供电, 硬盘加电后马达就立即加速, 待其加速到3600转/分, 磁头回归零道并准备就绪。其电路与I型硬盘电路类似。

检查电源插头J2, +12V电压正常, 检查去主轴马达插头J1, 发现+6V接近0V, 断电后测



+6V与地短路,进一步检查发现稳压二极管CR2(6.2V)击穿短路,用稳压为7V的玻封二极管代替,实测电压为6.3V,但马达仍未旋转。马达线圈A、B一边接+12V,一边接T1或T2,当T1、T2导通时,马达旋转加速,HALL IN (J1-1)为霍尔传感信号,当马达旋转加速后为高电平。要使马达旋转,首先T1须导通,即T1基极应有0.7V左右的电压,测T1基极电压为0V,而T1基极又由A1(TRQ2907)的1脚控制。A1为四个PNP型三极管的集成电路,1、2、3脚一组,5、6、7脚一组,8、9、10脚一组,12、13、14脚一组,管脚分别对应c、b、e极,4、11脚未用。检查A1的1、2两脚断路,2、3两脚短路,造成3脚c极无输出,即T1基极无电压,T1不导通,马达不旋转。由于TPQ2907较难购到,而其只有一组三极管损坏,故把TPQ2907的1、2、3脚剪断,用一般的PNP型三极管3CG5的c、b、e极,分别对应接至原电路,开机后恢复正常。

2. 故障现象:IBM PC/XT机用硬盘启动时,显示磁盘根故障(DISK BOOT FAIL),系统无法启动。

分析与排除:对硬盘反复进行格式化,硬盘可以启动,但经过几次开关机后,又显示根故障,用软件方法读0道0号磁头下的数据时,提示读错误,初步判断为00磁道划伤。此台硬盘的0道传感器在步进电机的外侧,为一光耦合电路传感器,当步进电机回归零道时,与步进电机主轴相连的档板挡住传感器的光源,从而判断出磁头已回归0道,我们将档板加长,就可改变0道位置。实际做法,采用软盘写保护标签,粘住档板,粘牢后,让标签长于档板2毫米左右,多余部分剪去,因标签提前挡住光源,就改变了愿2道位置,避开了被划伤的磁道。磁盘在出厂时都留有10个左右的备用磁道,因而格式化硬盘时,仍可按最大柱面数305来格式化硬盘,经这样处理后就排除了00磁道划伤故障。

长城0520—CH硬盘故障两例

湖南怀化铁路分局微机室 梁 军

故障例一:

开机自检到硬盘时,屏幕上出现“1701”,然后进入Basic状态。

解决方法:

用软盘启动后,打入C后可以进入硬盘并可

以对硬盘中的程序和数据进行正常访问,说明硬盘控制器部分是完好的,故障只有硬盘本身。仔细分析自检程序,可以断定问题出在硬盘的00道上。因此笔者采用FORMAT程序对硬盘进行格式化,并把操作系统拷入,然后热启动,故障消

RAM模块地址线和数据线固定故障的诊断

西南交通大学 马永强

摘要: 本文对计算机系统中RAM模块的等价于地址引线和数据引线的固定故障作了详细的讨论和理论分析之后, 提出了适用于同步和异步两种总线通讯方式的故障诊断算法。算法复杂性为 $O(\ln n)$ 。

一、引言

计算机系统中RAM部件的故障诊断, 是整个系统故障诊断的一个重要组成部分。关于RAM芯片本身的测试, 有关文献[1-4]已提出了多种不同的测试方法。本文将在计算机系统级把存储器RAM模块作为整个系统的一个部件, 针对RAM模块中等价于地址引线和数据引线的固定型(s-a-x ($x=0, 1$))故障部分, 提出一个对同步和异步总线通讯方式均适用的软件诊断算法。它可用于诊断RAM模块与主机之间地址总线 and 数据总线的固定型故障(如开路或短路), RAM扩展板与扩展插座之间的接触不良等故障。

二、诊断原理

不失一般性, 我们讨论图2.1所示的字长为8位、容量为 $2^{16}=64K$ 字节的RAM模块的数据引线和地址引线的s-a-x ($x=0, 1$)故障。

诊断图1所示的RAM模块的数据引线和地址引线的s-a-x故障, 可分为三个部分进行: 数据引线 D_7-D_0 ; (2)RAM模块内寻址的地址引线 $A_{15}-A_0$; (3)RAM的模块寻址的地址引线 $A_{N-1}-A_{16}$ 或模块选择信号CS。下面分述之。

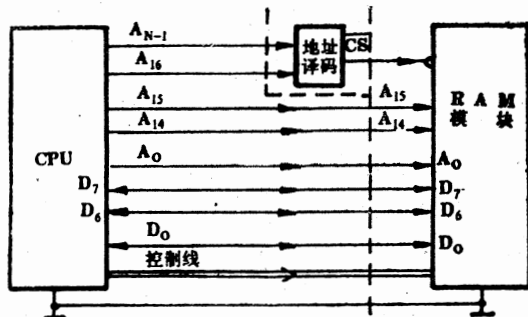


图1

1. 数据引线 D_7-D_0 的诊断

当RAM模块无故障时, 其CPU从存储器任一单元的读出内容总是与最后一次写入该单元的内容相同; 若RAM模块的某数据引线 D_i 发生s-a-x ($x=0, 1$)故障, 则不管CPU向RAM某单元发送写入的 D_i 位数据是0还是1, 其读出的数据总是 x 。因此, 可用以下方法诊断 D_7-D_0 。

除。

故障例二:

自检时正常, 但在调入操作系统时出现提示“Bad or missing command interpreter”, 并死机。

解决方法:

由提示可知是硬盘中的command.com有问题, 用dir命令和pctool进行检查, 都可以看到该程序的存在, 但就是不能进行拷贝, 这说明

文件目录区中的某一个字节损坏, 根据其原因笔者采用软件方法而没有用FORMAT命令。具体方法是先把原先的command.com进行更名, 这样这个程序就占用了坏簇区, 但是这个程序不能使用, 然后拷贝一个新的command.com到硬盘, 经过这一处理, 热启动后硬盘恢复正常, 故障解决。

以上两例的产生均由电源掉电而冲坏某一个存储区, 使得无法启动硬盘, 通过分析故障源而采用两种不同的方法而得到理想的结果。

的s-a-x故障。

(1) CPU任选一个模块内地址(例如00H)向该RAM模块写入全0数据,然后读出。若读出内容为全0,则数据引线 D_7-D_0 无s-a-1故障;

(2) 同样地, CPU任选一个模块内地址(例如00H)向该RAM模块写入全1数据,然后读出,便可诊断出数据引线 D_7-D_0 上s-a-0故障。

2. 地址引线 $A_{15}-A_0$ 的诊断

为叙述方便,我们先作如下的定义。

定义:用 $M_c(A_{15}, A_{14}, \dots, A_0)$ 表示CPU面向地址总线的16位地址线($A_{15}, A_{14}, \dots, A_0$)的寻址空间,简称为 M_c ;用 $M(A_{15}, A_{14}, \dots, A_0)$ 表示RAM模块的16位地址引线($A_{15}, A_{14}, \dots, A_0$)对RAM存储单元的寻址空间,简称为 M 。

按照图1的连接方式,若RAM模块地址引线 $A_{15}, A_{14}, \dots, A_0$ 中只存在固定型故障,则有如下引理:

引理1:

(1) 与RAM模块中无故障地址引线连接的地址总线,其对应位取值相同的 M_c 地址空间中的所有地址,必映射到 M 地址空间的同一个地址中;

(2) 映射到 M 地址空间中同一地址的 M_c 地址空间中的所有地址,其中与RAM模块中无故障地址引线连接的地址总线,其对应位取值必相同。

证明:假定RAM模块中有地址引线 A_i s-a- x_i 和 A_j s-a- x_j ($x_i, x_j=0$ 或 $1, i>j$)故障。

(1) 由于 A_i s-a- x_i, A_j s-a- x_j , RAM模块总是认为系统的地址总线 A_i 和 A_j 上的值分别为 x_i 和 x_j ,与 M_c 空间中 A_i 和 A_j 的地址位无关。因此,在 M_c 地址空间中,对任何一组形如下列的 M_c 地址空间中的地址:

$$\begin{aligned} & a_{15} \dots a_{i+1} 0 \ a_{i-1} \dots a_{j+1} 0 \ a_{j-1} \dots a_0 \\ & a_{15} \dots a_{i+1} 0 \ a_{i-1} \dots a_{j+1} 1 \ a_{j-1} \dots a_0 \\ & a_{15} \dots a_{i+1} 1 \ a_{i-1} \dots a_{j+1} 0 \ a_{j-1} \dots a_0 \\ & a_{15} \dots a_{i+1} 1 \ a_{i-1} \dots a_{j+1} 1 \ a_{j-1} \dots a_0 \end{aligned}$$

它们都映射到 M 地址空间中的同一个地址

$$a_{15} \dots a_{i+1} x_i \ a_{i-1} \dots a_{j+1} x_j \ a_{j-1} \dots a_0$$

上。

对整个 M_c 地址空间,当RAM模块地址引线 A_i s-a- x_i 和 A_j s-a- x_j ($x_i, x_j=0$ 或 $1, i>j$)故障时,它在RAM地址空间中的映象为 M 的子空间

$$M(A_{15}, \dots, A_{i+1}, x_i, A_{i-1}, \dots, A_{j+1}, x_j, A_{j-1}, \dots, A_0)$$

M 的其它子空间

$$M(A_{15}, \dots, A_{i+1}, x_i, A_{i-1}, \dots, A_{j+1}, \bar{x}_j, A_{j-1}, \dots, A_0)$$

$$M(A_{15}, \dots, A_{i+1}, \bar{x}_i, A_{i-1}, \dots, A_{j+1}, x_j, A_{j-1}, \dots, A_0)$$

$$M(A_{15}, \dots, A_{i+1}, \bar{x}_i, A_{i-1}, \dots, A_{j+1}, \bar{x}_j, A_{j-1}, \dots, A_0)$$

(x_i, x_j 为固定值)

均不能被 M_c 地址空间映射到。

(2) 设 $a_{15}a_{14} \dots a_0$ 和 $b_{15}b_{14} \dots b_0$ 为 M_c 地址空间中的两个地址,若存在 $a_k \neq b_k$ ($k=0, 1, \dots, 15$ 但 $k \neq i, j$),即 $b_k = \bar{a}_k$,则它们映射到 M 地址空间中的地址分别为

$$A = a_{15} \dots a_k \dots a_{i+1} x_i \ a_{i-1} \dots a_{j+1} x_j \ a_{j-1} \dots a_0$$

$$B = b_{15} \dots b_k \dots b_{i+1} x_i \ b_{i-1} \dots b_{j+1} x_j \ b_{j-1} \dots b_0$$

显然,由于第 k 位取值不同, $A \neq B$,运用反真原理即可得证。

上述情况可以推广到任意条地址引线固定故障的情况。

证毕!

推论1:

若RAM模块地址引线 A_{15}, \dots, A_0 中有 n 条引线发生s-a- x_{in} ($x_{in}=0, 1$)故障,则 M_c 与 M 的地址空间是 2^n 对1的映射关系。 M_c 的映象是 M 的 $1/2^n$ 地址空间。

根据引理1,我们可以得出以下的定理。

定理1:

对 M_c 地址空间中任意一个地址 $a_{15}a_{14} \dots a_0$,可以构造表1中第二列所示的地址序列。当RAM模块地址引线 A_i 无故障时,序号 i 的地址和其它序号的地址(特别是序号 \bullet 的地址)必映

射到M地址空间的不同地址中;当模块的地址引线 A_i s-a- x_i ($x_i=0, 1$)故障时,序号i的地址和序号*的地址必映射到M地址空间中的同一个地址中。

表1

序号	MC空间中的地址																诊断数据							
	A_{15}	A_{14}	A_{13}	A_{12}	A_{11}	A_{10}	A_9	A_8	A_7	A_6	A_5	A_4	A_3	A_2	A_1	A_0	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
*	a_{15}	a_{14}	a_{13}	a_{12}	a_{11}	a_{10}	a_9	a_8	a_7	a_6	a_5	a_4	a_3	a_2	a_1	a_0	0	0	0	0	0	0	0	0
0	\bar{a}_{15}	a_{14}	a_{13}	a_{12}	a_{11}	a_{10}	a_9	a_8	a_7	a_6	a_5	a_4	a_3	a_2	a_1	a_0	0	0	0	0	0	0	0	1
1	a_{15}	\bar{a}_{14}	a_{13}	a_{12}	a_{11}	a_{10}	a_9	a_8	a_7	a_6	a_5	a_4	a_3	a_2	a_1	a_0	0	0	0	0	0	0	1	0
2	a_{15}	a_{14}	\bar{a}_{13}	a_{12}	a_{11}	a_{10}	a_9	a_8	a_7	a_6	a_5	a_4	a_3	a_2	a_1	a_0	0	0	0	0	0	1	0	0
3	a_{15}	\bar{a}_{14}	\bar{a}_{13}	a_{12}	a_{11}	a_{10}	a_9	a_8	a_7	a_6	a_5	a_4	a_3	a_2	a_1	a_0	0	0	0	0	1	0	0	0
4	a_{15}	a_{14}	a_{13}	\bar{a}_{12}	a_{11}	a_{10}	a_9	a_8	a_7	a_6	a_5	a_4	a_3	a_2	a_1	a_0	0	0	0	1	0	0	0	0
5	a_{15}	\bar{a}_{14}	a_{13}	\bar{a}_{12}	a_{11}	a_{10}	a_9	a_8	a_7	a_6	a_5	a_4	a_3	a_2	a_1	a_0	0	0	1	0	0	0	0	0
6	a_{15}	a_{14}	\bar{a}_{13}	\bar{a}_{12}	a_{11}	a_{10}	a_9	a_8	a_7	a_6	a_5	a_4	a_3	a_2	a_1	a_0	0	1	0	0	0	0	0	0
7	a_{15}	\bar{a}_{14}	\bar{a}_{13}	\bar{a}_{12}	a_{11}	a_{10}	a_9	a_8	a_7	a_6	a_5	a_4	a_3	a_2	a_1	a_0	1	0	0	0	0	0	0	0
8	a_{15}	a_{14}	a_{13}	a_{12}	\bar{a}_{11}	a_{10}	a_9	a_8	a_7	a_6	a_5	a_4	a_3	a_2	a_1	a_0	0	0	0	0	0	0	1	1
9	a_{15}	\bar{a}_{14}	a_{13}	a_{12}	\bar{a}_{11}	a_{10}	a_9	a_8	a_7	a_6	a_5	a_4	a_3	a_2	a_1	a_0	0	0	0	0	0	1	0	1
10	a_{15}	a_{14}	\bar{a}_{13}	a_{12}	\bar{a}_{11}	a_{10}	a_9	a_8	a_7	a_6	a_5	a_4	a_3	a_2	a_1	a_0	0	0	0	0	1	0	0	1
11	a_{15}	\bar{a}_{14}	\bar{a}_{13}	a_{12}	\bar{a}_{11}	a_{10}	a_9	a_8	a_7	a_6	a_5	a_4	a_3	a_2	a_1	a_0	0	0	0	1	0	0	0	1
12	a_{15}	a_{14}	a_{13}	\bar{a}_{12}	\bar{a}_{11}	a_{10}	a_9	a_8	a_7	a_6	a_5	a_4	a_3	a_2	a_1	a_0	0	0	1	0	0	0	0	1
13	a_{15}	\bar{a}_{14}	\bar{a}_{13}	\bar{a}_{12}	\bar{a}_{11}	a_{10}	a_9	a_8	a_7	a_6	a_5	a_4	a_3	a_2	a_1	a_0	0	1	0	0	0	0	0	1
14	a_{15}	a_{14}	a_{13}	a_{12}	a_{11}	\bar{a}_{10}	a_9	a_8	a_7	a_6	a_5	a_4	a_3	a_2	a_1	a_0	1	0	0	0	0	0	0	1
15	a_{15}	\bar{a}_{14}	\bar{a}_{13}	a_{12}	a_{11}	\bar{a}_{10}	a_9	a_8	a_7	a_6	a_5	a_4	a_3	a_2	a_1	a_0	0	0	0	0	0	1	1	0

定理1说明,若RAM模块某地址引线 A_i 发生s-a- x_i 故障,则CPU对序号i地址的读写实际上也是对序号*地址的读写,反之亦然。

利用定理1,我们可得出如下检测和诊断地址引线固定故障的方法:

(1) 检测

① CPU选择一组互不相同的8位数据(如表2.1第3列)作为诊断数据,按照表2.1序号*, 0, 1, ..., 15的顺序,把诊断数据写入相应序号的地址单元中;

② CPU读出序号*地址中的内容。若读出值与序号*的诊断数据相同,则RAM模块地址引线 A_{15} - A_0 无固定故障,否则存在固定故障(可能是单故障或多故障)。

(2) 诊断

① CPU按照表2.1中序号0, 1, ..., 15, *的顺序,向序号中相应的地址单元写入诊断数据;

② CPU按照表2.1中序号0, 1, ..., 15的顺序读出相应地址单元的数据。若序号i读出内容

与原先写入的一样,则地址引线 A_i 无故障;若序号i读出内容改变为序号*的诊断数据,则地址引线 A_i 存在固定故障(s-a-0或s-a-1故障)。

3. 地址引线 A_{N-1} - A_{16} 或选择信号 \overline{CS} 的诊断

设图1中地址译码的输出逻辑为 $\overline{CS} = \overline{a_{N-1}} \dots \overline{a_{17}a_{16}}$ ($a_i=0, 1; 16 \leq i \leq N-1$)。当地址引线 A_{N-1} - A_{16} 或选择信号 \overline{CS} 发生固定故障时, RAM模块将导致以下二类故障之一: (1) CPU访问本模块地址时, RAM模块没能被选通(即 $\overline{CS}=1$); (2) CPU访问非本模块地址时, RAM模块被选通(即 $\overline{CS}=0$)。下面对它们分别进行讨论。

(1) 第一类故障的检测

对于异步总线通讯机(如PDP-11机),发生第一类故障时CPU将发出“总线超时错”信号。对于同步总线通讯机(如DBJ-Z80),我们假定RAM模块数据引线中允许发生固定故障的线数小于数据线数的一半,例如8位数据线中最多允许其中3位发生固定故障,则第一类故障

可在数据引线 D_7-D_0 的诊断(本节1)过程中检测出。当数据总线为正(负)逻辑时,若CPU向RAM模块某单元写入全0(1),而读出的内容为1(0)的位数 ≥ 4 ,则我们不再认为数据引线中有 ≥ 4 根线发生 $s-a-1(0)$ 故障,而认为发生了上述的第一类故障。

图2.1中如果地址译码部分在CPU一侧,即总线直接提供RAM模块选通信号 \overline{CS} ,则第一类故障即 $\overline{CS}-s-a-1$ 故障;如果地址译码部分在RAM模块内,则第一类故障即存在某些地址引线 $A_i(16 \leq i \leq N-1)$ 发生 $s-a-a_i$ 故障,但不能诊断到每一根引线上。

(2) 第二类故障的诊断

对于总线直接提供RAM模块 \overline{CS} 选通信号的情况,第二类故障即 $\overline{CS}-s-a-0$ 故障;对于模块地址译码在RAM模块内的情况,第二类故障即RAM模块存在某些地址引线 $A_i-s-a-a_i$

$(16 \leq i \leq N-1)$ 故障。上述 $\overline{CS}-s-a-0$ 故障等价于这种情况下的所有 $A_i-s-a-a_i(16 \leq i \leq N-1)$ 故障同时发生。

$A_i-s-a-a_i(16 \leq i \leq N-1)$ 故障的诊断原理与地址引线 $A_{16}-A_0$ 的诊断(本节2)原理基本相同。仿照表1可以构造诊断地址和数据如表2。

表2

序号	MC空间中模块的地址			诊断数据							
	$A_{N-1} \dots A_{16}$	A_{17}	A_{16}	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
**	$a_{N-1} \dots a_{16}$	a_{17}	a_{16}	0	0	0	0	0	0	0	0
16	$a_{N-1} \dots a_{16}$	a_{17}	a_{16}	0	0	0	0	0	0	0	1
17	$a_{N-1} \dots a_{16}$	a_{17}	a_{16}	0	0	0	0	0	0	1	0
18	$a_{N-1} \dots a_{16}$	a_{17}	a_{16}	0	0	0	0	0	1	0	0
\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots
$N-1$	$a_{N-1} \dots a_{16}$	a_{17}	a_{16}								

类似于定理1,若RAM模块存在地址引线 $A_i-s-a-a_i(16 \leq i \leq N-1)$ 故障,序号**和序号 i 的模块地址必都选通故障模块。否则序号 i 地址不选通该RAM模块。(证明方法同引理1,证明略)。

根据这个原理,便可对每条地址引线 $A_i(16 \leq i \leq N-1)$ 进行诊断。

诊断方法:

对于异步通讯的总线,CPU凡是访问到总

线上不接有序号16, 17, ..., $N-1$ 对应模块地址的地址空间,CPU如果发出“总线超时错”信号,则该序号(设序号为 i)的地址引线 A_i 无故障,否则 $A_i-s-a-a_i$ 故障。凡是总线上序号16, 17, ..., $N-1$ 对应模块地址上接有其它RAM模块或其它部件,诊断方法与下述的同步通讯总线方法相同。

对于同步通讯总线,先对序号**,16,..., $N-1$ 各模块任选一个相同的模块内地址 $a_{16}a_{14} \dots a_0$ (如00H),然后分别对序号 $i(i=16, 17, \dots, N-1)$ 的模块地址作如下操作,即可诊断 $A_i-s-a-a_i(16 \leq i \leq N-1)$ 故障:

① CPU对序号**模块的 $a_{16}a_{14} \dots a_0$ 单元地址写入其诊断数据;

② CPU对序号 $i(16 \leq i \leq N-1)$ 模块的 $a_{16}a_{14} \dots a_0$ 单元地址写入其诊断数据;

③ CPU读出序号**模块的 $a_{16}a_{14} \dots a_0$ 单元地址的内容。若读出内容与序号**的诊断数据相同,则RAM模块地址引线 $A_i(16 \leq i \leq N-1)$ 无故障;若读出内容改变为序号 i 中的诊断数据,则引线 $A_i-s-a-a_i$ 故障。

三、诊断算法

根据诊断原理,下面我们以流程图形式给出诊断RAM模块数据引线 D_7-D_0 及地址引线 $A_{N-1}-A_0$ 的固定故障诊断算法。假定被诊断的RAM的模块地址为 $A_{N-1} \dots A_{17}A_{16}=a_{N-1} \dots a_{17}a_{16}$ 。

图2中,对于状态 A^*1 ,程序已不能继续进行其它部分的诊断,必须中断诊断而退出。对于状态 A^*2 及 A^*3 ,只要表1和表2中的诊断数据能根据数据引线故障情况动态地产生(本文尚未给出),可转到图3程序B继续诊断。

若计算机系统地址总线为 N 位,其中模块的块内地址为 N_1 位,模块的地址为 $(N-N_1)$ 位,则流程图2, 3和4中执行RAM读写次数分别为4次、 $3 \cdot (N-N_1)$ 次和 $2 \cdot N_1+1$ 次,总次数 $< 3N+5$ 次。整个算法复杂性为 $O(\ln n)$ 。(n为RAM模块存储容量)。

四、结语

本文对RAM模块中等价于地址引线和数据

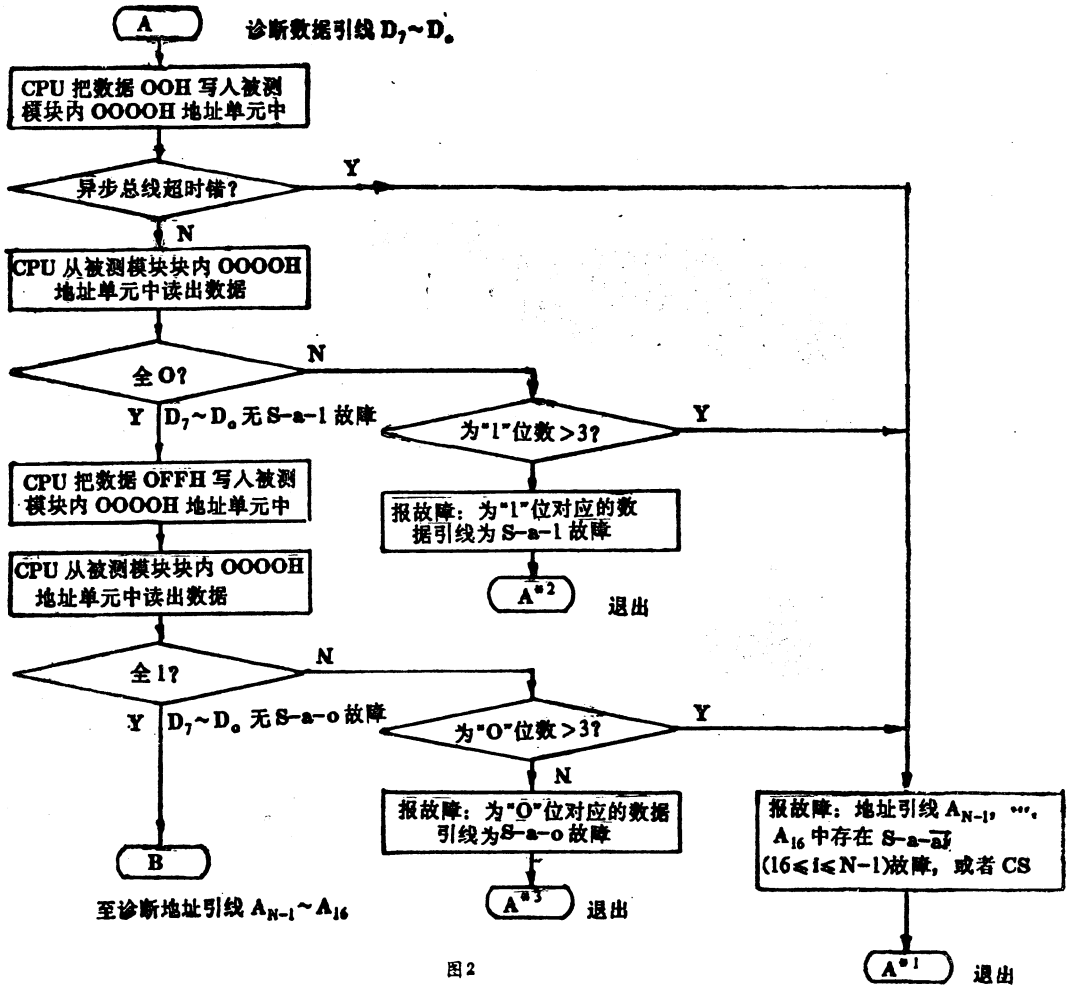


图2

引线的固定故障作了详细的讨论和理论分析之后,提出了适用于同步和异步两种总线通讯方式的故障诊断的算法。该算法可利用CPU及ROM自动地进行,也可通过CPU控制台面板手动方式进行,其复杂性为 $O(\ln n)$ (n 为RAM模块存储容量)。作者曾在DJS-183计算机半导体内存研究和制作^[6]过程中,通过控制台面板手动操作该诊断算法,对自制插件板的数据线和地址线引脚进行固定故障(主要是接插件接触不良和电缆线脱焊)的诊断,效果甚佳。本文对RAM模块中控制线部分的固定故障,尚未给出诊断算法,需要进一步探讨。目前已有一些文献^[6]在这方面作了不少工作。

参考文献:

1. 陈延槐、陈光熙, 数字系统的诊断与容错, 北京, 国防工业出版社, 1981年, 第132—134页。
2. M.A. Breuer & A.D. Friedman, Diagnosis & Reliable Design of Digital Systems, Computer Science Press Inc, 1976, PP139—146; PP156—160.
3. J.P. Hayes, "Detection of Pattern-Sensitive Faults in Random-access Memories", IEEE Trans. on Computer, Vol. C-24 PP150—157, Feb. 1975.
4. E.R. Hnatik, "4-Kilobit Memories Present a Challenge to Testing" Computer Design, PP118—125, May 1965.
5. 马永强, "DJS-183计算机半导体内存研制", 西南交通大学学报, 1984年第3期, 第58—61页。
6. L.H. Pollard & J.H. Patel, "Fault Tolerant Techniques for Control Signals in Bus Communication Protocols", FCTS-14, 1984, PP380—385.

打印机断针自动测试程序

黄河大学 叶志斌

打印机是计算机的重要外部设备,断针现象在打印机故障中并不少见,这在有图形输出及其它一些应用中是不允许的。以前我们都是把打印头拆下来仔细找出断针后再进行更换,这样既费时又不很准确。在工作实践中,笔者用大众语言Basic写了一个24针打印机针头自动测试通用程序,它在1分钟内把24针逐个检测一遍并告知检测结果:针号后有细短直线者为完好,否则,该针不是断了就是有问题。君不妨一试(最好在西文方式下使用)。

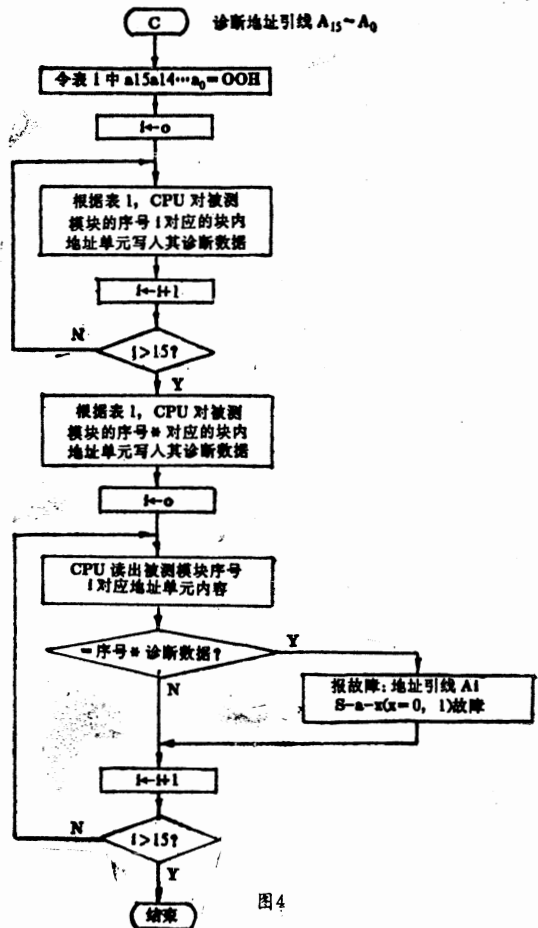
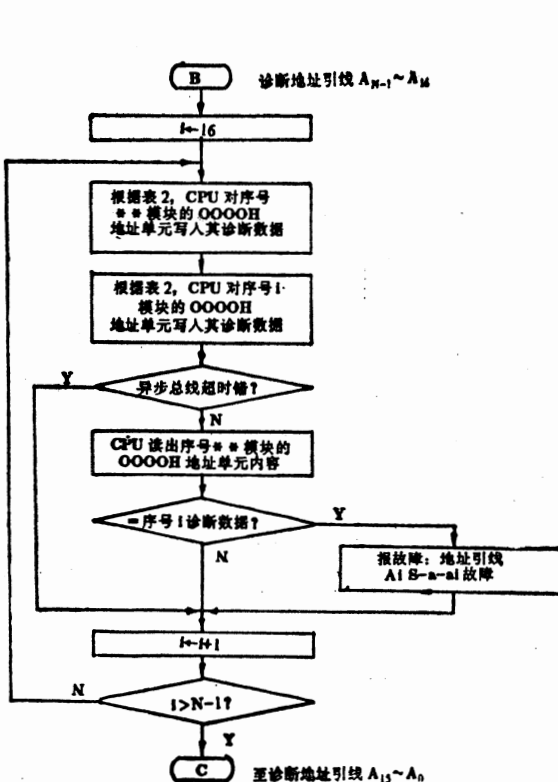
状态为准的,程序2则是测试DIP 7状态的程序。

```

5  REM Program 1: Printer pin testins
10 FOR K=0 TO 2
20 FOR I=1 TO 8
30 LPRINT "pin": 9-I+8*K;
40 LPRINT CHR$ (27): "4"; CHR$ (0); CHR$ (50);
50 P=2^(8-I)
60 L= (1-K) * (2-K) * P / 2
70 M=K * (2-K) * P
80 H= (K-1) * K * P / 2
90 FOR J=1 TO 50
100 LPRINT CHR$ (L); CHR$ (M); CHR$ (H);
110 NEXT J
120 LPRINT
130 NEXT I, K
  
```

程序1是以常用M2024打印机的DIP 7 off

程序1



FX—100 打印机不走纸故障检修一例

云南昆明工学院计算中心 张学云

故障现象：打印机自检时能开机，但不走纸（或走纸不对）。

分析：根据此故障现象，应从如下几个方面查找：

1. 走纸电机坏；
2. 有关元件坏，供给走纸信号不对；
3. 其它。

首先应判断走纸电机是否好，具体的方法是：用万用表的电阻档测试一下控制板上的 LF 插孔上时 1, 2, 3, 4, 5, 6 端是否对，即 1, 3, 5 是否相通，并且 2, 4, 6 是否相通，如果相通，则说明电机好，反之是不好的。在实际修理中，曾碰到过一例，测 1, 3, 5 这组通，但 2, 4, 6 这一组不通，将电机拆开，这组线圈已烧坏，重新换一个好的电机后，开机一切正常。

其次，如果在走纸电机好的情况下，开机自检时仍不走纸，则可具体查下 LE 插座上的 1, 2, 3, 4, 5, 6 的电平信号是否对，如果不对，再往前查，即查控制板上的 TRA2 的 2, 3, 4, 5, 6, 7, 8, 9 端电平是否对。因 TRA2 是一个集成功率管，其型号为 PU4120，它的 E 集连在一起接地，而 2 与 3, 4 与 5, 6 与 7, 8 与 9 分别构成四个管子，分别对应于 TR5, TR6, TR7, TR8。如果某一个管坏，可用分立元件管 D837 或 F03OG 来代替，只是需把 TRA2 对应端断开。在实际

修理中曾碰到 TRA2 的 2, 3 电平不对，在 TR5 位置上焊上一个 D837 管，并且将 TRA2 的 2, 3 断开，开机后一切正常，其它管坏，也可参照上述。

附：

为了便于查找我们给出如下

1. 黑的插孔(LF)

颜色	黄	白	兰	红	绿1	绿2
端数	1	2	3	4	5	9
	↑	↑	↑	↑	↑	↑

2. LF端的电平从左往右数

端数	1	2	3	4	5	6
正确值	0≈2V	0≈2V	0≈2V	0≈2V	0	0≈2V

3. TRA2各脚电平

脚数	1	2	3	4	5	6	7	8	9	10
电平	12.5V	12.5V	0≈2V	12.5V	0≈2V	12.5V	0≈2V	12.5V	0≈2V	12.5V

当它输出两条粗短线时为 off，一条粗短线时为 on。为 on 状态时，程序 1 输出的结果并不是打印头上针的物理顺序。此时把程序 1 的语句 30 改为 LPrint "Pin"; I+8*k; 50 改为 P=2^(I-1) 即可。

对于其它型号的打印机，仅把语句 40 改为相应的图形输出命令即可使用。

```

5 REM Program 2: Dip 7 on / off testins
10 LPRINT CHR$ (27): "4": CHR$ (0); CHR$ (80);
20 L=248
30 M=31
40 FOR J=1 TO 80
50 LPRINT CHR$ (L); CHR$ (M); CHR$ (0);
60 NEXT J
70 LPRINT
  
```

程序 2

《计算机应用研究》第7卷 第3期 (总第35期)

目 录

软 件 篇

一个简单的标准文稿格式打印程序	周 明 (1)
灵活的文件打印程序	刘建福 (2)
一个实用的方格纸格式文稿打印程序	徐雪斌 (5)
硬盘分区与DOS版本	彭 禾 (10)
DEBUG对硬盘分区的浅析	陈一侃 (12)
保护硬盘的DOS外部命令	王志中 (14)
逻辑硬盘“丢失”的预防及处理	彭 禾 (14)
再谈多种操作系统共享硬盘的方法	徐雪斌 (15)
抢占硬盘空间的有效方法	茹建平 (16)
长城0520-CH高中分辨率汉字系统同从硬盘自举的新方法	刘立强 (18)
提高IBM PC/XT硬盘空间利用率的一种有效方法	许传国 (19)
硬盘保护的一种方法	宋俊山 (21)
用BASIC语言调用汇编子程序的方法修复硬盘初始引导扇区	肖浩廷 郭松亮 (22)
程序动态加密	刘士军 (24)
在绘图机上绘写汉字的新方法	胡培民 (26)
道路选择辅助决策系统的功能与设计	王雄基 曾光初 (29)
一个自动生成表格的FoxBASE程序	樊金生 (31)
介绍一例简易实用的键控绘图程序	瞿 波 (33)
单缝衍射实验的计算机模拟分析	李志伟 (34)
APPLE-II上超长程序动态调试的实现	李 勇 (36)
软件汉化及一个辅助工具	杨传斌 (37)

硬 件 篇

如何用2716 EPROM代替16FDC板中的74918 ROM	叶富乐 (39)
--	------------

系 统 篇

计算机决策支持系统在铁路站场中的应用	朱怀芳等 (41)
一种简单易行的多机互连技术	龙卫红 (48)
微机控制X光测厚仪	胡士而 (51)

维 修 篇

3070打印机断针定位方法及8088汇编程序	龙兵生 (56)
IBM PC/XT硬盘维修及00磁道的调整	陈永青 (58)
长城0520-CH硬盘故障两例	梁 军 (59)
RAM模块地址线和数据线固定故障的诊断	马永强 (60)
打印机断针自动测试程序	叶志斌 (65)
PX-100打印机不走纸故障检修一例	张学云 (66)

信 息 篇

IBM PC/XT及其兼容机故障诊断仪问世	李相彬 (33)
电力工程建设信息管理系统制成	李相彬 (40)
介绍一个围棋软件	顾建林 (47)
微机计算计数器大批投产	李相彬 (54)
消息二则	(70)

APPLICATION RESEARCH OF COMPUTERS

Vol.7 No.3 (Total 35)

CONTENTS

SOFTWARE

A Simply Printing Program for Standard Paper Form of Manuscript	Zhou Ming (1)
The Flexible Printing Program of File.....	Liu Jianfu (2)
A Practical Printing Program for Squared Paper Form of Manuscript	Xu Xucbin (5)
Partitioned The Hard-disk and DOS Version	Peng He (10)
A Simply Analysis for Partition Hard-disk by DEBUG.....	Chen Yikan (12)
Protecting External Command of DOS on The Hard-disk	Wang Zhizhong (14)
Preventing and Handling about "Losing" of Logic Hard-disk.....	Peng He (14)
Once More Discussing about The Method for Many Operating System Shared Hard-disk.....	Xu Xuebin (15)
The Effective Method for Controlled Space of Hard-disk.....	Ru Jianping (16)
The New Method for Disk Self-imposed on GW 0520 CH High and Middle Resolution Chinese System.....	Liu Liqiang (18)
The Effective Method about Increasing Utilization Ratio of Hard-disk Space on IBM PC/XT.....	Xue Chuanguo (19)
A Method of Protecting Hard-disk.....	Song Junshan (21)
The Method for Resumed Initial Sector of Hard-disk by BASIC Language Calling Assembler Subprogram.....	Xiao Jiyan and Others (22)
The Method of Program Encryption with Dynamic.....	Liu Shijun (24)
The New Method for Drawing Chinese Character by Plotter	Hu Peimin (26)
The Function and Designing of Path-selecting by Auxiliary Policy-making System.....	Wang Xunji and Others (29)
A FoxBASE Program with Automatic Generated Table.....	Fan Jinsheng (31)
A Simply and Practical Example of Ploting Program by Keyboard Controlled.....	Qu Bo (33)
The Simulate Analysis for The Experiment of Diffraction with A Slit by Computer.....	Li Zhiwei (34)
Realizing for Debugging Longthy Program on APPLE-II Microcomputer	Li Yong (36)
Transforming to Chinese Character for Software and A Auxiliary Tool	Yang Chuanbin (37)

HARDWARE

How Replacing A 74918 ROM in The 16 FDC/S Card by A 2716 EPROM	Ye Fule (39)
--	----------------

SYSTEM

- The Application of The Computer Policy-making and Supporting System
in Railway Station..... Zhu Huaifang and Others (41)
- A Simply Technique for Multilevel Interconnection.....Long Weihong (48)
- The Testing Instrument for Controlled Thickness of X Ray by
Microcomputer.....Hu Shiyong (51)

MAINTENANCE

- The Method about Setting Locations for Cutting Points of 3070 Printer
and 8088 Assembler Program.....Long Bingsheng (56)
- The Maintenance for The Hard-disk of IBM PC/XT and Justification 00
Magnetic Track.....Chen Yongqing (58)
- Two Example about Maintenance The Hard-disk Fault on GW0520-CH
Computer.....Liang Jun (59)
- Diagnosising Faults of Fixed Address and Data Line on RAM Module
.....Ma Yongqiang (60)
- The Automatic Testing Program for Break Wire on Printer
.....Ye Zhibin (65)
- An Example for Overhaul The Breakdown of not Shifting Paper on
FX-100 Printer.....Zhang Xueyun (66)

INFORMATION

- Six Pieces of News..... (33, 40, 47, 54, 70)

APPLICATION RESEARCH OF COMPUTERS

(Publicly Publishing) (Bi Monthly)

May 1990 Vol.7 No.3 (Total 35)

Main Editor: Zhang Zhi Qian

Assistant Main Editor: Li Zemin

Responsibility Editor fhis this Issue: Qi Mo Zhi

Editor/Publisher: Editorial Department of This Periodical

Address: 11—1 Four Section Ren Ming Nan Road Chengdu, P.R.C

Press: South west metallurgical geology Printing house of

Ministry of Metallurgical Industry

Subscribe Address: The Post of the All Localities of the Country

Central Dispatching in Inside the Country; The Post of Chengdu

Unified Number in Inside the Country: CN 51—1196

Number of the Post Publishing: 62—68

Post Code: 610015

好消息

——全国首次期刊展览将在北京举办

为了检阅我国期刊出版事业的成就,加强对期刊出版工作的导向,扩大期刊的宣传,促进期刊繁荣,国家新闻出版署决定于1990年9月在北京举办全国期刊展览。参展期刊须是经过取得国内统一刊号公开发行的各类期刊。展览期间,除邀请各地高校、研究单位、图书馆人士和部分外国出版界人士参观外,还将举办多种有关期刊编辑、出版、发行、经营等方面的研讨、经验介绍活动,并将组织观众分类评选“我所喜爱的期刊”,组织有关专家评选“期刊整体设计奖”和“期刊印刷质量奖”等多项评奖活动。

3—990 型血液流变学系列仪器简介

血液流变学(Hemorheozogy)是研究血液及其组成成份的流动和变形的科学,近年来国外发展甚为迅速。它同基础医学、临床医学关系密切,在临床医学上测定血液、血浆或血清的流动和变形,以了解血液流动、变形及其生理、病理条件下的变化规律,因而对于许多疾病,尤其像冠心病、心肌梗塞、脑中风、高血压和肺心病等病因的研究、诊断和鉴别诊断,以及疾病的发生发展和愈后的判断、治疗、预防、药物等等作用原理的探讨和肿瘤的发生、转移的判断等均有及其重要的意义。

由四川省电子技术研究所和成都电子医仪厂联合研制和生产的血液流变学系列仪,品种齐全、质量可靠、操作简易、数据准确、性能价格比高,负责终身免费保修,并可免费接受华西医科大学专家、教授的培训和指导。

血粘仪: (电极式, 非电极式)

电极式 3-990型	1950元/台
------------	---------

非电极式 3-990 I 型	2850元/台
----------------	---------

3-990 II 型(含电脑、打印机全套, 可自动打印8个血流变学参数)	8000元/台
--------------------------------------	---------

细胞电泳仪: 3-990型(带电脑、打印机、显微镜等全套)	4850元/台
--------------------------------------	---------

血栓形成仪: 3-990 I 型	1970元/台
-------------------------	---------

3-990 II 型(带电脑、打印机等全套)	3850元/台
------------------------	---------

血小板粘附仪: 3-990 I 型	1970元/台
--------------------------	---------

3-990 II 型(带电脑、打印机等全套)	3850元/台
------------------------	---------

细胞压积仪: 3-990 I 型	1980元/台
-------------------------	---------

3-990 II 型(带电脑、打印机等全套)	3850元/台
------------------------	---------

血小板聚集仪: 3-990 I 型	1970元/台
--------------------------	---------

3-990 II 型(带电脑、打印机等全套)	3850元/台
------------------------	---------

肿瘤、冠心、中风、预报、监测仪: 3-990型(带电脑、打印机等全套)	7800元/台
--	---------

以上系列仪器经实际运行检测证明: 操作简便、测定快速、重复性好, 智能化程度高, 功能较全, 各项技术指标均符合各级医院和医学研究单位临床及科研实验要求, 欢迎来人来函来电联系洽谈。

通讯地址: 成都市人民南路4段11号附1号 四川省电子技术研究所

联系人: 李尧炯

邮政编码: 610015

电 话: 551858

电 挂: 成都 5938