

电脑 3 1990

中国软件行业协会会刊



美国SANTAK不间断电源系列产品



防静电地板

- 计算机软件, 计算机通讯系统, 多用户系统及局部网络管理系统系统的开发、推广与实施。
- 传感器, 计算机自动配料系统的开发与实施
- 各类电子装置, 自动装置和计算机部件的开发与生产。
- 电子控制系统, 地面卫星工作站, 程控电话系统的开发与实施。
- 大、中、小各类机房工程及机房配套工程的设计与实施。
- 计算机主机, 电子印刷与办公室自动化设备, 电子仪器, 计算机外设, 配件和消耗品批发零售与维修。
- 用美国HCR公司专利、全部进口原器件生产高能牌FSC-88自动配料计算机, 具有国际先进水平。
- 本公司为美国山特国际有限公司UPS系列产品代理。
- 本公司下属南海明珠防静电地板厂, 用国际规范、进口原材料生产高能牌AE-01型防静电地板, 畅销国内外。
- 本公司下设业务拓展部、计算机系统工程部、工业控制工程部, 电子工程部、通讯工程部、机房工程部、销售服务部。



FSC-88全功能秤量控制计算机



PS/2 微机

广州开发区高能计算机公司

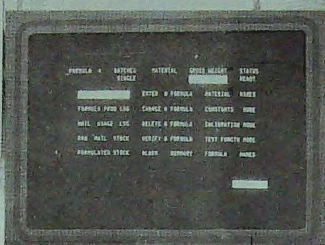
东莞市高能计算机应用工程公司

地址: 广州花园酒店东楼611室
电话: 338999-611 电挂: 4735-611
传真: 350467转7611 邮编: 510064
联系人: 廖仲琪 罗成伟

地址: 东莞市南城路
电话: 225397 电挂: 6933
传真: 225397 邮编: 510064
联系人: 高训嫻



▲我公司承接的广州电池厂工程



▲FSC-88 机正在出厂校验

◀ FSC-88 机显示

我公司按美国 HCR 专利全部进口件生产的

FSC-88 全功能配料控制计算机及配料控制系统

FSC-88 最大的特点是精度高、功能强、操作方便、适应面广、通用性强、抗干扰性强、稳定可靠；适用于任何配料及称量系统的控制，广泛应用于电池、玻璃、橡胶工业、合成洗涤剂工业、食品、医药工业、饲料及钢铁、冶炼工业，我们已成功地在电池、洗涤剂工业中应用并获得高度的评价。

The specialties of FSC-88 are high accuracy, full capability, easy to use, widespread flexibility, good repeat ability and dependable performance. It is supplied virtually every type of batching and weighing system. It is applied to the industries of rubber, glass, battery, synthetic detergent, food, medication, fodder, iron and steel, metallurgy etc. We have applied FSC-88 control system on the industries of detergent and battery and won high reputation.

■优点

- 称量准确，是产品质量的关键。
- 配料程序严格，保证生产稳定可靠。
- 操作简便，维修便利。
- 生产效率提高，增加产量，降低生产成本。

■功能及特性

- 操作简单——通过 CRT 显示，操作者可随时了解生产情况。
- 用简明的英语说明生产情况。
- 配方存储量大——可储存 96 个配方，每一个配方可有 24 种物料。

- 功能多样——有两块平面式键及打印机供您操作。

■称量及配料特点

1. 精确的自动截止控制保证高度的准确性。
2. 净重自动零位调节。

3. 自动除皮调节以便称斗残余量尽可能的少。
4. 自动零位误差校核。
5. 物料称量的超/差校核。
6. 自动补偿调节。
7. 比较参数自动调节。
8. 断电保护 1500 小时。
9. 每按动一次补进按钮，可有 1% 增量。
10. 称量数字显示大、易于观察。
11. 配料批量统计。
12. 显示总重、净重及预置重量。
13. 简便的接口，便于同上级计算机联络。
14. 结构紧凑，适用于工厂生产环境。

■技术性能

显示分度率	1-10000
称量分辨率	1 / 40000
精度	± 0.01% F.S
非线性	± 0.05% F.S
控制物料数量	24 种 (双速)
配方储存	24-96
工作环境温度	-10℃ - 40℃

■提供多种服务

- “钥匙”工程——对整个系统全面承包。
- 提供整套工程的设计，联合实施。
- 联合设计、分工实施。
- 提供系统技术及 FSC-88
- 承包原有配料系统的改造。
- 培训操作及维修人员，及时服务。
- 提供各种备品配件。
- 提供各种备品配件

广州经济技术开发区 高能计算机公司

中国科学院广州电子技术研究所

智通电子公司

(原电子系统工程部)

本公司是一个多年从事研究、开发和应用计算机网络系统,工业自动化控制与监测系统,计算机信息处理系统、遥测遥控系统,电子工程(包括卫星接收天线、公共天线等)的高技术单位,以中国科学院为后盾,拥有雄厚的经济实力和众多的各类高技术人材。曾经承接并完成了卫星地面站自动检测系统,大型水库水文遥测系统,电站自动监控系统,由二十七台 PS/2 微机构成的 3⁺以太网计算机信息管理系统等。有多年计算机应用的经验,可在较短时间内为用户提供电子系统工程的各种研究、开发、应用和系统设计。

公司竭诚为各界用户服务,并提供:

- 一、人材培训,技术咨询,各种计算机资料和软件。
- 二、计算机网络系统,工业自动化系统,遥测遥控系统,电子工程等软、硬件开发、研究、设计、工程。
- 三、中国科学院的各种新产品,各种进口元器件、数字仪表、通讯设备、示波器等。
- 四、美国 AST、PS/2 及各种兼容 486、386、286 微机。
- 五、高抗干扰电源、UPS 不间断电源、各种打印机、中英文打字机、各种电子电脑配件。

公司地址:广州市先烈中路 100 号大院内电子所实验楼五楼

公司电话:753245—4131,1551 775600—493

邮政编码:510070 图文传真:753247

联系人:李海华 姚智鹏

门市部地址:广州市天河科技街第二座 175 号

门市部电话:510450

邮政编码:510630

联系人:欧 德



AIHUA FLOPPY DISK



伯樂識駿馬 軟盤推 Aihua

愛華軟磁盤公司

AI HUA FLOPPY DISK CO.

地址：深圳市深南中路24號愛華24號樓 郵政編碼：518044
郵政信箱：深圳市3-018信箱 電話：351154 電掛：1311

Flat 14 Aihua Shenzhen Zhong Road Shen / China
Tel: 351154 Fax: 350122 Cable: 1314 Tlx: 350122 ABC CN

本公司為教育作貢獻，對全國大、中學校學生特價優惠360KB (Ⅲ) 軟磁盤¥ 2元/片

國內統一刊號：CN 44-1188 郵發代號：46-115



广州白云山电源设备厂

CWY系列高抗干扰稳压电源

我厂是生产稳压器、变压器、变压器铁芯的省电子局定点厂，具有十多年的生产历史，是我国生产各类电源设备及其配件的骨干企业。所生产的CWY系列高抗干扰交流参数稳压电源是我厂84年研制成功的国内首创稳压电源，通过国家技术部门鉴定。技术性能优越，比国内其他类型稳压电源有更明显的优越性。86年获国家科技进步奖，三相抗干扰稳压电源也由中国科学院广州分院通过技术鉴定。88年获产品专利权。89年获中国科学院技术进步三等奖。

本厂是生产CWY系列高抗干扰稳压电源的较早厂家，系列齐全。近年来经过工程技术人员攻关，对噪音、空载电流和漏磁干扰都取得了相当完善的解决，所以我厂产品与市场上同类型产品相比有更优越的性能，赢得了国内计算机用户、使用高精仪器设备等企、事业单位的信赖和广泛的应用，为我国电源系列发展作出卓越贡献。

购买时，请认明商标，提防有人冒盗本厂技术制造伪劣产品，使用户造成不必要的损失。

本产品已向中国人民保险公司办理了全国范围产品责任保险，用户可放心使用。

单相抗干扰稳压器系列

一、型号及规格

型号	350	500	1K	2.2K	3.2K	5.2K	10K	15K
容量	350VA	500VA	1KVA	2.2KVA	3.2KVA	5.2KVA	10KVA	15KVA

二、主要技术参数

●输入电压单相交流220V50Hz

●电压稳定度

输入电压范围	输出电压稳定度
176V~264V	$\Delta U/U < \pm 1\%$
160V~264V	$\Delta U/U < \pm 2\%$
140V~300V	$\Delta U/U < +2/-7\%$

全國範圍產品責任保險



- 无过压危险，有自动短路保护特性
- 总谐波失真度 $<4\%$ （开关式电源负载）
- 对电网振铃干扰或尖脉冲干扰抑制能力符合国际计算机电源要求
- 应变时间（输入电压跳变 $\pm 100V$ ） $<10\sim 30ms$
- 有效功率可达到80~90%
- 音频噪声低，位于1.6米距离处50dB

三相抗干扰稳压器系列

一、型号：3KV、6KV、10KV、15KV、30KV

二、主要技术参数：●干扰抑制抑制：输入脉冲 $<400V$ ，输出不发现；输入脉冲 $>1000V$ ，输出 $<100V$ 。

- 稳压度调整率：输入 $-40\%\sim +40\%$ ，输出 $-4\%\sim +1\%$
- 音频噪声低：位于1.6米距离处52db。
- 效率高：89%。比一般电子交流稳压器效率高1/5。这对节能很有意义。

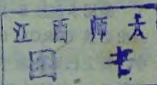
本单相、三相电源已与VAX11/750，VAX11/785，PDP11/44连机试验，运转正常，性能良好，广泛适用于计算机、自动控制设备，电子显微镜，X光CT，核磁共振断层扫描仪等高、精、尖设备等。与国外同类产品性能相等。

厂长：贝远娥 副厂长、工程师：陈自如 厂址：广州市沙河同和 邮政编码：510515

开户银行：广州农行白云营业所 账号：80-431032 电挂：0839 电话：705764转23 705665转343

注册商标 质量三包 欢迎来人来函订购 多谢惠顾

高抗干扰
安全可靠
功能特殊
電腦必備



深圳益中科技发展有限公司为
IBM-PS/2、30、50、70、80 等机型，开
发的具有通用型的外接 5.25 英寸软盘驱动
器，适用于 IBM-PS/2 上使用
IBM-PC/XT 和 AT 系列微机软件，自登
出广告后，受到 PS/2 用户的欢迎和支
持，为了方便用户，特设如下代理：

广州石牌华南师大《电脑》杂志社

邮码：510631 电话：516792

广州市北京路 356 号新大新公司电脑商场

邮码：510030 电话：335570

广州市广仁路四号之二广东省教育服务公司

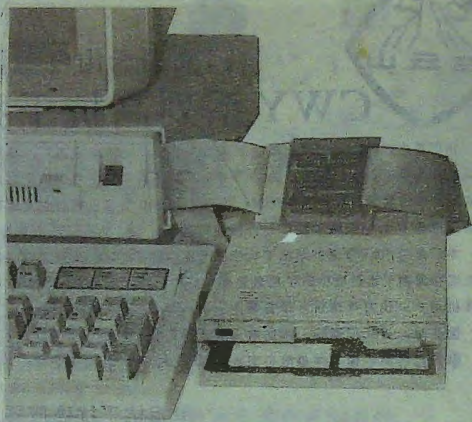
邮码：510030 电话：352553

北京中关村路 15 号中科院力学所新技术开

发公司 邮码：100080 电话：2564089

零售价：2650 元/套 批发面议

征询各地经销代理，详情请与《电脑》
杂志社联系



广州袖珍计算机技术服务中心报价单

PC-1500 计算机功能扩展系列器件

(一) GE-1 型万字汉卡 零售价 120 元

可用 CE-150 打印机输出矢量汉字 11000 多个。

(二) E-BASIC 系统卡 零售价 120 元

可为你增加近百个原 SHARP 公司机器没有的指
令，其中包括：

1、矩阵运算指令 10 种。

2、程序编辑指令 5 种。

3、新增函数和语句 32 种。

4、机器语言开发工具 DEBUG 指令 9 种，包括监
控、反汇编、内存码输出、单步跟踪等，使你有了学习机
器语言及进行二次开发必不可少的强劲工具。

5、正常区及井号区文件管理指令二十多种。

(三) GE-2 型两用卡 零售价 210 元

该卡既是万字汉卡又是 E-BASIC 系统卡，体积仅为
58×92×18MM，携带方便。

(四) GE-3 型多用卡 零售价 320 元

多用卡包括 E-BASIC 系统及万字汉卡功能，并配可
自由更换 EPROM IC 的插座，插上汉字库 IC 可打印汉
字。换上已固化程序的 EPROM IC，就可高速读出所固
化程序（1 秒钟可调出 8K 程序），可代替磁带机。

袖珍机及其配件

PC-1500A 计算机（日本原装）全套 1850.00 元

CE-152 磁带机

320.00 元

CE-161 模块（16K）

180.00 元

CE-158 接口

750.00 元

进口四色打印笔

18.00 元

打印纸（57×30×8）

0.55 元

60 针插座（弯脚）

45.00 元

60 针插头（弯脚）

45.00 元

60 针插头（直脚）

15.00 元

打印机小齿轮

5.30 元

打印头梅花簧片

5.20 元

CE-150 打印机 9V 电源

80.00 元

S15P 四色绘图机

1150 元

S15P 绘图机汉字卡

420 元

最新型“大力神”袖珍机

此为美国 ATARI 原装进口机，63 键 CPU 80C88，
内存 128K，可扩充到 640K。另有 128K 模块作为配件，
供选购，此机尺寸小，20mm×10mm×30mm，具有串、
并行接口，液晶显示 8×40，直流 4.5 伏供电。

工程计算器价格：

卡西欧 fx-4000P，程序步 550，固定存储器 26 个，
价：298 元

卡西欧 fx-4500P，程序步 1103，固定存储器 26
个，价：370 元

广州东风东路 745 号 510080 广州袖珍计算机技术服务中心 电话：750125-235 电挂：5051

长途直拨：020770353 开户银行：广州建行天河支行环市东办 帐号：208-2612123

向您推荐本公司专利产品

●万能拷贝 Copy Most

基于一种全新的拷贝思想，它撇开软件的加密手段和载体，当被加密软件运行时进行自动跟踪，直接拷贝程序的有用代码本身，功能强于 CopyRight。

经试验，能对 PC / XT 激光加密盘（如“指纹盘 PROLOCK, HSLOCK），特殊扇区技术加密盘（如 SuperLock），掩膜加密盘、智能加密卡及其他国内外流行的加密盘（如双重加密 LOCK 和 LOCKER）等进行备份拷贝，而且可以拷贝到硬盘。

零售价每份 120 元，批发价 100 元

●HEY-2 型汉英语音发生器

可使 APPLE 和中华学习机讲话

只需输入拼音字母串即可发出对应的声音，有四声变调，可在机器语言及高级语言使用情况下调用发音功能，声调、声速及插座号均可变动。

零售价每套 200 元，批发价 170 元

欢迎邮购

广州经济技术开发区华新技术开发有限公司

银行帐号：广州农行新技所 263-501030 邮码：510630 电话：510243 地址：广州五山路科技街 123 号

西安电子科技大学广州开发区分公司

愿为您提供下列器材和服务。

一、EE1108 逻辑分析仪：最高可测频率为 20MHz，触发延迟：0~9999 个时钟周期，存储容量为 1024 点 / 通道。该分析仪与示波器相连，可直接观测波形，与 PC / XT 机相连，可显示、存储、打印复读、比较波形。该分析仪有八十六通道两种都具有专用逻辑分析仪的功能，与同类产品相比价格低廉，是微机开发、数字系统研究和故障分析的理想工具。

二、打印机共享控制器：可为多至四台计算机服务，使它们共用一台打印机，而且无需您在使用时拨动开关，是非联网用户减少机房投资的理想选择。

三、JOB-1 型八通道肌电仪：采集、输出八道肌电信号，关节声监听装置，计算机标准接口，可配光线示波器、磁带记录仪、数据处理仪，用于面部神经肌肉、生物电的测试。

四、各种常用 IC（74LS 系列、400 系列、45 系列、74HCT 系列、74F 系列）各种协处理器、CPU、单片机、EPROM、PROM、RAM，报价单函索即寄。

本公司还可为各行业朋友提供计算机技术类、自动控制类、电子类等的技术服务，承接研制任务、协作开发产品等技术类工作更是本公司的特长，欢迎各界朋友来人来函洽谈。

本公司地址：广州天河石牌五山路科技街 294 号

邮政编码：510630 电挂：1949 转 294

电话：510326

联系人：周建中 王拴柱

电脑应用

- 控制系统的微分方程数字仿真..... 胡泽滋 (5)
自动发报程序..... 姚根龙等 (10)

综 述

- 县级计算机发展状况的调查与思考... 周林官等 (12)

软件纵横

- MS-DOS 彻底剖析 (七) DOS 内部命令的实现
原理 (下)..... 郭嵩山等 (13)
SAS 语言介绍..... 董大钧 (16)
修改 EXE 文件中说明部分的新方法... 王延亮 (18)

大学生之页

- IBM PC 机 BIOS 数据区的利用..... 王宇栋 (20)
APPLE EDIT 文本编辑程序..... 傅增明 (21)

使用与维修

- 3070 打印机断针定位方法及 8088 汇编程序
..... 龙兵生 (23)
中华机故障检修方法..... 任强 (26)
AR2463 打印机故障排除一例..... 陈文坤 (27)

计算机辅助教学

- 随机点分布图象通用程序..... 王克勤 (28)

中华学习机

- 中华机上配置简易 D/A 转换接口..... 杨宪泽 (31)

中学天地

- 高分辨率图形的放大打印..... 臧威成 (19)
对 $\mu P-1$ 微电脑套件的改进意见..... 关本立 (33)

竞赛辅导

- 谈谈数据结构..... 黄思曾 (35)

电脑用户

- 《Dbase-III 数据库文件的加密和解密试验》
的补充..... 林沛 (34)
IBM PC 机并行通信的实现..... 杨瑞生 (40)
在屏幕上同时显示三种字体的方法..... 翁晓刚 (45)

单板机与单片机

- 动手设计一个 6502 单板计算机..... 周仲玉 (46)

厂家与产品

- 广州经济技术开发区高能计算机公司... (封面、封二)
广州白云山电源设备厂 CWY 系列高抗干扰稳压
电源..... (1)
广州袖珍计算机技术服务部报价单..... (2)
西安电子科技大学广州分公司..... (3)
广州经济技术开发区华新技术开发有限公司..... (3)
IBM-PS/2 系列微机用户佳音..... (2)
苹果、中华机软件大赠送..... (48)
中国科学院广州电子技术研究所..... (封三)
爱华磁盘..... (封底)

☆☆☆☆☆☆☆☆☆☆☆☆☆☆☆☆☆☆☆☆☆☆

编 辑 《电脑》编辑部

(地址: 广州市石牌华南师范大学微电子学研究所)

电话: 516792 邮编: 510631

出 版 电脑杂志社

印 刷 韶关二九〇研究所地图彩印厂

总发行处 韶关市邮电局

订 阅 处 全国各地邮电局、所

定 价 1.00 元

出版日期 90年6月30日

控制系统的微分方程数字仿真

贵州工学院机械系 胡泽滋

提 要

本文主要研究在给定控制系统微分方程的情况下,通过其对应的传递函数进行数字仿真。文中重点的分析,论述了这种结构形式仿真的原理、方法;详尽地给出了具有通用性的仿真程序。此外,还通过一个实例说明了该程序的具体应用及注意事项,并给出仿真结果的数据和图形,分析仿真程序的调试,参数修改方法。

本仿真程序是在APPLE-II机上通过的。

关键词: 控制系统

数字仿真

传递函数

工程上的控制系统一般是由有限个具有集中参数元件所组成的动力学系统。这种系统均可用一个 n 阶常微分

方程来描述;在该方程中,时间 t 是独立变量。为了对该种单输入、单输出线性控制系统进行数字仿真,可应用矩阵方法,把 n 阶线性微分方程化成 n 个一阶微分方程组;而每个一阶微分方程都可以用数值积分法来求解。然后利用数字计算机进行 n 次循环,那么就很容易将整个系统的动态特性全部解出来;同时应用计算机的绘图功能,画出其动态响应曲线。此外,还可以非常方便地在计算机上修改输入量的大小,直接从计算机屏幕或打印机上获得动态特性所要求的指标,从而为控制系统实际的物理结构,提供必要和可靠的设计参数。

一、数学模型

众所周知,对于集中参数线性定常系统来说其动态特性总可以归结为一个 n 阶常微分方程,即:

$$\frac{d^n y}{dt^n} + a_1 \frac{d^{n-1} y}{dt^{n-1}} + \dots + a_{n-1} \frac{dy}{dt} + a_n y = C_0 \frac{d^{n-1} u}{dt^{n-1}} + C_1 \frac{d^{n-2} u}{dt^{n-2}} + \dots + C_{n-1} u \quad (1)$$

其中 y 是系统的输出量, u 是系统的输入量。若引进算子:

$\frac{d}{dt} \triangleq P$ 则(1)式可写为:

$$P^n y + a_1 P^{n-1} y + \dots + a_n y = C_0 P^{n-1} u + C_1 P^{n-2} u + \dots + C_{n-1} u$$

$$\text{即: } \sum_{j=0}^n a_{n-j} P^j y = \sum_{j=0}^{n-1} C_{n-j-1} P^j u \quad (\text{其中 } a_0 = 1) \quad (2)$$

或:

$$\frac{y}{u} = \frac{\sum_{j=0}^{n-1} C_{n-j-1} P^j}{\sum_{j=0}^n a_{n-j} P^j} \quad (3)$$

对(1)式两边取拉普拉斯变换,假设 y 及 u 的各阶导数(包括零阶)的初值均为零,则可得:

$$S^n Y(S) + a_1 S^{n-1} Y(S) + \dots + a_{n-1} S Y(S) + a_n Y(S) = C_0 S^{n-1} U(S) + C_1 S^{n-2} U(S) + \dots + C_{n-1} U(S) \quad (4)$$

其中: $Y(S)$ 是输出量 $y(t)$ 的拉普拉斯变换。

$U(S)$ 是输入量 $u(t)$ 的拉普拉斯变换。

即:

$$\frac{Y(S)}{U(S)} = \frac{C_0 S^{n-1} + C_1 S^{n-2} + \dots + C_{n-2} S + C_{n-1}}{S^n + a_1 S^{n-1} + \dots + a_{n-1} S + a_n} = G(S) \quad (5)$$

将(5)与(3)比较一下,可知在初值为零的情况下,用算子 P 所表示的式子与传递函数 $G(S)$ 在形式上是完全相同的。

若对(1)式引进 n 个状态变量: x_1, x_2, \dots, x_n
即 $x_1 = y$

$$x_2 = \dot{x}_1 = \frac{dy}{dt}$$

$$\dot{x}_n = \dot{x}_{n-1} = \frac{d^{n-1}y}{dt^{n-1}}$$

设

$$\sum_{j=0}^n a_{n-j} p^j x = u \quad (6)$$

$$\text{并令 } p^j = x_{j+1} \quad (j=0, 1, 2, \dots, n-1)$$

$$\therefore \sum_{j=0}^{n-1} a_{n-j} x_{j+1} + a_0 p^n x = u$$

若 $a_0 = 1$ 则得:

$$p^n x = \dot{x}_n = - \sum_{j=0}^{n-1} a_{n-j} x_{j+1} + u$$

写成矩阵向量形式为:

$$\dot{X} = Ax + Bu$$

$$\dot{X} = \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \vdots \\ \dot{x}_n \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \\ -a_n & -a_{n-1} & -a_{n-2} & \cdots & -a_1 \end{bmatrix} X$$

$$+ \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 1 \end{bmatrix} u$$

将(6)式代入(2)式得:

$$\sum_{j=0}^n a_{n-j} p^j y = \sum_{j=0}^n C_{n-j-1} p^j \sum_{i=0}^n a_{n-i} p^i x$$

则

$$y = \sum_{j=0}^{n-1} C_{n-j-1} p^j x = \sum_{j=0}^{n-1} C_{n-j-1} x_{j+1} = Cx$$

$$y = [C_{n-1}, C_{n-2}, \dots, C_0] \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$$

显然: A, B, C 三个系数矩阵为

$$A = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \\ -a_n & -a_{n-1} & -a_{n-2} & \cdots & -a_1 \end{bmatrix} \quad B = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix}$$

$$C = [C_{n-1}, C_{n-2}, \dots, C_0]$$

二、仿真模型

要将上述数学模型用计算机程序加以实现, 从而进行数字和图形的仿真, 首先要实现状态方程中各矩阵的存储和运算。因此程序中就必须安排一个二维数组 A (N, N), 一维数组 B (N), C (N) 作为 A, B, C 三个基本矩阵的存储空间。此外, 对于 A 阵中的系数 a_1, a_2, \dots, a_{n-1} 由于在运算中要经常调用, 所以还应有一个一维数组 Z (N) 来存放。而状态变量 x_1, x_2, \dots, x_n 则用数组 x (N) 加以保存。

仿真的核心在于解这一系列的一阶微分方程, 这就要用到数值积分法的有关内容。此时把仿真模型写成传递函数的形式最为方便, 由前 (4) 式知

$$G(S) = \frac{Y(S)}{U(S)} = \frac{C_0 S^{n-1} + C_1 S^{n-2} + \cdots + C_{n-2} S + C_{n-1}}{S^n + a_1 S^{n-1} + \cdots + a_{n-1} S + a_n}$$

使用时用户只要输入传递函数的阶次 n, 以及系数 $C_0, C_1, \dots, C_{n-1}; a_1, a_2, \dots, a_n$, 而不必去填写仿真模型, 这种做法当然比一个一个的微分方程填写其相应内容和数据要方便得多。

剩下的问题是如何解这一系列的一阶微分方程呢? 常用的是四阶 RUNGE—KUTTA 法, 因为它既有较快的收敛速度又有足够的计算精度。

设控制系统的微分方程组为:

$$\frac{dx_1}{dt} = f_1(t, x_1, x_2, \dots, x_n)$$

$$\frac{dx_2}{dt} = f_2(t, x_1, x_2, \dots, x_n)$$

$$\frac{dx_n}{dt} = f_n(t, x_1, x_2, \dots, x_n)$$

则四阶 RUNGE—KUTTA 法的公式, 以第 i 个状态变量的表达式为:

$$x_i^{(1)} = x_i^{(0)} + \frac{h}{6} (K_{i1} + 2K_{i2} + 2K_{i3} + K_{i4})$$

$$K_{i1} = f_i(t_0, x_1^{(0)}, x_2^{(0)}, \dots, x_n^{(0)})$$

$$K_{i2} = f_i(t_0 + \frac{h}{2}, x_1^{(0)} + \frac{h}{2} K_{i1}, \dots, x_n^{(0)} + \frac{h}{2} K_{i1})$$

$$K_{i3} = f_i(t_0 + \frac{h}{2}, x_1^{(0)} + \frac{h}{2} K_{i2}, \dots, x_n^{(0)} + \frac{h}{2} K_{i2})$$

$$K_{i4} = f_i(t_0 + h, x_1^{(0)} + h K_{i3}, \dots, x_n^{(0)} + h K_{i3})$$

$$\text{在本控制系统中 } f_i = \frac{dt_i}{dt} = a_{i1} x_1 + a_{i2} x_2 + \cdots + a_{in} x_n + b_i u$$

其中: $i = 1, 2, \dots, n$

$$\text{则 } x_i^{(1)} = x_i^{(0)} + \frac{h}{6} (K_{i1} + 2K_{i2} + 2K_{i3} + K_{i4})$$

其中:

$x_i^{(1)}$ 表示 $t = t_1 = t_0 + h$ 时的 x_i 值

$x_i^{(0)}$ 表示 $t = t_0$ 时的 x_i 值

$K_{11}, K_{12}, K_{13}, K_{14}$ 可通过右端函数 f_i 求出

为了使程序设计紧凑些, 我们引入一个有四个分量的一维矢量

$$h_j = (0, \frac{h}{2}, \frac{h}{2}, h)$$

和一系列零矢量

$$K_{10} = 0 (i = 1, 2, \dots, n)$$

这时四组系数的公式可以合成一个

$$K_{1i} = f_i \left(t_0 + h_j, x_1^{(0)} + h_j K_{11-i}, x_2^{(0)} + h_j K_{21-i}, \dots, x_n^{(0)} + h_j K_{n1-i}, u(t_0 + h_j) \right)$$

其中:

$$i = 1, 2, \dots, n \quad j = 1, 2, 3, 4$$

因为在计算各状态变量 x 时都要计算一次 K_{1j} , 所以把 K_{1j} 的运算单独作为一个子程序以供频繁调用最为方便。而 RUNGE—KUTTA 法中的系数 K_{1j} 用数组 $K(N, 4)$ 表示, 在设计 $K(N, 4)$ 的子程序时, 为使计算过程和程序简明起见, 引入两个中间工作单元 $P(N, 4)$ 和 $D(N, 4)$; 而引入的四个分量的一维矢量 h_j , 则由数组 $H(4)$ 储存。

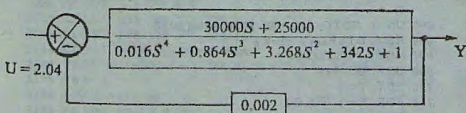
三、仿真程序及其结构特点

本程序采取模块式结构与菜单选择技术, 程序模块包含: 输入程序块、初始程序块、功能程序块、输出程序块及子程序块。

输入程序块是输入系统参数初值、计算步长、打印时间间隔、打印点数 (或计算数据数)、阶跃幅值。初始程序块主要对所需数组定维, 使所有工作单元置初值 (或置零)。功能程序块可根据用户需要由菜单选择不同的运算与输出; 计有: 屏幕显示仿真曲线、屏幕显示仿真数据、打印机打印仿真曲线、打印机打印仿真数据。每项功能程序执行结束后均可自动返回功能程序块, 再供用户选择要用的其他功能程序或退出。运算程序块用来求解系统的动态响应, 是本程序的核心。输出程序块与子程序块专供运算程序块及功能程序块随时调用。

四、程序调试及运算实例

现有一控制系统, 其传递函数方块图如下:



$$\text{则 } G(S) = \frac{30000S + 25000}{0.016S^4 + 0.864S^3 + 3.268S^2 + 342S + 1}$$

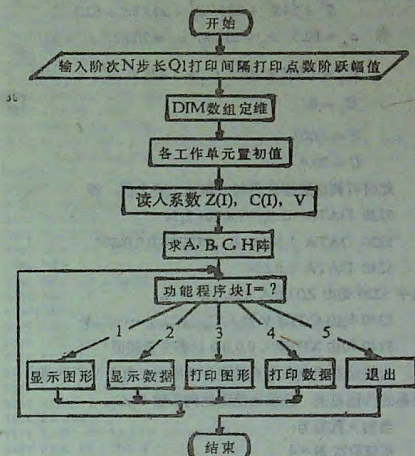


图1 主程序框图

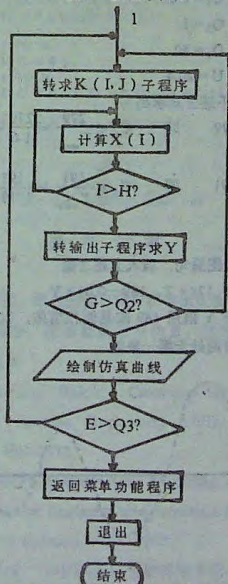


图2 功能程序框图

为使 $G(S)$ 符合标准形式, 即使 $a_0=1$, 可用 0.016 同除分子分母, 则上式变为:

$$G(S) = \frac{1.875 \times 10^{-6} S^4 + 1.562 \times 10^{-6} S^3 + 54 S^2 + 204.2 S^2 + 213.8 S + 62.5}{S^4 + 54 S^3 + 204.2 S^2 + 213.8 S + 62.5}$$

即 $a_4 = 62.5$, $a_3 = 213.8$, $a_2 = 204.2$, $a_1 = 54$

$$C_3 = 1.562 \times 10^{-6}, C_2 = 1.875 \times 10^{-6}, C_1 = 0,$$

$$C_0 = 0$$

$$V = 0.002$$

$$U = 20.4$$

此时可调用数据区语句, 填入上述之值, 即

5220 DATA 62.5, 213.8, 204.2, 54

5230 DATA 1.562E+6, 1.875E+6, 0, 0, 0.002

5240 DATA 0, 0, 0

其中 5220 句由 Z(I) 读入 a_n, a_{n-1}, \dots, a_1

5230 句由 C(I) 与 V 读入 $C_n, C_{n-1}, \dots, C_0, V$

5240 句由 X(I) 读入 0, 0, 0, 0 状态变量初值

为使动态响应曲线充满高分辨率第二页而又不致超出

屏幕的作图范围, 应求出绘图比例系数 C_1, C_2 .

当输入数据为:

系统阶次 $N=4$

计算步长 $Q_1=0.02$

打印间隔 $Q_2=1$

打印点数 $Q_3=80$

阶跃幅值 $U=20.4$

比例系数可按下述公式求出

$$T_{max} \cdot C_1 \leq 279 \quad \text{即} \quad C_1 \leq \frac{279}{T_{max}} = \frac{279}{1.6} = 174$$

$$Y_{max} \cdot C_2 \leq 191 \quad \text{即} \quad C_2 \leq \frac{191}{Y_{max}} = \frac{191}{17196}$$

$$= 0.011$$

此时可调用绘图语句, 填入上述之值

2250 HPLLOT 174 * T, 189 - 0.01 * Y

其中输出座标 Y 值用 189 减是倒相措施, 以使图形符合人们习惯而处于第一象限

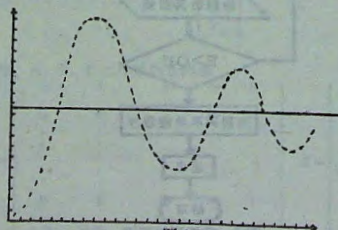


图 3

```

J LIST
10 HOME
20 VTAB (5), HTAB (3)
30 PRINT " use the transmit function proceed"
40 PRINT, PRINT, HTAB (3)
50 PRINT " digital simulation for control system", PRINT,
  PRINT, HTAB (12)
60 PRINT " program by H. z. z", PRINT, PRINT,
  HTAB (20)
70 PRINT " 1989. 8. 24"
80 VTAB (24), HTAB (20), GET QS
90 HOME
100 INPUT " input system number of order, n=", N, PRINT
110 INPUT " input integral size, q1=", Q1, PRINT
120 INPUT " input print interval, q2=", Q2, PRINT
130 INPUT " input print point number, q3=", Q3, PRINT
140 INPUT " input amplitude, u=", U, PRINT
150 DIM H (4), K (N, 4), X (N), P (N, 4), Z (N), D (N, 4),
  A (N, N), B (N), C (N)
160 FOR I=0 TO N
170 LET B (I)=0
180 LET C (I)=0
190 LET Z (I)=0
200 FOR J=0 TO 4
210 LET K (I, J)=0
220 LET P (I, J)=0
230 NEXT J
240 FOR J=0 TO N
250 LET A (I, J)=0
260 NEXT J
270 NEXT I
280 FOR I=1 TO N
290 READ Z (I)
300 NEXT I
310 FOR I=1 TO N
320 FOR J=1 TO N
330 IF I=N GOTO 380
340 IF J=I+1 GOTO 360
350 GOTO 390
360 LET A (I, J)=1
370 GOTO 390
380 LET A (I, J)=-Z (J)
390 NEXT J
400 NEXT I
410 FOR I=1 TO N
420 READ C (I)
430 NEXT I
440 READ V
450 LET B (N)=1
460 LET I=N
470 FOR J=1 TO N
480 LET A (I, J)=A (I, J)-V*K (J)
490 NEXT J
500 LET H (I)=0
510 LET H (2)=Q1/2
520 LET H (3)=Q1/2
530 LET H (4)=Q1
540 FOR I=1 TO N
550 READ X (I)
560 NEXT I
570 HOME, VTAB (5)
580 PRINT SPC (13), " FUNCTION PROGRAM", PRINT
590 PRINT SPC (12), " (1) DISPLAY HGR2", PRINT
600 PRINT SPC (12), " (2) DISPLAY DATA", PRINT
610 PRINT SPC (12), " (3) PRINT HGR2", PRINT
620 PRINT SPC (12), " (4) PRINT DATA", PRINT
630 PRINT SPC (12), " (5) EXIT", PRINT
650 PRINT TAB (10), INPUT " CHOOSE NUMBER (1--
  5)", I
660 ON I GOTO 670, 680, 2000, 3000, 4000
670 REM DISPLAY HGR2
680 HGR2, HCOLOR=7
690 GOSUB 5000
700 LET T=0
710 FOR E=1 TO Q3
720 FOR G=1 TO Q2

```



```

730 GOSUB 5070
740 FOR I=1 TO N
750 LET X(I)=X(I)+Q1*(K(I,1)+2*K(I,2)+2*
  K(I,3)+K(I,4))/6
760 NEXT I
770 LET T=T+Q1, LET T=INT(T*100+0.5)/100
780 GOSUB 5000
790 NEXT G
800 HPLLOT 3.1 TO 3.188 TO 277.188
810 FOR G=10 TO 188 STEP 10
820 HPLLOT 3.92 TO 277.92
830 HPLLOT 3.6 TO 5.6
840 NEXT G
850 FOR J=10 TO 276 STEP 10
860 HPLLOT J,188 TO J,185
870 NEXT J
880 HPLLOT 3.3 TO 5.9
890 HPLLOT 3.3 TO 0.9
900 HPLLOT 276.188 TO 270.185
910 HPLLOT 276.188 TO 270.191
920 HPLLOT 174*T,189-0.01*X
930 NEXT E, FOR I=1 TO N, X(I)=0, NEXT I
940 GET QS, POKE-16300,0, POKE-16303,0, GOTO 570
950 END
960 HOME
970 REM DATA
980 PRINT "T", TAB(20), "Y"
990 GOSUB 5000, T=0
1000 PRINT T, TAB(20), Y
1010 LET T=0
1020 FOR E=1 TO Q3
1030 FOR G=1 TO Q2
1040 GOSUB 5070
1050 FOR I=1 TO N
1060 X(I)=X(I)+Q1*(K(I,1)+2*K(I,2)+2*K
  (I,3)+K(I,4))/6
1070 NEXT I
1080 T=T+Q1, T=INT(T*100+0.5)/100
1090 GOSUB 5000
1100 NEXT G
1110 PRINT T, TAB(20), Y
1120 NEXT E, FOR I=1 TO N, X(I)=0, NEXT I
1130 GET QS, GOTO 570
1140 END
2000 REM PRINT HGR2
2010 HGR2, HCOLOR=7
2020 GOSUB 5000
2030 LET T=0
2040 FOR E=1 TO Q3
2050 FOR G=1 TO Q2
2060 GOSUB 5070
2070 FOR I=1 TO N
2080 LET X(I)=X(I)+Q1*(K(I,1)+2*K(I,2)+2*
  K(I,3)+K(I,4))/6
2090 NEXT I
2100 LET T=T+Q1, LET T=INT(T*100+0.5)/100
2110 GOSUB 5000
2120 NEXT G
2130 HPLLOT 3.1 TO 3.188 TO 277.188
2140 FOR G=10 TO 188 STEP 10
2150 HPLLOT 3.92 TO 277.92
2160 HPLLOT 3.6 TO 5.6
2170 NEXT G
2180 FOR J=10 TO 276 STEP 10
2190 HPLLOT J,188 TO J,185
2200 NEXT J
2210 HPLLOT 3.3 TO 5.9
2220 HPLLOT 3.3 TO 0.9
2230 HPLLOT 276.188 TO 270.185
2240 HPLLOT 276.188 TO 270.191
2250 HPLLOT 174*T,189-0.01*X
2260 NEXT E, FOR I=1 TO N, X(I)=0, NEXT I
2270 PR# 1
2280 POKE 1913,66
2290 PRINT CHR$(17)

2300 GET QS, POKE-16300,0, POKE-16303,0, GOTO 570
2310 END
3000 HOME
3010 REM DATA
3020 PR# 1
3030 PRINT "T", TAB(20), "Y"
3040 GOSUB 5000, T=0
3050 PRINT T, TAB(20), Y
3060 LET T=0
3070 FOR E=1 TO Q3
3080 FOR G=1 TO Q2
3090 GOSUB 5070
3100 FOR I=1 TO N
3110 X(I)=X(I)+Q1*(K(I,1)+2*K(I,2)+2*K
  (I,3)+K(I,4))/6
3120 NEXT I
3130 T=T+Q1, T=INT(T*100+0.05)/100
3140 GOSUB 5000
3300 NEXT G
3320 PRINT T, TAB(20), Y
3340 NEXT E, FOR I=1 TO N, X(I)=0, NEXT I
3360 PR# 0
3380 GET QS, GOTO 570
3400 END
4000 END
5000 REM OUTPUT COMPUTATION SUBROUTINE
5010 LET Y=0
5020 FOR I=1 TO N
5030 LET Y=C(I)*X(I)+Y
5040 LET Y=INT(Y*100+0.05)/100
5050 NEXT I
5060 RETURN
5070 REM INTERATE SUBROUTINE
5080 FOR I=1 TO 4
5090 FOR I=1 TO N
5100 LET P(I,J)=H(J)*K(I,J)-1
5110 LET Z(I)=P(I,J)+X(I)
5120 NEXT I
5130 FOR I=1 TO N
5140 LET D(I,J)=0
5150 FOR L=1 TO N
5160 LET D(I,J)=D(I,J)+A(I,L)*Z(L)
5170 NEXT L
5180 LET K(I,J)=D(I,J)+B(I)*U
5190 NEXT I
5200 NEXT J
5210 RETURN
5220 DATA 62.5,213.8,204.2,54
5230 DATA 1.562E+6,1.875E+6,0.0,0.002
5240 DATA 0.0,0.0

```

参 考 文 献

- [1] 绪方胜彦著, 现代控制工程, 科学出版社, 1976年
- [2] 熊光楞编著, 控制系统数字仿真, 清华大学出版社, 1982年
- [3] Kochenbures, Ralph J. Computer Simulation of Dynamic Systems. Englewood Cliffs, N. J., Prentice-Hall, 1972.
- [4] Jon M. Smith. Mathematical Modeling and Digital Simulation for Engineers and Scientists. New York, John Wiley and sons, INC., 1977.
- [5] 郑成义等编, APPLE II 微电脑使用手册, 广东高教出版社, 1986年

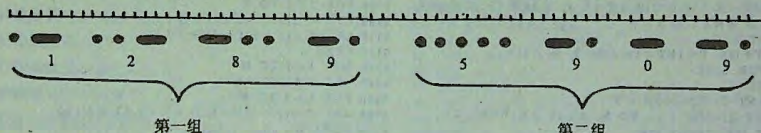
自动发报程序

辽宁农业管理干部学院计算机教研室 姚根龙 张金兰

无线电报务是一项极有意义的国防体育运动，抄报又是该运动中主要技术之一。通常是用自动发报机或人工发报进行抄报练习的。我们也可以使用计算机自动发报。

众所周知，计算机不但可以高速地运算、准确地绘图，而且还可以发出美妙动听的音乐。计算机自动发报就是利用它能够发声这一功能来模仿自动发报机发送电码的，报务员可以借此来提高抄报水平。

无线电报务通讯中是采用声音的长短来表示不同的字符的。短信号称为“滴”，长信号称为“达”，同时还规定“达”的信号长度是“滴”的三倍，每个“滴”、“达”信号间隔为一个“滴”的信号长；每个字符由一至五个“滴”、“达”信号组成，字符与字符之间的间隔正好为三个“滴”（即一个“达”）的信号长；每组字符由四至五个字符构成，每组字符间隔规定为五个“滴”的信号长。（如下图所示）



利用计算机自动发报不但能产生随机的电码，而且对程序稍加修改后还可以适应特殊电码的练习。如 Q、Y、X、P、J 和 Z 电码；E、I、S、H 和 5 电码；T、M、O 和 O 电码等这些易为混淆的电码可以集中在一起练习，这只要将选择语句后面的语句标号重新组合即可实现。

全程序大体上分为主程序和子程序两个部分，主程序的功能是不断产生随机电码，子程序功能就是发声，它是由机器语言编写。

10~60 语句为发声的子程序。

70~145 语句选择练习电码的类型，其中 1、2、3、4 分别表示长码、短码、字母及混合码四种练习电码。

200~245 语句是利用随机数产生长码和字母码的，300~345 语句是利用随机数产生短码的。

400~590 语句为不同数字和字母的入口。

600~660 语句则确定发声的长短，其中 600 语句为产生“滴”的入口，610 语句为产生“达”的入口。

该程序在 APPLE II 机上运行效果很好，对程序中的某些语句进行删改后还可以在 PC 机上运行，效果同样理

这些规定，我们可以通过控制发声的长短及间隔的比例来实现之，计算机自动发报程序就是根据这一原理，用 APPLE II BASIC 语言编写而成的。

在程序中我们利用了随机函数来产生不同的字符，其中包括十个阿拉伯数字和二十六个英文字母。根据练习的需要，我们又划分成四种类型的练习电码：即长码、短码、字母码及数字、字母的混合码。

运行本程序时，音调是可以改变的，输入音调参数范围可以为 1~255，但根据实际操作得知，当音调参数为 20~200 时发出的声音较为悦耳，速度也是可调的，速度输入参数范围是 1~85，发报的速度可低至 40 字符/分，高至 200 字符/分，一般已完全能满足练习的要求。

如果练习收报的人员较多，还可以将发出信号送至音频放大器放大，以提高抄报的效果。

数字、字母电码表

字母	电码	字母	电码	字母	电码
A	·—	J	·---	S	···
B	—···	K	—·—	T	—
C	—···	L	·—··	U	··—
D	—··	M	—	V	····
E	·	N	··	W	—··
F	····	O	---	X	—···
G	—·	P	····	Y	—···
H	····	Q	—---	Z	—···
I	··	R	·—		

数字	短码	长码	数字	短码	长码
1	·—	·—	6	····	····
2	··—	··—	7	—···	—···
3	····	····	8	—·	—·
4	····	····	9	—	—
5	····	····	0	—	—

想。

JLIST

```

10 DATA 173,48,192,136,208,5,20
   6,1,3,240,9,202,208,245,174,
   0,3,76,2,3,76,0,0
30 FOR N = 770 TO 792
40 READ A
50 POKE N,A
60 NEXT
70 HOME
85 VTAB 10: HTAB 10: INPUT "INPUT
  T FREQUENCY (1--255) " : IF
97 PRINT
90 HTAB 10: INPUT "INPUT SPEED (
  1--85) " : IF
92 HOME
95 VTAB 10: HTAB 10: PRINT "LONG
  CODE----- (1) "
97 PRINT
100 HTAB 10: PRINT "SHORT CODE--
  --- (2) "
102 PRINT
105 HTAB 10: PRINT "LETTER CODE-
  --- (3) "
107 PRINT
110 HTAB 10: PRINT "MIX CODE----
  --- (4) "
115 INPUT N:D = 0
120 ON N GOTO 130,135,140,145
130 N = 10: GOTO 200
135 N = 10: GOTO 300
140 N = 26:D = 10: GOTO 200
145 N = 36: GOTO 200
160 GOTO 110
200 FOR I = 1 TO 100
205 FOR J = 1 TO 5
210 ON RND (1) * N + 1 + D GOSUB
  490,470,495,500,400,425,430,
  575,540,450,515,435,565,585,
  420,560,545,405,415,475,530,
  590,460,550,455,555,525,570,
  410,465,510,505,485,520,480,
  580
215 FOR K = 1 TO 3 * V
220 NEXT K
225 NEXT J
230 FOR K = 1 TO 5 * V
235 NEXT K
240 NEXT I
245 GOTO 70
300 FOR I = 1 TO 100
305 FOR J = 1 TO 4
310 ON RND (2) * N + 1 GOSUB 51
  5,510,495,500,400,425,430,58
  5,550,465
315 FOR K = 1 TO 3 * V
320 NEXT K
325 NEXT J
330 FOR K = 1 TO 5 * V
335 NEXT K
340 NEXT I
345 GOTO 70
400 GOSUB 600
405 GOSUB 600
410 GOSUB 600
415 GOSUB 600
420 GOSUB 600: RETURN
425 GOSUB 610: GOTO 405
430 GOSUB 610
435 GOSUB 610: GOTO 410

```

```

450 GOSUB 610: GOSUB 610
455 GOSUB 610
460 GOSUB 610
465 GOSUB 610: RETURN
470 GOSUB 600
475 GOSUB 600: GOTO 455
480 GOSUB 610
485 GOSUB 600: GOTO 460
490 GOSUB 600: GOSUB 610: GOTO 4
  55
495 GOSUB 600: GOSUB 600: GOSUB
  600: GOTO 460
500 GOSUB 600
505 GOSUB 600
510 GOSUB 600
515 GOSUB 600: GOSUB 610: RETURN

520 GOSUB 610: GOTO 510
525 GOSUB 610
530 GOSUB 610: GOTO 515
540 GOSUB 610: GOSUB 610
545 GOSUB 610
550 GOSUB 610: GOSUB 600: RETURN

555 GOSUB 600: GOTO 545
560 GOSUB 600: GOSUB 600: GOTO 5
  50
565 GOSUB 610
570 GOSUB 600: GOTO 550
575 GOSUB 610
580 GOSUB 610
585 GOSUB 610: GOSUB 600: GOSUB
  600: RETURN
590 GOSUB 600: GOTO 585
600 T = V: GOTO 620
610 T = 3 * V
620 POKE 768,F
630 POKE 769,T
635 CALL 770
640 FOR K = 1 TO V
650 NEXT K
660 RETURN

```

启 事

为推广使用“快译通”英汉电脑字典，现大幅度降价酬宾，零售价：600元/台（包括邮费），款到即寄，批发面议。地址：广州市石牌五山路科技街29号，电话：510269，邮政编码：510630，开户银行：农业银行广州市新技术营业所263-501006。

天河音像科技有限公司

90.6.13

县级计算机发展状况的调查与思考

★ ★ ★ ★ ★ 江苏省射阳县委办公室 周林官 陆军 ★ ★ ★ ★ ★

随着我国现代科技的飞速发展,计算机已广泛应用于科研、生产、军事以及社会生活的各个领域,它对于加速我国的“四化”进程、缩短与先进国家的差距将发挥巨大的推动作用。但是,由于我国人口素质的偏低,科学发展水平的滞后,使得被誉为“电脑”的计算机在“人脑”的操纵下,却不能发挥它们应有的“聪明才智”。我们射阳县是个濒临黄海的经济发展县,在计算机的开发应用中就存在着这样的“心有余而力不足”状况,具体表现为“三难”。

一、开发应用难。购计算机的目的是为了应用,但我们在调查中却发现,射阳县着有的 27 台微机中,能发挥作用的只有 10 台,还不到总数的一半。还有 8 台处于“半应用”状态,剩下的 9 台则基本上是束之高阁,这就是说,每三台微机中就有一台完全是处于“沉睡”状态。而且即使得到应用,其用途也比较单一,据调查,上述 27 台计算机中,除 4 台属于专用外,有 5 台专门用于记帐,5 台用于每月一次的工资报表打印,4 台当着打字机在使用,“大马拉小车”的状况极为严重。究其原因主要有:①县里没有开发计算机应用软件的能力,在全县没有计算机站等类似普及计算机应用的组织。②计算机操作人员素质不高。全县共有计算机操作人员 34 人,其中大专以上文化程度的仅有 5 人,中专文化程度的 11 人,高中文化程度的 18 人,而其中学过计算机专业的只有 6 人,其余都是“半路出家”。因而,他们当中大部分人只能进行简单的、机械式的操作。比如,有个单位的一名具有高中水平的计算机操作员,他的任务是负责每月一次的工资报表打印,上机操作了半年多,只知道开机后,按一定顺序按几个键就完事,别的一概不知,有时按错了键,就得重新开机再来一遍。③外部环境差。计算机的开发应用是比较难的,计算机操作人员需要一个良好的外部环境才能出成果。如有些单位领导不够重视、经费不足等问题都严重影响操作人员开发应用的积极性。

二、管理难。计算机的管理是保证其发挥应有作用的必要措施,管理不好将直接影响到计算机的正常工作。对计算机的管理包括硬件管理、软件管理两方面。硬件管理主要是对计算机设备的管理,计算机设备对外部环境要求很高,如灰尘、温度、湿度及电源的稳定程度等对计算机的影响都很大。因此,要求有专门的计算机房。我们调查中发现,射阳县的 25 家用户只有 7 家用户具备基本符合

条件的计算机房。而其他的 18 个用户都将计算机放在普通的办公室或一般的房间里,不仅灰尘多,而且温度、湿度都不能保证在正常的范围内,大大影响了计算机的正常工作。如某单位将微机放在位置偏阴,而且室内条件较差的空房子里,致使冬天因室温太低而无法开机工作。另外,由于农村电网电压不稳定和经常性的突发停电给计算机的工作带来严重影响,而在这方面采取积极防护措施的用户却不多,25 家用户中只有 5 家买了 UPS 不间断电源,3 家是交流稳压电源,其余用户均无任何保护设施。更有甚者,在计算机的正常用途得不到发挥时,由于缺乏严格的管理制度,居然把计算机用来玩游戏、占卜算卦,有人嘲讽地说办公自动化变成了“办公游戏化”。在软件管理方面,由于大部分操作人员缺乏这方面的知识,对软件的管理也就比较乱。因乱拷贝,乱放磁盘丢失软件的现象屡见发生。

三、维修难。计算机是由多种电子元器件通过合理而又科学地设计组装起来的,虽然厂家千方百计提高产品质量,但由于使用环境等因素的影响,计算机系统时常会出现一些故障,由于受到地理位置、交通条件以及技术力量限制,有时为了一点小故障,往返数百里远送省城,甚至要千里迢迢送到北京等地去修理。而且修理费用也高得惊人;有时只是因为操作失误而造成的小故障,根本不需要打开机器修理,修理部门却索要“打开费”300~500 元不等;有些修理部门维修技术不高,无法给用户修好计算机,却照样厚颜无耻地乱收费。因此,部分用户反映:买得起计算机,花不起修理费,贴不起路费。造成这种状况的原因有:1 县级计算机的发展是近几年的事,维修服务以及技术培训等配套措施暂时还未跟上。②操作人员对计算机的硬件知识缺乏,有时只是一些小毛病自己也不能排除。

以上“三难”表明,随着县级计算机的发展,维修人员缺乏,操作人员素质偏差的问题也日趋突出。人才跟不上而导致的有用用不上的矛盾也就显露了出来,那么,怎样才能使县级计算机得以发挥应有的作用,不再出现花钱购“无用机”的现象呢。我们认为应从三个方面着手解决。

一、建立县级计算机应用协会。协会可吸收县内外各计算机用户为成员,既负责软件的开发,推广应用,又担

(转 15 页)

MS-DOS 彻底剖析 (七)

DOS 内部命令实现原理 (下)

郭嵩山 吴晓文 陈数学

七、批命令操作

DOS为保证批文件操作正常运行,提供了控制批处理的子命令,这些批命令包括PAUSE(暂停系统运行)、FOR(重复执行)、IF(条件执行)、SHIFT(移位替换参数)和GOTO(控制转向),有的资料将ECHO(显示开关)归入这类。但本文是从模块结构角度来分析,故将它划归开关操作类。下面分别介绍其功能实现的原理。

(一) PAUSE命令处理

命令格式: PAUSE [待显示提示信息]

其功能是暂停程序的执行,显示PAUSE后用户指定的提示信息及由系统给出的'strik a key when ready...'提示信息。如用户键入CTRL-Break,则停止程序的执行,如按其他键,则程序继续运行,不受影响。

PAUSE实现比较简单,主要是通过执行DOS的0C0IH号系统功能调用,清输入缓冲区内容并等待键盘输入,并根据用户不同输入作相应处理后,实现PAUSE功能程序在COMMAND暂驻段1332~1340H。

(二) FOR命令处理

命令格式:

FOR %%<变量名> IN [<文件名集>] DO 命令

该命令是重复执行命令,其功能是将%%后变量依次赋给文件名集(set)中每个元素,并执行DO后的命令,直到文件名集中的每一项都执行完。文件名集中允许使用多义文件,也允许使用路径名。

其实现原理是将磁盘传送缓冲区(PSP中0081H为首址)中读入的FOR命令进行拆解处理,也即将每一条FOR命令中文本名集内的文件名串送到常驻段979~9A0H的文件名集缓冲区中存放,并将之拆解;文件名间用回车码(0DH)分隔,整个文件名串用00结束。如果文件名串中文本名为多义文件(含通配符),则置通配符标志(常驻段09FEH字节置1)以作为后面处理的依据。同时将DO后的命令送常驻段9A1~9FAH的命令缓冲区存放。如果文件名集中含多义文件,则在常驻段09FF~0A1FH区域上逐个建立FCB。

FOR命令处理控制是通过设置几个标志单元来实现的,这包括前面提到的通配符标志(常驻段09FEH)、FOR嵌套标志(常驻段0977H)和批文件条件句柄(常驻段0973~0974H),该值如置为FFFFH,则退出FOR处理;如该

值设定为00FFH,表示重复执行;该值为具体地址指针时,则根据该指针停留在缓冲区中FOR语句中的位置,决定控制重复执行还是跳出循环。每次FOR命令处理执行后都返回调用点或COMMAND主程序入口。FOR命令处理在暂驻段147B~155BH。

(三) IF命令处理

命令格式: IF [NOT] <条件> <命令>

其中条件(CONDITION)为下列参数之一:

① ERRORLEVEL <Number>

返回码Number为十进制数。

② String 1=String 2(字符串相等)

③ EXIST [<路径名>] [<文件名>]

IF命令功能是有条件执行某条DOS命令。当条件成立时(为真值),执行本行的DOS命令,否则执行下一行DOS命令,该命令用来控制批命令的执行。

对于上述参数为真的条件是:

①如果前一个程序含有一个等于或大于Number的返回码,则ERRORLEVEL Number为真。

②当表达式string 1=string 2时为真。

③在指定目录上能找到所匹配的文件时,EXIST为真。

而当condition(条件)为假时,NOT condition为真。

在MS-DOS中,只有BACKUP和RESTORE这两条系统提供的外部命令设置了供测试用的ERRORLEVEL码,用户在自行设计中可自定程序运行结束的返回码以供测试之用。

IF命令实现原理是根据命令参数,查条件表,判定条件值真假,如为真值,则将条件后的命令送相应的命令行缓冲区,以待命令识别用,如为假值则返回。IF命令处理程序在暂驻段1563~1668H。

IF命令具体实现过程是(括号内为实现各段功能的程序段)

①对IF后命令参数语法分析,并查位于COMMAND暂驻段2A00~2A1BH的条件处理表,该表结构可参看89年第4期连载(二)(1563~15ACH)。

②如命令参数中有NOT,则将NOT标志单元(暂驻段2CC2H)取反,并继续查表(15AD~15B2H)。

③如条件为ERRORLEVEL,则将其后数字转换成二进制,并与前一程序的返回码比较,以确定条件值(真值AL=00H,假值AL=FFH,下同)(163B~1668H)。

④如条件为EXIST,则在暂驻段首址为2D68H区域建立FCB,并根据EXIST后的文件名打开该文件(15FD~16AH)。

⑤如查不到条件关键字,则执行比较字符串操作,并确定真假值(15B3~15FCH)。

⑥结合条件及NOT标志,判定条件值真假,如为假值,返回COMMAND主程序(160B~1618H)。

⑦把IF语句中命令写到首址为暂驻段的2B9EH的缓冲区中,重置PSP命令行缓冲区字符数及指针,转命令识别处理(161B~163AH)。

(四) SHIFT命令处理

SHIFT是移位替换参数命令,其命令格式是:

SHIFT

SHIFT命令功能是通过参数移位,改变形参与实参的形参表(执行SHIFT前)

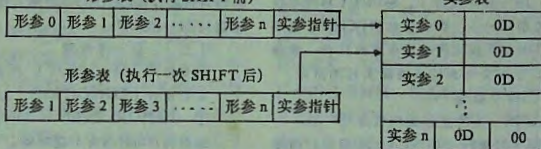


图1 SHIFT命令原理示意图

SHIFT命令实现过程如下:首先置批文件形参表指针及计数值,并把第2~10个参数在参数表内向前移1个参数位置,处理实参指针(跳过第一个实参),并在形参表最后两个字节上形成新的实参表指针(指向第2个实参)。在命令执行过程中,凡批文件为空(常驻段933H的标志值为FFFFH)或批文件中无对应的实参或实参表查完都作退出处理。

SHIFT命令处理程序在COMMAND暂驻段1669~16A8H。

(五) GOTO命令处理

命令格式:GOTO <标号>

其中标号是由' '后跟一个标号名所组成。

该命令功能是将控制转移到标号后的命令,以改变批命令执行的顺序。

其实现过程是:先打开批文件,然后根据标号,将文件读写指针移至批文件之首,再读批文件,当读到' '时,则比较' '后的标号是否与GOTO所指定的标号相等(大小写分别进行比较),如相等,则将批文件的读写指针移到标号后头一个回车码位置(也即标号后的一条命令之首),并关闭文件,返回调用程序,准备执行下一个DOS命令(外部或内部命令);如查不到' ',则关闭文件,显示'标号未找到'信息。上述操作中移动批文件读写指针是通过调用DOS 42H号功能来实现的。

GOTO命令处理程序在COMMAND暂驻段16A9~173FH,其中调用了1740~1762H的根据标号,移动读写指针的子程序。

八、I/O操作

DOS有关I/O操作的命令包括EXIT(出口命令)、CITY

对应关系,让一个批命令行可以使用超过10个的可替换参数。形参表结构如下图所示。

0D	参数0	参数1	参数2	...	参数n	实参表指针
----	-----	-----	-----	-----	-----	-------

即每个形参占2个Byte,形参表的最后两个字节为实参表指针。实参表的数据结构是一个不定长的数组,每个实参以0D作为结束标志,最后一个实参结束后以00作为整个实参表的结束。

SHIFT命令实现原理是每执行一次SHIFT命令,形参表向左移动一个位置,以%1取代%0(如图1所示),其实参表指针也随之相应改变(指向下一个实参)。

(改变控制台)和CLS(清屏)三条命令,其操作对象是控制台、屏幕和设备。

(一) EXIT命令处理

命令格式:EXIT

其功能是作为DOS命令的出口。EXIT主要任务是释放资源,恢复原来资源占有状态,进行程序结束的处理。例如,恢复进入DOS时程序段前缀表中父程序段址(PSP偏移0016~0017H),置21类中断向量入口,并调用DOS 49H号系统功能调用,释放程序运行时所占的内存空间,并执行DOS的4CH号功能调用,终止当前程序的运行并返回调用程序,其正常结束退出码为00H。该程序在内存暂驻段11DC~1209H。

(二) CITY命令处理

命令格式:CITY 设备名

其功能是定义新的主控制台。例如,使用CITY命令,可以将电传打字机、汉字显示终端、图形显示终端等设备定义为主控制台,而把原主控制台的设备(原屏幕、键盘等)改变为辅助控制台,也可以恢复原来作为主控制台的标准输入输出设备。

其实现原理是首先扫描命令参数,并根据参数中的设备名打开并读写文件,如果成功,将CITY命令参数所指示的设备改造为标准I/O设备,其改造过程包括关闭原来I/O及出错设备,置新的I/O和出错设备的文件号,以取代原来的设备,并关闭原来的设备文件。如果上述打开或读/写设备文件不成功,则关闭文件(如已打开的话)并显示设备无效信息。无论改造设备特性是否成功,最后都执行退出处理(即恢复PSP有关设备的内容),返回到程

序中止处理。CITY命令处理程序在内存暂驻段120A~1265H。

(三) CLS命令处理

命令格式: CLS

其功能是执行清屏,使光标回到第1行第1列。其实现原理是执行DOS的4400H号系统功能调用,取屏幕状态,如果不是屏幕而是磁盘文件,则执行直接控制台输出,否则取29H号中断向量,并执行BIO模块中10H类中断(显示器中断),完成诸如设置彩色号、彩色值,读当前显示状态,当前页上读及设置光标位置在屏幕左上角等操作。CLS命令处理程序在内存暂驻段1266~12B8H。

九、日期和时间操作

MS-DOS日期和时间操作包括DATE和TIME两条命令,这两条命令的实现都调用了DOS内核中相关的系统功能调用,这包括取/置日期(2AH/2BH号功能调用)和取/置时间(2CH/2DH号功能调用),它们的实现过程十分类似;如命令后不带参数,则执行取日期或时间的DOS调用(2AH或2CH),获取系统日期或时间并显示之;如命令后指定新的日期或时间,则首先检测所输入的格式是否正确,如正确,转换成二进制形式,然后执行2BH或2DH号功能调用,将新的日期或时间送入系统。

下面,据要介绍日期和时间操作的实现原理。

(一) DATE命令处理

命令格式: DATE [mm-dd-yy]

其中mm-dd-yy分别为月一日一年,年、月、日之间可用“/”或“-”分隔。

其功能是设置新的日期,日期输入形式为十进制,如在表中顺次为Sun, Mon, Tue, Wed, Thu, Fri和Sat。

执行2AH/2BH号功能调用所用到的数据结构如下:

CH	CL	DH	DL	AL
----	----	----	----	----

年 月 日 星期数

上述日期以二进制形式保存在相应的寄存器中。

在COMMAND暂驻段的2713~2727H共21个Byte中,存放一张星期表,它是一个等长的数组,一个星期的七天在表中顺次为Sun, Mon, Tue, Wed, Thu, Fri和Sat。

通过查表,找出星期数的文字表示,其计算公式为:

$2713H + \text{星期数} \times 3 = \text{对应的星期几文字表示}$

在执行DATE命令中,待显示日期在形成正确格式后送到首址为暂驻段2CD9H的显示缓冲区中存放,新输入的日期送至首址为2B9CH的命令行暂存区存放。

DATE命令处理程序在COMMAND暂驻段1848~18DFH,其主程序在184B~185CH。

(二) TIME命令

命令格式: TIME [hh:mm:ss[.xx]]

其中hh:mm:ss分别表示时、分、秒,xx表示百分比。

其功能是显示并设置时间。

执行2CH/2DH号功能调用(取/置时间)所用到的数据结构是:

CH	CL	DH	DL
----	----	----	----

小时 分钟 秒 百分比

上述时间以二进制形式保存在相应的寄存器中。

TIME命令处理程序在COMMAND暂驻段18E1~194FH。

其主程序在18E1~18F2H。

十、VER和REM命令处理

这两条命令比较简单。

VER命令格式: VER

其功能是显示版本号,其处理程序在COMMAND暂驻段0E6A~0EB7H。其实现原理是直接调用DOS 30H号系统功能调用,取出版本号(AH=版本号,AL=子版本号),并转换成十进制形式,版本号与子版本号之间用“.”分隔,取出后送到首址为2CD9H的显示缓冲区中等待输出。

REM为注释命令,其功能是批处理时显示REM所在注释行,然后转回COMMAND主程序入口(暂驻段104H),实际上并无作其他实质性处理。

至此,我们将DOS的内部命令实现原理全部介绍完毕。

(接12页)

负起计算机的维修服务工作,同时还可以举办一些培训班,普及计算机应用基础知识。该协会可以不设立专门办公机构,由使用计算机较早,经验较丰富,计算机人才较多的用户牵头主办,这样既可以发挥本地优势,又不增加国家负担。

二、提高计算机操作人员的素质,计算机应用得好坏,作用发挥得如何,取决于操作人员业务素质的优劣。从目前情况看,操作人员素质参差不齐,且基础普遍较差。因此,要求基础较好的操作人员应该在实践中不断探索,及时吸收新知识,争取早出成果;基础差的操作人员应该积极进取,经常参加各种形式的培训班学习或深造进修,以提高自己的素质;实在不能胜任的操作人员,应尽快改行,让能者上岗。这样操作人员的素质提高了,计算机的效用也就会相应地得到提高和扩展。

三、要从实际出发,先人后机,按需置机。今后,每个用户在购机之前要先将具备操作能力的计算机操作人员选定好,然后根据本单位的实际需要确定机型,再按照价廉物美的原则购置合适的计算机。这样,就可有效地避免“置则无可用”,“临时抱佛脚”等不良状况。

SAS 语言介绍

中国医科大学计算中心 董大钧

在信息时代的今天，人们在工作实践中会获取到大量的信息。如何存贮、整理和分析处理它们是一件很重要的工作。近几年来由于 dBASE 软件的普及，人们已普遍利用 dBASE 去存贮、整理和管理信息，甚至去分析它们。但利用 dBASE 或用高级语言进行信息的定量分析都要编制较大的程序，这对一般用户来说困难较大。即使是打印一个简单的表格也不是一件容易的事。

然而大量的数据分析工作都是基于基本的统计原理进行的。国内外学者多年来为进行数据定量分析编制了许多统计软件包。SAS 软件包则是诸多统计软件包中的佼佼者。它使用简单，几乎能用极简单的命令去作你想作的一切数据整理和分析工作。

SAS (Statistical Analysis System) 是一个用来分析数据和编写报告的系统，是美国 SAS 研究所于 1966 年开始开发的。使用 SAS 软件可以存贮、恢复、修改数据，对这些数据可以进行各种统计，并建立每次处理的报告，也可给出图形或进行预测。

与另两个最流行的统计软件包 BMDP 和 SPSS 相比，SAS 的岭回归分析、概率单位分析是 BMDP 所没有的，它在时序分析方面，更优于 BMDP 和 SPSS。它具有很强的图形显示功能，甚至三维显示和地图输出，而 BMDP 和 SPSS 都只有用字符显示的直方图和散点图。

SAS 系统具有积木式的结构，在 SAS/BASE 基础软件的基础上，可以任意增加象 SAS/GRAPH, SAS/STAT, SAS/IML 等功能块，组成完善的 SAS 系统。它不仅可运行在象 IBM PC/XT/AT 等微机上，而且可以运行在许多大中型计算机上。

在 IBM PC 上，SAS/BASE 软件约占 7M 硬盘空间，而 SAS/STAT (高级统计) 软件约占 5M 硬盘空间。

与其它几个世界上流行的统计软件包相比，SAS 系统在数据预处理方面具有独特之处。因此，除了广泛应用在统计上之外，SAS 在商业界也得到越来越多的应用。

世界上有一 SAS 协会，每年都要开会对该软件的使用进行交流和研究。而 SAS 公司每年都在改进 SAS 系统，目前 SAS 系统已经被完善成一种高级编程语言。

SAS 有 100 多种函数，有各种算术和逻辑操作符，

可以使用赋值语句，条件语句，数组和循环语句等对变量进行赋值、修改、删除和输出等操作。

下面简单介绍一下 SAS 软件包的使用方法。

在 SAS 系统中，编制的程序概括起来可分为两种：一种是建立和整理数据集，这样的程序段称为数据步；另一种是对存在于 SAS 数据集集中的数据利用现有的过程进行处理和分析，这种程序称为过程步。

一、窗口

SAS 软件使用了窗口技术，最常用的是程序编辑、日志和输出三个窗口。每一窗口都有一命令行。一般的操作是在程序编辑窗口中的标号行上，书写语句以组成程序。所有的 SAS 语句都以分号结尾。编好的程序通过在命令行上打入 SUBMIT 命令或按功能键发送给 SAS 系统。

运行过程中的各种信息都在日志窗口中显示并记录，而输出结果则由输出窗口给出。各窗口中的信息都可存入磁盘文件中。

你可很方便地改换当前窗口。

二、建立数据集

运行 SAS 处理数据之前，要首先利用 DATA 步建立一个 SAS 数据集。只有在数据集集中的数据才能被处理。建立 SAS 数据集时，要用 INPUT 语句说明各变量的名称、类型和变量长度等特性。

在 DATA 步中，输入数据建 SAS 数据集有三种方法：

1. 利用 CARDS 语句直接输入数据。如建立有 DEP、MACHINE 和 SALES 三个变量的各商店电视销售数据集 TVSALES。使用下列语句：

```
DATA TVSALES;
INPUT DEP $ MACHINE $ SALES;
CARDS;
D1 BL_TV 11105
D2 COL_TV 21157
D1 COL_TV 22134
D2 BL_TV 33125
```

RUN;

其中变量 DEP 和 MACHINE 是字符型数据，故在其名后加一美元号 (\$)。各项间用空格间隔而不使用逗号。

2. 利用 INFILE 语句, 从外部系统数据格式 (SDF) 文件中, 读取部分数据建立数据集。如:

```
DATA PAYSET;
  INFILE 'B.PAY.DAT';
  INPUT DEP $ 1-3 PAY 16-19;
RUN;
```

当该程序运行时, 就从 B 盘的 PAY.DAT 文件中, 将各相应数据位的值读入 SAS 数据集 PAYSET 中。

3. 利用 DBF 或 DIF 过程, 将由 DABSE 或 LOTUS1—2—3 建立起的数据文件中的数据传至 SAS 数据集。

如将存于 SAS 子目录下的 DBASEIII 库文件 PAY1.DBF 中的数据转入 SAS 数据集 PAYSET 中, 用如下语句:

```
PROC DBF DB3=PAY1 OUT=PAYSET;
RUN;
```

三、整理数据

在处理数据之前, 可能要在数据集中增加某些新变量, 如将原定量的变量进行量化处理, 也可能想修改或删除某些不合用的数据。或者只想分析原数据的一部分。你可在 DATA 步中对数据进行整理, 或者产生一个符合条件的子集。如对年龄 AGE 按 5 岁一分组, 得年龄组值 INTAGE, 使用如下语句:

```
INTAGE=INT(AGE/5);
```

VARIABLE	N	MRAN	STANDARD DEVIATION	MINIMUM VALUE	MAXIMUM VALUE
-----DEP=D1-----					
SALES	2	16619.50000	7798.680690	11105.00000	22134.0000
-----DEP=D2-----					
SALES	2	27141.00000	8462.653957	21157.00000	33125.0000

利用 TABULATE 过程, 可以按你在表格表达式中给定的格式, 打印出各种统计表格。该过程将统计与制表融为一体, 表格中允许有 18 种统计量。

如: 打印出各商店销售统计表格。

```
PROC TABULATE;
  CLASS DEP;
  VAR SALES;
  TABLE DEP ALL, SALES * (SUM PCTSUM);
RUN;
```

用 CLASS 语句指明分类变量, 用 VAR 语句指明统计变量, 运行后得到各商店销售情况表格 (SUM 为各单位销售和, PCTSUM 为各单位销售和百分比)。

四、调用 SAS 过程

为了方便各种数据处理和分析, SAS 已将各处理程序编制成过程。一旦你建立了一个 SAS 数据集, 就可以用 PROC 语句调用相应的 SAS 过程来分析和处理数据集

中的数据。例如: 用 SORT 过程对数据集进行排序; 用 MEANS 过程计算平均值及其它描述统计量, 并将其打印出来; FREQ 过程在频数表或列联表中给出变量值的计数统计; 利用 CORR 过程可求出各数值变量间的相关系数。在各种数据处理过程中, 也可以根据某一变量的取值而按组进行。如:

```
PROC SORT OUT=SALESORT;
  BY DEP;
RUN;
```

运行时, 就对当前数据集按商店进行了排序, 产生了新的数据集 SALESORT。

```
又如:
PROC MEANS;
  VAR SALES;
  BY DEP;
RUN;
```

运行后得到了各商店的销售值, 销售值的平均值, 最大最小值, 标准差等统计量, 如下表所示:

	SALES	
	SUM	PCTSUM
DEP		
D1	33239.00	37.98
D2	54282.00	62.02
ALL	87521.00	100.00

对数据作各种相关、回归分析及画各种统计图, 也都象上面的程序一样, 只要调用一个相应的 SAS 过程, 并给出必要的信息就可以了。对于系统中尚没有的处理过程, 用户可利用 SAS 语言自己编写程序进行处理。

五、SAS 过程菜单系统

该系统是为使每一过程不论对初学者还是经验丰富的程序员都容易使用而设计的。这个系统是由填空表格组

修改 EXE 文件中说明部分的新方法

哈尔滨冶金高等专科学校 王延亮

EXE 文件是经编译后产生的执行文件，一般很难用 type, wordstar 等显示、修改。过去曾有文章介绍过用 DEBUG 或汇编语言进行修改的方法，但一般微机用户难以掌握。

为此，笔者设计了一种简单易行的新方法，用 BASIC、Wordstar 等一般软件工具即可实现对 EXE 说明部分的修改。以下分几部分介绍这种方法，文中 Wordstar 用 WS 表示。

一、WS 等编辑工具不能编辑 EXE 的原因

WS 为什么不能编辑 EXE 文件？笔者把 EXE 文件当作随机文件，用 BASIC 程序将其读入，赋予 X\$ (K) 字符组，再用

```
PRINT ASC (X$ (K))
```

显示于屏幕，发现 X\$ (K) 的 ASCII 码不仅包含 WS 能编辑的部分，如 33~126 (西文字母、数字、符号)、161~254 (汉字)，而且还包含小于 33 的部分和 128~160 部分。

后两部分是 WS 难以处理的，如 12 号 ASCII 符显示的结果是清屏；9 号字符是声响，更为特殊的是 26 号 ASCII 符，WS 把它作为文件的终止符，当 WS 读到该字符时，就认为文件结束，不再承认以下内容。

这就是 WS 不能编辑、修改 EXE 文件的主要原因。

二、EXE 文件中说明字符串的提取

从上文的分析可知，只要从 EXE 文件中去除 WS 不能处理的 ASCII 字符，即可得到包含说明部分的可编辑字符文件。笔者为此设计了 GZWE1.BAS 程序。程序的使用过程如下：

1. 将 EXE 文件拷贝到工作盘上
2. 进入 BASIC，执行 GZWE1。
3. 输入欲修改的文件名。若有扩展名，必须同时输入。

4. 运行完，退出 BASIC。

这时，提取出的部分就形成了名为 WJSMZF\$.DAT 的文件。

100 句的 Kill 命令在首次使用程序时可删去。

265 句对 Y\$ (65) 赋 13 号 ASCII 符，对 Y\$ (66) 赋 10 号 ASCII 符是为了下步编辑 WJSMZF\$.DAT 时，每显示 64 字符后自动回车、换行。

270 句是在文件结尾处写上终止标志。

三、对 WJSMZF\$.DAT 的编辑修改

用 WS 或 Edlin 可方便地对 WJSMZF\$.DAT 进行各种操作。但要注意不能插入任何字符，不能改动非说明部分，新改写的字符串不能长于旧字符串。

四、WJSMZF\$.DAT 与 EXE 的合并

WJSMZF\$.DAT 与 EXE 的合并由 GZWE2.BAS 完成，操作过程如下：

1. 进入 BASIC，运行 GZWE2。

2. 输入修改后的文件名。该文件必须存在，且与原文件等长。

3. 退出 BASIC。

至此，全部修改结束，打入 EXE 文件名[回车]，即可看到文件运行时显示出新的说明，提示字符，而原有的功能丝毫未变。上述修改过程同样适用于 COM 文件和 OVL 文件。

成，它指导你在 SAS 系统中通过这个过程，按照表格上的句法，新的用户能够学习如何去使用这个过程。对于临时用户可不必记住他们需要用的各种可选项，即使是程序员也只需熟悉他们需要使用的 SAS 过程的概况就可以了。

在 SAS 系统中，你可使用 HELP 键迅速地得到联机帮助。你可以将中间结果和最终结果及运行信息存于磁盘文件中，以便今后调用或打印。

你可在命令行上打入 X 而暂时进入操作系统，进行

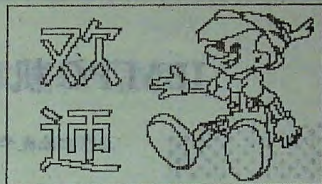
各种系统处理后，再返回 SAS。

在操作系统下，可以自动执行由 SAS 语句编写的命令文件。如能将 SAS 与 DBASE 结合使用会极大的方便信息处理。

推广和普及 SAS 软件包必将极大地提高我国各种领域中，尤其是医学领域中的信息处理水平。为使从事数据处理工作的同志都能了解 SAS 软件包，我们已将 SAS / BASE 软件的本三本说明书《SAS 导引》、《SAS 语言指导》、《SAS 过程指导》译成中文，并陆续出版。

高分辨率图形的放大打印

北京师范大学 戚戚成



我们用 APPLE-II 微机可以画出很漂亮的高分辨率图形，若把它们打印出来，也会具有欣赏价值。但是，通常打印出来的图象幅面较小，约为 80 列打印纸的 1/8 页。今设计下列程序，它可以把图象横向放大到 2 倍，同时将纵向放大到 4 倍，这样，我们就能打印出幅面为整页 80 列纸大小的图，可张贴在墙上。有兴趣的读者可以试试。

程序运行前，请你把拟打印的图象先在微机上画出生并存入磁盘，然后开启打印机，再运行本程序。运行后，屏幕上将出现“FILENAME:”字样，此时，请键入存盘图

象的文件名，并按一次“RETURN”键。这样，屏幕上将显示调入之原图，继而显示原图上半部的放大图，并开始打印。当打完这一部分，屏幕将显示原图下半部的放大图，并继续打印，直至打完完整幅图形后结束。

本程序在 APPLE-II 微机和 CPA-80 打印机支持下运行通过。程序清单如下，打印的图形见图。

笔者用本文介绍的方法汉化了 BASIC.COM 等文件，效果很好。存在的问题是，由于新的 EXE 文件是当作随机文件用 BASIC 程序形成的，其长度是 128 的整数倍，故新的 EXE 文件有时会超过原文件的长度。尽管这对文件的功能没有影响，但却多占了存储空间。对此有待于进一步改进。

附源程序两则

GZWE1.BAS

```
100 CLEAR:CLS:DIM X$(64),Y$(66):KILL"wjsmzf$.dat"
105 OPEN "R",#1,"wjsmzf$.dat",66
115 LOCATE 5,10:INPUT"请输入欲修改的文件名",N$
117 OPEN "R",#3,N$,64:ZJ=LOF(3):JS=INT(ZJ/64+9999)
:EZ=JS-64-ZJ+1
120 FIELD=
130 FOR K=1 TO 66:FIELD # 1,(K-1) AS AA$,1 AS Y$(K):NEXT K
140 FOR K=1 TO 64:FIELD # 3,(K-1) AS AA$,1 AS X$(K):NEXT K
250 '=====
257 FOR I=1 TO JS:GET #3,I
260 FOR K=1 TO 64:C=ASC(X$(K))
261 IF C>32 AND C<127 OR C>160 AND C<255 THEN LSET
Y$(K)=X$(K) ELSE LSET Y$(K)=" "
265 NEXT K:LSET Y$(65)=CHR$(13):LSET Y$(66)=CHR$(10)
270 IF I=JS THEN LSET Y$(EZ)=CHR$(26)
275 PUT #1,I
280 NEXT I:CLOSE:SYSTEM
```

GZWE2.BAS

```
100 CLEAR:CLS:DIM X$(64),Y$(66)
105 OPEN "R",#1,"wjsmzf$.dat",66
115 LOCATE 5,10:INPUT"请输入修改后的文件名",N$
117 OPEN "R",#3,N$,64:ZJ=LOF(3):JS=INT(ZJ/64+9999)
120 FIELD=
130 FOR K=1 TO 66:FIELD # 1,(K-1) AS AA$,1 AS Y$(K):NEXT K
140 FOR K=1 TO 64:FIELD # 3,(K-1) AS AA$,1 AS X$(K):NEXT K
250 '=====
257 FOR I=1 TO JS:GET #1,I:GET #3,I
260 FOR K=1 TO 64:C=ASC(X$(K))
261 IF C>32 AND C<127 OR C>160 AND C<255 THEN LSET
X$(K)=Y$(K)
265 NEXT K
275 PUT #3,I
280 NEXT I:CLOSE:KILL"wjsmzf$.dat":SYSTEM
```

LIST

```
10 TEXT: HOME
20 VIAB 10: HTAB 10
30 INPUT "FILENAME:":F$
40 PRINT CHR$(4);"BLOAD "F$";,AF4000"
50 FOR I = 768 TO 897: READ A: POKE I,A: NEXT I
60 DATA 169,32,133,230,160,0,162,0,138,133,254,
32,17,244,166,254,165,38,157,0
70 DATA 112,165,39,157,0,113,24,105,32,157,0,11
4,232,224,192,208,227,24,169,0
80 DATA 133,11,169,0,133,10,170,189,0,112,133,6
189,0,114,133,7,138,10,170
90 DATA 189,0,112,133,8,189,0,113,133,9,232,189
0,132,133,16,189,0,113,133
100 DATA 17,160,0,177,6,145,8,145,16,200,192,40
208,245,230,10,165,10,201,96
110 DATA 208,200,230,11,165,11,201,2,240,19,169
96,141,46,3,141,53,5,169,114
120 DATA 141,67,3,141,78,3,76,42,3,96
130 POKE 49232,0: POKE 49239,0: POKE 49234,0: POKE
49237,0
140 FOR I = 0 TO 3000: NEXT I
150 CALL 768
160 POKE 49236,0
170 PRI 1
180 POKE 1913,65
190 PRINT CHR$(17)
200 PRINT CHR$(127); CHR$(106); CHR$(66);
210 POKE 49237,0
220 POKE 1913,66
230 PRINT CHR$(17)
240 PRINT CHR$(24)
250 " "
260 TEXT: HOME: END
```

(K):NEXT K

```
140 FOR K=1 TO 64:FIELD # 3,(K-1) AS AA$,1 AS X$(K):NEXT K
```

```
250 '=====
```

```
257 FOR I=1 TO JS:GET #1,I:GET #3,I
```

```
260 FOR K=1 TO 64:C=ASC(X$(K))
```

```
261 IF C>32 AND C<127 OR C>160 AND C<255 THEN LSET
X$(K)=Y$(K)
```

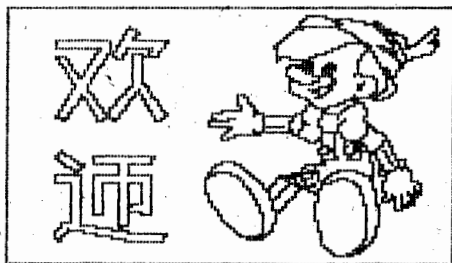
```
265 NEXT K
```

```
275 PUT #3,I
```

```
280 NEXT I:CLOSE:KILL"wjsmzf$.dat":SYSTEM
```

高分辨率图形的放大打印

北京师范大学 臧威成



我们用 APPLE-II 微机可以画出很漂亮的高分辨率图形，若把它们打印出来，也会具有欣赏价值。但是，通常打印出来的图象幅面较小，约为 80 列打印纸的 1/8 页。今设计下列程序，它可以把图象横向放大到 2 倍，同时纵向放大到 4 倍。这样，我们就能打印出幅面为整页 80 列纸大小的图，可张贴在墙上。有兴趣的读者可以试试。

程序运行前，请你把拟打印的图象先在微机上画出并存入磁盘，然后开启打印机，再运行本程序。运行后，屏幕上将出现“FILENAME:”字样，此时，请键入存盘图

象的文件名，并按一次“RETURN”键。这样，屏幕上将显示调入之原图，继而显示原图上半部的放大图，并开始打印。当打完这一部分，屏幕将显示原图下半部的放大图，并继续打印，直至打完整幅图形后结束。

本程序在 APPLE-II 微机和 CPA-80 打印机支持下运行通过。程序清单如下，打印的图形见题图

笔者用本文介绍的方法汉化了 BASIC.COM 等文件，效果很好。存在的问题是，由于新的 EXE 文件是当作随机文件用 BASIC 程序形成的，其长度是 128 的整倍数，故新的 EXE 文件有时会超过原文件的长度。尽管这对文件的功能没有影响，但却多占了贮存空间。对此有待于进一步改进。

附源程序两则

GZWE1.BAS

```
100 CLEAR:CLS:DIM X$(64),Y$(66):KILL"wjsmzf$.dat"
105 OPEN "R",#1,"wjsmzf$.dat",66
115 LOCATE 5,10:INPUT:"请输入欲修改的文件名",N$
117 OPEN "R",#3,N$,64:ZJ=LOF(3):JS=INT(ZJ/64+.9999)
:EZ=JS*64-ZJ+1
120 '-----field-----
130 FOR K=1 TO 66:FIELD #1,(K-1) AS AA$,1 AS Y$
(K):NEXT K
140 FOR K=1 TO 64:FIELD #3,(K-1) AS AA$,1 AS X$
(K):NEXT K
250 '=====
=====
257 FOR I=1 TO JS:GET #3,I
260 FOR K=1 TO 64:C=ASC(X$(K))
261 IF C>32 AND C<127 OR C>160 AND C<255 THEN LSET
Y$(K)=X$(K) ELSE LSET Y$(K)=" "
265 NEXT K:LSET Y$(65)=CHR$(13):LSET Y$(66)=CHR$(10)
270 IF I=JS THEN LSET Y$(EZ)=CHR$(26)
275 PUT #1,I
280 NEXT I:CLOSE:SYSTEM
```

GZWE2.BAS

```
100 CLEAR:CLS:DIM X$(64),Y$(66)
105 OPEN "R",#1,"wjsmzf$.dat",66
115 LOCATE 5,10:INPUT:"请输入修改后的文件名",N$
117 OPEN "R",#3,N$,64:ZJ=LOF(3):JS=INT(ZJ/64+.9999)
120 '-----field-----
130 FOR K=1 TO 66:FIELD #1,(K-1) AS AA$,1 AS Y$
```

FILE

```
10 TEXT: HOME
20 VTAB 10: HTAB 10
30 INPUT "FILENAME: ";F$
40 PRINT CHR$(4);"BLOAD ";F$;"A$4000"
50 FOR I=768 TO 897: READ A: POKE I,A: NEXT I
60 DATA 169,32,133,230,160,0,162,0,138,133,254,
32,17,244,166,254,165,38,157,0
70 DATA 112,165,39,157,0,113,24,105,32,157,0,11
4,232,224,192,208,227,24,169,0
80 DATA 133,11,169,0,133,10,170,189,0,112,133,6
,189,0,114,133,7,138,10,170
90 DATA 189,0,112,133,8,189,0,113,133,9,232,189
,0,112,133,16,189,0,113,133
100 DATA 17,160,0,177,6,145,8,145,16,200,192,40
,208,245,230,10,165,10,201,96
110 DATA 208,200,230,11,165,11,201,2,246,19,169
,96,141,48,3,141,53,3,169,114
120 DATA 141,67,3,141,78,3,76,42,3,96
130 POKE 49232,0: POKE 49239,0: POKE 49234,0: POKE
49237,0
140 FOR I=0 TO 3000: NEXT I
150 CALL 768
160 POKE 49236,0
170 PR#1
180 POKE 1913,65
190 PRINT CHR$(17)
200 PRINT CHR$(27); CHR$(106); CHR$(166);
210 POKE 49237,0
220 POKE 1913,66
230 PRINT CHR$(17)
240 PRINT CHR$(24)
250 '-----
260 TEXT: HOME: END
```

(K):NEXT K

```
140 FOR K=1 TO 64:FIELD #3,(K-1) AS AA$,1 AS X$
(K):NEXT K
```

```
250 '=====
=====
```

```
257 FOR I=1 TO JS:GET #1,I:GET #3,I
260 FOR K=1 TO 64:C=ASC(X$(K))
262 IF C>32 AND C<127 OR C>160 AND C<255 THEN LSET
X$(K)=Y$(K)
265 NEXT K
275 PUT #3,I
280 NEXT I:CLOSE:KILL"wjsmzf$.dat":SYSTEM
```


IBM PC 机 BIOS 数据区的利用

北京工业大学计算机科学系 86072 班王宇栋

大家知道,当 IBM PC 系列机在启动以后,内存中从 40.0H 开始有一段区域,存放着 BIOS 数据,我们通常称这一区域为 BIOS 数据区。在该区中,存放着有关硬件设备,如键盘、软盘、硬盘、显示器、打印机等的各种信息。BIOS 的中断调用和 DOS 操作系统都利用了这些数据。如果我们也能对这些数据的具体意义有所了解,并加以利用,则将会给我们开发一些应用程序带来极大方便。下面,由于篇幅所限,我们只选择其中的几个单元加以介绍,希望能引起大家对数据区的进一步重视。

1. 硬件设备配置字

该字在内存中的地址为 0040:0010H,各位的意义分别如下:

位	意义
0	装有磁盘驱动器时为 1;
1	协处理器安装时为 1;
2—3	系统板 RAM 容量: 00=16K 01=32K 11=64K;
4—5	显示器方式: 00=40 列彩色 10=80 列彩色 11=80 列单色
6—7	磁盘驱动器数目: 00=1 个驱动器 01=2 个驱动器 10=3 个驱动器 11=4 个驱动器;
8	DMA 芯片安装标志;
9—11	RS-232 串口数目;
12	游戏卡安装标志;
13	串行打印机安装标志;
14—15	打印机数目。

2. 内存容量字

该字在内存中的地址为 0040:0013H,存放着内存总容量,以 K 字节为单位。

3. 扩充板内存容量字

该字在内存中的地址为 0040:0015H,存放着扩充板内存总容量,以 K 字节为单位。

4. 当前 CRT 方式字节

该字节在内存中的地址为 0040:0049H,其中不同的值代表的意义分别如下:

00—40 * 25 黑白	04—320 * 200 彩色
01—40 * 25 彩色	05—320 * 200 黑白
02—80 * 25 黑白	06—640 * 200 黑白
03—80 * 25 彩色	07—80 * 25 单色板

通过以上简要的介绍,大家已经可以看出,这些数据能够为我们的程序设计提供重要的信息,便于我们在程序中根据不同的硬件配置作出相应的选择,从而提高程序的灵活性和应变能力。

最后,我们用 BASIC 语言编写了一段程序,其作用是显示出系统的硬件配置情况和当前屏幕所处的状态以及内存和扩充板的容量,大家可以从中间看出这些单元在程序中是如何被利用的。

程序清单如下:

```

10 DEFINT W
20 DEF SEG = &H40
30 W = PEEK(&H10)+256 * PEEK(&H11)
40 WT = W / 2 ^ 14
50 PRINT "The number of printers is";WT
60 IF (W AND &H2000) < > 0 THEN PRINT "Serial printer is present."
70 IF (W AND &H1000) < > 0 THEN PRINT "Game adaptor is present."
80 WT = W / 2 ^ 9:WT = WT AND &H7
90 PRINT "The number of RS-232 serial ports is";WT
100 IF (W AND 1) < > 0 THEN WT = W / 2 ^ 6:WT = WT AND 3:PRINT "The number of diskette drives is";WT+1
105 WT = W / 2:WT = WT AND 1:IF WT = 1 THEN PRINT "Co-processor is installed."
110 W = PEEK(&H49)
120 PRINT "Current display mode is";
130 IF W = 0 THEN PRINT "40 * 25 B/W"
140 IF W = 1 THEN PRINT "40 * 25 Color"
150 IF W = 2 THEN PRINT "80 * 25 B/W"
160 IF W = 3 THEN PRINT "80 * 25 Color"
170 IF W = 4 THEN PRINT "320 * 200 Color"
180 IF W = 5 THEN PRINT "320 * 200 B/W"
190 IF W = 6 THEN PRINT "640 * 200 B/W"
200 IF W = 7 THEN PRINT "80 * 25 MONO"
210 W = PEEK(&H13)+PEEK(&H14) * 256
220 PRINT "Total memory is";W;"K bytes"
230 W = PEEK(&H15)+PEEK(&H16) * 256
240 PRINT "Memory on extended board is";W;"K bytes"
250 END

```

APPLE EDIT 文本编辑程序

哈尔滨科技大学 105 信箱 傅增明

文本编辑程序用于输入文本, 修改订正, 打印输出和把文本存入文件中, 是字处理系统的重要组成部分。作者用 BASIC 语言编写了一个文本行编辑程序 APPLE EDIT, 并在苹果机上运行的 MCS-51 单片机交叉汇编 SUPER/8051 中得到应用。SUPER/8051 是目前苹果机上速度最快、功能最强、容量最大并可与许多常用的开发机(仿真器)通讯的交叉汇编, 其编辑程序就是以 APPLE EDIT 为基础, 再加上一个条汇编程序组成, 在编辑时, 利用键盘输入的空闲时间, 进行条汇编, 从而大大地提高了它的整体速度。

一、APPLE EDIT 文本编辑程序操作命令

[CR] 文本输入命令。键入该命令后, 自动显示行号即可输入文本。如果键入 [CR] 命令前, 机内已有文本若干行, 则 [CR] 命令不损坏原有文本, 行号从原有文本行数加 1 开始。APPLE EDIT 的每一个行最多可输入 255 个字符, [CR] 命令由 CTRL-Z 退出, 返回 COMMAND 状态。

n[CR] 行插入命令。该命令在第 n-1 行和第 n 行之间插入一行, 插入行的行号为 n。第 n 行以后的行号全部加 1。插入一行后自动返回待命状态。如果在输入回车符之前, 键入 CTRL-Z, 则刚输入的一行作废, 并把原来的第 n 行复制一行。

L[CR] 屏幕列表显示命令。列表从第一行开始, 列表过程中可由 CTRL-S 暂停, 可由 CTRL-Z 中断返回。为了能够显示完整的行, 列表过程中, 在每显示一个完整的行后, 才采样 CTRL-Z 信号。

nL[CR] 显示第 n 行, 并自动返回待命状态。

nL-[CR] 从第 n 行开始列表, 可由 CTRL-S 暂停, 可由 CTRL-Z 中断返回。

nE[CR] 修改命令。当需要修改第 n 行时, 键入 nE[CR] 命令, APPLE EDIT 显示该行的原有内容, 并请求输入新的内容, 回车后自动返回 COMMAND, 如果在回车前输入 CTRL-Z, 则新输入的行作废, 修改之前的行仍有效。

nD[CR] 删除命令。删除文本的第 n 行, 第 n 行以后的行号全部减 1。

nD-[CR] 删除掉第 n 行到最后一行之间的所有行。

$n_1 - n_2 D = [CR]$ 删除掉第 n_1 行到第 n_2 行之间的

所有行(包括第 n_1 行和第 n_2 行)。

P[CR] 打印输出命令。在键入该命令前, 打印应准备就绪, 打印时可选择是否打印行号。

S[CR] 文本存盘命令。键入该命令后需输入文件名。

R[CR] 读软盘文件命令。需键入文件名。当一个文件读入微机内存后, 机内原有文本即丢失。

C[CR] 显示软盘文件目录。

Q[CR] 退出文本编辑程序。

[CR] 显示操作命令简介。

二、APPLE EDIT 程序说明

10~20 句: 定义数组, 出错处理, 设置常数变量。

30~85 句: 操作命令输入, 判别。

90~100 句: I[CR] 命令解释执行子程序。通过调用行输入子程序实现编辑, A\$(0) 是行计数器。

105~190 句: 行输入子程序。利用 GET 语句接收键盘输入字符并转送给字符串变量 B\$。因“,”号及“:”号的使用有限制, 故分别以 CHR\$(158), CHR\$(159) 代替。行输入子程序输出的是字符串数组 A\$(w)。

400~402 句: 磁盘文件目录列表子程序。

500~550 句: 删除命令 $n_1 - n_2 D = [CR]$ 的解释执行子程序。

600~660 句: 屏幕列表显示子程序, 通过调用行输出子程序实现。

670~695 句: 行输出子程序。

700~740 句: 打印文本子程序, 通过调用行输出子程序实现。

800~840 句: 文本存盘命令 S[CR] 解释执行子程序, 文本以顺序文件的形式存入软盘。

860~895 句: 读软盘文件命令 R[CR] 的解释执行子程序。

900~920 句: 删除一行命令 nD[CR] 的解释执行子程序。

930~960 句: 修改命令 nE[CR] 的解释执行子程序。其中既调用了行输出子程序又调用了行输入子程序。

970~980 句: 插入一行命令 nI[CR] 的解释执行子程序, 其中调用了行输入子程序。

700~7010 句: 显示操作命令简介。

```

10 DIM A$(999)
15 ONERR GOTO 30
20 Z1=1:Z6=0:GOSUB 7000
30 PRINT:CALL=868:INPUT "COMMAND";L$
35 IF L$="T" THEN GOSUB 90
40 IF RIGHT$(L$,1)="L" OR RIGHT$(L$,2)="L-" THEN
    GOSUB 600
45 IF L$="P" THEN GOSUB 700
50 IF L$="S" THEN GOSUB 800
55 IF RIGHT$(L$,1)="D" THEN GOSUB 900
60 IF RIGHT$(L$,2)="D-" THEN A$(0)=STR$(VAL(L$
    )-1):GOTO 30
65 IF L$="R" THEN GOSUB 860
70 IF L$="Q" THEN END
75 IF L$="" THEN GOSUB 7000
76 IF RIGHT$(L$,1)="E" THEN GOSUB 930
78 IF L$="C" THEN 400
80 IF RIGHT$(L$,1)="I" THEN GOSUB 970
83 IF RIGHT$(L$,2)="D-" THEN 500
85 GOTO 30
90 W=VAL(A$(0))
100 W=W+Z1:GOSUB 105:A$(0)=STR$(W):GOTO 100
105 PRINT W;"TAB(5);
110 GET A$:IF A$=CHR$(13) OR A$=CHR$(26) THEN 180
120 IF A$ < > CHR$(8) THEN 132
125 IF LEN(B$)=0 THEN 110
130 IF LEN(B$)=1 THEN B$=""PRINT A$;"A$;GOTO
    125
131 B$=LEFT$(B$,LEN(B$)-Z1):PRINT A$;"A$;
    GOTO 110
132 PRINT A$;:IF A$=CHR$(21) THEN 110
140 IF A$="" THEN B$=B$+CHR$(159):GOTO 110
150 IF A$="" THEN B$=B$+CHR$(158):GOTO 110
160 B$=B$+A$:GOTO 110
180 IF A$=CHR$(26) THEN POP:PRINT"ZZ":GOTO 30
190 A$(W)=B$:B$=""PRINT:RETURN
400 INPUT ".DRIVE.NO.(1/2);L:IF L>2 OR L<1 THEN 30
410 PRINT CHR$(4);"CATALOG D";L
420 GOTO 30
500 J=VAL(A$(0)):K=VAL(L$):FOR I=2 TO 4:IF MID$(L$,
    I,1)="-" THEN 520
510 NEXT I:GOTO 30
520 L$=RIGHT$(L$,LEN(L$)-I):L=VAL(L$):IF K>L
    THEN 30
530 IF K>=J OR L>=J THEN 30
540 FOR I=L+Z1 TO J:A$(K+I-L-Z1)=A$(I):NEXT I
550 A$(0)=STR$(J-L+K-Z1):GOTO 30
600 IF L$="L" THEN R=1:GOTO 640
610 R=VAL(L$):IF RIGHT$(L$,1)="-" THEN 640
620 IF R>VAL(A$(0)) THEN RETURN
630 W=R:GOSUB 670:RETURN
640 IF R>VAL(A$(0)) THEN RETURN
650 FOR W=R TO VAL(A$(0)):GOSUB 670:IF PEEK
    (49152)=154 THEN POKE 49168,0:RETURN
660 NEXT W:RETURN
670 PRINT W;"TAB(5);
675 I=LEN(A$(W)):IF I=0 THEN PRINT:RETURN
677 FOR J=Z1 TO I:A$=MID$(A$(W),J,Z1):IF ASC(A$)=158
    THEN PRINT "

```

```

680 IF ASC(A$)=159 THEN PRINT "
690 PRINT A$;
695 NEXT:PRINT:RETURN
700 INPUT "TABLE (Y/N)";L$
710 PR#1:POKE 1657,40:PRINT
720 IF L$="Y" THEN FOR W=Z1 TO VAL(A$(Z6)):GOSUB
    670:NEXT W
730 IF L$="N" THEN FOR W=Z1 TO VAL(A$(Z6)):GOSUB
    675:NEXT W
740 PR#0:RETURN
800 PRINT "SAVE TO DISK":INPUT "NAME";R$:IF R$=""
    THEN 800
805 D$=CHR$(4)
810 PRINT D$;"OPEN";R$
815 PRINT D$;"WRITE";R$:PRINT A$(0)
820 FOR I=1 TO VAL(A$(0)):PRINT A$(I):NEXT I
830 PRINT D$;"CLOSE";R$
840 RETURN
860 PRINT "READ FROM DISK":INPUT "NAME";R$:IF R$
    "" THEN 860
870 D$=CHR$(4)
880 PRINT D$;"OPEN";R$
885 PRINT D$;"READ";R$:INPUT A$(0)
890 FOR I=1 TO VAL(A$(0)):INPUT A$(I):NEXT I
895 PRINT D$;"CLOSE";R$:RETURN
900 IF L$="D" THEN RETURN
910 K=VAL(L$):J=VAL(A$(0)):IF K>J THEN RETURN
920 FOR I=K TO J:A$(I)=A$(I+1):NEXT I:A$(0)=STR$(
    J-I):RETURN
930 IF L$="E" THEN RETURN
940 K0=VAL(L$):J=VAL(A$(0)):IF K0>J THEN RETURN
950 IF K0=0 THEN RETURN
960 W=K0:GOSUB 670:GOSUB 105:PRINT:RETURN
970 K0=VAL(L$):J=VAL(A$(0)):IF K0>J THEN RETURN
975 IF K0=0 THEN RETURN
980 FOR I=J TO K0 STEP -1:A$(I+Z1)=A$(I):NEXT I:A$
    (0)=STR$(J+Z1):W=K0:GOSUB 105:PRINT:RETURN
7000 HOME:PRINT "TEXT EDITOR.....FU-ZENGMING..
    1989.9.15..TONGXUN..HARBIN..KEJIDAXUE.105.XINXI
    AN"
7010 PRINT "I> INPUT":PRINT "L> LIST":PRINT "P>
    PRINT":PRINT "C> CATALOG":PRINT "S>
    SAVE":PRINT "D> DEL":PRINT "R> READ":PRINT
    "B> REENTER":PRINT "Q> QUIT":RETURN

```

小 广 告

转让 CEC-I 型中华学习机连一台驱动器 1200 元,
Z-80 卡 100 元, 游戏棒 30 元, 工具、游戏磁盘 (双面
拷贝) 每片 7 元。

联系地址: 四川纳溪县泸天化厂甲醇 周全
邮政编码: 646300

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碰到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 n_1 , n_2 表示, 如图

图三所示, n_1 和 n_2 不能同时为零。打印点列数 N 与 n_1 和 n_2 的关系如下: $N = 256 \times n_1 + n_2$ 。

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, 根据所选定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 $n_1 = 3$, $n_2 = 0$, 得到打印列数为 $N = 256 \times n_1 + n_2 = 3 \times 256 + 0 = 768$)。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]

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。


```

TYPE DAJC ASH
; DAJC.ASH ----- 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)? '
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'
LBPBG 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
INT 10H
MOV DX,3D9H
MOV AL,6
OUT DX,AL
IXKU: CALL 0LIX
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,1BX1
CMP AL,31H
JL IXEND
CMP AL,39H
JG IXEND
CALL JSW1
KUDY: CALL UJDYMK
JMP IXKU
IXEND:
LBPBG ENDP
PROC NEAR
MOV DX,OFFSET LBU6
MOV AH,0AH
INT 21H
MOV BX,OFFSET LBUFF1
MOV AL,1BX1
RET
JPUR ENDP
UJDYMK PROC NEAR
MOV CX,0EH
MOV BX,OFFSET LBU2

```

```

P000: MOV AL,1BX1
CALL DYNK
INC BX
LOOP P000
MOV BX,OFFSET LBU4
MOV CX,4
MOV AL,1BX1
CALL DYNK
INC BX
LOOP P01
MOV CX,300H
P00: MOV BX,OFFSET LBU3
MOV DX,3
PP2: MOV AL,1BX1
CALL DYNK
INC BX
DEC DX
JNZ PP2
LOOP P00
MOV BX,OFFSET LBU5
MOV AL,1BX1
CALL DYNK
INC BX
MOV AL,1BX1
CALL DYNK
RET
UJDYMK ENDP
DIS PROC NEAR
MOV DX,OFFSET LBU1
MOV AH,9
INT 21H
RET
DIS ENDP
DYNK PROC NEAR
XOR AH,AH
PUSH DX
XOR DX,DX
INT 17H
POP DX
RET
DYNK ENDP
JSIX PROC NEAR
MOV AL,1
CMP CX,1
JL JS10
MOV DL,2
JSP: MUL DL
LOOP JSP
JS10: RET
JSIX ENDP
QLIX PROC NEAR
MOV AL,0
MOV LBU3,AL
MOV LBU3+1,AL
MOV LBU3+2,AL
RET
QLIX ENDP
JSW1 PROC NEAR
CMP AL,30H
JG P11
SUB AL,30H
MOV CL,8
SUB CL,AL
MOV CH,0
CALL JSIX
MOV LBU3,AL
JMP P13
P11: MOV AL,0EH
MOV LBU3+1,AL
RET
P13: RET
JSW1 ENDP

```

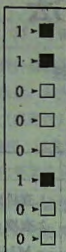
```
JSW2: PROC NEAR
      INC BX
      MOV AL, (BX)
      SUB AL, 30H
      MOV DL, 10D
      MUL DL
      INC BX
      MOV AH, (BX)
      SUB AH, 30H
      ADD AL, AH
      RET
ENDP
```

```
JSW3: PROC NEAR
      MOV CL, 16D
      SUB CL, AL
      MOV CH, 0
      CALL JSIX
      MOV LBU3+1, AL
      RET
ENDP
```

```
JSW3: PROC NEAR
      MOV CL, 24D
      SUB CL, AL
      MOV CH, 0
      CALL JSIX
      MOV LBU3+2, AL
      RET
ENDP
```

```
JSW4: PROC NEAR
      PUSH BX
      INC BX
      MOV CH, (BX)
      INC BX
      MOV CL, (BX)
      CMP AL, 1
      JE PPL0
      MOV LBU2+2, CH
      MOV LBU2+3, CL
      JMP PPL1
ENDP
```

```
PPL0: MOV AL, 30H
      MOV LBU2+2, AL
      MOV LBU2+3, CH
PPL1: POP BX
      RET
PPL: ENDP
CSEG: ENDS
      END LBPRG
```



打印一个点

不打印点

字节与打印针关系

图一



图二



n1n2 ≤ 2147 例

若 n1n2 取值范围大于 2147 列, 则超出的列在换行后打印。

图三

附录一

- 第 01 根针打印
- 第 02 根针打印
- 第 03 根针打印
- 第 04 根针打印
- 第 05 根针打印
- 第 06 根针打印
- 第 07 根针打印
- 第 08 根针打印
- 第 09 根针打印
- 第 10 根针打印
- 第 11 根针打印
- 第 12 根针打印
- 第 13 根针打印
- 第 14 根针打印
- 第 15 根针打印
- 第 16 根针打印
- 第 17 根针打印
- 第 18 根针打印
- 第 19 根针打印
- 第 20 根针打印
- 第 21 根针打印
- 第 22 根针打印
- 第 23 根针打印
- 第 24 根针打印

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

中华机故障检修法

四川南充市纺织职中 任 强

中华机是我国自行研制的广泛推广型八位学习机, 由于其功能强、造价低、维修相对容易而倍受微机爱好者和初学者青睐。然而由于中华机的年轻, 它的可靠性、连续性和稳定性仍是用户担心和苦恼的问题, 尤其是条件差的地方花了钱买机器, 坏了却无力再请人维修, 因此用户必须了解一些基本检修的方法。

首先要说明的是, 系统不能正常工作, 不一定就是主机本身问题。电源及其他电缆接口状态是否良好, 以及系统安装及调试是否合理应是检测第一步, 若确认是主机故障方才开启机器。由于中华学习机 CEC-I 基本电路思想是 APPLE II 的改动, 因此懂 APPLE II 电路和拥有 APPLE II 电路波形资料的同志可参照分析, 当然 CEC 上的专用大规模集成电路 MMU、IOU、CECWL 等是 APPLE II 上所没有的。

开启机盖, 拿去键盘, 即可看见主机板。其功能分布很简明, 左上方是 PAL 制式电路, 左下方是汉字系统, 右上方是驱动器接口, 右下方是键盘接口, 中部是主控部分。主机电源采用无工频变压器开关稳压电源。整机有两个晶振电路 (其主时钟信号源 Y_1 晶振值 14.31818MHz, 彩色基波信号源 Y_2 晶振值 4.43MHz)。三个集成电阻 RP_1 、 RP_2 、 RP_3 , 三种只读存储器 2716、2732、27256。整机共 481 个元器件、45 个集成块。EPROM U_7 中装有固化监控系统、CECBASIC、LOGO 及自检程序。

中华机故障检测方法很多, 首先必须判准是什么现象, 可能坏在哪个电路路上, 若条件允许应尽量作整机检查, 以免补东误西未治本。若尚能启动磁盘软件, 可采用《磁盘医生》之类的检测工具软件, 若不能启动盘, 尚能单机使用, 可同时按下“CTRL”、“RESET”、“TEST”三键对整机进行硬自检, 若主机根本不能正常启动, 就只有试探判断了, 现就常见故障现象及处理方法列表如下: (表一)

诚然, 检测的路子还很多, 你还需“因机而议”。笔者曾检修过一台 CEC 机, 开机屏幕只亮闪一下就没信息了。经检测, 电源各输出正常, CPU 亦正常; 于是检测到 IOU (U_{12}) 坏了, 换新后仍不行; 之后又查到 U_{11} PAL 逻辑定时阵列集成块坏, 换新后单机启动正常了, 进行硬自检也完全通过, 但一带驱动器, 就不行了, 因而将检测目标指向主机板右上方的驱动器电路 (即主机自带的驱动卡), 查出 74LS259 (U_{41}) 坏, 换新后就正常

表一

故障现象	常用维修检测方法
开机屏幕无任何显示	1. 查机内稳压电源各路输出 2. 查视 (射) 频输出, 查 CPU、MMU、IOU、 U_{16} 、 U_{17} 、 U_{13}
开机满屏显大方块	分别查 IOU、CPU、 U_5 、 U_6 、 U_{17} 、 U_7 、 U_{36}
开机随机乱显字符	查是否是 74LS374 (U_{17}) 使字符显示管理出故障
开机进入监控状态	查 U_7 集成块
按键字符不显示	1. 查 U_{25} 、 U_{26} 、MMU 和 IOU 等芯片 2. 查键盘性能
显示对, 但翻滚不止	查 IOU 是否被感应或电位为高, 此时可将 16 脚接地, 以增强抗扰力。
忽然花屏不能使用	查 CPU 的 $A_0 \sim A_{15}$ 是否局部短 (断) 路现象, 亦可再查 IOU 芯片
汉字状态出现乱显	查 U_{35} 及相关部件
汉字字形左右易位	将左字库 U_{33} 芯片与右字库 U_{34} 芯片互换插位
无彩色或色不正	查 L_5 、 Q_4 、 U_{18} 、 CR_1 、4.43MHz 振荡电路、 C_4 、 L_6 、 C_{41} (色稳定)
色不准	泛红可调 W_1 , 泛蓝可调 W_2 , 射频亮度调 W_3
DOS 不启动	查 U_{40} 、 Q_{10} 、 U_{41}
磁带机工作不正常	查 74LS251 (U_{20})、MC1458 (U_{19})
游戏口坏	查 74LS251 (U_{20})、NE558 (U_{19})

了。这台机器的故障很可能是因为用户使用驱动器不当, 使得一连串相关电路芯片被烧坏而致。

根据上述思想, 若一旦怀疑或确认某一部分、某一组件或集成块有问题, 需要检测分析一下, 那么可根据具体情况选用下表所述的一种或并用几种检测手法 (见表二)。

在维修中华学习机稳压电源后, 应检测组件的内阻,

表二

检测法	适用特点	检测说明
观察初探法	此法对新机器或被震动、拆卸过的机器有效	将可疑插件或带座芯片用手指适当压紧，避免断路；检查芯片（或组件）各级脚是否一一对应，以免短、断路。往往此举能立即见效。
电路替换法	宜于带座的插件或芯片	拿好的芯片（或组件）去替换对应可疑的电路器件或拿可疑芯片做对应替换
芯片重叠法	宜于不便拔插（双面精焊）的芯片或组件	将好的芯片放到可疑芯片背后，使两片的各引脚接触良好，再给电路加电，若判断准确，重叠芯片能使电路恢复正常
参数比较法	宜于有设备而不精通电路者	对可疑区进行波形和电平测试，并与电路波形图或样机测试数据比较、分析
逻辑分析法	复杂故障的基本检测手法	用示波器、万用表、逻辑笔、再生测试仪等仪器对机器进行逻辑测试及分析
测试内阻法	参数分析	一般，正常的组件输入端或输出端对地对电源都有一定内阻值。通常正向阻值在几十欧到一百欧左右，而反向电阻在几百至一千欧左右。正向阻值绝不会为零或近乎于零欧，而反向也不会无穷大，否则为元件故障。

用万用表 R × 100 档, 红笔接地时 +5V 端内阻应 400Ω~600Ω, 反响应 420Ω~450Ω; 红笔接地时 +12V 端内阻应 1K~1.1K, 反响应 800~860Ω; 红笔接地 -5V 端内阻应 6.5K~7.8K, 反响应 70K~80K; 每组电源对地间不能有短路现象存在, 方能上机通电。

对于主机的硬自检程序, 由于它是采用各部分参数累加, 再比较标准参数来判断是否 OK, 这种参数并非元件本身性能参数, 因而有它的局限性, 不可作具体检测中的唯一标准。

最后提一下，主机发生故障，有时原因和现象可能很简单明显，有时却很复杂隐蔽，以上谈到的故障分析检测技术属一般常用手法，不能墨守陈规，用固定模式去检测

AR2463打印机故障排除一例

山东54685部队 陈夕坤

故障现象：打印机输出的汉字、字符残缺不全，字迹模糊不清，漏点很多，且呈无规则出现。

故障分析：打开电源，打印机面板各指示灯显示均正常，并且打印机能进行正常自检测试操作，故初步判断故障在打印头和色带部位；取下打印机面板，仔细观察打印机在打印输出文稿时打印头和色带动作，笔者发现，打印头输出一行后往左复位时，打印针挂住色带，并把色带往左拉了一拉，取下色带盒，再仔细观察色带，发现有许多地方打穿并起毛，换上新色带后，故障现象依然；至此可以判断是打印头的问题，卸下打印头，细细查看打印针，并无断针现象，但在打印针及针孔附近有一层蜡纸油泥，因此我们分析可能是油泥堵塞了针孔，增加了打印针工作时的阻力，使打印针不能正常打印来不及复位而挂住色带，导致有点点打印不出来。

解决办法：将打印针朝下，用棉球蘸无水酒精轻轻擦洗打印针头部位和针孔，同时让少许酒精沿针从针孔流出；这样反复几次，直到洗净为止。注意：不能让清洗时的酒精液体流入打印头内部。

待酒精挥发后, 装上打印头, 重新开机打印, 故障消失。

(注：此法也同样适用于其它各类针式打印机和中文打字机) ■

任何故障。而正确使用和经常保养才是保证机器正常工作，延长使用寿命，减少故障的关键！

啟事

《电脑》90年第2期封二广告《双保险4片装5英寸磁盘盒》，邮购5盒起售，款15.00元，含邮包费。

随机点分布图象通用程序

内蒙古工学院物理教研室 王克勤

用计算机编制电子衍射, 气体分子速率分布, 电子云等原子, 分子现象的演示实验时, 用随机点分布构造图象往往取得直观、形象的效果。为此, 我编制了一个随机点作图通用程序, 它可以按给定的几率密度函数在某一区间的值, 在显示屏上一个选定区域构造一个设定形状 (矩形、圆、环状、椭圆, 以至由给定函数围成的形状) 的随机点分布。根据演示需要可用二种不同方式表现出随机点 (代表原子、分子) 的动态过程。

一、矩形域随机点分布的构造

设给定几率密度函数为 $f(x)$, 变量区间为 (x_1, x_2) , 该函数一般用矩形域表现。显示屏矩形域设为 $(x_3, x_4), (y_1, y_2)$ 。首先求出该函数在区间 (x_1, x_2) 的极大值, 极小值 Z_m, Z_n , 则

$$Z_n < f(x) < Z_m$$

为用点数表现该函数决定的随机点分布, 作变换:

$$f_1(x) = (f(x) - Z_n) / (Z_m - Z_n)$$

$$\text{则 } 0 < f_1(x) < 1$$

以上函数的预处理由子程序 950-1070 进行, 分别在

各模块作图并调用。

LIST 950, 1070

950 X=X1-(X2-X1)/100

960 FOR I=1 TO 100

970 X=(X2-X1)/100+X

980 Y(I)=FN Y(X)

990 NEXT

1000 A1=Y(I): B=Y(I)

1010 FOR I=2 TO 100

1020 IF Y(I) > A1 THEN A1=Y(I)

1030 IF Y(I) < B THEN B=Y(I)

1040 NEXT

1050 C=1/(A1-B): D=B/(B-A1)

1060 DEF FN B(X)=C*FN Y(X)+D

1070 RETURN

程序运行前提示在 900 行定义该密度函数用 DEFFNY(X), 处理后的 $f_1(X)$ 用 FNB(X) 定义出。

一般讲, 变量区间 (X_1, X_2) 是任意的, 而显示屏图象点却是固定的, APPLE II 中高分辨密度第一页, 图象点在 $0 \leq X \leq 279, 0 \leq Y \leq 159$ 之间, 直接用 (X_1, X_2) 作图是不行的, 其图象点或超出显示屏, 或是在屏上很小区域显示, 难以表现其分布特点。

这时, 用 $X' = X_1 + (X_2 - X_1) \text{RND}(I)$ 语句产生随

机变量 X' , 由此代入密度函数 $f(x')$ 计算其几率密度值, 再作变换:

$$X' = X_1 + (X_2 - X_1) (X' - X_1) / (X_2 - X_1)$$

$$Y = Y_1 + (Y_2 - Y_1) \text{RND}(I)$$

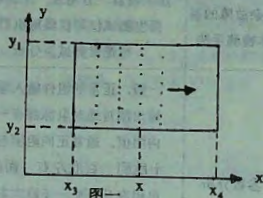
$$\text{这样, } X_3 \leq X' \leq X_4, Y_1 \leq Y \leq Y_2$$

即随机点分布在矩形域 $(X_3, X_4), (Y_1, Y_2)$ 了。

可用这种方式把任意区间的几率密度分布情况用随机点分布在矩形域中表现出来, 显示域可以放大, 缩小以符合编制要求。

作出随机点分布的方式可用设 X 方向顺序作出与在平面全随机作出两种方式, 可按编制需要选用。

1. 方式 1: 以设 X_3, X_4 顺序方式设 Y 方向构造由函数 $f(x)$ 决定的随机点分布。



由图一, 设屏上任一 X 处设 Y 作出最多点数为 N_1 , 这样, 任一 X 处按密度函数决定的实际点数为 $N_i(X) = N_1 f_1(X)$

$$f_1(x) = \frac{N_i(x)}{N_1}$$

即为随机点密度与密度函数 $f_1(x)$ 相一致。

程序 210-270 即为按此方式作点程序

LIST 210, 270

210 HGR: HCOLOR=3

220 FOR X=X1 TO X2 STEP (X2-X1)/(X4-X3)

230 N=FN B(X)*N1

240 FOR K=1 TO N

250 XT=(X4-X3)*(X-X1)/(X2-X1)+X3

252 IF A\$="AR" THEN GOSUB 1240

255 Y=Y1+(Y2-Y1)*RND(K)

260 HPOINT XT, Y

265 NT=NT+1: VTAB 23: HTAB 15: PRINT "NO. ";

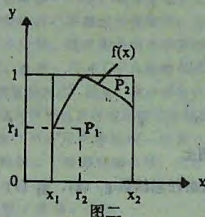
NT

270 NEXT: NEXT: NT=0

其中 $f_1(x)$ 由 FNB(x) 定义, 252 行后面说明。用这种方式很形象地编制出蔡特曼—葛正权实验中银原子打在弯曲玻璃上出现分子密度分布过程。

(2) 方式 2: 以全随机方式在矩形显示域 (x_3, x_4) (y_1, y_2) 构造由密度函数 $f(x)$ 决定的随机点分布。

由产生随机变量的余选抽样方法⁽¹⁾, 依次产生随机数 r_1, r_2 , 在矩形内任设一点 P, 若其位于曲线 $f_1(r_2)$ 下面, 即 $r_1 < f(r_2)$ 则取 r_2 为模拟随机变量, 否则, 不画点进行新实验。(图二)



图二

按此方法编制程序, 以 $X = X_1 + (X_2 - X_1) \text{RND}$ (I) 获得一随机数 r_2 , 代入 $f_1(r_2)$ 中, 用 $\text{RND}(k)$ 获得均匀分布的伪随机数 r_1 , 这时, $0 \leq r_1 \leq 1, 0 \leq f(r_2) \leq 1$, 若 $r_1 \leq f(r_2)$ 则画点, 若 $r_1 > f_1(r_2)$ 则不画点, 即可在矩形区域 $(x_3, x_4), (y_1, y_2)$ 中以全随机方式构造出按密度函数 $f(x)$ 分布的随机点了。程序 320—390 即为此段程序。其中 N_1, N_2 为设 x, y 方向作出的最大点数, 总点数 $N_1 \times N_2$

```
LIST 320, 390
320 HGR: HCOLOR=3
330 FOR I=1 TO N1: FOR K=1 TO N2
340 X=X1+(X2-X1)*RND(I)
350 N=FN B(X)
360 IF N<RND(K) THEN 340
370 Y=Y1+(Y2-Y1)*RND(K)
375 XT=(X4-X3)*(X-X1)/(X2-X1)+X3
380 HPLOT XT, Y
385 NT=NT+1: VTAB 23: HTAB 15: PRINT "NO.=";
NT
390 NEXT: NEXT: NT=0
```

用这二种方法构造的实例如图三:

图三

密度函数 $f(x) = X^2$

变量域: 100—700

显示域: $\begin{cases} (0-200) \\ (0-130) \end{cases}$

总点数: $N=1091$

二、圆域 (环形、扇形、椭圆) 随机点分布的构造

若给定径向, 角向密度函数 $f(r), f(\theta)$, 则一段在圆域内构造随机点分布。

设, 密度函数变量区间为 $(R_1, R_2), (\theta_1, \theta_2)$ 在微机显示屏上以 x_0, y_0 为圆心, 径向 (R_3, R_4) 角向 (θ_3, θ_4) 区域构造随机点分布, 首先作变换

$$\begin{cases} R' = R_1 + (R_2 - R_1) \text{RND}(I) \\ \theta' = \theta_1 + (\theta_2 - \theta_1) \text{RND}(I) \end{cases}$$

获得随机变量 R', θ' , 代入 $f(R'), f(\theta')$ 中得几率密度值, 再作变换

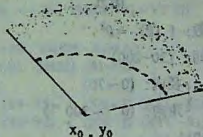
$$R'' = R_3 + (R_4 - R_3)(R' - R_1)/(R_2 - R_1)$$

$$\theta'' = \theta_3 + (\theta_4 - \theta_3)(\theta' - \theta_1)/(\theta_2 - \theta_1)$$

用语句:

$$\begin{cases} X' = X_0 + R'' \cos \theta'' \\ Y' = Y_0 + R'' \sin \theta'' \end{cases}$$

即可在显示屏上构造出一个以 x_0, y_0 为中心, 径向 (R_3, R_4) 角向 (θ_3, θ_4) 范围的环形、扇状区域的随机点分布了。(图四)



图四

密度函数 $f(r) = r^2$

变量区间: $(10 \sim 20)$

显示域: 径向: $(40 \sim 70)$

角向: $(230^\circ \sim 350^\circ)$, 总点数 407

若径向密度函数为 $f(r)$

用二种不同方式作出圆域分布的程序分别为程序行 500—560, 790—870 段

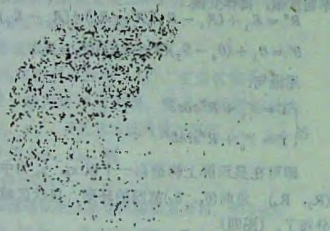
```
LIST 500, 560
500 FOR R=R1 TO R2 STEP (R2-R1)/(R4-R3)
510 N=R*FN B(R)*N1/R2
520 FOR K=1 TO N
530 Q=Q1+(Q2-Q1)*RND(K)
535 RT=(R4-R3)*(R-R1)/(R2-R1)+R3
540 X=X0+RT*COS(Q): Y=Y0+RT*SIN(Q)
550 HPLOT XT, Y
555 NT=NT+1: VTAB 23: HTAB 15: PRINT "NO.=";
NT
560 NEXT: NEXT: NT=0
```



```

LIST 790, 870
790 HGR: HCOLOR=3
800 FOR I=1 TO N1: FOR K=1 TO N2
810 R=R1+ (R2-R1) * RND (I)
820 N=FN B (R) * R / R2
830 IF N< RND (K) THEN 810
840 Q=Q1+ (Q2-Q1) * RND (K)
845 RT= (R4-R3) * (R-R1) / (R2-R1) + R3
850 X=X0+RT * COS (Q): Y=Y0+RT * SIN (Q)
860 HPOINT X, Y
865 NT=NT+1: VTAB 23: HTAB 15: PRINT "NO=";
NT
870 NEXT: NEXT: NT=0
    
```

若已知角向分布函数 $f(\theta)$ ，作出圆域分布的程序与上二段程序同，将 R 与 θ 变互换，仅举一例说明。(图五)



图五

密度函数: $f(\theta) = \theta^2$

变量区间: (30~200)

显示域: { 径向: (0~70)
角向: (0°~300°)

点数: 1911

需要说明的是，作圆域分布的，若密度函数 $f(r)=C$ (均匀分布)，这时，在半径 $r=R_1, R_2$ 处作出点数相同，但其密度分别为 $P_1=N_1/2\pi R_1, P_2=N_2/2\pi R_2$ ，当 $N_1=N_2$ 时 $P_1>P_2(R_2>R_1)$ ，即并不是均匀分布了。为得到均匀分布，需将密度函数改为 $f_1(r)=r/C$ 方可。若作密度函数为 $f(r)$ 的分布，需改为 $f(x)=r f(r)$ 。

显然，若取 $R_3=0, \theta_3=0^\circ, \theta_4=360^\circ$ ，则为圆形区域。举图六为例。

图六

密度函数: $f(r) = \cos^2(2\pi r)$

变量区间: (0~2)

显示域: 径向: (0~60)

角向: (0°~360°)

总点数: 800

若改程序行为:

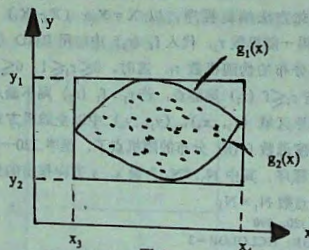
$X=X_0+A\cos\theta$

$Y=Y_0+B\sin\theta$

即可构造椭圆域了 (图七)。

图七

三、构造由任意函数 $g_1(x), g_2(x)$ 围成的形状随机点分布。



图八

如图八，若在显示屏 $(x_3, x_4), (y_1, y_2)$ 区域构造由函数 $g_1(x), g_2(x)$ 围成的形状的随机点分布，则需先求出二个函数在 (x_3, x_4) 区间的共同极值 Y_m ， Y_n ，则：

$$Y_n \leq g_1(x), g_2(x) \leq Y_m$$

作变换：

$$x = X_3 + (X_4 - X_3) \text{RND}(I)$$

$$y_1 = y_3 + (y_4 - y_3)(g_1(x) - y_n) / (y_m - y_n)$$

$$y_2 = y_3 + (y_4 - y_3)(g_2(x) - y_n) / (y_m - y_n)$$

再转入程序行 255 即可。

以上求极值与变程序用子程序 1100~1237, 1240~1260 作出，函数 $g_1(x), g_2(x)$ 在 1100 行定义。

下面给出由函数 $g_1(x) = (2x)^{1.5}$ 及 $g_2(x) = x^2$ 围成区域中作出几率密度函数 $f(x) = x$ 的随机点分布实例 (图九)。

中华机上配置简易 D/A 转换接口

西南民族学院 杨宪泽

本刊1990年第一期登载了“中华机直接用于数据采集和开关控制的实现”一文。笔者在这篇文章中，介绍了利用中华机游戏接口，不增加任何硬件，由软件编程实现开关信号、脉冲信号、模拟信号的多路数据采集和多路开关式控制方法。然而，使用D/A是微机实施工业控制和辅助电路实验的重要方式，因此，有必要给中华机配置D/A转换接口。考虑到中华机系统情况和价格低廉这一特殊性，配置的D/A接口不占用中华机唯一的接口槽（这个槽可能要安装打印机接口，绘图仪接口等，已经够紧张了），仍然使用游戏接口引脚，同时，这一设计采用软、硬件结合措施，力求价低、简单、易制。

一、D/A转换接口设计原理

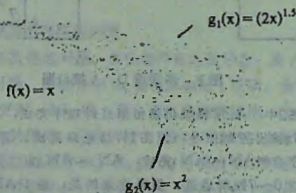
按上述原则构思的D/A转换方法是，利用6502汇编语言编一个延时程序作D/T转换，即将数字量转换成相应长的时间T，由中华机游戏口AN0脚输出一控制信号，控

制T/V转换电路，将T还原成相应的模拟电压，从而完成D/A转换。原理见图1。

图1原理：由方法构思可知，D/T转换依靠中华机编程完成。AN0发出的控制信号间隔时间T长，D的值大，T短，D的值小，T/V转换由D触发器IC₁，积分器IC₂保持器IC₃ IC₄及隔离电路G₁ G₂等组成。AN0发出一个控制信号时，作用于IC₁的CP端，使Q端为“0”，D₁ D₂截止，积分器开始积分，有：

$$V_1 = -\frac{1}{RC} \int_0^T (-V_R) dt = \frac{V_R}{RC} T \quad (1)$$

同时，由于Q为“0”，G₁导通，IC₃跟踪V₁值；Q为“1”，G₂截止，IC₃与IC₄通路断开，IC₄保持原值。当t=T时，AN0又发出一个控制信号，Q=1，D₁ D₂导通，C



图九

子程序1100-1237, 1240-1260如下：它们可在程序206-270中206, 252行分别调用。

```

LIST 110, 1260
1100 DEF FN C(X) = (2 * X) ^ 1.5: DEF FN D(X) = X ^ 2
1105 Y3=Y1: Y4=Y2
1110 X5=X3- (X4-X3) / 100
1120 FOR I=1 TO 100
1130 X5=X5+ (X4-X3) / 100
1140 Y1(I) = FN C(X5): Y2(I) = FN D(X5)
1145 NEXT
1150 A1=Y1(I): B1=Y1(I): A2=Y2(I): B2=Y2(I)
1160 FOR I=2 TO 100
1170 IF Y1(I) > A1 THEN A1=Y1(I)
1180 IF Y1(I) <= B1 THEN B1=Y1(I)
1190 IF Y2(I) > A2 THEN A2=Y2(I)

```

```

1200 IF Y2(I) <= B2 THEN B2=Y2(I)
1210 NEXT
1220 IF A2 >= A1 THEN A1=A2
1230 IF B2 <= B1 THEN B1=B2
1237 RETURN
1240 Y1=Y3+ (Y4-Y3) * (FN C(XT) - B1) / (A1-B1)
1250 Y2=Y3+ (Y4-Y3) * (FN D(XT) - B1) / (A1-B1)
1260 RETURN

```

同样值得注意的是由于 $g_1(x)$, $g_2(x)$ 围成的区域不是矩形区域，随机点密度也会与原设定密度函数 $f(x)$ 有偏差，也需视不同情况对 $f(x)$ 进行修正。

将以上各段程序有机地链结起来，并用菜单显示选择各项操作和参数，便成为完整的通用程序了。

在编制程序时作简化处理，定义密度函数在900行只需用 DEFFNY(X)=.....即可。FNY(X)可以是直角、径向、角度等函数，作均匀分布，则在提示是否作均匀分布时，打入“Y”即可。可根据需要设定随机点总数，并在作出时，逐一显示其点序数与总数。这里不一一列出。

本程序用 BASIC 语言编制，在 APPLE II 机上通过，效果良好。

参考文献

(1)数值计算方法 梁康等编 国防工业出版社 P217

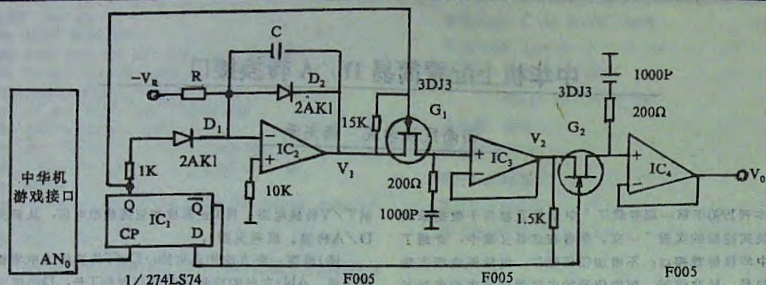


图1 D/A转换接口原理图

迅速放电，这时G₁截止，G₂因Q=0导通，于是IC₄带着IC₃在t=T时刻最大值V₁进入保持状态，有：

$$V_0 = V_2 = V_1 = \frac{V_A}{RC} T \quad (2)$$

(2)式说明，V₂跟踪的是积分器V₁的值，跟踪期间IC₄被G₂断开保持以前的值；V₂跟踪结束后得到V₁的最大值，而V₁受AN₀控制迅速为0，这时由于IC₃被G₁断开V₂值不变，G₂导通V₂值经IC₄输出V₀。显然，两次控制信号时间间隔T的大小决定了V₀的大小，D/A转换就这样实现了。

二、转换计算及控制信号编程

控制时间间隔为：

$$T = T_1 D \quad (3)$$

(3)式中，T₁为延时期序时间，D为D/A转换数字量。我们取D为一个字节二进制数，即为0~255。如果转换的电压为0~5伏，那么V_R=5伏。显然，T₁取得越小，D/A转换时间T越短。但是T₁参数受R、C数值限制，我们取R为100K，C为1μ，有：

$$V_0 = \frac{V_A}{RC} T = \frac{V_A}{RC} T_1 D = \frac{5T_1 D}{100 \times 10^3 \times 10^{-6}}$$

要V₀为5伏，D应为255。经计算T₁为39μs，这就是程序延时间。由此可知，这一转换器转换速度为39μs~100ms，分辨率为8位，转换精度约为0.1%。

控制信号实施D/A转换程序如下：

```
LDX # 00, 初始化
STA $C058
STA $C058, 送初始控制信号于AN0
LOOP, NOP
NOP, 共15个
NOP
INX
CPX (D), 与某一单元存入的转换数D比较
BNE LOOP, 1≠T重复
STA $C059, 送T
STA $C058, 送T
RTS
```

三、多通道D/A的扩充

如果需要实施多通道D/A控制，可把图1扩充为图2。

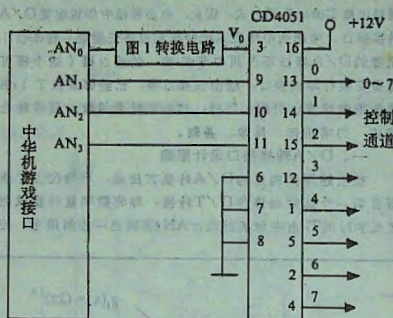


图2 多通道D/A接口图

图2中，A/T转换仍然由原延时程序完成，T/V转换由AN₀输出控制信号，原图1转换电路完成。通道数选择由程序控制AN₁~AN₃完成，AN₁~AN₃为二进制000~111对应0~7控制通道。值得注意的是，由于AN₁~AN₃译成000~111不如主机地址线那样方便，所以通道译码程序有些特殊，这是因为：

对地址\$C05A进行存数操作，AN₁输出低电平
对地址\$C05B进行存数操作，AN₁输出高电平
对地址\$C05C进行存数操作，AN₂输出低电平
对地址\$C05D进行存数操作，AN₂输出高电平
对地址\$C05E进行存数操作，AN₃输出低电平
对地址\$C05F进行存数操作，AN₃输出高电平
那么，选择0通道将使用程序

```
STA $C05A
STA $C05C
STA $C05E
选择1通道使用程序
STA $C05B
STA $C05D
STA $C05F
```

依此类推，可以写出选择2~7通道的程序。

对 $\mu p-1$ 微电脑套件的改进意见

广州市电子职业高级中学高三 (3) 班 关本立

指导教师 何文生

【编者按】《学装微电脑》对于广大青少年学生来说，是一个新的内容。为了全面掌握计算机硬件和软件的基础知识，更好发挥计算机的效能，由李仲先、潘懋德教授研制的 $\mu p-1$ 型微电脑套件，通过组装、实验和操作，可有效了解计算机各部分功能及其联系，并大大增强感性认识。

本文作者于今年元月在广东省惠州市参加了全国青少年计算机冬令营暨第五届全国青少年计算机竞赛，荣获硬件组装三等奖以及 $\mu p-1$ 型电脑开发应用三等奖。本文通过组装、调试对 $\mu p-1$ 微电脑提出的改进意见仅供参考。

我们在组装及使用 $\mu p-1$ 微电脑套件的过程中，发现本机存在一些问题，有待改进；现提出我们的设想和改进方法如下。

一、使用上存在的问题

首先是按键的问题。有些套件在组装好后，按 A_1 键会出现误计数。原套件中 A_1 键电路如图 1 所示，当按下 A_1 键再放开时， IC_{18} 的 3 脚会产生负跳变脉冲，发出使计数器加 1 的信号。但由于按 A_1 键时产生的跳变脉冲不止一个，因而必须用 R_6 、 R_7 及 C_9 组成的 RC 回路将其滤除。由于原套件中 C_9 取值太小，因而出现上述的误计数。试将原设计的 C_9 (1μ) 换用 10μ 后故障消失。

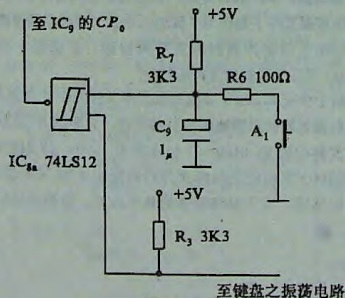


图 1

还有些套件在组装好后，按一次 RES 键，程序有时运行两次，原因与上述 A_1 键问题类似，将容量为 1μ 的 C_{10} 换成 10μ ，问题也顺利解决。

二、对内存分配的改进

本机在加入 I/O 接口后，部分存储单元被占用，使容量仅为 1K 的 RAM 少用了 1/4，这是不合适的。改进的关键问题是如何在尽量少改动电路的情况下重新分配内存，增大用户可用空间。

我们将电路改成如图 2 所示。其中三态门使用了 IC_{14} 中空置的一个，它能保证 DMA 状态下必然选中 RAM。具体改装方法如下：

1、将 IC_{18} 的 2、3 脚和 A_9 的连线断开并接至 CPU 的 20 脚。

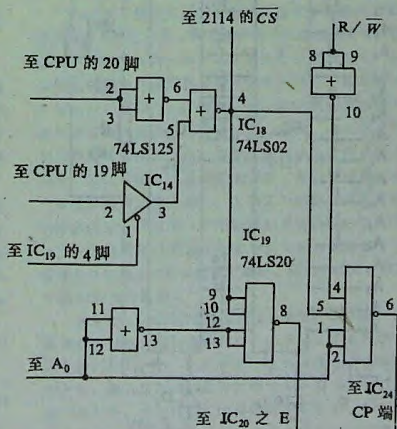


图 2

2. 将 IC₁₆ 的 5 脚和 A₉ 断开并接至 IC₁₄ 的 3 脚。

3. 连接 CPU 的 19 脚和 IC₁₄ 的 2 脚, 并将 IC₁₄ 的 1 脚和 4 脚相连。

至此用户可使用整个 RAM。由于只有当 A₁₀ 为高电平且 A₁₁ 为低电平才能选中 I/O 区地址为 \$800~\$BFF; 其中偶数单元为输入接口, 而奇数单元为输出接口, 用户可放心使用原 I/O 所占单元。

三、程序及数据的保护

由于 $\mu P-1$ 型微电脑只有 RAM 而没有 ROM, 因而关机后内存中所有程序和数据就会全部消失, 使用很不方便。解决方法可用干电池为它的 RAM 加保护电源, 使 +5V 电源断开后由干电池对 RAM 供电。

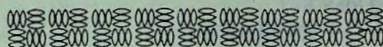
具体电路如图 3 所示。操作此机对 RAM 读或写时 K' 应拨向“读写”端, 否则应拨向“保护”端。原电路中的 RAM 应换用耗电较少的 5514, D₁ 和 D₂ 应选质量较好, 反向电阻大的整流二极管。

图中 D₁ 和 D₂ 作用是保证向 RAM 供电: 当接上 +5V 电源时由 +5V 电源对 RAM 供电, 自动断开 +3V 电源 (由两个 5 号电池串联而成), 而当 +5V 电源断开时则由 +3V 电源自动恢复供电。K' 的作用是保护数据, 减少耗电。若没有 K', 由于开、关机瞬间 R/W、CS、AB、DB 均处于随机状态, 因而当 R/W 和 CS 均为低电平时就会破坏 RAM 中的数据; 而当装上 K' 且将其置于“保护”端, CS 被强置为高电平, 保护了数据也减少了耗电, K' 是必不可少的。R' 作用是防止在拨动 K' 的瞬间 CS 出现高阻态, 避免 RAM 出现自激等现象。

总的来说, 将 1 μ 的 C₉、C₁₀ 换成 10 μ 对本机的 RES 键及 A₁ 键较有保障; 而对内存的重新分配则提高了 RAM 的使用效率; 对数据的保护则加强了本机的功能。

《Dbase-III 数据库文件的加密

和解密试验》的补充



广州钢铁有限公司经管处 林 沛

贵刊 89 年第 3 期黄光黎同志所作“Dbase-III 数据库文件的加密和解密试验”一文写得很好, 很有实用性。但是, 在数据库文件解密方面, 该解密程序对含有 MEMO 字段的数据文件不起作用, 其原因是该解密程序只考虑没有 MEMO 字段的数据文件的情形, 而忽略了含有 MEMO 字段的数据文件的情况。

对于含有 MEMO 字段的数据文件的解密方法, 可以仿照黄光黎同志所编制的解密程序, 只要把 XG2.TXT 文本文件中的 E 0100 03 修改为 E 0100 83 (因为含有 MEMO 字段的数据文件结构是以 83H 开始标志) 就可以实现。至于解密程序的具体内容, 这里就不详述了。

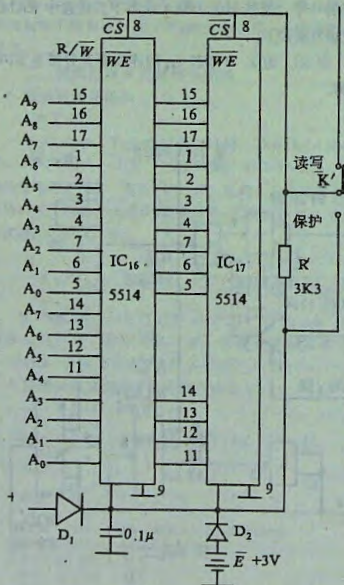


图 3

谈谈数据结构

黃思曾

一、概述

程序设计已从技巧发展为科学，其中包含三个密切相关的课题，即：数据结构、算法设计和分析、程序结构方法。

数据是描述客观事物的数、字符以及所有能输入到计算机中并被处理的信息集合。实质上，数据就是客观事物表示的一种抽象。数据结构这门学科就是研究如何把客观世界中待处理的对象逐层抽象为计算机可以接受的某种表示形式的。

算法是指解决某类特定问题的有穷步处理动作的序列。它最终要体现为能在计算机环境中执行的某种程序设计语言的语句序列，即程序。

数据结构和算法是互相连系的。所谓程序就是在特定的数据结构上用某种语言来表述的算法。所以，不了解施加于数据之上的算法就无法决定应使用的数据结构；反之，算法的设计和效率又依赖于它选择的数据结构。

要设计出一个高质量的程序尚需考虑程序结构问题。模块化原则是当前流行的结构方法。它强调自顶向下、逐步求精地用尽可能独立的模块来设计构造一个程序。

二、数据结构的分类

从数据的值在使用时是单一不可分割的，还是由更基本的可独立使用的成分值构造而成这一点，可以把数据结构划分为简单的和构造性的两大类。构造性数据的成分值本身可以是简单数据也可以是构造数据，这样它所表达的可以是十分复杂的数据。

换一个角度，一类数据其结构和占据的存储空间都是预先确定的，在程序执行期间不发生改变，可以称之为静态的数据结构。所有的简单数据和某些构造数据结构是静态的。另外的某些构造数据在程序的执行过程中，只保持特定的构造特征，大小规模却在变化，可以称之为具有动态的数据结构。

目前常见的数据结构分类如下：

在程序设计语言中，简单数据和静态的构造数据的表示通常体现为数据类型的定义。如PASCAL—共提供了十一种类型定义，除文件类型外其它都是静态数据结构的定义。BASIC也有三种类型的简单变量定义，以及数组和文件等数据结构组织。

对于动态的数据结构,描述和表示的基本方法是把它们看成是相互间存在某种特定联系模式的结点的集合。所谓结点是动态结构的基本构造单位,它本身是具有静态结构的数据,结点之间存在的联系模式就表征出不同的数据结构。在算法的执行过程中,结点可以增减,从而结构的规模可以发生变化;但结点间的联系模式保持不变,这样才能保持动态的数据结构的构造特征。

只考虑结点之间的联系构成特征,称为数据的逻辑结构;把逻辑结构映射到计算机中得到的存储表示形式称为数据的存储结构(或物理结构)。而在某种程序语言环境中,用这种或那种方法去表示某种数据,说成是“实现”该数据结构。

如前述,静态数据结构的实现通常体现为选择语言所提供的某种数据类型,而实现一般动态结构的通常方法是:选择某种静态结构(如记录)来表示结点的定义,结点内部的数据通常分为数据域和指针域两大部分,结点间的联系由结点内部包含的指针域来体现,指针域的值给定与本结点相联系的其它结点的存储位置。显然,一个具有动态结构的数据其取值与变动,要依据在算法执行中各结点的数据库和指针域的取值及变动来体现,无法加以静态定义。文件组织的表示是个例外。因为文件驻留在外存设备上,对它的访问基于对它的一个个构成元素的访问之上。所以,定义文件构成元素的类型(通常称为文件的记录型)以及元素在文件空间上的组织型式(顺序、索引、相对、散列...)即能表示文件组织。

动态数据结构的实现会受制于具体的语言环境。PASCAL语言具有记录类型和指针类型，适于定义结点的结构。而标准的NEW过程，使程序可动态地获得结点所需的存储空间。因此用PASCAL语言在内存中实现动态数据结构就特别方便。而在BASIC这样的语言环境中，就要使用变通的方法。如定义若干个数组用作作为结点的存储空间。每个数组元素表示一个结点的一个域（数据域或者指针域），因此具有相同序号的若干个数组元素就表示了一个结点。结点间的联系由指针值体现。此时，指针值就是某个数组元素的序号，因为数组元素的存储位置可由序号体现。

{ 简单的: 整型、实型、字符型、布尔型、 纯量型、子界型、指针型等等 构造的: { 数组、记录、集合 线性表、树、图 文件组织, 数据库的数据模型	} 静态结构
	} 动态结构

可见, 研究动态的数据结构首先要关心数据的逻辑结构, 即结点间的联系模式。而结构的实现往往是技巧问题, 而且和语言环境相关。

最常见的动态数据结构有三种, 即: 线性表、树、图。

三、线性表

一个线性表是 n 个($n \geq 0$)结点的有限序列, 除第一个和最末一个外, 表中的每一个结点有且仅有一个直接的前趋结点, 有且仅有一个直接的后继结点。这样, 每个结点都有一个后继结点和/或一个前趋结点, 这就是线性表中结点的联系模式。

因此, 线性表结构具有两个根本特征:

(1) 均匀性。所有构成结点必须具有完全相同的结构;

(2) 有序性。各结点线性有序。为了反映这种有序性, 表结点用下标来标识, 所有下标都是有序量, 比如说整数。

对于客观世界中一批构造相同的关联数据, 如果只有相邻的两个数据间才体现某种直接联系(如领先或滞后), 就可以用线性表结构来表达这批数据。

对线性表施加不同的限制就得到几种在程序设计上有着广泛应用的数据结构。列表如下:

名称	特点(施加的限制)	结点
(1) 数组	结点的个数预先加以确定不再动态地增减	数组元素
(2) 栈	只能在固定的一端进行结点的插入和删除	栈元素(栈顶元素、栈底元素)
(3) 队	在固定的一端(队尾)插入结点, 在另一端删除结点	队元素(队首元素、队尾元素)

(1) 数组

结点数目预先确定的线性表结构, 在各种程序设计语言中均以数组类型出现。这种限制使数组变为一种静态结构。典型结构特征是: 数组是线性表, 它的每一个元素又可以是一个数组, 这些子数组的元素还可以是一个数组……直到最低层次的数组只含有简单的数据元素为止。通常用数组的维数来描述数组的这种层次构成。

在广东省青少年计算机竞赛(以下简称竞赛)的历年试题中, 使用数组结构来表达处理数据的题目比比皆是。要概括的话, 那就是: 既然数组是一类特殊的线性表, 它就具有均匀性和有序性的特征, 即每个数组元素应具有相同的结构(简单的或构造的)以及用有序下标来标识数组元素。此外, 应分清几个概念: 数组(变量)和数组类型, 数组元素和数组元素的类型, 数组元素的下标和数组元素下标的类型。这些基本概念在不同的语言中有不同的体现形式。

(2) 栈

栈是在操作特征上加以限制的线性表。固定插入元素的一端称为栈顶, 另一端称为栈底。栈的这种限制并未使栈失去动态结构的基本特征, 栈的规模随元素的入栈出栈而变化, 但栈元素的出入操作显然遵循后进先出的原则。

利用一维数组来表述栈是方便的, 只是会引起另一个附加的限制, 栈的容量有了一个上限。这时, 如果数组再无空间容纳要进栈的元素, 就会出错, 称之为上溢; 如果执行出栈操作时栈内已经没有元素, 则表示栈空, 称之为下溢。显然, 进出栈操作均须知道当前栈顶元素的位置, 这个位置总是用一个栈顶指针来记录的。现在, 它的当前值就是栈顶元素所在数组元素位置的序号。

对栈结构的使用要紧紧抓住栈元素的后进先出特征。八九年竞赛题六(题目见上期)要求找出国际象棋盘上“马”的周游路线, 路线要遍历所有方格一次。程序应使用试探和回溯的算法。从每一方格, “马”最多可以有八个方向前进, 程序必须记下走过的路径(方格坐标)和方向, 以便在未遍历完而又跳到某个无法再前进的位置(因为每个方格只准经历一次的限制)时, 就必须沿路径回溯一个位置, 试图向另一个未尝试过的方向前进。那么, 用什么样的数据结构来记录遍历路径好呢? 考虑到每次回溯总是要求得到当前路径上最近一个出发方格的位置, 显然应设立一个栈来记录“马”经历的每个方格的坐标位置, 最后入栈的方格位置(栈顶元素)就是需要回溯时应退回到方格位置。

八八年竞赛复试题五中如果使用了栈结构, 算法就显得十分简明。题目要求描述拆下互连的 N 个环的过程, 除第一个环可任意拆外, 第1个环($2 \leq i \leq N$)必须在第1-1号环装上而且1-2号到1号环均已拆下的条件方能拆去。只要意识到为了拆下1号环而重新装上的第1-1号环是当前最后一个已拆下过的环这一点, 就不难定义两个栈, 一个从底到顶存放 N 到1号环, 另一个栈作为暂存当前已拆下的环的空间。显然, 在拆1号环时, 暂存栈从底到顶存放的是1到1-1号环。因此拆1号环的动作就变为: 从暂存栈退出1-1号环, 将其压入环栈, 再从环栈连续退出两个环(1-1号环和1号)压入暂存栈。

上面的两个例子充分表现了程序算法和数据结构的依存关系。正确而简明的算法有赖于选择合适的数据结构来表达算法所加工的数据。反过来说, 特定的数据结构总是和特定的算法相联系的。

(3) 队

对线性表施加另一种操作限制, 就得到称之为队的线性表。队是在固定的一端插入元素, 而在另一端删除元素的线性表。可见, 在一个队中元素具有先进先出的特征。

实现一个队的常用方法是把一个由一维数组确定的空间视为一个首尾相接的环形区, 用两个指针来分别表示队首元素和队尾元素的位置, 每加入一个元素, 尾指针先向某个方向移动一个存储位置; 而每删除一个元素, 头指针向相同方向移动一个位置。出入队操作的关键点在于, 当指针移动方向沿数组元素序号递增的方向时, 数组空间序号最大的位置的下一个位置是序号最小的那个位置。这就是所谓循环队的

结构。

在循环队中，无法由首尾指针当前值的相对关系来判定一个队是满的还是空的。解决的方法是，或者另设一个标志（如队列中当前元素的个数）作为判定的依据；或者“牺牲”一个存储位置，即有 n 个存储位置的队，最多允许存 $n-1$ 个元素这时队的满空可由队首尾指针的不同差值得到反映。

八八年和八七年竞赛小学组试题中，有两个程序使用的数据结构涉及数组首尾的逻辑相接，但未包含队列的典型操作——出队和入队。一道题要求在首尾环接的十个数中，找到相邻三个数之和的最大值。另一道题要求模拟20个人围坐一圈，给出每隔5人出列的序号序列。假设 N 个元素的存储序号为 0 到 $n-1$ ，那么从第1个元素位置前进1个位置后的位置序号为 $i_1 = (i+1) \bmod N$ 。根据这个访问关系就不难给出题目的算法，如果在后一题中，要求决定的序号序列要顺序地存储在原来的数组空间中，就会涉及入队操作。挑出当前出列元素的序号之后，就要把它之前尚未挑出的元素依次搬动一个存储位置，然后把挑出元素加入出列的队尾。这时在挑选出列元素时要跳过已排队的元素，设当前队尾指针值为 t ，即队中已有 $t+1$ 个已出列的元素，因此从出列位置 i 找下一个出列元素位置时如要跨越最大序号，则 $i_1 = (i+5) \bmod 20+t+1$ ；否则， $i_1 = i+5$ 即指示下一个出列元素的位置。

(4) 串

元素是字符串的线性表称为串。对串的编辑加工是常见的应用。八八年复赛第六题就是一道编辑串的题目。题目要求编写一个通用编辑程序，将从键盘上输入的句子，按一定的编辑格式打印出来。编辑格式是：每行从第 M 个位置开始打印，行宽为 N 个字符，每个单词不得跨行，为使每行末尾准确地落在第 N 个字符处，单词间可增加空格。在BASIC中利用字符串变量的定义以及字符串操作函数不难完成题目要求，如果在其它语言环境中，可能需要决定表达一个串所使用的存储结构。

(5) 链表

线性表的存储结构分为两类。一类是顺序分配结构，元素在存储空间上的物理排列次序和它们的逻辑顺序一致。这时容易找到一个公式，从元素的序号计算出元素的存储地址。因此顺序分配的线性表支持对表元素的随机访问而不一定要求按元素序号顺序访问线性表。但顺序分配要求预先确定空间，不利于元素的中间插入或删除操作，因此具有静态特征。显然，数组很适合采用顺序分配的存储结构，动态性较弱的栈和队也可以考虑，以简化出入操作和提高效率。线性表的另一类存储结构称为链表，它着眼于保证线性表的动态特性，即表的逻辑结构特征是固定的，但表的规模是动态可变的。因此，链表很适合表达结点增删插入频繁的线性表。

链表的结构要点是：结点的存储位置任意，不要求连续分配；结点间的顺序关系由每个结点的链域指向其直接后继或/和直接前趋结点来表示；用头指针来确定链表中第一个结点的存储位置；最后一个结点的链域设置特殊标记值，记为“空”。可见链表结点的有序性是体现在结点的结构当中的，因此链表便于结点的插入和删除，但增加了链域的空间

开销，而且只能从链首开始，顺序地依次访问每一个链表结点。

按照不同算法的需要，可以设计不同的链表结构。

·向前链表。每个结点只包含一个链域，其值指向其直接后继结点。

·循环链表。把向前链表最后一个结点链域的“空”值改为指向头结点的指针值。

·双向链表。每个结点设两个链域，一个指向直接后继，另一个指向直接前驱。

·双向循环链表。双向链表中，末结点的向前链域指向头结点，头结点的向后链域指向末结点。

·广义链表。至少有一个结点本身是链表（子表）的向前链表，而且子表的结点仍然可以再包含子表。广义链表适于表达带共享因素或者递归因素的线性表复杂结构。

·多重链表。每个结点有两个以上的链域，主要用来表示非线性结构，例如树和图。

在不同的语言环境中，链表的实现受语言机构的制约。在PASCAL中，结点结构用记录类型来定义，各数据类型按实际情况选择，而指针域的类型必须为用于结点记录类型的指针类型；结点存储空间通过标准的NEW过程来动态分配，每调用一次就获得一个结点的存储空间，可能还会提供一个非标准的回收过程，来动态收回删除结点的存储空间。

而在BASIC当中，只能在逻辑上用若干个数组中具有相同下标值的元素来组成一个结点。整个链表需要的空间需预先估计，并定义相应的数组空间。指针域应为整型量，其值就是指向结点所在的数组元素的下标值。

至于对链表的操作，包括在链表中插入一个结点，删除一个结点，沿链搜索一个结点等等，都是由程序语句的动作逻辑来体现的。

这里是个例子。设链表结点包含一个数据域一个指针域，链表空间由数组 V 和数组 P 充当，数组元素 $V(1)$ 和 $P(1)$ 组成一个结点 1 。已知存储空间上有两个链表，表 L 是数据结点链表，其头指针由整型变量 LP 充当；表 E 是由尚未分配使用的空结点组成的链表，其头指针由 CP 充当。各指针域的空值用零来表示，因此每个数组的序号从 1 到 n 。

在表 L 中位于 m ($m \neq 0$) 处的结点之后插入一个 u 结点的动作序列是：

If $cp=0$ then已不存在未使用的空结点，故无法得到 u 结点需要的空间；

else

(1) $I_1 = CP$; $CP_1 = P(I_1)$; $V(I_1) = u$;

(2) $P(I_1) = P(m)$; $P(m)_1 = 1$

动作序列(1)表示从表 E 中“脱下”第一个可使用的空结点，把数据 u 送入结点数据域中。注意这个结点的引用始终是通过指针 I 来完成的， I 的值就是 u 结点所在数组元素位置的下标值。

动作序列(2)表示把 u 结点“挂上”表 L 的指定位置。把位于 m 处结点的指针值送入 u 结点的链域中，即把原指定结点的后继变为 u 结点的后继；再把指针 I 的值送入原指定结点的

链域，即把它变为u结点的前趋，这就是u结点插入链表L指定位置的动。注意这两个动作不能颠倒（为什么？）。

如果要在表L中查找出u结点，动作可表示为：

- ```
(1) I1 = LP;
(2) If I = 0 then 查找失败，表L中无u结点;
(3) If V(I) = u then 查找成功;
 else I1 = P(I); GOTO (2)
```

对链表的访问只能从第一个结点开始顺序进行，引用的结点要用指向其存储位置的指针I来表示，所以I<sub>1</sub> = P(I)就表示了从I结点沿链前进到它的直接后继结点的动作。

类似地，从至少有两个结点的表L中删除u结点的后继结点，并把u结点归还到空结点链表E的动作序列是：

- ```
(1) 查找u结点；其位置值置入指针I1中；
(2) J1 = P(I1); P(I1) = P(J1);
(3) P(J1) = CP; CP1 = J
```

在向前链表里，删除一个结点需要两个指针，一个指向被删结点（J指针），另一个指向被删结点的前趋（I指针）。P(I₁) = P(J₁)就实现了删除动作，删除之后，该结点的位置仍保存在J指针中。有兴趣的读者可考虑一下在双向链表中删除结点需要几个指针？删除动作又如何实现？

至此我们已经从逻辑结构和存储结构两个不同的层次上叙述了线性表的一些结构要点，可以用一个简单的表来表示我们提到的一些结构的关系。

逻辑结构		线性表：数组 栈 队 串 广义表				
存储结构	顺序分配	向量	↓	↓	↓	↓
	链式分配	链表	↓	↓	↓	↓

四、树

第二类常用的动态数据结构称为树。树结构的严格定义是用递归方式给出的。树是由一个结点或多个结点组成的有限集合，其中有一个特定的称为根的结点，其余的结点分为m个（m≥0）互不相交的有限集合，每个集合又是一棵树，并称为根的子树。理解这个定义，应把握树结构的非线性特征，这就是一棵树中有且仅有一个根结点；每一个树结点可以有零个或多个与之直接关联的下层结点一称之为孩子；但除根结点之外的任意结点有且只有一个直接与之关联的上层结点一称之为双亲。

通常，采用多重链表来充当树的存储结构，每一个链表结点用作一个树结点，结点内的每一个链域指向一个孩子结点的存储位置，它们也就是各个子树的根。由于各结点的孩子个数不确定时，引起结点的链域数亦难以确定，因此人们更关心一类特殊的树结构，称之为二叉树。

二叉树是n个结点的有限集合。集合或者为空集，或者由一个根结点加上两棵互不相交的二叉树构成，分别称为根的左子树和右子树。就是说，二叉树的每一个结点可以没有小孩，可以有一个小孩（左或右），至多可以有两个小孩（左和右）。

由于树结构的递归性，因此在树结构上操作的算法也用递归方式给出往往会比较简便。例如按一定规律访问二叉树的

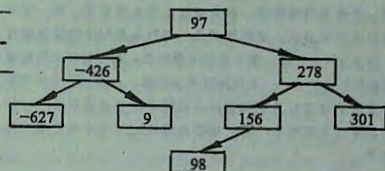
每个结点，每个结点只被访问一次的操作称为二叉树的遍历。如果规定先遍历左子树，然后访问根，再遍历右子树，则称为中序遍历。注意到遍历子树时，子树又有左/右子树和它们的根，因此算法应具有递归的特征。这个算法可以表示为：

- ```
Inorder (P) { 形式参数P是指针，初值应指向根 }
If P的值不为空 then
(1) Inorder (P指结点的左小孩链域);
(2) 访问 (P指结点的数域);
(3) Inorder (P指结点的右小孩链域);
```

(1)表示的递归调用过程表示先对指针P所指的树结点的左子树进行遍历，然后访问它的根(2)，再转向右子树(3)。这样简明的算法如果在不支持递归调用的语言中实现，就会复杂一些，而且需要用一个栈来保存各个子树根结点的存储位置，因为从根开始经过各子树根先访问左子树后，需要再经过子树的根转向右子树。

作为例子，看看二叉排序树的问题。设R={R<sub>1</sub>, R<sub>2</sub>, ..., R<sub>3</sub>}为一个线性序列，则R的二叉排序树是按如下规则构造的二叉树：令R<sub>1</sub>为二叉树的根；若R<sub>2</sub><R<sub>1</sub>，令R<sub>2</sub>作为R<sub>1</sub>左子树的根，否则为右子树的根；对于R<sub>3</sub>到R<sub>n</sub>递归地重复上一个步骤。

因此，序列97，-426，278，9，-627，301，156，98所对应的二叉排序树是



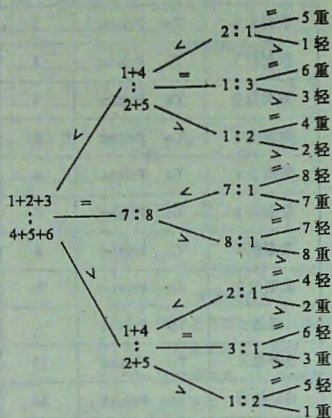
二叉排序树的特点是，用中序遍历树结点时，将得到R序列中诸元素的一个升序序列。在上例中就是-627，-426，9，97，98，156，278，301。可见，构造序列的二叉排序树的过程也就是对序列排序的过程，这正是这种树结构命名的原因。

在PASCAL这样的语言环境中，把序列中的一个元素挂到二叉排序树的过程，可以用一个简练的递归过程来表达。

```
add (int, t) (t为挂入结点的值，t为树结点指针，初值为空)
begin
 if t的值为空
 then
 (1) new (t);
 (2) t1.VALUE1 = int;
 t1.Leftchild = 空值;
 t1.Rightchild = 空值;
 else
 if int < t1.VALUE
 then add (int, t1.Leftchild)
 ELSE add (int, t1.Rightchild)
end;
```

算法中, new(t) 的使用最值得注意的是, 一方面它表示动态取得二叉树的一个结点空间, 同时又表示了把这个结点挂上树的动作, 即把它的存储位置的值置入其双亲结点的左小孩链域或者右小孩链域当中。

当然树结构在程序设计中的应用是各种各样的。八八年的复赛题四要求用天平称重三次, 挑出八个球中的一个特别球, 并指出它是重球还是轻球。显然, 可能的结果一共有16种(每一个编号的球都可能是轻球或重球)。全部可能的判定过程恰好构成一棵树, 称之为判定树。



由这棵判定树可见, 16种可能结果的任何一种的判定, 都可以通过适当的三次称重的结果来推断。

#### 五、图

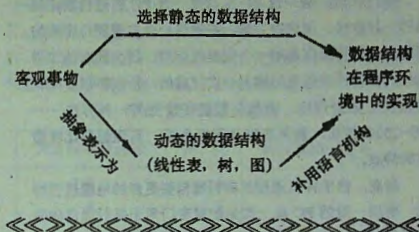
第三类动态数据结构称为图, 一个图是由一个顶点的非空有穷集合和一个边的有穷集合所组成, 所谓边是顶点的有序对或无序对。可见图的顶点就是抽象后的数据结点, 而边则体现了结点和结点之间的联系。因此, 图是由互相间具有任意联系的结点所组成的数据结构, 线性表和树都是图的特例。

由于图的结构任意性, 使得它成为很多算法问题中使用的数据结构。但也因为这一点引起某种复杂性。例如, 上面提到的“马”在棋盘上的周游路线问题, 如果把棋盘上的64个格子变成64个顶点, 两个格子之间有马步时, 对应的两个顶点就存在一条无向边。一个图上如果存在一条经过每个顶点一次的封闭路径, 则称之为哈密顿圈。所以, “马”的周游问题实质上就是在有一个64个顶点的无向连通图上寻找哈密顿圈的问题。目前还没发现能迅速判定马步图中是否存在哈密顿圈的方法。似乎用从某一点出发, 不断地探索和回溯, 总可以判定是否存在一个回到出发点的哈密顿圈。但分析表明, 一只“马”在棋盘上的移动组合数达  $9 \cdot 16 \times 10^{43}$  步,

以每秒走  $10^8$  步来计算, 亦要  $2 \cdot 9 \times 10^{27}$  年来穷尽可能的马步。因此人们使用其它的算法来发现马步图中的哈密顿圈, 而且成功地找到许多个。

#### 六、小结

作为一门学科, 数据结构涉及的概念、性质, 特别是许多相关的算法不可能在一篇文章中得到全面的叙述。总之, 数据结构研究的是如何在计算机上表达客观世界处理对象的抽象方法, 也许我们可以把抽象途径简单地概括为:



(接 47 页)

|      |        |           |
|------|--------|-----------|
| F80E | 2000F9 | JSR F900  |
| F811 | E010   | CPX #10   |
| F813 | F0F0   | BEQ START |
| F815 | 4C07F8 | JMP MORE  |

#### 时间延时子程序

|      |        |          |
|------|--------|----------|
| F900 | A9FF   | LDA #FF  |
| F902 | 8D2000 | STA 0020 |
| F905 | A9FF   | LDA #FF  |
| F907 | 8D2100 | STA 0021 |
| F90A | CE2100 | DEC 0021 |
| F90D | D0FB   | BNE      |
| F90F | CE2000 | DEC 0020 |
| F912 | D0F1   | BNE      |
| F914 | 60     | RTS      |

#### 数据

|      |    |
|------|----|
| F820 | 3F |
| F821 | 0G |
| F822 | 5B |
| F823 | 4F |
| F824 | 66 |
| F825 | 6D |
| F826 | 7D |
| F827 | 07 |
| F828 | 7F |
| F829 | 67 |
| F82A | 77 |
| F82B | 7C |
| F82C | 39 |
| F82D | 5E |
| F82E | 79 |
| F82F | 71 |

#### 复位变量

|      |              |
|------|--------------|
| FFFC | 00; 存放程序启动地址 |
| FFFD | F8           |



## IBM PC 机并行通信的实现

信阳陆军学院教务处 杨瑞生

并行通信,是一台 PC 机与另一台 PC 机进行数据传输的一种方法。用这种方法,在发送一个二进制位所需的时间内,能传输信息的一个完整的字节,因为所有的二进制位的传输是在各自的线上一次完成的,所以不仅支持相对较快的数据传输,而且与数据传输的另一种方法——RS-232C 相比,还具有编制程序容易,实现起来比较简单的特点。

然而,由于并行通信比串行通信要更多的电缆进行对话,所以一般的 PC 机上均未配置专门用于并行通信的适配器和相应的接口。笔者经过对 PC 机硬件的反复学习研究后,利用一般 PC 机上均配备的并行打印机适配器及其相应接口,实现了这一功能,现写出,供同行参考。

## 一、PC 机的标准并行接口

单色显示/打印机适配器或者单独的并行打印机适配器,这两者都是 PC 机的标准并行接口,可以用来联接打印机,实现数据的打印机输出。并行打印机适配器与打印机的接口是一个标准的 25 芯 D 型插头座,插座各脚排列及编号如图一所示,其中 1~17 脚为信号脚,信号均为标准 TTL 电平;18~25 脚为信号地脚。

图一 并行打印机适配器插座脚排列及编号

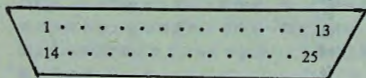


表 1 列出了各信号的名称和方向,其中 12 个脚(1~9 脚和 14、16、17 脚)的信号是由计算机输出到打印机的,5 个脚(10~13 脚和 15 脚)的信号是由打印机回送给计算机的。

计算机,不论是输出数据或输出控制给打印机,还是从打印机输入状态,都不是直接进行的,它需要通过输入输出口 I/O 来完成。并行打印机适配器的 I/O 地址及相应功能如表 2 所示。共有三个 I/O 地址,其中 378H 和 37AH 地址指向打印机,只能 OUT; 379H 地址指向向计算机,只能 IN。

表 1 并行打印适配器接口信号

| 信号名    | 方向          | 适配器插座脚号 |
|--------|-------------|---------|
| 数据位 0  | To Printer  | 2       |
| 数据位 1  | To Printer  | 3       |
| 数据位 2  | To Printer  | 4       |
| 数据位 3  | To Printer  | 5       |
| 数据位 4  | To Printer  | 6       |
| 数据位 5  | To Printer  | 7       |
| 数据位 6  | To Printer  | 8       |
| 数据位 7  | To Printer  | 9       |
| 选通     | To Printer  | 1       |
| 选择输入   | To Printer  | 17      |
| 自动换行   | To Printer  | 14      |
| 打印机初始化 | To Printer  | 16      |
| 肯定     | To Computer | 10      |
| 忙碌     | To Computer | 11      |
| 纸用完    | To Computer | 12      |
| 选择     | To Computer | 13      |
| 错误     | To Computer | 15      |
| 信号地    |             | 18-25   |

表 2 并行打印机适配器的 I/O 地址

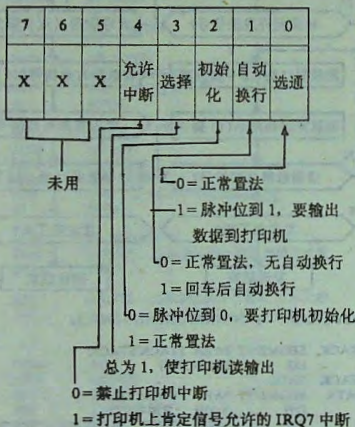
| 功能   | 单色显示器和打印机适配器上的 I/O 地址 | 并行打印机适配器上的 I/O 地址 |
|------|-----------------------|-------------------|
| 输出数据 | 3BCH                  | 378H              |
| 输出控制 | 3BEH                  | 37AH              |
| 输入状态 | 3BDH                  | 379H              |

图二至图四分别说明了三个 I/O 地址中各相应二进制位的含义, 其中除 378H 口的八个二进制位都有效外, 37AH 口只用了 D4~D0 五个二进制位, 379H 口也只用了 D7~D3 五个二进制位。

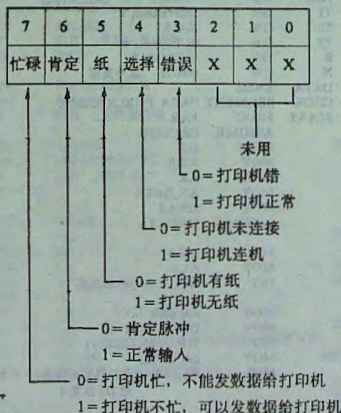
图二 并行打印机适配器输出数据

|     |     |     |     |     |     |     |     |
|-----|-----|-----|-----|-----|-----|-----|-----|
| 7   | 6   | 5   | 4   | 3   | 2   | 1   | 0   |
| 数据  | 数据  | 数据  | 数据  | 数据  | 数据  | 数据  | 数据  |
| 位 7 | 位 6 | 位 5 | 位 4 | 位 3 | 位 2 | 位 1 | 位 0 |

图三 并行打印机适配器输出控制



图四 并行打印机适配器输入状态



## 二、通信线路的连接

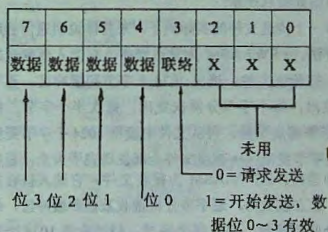
并行打印机适配器接口有 12 个脚的信号是由计算机输出的, 可以满足输出一个字节的需要, 但遗憾的是该适配器只提供了 5 个输入到计算机的信号脚, 不能满足输入一个字节的需要, 为了解决输入信号线不足的矛盾, 我采取了拆字传输的方法, 即将一个字节的八个二进制位“一分为二”, 分两次传送, 每次传送四个二进制位, 接收端收到后再来个“二合一”, 便可完成一个完整字节的传输了。图五是通信线路连接示意图, 5 脚和 15 脚交叉相连, 作为通信双方的联络线; 对于发送端 6~9 脚为发送数据线, 对于接收端 10~13 脚为接收数据线。

图五 通信线路的连接

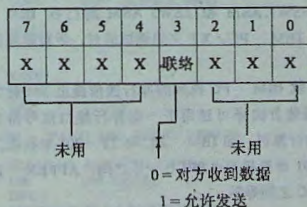
| 发送端插座脚号 | 方向 | 接收端插座脚号 |
|---------|----|---------|
| 5       | →  | 15      |
| 6       | →  | 13      |
| 7       | →  | 12      |
| 8       | →  | 10      |
| 9       | →  | 11      |
| 15      | ←  | 5       |
| 18      | →  | 18      |

当通信线路连接后, 并行打印机适配器 I/O 口中的各个二进制位的含义就如图六和图七所示了。

图六 发送端输出数据 (378H)



发送端输入状态 (379H)





图七 接收端输出数据 (378H)

|   |   |   |   |    |   |   |   |
|---|---|---|---|----|---|---|---|
| 7 | 6 | 5 | 4 | 3  | 2 | 1 | 0 |
| X | X | X | X | 联络 | X | X | X |

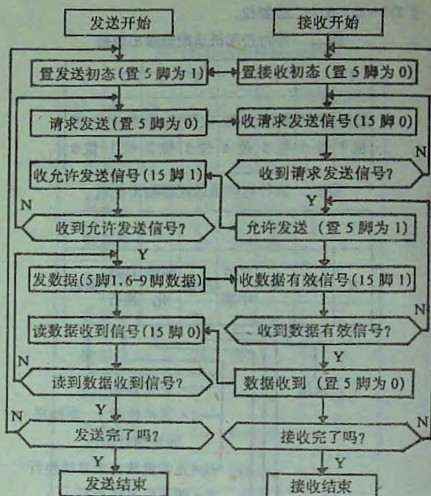
未用  
0 = 收到数据  
1 = 可以接收

接收端输入状态 (379H)

|    |    |    |    |    |   |   |   |
|----|----|----|----|----|---|---|---|
| 7  | 6  | 5  | 4  | 3  | 2 | 1 | 0 |
| 数据 | 数据 | 数据 | 数据 | 联络 | X | X | X |

未用  
0 = 等待接收  
1 = 开始接收, 数据位 0~3 有效

图八 通信程序设计框图



### 三、通信程序的设计

通信程序包括发送和接收两个部分, 发送程序负责完成字节的拆开和发送, 接收程序负责字节的接收和合成, 其设计框图如图八所示;

现举一个传送文件的实际例子, 来具体说明通信程序的设计方法。FSWJ.ASM 为发送程序, 它读入要发送的文件名, 打开该文件, 读出 1024 个字节到缓冲区, 逐字节分次发出, 每个字节分两次发出, 每次半个字节, 待 1024 个字节都发出后, 再从文件中读出 1024 个字节到缓冲区, 逐字节发出……直至文件全部发送完毕为止, 最后关闭文件退出。ZSWJ.ASM 为接收文件, 它读入接收的文件名, 建立该文件, 逐字节分次接收数据到缓冲区, 每个字节分两次接收, 每次半个字节, 待接收到 1024 个字节后, 写文件一次, 再逐字节接收……直至全部文件都接收完毕为止, 最后关闭文件退出。

程序 FSWJ.ASM 和 ZSWJ.ASM 均已在 IBM PC/AT 和 IBM PC/XT 之间调试通过, 运行效果良好。

本文仅就 IBM PC 机间的并行通信提出了一种方法, 这种通信方法还可延用于一切并行接口信号符合 TTL 电平的计算机, 如 IBM PC 与 TP-801 单板机之间, TP-801 单板机与 APPLE II 之间, APPLE II 与 IBM PC 之间等等。

```

;FSWJ.ASM YANG RUI SHENG 1989.19
STACK SEGMENT PARA STACK 'STACK'
DB 256 DUP(0)
STACK ENDS
DATA SEGMENT PARA PUBLIC 'DATA'
C DW 0000H ;发送字节数
D DB 0400H DUP(0) ;发送缓冲区 1024 字节
NA1 DB 0FH,0 ;发送文件名长度
NA2 DB 15 DUP(0) ;发送文件名
T1 DB 'File name: $'
T2 DB 'OK $'
T3 DB 0DH,0AH,'$'
E DB 'ERROR! $'
N DW 0002H ;脉冲宽度
DATA ENDS
CODE SEGMENT PARA PUBLIC 'CODE'
START PROC FAR
ASSUME CS:CODE
PUSH DS
MOV AX,0
PUSH AX
MOV AX,DATA
MOV DS,AX
ASSUME DS:DATA

MOV AH,0
MOV AL,6
INT 10H ;高分辨率屏幕

MOV AH,09H
MOV DX,OFFSET T1
INT 21H ;显示提示 T1
MOV DX,0378H
S0: MOV AL,08H ;并行打印适配器 I/O 输出数据口
 ;的 D3 位置 1

```

|       |      |                          |          |                            |                             |
|-------|------|--------------------------|----------|----------------------------|-----------------------------|
|       | OUT  | DX,AL ;设置发送初态 I          | PUSH     | AX                         |                             |
|       | MOV  | AH,01H                   | CALL     | FS                         | ;发送前四位 D7-D4                |
|       | INT  | 16H ;有键按下否               | POP      | AX                         |                             |
|       | JNZ  | S0 ;无键按下到 S0             | ROL      | AL,1                       |                             |
|       | MOV  | AH,0AH                   | ROL      | AL,1                       |                             |
|       | MOV  | DX,OFFSET NA1            | ROL      | AL,1                       |                             |
|       | INT  | 21H ;读发送文件名              | ROL      | AL,1                       |                             |
|       | MOV  | DI,OFFSET NA1            | CALL     | FS                         | ;发送后四位 D3-D0                |
|       | MOV  | SI,OFFSET NA2            | INC      | DI                         | ;指向下一发送字节                   |
|       | XOR  | AH,AH                    | LOOP     | A                          |                             |
|       | MOV  | AL,[DI+1]                | POP      | DI                         |                             |
|       | ADD  | SI,AX                    | POP      | DX                         |                             |
|       | MOV  | AL,00H                   | POP      | CX                         |                             |
|       | MOV  | [SI],AL ;文件名后的第一个字节置 0   | POP      | BX                         |                             |
|       | MOV  | AH,09H                   | POP      | AX                         |                             |
|       | MOV  | DX,OFFSET T3             | RET      |                            |                             |
|       | INT  | 21H ;显示换行                | ENDP     |                            |                             |
|       | MOV  | AH,0EH                   |          |                            |                             |
|       | MOV  | BX,0000H                 |          |                            |                             |
|       | MOV  | AL,'F'                   |          |                            |                             |
|       | INT  | 10H ;提示 F                |          |                            |                             |
|       |      |                          | PROC     | NEAR                       |                             |
|       |      |                          | PUSH     | AX                         |                             |
|       |      |                          | PUSH     | CX                         |                             |
|       | MOV  | AX,3D00H                 | MOV      | CX,WORD PTR N              | ;脉冲宽度                       |
|       | MOV  | DX,OFFSET NA2            | MOV      | AL,00H                     |                             |
|       | INT  | 21H ;打开发送文件              | MOV      | DX,0378H                   | ;并行打印适配器 I/O 输出数据口的 D3 位置 0 |
|       | JC   | ENDE                     |          |                            |                             |
|       |      |                          | OUT      | DX,AL ;即请求发送               |                             |
|       | MOV  | BX,AX                    | MOV      | DX,0379H                   |                             |
|       | MOV  | AH,3FH                   | IN       | AL,DX                      |                             |
|       | MOV  | CX,0400H                 | TEST     | AL,08H                     | ;检查并行打印适配器 I/O 输入状态口的 D3 位  |
|       | MOV  | DX,OFFSET D              |          |                            |                             |
|       | INT  | 21H ;读 1024 字节到发送缓冲区     | JZ       | FS0                        | ;是否为 1,即允许发送否               |
|       | JC   | ENDE                     | LOOP     | FS1                        |                             |
|       |      |                          | POP      | CX                         |                             |
|       |      |                          | POP      | AX                         |                             |
|       | MOV  | DI,OFFSET C              | RCL      | AL,1                       | ;至此允许发送                     |
|       | MOV  | [DI],AX ;实际读出的字节数,即发送长度  | CMC      |                            |                             |
|       | CALL | FSAX                     | RCR      | AL,1                       | ;四位中的第一位取反                  |
|       |      |                          | AND      | AL,0F0H                    | ;屏蔽 AL 的后四位                 |
|       | CMP  | AX,CX ;实际读出的字节数与 1024 比较 | OR       | AL,08H                     | ;置 D3 位 1,即发送有效             |
|       | JNZ  | ENDZ ;不相等,说明文件已发送完       | MOV      | BL,AL                      |                             |
|       |      |                          |          |                            |                             |
|       |      |                          | PUSH     | CX                         |                             |
|       | PUSH | AX                       | MOV      | CX,WORD PTR N              |                             |
|       | MOV  | AH,02H                   | MOV      | AL,BL                      |                             |
|       | MOV  | DL,''                    | MOV      | DX,0378H                   | ;往并行打印适配器 I/O 数据输出口发数据      |
|       | INT  | 21H ;提示''                |          |                            |                             |
|       | POP  | AX                       |          |                            |                             |
|       |      |                          | OUT      | DX,AL ;D7-D4 位和发送有效信号 D3 位 |                             |
|       | JMP  | SI                       | MOV      | DX,0379H                   |                             |
|       |      |                          | IN       | AL,DX                      |                             |
| ENDZ: | MOV  | AH,3EH                   | TEST     | AL,08H                     | ;检查并行打印适配器 I/O 输入状态口的 D3 位  |
|       | INT  | 21H ;关闭发送文件              |          |                            |                             |
|       | JMP  | END                      |          |                            |                             |
|       |      |                          | JNZ      | FS2                        | ;是否为 0,即是否收到数据              |
|       |      |                          | LOOP     | FS3                        |                             |
| ENDE: | MOV  | DX,OFFSET E              | POP      | CX                         | ;置此数据收到                     |
|       | MOV  | AH,09H                   |          |                            |                             |
|       | INT  | 21H ;文件操作错               |          |                            |                             |
|       |      |                          | PUSH     | CX                         |                             |
|       |      |                          | MOV      | CX,WORD PTR N              |                             |
| END:  | MOV  | AH,09H                   | MOV      | DX,0378H                   |                             |
|       | MOV  | DX,OFFSET T2             | MOV      | AL,08H                     |                             |
|       | INT  | 21H ;提示 T2               | OUT      | DX,AL                      |                             |
|       | RET  |                          | LOOP     | FS4                        | ;重新置发送初态                    |
|       |      |                          | POP      | CX                         |                             |
|       |      |                          | RET      |                            |                             |
| FSAX  | PROC | NEAR                     | ENDP     |                            |                             |
|       | PUSH | AX                       |          |                            |                             |
|       | PUSH | BX                       |          |                            |                             |
|       | PUSH | CX                       |          |                            |                             |
|       | PUSH | DX                       |          |                            |                             |
|       | PUSH | DI                       |          |                            |                             |
|       | MOV  | CX,AX ;发送长度              |          |                            |                             |
|       | INC  | CX                       |          |                            |                             |
|       | INC  | CX                       |          |                            |                             |
|       | INC  | CX                       |          |                            |                             |
|       | XOR  | AH,AH                    |          |                            |                             |
| A:    | MOV  | AL,[DI] ;读发送字节到 AL       |          |                            |                             |
|       |      |                          | START    | ENDP                       |                             |
|       |      |                          | CODE     | ENDS                       |                             |
|       |      |                          |          | END                        | START                       |
|       |      |                          | ZSWJ.ASM | YANG RUI HENG              | 1989.6.19                   |
|       |      |                          | STACK    | SEGMENT PARA STACK         | 'STACK'                     |
|       |      |                          | DB       | 256 DUP(0)                 |                             |
|       |      |                          | STACK    | ENDS                       |                             |
|       |      |                          | DATA     | SEGMENT PARA PUBLIC        | 'DATA'                      |



```

C DW 0000H ;接收字节数
D DB 0400H DUP(0) ;接收缓冲区 1024 字节
NA1 DB 0FH,0 ;接收文件名长度
NA2 DB 15 DUP (0) ;接收文件名
T1 DB 'File name: $'
T2 DB 'OK $'
T3 DB '0DH,0AH,$'
E DB 'ERROR!$'
N DW 0002H ;脉冲宽度
DATA ENDS
CODE SEGMENT PARA PUBLIC 'CODE'
START PROC FAR
 ASSUME CS:CODE
 PUSH DS
 MOV AX,0
 PUSH AX
 MOV AX,DATA
 MOV DS,AX
 ASSUME DS:DATA

 MOV AH,0
 MOV AL,6
 INT 10H ;高分辨率屏幕

 MOV AH,09H
 MOV DX,OFFSET T1
 INT 21H ;提示 T1
S0: MOV DX,0378H
 MOV AL,00H ;并行打印适配器 I/O 输出数据口
 ;的 D3 位置 0
 OUT DX,AL ;即置接收初态 0
 MOV AH,01H
 INT 16H ;有键按下否
 JNZ S0 ;无键按下到 S0
 MOV AH,0AH
 MOV DX,OFFSET NA1
 INT 21H ;接收文件名
 MOV DI,OFFSET NA1
 MOV SI,OFFSET NA2
 XOR AH,AH
 MOV AL,[DI+1]
 ADD SI,AX
 MOV AL,00H
 MOV [SI],AL ;文件名后的第一个字节置 0
 MOV AH,09H
 MOV DX,OFFSET T3
 INT 21H ;显示换行
 MOV AH,0EH
 MOV BX,0000H
 MOV AL,'S'
 INT 10H ;提示 S

 MOV AX,3C00H
 MOV CX,0000H
 MOV DX,OFFSET NA2
 INT 21H ;建立接收文件
 JC ENDE

SI: MOV BX,AX
 CALL ZSAXQ
 MOV AH,40H
 MOV DI,OFFSET C
 MOV CX,[DI]
 MOV DX,OFFSET D
 INT 21H ;接收缓冲区写盘
 JC ENDE

 CMP AX,0400H ;实际写入的字节数与 1024 比较
 JNZ ENDE ;不相等,说明文件已接收完

 MOV AH,02H
 DL,' '
 INT 21H ;提示'

 JMP SI

```

```

ENDZ: MOV AH,3EH
 INT 21H
 JMP END ;关闭接收文件

ENDE: MOV DX,OFFSET E
 MOV AH,09H
 INT 21H ;文件操作错

END: MOV AH,09H
 MOV DX,OFFSET T2
 INT 21H ;提示 T2
 RET

ZSAXQ PROC NEAR
 PUSH AX
 PUSH BX
 PUSH CX
 PUSH DX
 PUSH DI
 MOV DI,OFFSET C
 MOV CX,0002H ;先接收 2 个字节,即长度
 ZSAX ;接收规定长度个字节
 MOV CX,[DI-2] ;收到的长度
 CMP CX,0000H
 JZ ZSAXQE ;长度等于零,说明接收的文件完
 ZSAX ;接收规定长度个字节
 CALL DI
 POP DX
 POP CX
 POP BX
 POP AX
 RET
ZSAXQ ENDP

ZSAX PROC NEAR
A: CALL ZS ;接收前四位
 AND AL,0F0H ;屏蔽后四位
 PUSH AX
 CALL ZS ;接收后四位
 AND AL,0F0H
 ROR AL,1
 ROR AL,1
 ROR AL,1
 POP BX
 OR AL,BL ;合成一个完整字节
 MOV [DI],AL
 INC DI
 LOOP A
 RET
ZSAX ENDP

ZS PROC NEAR
 PUSH CX
 MOV CX,WORD PTR N ;脉冲宽度
 MOV AL,00H
 MOV DX,0378H ;并行打印适配器 I/O 输出数据
 ;口的 D3 位置 0
 OUT DX,AL ;即请求接收
 MOV DX,0379H
 IN AL,DX
 TEST AL,08H ;检查并行打印适配器 I/O 输入状
 ;态口的 D3 位
 JNZ ZS0 ;是否为 0,即允许接收否
 LOOP ZS1
 POP CX ;至此允许接收

 PUSH CX
 MOV CX,WORD PTR N
 MOV AL,08H
 MOV DX,0378H ;并行打印适配器 I/O 输出数据
 ;口的 D3 位置 1
 OUT DX,AL ;即开始接收
 MOV DX,0379H
 IN AL,DX
 TEST AL,08H ;检查并行打印适配器 I/O 输入状
 ;态口的 D3 位

```

# 在屏幕上同时显示三种字体的方法

上海挤出机械厂 翁晓刚

我们知道, CCDOS在640×200分辨率的屏幕上只能显示16×16点阵的汉字。为了显示较大的汉字,当然可以专门设计一段程序对汉字进行放大处理,但放大过程往往较复杂,不仅放大程序较长,而且处理速度也较慢。本文向大家介绍一种尽量利用BASIC提供的功能来读取放大字型的方法,该方法可在640×200分辨率的屏幕上同时显示16×16、32×16、32×32三种点阵的字体,其中16×16是原字型,32×16是横扩型,32×32是纵横扩字型。利用这三种字型可使屏幕显示字型富于变化,构成较美观的软件篇头。

## 一、32×16字型的获取及显示

在BASIC中用SCREEN命令可很方便地转换屏幕工作方式。命令SCREEN 1可使屏幕处于320×200中分辨率工作方式。此时显示的汉字即为32×16点阵字型。我们可在此状态下先用OUT & H3D8:2 (或OUT & H3D8:6)命令来关闭屏幕显示,但工作方式不变。然后用PRINT命令向屏幕缓冲区写上所需的横扩汉字,再用GET命令把屏幕缓冲区的横扩汉字图形吸取到一个数组中,随后用SCREEN 2命令把屏幕转回高分辨率工作方式,此时只要用PUT命令把数组内容投影到屏幕上指定的区域,就可在高分辨方式下显示横扩字型了。

## 二、32×32字型的获取及显示

先用上述方法获得32×16点阵的汉字图形,并存入一

整形数组中。然后对该数组图形进行纵扩一倍,存入另一个大一倍的数组中。为了对存于数组中的图形进行纵扩,首先要了解屏幕信息存于数组中的格式:数组的最前2字节给出图形以位表示的水平长度,其后2字节给出图形以位表示的垂直长度,再后面就是图形第一象水平线的数据,在一条水平线数据存完后再存下一条水平线数据,直至结束。因此要将图形纵扩一倍,放大的数组的最前2字节内容应与原数组对应值相同,其后2字节值应为原数组对应值的2倍。接下去应对原数组每一条水平线的值重复存放2次,这样处理完后在放大的数组中就得到了32×32点阵的汉字图形了。之后可用PUT语句在高分辨方式下进行显示。

## 三、一个显示三种字体的实例

下面给出一个显示三种字体的软件篇头实例,供读者编程参考。对该程序简单说明一下。

语句20定义的三个整形数组用于存放屏幕图形。数组A%存放32×16点阵汉字串,数组C%存放32×32点阵汉字串,B%是过渡数组。数组大小的确定可参照BASIC中GET语句的说明。语句40用后台方式演奏一段音乐。语句90~120将数组B%中的32×16字型纵扩一倍后存于数组C%。这四句的循环变量值必须计算正确。语句100的循环步长应等于图形水平长度(以字节为单位),这里显示的字符串为22字节,故取22。循环终值应略小于B%的最大下标,但不能小于一个步长,这里取小1。语句110中J循环的终值应为I+(I循环的步长-1)。程序用PC BASIC A 2.0版通过。

```

JZ ZS2 :是否为1,即是否已经收到数据
LOOP ZS3
POP CX :置此收到数据,D7-D4位

PUSH AX
PUSH CX
MOV CX,WORD PTR N
ZS4: MOV DX,0378H
 MOV AL,00H
 OUT DX,AL
LOOP ZS4 :重新置接收初态
POP CX
POP AX
RET
ENDP

ZS
START ENDP
CODE ENDS
 END
 START

```

```

10 REM 显示三种字体表演程序'XSSZZT. BAS'
20 DIM A%(220),B%(350),C%(700)
30 KEY OFF, SCREEN 1: OUT & H3D8:2
40 PLAY "MBT20002G03CDDDE. F12GFEEDDC2"
50 PRINT "上海挤出机械厂"
60 GET (0,0)-(111,15),A%
70 PRINT "生产管理系统"
80 GET (0,18)-(175,33),B%
90 C%(0)=B%(0),C%(1)=B%(1)*2,K=2
100 FOR I=2 TO 353 STEP 22
110 FOR M=1 TO 4: FOR J=1 TO 1+21
120 C%(K)=B%(J),K=K+1: NEXT J,M,1
130 SCREEN 2: PUT (210,40),A%,PUT (143,70),C%
140 LOCATE 8,45: PRINT "1986年8月开发"
150 LINE (100,20)-(540,150),,B
160 LINE (80,10)-(560,160),,B

```



# 动手设计一个 6502 单片计算机

周仲玉

## 一、引言

本文主要讲述了如何设计一个 6502 单板机电路及有关程序，并介绍了实现办法，该文章主要是为那些从事微型计算机教学任务的人而准备的，不过根据实践证明，本文对那些刚刚熟悉计算机和为了巩固学到的微计算机知识而希望学到动手技巧的人来说，也是很有用的。另外，一些微机业余爱好者和学习微机基本课程的学生也对本文深感兴趣。因为设计并完成一个单板机，不仅能增加人们对计算机理论的感性认识，而且单板机本身还能作为一个开发产品的计算机来使用。

## 二、设计要点

设计一个单板机系统的第一步是画出系统框图，单板机系统框图 1 通常如下所示：

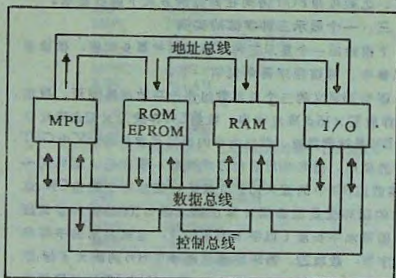


图 1

框图中列出了单板计算机 (SBC) 的基本组成部分，其中各部采用的具体型号是：

MPU——6502 NMOS

EPROM——2516 (2K 字节) NMOS

RAM——5516 (2K 字节) CMOS

I/O——6522 NMOS

然后开始设计电路框图，将各芯片所有对应的数据总线和地址总线一一连接起来，形成 SBC 的数据和地址总线。接着把注意力集中到 MPU 的控制总线，6502 需要注意的控制总线有以下几条：

R/W——与其它芯片的 R/W 信号线相连。

IRQ——与 6522 的 IRQ 相连。

RES——与 6522 的 RES 相连。

RDY——如该信号为低电平，将停止处理机工作，RDY 通过一个 10KΩ 电阻与 Vcc (+5V) 相连。

NMI——优先级中断链，一般用于电源故障，它通过 10KΩ 电阻与 Vcc 相连。

SYNC——这是一根输出信号，平时处于浮空状态。

SO——平常不使用，通过 10KΩ 电阻与 Vcc 相连。

注意，IRQ 同时通过电阻与 Vcc 相连，因为平时要保证该信号为高电平，当 I/O 外设产生中断请求时，该信号为低电平。

## 电路

图 2 是 6502 SBC 的电路图：

其中有几个部分介绍一下。

### a. 时钟电路

6502 的 39 脚是时钟信号，它是通过将晶振与 Q<sub>0</sub> 和 Q<sub>2</sub> 并联连接后才产生的，然后经过两个史密特反向器和一个 10KΩ 电阻，将信号整形。

### b. 复位电路

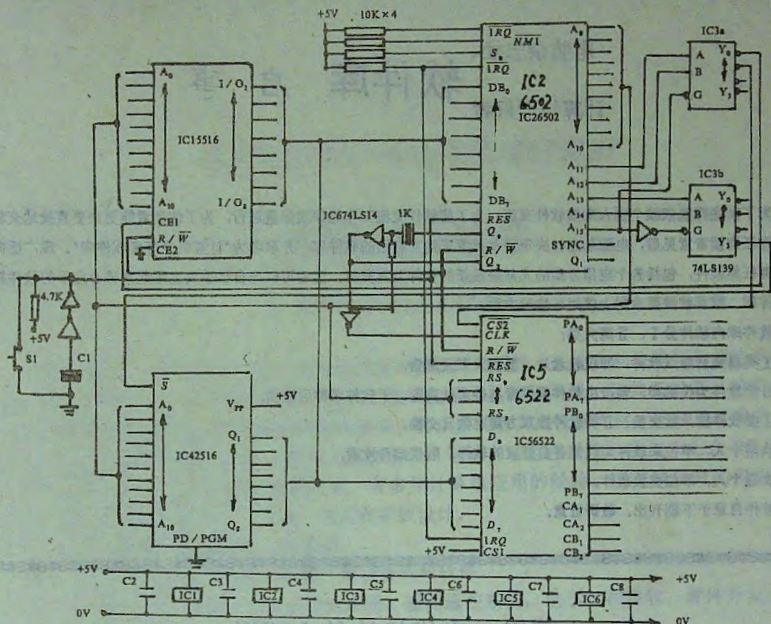
当 S<sub>1</sub> 按钮按下时，复位信号为低电平，由两个史密特反向器对信号进行整形，当给 SBC 加上电源时，R<sub>1</sub> 和 C<sub>1</sub> 能产生自动复位信号。

### C. 地址译码电路

这儿使用了一个 74LS139 译码器，其中电路图中的译码器 (IC3a) 用来对存储器地址的后半部分进行译码 (如 00~7F)，而另一个译码器 (IC3b) 通过一个反向器，用来对存储器地址的前半部分进行译码 (如 80~FF)。IC3a 给 RAM 和 I/O 设备提供芯片选信号，而 IC3b 给 EPROM 提供片选信号，其中对 2516 EPROM 编程时要使用顶端的 2K 内存，本 SBC 的内存空间映射图如图 3 所示：

## 三、实现

在动手实现时，可以采用带有 IC 插座的微处理机开发板 (RS345—434)，这儿强调要采用插座而不是直接将芯片焊在板子上。这是因为：不正确的连线很容易损坏芯片，而焊好的芯片要取下来，又是很困难的；另外在板子完成前，可以不去考虑芯片，以便在构造单板机时避免芯片损坏，而芯片也可以在以后重复使用，最后，最重要的一点是在板子构造完后，要采用测试设备对所有的电路进行连线正确性检查，确保没有短路。



还要将每个 IC 的电源信号复接一个  $0.1\mu\text{F}$  电容。当板子搭好并进行检查后，先将处理器和六脚反向器插上，加上电源，进行两个比较简单的测试：是否有  $1\text{MHz}$  的时钟信号和当复位按钮按下时是否 RES 为低电平。接着对 EPROM 进行编程，表 1 提供的程序能在七段 LED 显示从 0 到 F 的字符。当所有芯片插好后，可将七段 LED 与 6522A 口相连（见下图 4）相连，中间要加一个驱动电路。当 SBC 加上电源，按复位钮后，SBC 会在七段 LED 自动显示 0~F 字符。

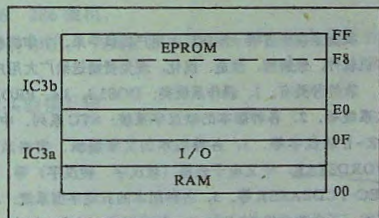


图 3

喪 1

PROGRAM TO INCREMENT A 7-SEGMENT DISPLAY FROM 0 TO F

| 地址   | 代码          | 指令         |
|------|-------------|------------|
| 主程序  |             |            |
| F800 | A9FF        | LDA #FF    |
| F802 | BD030B      | STA 0803   |
| F805 | A200 START  | LDX #00    |
| F807 | BD20F8 MORE | LDA F820,X |
| F80A | 8D010B      | STA 0800   |
| F80D | E8          | INX        |

(转 39 页)

图 4



电脑杂志社

计算机爱好者

# 软件库 启事

为了彻底摆脱依赖别人来搞软件交流，为了使软件交流工作能保质保量进行，为了使读者能更快更直接地交流软件，经征询读者意见后，电脑杂志社决定发动读者来建立自己的软件库，并取名为“计算机爱好者软件库”。现广泛向各地读者征集软件，包括各个应用方面的尤其是最新的软件及其资料。读者可以把自己手头上用起来得心应手的软件推荐给软件库。推荐软件要求附上使用环境和方法。

软件库内软件分 I、II 两大类：

I 类属免费赠送软件，即仅收盘片、复制人工及邮费。

II 类软件为代销类，售价由软件提供者与杂志社商定，于软件名称后注明。

I 类软件按片数交换，II 类软件按双方商定数目交换。

欢迎个人、单位来软件库代销各自开发的软件。版权归开发者。

欢迎个人、单位交换软件。

软件目录于下期刊出。敬请留意。

## 苹果、中华机软件大赠送

我是湖南省唯一能向广大用户提供苹果、中华机各类软件的教学单位。收集了大量国内、外最新版本的苹果、中华机软件，经解密、改造、汉化，现免费赠送给广大用户，以推动苹果、中华机的普及和应用。

软件种类有：1、操作系统类：DOS3.3, 3.4, PRODOS 系列，CEC-I 扩展系统，汉卡超级 CCDOS，汉卡拼音输入系统等。2、各种版本的软汉字系统：STC 系列，中华、高级软汉字、星河软汉字，多功能软汉字，快速软汉字，GZ-II 软汉字等。3、各种版本的文章编辑、表格系统：APPLE 软汉字、硬汉卡 WORDSTAR，中华机硬汉卡 WORDSTAR，中文电子表格（软汉字、硬汉字）等。4、中文数据库系统：APPLE DBASE II（软汉字、硬汉卡），CEC-I CDBASE II 等。5、各种版本的五笔字型系统：APPLE II 五笔字型（软汉字、硬汉卡），CEC-I 五笔字型输入系统（不必改主机 IC 块）。6、各种拷贝、加密、解密、磁盘管理、修复、编辑、APPLE、CEC-I 硬件检测等工具软件，绘图软件。7、各种辅助教学工具软件、教学、应用、游戏软件。

软件每片 10 元，个人用户每片 9.5 元，邮寄包装费每次 3 元，款到一星期内软件付邮。有正式税务发票，需要者请注明。

本室还向用户优惠提供以下硬件：APPLE II、CEC-I、IBM PC/XT 286、386 系列、ZH-2 型联想式拼音词汉卡、CEC-I 驱动器一转二接口、CEC-I 游戏棒、优质双面双密空白磁盘（有彩色封套的）。

需要软件或硬件者请来函联系索取软件目录和硬件报价单。

联系地址：长沙重型机器厂厂长重中学计算机室 联系人：单振宇 黄运莲 邮政编码：410014 电话：34211 转学校  
银行开户：长沙长重学校生活文化用品服务部 帐号：060006603241 开户行：长沙市工行树处