

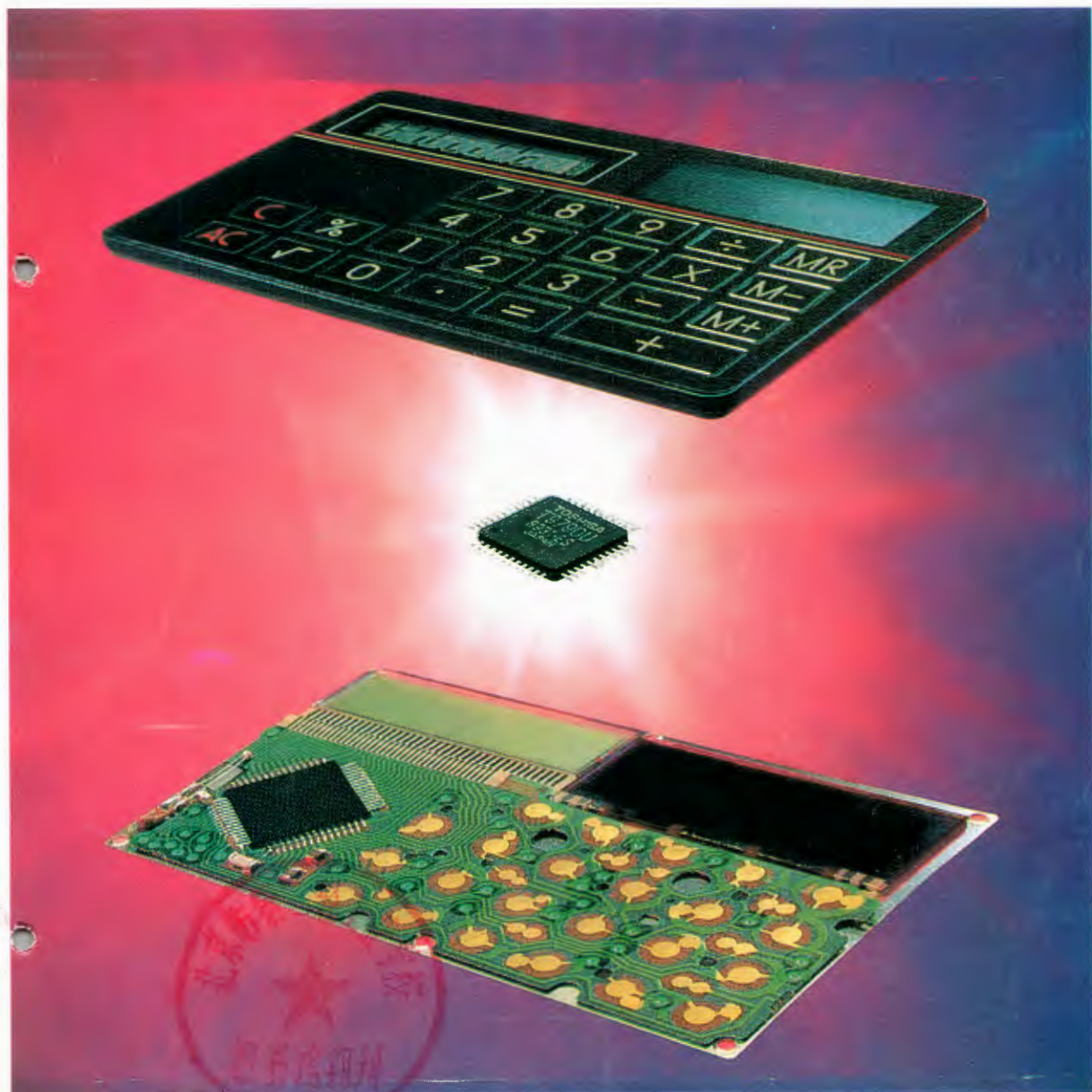


電子

ISSN 1000-1077

744

與 電腦



• ELECTRONICS AND COMPUTERS •

家庭电脑教学键盘+BS卡 使游戏机变成了家用电脑



四大功能：

- 可进行F B A S I C语言程序设计。
- 输入曲谱即可自动演奏的电子琴功能。
- 存储有104个动画图案，有丰富的绘图功能。
- 具有计算器功能，可直接进行四则运算。
- 英文打字练习功能。

K B 200电脑教学键盘
学习电脑知识的良友

生产厂家：机电部46所新星电子公司
地 址：天津新技术产业园区
电 话：31. 8342
邮 编：300192 天津55号信箱



Z80 的“新”指令

表一, 重复功能指令

指令码	助记符	字节数	指令码	助记符	字节数
ED00		2	ED4C		2
⋮			ED54		2
ED3F		2	ED5C		2
ED70		2	ED64		2
ED71		2	ED6C		2
ED77		2	ED64		2
ED7F		2	ED7C		2
ED80		2	ED55		2
⋮			ED65		2
ED9F		2	ED75		2
EDA4		2	ED5D		2
⋮			ED6D		2
EDA7		2	ED7D		2
EDAC		4	ED3nn		4
⋮			ED6Bnn		4
EDAF		2	ED66		2
EDB4		2	ED76		2
⋮			ED7E		2
EDB7		4	11 011 101		
EDBC		4	11 001 011		
⋮			←+d→		
EDBF		4	01 b XXX		
EDC0		4	11 111 101		
⋮			11 001 011		
EDFF		4	←+d→		
			01 b XXX		

注: 测试位 b 见表七。

表二, 新的传送运算和比较指令

指令码	助记符	字节数	指令码	助记符	字节数
DD 26 n	LD IXH, n	3	DD 24	INC IXH	2
FD 26 n	LD IYH, n	3	DD 2C	INC IXL	2
DD 2E n	LD IXL, n	3	FD 24	INC IYH	2
FD 2E n	LD IYL, n	3	FD 2C	INC IYL	2
11 011 101			DD 25	DEC IXH	2
01 r' 100		2	DD 2D	DEC IXL	2
11 011 101		2	FD 25	DEC IYH	2
01 r' 101		2	FD 2D	DEC IYL	2
11 111 101		2	11 011 101		2
01 r' 100		2	10 x 100		2
11 111 101		2	11 011 101		2
01 r' 101		2	10 x 101		2
11 011 101		2	11 111 101		2
01 100 r'		2	10 x 100		2
11 011 101		2	11 111 101		2
01 101 r'		2	10 x 101		2
11 111 101		2			
01 100 r'		2			
11 111 101		2			
01 101 r'		2			
DD 64		2			
DD 65		2			
DD 6C		2			
DD 6D		2			
FD 64		2			
FD 65		2			
FD 6C		2			
FD 6D		2			

注: 运算符 [x] 与寄存器 r' 见附表。

Z80 的“新”指令

表三、双功能指令

指令码	助记符	字节数
11 111 101 11 001 011 ←---+d---→ 00 S r''	{(S)(IX+d)} {LD r'', (IX+d)}	4
11 111 101 11 001 011 ←---+d---→ 00 S r''	{(S)(IY+d)} {LD r'', (IY+d)}	4
11 111 101 11 001 011 ←---+d---→ 00 b r''	{RES b, (IX+d)} {LD r'', (IX+d)}	4
11 111 101 11 001 011 ←---+d---→ 00 b r''	{RES b, (IY+d)} {LD r'', (IY+d)}	4
11 111 101 11 001 011 ←---+d---→ 11 b r''	{SET b, (IX+d)} {LD r'', (IX+d)}	4
11 111 101 11 001 011 ←---+d---→ 11 b r''	{SET b, (IY+d)} {LD r'', (IY+d)}	4

注:测试位 b, 寄存器 r'' 见表六、表七。

表四 其它新指令

指令码	助记符	字节数
ED4E	{IM OFF}	2
ED6E		2
CB30	SLL B	2
CB30	SLL B	2
CB31	SLL C	2
CB32	SLL D	2
CB33	SLL E	2
CB34	SLL H	2
CB35	SLL L	2
CB36	SLL (HL)	2
CB37	SLL A	2
DDCB+d 36	SLL (IX+d)	4
FDCB+d 36	SLL (IY+d)	4

表五 移位过程

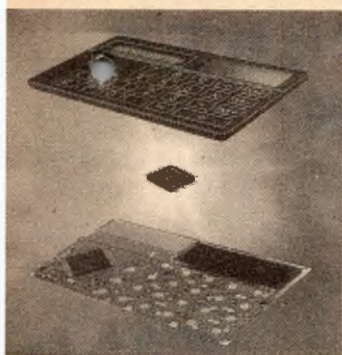
移位指令(s)	S	操作过程
RLC	000	
RRC	001	
RL	010	
RR	011	
SLA	100	
SRA	101	
SLL	110	
SRL	111	

表六

寄存器	r''
B	000
C	001
D	010
E	011
H	100
L	101
...	...
A	111

表七

测试位	b
1	000
2	001
3	010
4	011
5	100
6	101
7	111



ELECTRONICS AND COMPUTERS

一九九一年

总期第78期

电子与电脑

目 录

· 综述 ·

- 参加 91' 第三届国际信息学奥林匹克 (IOI' 91) 竞赛的体会 吴文虎 薛宏熙 (2)
- 众说纷云: 8 位学习机是否已过时?
- 专家认定: 君在教育界仍是主力军! (3)

· PC 用户 ·

- IBM PC/XT 定时中断服务程序设计 张辉 李东升 李奎贤 (5)
- 通用数据库修改程序 周日初 (7)
- PC-DOS 命令 WHEREIS 的改正 李 敏 (8)
- 巧算 1000! 李 帆 (9)
- 又发现一种新病毒 王家春 (10)
- 杨基病毒快速解毒法 李凤明 (10)
- 设备驱动程序 ANSI.SYS 的使用 王立勇 (11)
- 巧用虚拟盘 李焕民 (12)
- 用递归解九连环 钟 武 (12)

· 学习机之友 ·

- 将 CEC-I 中华学习机改制成专用机 姜良夫 (14)
- 用栈结构优化程序 陈合力 (17)
- 寻找无理数 e 的约率和密率 施为钢 (17)
- 怎样节省 CEC-I 存储空间 尹向阳 梁 曦 (18)
- ProDOS 磁盘操作系统入门 (续) 廖 凯 (19)
- 数制转换简法 李 锋 (20)
- 也谈扩展 COPY II PLUS 4.0 关炳坤 (21)
- 从 CP/M 转移文件到 APPLE DOS 下 方福兴 (21)

- 多功能 CEC-I 高速排序 王庆顺 (22)
- 小学生速算训练 李永升 (23)

· C 语言初阶讲座 ·

- 第七讲 结构数据 (二) 李文兵 (24)

· 学用单片机 ·

- MCS-51 单片机应用中二种 LED 显示方法 傅 篱 (27)

· 学装微电脑 ·

- 相位控制部件的制作 易齐干 (29)

· 电脑巧开发 ·

- 怎样用电脑检修录像机 李玉全 (32)
- 用娃娃机开发单片机程序 张 涛 (32)

· 维修经验谈 ·

- TH3070 打印机接口电路工作原理与故障分析 李燕华 (35)
- 长城 0520CH 机打印机适配器故障维修 邱 槟 (37)

· 初级程序员级水平考试辅导讲座 ·

- 汉字 dBASEIII 使用技巧 (下) 王路敬 (39)
- 91 年第三届国际信息学奥林匹克 (IOI' 91) 试题 吴文虎 (46)
- Z80 的“新”指令 章永得 (47)

封一: 薄型计算器的内幕

封二封三: Z80 的“新”指令

封四: 家庭电脑教学键盘

机械电子工业部电子工业出版社主办

编辑、出版:《电子与电脑》编辑部

(北京 173 信箱 邮政编码: 100036)

印刷: 北京三二〇九厂

国内总发行: 北京报刊发行局

国内统一刊号: CN11-2199

邮发代号: 2-888

国外代号: M924

出版日期: 每月 23 日

主编: 王惠民 副主编: 王昌铭

责任编辑: 施玉新

订购处: 全国各地邮电局

国外总发行: 中国国际图书贸易总公司

(北京 399 信箱 邮政编码 100044)

广告经营许可证: 京海工商广字 147 号

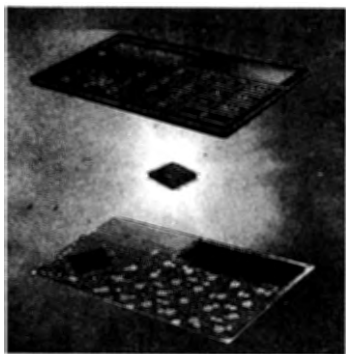
定价: 0.95 元



ISSN 1000-1077

电子与电脑

9744



ELECTRONICS AND COMPUTERS

一九九一年

总期第78期

电子与电脑

目 录

· 综述 ·

- 参加 91' 第三届国际信息学奥林匹克 (IOI'91) 竞赛的体会 吴文虎 薛宏熙 (2)
众说纷云: 8 位学习机是否已过时?
专家认定: 君在教育界仍是主力军! (3)

· PC 用户 ·

- IBM PC/XT 定时中断服务程序设计 张辉 李东升 李奎贤 (5)
通用数据库修改程序 周日初 (7)
PC-DOS 命令 WHEREIS 的改正 李 敏 (8)
巧算 1000! 李 帆 (9)
又发现一种新病毒 王家春 (10)
杨基病毒快速解毒法 李凤明 (10)
设备驱动程序 ANSI.SYS 的使用 王立勇 (11)
巧用虚拟盘 李焕民 (12)
用递归解九连环 钟 武 (12)

· 学习机之友 ·

- 将 CEC-I 中华学习机改制成专用机 姜良夫 (14)
用栈结构优化程序 陈合力 (17)
寻找无理数 e 的约率和密率 施为钢 (17)
怎样节省 CEC-I 存储空间 尹向阳 梁 曦 (18)
ProDOS 磁盘操作系统入门 (续) 廖 凯 (19)
数制转换简法 李 锋 (20)
也谈扩展 COPY II PLUS 4.0 关炳坤 (21)
从 CP/M 转移文件到 APPLE DOS 下 方福兴 (21)

- 多功能 CEC-I 高速排序 王庆顺 (22)
小学生速算训练 李永升 (23)

· C 语言初阶讲座 ·

- 第七讲 结构数据 (二) 李文兵 (24)

· 学用单片机 ·

- MCS-51 单片机应用中二种 LED 显示方法 傅 篱 (27)

· 学装微电脑 ·

- 相位控制部件的制作 易齐干 (29)

· 电脑巧开发 ·

- 怎样用电脑检修录像机 李玉全 (32)
用娃娃机开发单片机程序 张 涛 (32)

· 维修经验谈 ·

- TH3070 打印机接口电路工作原理与故障分析 李燕华 (35)
长城 0520CH 机打印机适配器故障维修 邱 楦 (37)

· 初级程序员级水平考试辅导讲座 ·

- 汉字 dBASEIII 使用技巧 (下) 王路敬 (39)
91 年第三届国际信息学奥林匹克 (IOI'91) 试题 吴文虎 (46)
Z80 的“新”指令 章永得 (47)

封一: 薄型计算器的内幕

封二封三: Z80 的“新”指令

封四: 家庭电脑教学键盘

机械电子工业部电子工业出版社主办

编辑、出版:《电子与电脑》编辑部

(北京 173 信箱 邮政编码: 100036)

印刷: 北京三二〇九厂

国内总发行: 北京报刊发行局

国内统一刊号: CN11-2199

邮发代号: 2-888

国外代号: M924

出版日期: 每月 23 日

主编: 王惠民 副主编: 王昌铭

责任编辑: 施玉新

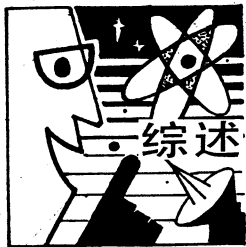
订购处: 全国各地邮电局

国外总发行: 中国国际图书贸易总公司

(北京 399 信箱 邮政编码 100044)

广告经营许可证: 京海工商广字 147 号

定价: 0.95 元



参加 91' 第三届国际信息学奥林匹克 (IOI'91) 竞赛的体会

领队 吴文虎 薛宏熙

今年 5 月 19—25 日在希腊的雅典举行了第三届国际奥林匹克信息学竞赛,有 23 个国家的 74 名选手参赛。由中国科协,国家教委,国家自然科学基金会委托中国计算机学会主办参赛选手选拔,集训和赴希腊参赛全过程。中国队的 3 名选手全部获奖,金牌两块,银牌 1 块,总积分名列第一。

一、普及是基础,组织是关键

学科奥林匹克的宗旨是:促进交流,建立友谊与联系,推动优秀人材的成长。中国队本着这一宗旨,力求反映中国青少年努力学习先进科学知识,奋发向上,全面发展的精神风貌,第三次出国参赛。在希腊的两轮竞赛中水平得到比较正常的发挥,取得了比较好的成绩。评委和各国领队们说:中国队是 IOI 中最强的队。在闭幕发奖会上组委会表扬了中国队,说从解思路,编程能力到调试速度上,中国的三名选手都是非常出色的。

这次取得比较好的成绩不是偶然的,因为我们既有普及基础,又有良好的组织和训练。

从组织上讲,我们有得天独厚的优越条件,那就是党和国家高度重视,计算机的普及从娃娃抓起,有全国性的竞赛活动作为选拔高水平选手的依据。这次去希腊参赛的三名选手是经两次选拔产生的。第一次是全国统一命题的通讯赛。各省(自治区,直辖市)设考场,同日同时开始测试。每个省市的前三名的成绩和答卷(含有解题程序的软盘)报到学会,由学会统一测试,排序,选出参加复试的选手;第二次 11 名选手集中北京,由学会主持参加第三届国际信息学奥林匹克中国队选手的选拔赛(见中国计算机学会通讯(91)第 2 期)。通讯赛的题目和选拔赛题目都有相当难度。从这两次竞赛看 91 年全国青少年计算机竞赛中的高手们较之 90 年又有了比较大的进步,表现在掌握计算机语言,算法知识,数据结构知识,以及解题的能力上。91 年的组队有了比 90 年更为广阔的基础,挑选人才的余地增大了。

二、训练中加强针对性,重在提高能力

在训练的策略思想上强调加强针对性,“对症下药”,有目的有目标地进行培训。一是我们分析了选手的情况;二是预测了赛事情况进行有针对性地准备。

选手的情况是:两名老选手杨澄(清华附中,高

二),杨云和(上海延吉中学,高二)曾参加 IOI'90,分获银牌和铜牌,新选手是夏西远(南京师大附中,高二)。两名老选手可能有的优势是:经历过一次世界级的大赛,见过世面,有了胆识,获得了如何备战的经验;但也可能会背上包袱,脑子里会形成固定的解题模式,临场放不开手脚。这两位选手在上述的全国通讯赛和组队选拔赛就有这方面的表现,比如选拔赛中的第一题,本来可以用“宏”的思想来做,这两名选手在 90 年的中国队培训中曾经学习过这种思路,但偏偏在选拔赛中不敢用这种思路解题,反映了一种套搬公式,求稳而不求新的思想,创造才能发挥不出来。我们首先加强心理素质训练,让选手从零开始,以首次参赛的心态去迎接 IOI'91。要求每一个选手要力争上游,瞄准世界第一的目标;顽强拼搏,发挥最佳的创造才能;勤奋学习,努力实践,掌握真本事,把必胜的信心建立在解题能力的稳步提高上。

为了提高解题能力,采取了四项措施:

1. 引导选手“温故知新”,精做通讯赛和选拔赛中的三道难题,深入钻研,总结更好的思路与方法;
2. 自学一些参考资料,学一些有关图论和组合数学的知识,将一些解题方法上升到理论高度,以起到举一反三的作用。
3. 扩充题型进行练习,以掌握广博的知识和算法;

4. 掌握最新的语言版本,提高编程与调试效率。

为了应付赛场上出现的一些特殊情况,训练与掌握所应采取的对策与思路,譬如在第二届 IOI 上,解第一题普遍出现由于算法的数据量大而溢出的问题,我们从两个方面引导选手:一是开数组时,选用大模式;二是当运行时间或搜索步数超过某个阈值时,转而采取不致出现溢出的“笨方法”。当然,这些方法要结合赛题的实际当场灵活运用。

通过短时间的赛前训练,从总体上看中国队的水平在以下四个方面有所提高:

1. 拼搏与竞争的意识增强了;
2. 解题的思路拓宽了;
3. 编程调试与自测能力提高了;
4. 应变的能力增强了。

三、试题难度加大,训练的视野尚须扩大

这届竞赛第一轮题目比较容易,这道题要求据三

个关系式往 5×5 的矩阵中填入数字,编程输出所有可能的方案。中国的三名选手采用深度优先的思路来解决这道题,程序风格和屏幕效果都很漂亮,三个人都获满分。第二天组委会有意将第二轮赛题难度加大,经仲裁委员会表决后选出的一道题目是用递归方式定义 S 项,S 项是字符 S 和括号组成的序列,对 S 项的子项形式给出了一个缩简规则,要求写出生成,读,显示,缩减等五个过程,而且要求程序能演示每一个过程,最后将上述过程连成一个程序,将生成的 S 项以规格化的形式输出。这道题目涉及到编译原理中的某些内容,要编程的内容多,给的时间短,想不到用什么数据结构就做不出来。中国队的杨澄和夏西远选择二叉树的结构,杨云和选择表达式化简的方法,都做得比较好。杨澄编了 400 行 PASCAL 语句,杨云和编了 420 行 C 语言的语句。每小时平均 100 行。负责对中国队评分的希腊评委看了我们三位小将的程序,翘起了大拇指说:中国的孩子编出的程序,真像是专业人员编的。第二轮赛题的评分有的队进行得很快,中国队拖的时间很长,从下午 2:30 到 6:30。后来才知道谁做得多,自然被测试的项目就多,测得就细,一点一点扣,那真是“一丝不苟”,拿高分绝对不容易。

第二轮竞赛未被选中的题目有的难度可能比这道题还要大。其中的一道是涉及各国文字文本字频统计的,另一道是涉及医院门诊预约的。从题型看涉及的领域广,灵活多变,这从一个侧面反映了计算机的应用渗透到了社会生活的各个领域的发展状况,引导孩子们了解这方面的情况,开阔视野,对于启发思路是会有好处的。

四、仍然需要站到起跑线上

在希腊逗留十天,有幸到奥林匹克的发源地,踏一踏马拉松发源地的起跑线,感慨万千。想到普及计算机教育的事业也应该有锲而不舍的马拉松精神。从体育的奥林匹克到智力的奥林匹克,世界发生了多么大的变化!几届数,理,化,信息学奥林匹克又连连报捷,长了中华儿女的志气。众所周知,国际间的竞争说到底还是综合国力的竞争,关键是科学技术的竞争。科学技术是第一生产力,科学技术需要人去掌握,去运用,去发展,人才又是至关重要的。科技进步的基础在教育,我们应该把开展学科奥林匹克和普及计算机教育,提高未来四化建设人才的素质紧密联系起来。要看到我们的不足,更要激发我们的勇气,作为教练和领队,我们从小将身上也学到不少东西。前三届国际信息奥林匹克取得的成绩已成为历史,第四届将于 92 年 7 月在波恩举行,我们想:中国队还应认真准备,从零开始,重新站到起跑线上,信心来源于勇气,来源于实力,来源于千百万孩子,教师与家长的支持,可以这样说:在写这篇总结时,我们是满怀信心的。

五、值得总结的一条经验

参加学科奥林匹克竞赛,参赛选手和领队在精神和体力上负担都很重,还要克服时差、环境等造成的困难。为了保证选手有充沛的精力投入紧张的两轮竞赛,中国科协多方设法联系机票和选定转机路线,使代表团能提前两天到达雅典,给选手以恢复体力的充分时间,更重要的是,使我们能及时了解到竞赛规则的变更细节,有助于我们及时调整参赛的策略,为取得良好成绩创造了有利条件。否则,将会对我们极为不利。因此,我们建议下一届竞赛,时间上仍要留有充裕余地。

众说纷纭:8 位学习机是否已过时? 专家认定:君在教育界仍是主力军!

北京计算机五厂专家座谈会纪要

最近在社会上,特别是学校、家庭中,讨论以 6502 芯片为主体构成的 8 位学习机是否过时了,是否应该用 16 位的 PC 机取代这一问题成了热门话题。有些学校急需配置教学用计算机,有些学生家长也持币等购,但被众说纷纭的议论弄得主意难定。购现在流行的中华牌或天坛牌学飞机,又怕真的会很快被 PC 机淘汰;若购 16 位 PC 机又怕目前教育软件的匮乏而影响盼子成龙之大计不能成就,究竟买什么样的学习机真是

左右为难。

为了帮助学校及家庭将有限的资金用于购置有用的设备,使之发挥最大效率,真正实现少花钱办实事的目的。六月下旬,由生产北京市优质产品天坛学习机的北京计算机五厂发起并邀请一部分计算机专家及教育界的教授、学者,就以上问题召开了一个专题讨论会。与会者各抒己见,经过热烈的讨论大家一致认为:

一、8 位机并不过时,在中国如此,在教育界更是如此。

从改革开放至今,我国计算机引进有过三次高潮,其中 8 位机、准 16 位机占相当大的比重,我国市场上 8 位机的拥有量达 50 万台,因此分析、掌握、应用 8 位机是我们的基础工作。特别是在教育领域,目前中小学配置的大部分计算机是 8 位机,国产化的中华学习机和天坛学习机的生产已逾十万台,为中小学开展计算机教育提供了硬件环境。在谈到五年前教学用计算机选型时,国务院电子信息推广办公室副主任(高级工程师)陈正清同志指出:“当年在确定机型时选择 8 位机,而不是 16 位机,就是从中国的国情出发,并已经考虑到满足教学的需要,否则为什么不选择 4 位机呢。当年教育界和工业界共同商定把 8 位机作为学习机生产和开发的应用机型,现在还没有到修改、更正的时候,政策上要有稳定性。”

大家一致认为:虽然计算机教育发展这么多年了,但是从全国看,中小学计算机教育总数并不高,全国有计算机的中小学还不到 10%,这说明全国经济情况发展的不平衡。北京属于发展迅速普及较好的地区,但仅中学的配置还需翻两番才能达到标准。因此,从国情上看,我们是一个比较穷的国家,另外从我国目前中小学计算机教育内容来看,8 位机还有相当大的潜力需要挖掘,以提高它的使用率。所以说在教育领域 8 位机是完全可以胜任的。

机电部微机处处长、高工于万源同志一席话,给人以很深的印象,发人深省,他认为:“中国目前的教育水平不见得一定要用 16 位机最好。就象小学生不能一开始就用计算器一样,因为忘掉了数学本身的发展规律,仅仅变成使用“傻瓜”的畸形,孩子是不会脱颖而出的。只有知道了事物的发展过程,再使用先进的科学技术和工具才能有更大的创造。因此从这个意义上讲,用 8 位机甚至更低档的计算机对孩子进行启蒙教育是很有益的。”事实也确是如此,我们连续三届参加奥林匹克竞赛,参赛学生都是 8 位机培训的,只在出国前一个月才上 PC 机培训,但也同样取得了很好的成绩。因此与会者认为在教育领域 8 位机不但没有过时,而且是当之无愧的,今后的工作不是考虑换代的问题,而是应该考虑怎样更好的利用 8 位机为教育界服务,充分挖潜,在应用软件的开发上下功夫。

二、不能忽视 8 位机有丰富的软件资源、技术资源这一事实。

五年来我们在软件开发、应用方面做了很多工作,且 90% 的软件是在 8 位机上开发的,这些软件特别是教学软件已经比较符合当前中小学的教学环境,而且由于这些年计算机的普及,使部分中小学都是能够接

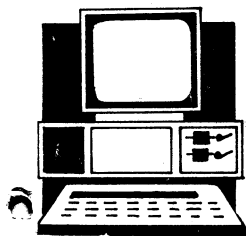
触和使用计算机,并且在开展 8 位机的辅助教学方面有所建树。今明两年北京市将在四所中学、二所小学进行计算机辅助教学实验,实验内容范围很广泛,几乎涉及各个学科,而所有教学软件也都是在 8 位学习机上开发的。全国中小学计算机研究教育中心从 87 年开始由机电部投资 30 万元作为“七五”攻关项目,设立了软件评审制度。这些软件在中小学教学中发挥了很大的作用。所有这些工作为 8 位机在教育界推广应用提供了良好的软件环境并打下了基础。

正如陈正清同志所指出的:“计算机技术发展到今天是以软件为核心的,这是一个趋势,也是现实。”如果硬件设备很好,而师资没有,与之配套服务的软件没有,那么计算机将是一事无成的。我们搞教学软件在选定教材方面的考虑不是一年两年了,而是相当多年积累起来的,是经过循环往复的实践、反复修改才成功的,这是一笔宝贵的财富,所以我们不应该轻易地抛弃自己的财富,要重视人类的一切资源,要珍惜自己的劳动所得;另一方面目前 PC 机上汉化的教育软件几乎没有,如果花很多的精力在 PC 机上去重新开发或移植教育软件不仅是要花许多人力、物力和时间,而 8 位学习机上的丰富软件以及硬件就会被丢弃,造成狗熊掰棒子的后果,延误计算机辅助教学及中小学计算机普及事业的发展。与会者一致认为对这种倾向,教育界要重视,工业界要重视,搞应用开发的也要重视,要中小学计算机普及的工作扎扎实实地搞下去,充分用好现有 8 位学习机及软件资源,切忌见异思迁和不切实际的喜新厌旧。

三、向社会开放、全面发展。

在谈到向社会开放这一题目时,于万源同志深有感触地说:“我们在软件开发方面还是做的不够,特别是在向社会开放,收集更多的软件等方面做的不够。尽管教委、科协、机电部做了大量工作,但光依靠专家、生产厂家是不够的,更重要的是要向社会开放,向用户开放。”我们知道苹果公司之所以拥有如此丰富的软件资源,很重要的一点就是借助社会力量发展自己的。在这一方面北京计算机五厂搞的很好,他们去年成立了“天坛机应用学组”,组织用户之间的技术、信息的交流,而且每年向社会开展学习机的应用成果有奖征集活动。通过这一活动首先鼓舞了社会上一大批人员,使他们认为国家需要,重视这一机型的发展问题,同时自己的劳动成果能够被肯定,且为社会作出贡献;从工厂的角度看,投资不多,收益很大,产生了异想不到的效果。

参加会议的有清华大学教授朱家维、北师大付教授张士英、北师大教授贺龙光、北京市电子振兴办高工华平澜、全国中小学计算机教育研究中心陈星火,一些新闻界同志也参加了讨论会。



PC 用户

IBM PC/XT 定时中断服务程序设计

东北工学院机械二系 张辉 李东升 李奎贤

近几年来,PC 系列微机除用于办公室管理和科学计算外,还越来越多地用于自动控制、在线监测、故障诊断及数据采集处理、频谱分析等各个领域。这些应用在大多数场合下均要求应用程序有定时处理功能,如 A/D、D/A 转换板的定时采样和输出等。有些程序中还要求具有时钟功能。本文针对 IBM PC/XT 系列微机介绍定时中断服务程序的编写方法。

一、8253 定时器的编程

IBM PC/XT 机内装有一片 8253 定时/计数芯片。该芯片内有三个彼此独立的定时/计数器,每一定时器通道都被系统指定为特殊用途。通道 0 的输出线连到系统板的 IRQ0 线上,使其能产生 0 级中断(INT 08H);通道 1 用来发出数据请求信号,供存储器动态刷新用;通道 2 的输出接到机内扬声器,以产生声音效果。

定时器每一通道的时钟输入连到一个 19318MHz 的晶振上。因此,定时器每一时钟周期是 $1/1.19318\text{MHz} = 840$ 毫微秒。通道 0 在计算机上电时被 BIOS 初始化,输出频率为 18.2Hz (定时常数为 65536),这个周期性中断被 BIOS 用来保持日历钟的走时。在定时器的三个通道中,只有通道 0 可供我们作定时中断用。如果我们在编程定时器中断时不需顾及日历钟,那么就可以重新对其进行编程。但通道 0 的中断还控制软磁盘的 I/O 操作,所以在对通道 0 编程后是不能进行磁盘的 I/O 操作的,完成中断服务任务后必须恢复通道 0 的原定时参数(18.2Hz)。

8253 定时器各寄存器的口地址为:

定时器通道 0 锁定寄存器存取口:40H;

定时器通道 1 锁定寄存器存取口:41H;

定时器通道 2 锁定寄存器存取口:42H;

定时器命令寄存器:43H。

修改定时器通道 0 参数的汇编语言程序如下:

```
MOV AL,36H          ;定时器命令:选通道 0,读写
                     ;LSB—MSB(低 8 位—高 8
                     ;位),二进制计数
OUT 43H,AL          ;命令发给 8253 命令寄存器
MOV BX,COUNT        ;定时器定时参数送入 BX
MOV AL,BL           ;先发送 LSB(低 8 位)值
OUT 40H,AL
MOV AL,BH           ;发送 MSB 值
OUT 40H,AL
```

定时器定时参数的计算公式为:

$$\text{COUNT} = 1193180 / \text{每秒中断次数} (0 \leq \text{COUNT} \leq 65535)$$

二、中断处理及中断服务程序

掌握 8253 定时器编程方法后,还需知道中断处理和中断服务程序的编写规则。一个要准备中断的处理程序必须做到:

1. 若以前为开中断则应关闭中断,以防止修改中断向量;
2. 初始化要处理的中断向量,以指向自己的中断服务程序;
3. 若中断以前已被关闭,应保证其它中断向量指向一些有效的中断服务程序;
4. 再次开中断。

修改 8253 定时器通道 0 的中断向量就是要修改 INT 8H 的中断向量,通过调用 DOS 功能 INT 25H(置中断向量)和 INT 35H(取中断向量)来完成。若一个中断通过 PC 机内的 8259 中断控制器请求处理,则中断处理结束后,必须给 8259 的控制口发送一个叫做中断结束的特殊码 EOI(8259 控制口的地址为 20H,EOI 的值为 20H),通知 8259 中断结束。最后,中断服务程序执行特定的中断返回指令 IRET。中断服务程序的结构如下:

```
STI                ;开中断
PUSH AX            ;保存中断服务中修改的寄存器
:
PUSH ES            ;内容
:
MOV AL,20H         ;EOI 值;
MOV DX,20H         ;8259 控制口地址;
OUT DX,AL          ;将 EOI 送 8259
POP ES             ;恢复寄存器内容
:
POP AX             ;
IRET               ;中断返回
```

中断服务程序必须遵守以下简单而严格的步骤:

1. 保存系统内容(寄存器、标志等);
2. 如果有些中断允许在该处理程序运行时发生,但这些中断会引起干扰,则应禁止这些中断;
3. 对该服务程序运行时允许的中断应开中断;
4. 确定中断原因;
5. 完成该中断要求的工作;
6. 恢复系统内容;
7. 对服务程序运行时封锁的各级中断再开中断;
8. 恢复运行被中断的过程。

在定时器中断服务完成并返回主程序后,还要将 INT 8H 原来的中断向量恢复,以便开始 BIOS 的时钟功能,对磁盘进行 I/O 操作。另外,在编写中断服务程

序时还需注意的是在中断服务中,硬件中断服务(如键盘、打印机、屏幕及通讯等)程序决不允许调用 DOS 功能,只能使用 BIOS 例行中断处理程序。

三、程序编写实例

为使读者能更快的掌握上述定时器中断服务程序的编写方法,笔者针对某一型号的 A/D 转换板编写了定时数据采集程序,供读者参考。其中数据定时采集程序用 8088 汇编语言编写,而数据处理及输出用 FORTRAN 编写。程序如下:

```
C   FORTRAN 语言采样主程序
      INTEGER * 2 CHN,AAR(1024),CCC(1024,100)
      INTEGER ACCOUNT
      CHARACTER * 20 OUTFIL,OUT1
      CHARACTER BI * 1,BC * 4,BC1 * 3
1    WRITE(*,2)
2    FORMAT(1X,'请输入采样频率=')\
      READ(*,'(15)') IFREQ
      IF (IFREQ.LT. 19. OR. IFREQ. GT. 7100) GOTO 1
3    WRITE(*,4)
4    FORMAT(1X,'请输入采样点数(1—1024)=')\
      READ(*,'(15)') NUM
      IF (NUM. GT. 1024. OR. NUM. LT. 1) GOTO 3
5    WRITE(*,6)
6    FORMAT(1X,'请输入采样通道号(0—15)=')\
      READ(*,'(15)') CHN
      IF (CHN. GT. 15. OR. CHN. LT. 0) GOTO 5
7    WRITE(*,8)
8    FORMAT(1X,'请输入采样数据块数(1—20)=')\
      READ(*,'(15)') IBCK
      IF (IBCK. GT. 20. OR. IBCK. LT. 1) GOTO 7
      ACCOUNT = INT(1. 19318E6/IFREQ)
C   ACCOUNT 是 8253 定时器的定时常数
      WRITE(*,*) '请按任一键开始采样!'
      READ(*,'(A1)') BI
      WRITE(*,*) '请稍候,正在采样……'
      DO 10 II=1,IBCK
C   调用汇编语言采样程序
      CALL SAMPLE(AAR,ACCOUNT,NUM,CHN)
      DO 9 I=1,NUM
9     CCC(I,II)=AAR(I)
10    CONTINUE
      DO 11 II=1,IBCK
      DO 11 I=1,NUM
      IF (CCC(I,II). GT. 2047) CCC(I,II)=CCC(I,II)-4095
11    CONTINUE
      WRITE(*,12)
12    FORMAT(1X,'采样结束,存储数据吗(Y/N)?')\
      READ(*,'(A1)') BI
      IF (BI. NE. 'Y'. AND. BI. NE. 'y') GOTO 18
      WRITE(*,13)
13    FORMAT(1X,'请输入数据文件名=')\
      READ(*,'(A20)') OUTFIL
      OPEN(6,FILE=OUTFIL,STATUS='NEW')
      DO 17 II=1,IBCK
      WRITE(6,14) II
14    FORMAT(1X,'BLOCK NO. ',I2)
      DO 16 I=1,NUM
      CCCC=2500./2047 * CCC(I,II)
      WRITE(6,15) CCCC
```

```
15  FORMAT(1X,F8.2)
16  CONTINUE
17  CONTINUE
18  CONTINUE
      STOP
      END
```

汇编语言采样子程序

```
      PUBLIC SAMPLE
FRAME  STRUC
SAVEBP DW ?
SAVEDS DW ?
RETADD DD ?
CHN     DD ?
NUM     DD ?
ACCOUNT DD ?
AAR     DD ?
FRAME  ENDS
DATA    SEGMENT PUBLIC 'DATA'
CSS     DW ?
OFF     DW ?
DATA    ENDS
DEROUP  GROUP DATA
CODE    SEGMENT PARA 'CODE'
SAMPLE  PROC FAR
      ASSUME CS, CODE, DS, DATA
      PUSH BP
      PUSH DS
      MOV BP, SP
      PUSH DI
      PUSH SI
      LES DI, [BP]. CHN      ;取采样通道号,
      MOV AX, ES: [DI]      ;存入寄存器 SI
      MOV SI, AX
      LES DI, [BP]. ACCOUNT ;取 8253 定时器时间常数,
      MOV BX, DS: [DI]      ;存入寄存器 BX
      LES DI, [BP]. NUM     ;取采样点数,
      MOV CX, DS: [DI]      ;存入寄存器 CX
      LES DI, [BP]. AAR     ;取数据存放的首地址
      PUSH DI
      PUSH CX
      PUSH BX
      MOV AX, DATA
      MOV DS, AX
      CLI                    ;清中断标志,禁止一切中断
      PUSH ES
      MOV AX, 3508H          ;取原 INT 08H 的中断向量
      INT 21H
      MOV AX, ES            ;中断向量的段地址存入存储器
      MOV CSS, AX          ;CSS,偏移量存入存储器 OFF
      MOV OFF, BX
      PUSH DS
      MOV AX, SEG SAM       ;取采样程序的段地址及偏移量
      MOV DS, AX           ;利用 DOS 功能 INT25H 使 INT08H
      MOV AX, 2508H         ;的中断向量指向采样程序
      MOV DX, OFFSET SAM
      INT 21H
      POP DS
      POP ES
      POP BX
      MOV AL, 36H           ;编程 8253 定时器通道 0
      OUT 43H, AL
```



```

MOV AL,BL      ;修改 8253 定时器通道 0
OUT 40H,AL     ;的定时常数,使之产
MOV AL,BH      ;生所要求的每秒中断
OUT 40H,AL     ;次数
MOV AL,0FCH    ;编程 8259 中断控制器
OUT 21H,AL     ;允许定时器和键盘中断
POP CX
POP BX
STI            ;开中断
CC1: CMP CX,0   ;检查是否采完所有数据,
JNZ CC1        ;未采完继续等待
CLI           ;清中断标志
MOV AX,CSS     ;取出原 INT 08H 的中断向量
MOV DX,OFF     ;恢复原中断服务程序
MOV DS,AX
MOV AX,2508H
INT 21H
MOV AL,36H     ;恢复 8253 定时器通道 0
OUT 43H,AL     ;的定时常数
MOV BX,65535
MOV AL,BL
OUT 40H,AL
MOV AL,BH
OUT 40H,AL
MOV AL,0
OUT 21H,AL
STI            ;开中断
POP SI
POP DI
POP DS
POP BP
RET 16         ;返回 FORTRAN 主程序
SAMPLE ENDP
;采样程序—中断服务程序
SAM PROC FAR
MOV AX,SI
MOV DX,2A0H
OUT DX,AL
INC DX
XOR AX,AX      ;采样过程
OUT DX,AL
INC DX
S1: IN AL,DX
AND AL,1
JNZ S1
DEC DX
IN AL,DX
AND AL,15
MOV AH,AL
DEC DX
IN AL,DX
MOV ES:[BX],AX
INC BX
INC BX
DEC CX
MOV AL,20H    ;通知 8259 中断控制器
OUT 20H,AL    ;中断服务完毕
IRET          ;从中断返回
SAM ENDP
CODE ENDS
END

```

通用数据库修改程序

湖北天门市 121 号信箱 周日初

用 FOXBASE 编写较大软件时,为了屏幕的美观和软件设计的完整,采用窗口,以便把一些有用的信息始终保留在屏幕上,因此对数据库的操作并不用 EDIT、BROW 等全屏幕操作命令,而专门编写一些程序来取代它们。但象这样做,对不同的数据库或不同结构的数据库都要编一个修改程序,不仅浪费磁盘空间,还加大了编程的工作量。为了克服这些缺点,本人编了一个数据库的通用修改程序 TYXG.PRG,可以作为一个过程进行调用,调用时只需给出要修改的数据库名(后缀省略)、显示范围即左上角的位置(行、列数)和右下角的位置(行、列数)即可。程序清单附后,如要对数据库 SJK.DBF 进行修改,显示范围从第 2 行 20 列开始至 20 行 78 列时,调用格式为:

DO TYXG WITH "SJK",2,20,20,78

当数据库的字段数不超过 100,字段长不超过 80 时,可直接调该程序。当字段数大于 100 时,需对数组变量 VXX 重新定义,字段长超过 80 时,需另作处理。

该程序已在 AST 286/386 上运行通过。

```

PARAM VNA,VM1,VN1,VM2,VN2
SET SCOR OFF
SET MENU OFF
SET TALK OFF
SET STAT OFF
DIME VXX(100)
IF VM1>VM2.OR.VN1>VN2
    RETU
ELSE
    IF FILE(TRIM(VNA)+'.DBF')
        USE &VNA
    ELSE
        RETU
    ENDI
ENDI
VI=1
DO WHILE LEN(TRIM(FIELD(VI)))<>0
    VXX(VI)=FIELD(VI)
    VI=VI+1
ENDD
VA=VI-1
IF VM2-VM1>VA
    VB=1
    VE=VA
ELSE
    VB=INT(VA/(VM2-VM1+1))+.9)
    VE=VM2-VM1+1
ENDI

```

```

DO WHIL . NOT. EOF()
STOR 0 TO VC,VD
DO WHIL VC<VB
  VD=0
  DO WHIL VD<(VE. AND. VC * VE+VD+1<=VA
    STOR VXX(VC * VE+VD+1) TO VYY
    @VM1+VD,VN1 SAY VYY GET &VYY
    VD=VD+1
  ENDD
  READ
  VC=VC+1

```

```

@ VM1,VN1 TO VM2+1,VN2 CLEA
ENDD
SKIP
@VM2,VN1 SAY "继续修改吗(Y/N)?"
WAIT " " TO VXG
@VM2,VN1 SAY SPAC(VN2-VN1)
IF UPPE(VXG)<>'Y'
  EXIT
ENDI
ENDD

```

PC-DOS 命令 WHEREIS 的改正

天津电子计算机研究所 李 敏

DOS 命令 WHEREIS 用于搜索由用户定义的文件名,但这种搜索只能在硬盘中进行。例如,输入:

C)WHEREIS P * . *

WHEREIS 命令就会自动地搜索整个 C 盘,显示出所有以 P 字母开头的文件的位置(包括路径名,目录等),这对用户查找文件是相当方便的。但是这儿也存在一个问题,即 WHEREIS 并没有显示文件的大小、日期和修改的时间。有时,同一个文件的不同版本分布在许多子目录下,如果没有时间显示,就很难判断哪一种版本是最新的。本文设计的 WHEREIN 命令可弥补这一点,而且功能还有所扩展,它能自动地搜索 A 盘和 B 盘(这可以在 WHEREIN.COM 生成过程中自动设置)。

WHEREIN 是在 WHEREIS 基础上开发的,该命令由一个 BASIC 程序 WHEREIN.BAS 生成,需要在 BASICA 解释程序下运行。如果你不想用 BASIC 程序,也可以在 DEBUG 环境下输入,具体步骤如下:

C)DEBUG WHEREIN.COM

(机器会显示“File not Found”信息,不必理会)。

然后输入: E 100,接下去,你可以利用文后的 BASIC 程序清单,输入 420 至 800 行 DATA 语句中的所有十六进制数,每一行 BASIC 数据对应一行 DEBUG 程序,BASIC 数据行中的逗号用空格代替。当你输入完毕就输入回车,程序输入就完成了。然后再输入 R CX,DEBUG 就会显示 CX 0000,最后依次输入:

26B
-W
-Q

WHEREIN 命令就自动生成了。

```

110 DIM T(38):FOR A=1 TO 38:READ S:T(A)=S:G=G+S:NEXT
120 IF G=69210! THEN 140
130 PRINT "Error in lines 380-410":END
140 FOR A=1 TO 38:FOR B=1 TO 16
150 READ C$:C=VAL("&H"+C$)
160 TOTAL=TOTAL+C:NEXT
170 IF TOTAL=T(A) THEN 190

```

```

180 PRINT "Error in line ";A*10+410:END
190 TOTAL=0:NEXT
200 RESTORE 420:CLS:KEY OFF:LOCATE 8,1,1:INS%=0
210 ADD%=4:DRIVE$="A":GOSUB 320
220 ADD%=2:DRIVE$="B":GOSUB 320
230 ADD%=1:DRIVE$="C":GOSUB 320
240 IF INS%=0 THEN INS%=1
250 LOCATE 16,30,1:PRINT "Writing Byte No."
260 OPEN "WHEREIN.COM" AS #1 LEN=1:FIELD #1,
  AS BYTE $
270 FOR I=1 TO 618
280 LOCATE 16,50,0:PRINT USING "###";I
290 READ HEXTWO$:IF I=609 THEN HEXTWO$=MID
  $(STR$(INS%),2)
300 LSET BYTE$=CHR$(VAL("&H"+HEXTWO$)):
  PUT #1
310 NEXT:CLOSE #1:LOCATE 18,20:PRINT "Done":
  PRINT:END
320 PRINT TAB(20)"Include drive "DRIVE$ " in search Y/N?";
330 A$=INKEY$:IF A$="" THEN 330
340 IF A$="Y" OR A$="y" THEN INS%=INS%+
  ADD%:GOTO 370
350 IF A$="n" OR A$="N" THEN GOTO 370
360 BEEP:GOTO 330
370 PRINT A$:PRINT:RETURN
380 DATA 1711,1790,2094,1211,1123,1235,1254,1382,
  1556,2028
390 DATA 1589,1661,2042,2654,2099,1633,1993,1938,
  1430,1954
400 DATA 1826,1736,1970,1533,2312,1760,2439,2081,
  1989,2097
410 DATA 2258,1514,2478,1479,1695,1604,2101,1961
420 DATA B2,0A,E8,E3,01,BE,82,00,BF,B9,03,AC,3C,
  0D,74,03
430 DATA AA,EB,F8,30,C0,AA,A0,60,03,24,01,74,14,
  BF,65,03
440 DATA 30,C0,FC,B9,40,00,F2,AE,8B,DF,4B,BA,00,
  00,E8,52

```

450 DATA 00,A0,60,03,24,04,74,21,BF,65,03,C7,05,
41,00,C7
460 DATA 45,01,3A,00,C7,45,02,5C,00,C7,45,03,00,
00,8B,DF
470 DATA 83,C3,03,BA,00,00,E8,2A,00,A0,60,03,24,
02,74,21
480 DATA BF,65,03,C7,05,42,00,C7,45,01,3A,00,C7,
45,02,5C
490 DATA 00,C7,45,03,00,00,8B,DF,83,C3,03,BA,00,
00,E8,02
500 DATA 00,CD,20,56,52,E8,65,01,E8,82,01,72,0D,
E8,5F,00
510 DATA E8,98,01,72,05,E8,57,00,EB,F6,5A,52,E8,
57,01,E8
520 DATA 6B,01,72,24,8B,F2,F6,44,15,10,75,0B,E8,
7C,01,72
530 DATA 17,F6,44,15,10,74,F5,80,7C,1E,2E,74,EF,
E8,0B,00
540 DATA 50,B4,1A,CD,21,58,EB,E4,5A,5E,C3,57,56,
50,53,FC
550 DATA 8B,F2,83,C6,1E,8B,FB,AC,AA,08,C0,75,FA,
8B,DF,FD
560 DATA AA,B0,5C,AA,E8,9C,FF,5B,C6,07,00,58,5E,
5F,C3,50
570 DATA 52,56,53,51,83,C2,1A,8B,F2,8B,04,8B,54,
02,BF,0A
580 DATA 00,E8,86,00,B2,20,E8,DF,00,E8,DC,00,83,
EE,02,8B
590 DATA 04,25,E0,01,B1,05,D3,F8,E8,90,00,B2,2D,
E8,C8,00
600 DATA 8B,04,25,1F,00,E8,A5,00,B2,2D,E8,BB,00,
8B,04,25
610 DATA 00,FE,B1,09,D3,E8,05,50,00,E8,91,00,B2,
20,E8,A7
620 DATA 00,E8,A4,00,83,EE,02,8B,04,25,00,F8,B1,
0B,D3,E8
630 DATA E8,58,00,B2,3A,E8,90,00,8B,04,25,E0,07,
B1,05,D3
640 DATA F8,E8,69,00,B2,20,E8,7F,00,E8,7C,00,59,
5B,5E,BA
650 DATA 65,03,8A,07,C6,07,00,E8,CF,00,88,07,5A,
52,83,C2
660 DATA 1E,E8,C5,00,E8,B3,00,5A,58,C3,31,C9,F7,
F7,83,C2
670 DATA 30,52,31,D2,41,09,C0,75,F3,BA,20,00,52,
41,83,F9
680 DATA 07,75,F9,B4,02,5A,CD,21,E2,FB,C3,31,D2,
F7,F7,83
690 DATA C2,30,52,31,D2,F7,F7,09,D2,74,03,83,C2,
10,83,C2
700 DATA 20,52,B9,02,00,B4,02,5A,CD,21,E2,FB,C3,
31,D2,F7
710 DATA F7,83,C2,30,52,31,D2,F7,F7,83,C2,30,52,
B9,02,00
720 DATA B4,02,5A,CD,21,E2,FB,C3,B4,02,CD,21,C3,

56,BE,B9
730 DATA 03,E8,0B,00,5E,C3,56,BE,61,03,E8,02,00,
5E,C3,50
740 DATA 57,8B,FB,FC,AC,AA,08,C0,75,FA,5F,58,C3,
51,83,FA
750 DATA 00,77,03,BA,9B,03,83,C2,2B,B9,10,00,B4,
1A,CD,21
760 DATA 52,BA,65,03,B4,4E,CD,21,5A,59,C3,51,52,
BA,65,03
770 DATA B9,10,00,B4,4F,CD,21,5A,59,C3,50,52,B4,
02,B2,0A
780 DATA CD,21,B2,0D,CD,21,5A,58,C3,50,52,56,FC,
8B,F2,B4
790 DATA 02,AC,8A,D0,CD,21,AC,08,C0,75,F7,5E,5A,
58,C3,00
800 DATA 05, 2A, 2E, 2A, 00, 43, 3A, 5C, 00, 00

巧算 1000!

东北电力学院 李 帆

本人采用一种特殊的技巧,求解 1000 以内的阶乘,不但程序量小,而且运算速度极快。现介绍原理如下:根据求 1000 阶乘的特点,是由一个不大于 1000 的数乘以一个位数极多的数,把这个大数的每一位都放到数组 a 中的每一个单元,而把不大于 1000 的这个数放入一个单元 i 中,用 i 中的数按顺序乘以数组中的每一个数,当数组中的数大于 9 时,留下个位,并记进位,在计算下一个数后,加上前一个数的进位,并留下个位,记进位,一直到计算完数组中的所有数。当 i 从 1~1000 变化,按以上规则计算,最后 a 中的数即为 1000!,当 i=1~n 变化时,就可以求出 n!。本程序用 C 语言编写,在 Micro VAX 机上计算 1000! 需用 50 秒,而在 286 微机上不到 30 秒。

```
#define max 5000
#include <stdio.h>

main () {
    int i,j,x,c,n,a[max],o=0,d=1;
    while (1==1) {o++;
        for (i=0;i<max;i++) a[i]=0;a[0]=1;
        printf("\n\n No. %d Input N=",o);
        scanf("%d",&n);
        printf("\n %d ! = \n",n);
        if (n<0)break;
        if (n==0)n=1;
        for (i=1;i<=n;i++){c=0;
            for (j=0;j<=d;j++){x=a[j]*i+c;c=x/10;a[j]=
                x%10;}
            while (a[j]==0)j--;d=j+5;
            while (a[j]==0)j--;d=0;
            for (;j=0;j--){printf("%d",a[j]);if (d++%79==
                0)putchar('\n');}
        }
    }
```


又发现一种新病毒

沈阳纺织工业非织造布技术开发中心 王家春

一、病毒的发现

开机后,计算机工作一切正常,输完程序后想打印,突然发现打印机不能联机。当初并没有认为有病毒,以为系统文件、打印模块或打印机等有问题,所以就逐个问题查找,过程是这样:

1. 退出汉字系统后仍不能联机,重新冷热启动计算机都不能联接打印机;排除了汉字系统有问题。

2. 更换主机检查,打印机和电缆都没问题。

3. 认为硬盘系统文件有问题,用软盘(贴写保护的硬盘)启动,打印机可以联机。传送系统文件到硬盘,但从硬盘启动仍不能联机。

从以上检查可以怀疑硬盘有病毒。但以前已知的病毒并没有不联打印机的。通过 PCTOOLS 发现,从硬盘启动时,系统内存被占用了 1KB。通过 DEBUG 发现,不联接打印机是由于 ROM BIOS 数据区的串、并行口地址被改变了。这时才断定,计算机染上了以前未发现的新病毒。由于此病毒更改系统串、并行口地址,这里暂称之为“IO”病毒。

“IO”病毒属操作系统型病毒,它藏在磁盘(磁盘指软盘和硬盘,以下同。)的第一物理扇区,即引导扇区。只要用带有“IO”病毒的磁盘启动计算机,“IO”病毒就常驻内存,而有传染“IO”病毒的能力,能感染任何没贴写保护的软盘。用带病毒的软盘启动,不论是否是系统盘,只要一次,就能感染硬盘。

“IO”病毒感染硬盘后,从硬盘启动 20H 次后,接在串、并行口上的外设都不能联机。

软盘感染后,某个文件可能有一个扇区内容被改变。如果系统盘感染“IO”病毒后又写某文件,可能因文件覆盖了引导扇区,此系统盘将不能启动系统。

“IO”病毒的特点是,硬盘感染后并没有什么症状。从带病毒硬盘启动系统 20H 次后,串、并行口外设才不联机。待到发现“IO”病毒时,已有很多软盘被感染。

“IO”病毒另一特点是,它把软盘的引导扇区写到 1 面 27H 道 8 扇区,覆盖了该扇区的数据,无可挽救的破坏了该区原数据。不象有些病毒仅破坏 FAT 表或文件目录,但文件体还存在盘上,还有挽救的机会。

对于 360KB 软盘,1 面 27H 道 8 扇区是在软盘的尾部,一般情况下问题不大。但对 1.2MB 的软盘,1 面 27H 道位于软盘前部,问题就经常出现了。

二、“IO”病毒的清除和免疫

1. 检查有无病毒

“IO”病毒藏在磁盘第一物理扇区,因为它修改引导扇区时,保留系统版本 OEM 标志和后部错误提示,初看是看不出病毒的,这也正是“IO”病毒的险恶之处。

读磁盘引导扇区到内存(硬盘为主引导扇区)。如

头三个字为 E9 8B 00,偏移 89H,8AH 两字节为 B1H,90H,即可断定磁盘已感染“IO”病毒。

2. 清除“IO”病毒

对于软盘很容易清除,可有多种方法。在清除“IO”病毒时一定要在系统无病毒情况下清除,方法是用一无病毒的硬盘启动系统。可用 DEBUG、PCTOOLS 等程序清除病毒,即把 1 面 27 道 8 扇区的内容写到 0 面 0 道 1 扇区。要注意的是:1 面 27H 道 8 扇区还是引导扇区否,如已被文件覆盖则不是原引导扇区,可用别的同类软盘引导扇区替代即可。

对于硬盘有点问题,但也容易消毒。因为“IO”病毒是直接改写硬盘主引导程序,没有保护原主引导扇区,所以要费点事。如果以前备份了硬盘主引导扇区,那么重新拷贝回去即可。如果未备份,可以把分区情况相同的其它硬盘的主引导程序备份出来,写到带病硬盘即可消毒。因为病毒程序仅修改引导程序的中间一部分,所以可以直接修改引导扇区,使之能引导活动分区就行了。最后也是最糟的方法是格式化硬盘,要低级格式化,高级格式化不能消除“IO”病毒。格式化前一定要备份硬盘上的所有文件。

3. 免疫

“IO”病毒是用引导扇区偏移 89H,8AH 两字节进行判别是否感染。如这两字节是 B1H,90H 则已感染,如不是则未感染。所以可以修改原引导扇区,把偏移 89H,8AH 修改为 B1H,90H 即能免疫。

扬基病毒快速解毒法

绵阳市职工社会保险事业管理处 李凤明

扬基病毒是一种文件型病毒,攻击所有 .COM 和 .EXE 程序。它采用了许多反跟踪措施,一旦发现跟踪,就将带毒程序还原。因此,可以利用它的这一特性来解毒,具体步骤如下(假设程序 A. EXE 感染了扬基病毒):

一、运行有毒程序,使系统感染上扬基病毒;

C>A

二、用 DEBUG 程序来加载染毒程序,再退出 DEBUG;

C>DEBUG A. EXE

—Q

此时染毒程序已被病毒程序还原成无毒的了。然后,用同样方法对 DEBUG. COM 本身解毒,再重新启动。应特别注意检查 COMMAND. COM 程序,否则一启动又有毒了。

设备驱动程序 ANSI.SYS 的使用

青岛 山东纺织工学院 88 自动化二班 王立勇

MS-DOS 2.0 及以后的版本,扩充了可安装设备驱动程序的功能,使计算机用户可以根据需要替换或编制自己的设备驱动程序,增强了灵活性及功能。

ANSI.SYS 是扩充键盘及屏幕显示的一个设备驱动程序,它载入内存后,替换了常驻内存的 CON 设备的驱动程序。ANSI.SYS 不仅支持原驱动程序所有的功能,而且还做了较大的扩充,提供了光标定位、移动光标、屏幕模式设置及键盘宏定义等功能。

下面就介绍一下 ANSI.SYS 的命令及使用方法。

一、命令格式:

①光标定位:

ESC[#;# 把光标移至指定#行、#列处,默认值为屏幕原点。

②光标上移:

ESC[#A 光标上移#行,默认为一行。若光标已在顶行则不动。

③光标下移:

ESC[#B 光标下移#行,默认为一行,如已在最后一行,则光标不动。

④清屏:

ESC[2J 清屏,光标回到原点。

⑤清行

ESC[k 从光标位置清至行末。

⑥设置屏幕模式:

ESC[= #h 其中#的含义为:

0	40×25	黑白方式
1	40×25	彩色方式
2	80×25	黑白方式
3	80×25	彩色方式
4	320×200	黑白图形
5	320×200	彩色图形
6	640×200	黑白图形
7		行结束时换行。

⑦键的重定义:

ESC[#;#;.....;p 或 ESC[#;#; 'string';#p

其中#为键值,string为定义的字符串。

注:①#为一数字参数,没指明时,使用默认值。

②ESC为换码符,即ASCII的27,而不是ESC三个字符。

③ANSI.SYS的命令还有许多,详见DOS手册。

④使用ANSI.SYS的命令,必须在磁盘根目录中有ANSI.SYS文件并在设备配置文件CONFIG.SYS中加有命令DEVICE=ANSI.SYS。

二、使用方法:

ANSI.SYS的命令字符串可以由系统功能调用

INT 21H 的 1、2、6、9 号子功能发出,因而我们可以在汇编语言或高级语言中使用 ANSI.SYS 命令。在高级语言中的应用可以替代调用汇编语言子程序,完成移动光标,清屏等高级功能。

值得一提的是,载入 ANSI.SYS 后,可以解决在 CCDOS 中使用 CLS 命令造成屏幕全黑的问题。

在汇编语言中使用 ANSI.SYS 命令十分简单,只要在数据段中加上以 '\$' 结尾的命令字符串,用 INT21H 的功能 9 显示该字符串即可,我们就不多说了。

下面我们举例说明在高级语言中的应用:

①屏幕模式的设置:

命令串为:ESC[=#h

i 在 FORTRAN 语言中可以用以下子程序实现:

```
SUBROUTINE SCREEN (mode)
INTEGER mode
WRITE (*, '(1X,A1,2H[=,I1,1Hh)') CHAR(27),
mode
RETURN
END
```

ii 在 C 语言中表示为:

```
void screen (mode);
unsigned char mode;
{
printf("\033[= %dh",mode);
}
```

iii 在 PASCAL 中的应用:

```
procedure screen (mode;integer);
BEGIN
writeln(char(27), '[=',mode, 'h');
END;
```

在主程序中只要正确设置 mode 的数值,再调用 screen 就能设置屏幕模式。

②清屏:

命令码为:ESC[Z]Z

i FORTRAN 语言的应用:

```
SUBROUTINE CLS
WRITE (*, '(1X,A1,3H[2J)') CHAR(27)
RETURN
END
```

ii C 语言中的应用:

```
void cls()
{ printf("\033[2J");
}
```

iii 在 PASCAL 中的应用:

```
procedure cls;
BEGIN
writeln(char(27), '[2J');
END;
```

三、键盘重定义:

键的重定义是 ANSI.SYS 提供的较为重要的功能,也是较常用的功能。我们可以将一些键设置为编程语言的保留字,而另一些键则设置为相应的编译命令或 DOS 系统命令,从而加快了软件开发的进度。也可以将键定义为汉字的词组或句子,从而加快了汉字录入的速度,而且还不用调入词组。当然我们也可定义一些制表符,方便我们使用。

键重定义的 ESC 换码序列使用编程语言发出较为烦琐,使用不便,下面介绍两种较为方便的方法。

方法一:用 DOS 的内部命令 prompt。该命令是设置 DOS 系统提示符的,其格式为:prompt[text]。其中 text 是将作为系统提示符的文本字符串,在此字符串中可以加上以 '\$' 打头的特殊元字符。

对于特殊元字符的定义可以参阅 DOS 手册,其中元字符 e 可以引导 ESC 的换码序列,用于定义键。

如我们设置 ALT-F10 为 BASICA (表示回车),可以用 prompt \$e[0;113;'BASICA';13p。

使用这种方法会取消提示符,因而在定义完键后,要用 prompt \$n\$g 恢复原来的提示符,当然也可用元字符设计自己的提示符,方便使用。

《电子与电脑》91 年第 4 期,刊登何管略的文章中有一处错误,用 prompt 进行键的重定义,并不会被下一条 prompt 命令冲掉,除非下一条 prompt 又重新定义了这一键。因而我们可以在一个批文件中定义所有的键,而不必为每一条 prompt 写一个批文件,这样不但使用方便,而且还可以节约大量的盘空间(DOS 为每一文件至少分配一簇盘空间,对于不同的盘将可能是 0.5K 或 1K 或 2K 或 4K 空间)。

方法二:这是一种新的方法,即将 ESC 换码序列直接编辑成一个文本文件,需定义键时,用 type 命令显示该文本文件即可。

在编辑时,ESC 码的输入可以采用以下方法

①先键入 CTRL-V,再键入[,此时屏幕显示 ^ V [然后再输入以 "[" 引导的字符串。

②按下 ALT 键,用数字小键盘键入 27,再放开 ALT 键,此时屏幕显示 "←"。

③键入 CTRL+[此时屏幕显示 "^ ["。

输入了 ESC 码后,再输入以 "[" 引导的字符串即可。如我们定义 ALT-F10 为 BASICA 可输入 ^ [[0;113;'BASICA';13p,当要恢复键的原定义时可键入键的扫描码,如 ^ [[0;113;0;113p 能使 ALT-F10 恢复 DOS 的原定义。

求 购

1802 程序设计手册、COMX PCI 机软硬件资料。

联系地址:西安市东郊东方机械厂子弟小学 杨兴岐

邮政编码:710043

巧用虚拟盘

北京工业大学 901031 班 李焕民

目前市场上的普及型 PC 比较多,但是许多用户只配了一台驱动器,这在文件拷贝的时候非常不方便,要多次抽、插盘。利用虚拟盘就能解决这个问题,首先在主盘上建立 CONFIG.SYS 文件,使它包含下列命令:

```
DEVICE=VDISK.SYS 363 512 64
```

然后用主盘启动,机器会自动建好虚拟盘,并告诉用户分配给这个驱动器的字母是什么。注意,虚拟盘不能进行 FORMAT。本人使用的是 BF-PC Boy,内存为 512K,如果您的机子内存较小,可把盘容量 363K 改为适当的数。

递归解九连环

辽宁大学 计 3 班 钟 武

九连环是一种古老的智力玩具,相传是公元前 234 年前后我国的一位武士发明的,两千多年来在民间广为流传。在十六世纪中叶传入欧洲,并被意大利数学家载入典籍,誉之为“中国套环难题”。

九连环构思精巧、结构奇特,它是由九个关连着的环、一根套柄和一些附件构成。玩法分套上九连环和解开九连环两个过程。九连环从完全脱离的初始状态一直“走”到套柄将九个环全部套上的状态的过程,称为套上九连环;其逆过程称为解开九连环。

完成解开九连环的基本思想是:首先,将九连环的前七个环解开,接着将第九个环解开,再将前七个环套上,最后将第八个环解开。这就完成这一过程。

套上九连环的基本思想是:首先,将前八个环套上套柄,再解开前七个环,接着将第九个环套上,最后将前七个环套上。

按此类推,九连环问题归结为一个环和二环的套上和解开问题。显然,这两种情况是可直接解决的。

从以上分析可知:九连环问题是一个递归问题。递归有很强的能力,也给程序设计带来很大方便。递归表述程序的必要且充分的工具是过程或子程序,因为它允许语句被赋以一个名字并按其名字进行调用。设过程 unload(9)表示解开九连环,过程 load(9)表示套上九连环。

递归有直接递归和间接递归两种。若过程明显包含着对自己的引用,称为直接递归。若过程 A 包含着对另一过程 B 的调用,而 B 又直接或间接地调用 A,则称 A 是间接递归的。从分析可知,九连环问题的解决

既含有直接递归又含有间接递归。

下面描述 n 连环问题的一般算法: ($n > 2$) 首先描述过程 unload(n):

- (1) unload($n-2$)
- (2) 解下第 n 个环
- (3) load($n-2$)
- (4) unload($n-1$);

再有过程 load(n):

- (1) load($n-1$)
- (2) unload($n-2$)
- (3) 套上第 n 个环
- (4) load($n-2$).

下面, 分别用 C 语言和 PASCAL 语言编写解开九连环的程序。若编写套上九连环的程序, 只要在主程序中稍加改动, 直接调用套上九连环过程即可。

C 语言源程序:

```
#include <stdio.h>
#define NCIRCLE 9
int step=0;
void unload __all(int)
void load __all(int)
main()
{
    unload __all(NCIRCLE);
}
void unload __all(int n)
{
    if (n>2) {
        unload __all(n-2);
        printf("%16d\t%d---> down\n", ++step, n);
        load __all(n-2);
        unload __all(n-1);
    }
    else
        printf("%16d\t%s---> down\n", ++step, (n==
2)? "1 & 2": "1");
}
void load __all(int n)
{
    if (n>2){
        load __all(n-1);
        unload __all(n-2);
        printf("%16d\t%d---> up\n", ++step, n);
        load __all(n-2);
    }
    else
        printf("%16d\t%d---> up\n", ++step, (n==1)?
"1": "1 & 2");
}
```

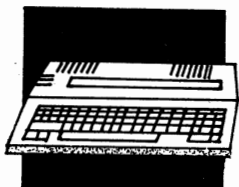
PASCAL 语言源程序:

```
PROGRAM ninecircle (output);
CONST
```

```
ncircle = 9;
PROCEDURE load (n:integer);
forward;
PROCEDURE unload (n:integer);
BEGIN
    CASE n OF
        1: writeln('1 ---> down';20);
        2: writeln('1 & 2 ---> down';24);
    OTHERWISE
        BEGIN
            unload(n-2);
            writeln(n;10, ' ---> down');
            load(n-2);
            unload(n-1)
        END
    END
END;
PROCEDURE load;
BEGIN
    CASE n OF
        1: writeln('1 ---> up';18);
        2: writeln('1 & 2 ---> up';22);
    OTHERWISE
        BEGIN
            load(n-1);
            unload(n-2);
            writeln(n;10, ' ---> up');
            load(n-2)
        END
    END
END;
BEGIN
    unload(ncircle)
END.
```

以上两个程序都在 VAX-8350 机上运行成功。当环数取 5 时, 即 ncircle=5 时, 运行结果为:

```
1 1 ---> down
2 3 ---> down
3 1 ---> up
4 1 & 2 ---> down
5 5 ---> down
6 1 & 2 ---> up
7 1 ---> down
8 3 ---> up
9 1 ---> up
10 1 & 2 ---> down
11 4 ---> down
12 1 & 2 ---> up
13 1 ---> down
14 3 ---> down
15 1 ---> up
16 1 & 2 ---> down
```



将 CEC-I 中华学习机改制成专用机

学习机之友

陕西电子技术研究所 姜良夫

CEC-I 中华学习机已作为家庭教育、娱乐、事务用机进入千家万户,它仍以简单易学,操作方便而适于改作各种专用机。对于改制专用机,人们希望尽量发掘机内潜力而不重新投资,一来可以降低成本,二来避免增加技术上的复杂程度。这就首先要求机内有提供存放专用程序的内存空间。本文介绍如何充分利用 CEC-I 机内存空间,置放自己的专用程序,将其改制成使用方便的专用机。

一、CEC-I 内存分配

CEC-I 机的存储器系统要比 APPLE II 机复杂,容量也大许多。CEC-I 机内存分主存和辅存,主存储器有 16KROM 和 64KRAM。利用主存储器和辅存储器之间的切换软开关

RAMRD; RAMWRT; ALTZP

还可访问辅存空间,其地址为 \$ 0000 ~ \$ BFFF 以及 \$ D000 ~ \$ FFFF。CEC-I 机中辅存空间分为四块,三块为 ROM 区,一块为 RAM 区。

辅存中, \$ 0000 ~ \$ 3FFF 是 RAM 区,并映射到主 RAM 所对应的存储器上,可以说是主存和辅存的公共区。其余空间皆为 ROM 区,存放汉字系统软件和 LOGO 语言,它们分别存储于 U_7 和 U_{35} 两片 27256 芯片中。汉字库则存于两片 1M 位的 U_{33} 、 U_{34} 芯片中。

芯片 U_7 中,物理地址 \$ 5000 ~ \$ 7FFF 的空间,固化着占系统逻辑地址 \$ D000 ~ \$ FFFF 的监控程序、BASIC 解释程序等,这是主 ROM 区。物理地址 \$ 4100 ~ \$ 4FFF 的空间是内部 ROM 区。物理地址 \$ 0000 ~

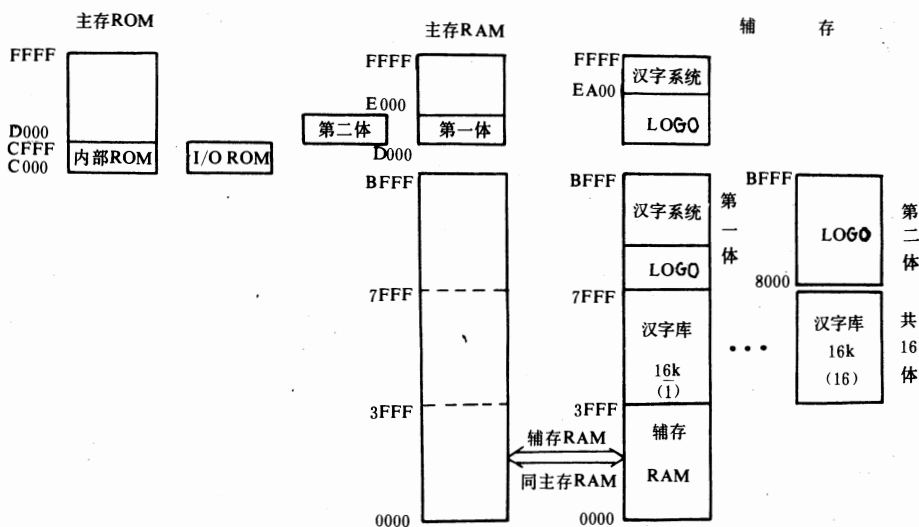
\$ 3FFF 的是逻辑地址 \$ 8000 ~ \$ BFFF 第二体的辅存 ROM,固化有 LOGO 语言。芯片 U_{35} 中,在 32K 的存储空间内,固化有汉字系统及 LOGO 语言等。占物理地址 \$ 0000 ~ \$ 3FFF 是辅存 \$ 8000 ~ \$ FFFF 的第一体,固化有汉字拼音码表和 LOGO 语言, \$ 5000 ~ \$ 7FFF 的地址空间是辅存 \$ D000 ~ \$ FFFF ROM 区,固化有汉字管理程序和 LOGO 语言程序。\$ C300, \$ C600 的 I/O ROM 也存放在 U_{35} 芯片内。而辅存 \$ 4000 ~ \$ 7FFF 的空间通过体切换,扩充达 16 体,总共计 $16K \times 16 = 256K$,由两个 $128K \times 8$ 的芯片组成,内装有国标一、二级汉字点阵字库。

二、LOGO 语言程序的存放及调用

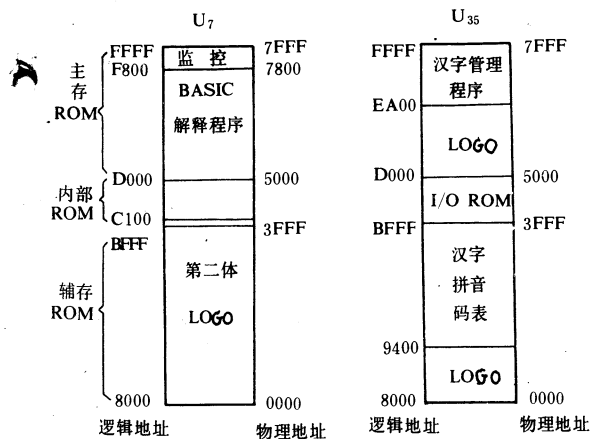
通过对内存配置的分析,不难想到 LOGO 占用的空间可以存放我们的专用程序。从图 2 可知,LOGO 所占空间约 $16K + 5K + 6.5K = 27.5K$,这对于存放一般的应用程序是可以满足的。

那么 LOGO 是怎样启动和工作的? 在键盘上敲击 LG 命令,程序就转 LG 的入口地址 \$ D10C,程序如下:

D10C-A9 10	LDA # \$ 10
D10E-8D B0 C0	STA #C0B0 ;选第一体
D111-8D 07 C0	STA \$ C007 ;切换到内部 ROM
D114-20 00 C6	JSR \$ C600 ;跳到转存子程序
D117-8D 06 C0	STA \$ C006 ;切换到 I/O ROM
D11A-4C D1 D2	JMP \$ D2D1



图一 CEC-I 系统地址分配图

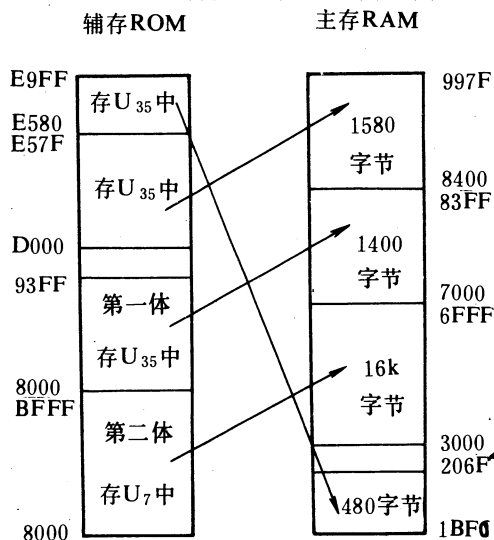


图二 U₇、U₃₅内存程序配置图

这段程序主要作用是跳转到内部 ROM 区从 \$C600 开始的子程序。读者可以在监控状态下,对软开关 \$C007 写操作,即可切换到内部 ROM,如:

```
* C007:1 ✓
* C600L ✓
C600-8D 03 C0    STA $C003
C603-8D 05 C0    STA $C005
:
```

这个子程序分四部份,第一部份把辅存 ROM \$8000~\$BFFF 第二体中的程序转存到主存 RAM \$3000~\$6FFF,第二部份将辅存 ROM \$8000~\$93FF 第一体中的程序转存到主存 RAM \$7000~\$83FF,第三部份将辅存 ROM \$D000~\$E57F 中的程序转存到主存 RAM \$8400~\$997F,第四部份将辅存 ROM \$E580~\$E9FF 中的程序转存到主存 RAM \$1BF0~\$206F。转换对应关系如下图所示:



图三、LOGO 程序转存示意图

执行完转存子程序后,跳转到 \$D2D1 入口地址。对转存到 RAM 中的程序作了校核,无误后,即转到 LOGO 语言的入口地址 \$1BF6。这样 LOGO 程序从辅存调出,转存入主存 RAM,并进入正常运行。

三、BASIC 程序在内存中如何存放。

我们用 LIST 命令可以看到内存中的 BASIC 程序,可进行编辑、修改。但是它们在内存中是如何存放的呢?原来,BASIC 程序在零页使用了一些标志单元,指示这个 BASIC 程序的起始位置、结束位置、变量、数组、字符串存储起始地址等,如:

\$67~\$68:指向 BASIC 程序的起始地址

\$69~\$6A:指向程序中简单变量的起点。BASIC 解释程序冷启动时,指向 \$803;执行 NEW 后,指向 \$804,以后随着程序的输入而自动调整,总是指向程序末地址加 1。

\$6B~\$6C:数组变量起始指针,未执行程序前和 \$69~\$6A 一样,执行后会根据 \$69~\$6A 的指针自动重新设定新值,故修改 \$69~\$6A 会影响此指针。

\$6D~\$6E:指向数组变量的结束地址加 1。

\$6F~\$70:指向字符串使用区的起始点,无 DOS 时为 \$C000,在中文状态时为 \$9200。

\$73~\$74:BASIC 用户可使用的最高地址加 1。无 DOS 时最高指向 \$C000。

\$AF~\$B0:BASIC 程序的终点,并随程序的修改而自动调整。

BASIC 程序一般从 \$801 开始存放。每一行语句的结束用 00 来表示,程序最后以三个 00 作结束。

程序在内存中存放地址可由 \$67~\$B0 中有关参数确定,这就为我们从内存中取出 BASIC 程序或者把 BASIC 程序放入内存提供了依据。

四、应用程序的存放

搞清了 LOGO 语言及 BASIC 程序在内存中的存放情况,就可以着手应用程序的固化工作。

我们用中文 BASIC 编制了一个专用的计算程序,它在内存中的起始地址在 \$801,末地址在 \$21DC,程序全长 \$19DC,约 6K 左右,因而可将其固化在芯片 U₇ 中,放置在物理地址 \$0000~\$3FFF 的 16K 空间内,这是辅存 ROM \$8000~\$BFFF 的第二体空间。程序实际只占用逻辑地址 \$8000~\$99DC 的这部份空间,由于程序容量小,就不需改动 U₃₅ 芯片。另外还需修改、编写从 \$C600 起的转存程序;从 \$D2D1 起的予置、校核程序。对于 \$C600 起的转存程序改写并不困难,可以参照原 LOGO 语言使用的 \$C600 子程序的编写方法。需要注意的是:从主存切换到辅存和从辅存切换到主存,以及辅存中选择第一体与第二体时,软开关使用不当会产生意料不到的情况,当然也就不能正确地执行程序的转存。关于软开关的详细定义,参见“中华学习机 CEC-I 技术手册(硬件)”144~146 页有关内容。

\$D2D1 起始的程序,在原 LOGO 中主要是校验从辅存转存到主存 RAM 中的程序正确与否,正确,则执

行 LOGO 语言,不正确则返回到 BASIC 状态。对于我们的专用程序,则应对零页中有关的标志单元予置参数。即 \$ 67、\$ 68 放入 BASIC 程序首址 \$ 801,在 \$ 69、\$ 6B、\$ 6D、\$ AF 中放入程序末地址低字节,在 \$ 6A、\$ 6C、\$ 6E、\$ B0 中放入程序末地址高字节。然后可校验 RAM 中任意地址的内容是否为应存入的内容,若是,转到 \$ D566,若不是,再转到 \$ D10C,重新装入 BASIC 程序。

我们为上述专用程序编制的转存程序是:

```

4600- 8D 03 C0 STA $ C003
4603- 8D 05 C0 STA $ C005
4606- A9 00 LDA # $ 00
4608- A8 TAY
4609- 85 FC STA $ FC
460B- 85 FE STA $ FE
460D- A9 80 LDA # $ 80
460F- 85 FD STA $ FD
4611- A9 08 LDA # $ 08
4613- 85 FF STA $ FF
4615- 8D 03 C0 STA $ C003
4618- 8D 05 C0 STA $ C005
461B- A9 20 LDA # $ 20
461D- 8D B0 C0 STA $ C0B0
4620- B1 FC LDA ($ FC),Y
4622- 8D 02 C0 STA $ C002
4625- 8D 04 C0 STA $ C004
4628- 91 FE STA ($ FE),Y
462A- C8 INY
462B- D0 E8 BNE $ 4615
462D- E6 FD INC $ FD
462F- E6 FF INC $ FF
4631- A9 99 LDA # $ 99
4633- C5 FD CMP $ FD
4635- D0 DE BNE $ 4615
4637- 8D 03 C0 STA $ C003
463A- 8D 05 C0 STA $ C005
463D- A9 20 LDA # $ 20
463F- 8D B0 C0 STA $ C0B0
4642- B1 FC LDA ($ FC),Y
4644- 8D 02 C0 STA $ C002
4647- 8D 04 C0 STA $ C004
464A- 91 FE STA ($ FE),Y
464C- C8 INY
464D- C0 D0 CPY # $ D0
464F- D0 E6 BNE $ 4637
4651- 60 RTS

```

校验、予置程序是:

```

52D1- A9 01 LDA # $ 01
52D3- 85 67 STA $ 67
52D5- A9 08 LDA # $ 08
52D7- 85 68 STA $ 68
52D9- A9 DC LDA # $ DC
52DB- 85 69 STA $ 69
52DD- 85 6B STA $ 6B
52DF- 85 6D STA $ 6D

```

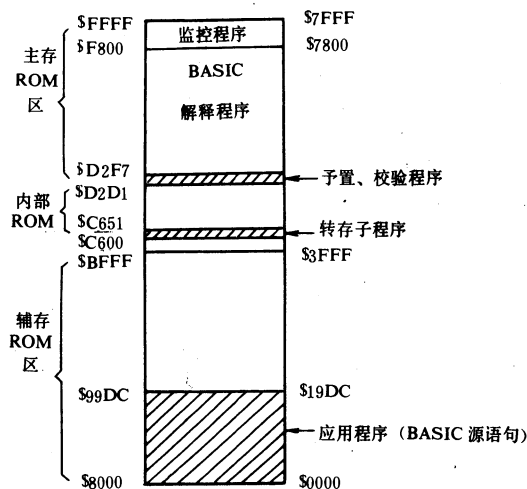
```

52E1- 85 AF STA $ AF
52E3- A9 21 LDA # $ 21
52E5- 85 6A STA $ 6A
52E7- 85 6C STA $ 6C
52E9- 85 6E STA $ 6E
52EB- 85 B0 STA $ B0
52ED- AD 02 08 LDA $ 0802
52F0- C9 07 CMP # $ 08
52F2- D0 03 BNE $ 52F7
52F4- 4C 66 D5 JMP $ D566
52F7- 4C 3C D4 JMP $ D43C

```

注:以上地址是程序在 U₇ 中的存放地址。

BASIC 应用程序、转存程序、予置、校验程序在 U₇ 芯片中的存放位置如图所示:



图四 新 U₇ 中程序分配示意图

五、程序的固化及编程技巧:

下一步便是将这三部程序写入 U₇(27256) 芯片内。可采用的办法很多,主要取决于读者手中现有的编程工具。我们采用的办法是:首先利用苹果机写入卡(只能写 2764 以下的芯片),将三部程序写入 2764 中,然后分别将 U₇ 中的程序和 2764 中程序读入 PC 机内存,经编辑后,使其存放在各自规定的地址上,形成新的 U₇ 内容,用 PC 机编程卡写入 27256 芯片中即可。

如果是用机器语言编写的应用程序,存放和调出的基本方法同上,但不用对零页标志单元进行置数,校验后,直接转到程序入口地址即可。

怎样启动和运行应用程序呢?如果想保留 CEC-I 机的通用计算机功能,那么只要按下“LG”键,即可启动,运行专用程序。要退出或中途想停止执行,只要按下 CTRL-Reset,即返回 BASIC 状态。但若想使其真正成为一台专用计算机,尽量减少不必要的操作,且一开机就自动进入专用程序,那就需对 U₇ 中的程序内容作以下修改。

\$ F1CB 3C→0C

\$ F1CD D4→D1

同时将 \$D2D1 起始的予置程序中最后一句 \$D2FF-4C 3C D4 改成: \$D2FF-4C 0C D1, 以上应在写 27256 前就修改好, 一并写入 U₇ 中。这样, 开机后, 就能自动进入专用程序的执行。

CEC-I 机冷启动或进入汉字状态后, 会在屏幕上出现提示信息, 启动中文专用程序后, 由于使用了高分辨第二页, 屏幕上出现杂乱无章的横道、竖道, 会使人感到很不舒服, 若欲去掉, 可在 \$D23C 程序中加一段清 \$4000~\$5FFF 的程序, 改为:

```
$D23C-20 58 FC A9 40 85 FD A9 00
$D245-A8 85 FC 91 FC C8 D0 FB E6
$D24E-FD A2 60 F4 FD D0 F3 60
```

同时在 \$D300 子程序中, 将 \$47B 置成 3, 或在专用程序中, 程序头上加一条 POKE 1147, 3 语句, 这样开机后, 就会“干净”地进入专用程序。

“LG”是系统的保留字, 存放在内存 \$C498 和 \$C499 中。若想改成自己所希望的保留字, 如“JS”, 则只需将 \$4A, \$D3 放入 \$C498, \$C499 中。这以后, 按下“JS”键, 就可以启动专用程序了。

用栈结构优化程序

浙江绍兴一中 陈合力

《计算机教与学》90 年第 6 期上刊登《用微机模拟十一 * / ^ () E 运算》是采用双向链表结构。笔者用栈结构优化了程序。改进后的程序不但比原来的短, 而且运算速度要快得多。

程序说明:

5 句, 输入待求的算术表达式, 如 $1+2*(3-4)/(7-8*3/2)-5^7)/8-6$ 。为保证完成全部运算, 在算术表达式后面加上一个 + 号。定义二个数组, 数组 A 存放运算符, 数组 B 存放算式表达式中的数码。

10 句, 为保证运算不误, 将 A(0) 赋值 -1。变量 M 记括号的重量。算术表达式中的数码暂存于 T\$ 中。

20 句, 把运算符用一个特定的数值表示。约定 + * / ^ 分别用 01234 表示, 在一括号内相应的数值增加 6, 二括号内增加 12, 依次类推。运算次序是根据运算符对应的数值大小, 大的优先运算。同一级运算如 (+-) 应从左到后, 故采用除以 2 取整再比较的办法, 并把数码存于 T 中。

30 句, 确定运算顺序, 出栈。

40 句, 进栈。

60~100 句, 具体执行运算并返回。

```
5 INPUT A$:PRINT A$="":A$=A$+"+":DIM
A(100),B(100
```

```
10 A(0)=-1:FOR I=1 TO LEN(A$):K$=MID$(A$,I,1):M=M+(K$="("-(K$=")"):IF K$="
"."OR K$="E"OR K$>="0"AND K$<="9" THEN T
$=T$+K$
```

```
20 Z=(K$="+")+2*(K$="-")+3*(K$
="*")+4*(K$="/")+5*(K$="^"):ON Z=0
GOTO 50:Z=Z+M*6-1:T=VAL(T$):T$=""
```

```
30 IF INT(Z/2)<=INT(A(P)/2) THEN ON A(P)-
INT(A(P)/6)*6+1 GOSUB 60,70,80,90,100:P=P-1:
GOTO 30
```

```
40 P=P+1:A(P)=Z:B(P)=T
```

```
50 NEXT:PRINT T:END
```

```
60 T=B(P)+T:RETURN
```

```
70 T=B(P)-T:RETURN
```

```
80 T=B(P)*T:RETURN
```

```
90 T=B(P)/T:RETURN
```

```
100 T=B(P)^T:RETURN
```

寻找无理数 e 的约率和密率

上海崇明县城中学 施为钢

大家知道无理数 π 的约率为 $\frac{22}{7}$, 密率为 $\frac{355}{113}$, 那么是否也能寻找到无理数 e 的约率和密率呢?

我们知道, 数列 $\{(1+\frac{1}{n})^n\}$ 的极限为 e 但是, 这个数列的分子分母递增太快而且向 e 逼近的速度很慢, 如它的第七项为 $\frac{2097152}{823543} = 2.546\dots$, 由此看出, 我们不能借助于这个数列。

要想找到分子分母不很大而且又非常接近于 e 的分数, 我们不妨作如下的思考:

设 ϵ 是一个很小的正数, 如果存在分数 $\frac{p}{q}$ 非常接近于 e (其中 p, q 是自然数), 那么应该满足不等式

$$|e - \frac{p}{q}| < \epsilon \text{ 即 } |eq - p| < q\epsilon \dots\dots (*)$$

由于 $q\epsilon$ 很小, 所以满足 (*) 式的自然数 p 实际上是最接近于 eq 的自然数, 这可以分两种情况讨论:

当 $eq > p$ 时 $p = \text{INT}(eq)$ 代入 (*) 式得

$$eq - \text{INT}(eq) < q\epsilon \dots\dots (1)$$

当 $eq < p$ 时 $p = \text{INT}(eq) + 1$ 代入 (*) 式得

$$\text{INT}(eq) + 1 - eq < q\epsilon \dots\dots (2)$$

注: 以上 INT(eq) 表示不超过 eq 的最大整数。

经过以上分析, 我们把寻找接近 e 的分数问题转化成寻找满足 (1) 式或 (2) 式的自然数 q 的问题, 可以用 BASIC 语言编写一段程序, 让计算机去寻找答案。

程序说明: 10 语句行用来键入分母 q 的上界 M; 以及 EP (即 e) 的值; 210 语句行的作用是, 每当找到一个接近 e 的分数后, 马上修改 EP 的值, 从而保证了以后找到的分数更近于 e。

由程序运行的结果看出, 接近 e 的分数很多, 我们不妨把其中的 $\frac{19}{7}$ 作为约率, 把 $\frac{49171}{18089}$ 作为密率。

如果对程序的 15 语句稍作修改, 就可以用来找出

与其它无理数相接近的分数。

```
10 INPUT M,EP
15 E= EXP (1)
20 FOR Q=1 TO M
30 X =E * Q;D= EP * Q;P=INT(X)
40 IF X-P<D THEN GOSUB 200;GOTO 60
50 IF P+1-X<D THEN P=P+1;GOSUB 200
60 NEXT Q
77 END
200 PRINT P;"/";Q,"=";P/Q
210 EP=ABS (E-P/Q);RETURN
]RUN
? 20000,0.005
19/7          =2.71428572
49/18         =2.72222222
68/25         =2.72
87/32         =2.71875
106/39        =2.71794872
173/71        =2.71830986
685/252       =2.71825397
878/323       =2.71826625
1071/394      =2.71827411
1264/465      =2.71827757
1457/536      =2.71828358
2721/1001     =2.71828172
12341/4540    =2.71828194
15062/5541    =2.7182817
17783/6542    =2.71828187
20504/7543    =2.71828185
23225/8544    =2.71828184
25946/7545    =2.71828182
49171/18087   =2.71828183
```

怎样节省 CEC-I 存储空间

杭州自动化研究所 尹向阳 梁 曦

采用 CEC-I BASIC 语言开发数据量大的管理软件常常感到内存和软盘容量不足,影响了这些软件的实用性。笔者在实际工作中,针对具体情况,采用了多种节省存储空间的有效方法。本文介绍其中三种效果明显并具有一定普遍性的方法。

一、用整型数组代替实型数组

CEC-I BASIC 的实型数组的每个元素占五个字节,数值表示范围达 $\pm 9.999 \times 10^{37}$,整型数组的每个元素占二个字节,数值表示范围为 ± 32767 。如果把本来要用实型数组存储的数据,经一定处理能用整型数组存储,则可节省五分之三的存储空间。例如工资额、学生成绩这类数据,一般情况下要用实型数表示,但经某种变换(如乘 100)就变成整数,而且数值大小在整

型数的表示范围内,这类数据就可用整型数组存储。参加运算前,从整型数组中读出,再经逆变换(如除 100)即可恢复原值。

有些数据(如工资总额)经上述处理后,会超出整型数的表示范围,但仍是小于 65536 的正整数,这时可以把整型数的表示范围调整到 $[0, 65536]$,即把大于 32767 的整数用负整数表示,方法如下:

设 N 是整数,且 $0 \leq N \leq 65536$

把 N 存入整型数组前,执行语句

```
10 N=N-65536*(N>32767)
```

从整型数组中读出,执行语句

```
10 N=N+65536*(N<0)
```

二、以 B 类文件代替顺序或随机文件

CEC-BASIC 的顺序文件和随机文件都是以 ASCII 码形式存储在软盘中的,占用空间大,存取速度慢。在处理诸如职工档案、学生成绩等管理对象比较固定的管理软件中,可以用 B 类文件代替顺序文件或随机文件。我们知道,B 类文件是以文件名、内存地址和文件长度三要素进行存取的,而对数组的操作完全不涉及这些信息。怎样建立 B 类文件和数组之间的联系?现在假定有这样一个程序:

```
10 DIM A(200),B%(50,12),C(40)
```

```
20 A=PEEK(107)+256*PEEK(108)
```

```
30 B=A+(200+1)*5+7
```

```
40 C=B+(50+1)*(12+1)*2+9
```

A 数组定义为实型数组,共有 201 个元素,占用 1005 个字节;B 数组定义为二维整型数组,共有 51×13 个元素,占有 1326 个字节。

每个数组都有一个表头,用以描述该数组的特征,表头的长度由数组的维数决定。这里 A 数组的表头占 7 个字节,B 数组的表头占 9 个字节。因此,A 数组数共占 1012 个字节,B 数组共占 1335 个字节。

零页的 \$6B, \$6C 二个单元是指向数组区域开始地址的指针,语句 20 给 A 赋的值就是数组区域的首地址,也就是 A 数组在内存中分配的首地址。B、C 分别是 B 数组和 C 数组在内存中分配的首地址。

知道了数组在内存中的首地址和字节数,再命名一个文件,就可以用 BLOAD、BSAVE 命令存取数组。

三、压缩汉字引荐符

CEC-BASIC 语言处理的汉字是用三字节引荐符方式表示的,其汉字内码格式为 7F+区码+位码。这里的区位码和国标区位码略有区别。

对于人名、项目名称等用纯汉字表达的字符串,用 INPUT 语句接受汉字字符串后,用 ASC、MID\$ 函数把区码和位码转换为整型数存入数组。在显示或打印前自动拼接上引荐符 7F,这样既节省了存储空间,也不影响显示和打印速度。

转 让

中华学习机,日本原装驱动器,打印机,打印卡等。学习机已用半年,其余皆暂新。原价 2376 元,愿以 1800 元转让,并送所有资料与软件。也可单项转让,欲者从速。

地址:北京西单东斜街 53 号杨建鸿 邮编:100032 电话:602.2238

ProDOS 磁盘操作系统入门(续)

廖 凯

5. CHAIN

格式: CHAIN 路径名[,@行号][,S#][,D#]

CHAIN/卷名/子目录/BASIC 文件名

功能:此命令可使程序沿用上一个程序所产生的变量值。在一个 BASIC 程序太大而内存不够时,我们将程序分成一些易处理的程序段,用 CHAIN 命令运行每个程序段。当 CHAIN 时,第一个程序段所有变量包括名称及其值被保存在内存内,而后清除程序,接着执行第二个程序,使用前面程序所留下的变量值。你可以再次使用 CHAIN 命令,返回到第一个程序段。注意不能重复定义 CHAIN 程序的数组。你可以使用“@行号”参数来 CHAIN 一个程序的一部分。

@行号:使 ProDOS 在指定的行号执行。例如,设为“,@95”,则程序会在行号 95 开始,如果没有此参数,那么 ProDOS 将从最小的行号开始。

在立即方式下,CHAIN 命令用在程序除错时才有实际意义。通常,CHAIN 是用于一个程序内的。

6. CLOSE

格式: CLOSE [路径名]

CLOSE/卷名/子目录/文本文件名

功能:此命令用于程序结束 READ 或 WRITE 操作之后,关闭打开的文件。如果使用此命令而没给出路径名时,所有打开着的文件都将被关闭。在文件被关闭之前,1K 字节的字符缓冲区将被清除。自 ProDOS 在 512 字节的区域内读写以后,CLOSE 命令将清除少于 512 字节的数据。注意在文本文件使用之后,一定要关闭文件,否则数据会丢失。

7. FRE

格式: FRE

功能:此命令是用于快速收集无用存储单元。此命令只能在程序调用方式下使用。在立即执行方式下,FRE 是一个 APPLESOFT 命令,并必须放在 PRINT 语句的后面,结果是一个在屏幕上所显示的数字。具有 64K 的计算机这数字是 -29188,这数字表示供给用户可使用空闲 RAM 的字节数,-29188 相应的正数是 36348(加 65536 所得的结果),约为 35.5K 字节。

8. IN# 和 PR#

格式: IN# 槽口号[,A#]

PR# 槽口号[,A#]

功能:这两个命令除了用来处理输入和输出的接口控制外,还可与机器语言沟通,重新规划 I/O 口。

如果你正使用一个 80 列卡,在改变输出到另一个外围槽口之前,一定要关闭 80 列卡,其命令为: ESC CTRL _ Q

下面举例说明:

如果在内存地址 768(\$300)有一子程序,你想发送输出到该子程序,使用的命令为:

PR# A768 或 PR# A\$300

输出子程序可以被指定给一个槽口,访问该槽口将被改变方向到该子程序。

PR# 4,A768 恢复为 PR# 4,A\$C400

内存分配给每个扩展槽的起始地址是 \$C#00, # 是槽口号,你也可以将所有送到一个槽口的输出均被转到另一个槽口之内,要转换槽口,就使用:

PR# 槽口号,A\$C#00 (#是槽口号)

子程序结束工作时,通过用 PR# 0 来恢复控制权到控制台。用 INT# 命令来使用相同情况的设备以接收输入信息。

9. OPEN

格式: OPEN 路径名[,S#][,D#]

OPEN/卷名/子目录/文本文件名

功能:此命令用于在 READ 或 WRITE 一个文件之前,打开一个文本文件。此命令只能用在程序调用方式下。ProDOS 允许用户同时打开八个文件。当每个文件被打开时,ProDOS 自动分配 1K 字节的缓冲区空间(1024 字节)。在首次 OPEN 一个文件时,ProDOS 在指定的目录内产生一个类型为 TXT 的文件。

10. POSITION

格式: POSITION 路径号,F#/,R#

POSITION/卷名/子目录/文本文件名,R#

功能:允许用户存取一个文件内的任何字段(数据域)或字节。在使用此命令之前,文件必须是打开的。此命令只能用在程序调用方式下,在文件内通过读取 RETURN 代码,将从当前的位置向前移动指针,这指针被定位在第 #RETURN 后的第一个字节上。

F# / R#:是先读取而后放弃并被 # 所指定的数个字段(数据域)。为了与 DOS3.3 兼容,F#(字段)与 R#(记录)是相等的。

11. READ

格式: READ 路径名[,F#][,B#]

READ/卷名/子目录/文本文件名

功能:通知 ProDOS 读取一个文本文件的数据。此命令只能用在程序调用方式下。READ 命令允许用户使用 B# 或 F# 参数,以向前定位若干字节或字段(数据域)。要被读取的起始位置,可以由用户来指定。在 READ 文件前,必须先将此文件打开。在使用 READ 以后,将一直保持到给出另一个 ProDOS 命令为止。

F#:是将被读取而后放弃的字段(数据域)的

数量。回车代码(CR)是用于指示被读取的字段(数据域)的号码。

B#是将被读取而后弃的字节或字符。

12. RUN

格式: RUN 路径名[,L 行号][,S#][,D#]

RUN/卷名/子目录/BASIC 文件名

功能:此命令用于装入并运行一个 BAS 类型的 BASIC 文件。用户可以经指定行号有选择地运行部分程序。

L#:是用于指定要被执行的程序内的行号。

13. WRITE

格式: WRITE 路径名[,F#][,B#]

WRITE/卷名/子目录/文本文件名

功能:是通知 ProDOS 将数据写入磁盘(使用一个后续的 PRINT 语句)。在 WRITE 文件之前,必须先 will 文件打开。使用此命令后,将一直保持到给出另一个 ProDOS 命令为止,此命令只能用在程序调用方式下。

B#和 F#参数能用于向前定位若干字节或字段(数据域)于执行 WRITE 操作之前。B#和 F#参数的含义见 READ 命令的介绍。

三、ProDOS 取消的命令

1. FP 和 INT

ProDOS 不再支持 INTEGER BASIC,所以没有必要再在两个 BASIC 版本之间作转换了。

2. INIT

ProDOS 已将磁盘格式化子程序放置在 FILER 文件内。如果将它作为一个正规命令放入 ProDOS 的话,那么这格式化程序会占用很大的空间。

3. MAXFILES

ProDOS 会自动地分配缓冲区空间给八个打开的文件,没有必要再用 MAXFILES 分配缓冲区的空间。

4. MON 和 NOMON

为了使 ProDOS 尽可能地少占内存,所以将 MON 和 NOMON 从解释程序中删去了。

四、ProDOS 的其它命令

1. DELETE

格式: DELETE 路径名[,S#][,D#]

DELETE/卷名/子目录/文件名

功能:从磁盘上删除一个文件。此命令可以删除任何类型的文件。在删除一个目录文件时,此目录内必须不包含任何文件。

2. EXEC

格式: EXEC 路径名[,L#][,S#][,D#]

EXEC/卷名/子目录/文本文件名

功能:此命令用于从磁盘上执行一个顺序文本文件。

L#:可用此参数指定 EXEC 执行整个文件或文件的一部分。例如:L105,是从顺序文件中第 105 字段开始执行。

3. LOAD

格式: LOAD 路径名[,S#][,D#]

LOAD/卷名/子目录/BASIC 文件名

功能:此命令是将磁盘上的一个 BASIC 文件装入内存。

4. LOCK 和 UNLOCK

格式: LOCK 路径名[,S#][,D#]

LOCK/卷名/子目录/文件名

UNLOCK 路径名[,S#][,D#]

UNLOCK/卷名/子目录/文件名

功能:LOCK 命令用于给一个文件上锁,使其内容不能被改变。对于锁着的目录文件来说,只能保护空目录不被改变,而其包含的文件可以被改变。上锁标记是一个“*” (星号),它放在文件名的左边。UNLOCK 命令用于给一个锁着的文件解锁。

5. RENAME

格式: RENAME 路径名, 路径名[,S#][,D#]

RENAME/卷名/子目录/文件名,/卷名/子目录/文件名

功能:此命令用于更改文件名。注意“,”(逗号)必须放在两个路径名之间。在程序调用下使用此命令时,这逗号必须放在引号外边。

6. SAVE

格式: SAVE 路径名[,S#][,D#]

SAVE/卷名/子目录/BASIC 文件名

功能:此命令是将已建立在内存中的 BASIC 程序存储到磁盘上。在磁盘上的文件类型为 BAS。如果在存盘时显示一个 DISK FULL 信息,则表示磁盘已满,只有程序的最初部分留在磁盘上。

(未完待续)

数制转换简法

福建 南平师范 李锋

有时临时需将 10 进制数转换为 16 进制数,编个程序总感到麻烦。下面介绍一种直接转化的方法(以十进制数 768 转为 16 进制数为例)

1. 在 BASIC 状态下键入 NEW

2. 输入 768?

3. 进入监控,查看 803、804 单元的内容为 00 03,由程序在内存的存放规则可知 $(768)_{10} = (300)_{16}$

需要强调的是,需转化的数据一定要做为行号,而不能放在语句体中。

逆用以上思路,可实现 16 进制到十进制的转换。

求 援

我部为修理福建产 LQ-1500K 电源,急需 3CT2KAT 可控硅元件及其代用器件。

浙江杭州延安北路 浙江省机电设备公司办公自动化经营部 刘琳 邮编:310006

也谈扩展 COPY II PLUS 4.0

广西柳州市第二中学 关炳坤

贵刊 1990 年第 8 期刊登了牛保军同志的文章《使 COPY II PLUS 4.3 工作到 41 磁道》。我参照文中介绍的方法,对《COPY II PLUS 4.0》进行了修改,结果未能如愿。原因是《COPY II PLUS 4.0》是以文件的形式存在于磁盘上的,由于每个用户得到软件的途径不同,故存放在磁盘上的位置也不同,按文中的方法修改自然不能成功。

我对该软件进行了分析,发现只要修改程序中的一些参数,就可使它的工作范围扩大到 \$29 磁道。为了使大家能方便地分析源程序,特把《COPY II PLUS 4.0》各功能的入口地址及相应单元的作用列于下表:

序号	功能	入口地址	相关地址	原值	新值
1	CATALOG DISK	\$1203			
2	COPY	\$1415	\$145A	\$44	\$52
3	BIT COPY	\$18BE	\$1F0C	\$23	\$29
4	DELETE	\$1905			
5	LOCK/UNLOCK	\$1A60			
6	RENAME FILES	\$1A71			
7	FORMAT DISK	\$1A97			
8	VERIFY	\$1B99	\$1C2E	\$23	\$29
9	TRACK/SECTOR MAP	\$202C	\$212D	\$8C	\$A8
10	VIEW FILE	\$22AD			
11	FIX FILE SIZES	\$251D			
12	CHANGE BOOT PROGRAM	\$26A4			
13	UNDELETE FILES	\$2743			
14	SECTOR EDITOR	\$28A4	\$2B16	\$23	\$29
15	NEW DISK INFO	\$2BFB			
16	BOOT DISK	\$2C6C			

修改源程序的方法如下:

1. 用 BLOAD COPY II PLUS 4.0 把程序调入内存,按表中新值修改相应单元的值(序号为 3 的不改);
2. 从 \$4800 处键入下面程序:
4800:A9 2A 8D FE BE 8D EF B3 0A 0A 8D B5 AE
20 18 11 60
3. 用 BSAVE COPY II PLUS 4.0, A \$7F8, L \$4019 把程序存盘;
4. 用 BLOAD COPY II PLUS BIT COPY 4.0 把程序调入内存,把 \$1F0C 的值改为 \$29;
5. 用 BSAVE COPY II PLUS BIT COPY 4.0, A \$900, L \$2808 存盘。

完成了以上五步,就可使 COPY, BIT COPY, FORMAT DISK, VERIFY, TRACK/SECTOR MAP, SECTOR EDITOR 等六项功能工作到 \$29 磁道。

从 CP/M 转移文件 到 APPLE DOS 下

成都 201 信箱 方福兴

在 APPLE 机中,将文件从 APPLE DOS 转移到 CP/M 下可用 APDOS 方便地进行,而从 CP/M 中的文件转移至 APPLE DOS 下没有专门的软件,转移起来就不那么方便了,特别是 ASCII 码文件。有时要转移一个文件还得从头键入。这里介绍一种方法,可以方便地将文件从 CP/M 中转移至 APPLE DOS 下。

一、ASCII 码文件的转移

先准备一张 HELLO 程序为如下内容的 APPLE DOS 盘,然后将 CP/M 下要转移的文件装入内存,再放入有该程序的盘到主驱动器中,热启动,运行该程序,输入长度、文件名,运行完毕则转移结束。例:

```
10 D$=CHR$(4):INPUT"LEN:";I1:I1=4352+I1
20 INPUT"FILE NAME:";F$
30 PRINT D$;"OPEN";F$
40 PRINT D$;"WRITE";F$
50 FOR I=4352 TO I1:A=PEEK(I):IF A=10 THEN 58
52 PRINT CHR$(A);
58 NEXT
60 PRINT CHR$(4);"CLOSE";F$
1. 放入要转移的文件到内存中(设文件名为 F1)。
A>DDT F1
DDT VERS 2.2
NEXT PC
0180 0100
```

2. 放入具有该 HELLO 程序的盘到主驱动器中,热启动,在 LEN:后键入如上显示差值的十进制值(即 $(0180)_H - (0100)_H = (80)_H = 128$),再键入要存入磁盘的文件名,运行完毕则转移结束。

3. 可从 RUN EDASM 装入已转移的文件,仅在文件的头和尾几个字节需修改一下。

二、二进制代码的转移。

先在 CP/M 下按上述方法装入内存,再放入具有较短 HELLO 程序的 APPLE DOS 盘到主驱动器中,热启动,直接用 BSAVE 文件名, A \$1100, L \$长度,该长度为上述方法计算的十六进制值。

用这种方法可方便地进行文件的转移。特别是目前单片机的普遍应用,在 APPLE DOS 下的仿真源文件与 CP/M 下的仿真源文件可以结合 APDOS 方便地进行相互转移,这对使用者将带来很大的方便。

多功能 CEC-I 高速排序

沈阳冶金机械专科学校 187 号 王庆顺

关于排序问题,大多使用 BASIC 语言程序来处理。但由于 BASIC 语言运行速度的限制,占机时间都较长,且功能单一,适用性不强。本排序程序用机器语言编写的,特点如下:

1. 速度快,对 1000 个数据排序仅需一、二分钟。
2. 适用于对实型、整型和字符串变量排序。
3. 能对一、二维数组排序,对二维数组的某一列排序时,排序后同行的相关数据不变。

程序一是 BASIC 主程序,程序二是机器语言子程序。现以二维实型数组 A(M,N)为例介绍使用方法。

首先按程序二给出的机器码输入 \$9000 开始的内存单元中,然后用“BSAVE PX.OBJ,A \$9000,L \$190”存盘,供程序一调用。

20 语句:定义数组大小,该语句必不可少。

40 语句:建立“&”跳转地址。

41 语句:为程序二设置参数,并由 45、46 和 47 语句置入指定内存单元。L2 和 L3 的计算公式是不变的。L1 为数据类型代码,取值为整型 2,实型 5,字符串 3。

42 语句:设置排序参数供 50 语句中使用。K2 表示按数组第 K2 列元素进行排序。K1 和 K3 表示两个方面的意义:1、第 K2 列中有 (K3-K1)+1 个元素参与排序;2、排序由 K1 行第 K2 列开始到 K3 行第 K2 列结束。调整 K1 和 K3 的值,可以方便地对数组第 K2 列的任意部分进行排序。各参数有如下要求:

$K1 < K3$ $0 \leq K1 \leq M$

$0 \leq K3 \leq M$ $0 \leq K2 \leq N$

当 $K1=0, K3=M$ 时, K2 列所有元素都参与排序。

50 语句:调用程序二进行排序。

对一维数组排序时,要把程序一中的二维数组改

成一维数组,把 50 句改成如下形式:

&A(K1),A(K1),A(K3)

各参数取值与上述二维数组相同,但在 41 语句前应加上“N=0”。改变排序数组维数或类型时,20、30、60 语句也要相应改变。

程序一是实型数组 A(M,N),对第 1 列所有元素都参与排序的范例。对于其他情况,用户可以参照程序一编程。

程序一

5 HIMEM:36863

7 PRINT CHR\$(4);“BLOAD PX.OBJ,A \$9000”

10 INPUT “M=,N=”;M,N

20 DIM A(M,N)

30 FOR J=0 TO M:FOR I=0 TO N:A(J,I)=100 *
RND(1):PRINT A(J,I);“”;NEXT:PRINT:NEXT

35 PRINT CHR\$(7)

40 CALL 36864

41 L1=5:L2=(M+1)*L1:L3=N+1

42 K1=0:K2=1:K3=M

45 POKE 249,L1

46 POKE 29,INT(L2/256):POKE 28,L2-INT(L2/256)

* 256

47 POKE 233,INT(L3/256):POKE 232,L3-INT(L3/
256)*256

50 &A(K1,0),A(K1,K2),A(K3,K2)

55 PRINT CHR\$(7)

60 FOR J=0 TO M:FOR I=0 TO N:PRINT A(J,I);“
”;NEXT:PRINT:NEXT

70 END

程序二

9000-A9 0B 8D F6 03 A9 90 8D

9008-F7 03 60 20 E3 DF 84 1F

9010-85 1E 20 BE DE 20 E3 DF

9018-84 07 85 06 38 E5 1E 85

9020-1E 98 E5 1F 85 1F 20 BE

9028-DE 20 E3 DF 84 1B 85 1A

9030-A4 07 A5 06 84 09 85 08

9038-20 07 91 A5 F9 C9 05 F0

9040-0A C9 02 F0 0C 20 A7 90

9048-4C 54 90 20 98 90 4C 54

9050-90 20 EC 90 C9 01 D0 03

9058-20 07 91 A5 09 A4 08 20

9060-76 91 D0 0E A4 09 A5 08

9068-20 6B 91 84 09 85 08 4C

9070-3B 90 20 10 91 A5 07 A4

9078-06 20 76 91 D0 0E A4 07

9080-A5 06 20 6B 91 84 07 85

9088-06 4C 30 90 60 18 D8 65

9090-1C AA 98 65 1D A8 8A 60

9098-A4 FD A5 FC 20 F9 EA A4

90A0-09 A5 08 20 B2 EB 60 A0

90A8-00 B1 08 85 A8 D1 FC 90

90B0-0A F0 0C B1 FC 85 A8 A9

90B8-FF D0 06 A9 01 D0 02 A9

90C0-00 85 A7 C8 B1 FC 85 A5

90C8-B1 08 85 42 C8 B1 FC 85

90D0-A6 B1 08 85 43 A0 00 B1

90D8-42 D1 A5 F0 07 A9 FF B0

90E0-02 A9 01 60 C8 C4 A8 D0

90E8-EE A5 A7 60 A0 00 B1 08

90F0-D1 FC F0 02 D0 09 C8 A2

90F8-00 B1 08 D1 FC F0 06 A2

9100-FF B0 02 A2 01 8A 60 A5

9108-08 85 FC A5 09 85 FD 60

9110-A5 06 A4 07 20 88 91 86

9118-3C 85 3D A5 FC A4 FD 20

9120-88 91 86 3E 85 3F A0 00

9128-84 42 84 43 B1 3E 48 B1

9130-3C 91 3E 68 91 3C C8 C4

9138-F9 D0 2C E6 42 D0 02 E6

9140-43 A5 42 C5 E8 F0 02 D0

9148-06 A5 43 C5 E9 F0 1B A4

9150-3F A5 3E 20 8D 90 84 3F

9158-85 3E A4 3D A5 3C 20 8D

9160-90 84 3D 85 3C A0 00 4C

9168-2C 91 60 D8 18 65 F9 AA

9170-98 69 00 A8 8A 60 C5 1B

9178-90 0B F0 02 B0 03 98 C5

9180-1A A9 FF B0 02 A9 00 60

9188-38 E5 1E AA 98 E5 1F 60

小学生速算训练

武汉江岸车辆厂银行 李永升

本程序可由学生自己设定数值的大小,训练项目多,适应不同年级的小学生运用。

70~100 句为选择类型,做多少题,任意设定数值,K 计算出每题分值。

130 句 根据选择的类型转随机出题的子程序。

140~180 句 对答案进行判断,加分。

210 句 计算答案,打印被乘数,乘数。

280~310 句 出两数相减的随机数。

390~420 句 出随机平均数。

590~620 句 出随机若干个连续自然数。

程序在 CEC-I 中华学习机上通过。

```

10 PRINT CHR$(4) ("PR#3");PRINT
20 T=0;HGR2;HOME
30 PRINT"1. 两数相加","2. 两数相减","3. 多位数相加",
"4. 求平均数","5. 两数相乘","6. 两数相除","7. 11-99 的平方",
"8. 连续整数相乘","9. 尾数 5 的平方",
40 PRINT"10. 尾数 5 的乘法","11. 100 以上相乘",
"12. 100 以下相乘","13. 求若干个连续整数之和"
50 PRINT"14. 个位数字之和为 10 的两数相乘"
60 PRINT"15. 个位相同其余数之和为 10 的数相乘"
70 INPUT X;HOME;INPUT"你要做多少道题?(3-100)";P;K=INT(100/P)
80 INPUT"请输入位数=";M
90 IF X=5 THEN INPUT"请输入乘数=";B
100 IF X=6 THEN INPUT"请输入除数=";B
110 FOR I=1 TO P;HOME;VTAB 5;HTAB 5
120 A=INT(10^(M-1)+RND(1)*9*10^(M-1))
130 ON X GOSUB 260,280,320,390,210,380,430,450,460,530,550,570,590,470,500
140 INPUT C;IF C=Z THEN 170
150 PRINT;PRINT TAB(10)"计算错误!"
160 PRINT;PRINT TAB(8)"正确答案是:";Z;GOTO 180
170 VTAB 7;HTAB 10;PRINT"计算正确! 加";K;"分":T=T+K
180 FOR J=1 TO 1000;NEXT J
190 NEXT I
200 HOME;GOTO 220
210 Z=A*B;PRINT A*"×"B="";RETURN
220 VTAB 8;HTAB 10;PRINT"成绩=";T;"分"
230 PRINT;VTAB 9;HTAB 10;PRINT"是否还做?(Y/N)";
240 GET A$;IF A$="Y"OR A$="y" THEN 20

```

```

250 END
260 B=INT(10^(M-1)+RND(1)*9*10^(M-1))
270 Z=A+B;PRINT A"+"B="";RETURN
280 A=INT(10^(M-1)+RND(1)*9*10^(M-1))
290 B=INT(10^(M-1)+RND(1)*9*10^(M-1))
300 IF B>=A GOTO 280
310 Z=A-B;PRINT A-"B="";RETURN
320 B=INT(10^(M-1)+RND(1)*9*10^(M-1))
330 D=INT(INT(1+5*RND(1))*10^M-A+.5)
340 E=INT(INT(1+5*RND(1))*10^M-B+.5)
350 F=INT(10^(M-1)+RND(1)*9*10^(M-1))
360 Z=A+B+D+E+F
370 PRINT A"+"B"+"D"+"E"+"F;"="";RETURN
380 Z=A/B;PRINT A"÷"B="";RETURN
390 NU(1)=101+INT(850*RND(1));CC=NU(1);VTAB(3);HTAB(4);PRINT NU(1)
400 FOR J=2 TO 10;NU(J)=NU(1)+INT(20*RND(1));CC=CC+NU(J);VTAB(3+2*(J-5));HTAB(6*(J-5*(J-5))-2);PRINT NU(J);NEXT J
410 VTAB(10);HTAB(1);CALL 64578;HTAB(20)
420 Z=CC/10;PRINT"平均数=";RETURN
430 A=11+INT(89*RND(1))
440 Z=A*A;PRINT A;"的平方=";RETURN
450 B=B+1;GOTO 210
460 A=10*A+5;Z=A*A;PRINT A;"的平方=";RETURN
RETURN
470 A=INT(10^(M-2)+RND(1)*9*10^(M-2))
480 B=1+INT(9*RND(1))
490 A=A*10+B;B=A+10-2*B;GOTO 210
500 C=10+INT(9*RND(1))*10
510 B=1+INT(9*RND(1))
520 A=C+B;B=100-C+B;GOTO 210
530 A=10*INT(40*RND(1))+5
540 B=10*INT(20*RND(1))+5;GOTO 210
550 A=100+INT(20*RND(1))
560 B=100+INT(20*RND(1));GOTO 210
570 A=100-INT(20*RND(1))
580 B=100-INT(20*RND(1));GOTO 210
590 A=1+INT(300*RND(1))
600 B=5+INT(100*RND(1))
610 Z=(A+A+B-1)*B/2
620 PRINT A"+"A+1"+"A+2"+.....+"A+B-1"="";RETURN

```

更正

本刊 1991 年第 4 期《在 dBASE III 中实现多维数组功能》一文,因对作者名字辨认不清,误将“石洪泰”看成“石供泰”,谨向石洪泰先生致歉。

又:同期《工资票面张数计算程序的简化》一文,作者陈承灿先生的名字被误排成“孙承灿”,校对时未能发现,特向陈承灿先生表示歉意。

第七讲 结构数据(二)

李文兵

3. 结构

结构是结构数据的一种常用形式,它是用户自由设定,由基本数据类型构造出来的组合数据类型。

(1)结构类型的定义 结构数据类型的定义方法有两种。

〈第一种定义方法〉 其定义格式如下:

```
struct 结构数据类型名{组员表}
```

其中结构数据类型名也叫标记,是人为设定的;成员是根据需要设置的;成员的组合就是结构类型的实在模式。例如:

```
struct date{
    int month;
    int day;
    int year;
};
```

就定义了由 3 个整型数据组成的一个结构数据类型,起名就叫 date。

〈第 2 种定义方法〉 使用数据类型定义关键字定义,如:

```
typedef struct{
    int month;
    int day;
    int year;
}date;
```

这里同样定义了包含有 3 个整型数据组成的一个结构数据类型 date。注意,typedef 可以给已经存在的数据类型起新的名字,但不能定义新的存储类别。

(2)结构变量的定义 结构变量有 2 种定义方式,比如,要定义具有 date 这种数据类型的变量,方法如下。

〈第一种定义方法〉 与结构类型分开定义,即:

```
struct date d1,d2[5];
```

其中 d1 和 d2 分别是 date 这种类型的一个变量和一个数组。不过,注意 d2[5] 的每个元素均为 date 这样的结构数据,共有 5 个这样的元素。

〈第二种定义方法〉 与结构类型同时定义,如:

```
struct date{
    int month;
    int day;
    int year;
}d1,d2[5];
```

(3)结构成员的初始化与调用 结构变量定义后,便可使用成员运算符及赋值运算符,对其成员赋初值,如:

```
d1.month=2;
```

```
d1.day=7;
```

```
d1.year=1991;
```

可见,使用结构成员运算符·,便可指定结构的成员,即

结构变量·成员名

使用 printf() 和 scanf() 这两个库函数,也可以对结构成员进行输入与输出,如练习 7.11 所示。

```
A)TYPE EXP7 _ 11.C
```

```
struct date{
    int month;
    int day;
    int year;
}

main()
{ struct date d;
  scanf("%d",&d.year);
  scanf("%d",&d.month);
  scanf("%d",&d.day);
  printf("%d",d.year);
  printf("%d",d.month);
  printf("%d",d.day);
}
```

```
A)EXP7 _ 11
```

```
1991 1 23
```

```
1991 1 23
```

对于结构数组,其赋值方法与一般变量相同。例如:对于结构数组 d2[5],要对其 3 号元素 d2[3] 的各成员赋值,可按如下进行:

```
d2[3].month=5;
d2[3].day=27;
d2[3].year=1990;
```

(4)结构的嵌套使用 结构还可以做另一结构的成员,即结构可以嵌套使用。在 Turbo C 中,结构嵌套层次不限,但常用的为 2 重和 3 重,如:

```
struct date{
    int month;
    int day;
    int year;
};

struct address{
    char name[20];
    char city[50];
    struct date zip;
}p;
```

这里,在结构 address 中就嵌套了结构 date。这时,对变量 p 的赋值和调用,则如下所示:

```
p.zip.month=9;
printf("%d",p.zip.month);
```

结构嵌套的例子如练习 7.12 所示。

```

A) type exp7 __12.c
struct date {
    int month;
    int day;
    int year;
};
struct date a1, a2[5];
struct class {
    char name[10];
    struct date a3;
    int grade;
};
typedef struct {
    char address[20];
    int tel;
    float num;
    struct date a4;
    struct class a5;
} ddate;

main()
{
    printf("size of date=%d\n",
        sizeof(struct date));
    printf("size of class=%d\n",
        sizeof(struct class));
    printf("size of ddate=%d\n",
        sizeof(ddate));
}

```

```

C) exp7 __12
size of date=6
size of class=18
size of ddate=50

```

4. 字段结构

C 语言可以进行字单元内的位操作,因此,在象对待布尔(Boolean)变量那样,进行 1 位(true 或 false)操作の場合,可在一个字单元内放置多个 Boolean 变量,这样,可节省存储空间。此外,在控制外设时,也必须进行位操作。结构是适合位操作的数据类型。这样的结构就叫字段结构,也叫位域。

(1) 字段结构的定义 其定义格式如下所示:

```

struct 字段结构类型名 {
    数据类型 字段 1 名:长度
    数据类型 字段 2 名:长度
    ;
} 字段结构变量名;

```

这里,结构成员都是不满一个字的整型数据,故叫字段。字段名后(冒号后)的长度表示该字段的长度,即占几位,其值当然要小于 cpu 字长。注意,数据类型为 int, unsigned。例如练习 7.13 所示。

```

C) type exp7 __13.c
main()
{ struct bit

```

```

{ unsigned f1;3;
  unsigned f2;5;
  int f0;5;
  unsigned f3;1;
  unsigned f4;2;
  } bit __code;
  bit __code.f1=0x2;
  bit __code.f2=0x6;
  bit __code.f3=1;
  printf("%x %x %x\n",
    bit __code.f1, bit __code.f2, bit __code.f3);
  printf("%x\n", bit __code);
}

C) exp7 __13
2 6 1
2032

```

(2) 字段结构的用法及注意事项:

① 字段不存在地址,故不能使用运算符 &,因而,也无指向字段的指针。

② 在 PC 机上,字段在一个字上的分配方向是从右至左,如上例,其字段 f_1 、 f_2 、 f_0 、 f_3 、 f_4 的分布情况如图 7.3 所示。

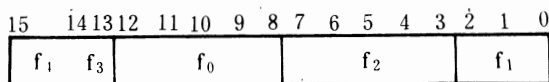


图 7.3 字段分配方向

③ 不能构造字段数组。

④ 如果字段未占满一个字存储单元,余下的位将保持原值不变。若余下的位已放不下某字段,则该字段开辟下一个存储单元使用。

⑤ 若想强制某字段放在一个字的开始部分,对 Turbo C 来说,只要在该字段的前面,放置一个长度为 1 的字段即可。

⑥ 无名字段(即无字段名)用来填空,所填的位数即该字段的长度。

上述用法如练习 7.14 所示。

```

C) type a; exp7 __14.c
struct bitfield {
    unsigned d1;3;
    unsigned d2;4;
    unsigned d3;4;
    unsigned d4;16;
    unsigned d5;8;
    unsigned :4;
    unsigned d6;5;
    unsigned d7;5;
} bit __code;

main()
{ unsigned *p;
  p=&bit __code;
  bit __code.d1=5;

```

```

bit __code.d2=10;
bit __code.d3=9;
bit __code.d5=0xaa;
bit __code.d6=12;
bit __code.d7=0x0a;
printf("%x\n", *p);
printf("%x\n", *(p+1));
printf("%x\n", *(p+2));
printf("%x\n", *(p+3));
}
C>exp7 __ 14
4d5
0
c0aa
14

```

5. 共体 (union)

共体 (union) 是由使用同一存储空间的多个变量组成的结构数据。由于其各组成变量使用同一存储空间, 故叫共体; 有些专家直接翻译为联合。其定义例子如下所示:

```

union rec{
    int var1;
    int var2;
    char var3;
    char var4;
}

```

该定义确定了一个由 var1、var2、var3、var4 四个变量组成的共体类型 rec。四个变量的存储分配, 如图 7.4 所示。

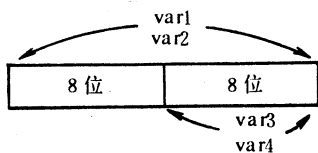


图 7.4 共体 rec 的存储分配

又如, 下面的定义:

```

union rec{
    int vari;
    char var2[2];
}

```

使变量 var1 所占用的低位字节和高位字节同时用来存储 var2[0] 和 var2[1], 其程序如练习 7.15 所示。

A>TYPE EXP7 __ 15.C

```

union rec{
    unsigned int var1;
    char var2[2];
}
main(){ union rec v;
    v.var1=0xaa66;
    printf("v.var1=%x\n",v.var1);
    printf("v.var2[0]=%x\n",v.var2[0]);
    printf("v.var2[1]=%x\n",v.var2[1]);
}
A>EXP7 __ 15
v.var1=aa66
v.var2[0]=66
v.var2[1]=ffaa

```

可见, 在共体中, 同一存储空间可被多个变量使用。

模拟 8086 cpu 寄存器的结构设在标题文件 dos.h 中。该结构如下所示:

```

struct WORDREGS{
    unsigned int ax,bx,cx,dx,si,di,cflag,flags;};
struct BYTEREGS{
    unsigned char al,ah,bl,bh,cl,ch,dl,dh;};
union REGS{
    struct WORDREGS x;
    struct BYTEREGS h;
};
struct SREGS {
    unsigned int es;
    unsigned int cs;
    unsigned int ss;
    unsigned int ds;
};
struct REGPACK{
    unsigned r_ax,r_bx,r_cx,r_dx;
    unsigned r_bp,r_si,r_di,r_ds,r_es,r_flags;
};

```

其中, 共体 REGS 是由两个结构组成的。AX 寄存器的高 8 位和低 8 位分别用 ah 和 al 表示。使用共体 REGS, 表示系统时间的程序如练习 7.16 所示。

```

A>TYPE EXP7 __ 16.C
#include "dos.h"
main()
{union REGS in,out;
    in.h.ah=0x2c;
    intdos(&in,&out);
    printf("time: %2d: %2d: %2d",
        out.h.ch,out.h.cl,out.h.dh);
}
A>EXP7 __ 16
time:16:32:14

```

(天大分校蓝智斌同志参加了程序调试)



MCS-51 单片机应用中二种 LED 显示方式

傅 篱

在单片机应用中,常常要采用多位八段 LED 显示方式。较为常见的 LED 方式,其电原理如图 1。

软件流程图如图 2。

在这种 LED 显示方式,每位 LED 是依次刷新,每位显示时间约 1ms。这样,显示电流变化较大,频率较高,当开机无键入时,LED 显示器以毫秒级时间不断刷新显示。如果系统中接有喇叭,这种方法就不适用。喇叭中

中将产生明显的交流噪声。

为此,在 MCS-51 系列单片机应用中,可利用串行 I/O 口,外接 74LS164 移位寄存器,在方式 0 的工作方式下,将显示段码数据,串行发送出去,使 LED 显示器根据需要显示相应数据。其电原理图如图 3。

软件流程如图 4。

这种 LED 显示方式和上种不同之处,在于后者为

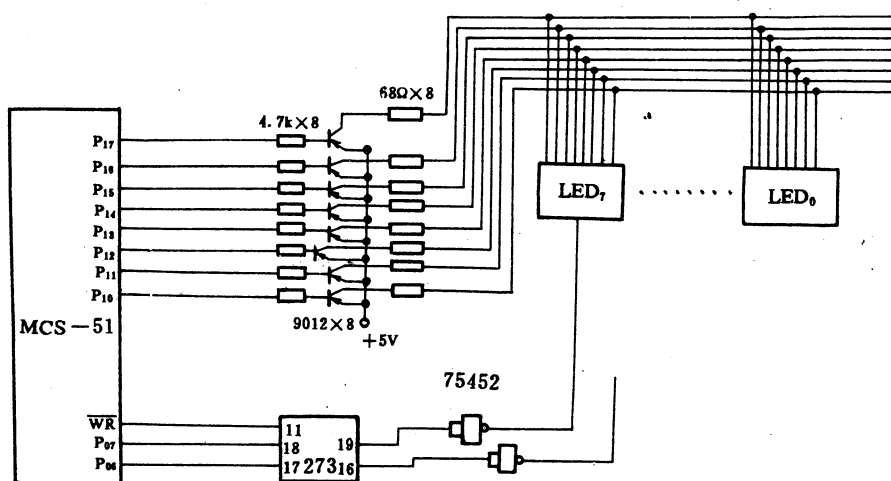


图 1

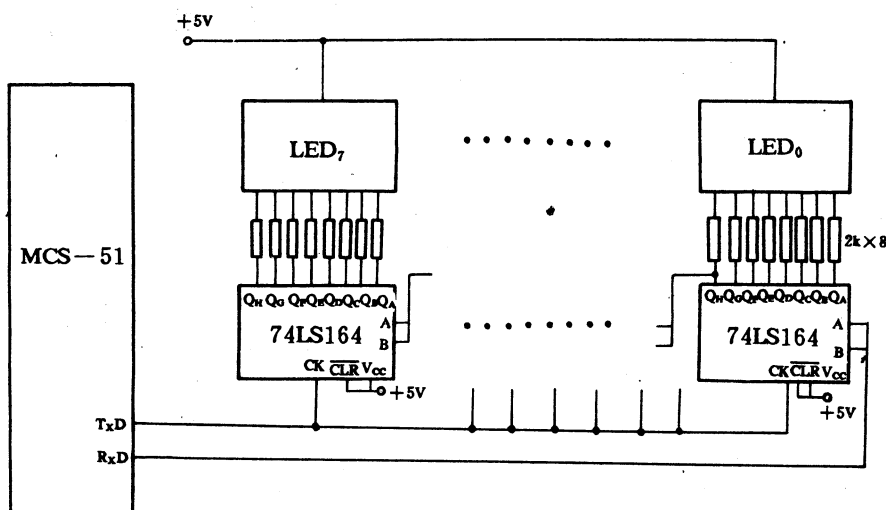


图 3

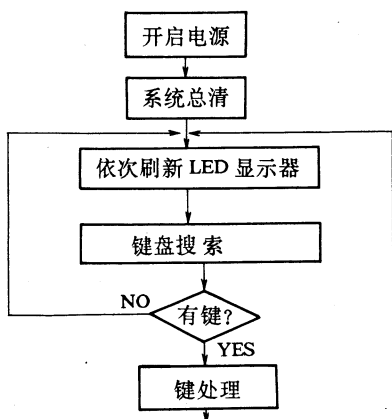


图 2

静态显示。如果 M_0 和 M_1 相同，则保持原显示数据；只有 M_0 不等于 M_1 ，显示器显示数据才需要改变。这样，LED 显示器的显示变化在秒级范围内变化，只要不按动键盘，显示器就不需要改变显示。这种显示方式虽克

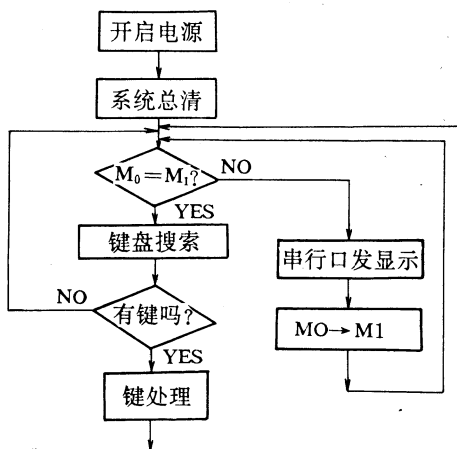


图 4

服了第一种显示方式的不足，但显示器的电流消耗却大了许多。所以，读者应根据自己设计要求，选择 LED 显示方式。

(上接 31 页)

表 3

标记	助记符	地址	机器语	注 释
START	LD A,90H	0000	3E 90	设置输入输出
	OUT(03H),A	0002	D3 03	
	LD SP,0100H	0004	31 00 01	SP←0100H
	XOR A	0007	AF	A←FFH
	CPL	0008	2F	
	OUT(02H),A	0009	D3 02	C口为高电平
	LD C,33H	000B	0E 33	33H送入C寄存器
	IM1	000D	ED 56	中断1型
	EI	000F	FB	开中断
LOOP	NOP	0010	00	保存
	JP LOOP	0011	C3 10 00	
INT	DI	0038	F3	关中断
	XOR A	0039	AF	A←00H
	RLC C	003A	CB 01	C寄存器左移
	JP NC,INT1	003C	D2 47 00	
INT2	OUT(02H),A	003F	D3 02	C口输出低电平
	CALL TIM	0041	CD 60 00	延时
	EI	0044	FB	开中断
	RETI	0045	ED 4D	中断结束
INT1	CPL	0047	2F	A←FFH
	JP INT2	0048	C3 3F 00	
TIM	LD E(60H)	0060	1E 60	延时
J1	DEC E	0062	1D	
	JP NZ,J1	0063	C2 62 00	延时常数送入E寄存器
	RET	0066	C9	

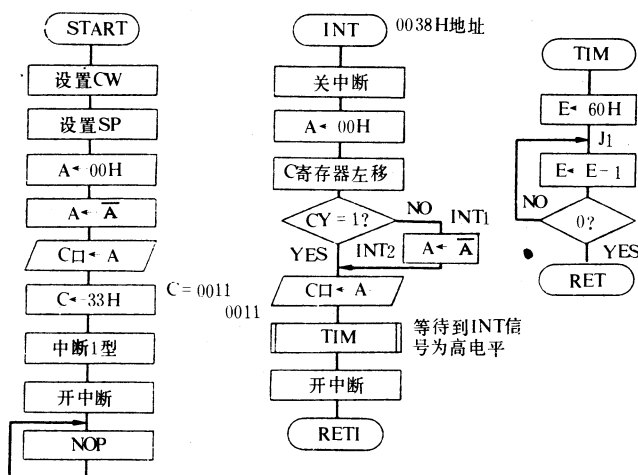


图 12

相位控制部件的应用不局限于上述事例。有兴趣的读者，不妨尝试一下。



学装微电脑

学装 μ p-80

相位控制部件的制作

易齐干

交流电为正弦波,如果使用可控硅遮断正弦波的一部份,以无级方式控制相位变化,如图1所示。再引进 μ p-80 微电脑,则能以简单的电路实现对交流电进行准确控制。

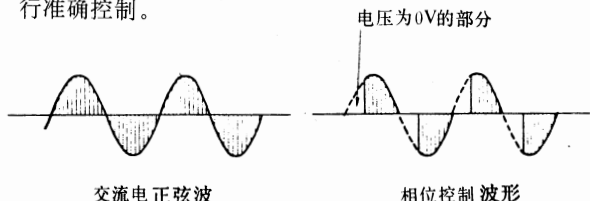


图1. 交流电正弦波和相位控制波形

1. 相位控制部件

相位控制部件电路原理图如图2所示。

图中 S102DD2 和 PC827 是夏普公司制作的光可控硅耦合器和光敏晶体管耦合器。PC827 将交流电正弦波以 0V 时序输入给微电脑。S102DD2 是通过微电脑对交流电正弦波进行相位控制。

晶体管 2SC1815 接受微电脑信号,控制驱动 S102DD2。

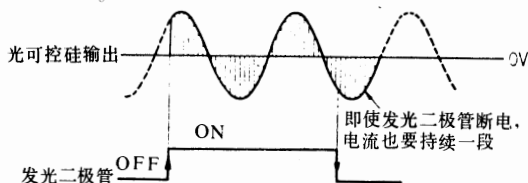


图4 光可控硅的动作

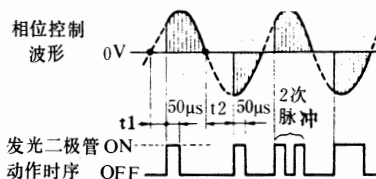


图5 相位控制的方式

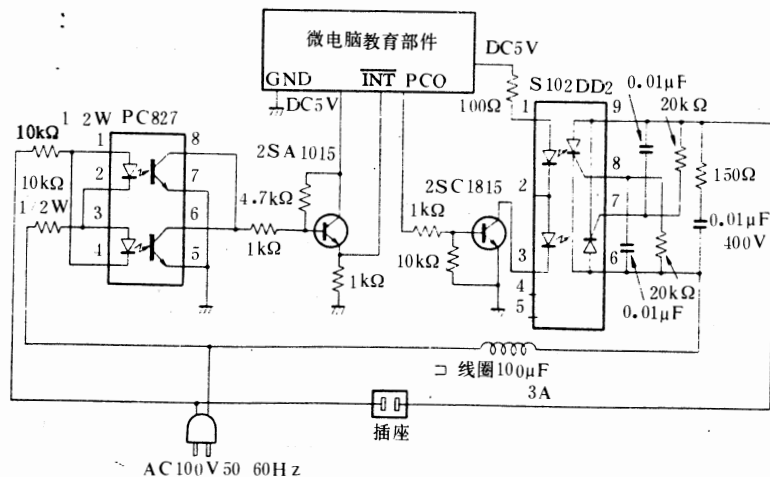


图2

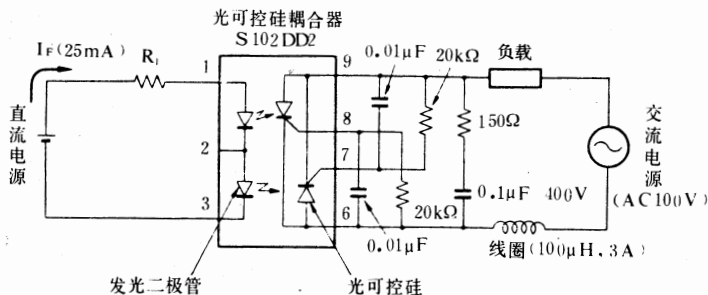


图3

(1) 光可控硅耦合器 — S102DD2

光可控硅耦合器(S102DD2)如图3所示。内部结构是两组发光二极管和光可控硅半导体装在一个盒内。如果有 25mA 的电流 I_F 由引脚 1 流向引脚 3,则发光二极管发光,打开光可控硅,引脚 6 与引脚 9 之间有交流电流通过。

引脚 6、7、8、9、之间设置电阻电容,是防止由于引脚 6 与 9 之间电压急剧增加,正弦波形增大而产生误动作。交流正弦波在通一断变化时产生噪声,为防止噪声向外部释放,电路中设置 100μH 的线圈。

引脚 1 串联的电阻 R_1 起限流作用,使发光二极管的电流在 25mA 左右。

光可控硅的特点是无触点,寿命长,高速动作,输出延续。如图4所示,发光二极管即使熄灭,交流电变为 0V 之前,光可控硅输出仍然持续一段时间。

(2) 相位控制

由光可控硅耦合器的动作很

容易理解:交流电的正弦波由 0V 开始的任意时间由微电脑驱动发光二极管产生触发脉冲实现相位控制,如图 5 所示。

触发脉冲宽度过窄,光可控硅可能没有应答。所需要的最低限度由负载种类和相位状态决定。必须通过实验验证。例如,触发脉冲宽度为 50 μ S 以上,均有应答。为提高可靠性,使脉冲连续触发两次更好。

(3)交流电 0V 检测

PC827 光晶体管耦合器也是一种半导体元件。结构也是一个盒内装有两组发光二极管和光敏晶体管。发光二极管一发光,光敏晶体管导通。

它检测交流电 0V 时序,输入给微电脑的电路原理图如表 1。

2. 应用例

(1)照明灯亮度调节

照图 6 连接 μ p80 微电脑套件和相位控制部件,使

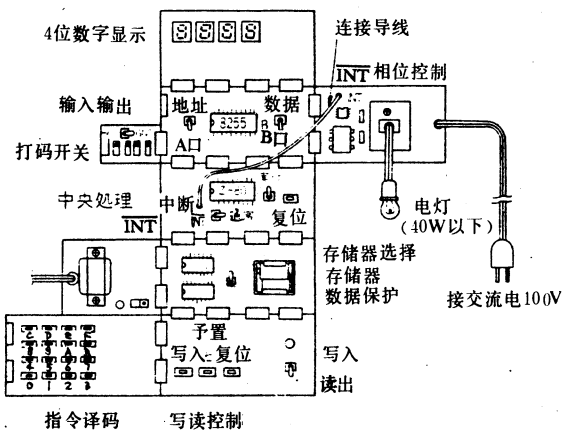


图 6

表 1

信号波形 动作内容		动作内容
1 光耦合器	① ON	(1) 随交流电电流方向(i或i')不同,发光二极管①和②交替发光
	② OFF	
2 晶体管	③ ON	(2) 光晶体管③、④接受发光二极管的发光,交替为导通状态,但是,交流电在 0V 附近,发光二极管①、②都不发光
	④ OFF	
①点	H	光晶体管③和④只要有一支导通,①点则为低电平
	L	
②点	H	②点为低电平,晶体管 2SA1015 导通,则②点(INT 信号)变为高电平
	L	

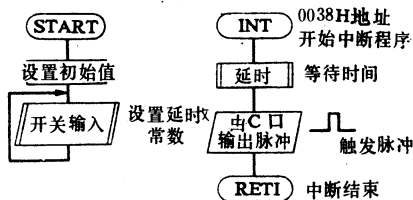


图 7

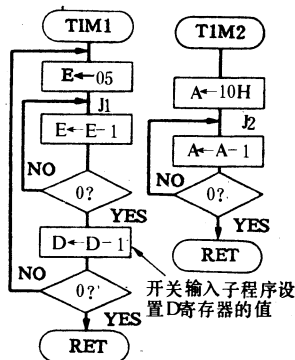
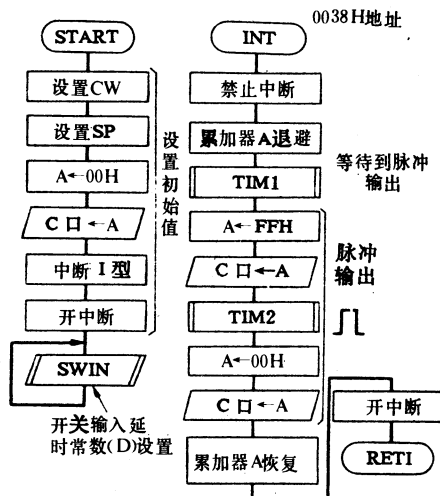


图 8(1)

用打码开关实现照明灯亮度 4 级调节。

完成这个实验很容易。由前进相位控制部件工作原理可知,交流电为 0V 的瞬间向微电脑申请中断。CPU 自动执行 0038H 地址开始的中断程序。中断程序的内容是光可控硅耦合器的触发脉冲产生程序,它出现於由打码开关控制所决定的既定时间之后。

程序流程图如图 7、图 8 所示,程序清单如表 2。

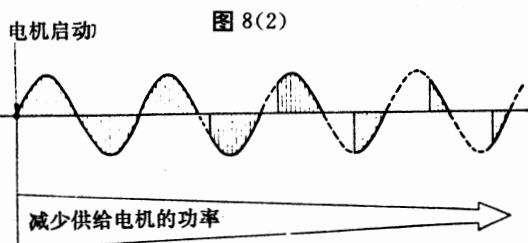
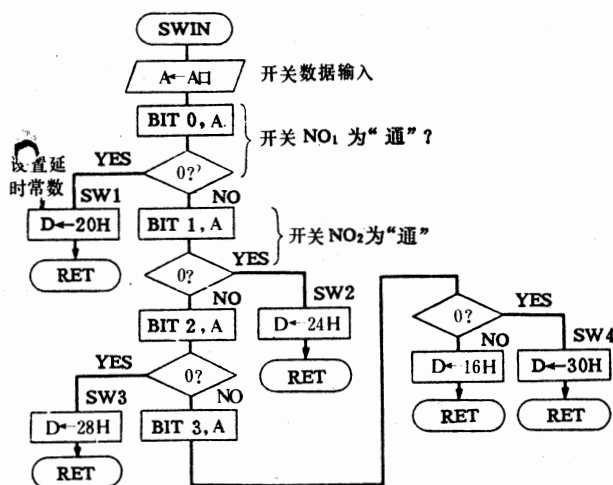


图 9

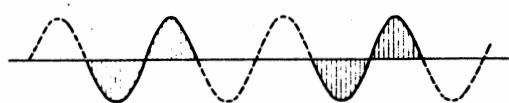


图 10



图 11

这种相位控制部件能够直接控制 40W 照明灯。

(2) 交流电机转速控制

用交流电机代替照明灯,上述程序可以控制电机的转速。

如果适当变动程序,如图 9 所示,电机启动时正弦波不变,启动之后,进行相位控制,可以实现电机低速运转。

(3) 零伏电压开关

除相位控制之外,如图 10 所示,交流电正弦波以

1 个周期为单位实现通、断控制,这就是零伏电压开关。与相位控制的波形不一样,正弦波无急躁变化,所以对外部噪声影响很小。

它主要用于加热器的温度控制,或用于对噪声干扰要求较严格的地方代替继电器。

程序流程图如图 11 所示。将 33H 放入寄存器 C,作为波形状态向外输出。在正弦波为 0V 时,每发生一次中断,寄存器 C 的数值向左移位,有进位输出时正弦波半周期为 OFF,没有进位输出时,正弦波的半周期为 ON。这样,实现交流电正弦波一个周期中为通一断状态。程序清单如表 3 所示。(下转 28 页)

表 2

标记	助记符	地址	机器语	注 释
START	LD A,90H	0000	3E 90	设置输入输出
	OUT(03H),A	0002	D3 03	
	LD SP,0100H	0004	31 00 01	SP ← 0100H
	XOR A	0007	AF	C口输出低电平
	OUT(02H),A	0008	D3 02	
	IM 1	000A	ED 56	中断 1 型
	EI	000C	FB	开中断
LOOP	CALL SWIN	000D	CD 80 00	
	JP LOOP	0010	C3 0D 00	
INT	DI	0038	F3 0	
	PUSH AF	0039	F5	保存 A 累加器
	CALL TIM 1	003A	CD 60 00	延时
	LD A,FFH	003D	3E FF 0	C口输出高电平
	OUT(02H),A	003F	D3 02	
	CALL TIM 2	0041	CG 70 00	脉冲宽度
	XOR A	0044	AF 0	C口输出低电平
	OUT(02H),A	0045	D3 02	
	POP AF	0047	F1	
	EI	0048	FB	
	RETI	0049	ED 4D	中断结束
TIM 1	LD E,05H	0060	1E 05	延时子程序
J1	DEC E	0062	1D	
	JP NZ,J1	0063	C2 62 00	
	DEC D	0066	15 0	
	JP NZ,TIM 1	0067	C2 60 00	
	RET	006A	C9	
TIM 2	LD A,10H	0070	3E 10	脉冲宽度延时
J2	DEC A	0072	3D	
	JP NZ,J2	0073	C2 72 00	
	RET	0076		
SWIN	IN A,(00H)	0080	DB 00	A口输入
	BIT 0,A	0082	CB 47	
	JP Z,SW1	0084	CA 99 00	
	BIT 1,A	0087	CB 4F	
	JP Z,SW2	0089	CA 9C 00	
	BIT 2,A	008C	CB 57	
	JP Z,SW3	008E	CA 9F 00	
	BIT 3,A	0091	CB 5F	
	JP Z,SW4	0093	CA A2 00	
	LD D,16H	0096	16 16	设置延时常数
	RET	0098	C9	
SW1	LD D,20H	0099	16 20	设置延时常数
	RET	009B	C9	
SW2	LD D,24H	009C	16 24	设置延时常数
	RET	009E	C9	
SW3	LD D,28H	009F	16 28	设置延时常数
	RET	00A1	C9	
SW4	LD D,30H	00A2	16 30	设置延时常数
	RET	00A4	C9	



怎样用电脑检修录像机

李玉全

用电脑检修录像机有自动检测和辅助检测二种方式。

一、自动检测

实现电脑自动检测录像机,关键是先把录像机正常时的各有关测试点电压、波形、机内有关元件的各种相关值,按一定程序固化在微处理器中检修录像机时,把探头接入测试点,取样(取电压、波形)送给电脑与正常数据比较,找出可能发生故障的元件。也可把录像机工作正常时,各种参数(主要相关元件参数)编出程序,存入电脑中。使用时,通过模/数(接口电路)转换电路,把电压、波形信号变成数字信号送给电脑。通过电脑把检测到的信号与标准信号(机内存的程序)进行比较,显示器显示出故障所在。

录像机工作时,磁鼓的转速、主导轴转速、带盘的转速要保持一个恒定速度。当这些关键部件转速发生变化说明录像机出现故障。录像机的各种转动部分的检测也是维修中的一大环节。测转速时,用霍尔元件、磁敏元件或光敏元件作成检测探头。在检修时,探头检测到转动信号,通过转换电路把单位时间内电流的变化,变换成数字信号。把数字信号输入电脑,进行比较处理,判断故障所在,显示在显示屏上。

二、辅助检测

用电脑辅助检测录像机,已达到了实用阶段。这种检测方法大大地提高了故障判断时间,提高了故障判断的准确率(因录像机中设计了多种自动保护电路,有时无法分清是否出现故障)可实现人和机“对话”。把录像机各种故障现象都编制成程序,把程序存入电脑中,使用时将录像机的各种故障现象,送入(键入)电脑,在电脑的指导下找出故障所在。如“程序一”是 NV-450 录像机“电源自动保护检修过程”。

```
5 REM“程序一”
10 LPRINT“松下 NV-450 录像机”
20 LPRINT“电源自动保护检修过程”
25 LPRINT“请先生遵照下列步骤进行”
```

```
30 LPRINT“首先检查 VEP03236H 电路板上的电源”
35 LPRINT“电路板的 Q1001 基极是否为高电平?”
40 INPUT“V1$=”;V1$
50 IF V1$=“是”THEN 80
60 LPRINT“请检查 VEP03236H 电路板上系统控制”
61 LPRINT“电路 IC6001(MN1542VG1)42 脚是否为”
62 LPRINT“低电平?”
65 INPUT“V2$=”;V2$
67 IF V2$=“是” THEN 90
70 LPRINT“请测量 IC6001 的 22 脚是否为低电平?”
75 INPUT“V3$=”;V3$
77 IF V3$=“是” THEN 100
78 LPRINT“应检查电源开关(POWER SW)S6559”
79 GOTO 130
80 LPRINT“应检查 Q1001(2SD1275)”
85 GOTO 130
90 LPRINT“应检查 QR6005(DTC124A)”
95 GOTO 130
100 LPRINT“请测量 IC6001 的 i6 脚在电源接通瞬”
101 LPRINT“间是否为低电平?”
110 INPUT“V4$=”;V4$
112 IF V4$=“是”THEN 120
115 LPRINT“应检查 Q6002(2SD636)”
116 GOTO 130
120 LPRINT“请检查 IC6001”
130 LPRINT“祝你成功! 欢迎合作!!”
140 END
```

三、用电脑作信号发生器

用电脑可输出各种图形信号,如屏幕中间一条横线、一条竖线、方格信号、垂直条信号、水平条信号、圆形信号,有的还能输出彩条信号。用电脑给录像机输入一图形信号,记录后再重放,观察图象情况,并和记录时的图形进行比较,判断是记录还是重放故障。

这种方法和用图形信号发生器相同,这样可提高电脑的利用率,给录像机维修带来方便。

用娃娃机开发单片机程序

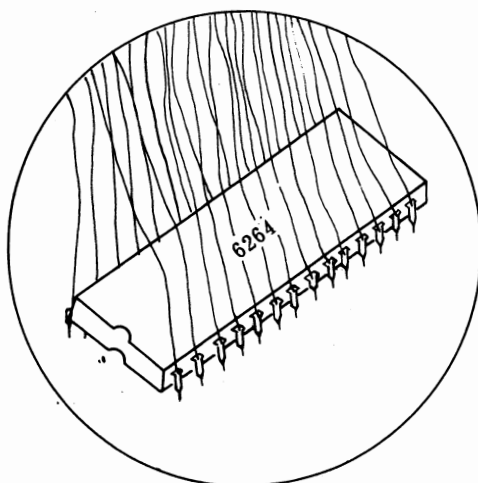
河南省新闻出版局办公室 张 涛

目前, MCS-51 单片机在电脑爱好者中日益普及, 早期已经普及的各种娃娃电脑却大有日趋闲置之

势。但业余爱好者手中的单片机系统, 程序只能手工汇编, 费时费力。而 LASER、COMX 等各种娃娃机, 大都

扩展内存插口各引脚排列如下(面对插口时):

+ IO		WA RF										RE									
NC	5V	RQ	RDN	MII	IT	M1	SHD	7	D2	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	SET	GND
22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
44	43	42	41	40	39	38	37	36	35	34	33	32	31	30	29	28	27	26	25	24	23
NC	+	NC	WR	MR	HA	INT	D1	D0	A0	NC	D6	D5	D3	D4	Φ	A15	A13	A12	A11	GND	
9V					EQ	LT										A14					



制作要点:

导线从集成块上方焊到各引脚根部,利用此集成块兼作仿真插头,注意焊点不要太靠下,以免影响集成块的插拔。

图 2

具有比较完善的屏幕显示和英文键盘,如能配上合适的汇编软件和接口电路来为单片机开发程序,必将给广大单片机爱好者带来莫大的方便。由于各种娃娃机的内存都比较小,如果汇编软件占内存量过大,则许多低档娃娃机就难以利用,因此,开发此类软件必须最大限度地减少其占内存量。

笔者经过潜心研究,在LASER机上开发成功了占内存仅1.5K的汇编程序,具有汇编、反汇编、源程序编辑、小汇编和内存编辑五大功能,配合笔者设计的超简易存储器仿真电路,使单片机的程序开发大为简便。下面介绍该程序的设计思想和设计中运用的主要技巧,希望有助于广大电脑爱好者在各种娃娃机上开发出更好的汇编软件,使单片机在业余爱好者中的应用水平得到进一步提高。

1、反汇编:在反汇编部分

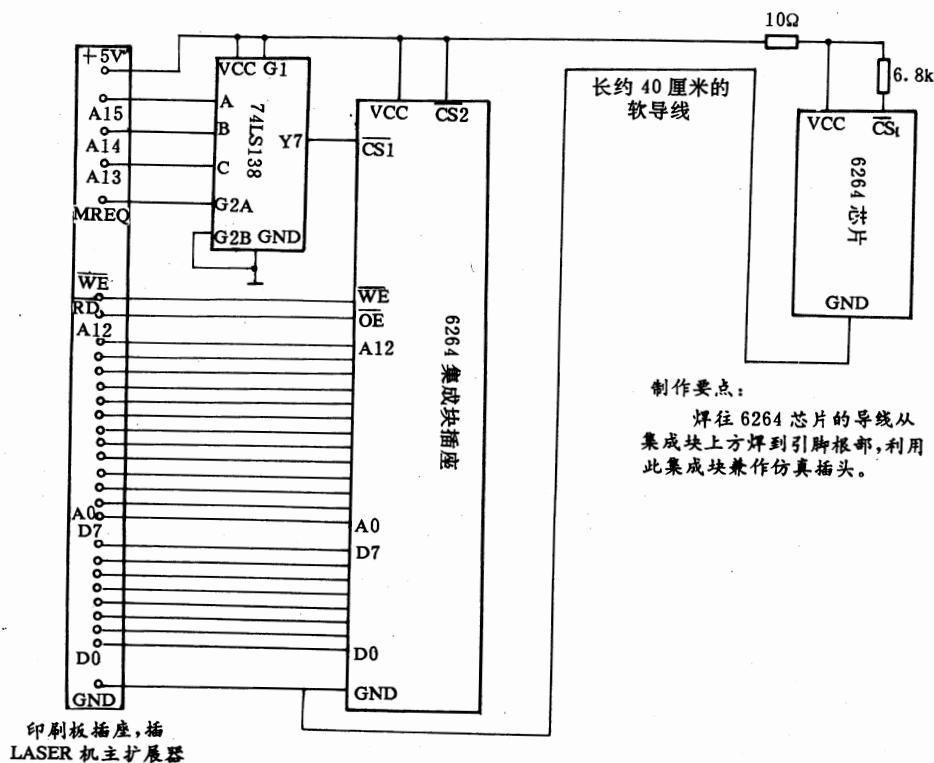


图 1

的设计中,笔者采用了“零件装配法”,大大节约了内存,其主要思想是:

①把一条助记符形式的指令看作是由几个“零件”组成,比如:MOV A, #DATA 可看作是由 MOV、A 和 #DATA 三个“零件”组成。在内存中设一个“零件表”,其内容是 MCS-51 指令集中用到的所有“零件”,共有约 80 个。

②由一个子程序求出待反汇编机器码对应助记符的各部分在零件表中的序号。MCS-51 指令矩阵有较强的规律性,在此子程序中要尽可能地利用,以缩短程序长度。

③由“装配”子程序按照序号从“零件表”中取出各个“零件”,在装配区中装配起来。

④由“配数”子程序把被反汇编操作码后所带的数据(如果有的话)取出来转换成十六进制 ASC 码写入装配好的指令中的适当地址。

⑤则装配区中的这条指令就可以显示输出了。

2、汇编一条指令:笔者在这部分的设计中利用反汇编功能来产生 MCS-51 指令集,避免了在程序中设置庞大的指令表。

①对数据 0 到 FFH 进行反汇编,逐条反汇编出 MCS-51 的每一条指令与待汇编的指令比较。

②若某一数据反汇编出的指令与待汇编指令完全一样,则这个数就是待汇编指令对应的操作码。

③把待汇编指令中的操作数读取出来(如果有的话)写到操作码后边。

④我们就得到了待汇编指令的完整机器码。

3、标号、伪指令:标号利用 BASIC 的变量系统来存储和查找,这样虽然限制了标号的长度只能是两个字符,却可省去大量的程序开销。

伪指令都用一个“&.”作为前缀,后面跟一个伪命令字,如 &O 代表 ORG; &E 代表 EQU; &F 代表 DEFB

等等,这样可以大大方便伪指令的识别与处理。

4、汇编起始地址的处理

各种单片机应用系统的地情况千变万化,而娃娃机中可用的 RAM 又都极其有限,笔者采用“双起始地址”的办法,解决了这一问题,其设计思想是:在源程序开头由用户设定汇编起始地址和虚拟起始地址(即实际运行地址),汇编时,将机器码生成在汇编起始地址开始的内存中,但对程序中的转移类指令,其目标地址都按实际运行地址来处理,这样汇编出的机器码,只要放入单片机中的实际运行地址处,就可正常运行。

5、源程序的处理:利用机器原有的 BASIC 编辑系统来编辑源程序,源程序为带有行号的 BASIC 形式,这样可以充分利用机器原有的编辑、存取磁带及打印等功能。

6、在程序中要尽可能地利用系统 ROM 中的子程序,比如显示输出、键盘输入等等常用功能在 ROM 中一般都有可用的子程序。

下面谈谈接口电路。笔者本着使制作尽可能容易的原则设计了这个程序存储器仿真电路如图 1。将此电路外插在 LASER 机上,将 6264 芯片插在插座中,可为 LASER 机扩展 8KRAM,地址在 E000—FFFF(用户也可通过改变三八译码器的输出端来将这 8KRAM 安排在其它地址上),用户可将程序汇编生成到这段内存中,然后将 6264 芯片从插座上拔下来,插到单片机系统中,就可以将这 8K 存有程序的存储器出借给单片机系统。本电路直接利用存储器芯片本身作为仿真插头,简化了制作工序,电路中所有零件可直接焊在印刷板插座和存储器芯片上,不必使用印刷电路板如图 2。此电路通用性极强,略加改动就可适用于各种常见娃娃机,用户还可以改用 6116 或 6132 等存储器芯片来作这个电路。

91 年全国少年儿童计算机应用竞赛 在河北廊坊举行

由全国妇联、团中央、中国科协、中国石油天然气总公司管道局、全国少年儿童计算机教育研究会主办的全国少年儿童计算机应用竞赛于 8 月 2 日至 4 日在河北省廊坊市中国石油天然气总公司管道局少年宫举行。来自全国各地少年宫的 28 个代表队,140 名小选手分别进行了“五项全能”、“动画制作”和“智慧小车”三个单项竞赛。

五项全能竞赛获一等奖的是:陆靖(男,中福会少年宫一队);慕巍(男,天津河西区少年宫队);夏甜(男,管道局青少年宫一队)。

寓言动画竞赛获一等奖的是:张彬(男,中福会少年宫一队);徐扬(男,大连少年宫队)。

智慧小车竞赛获一等奖的是:杨东林(男,中国儿童少年活动中心);黄春江(男,佛山青少年宫)。

这次竞赛内容丰富,突出应用和动手能力。竞赛包括了汉字输入、数据处理、计算机硬件系统连接、利用工具软件进行动画的设计与制作,以及将计算机应用在实际的控制系统中等。

这次竞赛较全面地考查了选手们对计算机系统的了解和使用能力并在青少年计算机竞赛的竞赛形式上、计算机软件与硬件的结合上都做了有益的尝试。

(吕 迈)



TH3070 打印机接口电路 工作原理与故障分析

李燕华

一、故障现象

打开电源,通电后打印自检正常,但和联机后不打印,并且使计算机死机。

二、电路分析

由于打印机自检正常,故障可能在和主机相联接的接口电路上。以下首先对接口电路的工作原理做简要分析。

主机和打印机之间应具备数据信息、控制信息、这两部分信息的传输功能,打印机的接口电路正是为了

查询方式。硬件线路通常也有两种形式,一种是并行接口,一种是串行接口。在一般情况下,TH3070 采用的是 Centronics 标准并行接口。图一给出了 TH3070 并行接口电路图。

从图一可以看出,主机和打印机通过扁平电缆线将信号传输到打印机一侧的 36 线插座 PJ32 上,打印机采用 8 位并行方式接收主机输入的数据(DATA0~DATA7)。控制及相互通信的信号有:忙信号 BUSG、应答信号 ACK、数据选通信号 DATASTB、选中信号 SE-

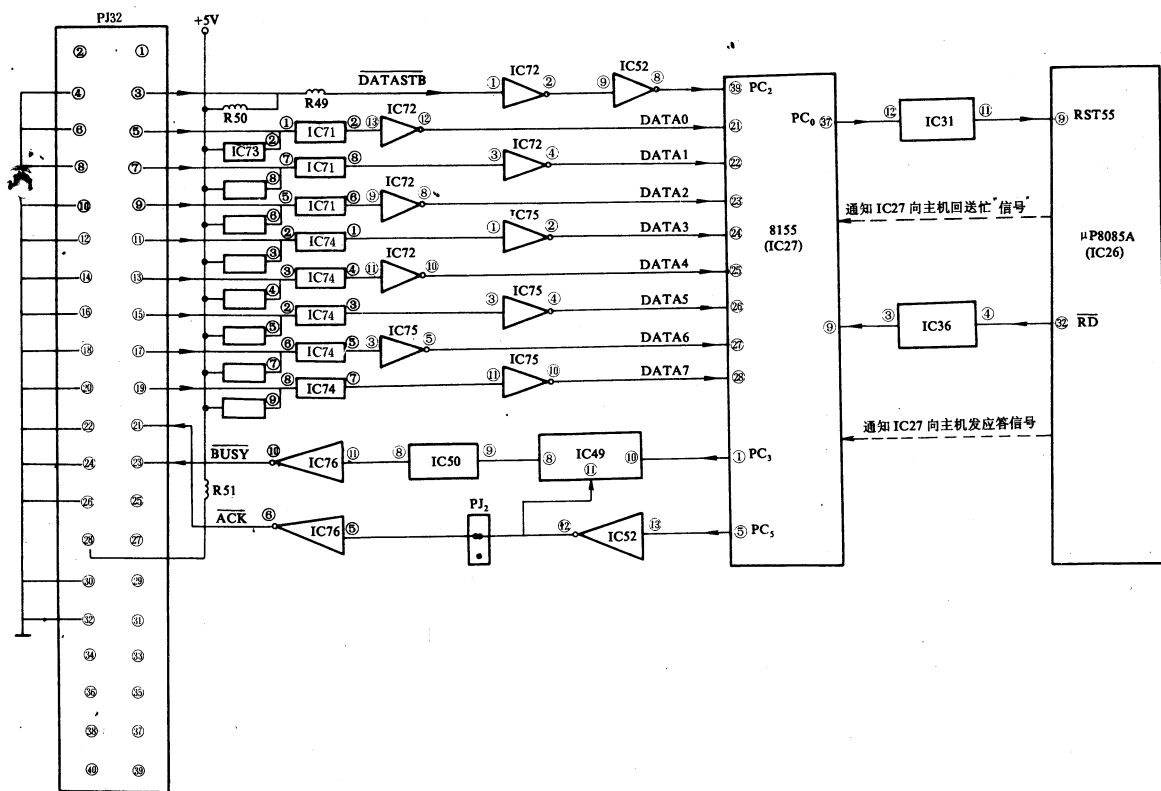


图 1 接口电路

完成这一功能而设置的。就“接口”而言,一般包括控制程序和硬件线路两部分。控制程序,存于计算机中,是主机对打印机的控制,通常有两种方式:查询方式和中断方式。在 IBM PC 机中,打印机的控制方式采用的是 LECT、初始化信号 PRIME、以及纸尽信号 PE、出错信号 FAULT 等。

图上表示了接口信号间的时序关系,接口电路的工作过程遵循信号时序的顺序。当打印机处于 BUSY = 即“非忙”状态时,主机将 8 位数据通过 PJ2 的第 5、7、9、11、13、15、17、19 脚送到打印机,经 IC72(74LS14)非门反相后送到接口 I/O 芯片 IC27(8155)的 A 口,接

(74LS14)、IC52(74LS04)两极非门反相后送到 IC2F 的 PC₂。由于数据选通信号的作用,打印机将已到达 IC27A 口的数据打入 IC27,同时 IC27 的 PC₀ 端发出 A 口数据选通中断信号,该信号经 IC31(或门)到达 CPU 8085A(IC26),向 CPU 提出 RST5.5 中断。CPU 接受这一中断后,先屏蔽 RST5.5 中断,使得在此期间数据的处理不受外来信号的干扰。然后 8085A 判断该中断是接口提出的初始化中断,还是数据接收请求中断。由于此时是数据请求中断,因而 CPU8085A 通知 IC27(8155)向主机回送忙信号,此“忙”信号由 8155(IC27)的 ①脚 PC₃ 发出,经 IC49 的 ⑩脚输入、置 IC49(74LS074)的 \bar{Q} 端(⑧脚)为“0”,然后 IC50、IC76 后,“忙”信号送往主机。至此,CPU8085A 可以对接收的数据做相应的处理。当数据处理完毕,CPU8085A 通知 IC27(8155)向主机发应答 \overline{ACK} 信号,IC27(8155)的 PC₅ ⑤脚发出的应答信号,经 IC52(74LS04)反相后,经 PJ2,一路送往 IC76 再次反相,然后从 PJ32 的 ②脚送往主机;另一路经 PJ₂ 送往 IC49(74LS074)的 CLR(时钟)端,将 IC49 的 \bar{Q} 端置“1”,使 BUSY=0 并送往主机,以便主机可以发送下一行数据。

三、故障的查找与排除

根据以上的故障现象和对电路的分析,我们考虑首先应从 I/O 接口电路的控制信号入手查找。参照图一的电路图和图二接口的时序关系,使用逻辑笔或示波器对“忙”信号(BUSY)、数据选通信号(DATASTB)、应答信号(ACK)等控制信号的通路逐点测量,经测量发现应答信号 \overline{ACK} 在 IC76(74LS06)的 ⑤脚处有输入,而 ⑥脚处无输出。IC76(74LS06)是一个六相反相器集成电路,六个输入端对地电阻应相本相同,六个输出端对地电阻应相本相同,用万用电阻档测输入端、输出端各对地电阻,结果 ⑤脚对地电阻为 $+\infty$,进一步断定 IC76(74LS06)集成片坏了,更换 IC76,打印机故障被排除,联机后打印一切正常。

对于有些经验的维修人员来说,也可以在使用逻辑笔和示波器之前,用万用电阻档,对接口电路中传输控制信号的几个反相器集成片的输入对地、输出对地各电阻值进行测量。因为它们较其他电阻性、电容性元器件容易发生故障。这样做的目的是为了更快、更简便地寻到故障的原因。

北京天翔新技术开发公司新近推出

国产化、廉价普及型九针中西文打印机

随着计算机的日趋普及,作为与计算机配套的外围设备——打印机的需求量增加。在打印机领域中,24 针打印机固然有其优秀的特点,与九针打印机相比,其速度及字型等功能是九针机无法比拟的。但价格较高,对一些应用领域难以接受。而九针打印机却以其低廉的价格在中国市场上占据了一席之地。天翔公司为振兴我国的打印机产业,以优质廉价的打印机替代进口产品,在消化、吸收国外先进技术后推出 TX-800⁺、TX-800K 九针打印机。

TX-800⁺中西文打印机是 TX-800 九针中西文打印机的二次开发产品。它提高了打印速度,配置了 IBM 及俄文方式,增加了串行口选件等。作为 TX-800 的增强型,机器的可用性及可靠性均向前迈进了一步,其主要性能均已赶上甚至超过某些进口机型,如 KY-980 等。

TX-800K 是国内现行唯一的中西文九针打印机。选用 TX-800K 与计算机配成系统,可以大幅度提高计算机效率,降低其系统的成本。比

如:使用西文打印机的图形方式打印汉字时,计算机必须备有硬汉卡或软字库,同时主机要有较大的内存,且配有相应的驱动程序。而 TX-800K 本身带有硬汉字库及其相应的控制软件,这就降低了计算机系统的成本。此外,每向其它打印机输出一个 16×16 点阵汉字,最少输出 32 个字节,而采用 TX-800K 仅需 2 个字节。提高了计算机的效率。它在软件上采用多种无级演变措施,丰富了可打印字型。

TX-800K 广泛适用于中小过程系统用户,程控交换机用户,小型文字处理系统及个人用户,不要求主机具有硬汉卡或软字库,就可进行汉字处理,这对我国计算机的普及应用,起到了一定的配套作用。目前,在邮政系统的电报、电话计费开据机、工业控制和饭店管理应用中,受到了广泛的欢迎。

目前,TX-800、TX-800⁺和 TX-800K 的零售价约在 1200~1800 元左右。今后,随着机器产量的增加,售价将会逐渐下降。

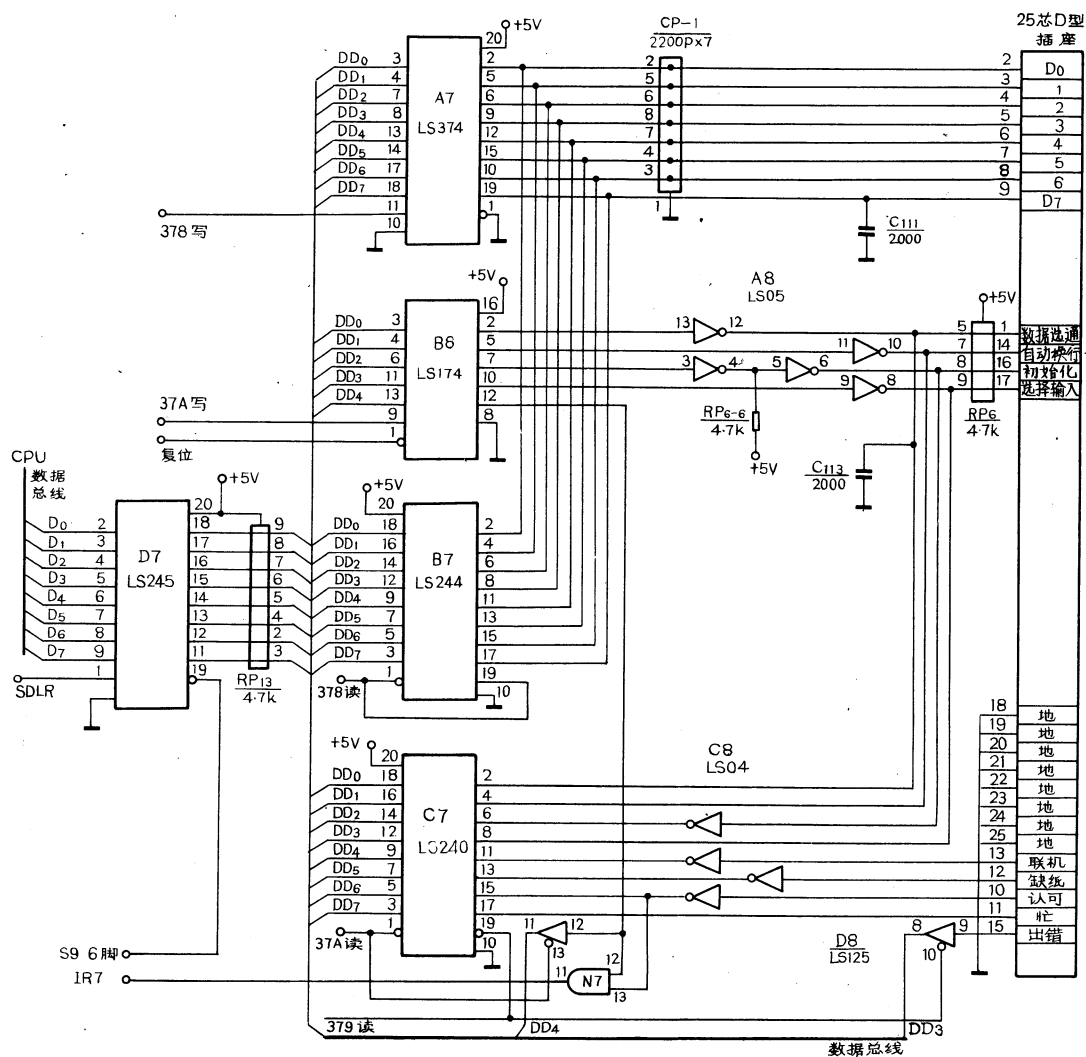
长城 0520CH 机打印机适配器故障维修

邱 檠

一、长城 0520CH 机打印机适配器的简要工作原理及电路

为了能够准确地判断故障部位,同时,也考虑到长城 0520CH 机(以下简称 CH 机)硬件资料不多见,我们先来简单了解一下 CH 机打印机适配器的工作原理。其电路原理图如图一所示。它由命令译码器、总线缓冲

器、数据锁存器、控制锁存器、控制驱动器和接收发射器等组成。考虑到与 IBMPC 机兼容,CH 机打印机适配器的逻辑结构与端口地址与 IBMPC 机都相同,但 CH 机所选器件以及用作译码的地址线又与 IBM PC 机不尽相同。如图二所示。



图一 长城 0520CH 机打印机适配器原理图

CH 机命令译码由 N8 和 E7 两个 74LS138 三——八译码器完成(N8 表示该器件在系统板的第 N 行,第 8 列上,这是位置代码。其它同),分别译出写数据(地址为 378W)、读数据(378R)、读状态(379R)、写控制(37AW)、读控制(37AR)等命令。这些信号分别作用到各锁存器和缓冲器上。

数据收发器 D7/74LS245 的数据传送方向由来自 S9 的 6 脚信号及 C10 的 2 脚的 SDIR 信号控制。当前者信号为低电平时,SDIR 为高电平,则数据由 CPU 系统总线写入适配器,若 SDIR 为低电平,则数据由适配器读入系统总线上。

A7/74LS374 是输出数据锁存器,锁存主机向打印机送出的数据。B7/74LS244 是输入数据缓冲器,输入 CPU 的数据一般就是主机向打印机刚送出的数据,用于故障测试与诊断。

控制锁存器是 D8/74LS174, A8/74LS05 是控制驱动器。控制字的格式为:

7	6	5	4	3	2	1	0
×	×	×	允许中断 (+)	选择输入 (-)	初始化 (+)	自动换行 (-)	数据选通 (-)

(+表示高电平有效,-表示低电平有效)它们接受主机对打印机的控制信号,并通过 25 芯 D 型接口送到打印机上。

控制/状态信号输入缓冲器由 C7/74LS240、N4/74LS125 及 C8/74LS04 组成。分别接受控制字和状态字的信息送到数据线上,通过 D7/74LS245 送给 CPU。读入的控制字节就是刚输出的控制字节,目的也是用于故障诊断。状态字格式为:

7	6	5	4	3	2	1	0
打印机忙 (+)	认可(-)	缺纸(+)	联机(+)	打印机出 错(+)	×	×	×

CH 机面板打印指示是由 379R 地址译码线提供的。

二、故障维修二例

下面我们来看两例打印机适配器的故障及维修。

• 故障现象一:打印机与 CH 机的打印接口接好后,进入系统,进行屏幕拷贝或文件打印时,打印机均没有反应,但 CH 机面板 I/O 口的打印机工作指示灯亮。当在总线槽上再插入一块打印卡时,此卡能够正常打印。

• 分析、维修:当进行打印时,虽然打印机无反应,但面板指示正常,这说明打印机适配器的译码正确,控制字及数据校验也正确。插入打印卡后又能正常打印,说明打印机是好的,状态字也能正常读入,那么就可以肯定故障就在 CH 板打印机适配器的数据发送器 A8/74LS05 或排电阻 RP6/47K 上。具体排放前,先用 DE-BUG 的 O、I 命令向 378H 地址口送数和读数,用示波器(或三用表)测试 25 芯 D 型接口的第 1、14、16 和 17

脚的电平。经测试发现,无论送什么数,脱机时,25 芯 D 型接口,脚总是低电平,再测 72LS05 的 12 脚对 +5V 电阻为 2.2K(正常为 4.7K),对地电阻为 140Ω(其它反向器输出端对地电阻为 560Ω),说明 74LS05 的一个门坏了,换上好的后,故障排除。这里应注意,排电阻一般不易损坏,应先检查 IC 片子,最后检查排电阻。另外,应急修理时,也可利用原来未用的 74LS05 的 1、2 脚的一个反向器,免得从系统板上将坏片子焊下。

• 故障现象二:打印机联机正常,进入系统后,屏幕拷贝和文件打印均不能进行,但打印机是好的,发出打印命令后,面板打印指示灯不亮。另插一个打印卡打印时,现象与原来一样。

• 分析、维修:不能正常打印,同时在打印时面板打印指示灯不亮,从电路上可以看出,故障可能出在译码器 N8 和 E7 的 74LS138 上,也可能出在锁存器 74LS174、74LS374 或总线缓冲器 74LS244、74LS240 和 74LS245 上。尤其是再插入打印卡打印时,故障依旧,更说明了故障器件在命令译码控制的几个片子上。为了确定故障部位,还应进一步测试。进入 EDEBUG 后,用 C 和 I 命令检查 378H 和 37AH 口地址输出和输入的数据是否一致。经查,脱机时,当向 378H 送 00 时,读入为 05,当送 FF 时,读入的为 FF;当向 37AH 口送 00 时,读入的为 E0,当送 FF 时,读入为 FF。由于 37AH 口送出和读入的是控制字,由其格式可知,其高三位未用,所以送 00,读入为 E0 是正确的。而 378H 口的数据就有问题了,从测试结果看,74LS374 的 DD₀ 和 DD₂ 锁存有问题,或者是 74LS244 的 DD₀ 和 DD₂ 读入有问题。将 25 芯 D 型接口的 2 脚和 4 脚接地,再向 378H 口送 00,读入为 00,这说明 74LS244 是好的,将 74LS374 换掉后,故障排除。值得注意的是,由于 CH 机的打印机适配器和总线座上的打印卡的端口地址一样,当出现故障时,将会互相影响。

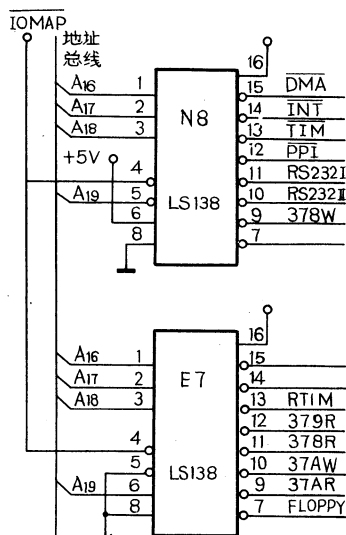


图 2 命令译码器

初级程序员级水平考试辅导

汉字 dBASE III 使用技巧(下)

王路敬

54. dBASE III 应用程序设计值得提倡的方法是什么? 操作的基本步骤有哪些?

利用 dBASE III 设计一个应用程序时,对于初学程序设计的人来说,感到困难的是缺乏程序设计的方法。而设计一个应用程序采用什么方法又往往是相当重要,如果方法不当,会导致事倍功半的结果。dBASE III 提供了一套先进的结构式应用开发语言,它不但语句丰富、自成体系,具有灵活的人机会话功能,而且它的语言体现了结构式程序设计的特征。用 dBASE III 语言编写的程序是一个结构式的程序。通常一个 dBASE III 的实用程序由一个主控程序和若干个子程序组成的。一个稍微复杂的应用管理系统中有许多大小不同的程序,这里就存在从何处入手的问题。传统的方法是先从下层入手,也就是说先实现下层程序(模块),因为下层某一程序功能比较具体,可以实现一部分就调试一部分。但是这种方法包含着很大的危险性,因为它不是从整个系统的功能出发。使用这种从下往上的方法,很可能出现各个模块单独调试运行都可通过,但是系统联调时却产生通不过的现象,其主要毛病产生于整体结构与各程序块间的接口处。

dBASE III 实用程序的设计值得提倡的基本方法是“从上到下,先粗后细,化整为零,分而治之”。这种方法的基本思路是:

(1)先实现上层,后实现下层。具体来说就是首先调试整个系统的结构及各程序之间的接口,在确保系统结构及各模块间接口正确无误之后,再实现下层程序的功能。

(2)在实现上层模块时,下层模块只保留其名字及输入参数和输出参数。而模块本身的处理先空着,或者只保留一两处象征性的语句或简单的显示信息。这样,上层模块可以调用下层模块,虽然其具体处理功能还未实现,可以验证调用是否成功,输入输出的传输是否正确等;这样就可以调试系统结构,保证其正确性。

(3)化整为零就是先部分后全部。即把整个系统分成若干“版本”,先实现一个只反映系统骨架的版本,而后不断扩

充,最后实现系统全部功能。

采用从上到下的方法既能有效地解决接口问题,又便于尽早发现系统结构整个上存在的问题,从而收到理想的效果。

实现上述方法的操作可分以下 7 步进行:

①明确管理系统应具有的功能,按这些功能划分程序模块,画出系统总体框图。

②要管理的信息由哪几种数据类型组成。字段多少,长度怎样,设计确定数据库的结构,并输入一定数量的数据。

③根据程序的功能画出程序的细框图。每个功能子程序的框图描述应尽量清晰详细,把每一步要做什么都在框图上标明。

④若需输出报表,则要设计报表格式。若屏幕输入数据,则设计屏幕格式文件。

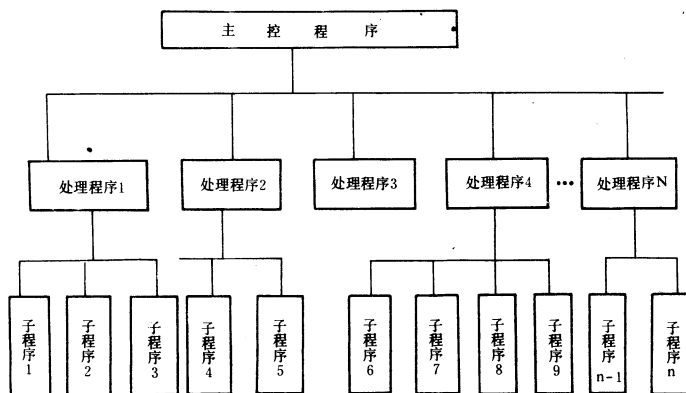
⑤检查程序框图没有错误,便可动手分模块编写程序。若几个人共同编程序,要按人分模块,完成功能比较完整的模块,不要将一个功能模块分给几个人来写,以免引起混乱。

⑥程序编写完以后,先分模块进行调试。调试的目的是检查该模块功能上是否达到了预定的要求,并排除错误。当各模块都通过分调之后,再联起来进行统试。

⑦统试通过后试运行,试运行无误后便可投入正常使用。

55. dBASE III 菜单程序的结构特点是什么?

在进行 dBASE III 实用程序设计时,菜单方式是经常使用的,因为实用程序主要是为了使用,而用户很多是不懂计算机的,不可能要求他们都会打入 dBASE III 命令,如果利用菜单方式,即将程序的功能全部在屏幕上显示,并将每一功能对应一个数字或字母,使用户象点菜一样,按某一键即可完成某一功能。菜单分大菜单和小菜单,大菜单即为主模块(或主控模块);小菜单即子程序(子模块)。菜单程序的一般结构框图如图所示:



在一个比较完善的微机管理系统中,往往包含了数十个,甚至上百个菜单程序,因此,菜单程序如何设计,对系统的功能起着举足轻重的作用。众所周知,dBASE III 命令功能很强,往往一条命令就可以独立完成一类复杂的操作,例如数据处理中常用的分类、排序、检索、显示、打印报表等操作,都可以由一条命令完成,如果再加上范围、条件、函数等短语,其功能就更强了。因此,操作使用人员几乎可以不必考虑操作功能如何实现,而把主要精力放在菜单程序设计上,只要把一个个菜单设计好,整个程序设计的问题也就迎刃而解了。

dBASE III 的菜单程序通常有如下特点:

(1)多分支选择结构。“菜单”说到底是一个多分支程序,即程序执行到此先暂停,等待使用者决定下边执行哪个分支程序,而后继续执行。dBASE III 的条件转移命令有两个,IF 和 CASE。IF 命令要求 ENDIF 与 IF 成对使用,一般在分支较少时可以使用,而在分支较多时,使用 IF 远不如使用 CASE 编程简洁,在菜单程序中,大多使用 CASE 命令。

(2)单入口单出口结构。尽管菜单程序是多分支的,但其入口和出口是唯一的,它是由单支发散为多支,再由多支汇总为单支,程序在 DO CASE 后发散,在 ENDCASE 时汇总。通常把“一道菜单”程序编为一个独立的“子命令文件”,需要时调用,执行某分支后由唯一的 RETURN 返回主程序。

(3)采取了“容错”设施。实践表明,在微机系统运行中,绝大多数错误来自于操作失误。为了使微机系统在操作使用人员对“菜单”提问回答错误时仍能继续运行,并给操作人员以错误提示信息和改正错误的机会,而不至“死锁”或停止运行。在菜单程序中,一般都含有循环程序,在选择错误时,用 LOOP 命令返回程序开头,再显示该“菜单”,要求操作人员重新选择。

56. 利用 TEXT...END TEXT 设计菜单的优点是什么?

在 dBASE III 中提供了一组很简单的菜单设计命令 TEXT...END TEXT,该组命令的功能是显示命令文件的一个文本数据块,此命令用于屏幕菜单提示设计既方便又简单。使用时 TEXT 与 ENDTXT 必须成对出现,例如主控命令文件中的菜单设计:

```
SET TALK OFF
CLEAR
TEXT
```

```
1. 显示    2. 查找
3. 增、删记录
4. 打印报表  5. 退出
```

```
ENDTEXT
```

```
⋮
⋮
```

```
RETURN
```

如果在 TEXT 之前加上 SET PRINT ON 语句,还可以输出菜单到打印机。

57. 主菜单程序应包括哪些基本内容

主菜单程序又称主控模块,它是管理系统的控制核心,它要调用和运行子程序,要为整个程序建立工作环境,在返回圆点提示符之前还要关闭各类文件。管理系统运行时,首先进入主菜单程序,主菜单程序应显示出系统所具备的各项功能名称,给出必要的提示信息。每一种功能子程序有其自

己的编号和名称,操作者根据自己的需要键入相应的功能编号,使其控制转向功能号所对应的子程序,执行完后返回主菜单程序。

主菜单程序应该包括下面的基本内容:

(1)设置工作环境

首先要用 CLEAR 获得一个清晰的初始状态,再逐个设置所需要的 SET 状态。例如,控制台响铃,命令的对话状态,目录路径,小数位数及功能键的设置等。

(2)初始化内存变量

要定义整个程序中使用的内存变量,给出它们的初始值。

(3)设计主菜单

为了保证单入口单出口的程序结构,一般都采用循环来设计主控程序。在主控程序中通常必须有如下四种类型的语句:

①DO WHILE(条件)语句。一开始应确保条件为“真”,以保证执行主控程序时能进入主菜单,通常选用 DO WHILE . T. 的形式。而在“退出”选择项时使用 EXIT 退出主控程序。

②?,TEXT...ENDTEXT 或@语句。用以显示主控菜单的各个选择项。

③人机交互性语句。例如,ACCEPT,INPUT,WAIT 等。用以接受用户键入的选择。若各选择项的编号为数字,则使用 INPUT 较为方便;若为字母,则使用 ACCEPT。

④DO CASE 或 IF 语句。用以根据不同的选择调用不同的子模块。在选择项较多时,使用 DO CASE 语句最为方便。

(4)程序结束前的内务处理

在返回圆点提示符之前,应恢复 SET 设置的标准默认状态,关闭各类文件。注意,RETURN 语句并不关闭文件。除此之外还应释放全局性的内存变量。

58. 主菜单程序编制的常用方法有哪几种?

编制菜单的方法有很多种,常用的一种方法是利用“?”和“CASE”语句相结合。用?在屏幕上显示各种功能及其对应选择键,然后用 CASE 语句来分别执行对应功能。

例如一个简单的工资管理程序的主菜单程序如下:

```
CLEAR
DO WHILE . T.
? "××单位工资管理系统"
? "1.....追加、插入、编辑"
? "2.....查询"
? "3.....删除"
? "4.....显示"
? "5.....打印"
? "0.....退出"
? "置入您需要的选择"
WAIT TO SELECT
DO CASE
CASE SELECT="1"
.....
CASE SELECT="2"
.....
.....
CASE SELECT="0"
EXIT
ENDCASE
ENDDO
```

执行该程序时,用户按某一数字键 CASE 语句就执行某一数字所对应的功能,当按入“0”时即退出主菜单程序。

第二种是利用@命令、READ 命令与 CASE 命令配合,其优点是使屏幕上显示的格式美观。例如下面这个主菜单程序:

```
CLEAR
STORE " " TO OP
DO WHILE OP<>"N" . AND . OP<>"n"
CLEAR
@0,27 SAY "(管理系统名称)"
@1,27 SAY "*****"
@2,4 SAY " "
@3,4 SAY "A.....XXXXX"
@4,4 SAY "B.....YYYYY"
.....
@PROW()+1,4 SAY "N.....结束"
@PROW()+1,4 "请按键进行选择" GET OP
READ
DO CASE
CASE OP="N" . OR . OP="n"
CLEAR
? "欢迎您再次使用本系统"
EXIT
CASE OP="A"
.....
CASE OP="B"
.....
ENDCASE
ENDDO
```

第三种方法是利用 TEXT.....ENDTEXT 与 DO CASE 语句相结合,用 TEXT.....ENDTEXT 在屏幕上显示主菜单,用 DO CASE 语句选择某一功能。下面这个实用程序的主菜单就是这样设计的。

```
* 主菜单程序 CONTROL. PRG
SET DEVICE TO SCREEN
SET PRINT OFF
SET TALK OFF
DO A, FIRST
DO A, SECOND
DO WHILE . T.
CLEAR
? "      为 您 服 务      "
? "      =====      "
WAIT "敲任一键开始"
TEXT
CLEAR

*****
*      功能如下:      *
* 1---人事档案材料的编辑 *
* 2---人事档案材料的显示 *
* 3---人事档案材料的查询 *
* 4---人事档案材料的统计 *
* 5---人事档案材料的报表输出 *
* 0---退出 *
*****
```

ENDTEXT

INPUT "请按 0—5 选择功能" TO TT

```
DO CASE
CASE TT=1
DO A, BJ
CASE TT=2
DO A, XS
CASE TT=3
DO A, CX
CASE TT=4
DO A, DJ
CASE TT=5
DO A, DY
CASE TT=0
DO A, LAST
ENDCASE
ENDDO
```

从上面几个实例我们不难看出在 dBASE III 应用程序中通过菜单这种人机接口让用户选择执行各种功能,给使用带来了很大的方便,使得操作简单而又直观,同时给阅读程序和分析程序带来很多好处,是值得提倡的。

59. 使用分支语句进行分支结构程序设计时应注意哪些问题?

dBASE III 提供了三种典型的分支语句:

简单分支语句:IF——ENDIF;

选择分支语句:IF——ELSE——ENDIF;

结构分支语句:DO CASE——ENDCASE。

使用这些分支语句进行结构程序设计时,以下几个问题不可忽视:

(1)分支语句是一种结构化程序,其入口语句例如:IF, DO CASE 与其相对应的出口语句,例如 ENDIF, ENDCASE 一定要配对出现。

(2)作为出口语句,例如 ENDIF 和 ENDCASE,在其语句后可增加一些文字性的注释内容,目的是提高应用程序的可读性。这些注释内容对程序的运行不会产生任何影响。

(3)DO CASE 与第一个 CASE 项之间不能有任何可执行的命令,系统永远不执行这个位置上的语句。

(4)三种分支结构,彼此都可以作为结构中待执行的语句行序列,组成嵌套形式,但嵌套的层次要清晰,不能交叉。否则造成逻辑混乱。

(5)为了防止仅有入口语句没有出口语句,或嵌套混乱情况发生,使程序的结构层次清晰,减少编写错误,增加程序的可读性,编写结构程序时,在书写上最好采用“锯齿”形的写法。即将结构的入口和出口语句写在同一纵座标的位置上,而嵌套在里面的其他结构语句的入口和出口向右移若干空格,写在另一同纵座标的位置上等等。这样一层一层的结构十分清晰,给查错也带来很大的方便。

60. 在使用循环语句设计循环程序时应注意哪些问题?

在 dBASE III 应用程序设计中常常需要重复执行某些相同的操作,即对一段程序进行循环操作。DO WHILE——ENDDO 循环语句保证了这一功能的实现。但在使用循环语句设计循环程序时需要注意以下四点:

(1)循环入口语句 DO WHILE 与出口语句 ENDDO 必须配对使用。

(2)在出口语句 ENDDO 之后可附加文字性的说明内容,

以便增加程序的可读性。其说明内容不影响程序的执行。

(3)循环程序中可以包含循环程序,即循环程序可以嵌套。循环嵌套层次不限,但内层循环必须完全嵌套在外层循环之中,不允许出交叉循环,否则,造成逻辑上的混乱,这是不允许的。

(4)循环语句和条件分支语句可以一起联用,同样允许嵌套,即在循环结构语句中可以包含条件分支语句,反之亦然。但是不允许交叉。为防止这类错误出现,在编写程序时亦可采用“锯齿”法书写(参看上个问答的有关部分)。

61. 为什么要使用过程文件? 过程文件中的过程如何调用?

在一般调用中,过程是当作一个文件独立存放在磁盘上的,所以每调用一次过程,都要打开一个磁盘文件,而每和磁盘打一次交道都是需要一定时间的。这样一来,当频繁调用过程(子程序)时,信息管理系统的运行就会走走停停,影响系统的运行速度。从减少磁盘访问时间出发,可以把多个常用的过程合并成一个大的程序文件,将这个大的程序文件一次打开,之后可以多次调用其中包含的过程。在这个大程序文件中,每个过程仍然是独立的,这个大的公用程序被称之为过程文件。一个过程文件可以容纳 32 个过程。过程文件被打开后,dBASE III 系统就能找到该文件所容纳的各个过程的所在位置,不需要频繁地进行磁盘操作,从而大大提高了调用过程的速度。同时一个过程文件由许多过程组成,既便于编写易于调试运行,也大大缓解了允许系统打开文件个数的要求。

在过程文件被打开前,其所包含的过程是不能被任何程序调用的。因此,在过程文件中进行过程的调用,在调用前,过程文件必须预先用打开过程文件的语句:

SET PROCEDURE TO <过程文件名>

打开。

过程文件被打开后,它所包括的过程便可为其他程序调用,调用时仍使用 DO 语句,只不过这时在 DO 语句中使用的是由 PROCEDURE 所指明的过程名。值得注意的是,过程文件名和过程名是两个不同的概念。过程文件仍是 .PRG 文件,过程名可由 1—8 个字符组成,但没有扩展名,每个过程以语句 PROCEDURE<过程名>开始,以 RETURN 语句终止。

62. 设计屏幕格式文件应注意些什么问题?

所谓屏幕格式文件实际上是编写若干条 @—SAY—GET 语句所组成的一种文件。dBASE III 执行时便在屏幕上显示用户自己满意的画面格式,供操作者输入和修改编辑数据。这种文件的扩展名为 .FMT,设计该类型的文件应注意以下几点:

(1)屏幕格式文件是文本文件,可以在 dBASE III 的 MODIFY COMMAND 字处理方式下建立,也可以在其他编辑程序下建立,但文件的扩展名 .FMT 必须在命令格式中明确的给出,以标明建立的是屏幕格式文件。否则 dBASE III 将自动地认为是命令文件,并赋给文件的扩展名为 .PRG。

(2)在调用屏幕格式文件之前,必须用 SET FORMAT TO <格式文件名>将格式文件打开。(可不带扩展名)

(3)屏幕格式文件只能由两种语句组成,其一是注释语句 *, 用来对格式文件的功能及所使用的语句加以说明;其二是 @—SAY—GET 语句,用来设计用户屏幕格式。

(4)调用屏幕格式文件只有下述五条命令:APPEND, INSERT, EDIT, CHANGE 和 READ 命令。但要注意,一个 READ 语句能关联的 GET 变量,在 dBASE III 中最多为 128 个。

(5)屏幕格式文件用完之后,如果不再需要应该使用 CLOSE FORMAT 语句及时的关闭。

63. 在进行信息查询时,被查询的内容确定存在,但有时就是查不到,为什么?

出现这种情况往往是由于当前盘和系统定义的默认盘不一致而造成的。每当汉字 dBASE III 被启动之后,汉字 dBASE III 即自动地在其所在盘上寻找是否有 CONFIG. DB,若有这个文件,则首先执行 CONFIG. DB 所含的各条命令,执行完,进入汉字 dBASE III 命令状态;如果没有,便立即进入汉字 dBASE III 命令状态。若被查询的内容确实存在,但又查找不到,解决的方法是将当前盘和系统默认盘取得一致,或者被查询的文件放在默认盘上,就不会发生查找不到的现象。

64. 什么叫模糊查询,怎样实现?

模糊查询是相对明确查询来说的。一般说来,明确查询要求查询人员提出明确的查询条件,而且被查到的信息要和查询人员提出的查询条件一字不差地完全相同,而模糊查询则要求查询人员提出的查询条件可以是不完整的,零碎的,只字片语的,而被查到的信息也不要求和被查询的条件完全一致,只要包含符合查询条件的内容,满足用户要求就行了。

模糊查询主要是用字符串包含运算符 \$ 实现的。

例一,用 LOCATE 进行模糊查询:

- USE <库文件名>✓
- LOCATE ALL FOR “高”\$职称✓
- DISP✓
- COUNTINUE✓
- DISP✓

一个信息管理系统的设计,如何做到既能实现明确查询,也能进行模糊查询,这无疑是一个十分重要的问题。

例 2,用 LIST、DISPLAY 进行模糊查询

- USE<库文件名>✓
- LIST ALL FOR “王”\$姓名

例 3,用 FIND 和 SEEK 进行快速模糊查询

这时需要建立多个按某个字段中的各个单字为索引关键字的索引文件。

例如,按姓名字段第二个汉字为“秀”的单字查询:

- INDEX ON SUBSTR (姓名,3,3) TO XMS✓
- FIND 秀✓
- DISP✓

65. 提高程序执行速度可采取哪些措施?

对于 dBASE III 的大多数应用软件,影响其速度的主要是查询检索操作,而不是访问磁盘。提高程序执行速度的措施可以从以下几个方面着手。

(1)利用 dBASE III 建索引文件和用程序段代替命令的方法。

一般来说,用单条命令处理时,在单位时间内所处理的记录数要比用程序所处理的多。但在实际工作中,大多数情况并不是对库中连续的记录进行一定的操作,而只是对库中满足条件的记录进行操作。此时,如果用单一的命令来完成一定的操作,则相当一部分时间花在对无关记录的判断上,而不是进行指定的操作,从而影响了整个程序执行的速度。如果充分利用 dBASE 索引文件的特性,同时用程序段来代替相应的命令,则可以大大地缩短完成一定的操作所需的时间。条件越复杂,其效果越明显。

众所周知,索引文件具有将关键字段相同的记录排在一个连续的区域中的特性,因此,根据不同的目的,对相应的关

键字段进行索引,就可将零乱的数据库按要求整理的很有规律。利用建索引文件,同时用具有特定功能的程序段代替具有相同功能的单一命令,就可实现在最短时间内完成某项具体的操作。

例如有一个 1000 个记录的人事档案数据库 JTGL,分为 20 个部门,要求按库 BM 的部门顺序分部门将每个人的基本情况打印输出。可采用两种方法:

其一,直接在基本上无规律排列的原始数据库

JTGL 上操作(见程序段 PROG1. PRG)。这种操作过程是对某一个部门的所有记录即将整个数据库的 1000 个记录从头到尾扫描一遍,符合条件的操作,不符合条件的跳过。对 20 个部门的所有记录进行扫描,就要将数据库从头到尾过 20 遍,因而大量的时间花在对无关记录的判断上。请看 PROG1. PRG:

```
CLEAR
SET TALK OFF
SET PRINT ON
SELECT A
USE JTGL
SELECT B
USE BM
DO WHILE .NOT. EOF()
    BUMEN=部门
    SELECT A
    DISPLAY FOR 部门=BUMEN
    SELECT B
SKIP
ENDDO
```

其二,首先对数据库在部门上进行索引,然后在索引数据库上进行操作,操作哪个部门,只对索引文件上相应区段内的记录过一遍,因此,对 20 个部门的所有记录按要求操作一遍,只相当于对整个数据库从头到尾过一遍(见程序段 PROG2. PRG)。

PROG2. PRG 清单如下:

```
CLEAR
SET TALK OFF
SET PRINT ON
USE JTGL
INDEX ON 部门 TO BUMEN
USE JTGX INDEX BUMEN
SELECT B
USE BM
DO WHILE .NOT. EOF(<>)
    BUMEN=部门
    SELECT A
    FIND & BUMEN
    DISPLAY
    SKIP
ENDDO
SELECT B
SKIP
ENDDO
```

这两种方法的功能完全相同,只是 PROG2. PRG 在利用索引文件的同时用了程序段:

FIND & BUMEN

```
DO WHILE .NOT. EOF( ) .AND. 部门=BUMEN
DISPLAY
SKIP
ENDDO
```

代替了 PROG1. PRG 中的单一命令

DISPLAY FOR 部门=BUMEN

实际运行结果表明,完成同样的操作 PROG2. PRG 比 PROG1. PRG 的速度快约 6~7 倍。

(2)利用 COPY 命令生成临时库的方法。

利用 COPY TO (文件名)FOR (表达式)命令,把要查询或统计范围内的记录放到一个临时库文件名下,然后再对该文件进行查询或检索。若使临时库文件不占用磁盘空间,一旦完成指定操作后可用 ERASE 命令将其删除。操作时所用的程序段可以这样编制:

```

:
SELECT A
USE (原始数据库名)
COPY TO (临时库名) FOR (表达式)
SELECT B
USE (临时库名)
DISPLAY
ERASE (临时库名) .DBF
:
(3)巧用 PACK 命令。
```

在删除数据库的记录时一般分为二步:第一步先对要删除的记录作删除标记“*”,即执行 DELECT [范围][FOR (条件)]命令;第二步执行 PACK 命令,对作删除标记的记录真正的从数据库中删除。若删除的记录不连续,不要逻辑删除一条或几条就执行 PCAK 命令,而是要删除的所有记录都选进行逻辑删除,最后一次使用 PACK 命令将凡是作逻辑删除的记录都删除掉。这样要比逻辑删除一条,就执行一次 PACK 命令要节省时间。因为 PACK 命令又称打包命令,或紧缩命令,多次打包所费时间总是比一次打包要多得多,尤其是数据库文件中记录总数相当多的情况下就更明显。

如果要删除一个数据库的全部记录,不必用 DELECT ALL 命令,而最好用 COPY TO(文件名)STRU 命令,这样可大大提高删除的速度。

(4)控制打开文件的个数

在使用 dBASE III 后,人们往往会发现,虽然它能够同时打开多个文件(最多 15 个),但当打开一定数量的数据文件后,程序运行的速度却降低了,特别是在子程序嵌套层次多的情况下尤为明显。解决的办法一是将数据缓冲区个数开辟到最低限度,也就是需要多少就开辟多少。具体做法是改写 CONFIG. DB 文件中的 BUFFERS 数值。因为,开辟大的缓冲区使得 dBASE III 能保留更多的信息,因而运行速度快。但缓冲区比文件要占更多的空间。另一个办法是尽量减少子程序的嵌套次数。具体做法是编写命令文件尽量长一些,以利于减少文件个数,减少程序嵌套次数。但是也应当注意,程序也不能太大,太大了一次调入的时间增加,反过来也会影响运行的速度。

(5)消除存储碎片

存储碎片是由于文件分散在盘上不连续的块中而产生的。读写这些碎片中的数据要花更多的时间。消除碎片的方法是用 COPY 命令将其文件先拷贝到一块新盘中,然后再拷

回到原盘中。对于硬盘,可以使用 BACKUP 和 RESTORE 命令来达到连接文件物理位置的目的。

(6)灵活使用 RUN

RUN 命令用来运行外部命令或其他可执行文件。运行外部命令比执行 dBASE III 命令快,可利用这一特点来提高执行速度。例如,在 dBASE III 下把 ABC.DBF 复制到 XYZ.DBF 中,可以使用二种方法:

其一:·USE ABC

·COPY TO XYZ

其二:·RUN COPY ABC.DBF XYZ.DBF

这里的 COPY 是 DOS 文件复制命令,这种方便的速度远远高于其一所用的方法的速度。

(7)善于使用过程文件

应用系统往往由多个文件组成。dBASE III 一次只调用一个命令文件。调用命令文件,系统必须到磁盘上寻找一次文件目录,再将所需的文件调入内存。随着访问磁盘次数增加,系统的运行速度就降低。使用 dBASE III 过程文件可以减少访盘次数。程序在打开过程文件时就将该文件所包含的过程一同调入内存,在调用每一个过程时就不需进行磁盘操作,因此,可大大提高系统的运行速度。

过程文件的组成形式如下:

PROCEDURE (过程名)

;

RETURN

PROCEDURE (过程名 2)

;

RETURN

;

在用之前,用 PROCEDURE TO (过程文件名)打开过程文件,调用结束后,用 CLOSE PROCEDURE 命令关闭过程文件。

一个过程文件,其中可以有最多 32 个过程。过程文件一旦被打开,各个子程序在盘上的位置就知道了,从而减少每次访盘再去查目录占用的时间。

(8)REPLACE 命令与 SUM 命令使用技巧

在实际应用中尽可能用一个 REPLACE 命令实现对多个字段的修改,比多次用 REPLACE 命令每次只对一个字段修改要快得多。例如,用

REPLACE (字段 1) WITH (表达式 1), (字段 2)

WITH (表达式 2), ... (字段 n) WITH (表达式 n)

比用:

REPLACE (字段 1) WITH (表达式 1)

REPLACE (字段 2) WITH (表达式 2)

;

REPLACE (字段 n) WITH (表达式 n)

的速度快得多。

在统计求和时,用:

SUM (字段 1), (字段 2), (字段 3) TO Y1, Y2, Y3 FOR (条件)时,可用

Y1=0

Y2=0

Y3=0

DO WHILE (条件)

Y1=Y1+字段 1

Y2=Y2+字段 2

Y3=Y3+字段 3

END DO

代替,其速度也能加快。

(9)利用屏幕格式文件

利用屏幕格式文件可以定义语句输入格式。所谓屏幕格式文件就是在一个以 .FMT 为扩展名的文件中,利用格式化输入语句在屏幕的适当位置上接收数据。这样可以避免在用 APPEND . EDIT 命令输入、修改数据时,以 dBASE III 提供的固定屏幕格式接收数据,这样输入数据更为方便,调用屏幕格式文件,可以提高速度,且可以提高应用程序的质量。

使用屏幕格式文件时,用 SET FORMAT TO (格式文件名)命令打开,用完后用 CLOSE FORMAT 命令关闭。

(10)循环语句中 LOOP 命令的使用

在 DO WHILE ENDDO 循环结构语句中使用 LOOP,使程序短路的命令跳过循环体中 LOOP 之后的命令,比不用 LOOP 命令而用 IF 等命令跳过其后的命令,程序的执行速度快。这是因为在使用 IF 命令时,不管 IF (或 ELSE) 后的命令是否要执行,都要调入内存解释一次,而 LOOP 命令就不需要这样。

66. 用 dBASE III 编制报表需要解决哪几个问题?

(1)走纸问题

换行或走纸是 dBASE III 制表中较头痛的问题之一。当前流行的各种 dBASE III 版本中,用 @ 命令制表时,都存在打印机不能连续打印超过 255 行的现象。若超过 255 行即会出现死机或打印机空走纸。虽然 dBASE III 提供了一个换页命令 EJECT,但若使用该命令换页时,打印机走纸并不是一页而是两页。解决这一问题可以采用以下的方法:

在用格式输出命令制表时,打印机通过当前行函数 FROW() 确认纵坐标。此函数的允许值是 0~255。一般此函数的值是连续累加的,当超过 255 时就会使打印机空走纸。可以通过以下的命令使该函数的值为零:

@ (行坐标,列坐标) SAY CHR(27)+CHR(12)

利用上述命令换页,避开 EJECT 命令,可解决换页和空走纸的问题。参看下列小程序段:

SET DEVICE TO PRINT

SET PRINT ON

@PROW(), 0 SAY CHR(27)+CHR(12)

;给当前行函数置零

DO WHILE .T.

@PROW()+1, PCOL()+1 SAY

;假定打印若干内容

IF PROW() >= 66

;规定一页打印 66 行

@66, 0 SAY CHR(27)+CHR(12)

;将打印头移至次页

第一行并将当前行函数 PROW() 置零

ELSE

ENDIF

ENDDO

SET DEVICE TO SCREEN

(2)变换字型

通过 dBASE III 提供的函数 CHR() 向打印机发送控制码可以改变汉字字型。不同的打印机改变字型的控制码可能不同,即使相同的打印机使用不同的汉字驱动程序,控制码也可能不同。字型变换的种类决定于所使用的驱动程序。TH-3070 打印机在 3070C . EXE 驱动程序下,可以打印 A、B、C、D、E、F、G、H 和 P 九种字型,而使用 3 . COM 驱动程序则只打

印 A~H 8 种字型,而在 CCDOS 4.0 支持下的 TH3070E.EXE 则可打印 A~N 和 a~n 共 28 种字型。所以在进行字型变换操作时,一定要注意汉字驱动程序要求;驱动程序要与打印机型号配套;驱动程序要与所要求的汉字库配套;驱动程序要与使用的汉字操作系统配套。在向打印机发送控制码前,要先接通打印机。

① TH-3070 打印机,3070C·EXE 驱动程序

改变字体的方法:

SET PRINT ON

@〈行,列〉SAY CHR(27)+"I"+"字型号"例如,选择 B 型字体:

? CHR(27)+"I"+"字型号"

或

SET DEVICE TO PRINT

SET PRINT ON

SET PRINT ON

? CHR(27)+"I"+"B"

或

SET DEVICE TO PRINT

SET PRINT ON

@〈行,列〉SAY CHR(27)+"I"+"B"

② M2024 打印机,P2024·COM 驱动程序

改变汉字字型的方法:

SET PRINT ON

? CHR(27)+"I"+"字型号"

SET PRINT OFF

例如,选择 D 型字体:

SET PRINT ON

? CHR(27)+"I"+"D"

SET PRINT OFF

(3) 在一行中打印不同字型的问题

当需要在同一行中打印不同的字型时,可以采用格式命令输出,也可以采用非格式命令输出,还可以采用格式命令和非格式命令混合输出的方法。下面通过具体程序实例来说明在不同情况下的操作。

① 采用非格式输出命令

SET DEVICE TO PRINT

SET PRINT ON

? CHR(27)+"I"+"A"

? "长城系列"

?? CHR(27)+"I"+"B"

?? "微型计算机"

?? CHR(27)+"I"+"A"

?? "使用技术培训班"

? " "

SET PRINT OFF

② 采用格式输出命令

SET DEVICE TO PRINT

SET PRINT ON

? CHR(27)+"I"+"A"

@PROW()+1,1 SAY "长城系列"

?? CHR(27)+"I"+"B"

@PROW(),PCOL() SAY "微型计算机"

?? CHR(27)+"I"+"A"

@PROW(),PCOL() SAY "使用技术培训班"

@PROW()+1,1 SAY " "

SET PRINT OFF

SET DEVICE TO SCREEN

③ 格式命令和非格式命令混合方式

SET DEVICE TO PRINT

SET PRINT ON

? CHR(27)+"I"+"A"

@PROW()+1,1 SAY "长城系列"+CHR(27)+"IB";

+ "微型计算"+CHR(27)+"IA"+"使用技术培训班"

SET PRINT OFF

(4) 打印实践问题

要使打印机能打印出实线表格需解决以下 3 个问题:

① 选择合适的制表符

制表符可以由打印机本身提供的制表符、ASCII 代码、中文制表符等几种方法得到。提供图形符号的机器,例如长城 0520CH,则可直接用图形符号做制表符。在制表时,横线制表符最好直接用西文符号"—",仅占一个字符宽度,而汉字库中的制表符则占两个字符宽度,当表格中有些栏目宽度是奇数位时,若用专门制表符需给奇数位的栏目加一个空格使其变为偶数,打印出来的表格才能对齐。当表格栏目数较多或表格较宽时,若横线采用专门的图形制表符,就易造成一行打不下的现象。

② 字间距压缩

即使采用中文制表符,如不压缩字间距仍不能打印出横向的实线。压缩字间距的命令形式:

? CHR(27)+"P";P 为小写字母

③ 行间距压缩

必须将行间距压缩才能使竖线成为实线。行间距压缩的命令形式是:

? CHR(27)+"行压缩码"

不同的打印机,其行压缩代码不同。3070 打印机,其行压缩代码为:U08 或 U12。

下面给出一个打印实线表格的小程序段。

主机为长城 0520CH,打印机为 3070,打印驱动程序为 3070C·EXE,中文操作系统为 GWBIOS3.0。

SET DEVICE TO PRINT

SET PRINT ON

? CHR(27)+"P"

? CHR(27)+"U08"

@PROW()+1,1 SAY "┌────────┐";

──────────┐"

@PROW()+1,1 SAY "│学习班名称│";

课程设置│"

@PROW()+1,1 SAY "┌────────┐";

──────────┐"

@PROW()+1,1 SAY "│人事财务管理│;操作系统、汉字 dBASE II 等│"

@PROW()+1,1 SAY "┌────────┐";

──────────┐"

@PROW()+16,1 SAY " "

SET DEVICE TO SCREEN

SET PRINT OFF

91 年第三届国际信息学 奥林匹克(IOI'91)试题

吴文虎

第一轮试题

以下列方式向 5×5 矩阵中填入数字, 设数字 $i (1 \leq i < 25)$ 已被置于坐标位置 (X, Y) , 则数字 $i+1$ 的坐标位置应为 (Z, W) 。

(Z, W) 可根据下列关系由 (X, Y) 算出:

- (1) $(Z, W) = (X \pm 3, Y)$
- (2) $(Z, W) = (X, Y \pm 3)$
- (3) $(Z, W) = (X \pm 2, Y \pm 2)$

求解问题如下:

(A) 编写一个程序, 当数字 1 被指定于某个起始位置时, 列举出其它 24 个数字应在的位置, 列举出该条件下的所有可能方案。输出方式如图 1 所示。

图 1

(B) 使数字 1 的起始位置分别处于矩阵的含主对角线的右上三角的每一位置, 计算出每一种情况下的所有可能的方案数。

举例: 如果数字 1 的起始位置坐标被定为 $(2, 2)$ 则数字 2 的可能的坐标位置应为 $(2, 5)$, $(5, 2)$ 或 $(4, 4)$, 上述位置在图 1 中用星号(*)标出。

注意: 输出方式如与图 1 一致则较好。

评分标准: (A) 50 分

(B) 25 分

输出效果 15 分

JURY 10 分

(仲裁委员会) 10 分(程序风格, 清晰程度, 优美程度)

第二轮试题

一个 S 项(S-Term)是字符 S 和括号的序列, 并以递归方式定义如下:

- 字符 S 是一个 S 项。
- 若 M 和 N 是 S 项, 则 (MN) 也是一个 S 项。

S 项的例子如下:

$((((SS)(SS))S)(SS))$

右括号没有提供任何新信息, 所以可被删去。即可用

(MN) 代替 (MN) 。因此上述例子可以变成:

$((((SS)(SSS)(SS$

1. 写一个名为 'gensterm' 的过程以产生 S 项: 该过程应产生 n (n 是字符 S 的数目, 即长度) 个正文文件。这些文件分别含有字符 S 的个数为 $1, \dots, n$ 的所有的 S 项。S 项由 ';' 隔开, 每个文件中的最后一个 S 项后面应跟有符号 '.'。(文件的序号对应 S 字符的个数) 写一个程序, 接受整数 $n (n \leq 10)$, 使用上述过程并在屏幕上显示全部生成的 S 项。

考虑一种对 S 项的运算, 唯一的运算规则(S 规则)为: 若 S 项的子项形式为: $((SA)B)C$

(其中 A, B 和 C 是 S 项) 则该子项可被改写为: $((AC)(BC))$ 也就是说,

$Context1(((SA)B)C)Context2 \rightarrow Context1((AC)(BC))Context2$

对 S 项应用上述规则称为对 S 项的缩简。应用 S 规则来选择子项有不同的方法(策略)。

对 S 项规格化是指: 对 S 项连续使用 S 规则直到不能再使用 S 规则为止。

一个缩减过程的实例如下:

$((((SS)(SS))S)(SS)) \rightarrow (((SS)((SS)S))(SS)) \rightarrow ((S(SS))((SS)S)(SS)) \rightarrow ((S(SS))((S(SS))(S(SS)S)))$

2. 选择适当的数据结构来表示 S 项(要便于应用 S 规则)写两个过程 'read term' 和 'printterm'。'readterm' 把由 'gensterm' 所产生的 S 项的表示转换为你自己的表示形式; 'printterm' 把你的表示形式转换为 S 项的形式。你的程序应能演示上述转换过程。

3. 写一个名为 'reduce' 的过程。对一个指定的 S 项的子项(以你的表示形式给出)使用 S 规则, 完成一个缩减步骤。你的程序应能演示此过程。

4. 写一个名为 'normalize' 的过程。给定一个 S 项, 该过程应能重复地使用 S 规则选择子项, 直到不能再缩减为止; 或者当缩减步骤超过某个最大值(现在给定为 30)时为止, 你的程序应能演示此过程。

5. 最后, 把上述程序合并成一个程序。应能:

- a) 请求用户输入一个长度 n 。
- b) 使用由过程 'gensterm' 产生的长度为 n 的 S 项。
- c) 把 S 项转换为你自己的表示形式。
- d) 若有可能, 将其规格化。
- e) 输出经规格化的 S 项。

f) 输出规格化过程中对每一 S 项进行缩减的步数; 若在 30 步内不能成功地完成规格化, 输出信息 'not normalized'。

g) 对给定长度为 n 的 S 项, 输出: 不能规格化的项的数目相对于总的 S 项个数的关系。即:

不能规格化项数; 总的 S 项数; 不能规格化项数/总的 S 项数。

评分标准:

序号 分数

1 20

2 25

3 15

4 20

5 10

Jury 10(清晰程度、优美程度、程序风格)

Z80 的“新”指令

章永得

近些年来,人们在实践中探索到一批 Z80 的“新”指令。其中一部分可能是制造厂为了简化逻辑结构,从而删减的重复老功能的指令。也有一部分可能是厂家没有公开发表的“新”功能指令。这些指令对于开发 Z80 的软件是很有用的。

其中重复功能指令 310 条,大部分分布在 ED 带头的区域,也有一部分分布在 DD 和 FD 带头的区域,见表一。应当指出的是 DISASSM88 把 ED4E、ED6E、ED66、ED76 和 ED7E 等五条指令都反汇编为 NOP 指令是与实际情况不符的。经过本人验证,其中 ED4E 和 ED6E 本人把它叫做中断方式无定义指令。其功能为取消程序前面定义过的中断方式,直到后面再定义时止。它对可屏蔽中断是有影响的。所以应属于新功能指令。本人把它列于表四中。后三条指令应分别为 IM

0、IM 1、IM 2。这些都是属于重复功能指令,本人把它们列于表一中。

新功能指令共有 440 条,主要分为三部分,一部分是单功能指令,主要是对变址寄存器 IX 和 IY 的高八位及低八位的传送、运算和比较指令共有 92 条,见表二。表中 IXH、IYH 分别表示为 IX、IY 的高八位而 IXL、IYL 是低八位。另一部分是所谓双功能指令,它是由位操作与传送合在一起的复合功能指令,共有 336 条,见表三。最后一部分是一组新的移位指令即逻辑左移指令,见表四。它与原来的一些移位指令组合在一起构成完整的八组移位指令,其功能见表五。

最后谈一下 Z80CPU 对所谓非法指令的响应。它们都分布在 DD 和 FD 带头的区域。当 CPU 遇到非法指令时,则去掉第一个字节看后面各字节,若还是非法的,则再去掉前面一字节,直到后面的是合法时,再执行该指令。如遇到 DDFD1C,则应把 DDFD 都去掉,剩下 1C 是可执行指令。最后,三字节合在一起执行的结果为 INC E。这一点可以由单步执行来验证。在单步执行中,整个三字节为一步,执行完后检查各寄存器的内容,可以发现 E 寄存器的数增加了 1。就这样,Z80CPU 对所有机器码都是能响应的。

本文指令表见本期封二、封三。

《电子质量》杂志

地址:广州沙河东莞庄路竹丝岗 1 号(广州 1501 信箱 9 分箱)

邮政编码:510610

全国各地邮局订阅,邮发代号:46-39。又月刊,年订价:7.50 元

更正

本刊 90 年 8 期《中华学习机通用 EPROM 读写卡》一文刊出后,发现在排印和原文稿两方面都有错误,特更正和修改如下:

1. 290 行中“;A=A+2”为“;A=A+3”之误
2. 340 行中“CETA\$”为“GETA\$”之误
3. 500 行中“RESTURE”为“RESTORE”之误
4. 560 行中“N\$=MID(A\$,K,1)”为“N\$=MID\$(A\$,K,1)”之误
5. 580 行中“RUTURN”为“RETURN”之误
6. 20 行中“H=208”语句移到 40 行行首
7. 200 行中“PRINTN”语句移到 220 行行首,并修改 220 行为:220PRINTN:A=N*16+128:PRINT”选择项正确否?(Y/N);:
- GETA\$:PRINTA\$:IFA\$=“N”ORA\$=“n” THEN200
8. 360 行中“IFB<=C...”应修改为“IFA>=B...”
9. 机器语言主程序自 124 地址开始修改 3 个数据:124;EA EA A9.00

电子工业出版社软件部征稿启事

为丰富中华机、苹果机的软件市场,为广大用户提供更多更好的软件产品,电子工业出版社软件部特向中华机、苹果机软件的开发个人或单位广泛征稿。

一、征稿范围:

1. 系统软件(语言系统、汉字系统、运行环境等);
2. 工具软件(各种实用工具);
3. 应用软件(管理软件、工程计算);
4. 辅助教学软件;
5. 幼儿教育软件;
6. 游戏软件。

二、征稿要求:

1. 提供不加密的软盘(可另附拷贝加密盘);
2. 提供通俗易懂的使用说明(用稿纸写,字迹要清楚);
3. 软件不得一稿两投或已在社会上广泛流传;
4. 征稿软件不包括汉化的国外软件。

三、出版发行:

1. 投稿的软件经“软件评委会”评审通过后,本部寄出软件登录表及出版委托书由作者填报。我社对出版的软件给予版权号;向国内外公开发售;颁发软件出版证书;保护作者权益。

2. 对于出版的软件按软件的发行量确定稿酬。

3. 没有被采用的软件,本部在二个月内退回修改或退稿。

注:作者投稿,可先不寄磁盘而寄软件简介一份,以供我部预审。

联系地址:北京 173 信箱 电子工业出版社 软件部
联系人:吕 迈 电话:815342 邮编:100036

欢迎作(译)者和读者

参加 91 年本刊评刊活动

诸位作(译)者和读者:

时光过的很快,转眼间 91 年已过去大半,根据办刊特点,现在我们已着手安排 92 年的刊物工作了。为此,我们首先想到了热情为本刊撰稿的各位作(译)者们及喜爱本刊的广大读者,希望大家就一年来本刊各方面的工作,提出评审意见和建议,参加到我们办刊工作的队伍中来,运用大家的智慧,使我们自己的刊物越办越好,年年有新意,有进步!

“科学技术是第一生产力,是推动经济和社会发展的强大力量”。

在普及科学技术方面,《电子与电脑》愿以自己微薄之力,奉献给我国经济和技术的发展。

《电子与电脑》读者、作译者意见征询表

姓名		职业		地址及邮编	
您购有或使用何种微电脑	CEC-I□, 天坛□, 小蜜蜂□, APPIe II □, PC□, LASER□, PC-1500□, 单片机□, 单板机□, 其它(机型)_____。				
您订阅本刊有: 1987□, 1988□, 1989□, 1990□, 1991□。					
您对本刊现有栏目的看法及建议	综述□, 学习机之友□, PC 机园地□, 电脑巧开发□, 学装微电脑□, 维修经验谈□, 学用单片机□, C 语言初阶□, 读者联谊□, 新书与软件□, 初级程序员级水平考试辅导讲座□, 信息与服务(广告)□。 建议增加_____, _____, _____, _____。				
您喜欢本刊已刊出文章中的哪几篇	软件方面				
	硬件方面				
您希望刊登哪方面的讲座					
您对本刊印制质量的看法	封面设计□, 版式设计□, 广告设计□, 校对质量□, 正文印刷□, 封面印刷□。 改进意见: _____				
您对本刊开展的读者服务工作有何看法及建议					
您认为本刊在提高知名度方面应做哪些工作					
您有何希望和要求					

表中□内, 同意√, 不同意× 回函寄: 100036 北京 173 信箱《电子与电脑》编辑部