

計算機應用研究

1990

6

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





北京四通新技术产业股份有限公司

计算机事业部

隆重推出

● 万能编程测试卡

无所不能
开发必备
十位一体

● MICE-II 微机在线仿真器

硬件侦错
系统调试
逻辑分析



MICE-II

- | | |
|------------------|----------------|
| 1. EPROM | 2. EEPROM |
| 3. 8751 系列 | 4. 8748 系列 |
| 5. Z86 系列 | 6. BPROM(PROM) |
| 7. 逻辑电路测试 | 8. RAM 测试 |
| 9. PALS GAL 编程 | |
| 10. PEEL/ PLD 编程 | |

仿真CPU: 8085 8086
8088 Z80等
是开发调试微电脑的必要
工具。

广西计算中心简介

广西壮族自治区计算中心(广西计算机技术应用研究所)成立于1978年5月,是专门从事计算机技术应用研究和开发的全民所有制科研单位,目前,共有职工82人,专业技术人员占80%,其中高级工程师、副研究员以上的7人,工程师26人。近年来先后派出了13人到美国、荷兰、新加坡、日本等国学习与交流。该中心利用其毗邻港澳和沿海开放城市的优势,不断加强新、高技术的引进、消化和吸收工作。

该中心目前有3400平方米的科研实验大楼,有2套超级小型计算机,一套Sun工作站,一套AT&T多用户系统,数十套包括286、386在内的高档微型机,有几十种开发、测试、实验和维修设备,有2套轻印刷系统。几年来该中心共完成126项科研课题,获得成果奖46项,其中2项获国家成果证书,广西科技进步奖14项(二等奖6项、三等奖8项),广西计算机应用成果奖16项(一等奖2项,二等奖4项,三等奖10项),广西科学院科技进步奖10项(一等奖2项,二等奖2项,三等奖6项),南宁市科技成果奖1项,其他厅级成果奖5项。其中TGCAD中小型水轮发电机组CAD系统、自治区科委计算机网络管理系统、微机控制垫处理调质线、医药商业企业微机网络管理系统、中医计算机诊疗系统、卷烟厂计算机辅助管理系统、冷库温度记录仪、营养评价与配餐电脑多项成果均在区内外推广应用,取得了显著的社会、经济效益。

该中心在承担各种科研项目时,注意为各行业培养技术骨干,几年来共培训学员7300余人次,其中不少人已成为其单位计算机应用的骨干力量。

该中心1985年成立海蓝电脑公司,利用计算中心雄厚的技术力量,面向全自治区开展计算机销售业务。现在,海蓝电脑公司已成为广西技术力量最强、服务质量最好的计算机专业公司。

为了更广泛地推广普及计算机技术,该中心组织编写、翻译和排印了40多种讲义、计算机应用手册和参考资料,约800万字以上,交流总量超过6万册,范围遍及全国二十多个省市。该中心科技人员扎实工作所取得的成绩,受到各行业的高度赞誉和广大用户的好评。

该中心目前开展以下业务:1、进行软件及其有关技术的研究与开发:承担计算机管理信息系统(MIS)、决策支持系统(DSS)、办公自动化系统(OAS)、计算机辅助设计、制造、测试(CAD/CAM/CAT)、网络技术应用、知识工程和专家系统、高级信息处理、图像处理、语音合成等课题。2、进行硬件研究与开发:承担计算机实时控制、智能仪器仪表研制、机电一体化产品研制等项目。

3、引进先进的软硬件,进行消化、开发和推广应用;研究、开发并向用户提供软硬件的系列产品。

4、销售电子计算机设备、开展计算机技术培训,提供计算机维修和机房装修服务。

5、提供计算机录入、排版、印刷和复印等服务。

6、向用户提供计算方法和计算机机时。

7、开展有关电子、电脑的科、工、贸方面的国内、国际合作。

8、组织计算机学术技术交流。

通讯地址:广西南宁市星湖路五号

邮政编码:530022

电话:(0771)42945,42627,23961

电报挂号:南宁 5454

《计算机应用研究》杂志办刊单位

四川省电子计算机应用研究中心
安徽省计算中心
吉林省计算中心
甘肃省计算中心
四川省电子学会

贵州省科委计算中心
新疆电子计算中心
广西计算中心
青海省测试计算中心

《计算机应用研究》杂志编辑委员会

主任委员：张执谦
副主任委员：李泽民
委员：贾洪钧 张国栋 罗海鹏
曾光初 王小华 朱景生
余 凯

《计算机应用研究》杂志社董事会

董事长：周赛渝
董 事：唐 珍 郑国基 陆慰椿
秦小竹 龚宇清 吴地兴
黎瑰常

计算机应用研究（双月刊）

（公开发行）

一九九〇年

第七卷 第六期（总第38期）

主 编：张执谦

副 主 编：李泽民

本期责任编辑：张 钢

编 辑 出 版：《计算机应用研究》杂志社

内 文 印 刷：西南冶金地质印刷厂

封 面 印 刷：四川省印刷制版中心

出 版 日 期：1990年11月30日

本刊通讯地址：成都市人民南路4段11号附1号

本刊邮政编码：610015

订 阅 处：全国各地邮局

总 发 行：成都市邮政局

每册定价：1.30元

本刊邮发代号：62—68

国内统一刊号：CN51—1196

广告经营许可证：川蓉工商广字005号

《计算机应用研究》第7卷第6期 (总第38期)

目 录

软 件 篇

Turbo Prolog 程序设计技巧	李春葆 章启俊 马玉枫	(1)
Turbo Pascal 程序设计中几个难题的最新解决方法	唐福益	(4)
Turbo Pascal 图形数组及应用	沈国坤 左尧凤 杨 丽	(8)
用高级语言编制过程控制软件	阎建明	(12)
利用软件黑盒子对PC机文本文件加密的原理与方法	于功弟 路 枝	(14)
硬盘加锁技术及实现方法	王晓武	(17)
最新的FOXBASE+2.10和它的开发工具	杨剑波 西 捷	(19)
dBASE III—PLUS下的窗口操作的设计	周步祥 阮树骅	(22)
在数据库中如何快速处理多层次合计的表格	蒋仁兴	(24)
防断电可中断的登帐程序设计	杨传斌	(25)
“地震活动信息熵与分数维”研究的软件设计	莫宇清 周仕勇	(26)
一个多机环境下的集成CAD系统	张小平 巫晓龙	(30)
印制电路板图形的分块输出	方旭明 徐 锐	(32)
电视动态字幕的微机实现	李志伟	(35)

硬 件 篇

MCS—51单片机多外中断扩展方法	骆德汉	(36)
PC—1500两套供电系统	曹来发	(55)

系 统 篇

贵州省科技管理信息系统数据库的设计与实现	贾洪钧	(39)
集散型空调计算机控制系统	叶定治 李淑莲	(43)
超长重载列车制动和缓解时间采集与处理	隗永安	(50)

维 修 篇

用软件方法查找硬盘适配卡故障	梁 军	(52)
PC/XT故障维修两例	戴 永	(54)
测试异步通讯适配器	冉春林	(23)

信 息 篇

《中华学习机系统盘》(CEC—DOS)简介	邵爱柏	(56)
SMD—6410系统捷足先登	马在强	(57)
《Z80汇编语言BCD码长浮点常用函数运算符程序库》简介	张铁钢	(57)
微机测试婴幼儿智能状况程序问世	李相彬	(17)
《计算机应用研究》一九九〇年总目录		(58)

APPLICATION RESEARCH OF COMPUTERS

Vol.7 No.6 (Total 38)

CONTENTS

SOFTWARE

The Technique about Programming of Turbo Prolog	Li Chunbao and Others (1)
The New Method of Solving Problem with Programming of Turbo Pascal.....	Tang Fuyi (4)
The Graphic Array of Turbo Pascal and It's Apply	Shen Guokun and Others (8)
Programming the Software with Procedure Controdding by High Level Language.....	Yan Jianming (12)
The Fundamentals and Method of ASC II File Encryption by Software Black-box on PC Computer.....	Yu Gongdi and Others (14)
The Encryption Technique of Hard-disk and It's Realize	Wang Xiaowu (17)
The New Foxbase Plus 2.10 and It's Developing Tools	Yang Jianbo and Others (19)
Designing of Aperture Operation on dBASE III Plus	Zhou Buxiang and Others (22)
How Quick Deal with Total Table in Data Base.....	Jiang Renxin (24)
Programming with Registration Could Interruput for Prevent Cutting off the Power Supply.....	Yang Chuanbin (25)
The Designing Software for the Research of "Seismic Activity Infor- mation and Entropy & Fraction Dimension.....	Gong Yuqing and Others (26)
Integrated CAD System in Multicomputer Environment	Zhang Xiaoping and Others (30)
The Output with Separated Block by PCB Graph	Fang Xuming and Others (32)
Realizing of TV Dynamic Character Screen on Microcomputer	Li Zhiwei (35)

HARDWARE

The Expanded Method for Multilayer Interrupt of MCS-51 Single Chip Processor.....	Luo Dehan (36)
Two Power Supply Systems on PC-1500.....	Cao Laifa (55)

SYSTEM

Designig and Realizing for Data Base of Science and Technology Management Information System of Guizhou Province.....	Jia Hongjun (39)
--	--------------------

Controlled System of Collecting and Distributing Air Conditioner by Microcomputer.....	Ye Dingzhi and Others (43)
Collecting and Processing of Braked and Relaxed Times of Overload Train.....	Kui Yongan (50)

MAINTENANCE

Searching the Fault of Hard-disk Adapter by Software.....	Liang Jun (52)
Two Examples about Breakdown Maintenance on PC/XT	Dai Yong (54)
Testing for Asynchronous Communication Interface Adapter	Ran Chunlin (23)

INFORMATION

Brief Introduction about DOS of China Education Computer	Shao Aibai (56)
SMD-6410 System-The Race is to the Swiftest.....	Ma Zaiqiang (57)
Brief Introduction about "The Operation Subprogram Library in Common Use Function of Long Floating Point of BCD Code of Z80 Assembler"	Zhang Tiegang (57)
Testing Program for Intelligence State of Baby and Child Come out	Li Xiangbin (17)
Comprehensive Table of Contents of "Application Research of Computers" in 1990	(58)

APPLICATION RESEARCH OF COMPUTERS

(Publicly Publishing) (Bimonthly)

November 1990 Volume 7 Number 6 (Total 38)

Main Editor: Zhang Zhi Qian, Assistant Main Editor: Li Ze min

Responsibility Editor for this Issue: Zhang Gang

Editor/Publisher: Editorial Department of This Periodical

Address: 11-1 Four Section Ren Ming Nan Lu, Chengdu, P.R.C

Press: South West metallurgical geology Printing house of

Ministry of Metallurgical Industry

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

Central Dispatching in Inside the Country: The Post of Chengdu

Unified Number in Inside Country: CN 51-1196

Number of the Post Publishing: 62-68 Post Code: 610015

Turbo Prolog 程序设计技巧

武汉测绘科技大学 李春葆 章启俊 马玉枫

摘 要 本文介绍了Turbo Prolog的递归程序设计和非完全递归程序设计方法,详细地讨论了动态数据库在程序设计中的应用以及在使用动态数据库时应注意的若干问题。

prolog语言是一种描述型的程序设计语言,它提供了模式匹配和回溯机制,这些特点构成了prolog与其它高级语言如C、pascal等过程型语言在程序设计风格上的主要差别,Turbo prolog保留了prolog的所有特点,另外除提供强有力的编辑器、编译器和跟踪器外,Turbo pascal采用了非确定性程序设计技术,即一次找到问题的所有解。因此,Turbo prolog是进行人工智能程序设计强有力的工具。下面我们介绍使用 Turbo prolog 在编程技巧方面的问题。

一、递归程序设计

采用 Turbo Prolog 编程,首先要将原问题转化成逻辑形式,然后再把逻辑形式转化为递归形式,而不是象过程型语言直接求解原问题。递归是Turbo Prolog最重要的特点。

一般地,递归程序是由递归出口和递归体两部分构成。递归出口反映了递归终止的条件,此时递归结束,同时开始求值过程,递归与求值是两个相反的过程。递归体反映了原问题递归的组织形式,Turbo Prolog 的问题求值过程就隐含在这种递归组织之中,实质上讲递归求解问题就是把原问题分解成若干个子问题来求解,每个子问题求解的方式和原问题求解的方式是相同的,这就必须具备一种递归型的数据结构存放原问题。我们知道,Turbo Prolog 中的重要数据结构表满足这个条件。图1所示是递归的一般形式,其中f是递归函数,L是表,car是表头函数,cdr是表尾函数,op是一种运算,其递归含义为:

$$f(L) = op(f([car(L)]), f(cdr(L)))$$

这样递归函数完全由 op 和递归出口条件确定。

上面递归形式由Turbo Prolog 写成为:

$$f(L) :- op(f([car(L)]), f(cdr(L))).$$

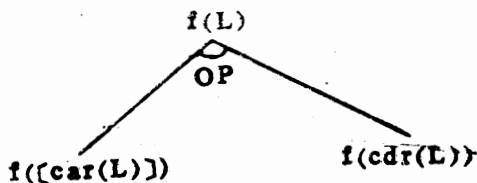


图1 递归的一般形式

一般地,把一张表分解成 car(L) 和 cdr(L) 是在规则的左部进行的,上面形式等价地写成为:

$$f([A | L]) :- op(f([A]), f(L)).$$

另外还加上递归出口,整个程序就生成了,我们看看几个例子。

例1. 表长函数 length.

通过分析有如下递归形式:

$$\text{Length}(L) = \begin{cases} 0, & \text{if } L = [] \\ \text{add}(\text{length}([car(L)]), \\ \text{length}(\text{cdr}(L))). & \text{else.} \end{cases}$$

显然 $\text{length}([car(L)]) = 1$

这样对应的Turbo Prolog程序为:

$$\text{length}([], 0) :- !.$$

$$\text{length}([_ | L], M) :- \text{length}(L, N), \\ M = N + 1.$$

例2. 求整数表中最大元素max函数。

递归形式为:

$$\text{max}(L) = \begin{cases} 0, & \text{if } L = [] \\ \text{maxnumber}(\text{max}([car \\ (L)]), \text{max}(\text{cdr}(L))), & \text{else} \end{cases}$$

其中 maxnumber 运算是两个整数中取大者。

对应的Turbo Prolog程序为:

$$\begin{aligned} \text{max}([], 0) & :- !. \\ \text{max}([A | L], M) & :- \text{max}(L, M1), \text{maxnumber}(A, M1, M). \\ \text{maxnumber}(A, B, A) & :- A > B, !. \\ \text{maxnumber}(_, B, B) & . \end{aligned}$$

例3. 判定两个表 L_1, L_2 是否相同的函数 `compare`.

递归形式为:

`compare(L1, L2)`

$$= \begin{cases} \text{yes, if } L_1 = L_2 = []; \\ \text{no, if } L_1 = [], L_2 \neq []; \\ \text{no, if } L_1 \neq [], L_2 = []; \\ \text{and, (compare(car(L}_1), \\ \quad \text{(car(L}_2) \text{compare} \\ \quad \text{(cdr(L}_1), \text{cdr(L}_2))). \end{cases} \quad \text{else.}$$

Turbo Prolog 程序为:

```
domains
charlist=char*
predicates
compare(charlist,charlist)
same(charlist,charlist)
clauses
compare([],[]):-!.
compare([],_):-not(same(L,[])),!.
compare(L,[]):-not(same(L,[])),!.
compare([A|_],[B|_]):-A<>B,!,fail.
compare([_|L1],[_|L2]):-compare(L1,L2).
same(L,L).
```

二、非完全递归程序设计

前面我们看到了递归程序设计的一般方法, 我们把这样的递归过程称为完全递归过程, 现在让我们再看另一种递归过程, 如图2所示, 同样, 表 L 分解成 `car(L)` 和 `cdr(L)`, 从这里看, 它是递归的, 但 f 还与一个全局变量 G 有关, G 不是随 L 的递归分解而分解, 因此 G 是非递归的, 我们称这种过程为非完全递归过程。

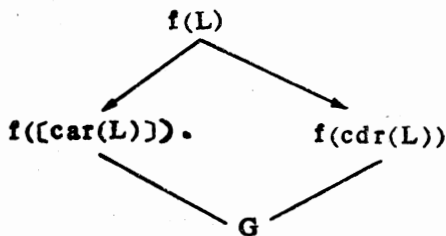


图2 非完全递归的一般形式

我们知道, 在 Turbo prolog 中, 不存在全局变量, 所有的变量都是局部变量, 因此为了达到同样效果, 只有通过局部变量的值传递来完成。另外, 在 Turbo Prolog 中的 “=”, 不仅具有赋值功能, 同样还取谓词作用, 即判定左

右两边是否相等, 因此 “ $G=G+1$ ” 为永假。为了达到变量的值变化的效果, 可以通过函数之间的调用来完成, 这就增加了编写非完全递归程序的难度。一般地, 图2所示的非完全递归可以等价地转化为图3所示的递归形式。这样, 非完全递归形式所对应的递归形式为:

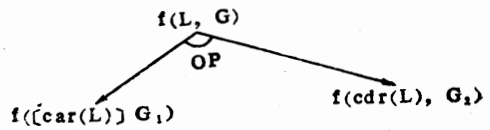


图3 与图2等价的递归形式

$$f(L, G) : -op(f([car(L)], G_1), \\ f(cdr(L), G_2)).$$

上述形式对应的 Turbo Prolog 语句为:

$$f([A|L], G) : -f(L, G_1), f([A], \\ G_2), op(G_1, G_2).$$

例4. 把表 L 中为字母的元素放入 CL 表中的函数 `newlist(L, CL)`.

由上面的转化过程得到:

`newlist([A|L], CL)`

$$= \begin{cases} \text{newlist(L, CL}_1\text{), put-in(A,CL}_1\text{,} \\ \quad \text{CL), if A为字母;} \\ \text{newlist(L, CL), else.} \end{cases}$$

因此对应的 Turbo prolog 程序:

```
domains
charlist=char*
predicates
newlist(charlist,charlist)
is_char(char)
clauses
newlist([],[]):-!.
newlist([A|L],[A|CL]):-is_char(A),!,newlist(L,CL).
newlist([_|L],CL):-newlist(L,CL).
is_char(A):-A>='a',A<='z';
A>='A',A<='Z'.
```

例5. 判定一个表 L 中左右括号是否配对的函数 `match(L)`.

依照上面的程序设计思想可以产生程序:

```
match(L):-match1(L,[]),!.
match1([],[]):-!.
match1([A|L],S):-A='(',!,match1(L,S1),
add(A,S1,S).
match1([A|L],S):-A=')',!,match1(L,S1),
not(same(S1,[])),delfirst(S1,S).
match1([_|L],S):-match1(L,S).
same(L,L).
delfirst([_|S],S).
add(A,S,[A|S]).
```

但是上述程序在运行中产生错误的结果,即对括号配对的表 L,都回答 no. 为什么产生这种结果呢?主要是这里使用的全局变量 S 是取堆栈作用,它的初值为空,而只有当最终结果亦为空时,才能得出 L 中括号匹配成功的结论,也就是说 S 的初值与终值是相关的。这种非完全递归问题用图 3 所示的方法不易解决。

我们知道 Turbo Prolog 中提供了动态数据库,动态数据库的内容是全局有效的。这样我们把上面程序修改为:

```
match(L):-cut_all,assertz(stack([])),
            match1(L,! ,stack(R),same(R,! ),!
match1({})::-!.
match1([A|L]):-A='(',! ,push,match1(L).
match1([A|L]):-A=')',! ,pop,match1(L).
match1([_|L]):-match1(L).
push:-stack(S),retract(stack(S)),
      assertz(stack(['('|S])).
pop:-stack(R),!,not(same(R,!)),
     stack([A|S]),retract(stack([A|S])),
     assertz(stack(S)).
cut_all:-retract(stack(_)).
cut_all.
same(L,L).
```

其中 stack(R) 是动态数据库谓词。这种设计方法是把图 2 的非完全递归形式转化成图 4 所示的递归形式。图 4 与 stack(S) 相关联的过程不参与表 L 的递归分解,故它对应的是递归过程。

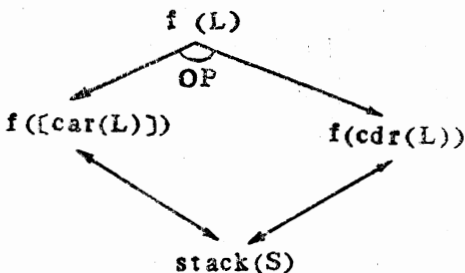


图4 与图2等价的递归过程

可以说所有的非完全递归过程都可以化为图 4 所示的递归过程来求解。例 4 采用动态数据库设计的程序为:

```
char_list(L,L1):-cut_all,assertz(stack([])),
                  newlist(L),stack(L1),!.
newlist({})::-!.
newlist([A|L]):-is_char(A,! ,updata_stack(A)
                  newlist(L).
newlist([_|L]):-newlist(L).
is_char(A):-A>='a',A<='z';
            A>='A',A<='Z'.
updata_stack(A):-stack(L),retract(stack(L)),
                 assertz(stack([A|L])).
cut_all:-retract(stack(_)).
cut_all.
```

在 Turbo Prolog 中,在使用动态数据库时,一定要注意动态数据库的定义、初使化和修改。动态数据库的定义为:

```
database
stack(data_type)
```

其初使化和修改过程分别为:

```
cut_all:-retraet(stack(_))
cut_all.

proc:-stack(S),retract(stack(S)),
      updata(S,NS),assertz(stack(NS)).
```

三、非完全递归设计方法的选择

我们前面介绍了两种非完全递归设计方法,分别称为方法 1 和方法 2。一般情况下两种方法可以相互转换,下面是采用两种方法进行设计的程序结构。在功能上是等价的。

```
program({})::-!.
program([A|L]):-cond-1(S),
                process-1(S),
                call-1(A),
                program(L).
```

```

:
:
:
```

```
program([A|L]):-cond-n(S),
                process-n(S),
                call-n(A),
                program(L).
```

```
program([],[])::-!.
program([A|L],S):-cond-1(S),
                  process-1(S,S1),
                  call-1(A),
                  program(L,S1).
```

TURBO PASCAL 程序设计中几个难题的最新解决方法

石家庄铁道学院 唐福益

摘要 程序代码长度或参加运算的数据量超过内存限制, 可调数组的实现, 以及过程和函数作为参数传递: 这三个难题时常困扰着用 TURBO PASCAL 的程序设计者。经过对 TURBO PASCAL 的分析研究, 和在软件设计中的反复探索, 本文提出用覆盖、强制类型转换和过程类型作为解决上述难题的最新方法。从而使得小内存运行大程序和设计通用程序都变得轻而易举。

TURBO PASCAL 语言以其功能强, 类型丰富, 数据结构灵活, 能递归, 便于结构化程序设计而倍受青睐。但由于它本身的局限性, 使得在大型、复杂的程序设计中, 常遇到摘要中所提的三个难题, 使它的应用受到一定限制。近几年来, Borland 公司不断发展更新该语言, 使其成为国际微机 PASCAL 语言的主流。1988 年推出的 TURBO PASCAL 5.0 版更具特色, 它集中了以前版本的全部优点, 增加了过程类型,

覆盖单元和覆盖管理工具, 特别是它给用户提供了一个极好的集成开发环境。在集成开发环境中, 无论是在编译或运行时, 都能自动定位出错位置并给出错误信息; 用户能在源程序级上调试程序: 能单步执行, 设置断点, 观察变量或表达式值的变化, 以及能跟踪程序的执行过程。所以 TURBO PASCAL 5.0 无论是对初学者或是对于大型、复杂程序的设计者, 均是强有力的程序开发工具。本文的方法就是基于 5.0 版而提

```

:
:
:
program([A|L],S):-cond-n(S),
               process-n(S,Sn),
               call-n(A),
               program(L,Sn).

```

```

run(N,M):-assertz(stack(N)),exec(M).
exec(M):-stack(N),N<0,N1=N+1,
          retract(stack(N)),
          assertz(stack(N1)),
          M=3.
exec(M):-stack(N),N>0,N1=N-1,
          retract(stack(N)),
          assertz(stack(N1)),
          M=1.
exec(M):-stack(N),N=0,M=2.

```

实验证明, 对于累加型的全局变量, 即其初值为空或某定值, 然后在递归中不断增加其值, 递归结束后, 其值就是所要求的结果。这种情况使用方法 1 是比较好的。相反, 对于作为堆栈的全局变量采用动态数据库更合适, 即采用方法 2。

在采用方法 2 进行程序设计时, 容易忽略堆栈的变化对递归的影响, 看如下例子。

例 6. 函数 $f(n, m)$ 为:

$$f(n, m) = \begin{cases} f(n-1, 3) & \text{if } n > 0; \\ f(n, 2) & \text{if } n = 0; \\ f(n+1, 1) & \text{if } n < 0. \end{cases}$$

程序为:

结果是错误的, 当运行 $\text{run}(1, M)$ 时, 结果为 2 个解, 即 $M=3, M=2$ 。这是由于忽略栈里值已经变化而造成的。这种程序用方法 1 是十分容易的。

四、结论

Turbo Prolog 是一种适合于递归程序设计的系统, 在问题求解时, 要把原问题转化成递归问题。有些原问题并非是递归的, 这样就引入了递归程序设计和非完全递归程序设计。我们介绍了非完全递归程序设计的两种方法, 对于增值型的全局变量采用方法 1, 对于堆栈型的全局变量采用方法 2 更适合。

* 本文所有程序均上机调试通过。

出的。

一、程序代码或数据总量超过内存限制的解决办法

由于 TURBO PASCAL 的内部结构和寻址方式决定了代码段、数据段和任何数据结构均不能超过 64KB。一个程序可以分成一个主段和多个单元,只要它们各自的代码段不超过 64KB,就解决了大型程序代码超过 64KB 的问题。但 TURBO PASCAL 编译器把所有类型的常量和各种变量(注意:不仅是全局变量,局部变量也在内)全部集中在数据段。这使得程序中数据量超过 64KB 或代码总和超过机器内存量时,程序设计走到山穷水尽的地步。通过实践,发现覆盖技术是使柳暗花明的好途径。覆盖技术就是让不必同时驻留内存的程序代码或数据共享某一固定内存区域,只在它们各自活动时才进入内存,不活动时,内存让给别的活动代码或数据而覆盖掉它自己。覆盖分代码覆盖和数据覆盖,下面分别介绍其实现方法:

1. 代码的覆盖技术

TURBO PASCAL 5.0 版增加了标准单元 Overlay 单元,用来实现覆盖的管理。它允许以单元为单位进行覆盖,也就是说单元是程序中能够装入覆盖缓冲区的最小单位,覆盖缓冲区是所有可以覆盖的单元共享的一块内存区域。为此,我们首先应根据问题的需要,把那些不需同时驻留在内存的程序代码段,按逻辑关系分成若干个单元,分别把它们编写并编译成可以被覆盖的单元代码。覆盖程序的设计与普通程序的设计差别不大,只需遵循下列规则就能实现代码覆盖。

(1) 在可覆盖单元开头处加上编译指令 `{O+, F+}`, 在所有要调用覆盖单元中的过程或函数的单元开头处加上编译指令 `{F+}`, 单独编译。

(2) 在程序开头处加上 `{SF+}` 编译指令;用 `Uses` 子句指明所使用的单元名,单元 Overlay 应列在所有覆盖单元名之前;在 `Uses` 子句之后用编译指令 `{$Unitename}` 列出所有可覆盖的单元名,以告诉编译器把它们编译成可

覆盖代码。

(3) 先初始化覆盖管理程序,必须保证在任何覆盖单元的初始化之前,先执行过程调用 `Ovrinit('file name · OVR')`。最好的办法是:把所有可覆盖单元的初始化代码集中起来,编写成一个非覆盖单元,它的名子列在 `Uses` 子句中其它可覆盖单元名前而在 `Overlay` 之后,且它的第一个初始化语句是过程语句 `Ovrinit('file name · OVR')`

(4) 把文件 `file name · PAS` 编译到磁盘,即把集成环境中编译菜单命令项 `Compil/Destination` 设置成 `Disk`。然后编译主文件 `file name · PAS`。

下面的例 1 是个简单的覆盖程序例子。单元 `Overu1` 是个非覆盖的初始化单元,它给数组 `a` 和 `b` 置初值,过程语句 `Ovrinit('Overtest · OVR')` 初始化覆盖管理程序,这些初始化代码是在程序开始执行前完成的。单元 `Overu2` 和 `Overu3` 是可覆盖的,在它们的代码开头均有 `{O+, F+}` 编译指令。在程序 `Overtest` 的开头有 `{SF+}` 编译指令,在程序头后的 `Uses` 子句中依次列出了单元 `Overlay`、`Overu1`、`Overu2`、`Overu3` 和标准单元 `printer`,紧接着用 `{O Overu2}` 和 `{O Overu3}` 编译指令,让编译器把单元 `Overu2` 和 `Overu3` 编译成可覆盖的代码。把 `Compil/Destination` 设置成 `Disk` 后,编译(F9 热键)文件 `Overtest · PAS`,就在磁盘上产生一个可执行文件 `Overtest · EXE` 和覆盖文件 `Overtest · OVR`,后一个文件就是初始化语句 `Ovrinit('Overtest · OVR')` 要打开的覆盖文件。

2. 数据的覆盖

覆盖单元的数据(无论全局量或局部量)都不会在覆盖缓冲区出现而被覆盖掉,编译器把程序和各单元(无论是否可覆盖)的任何类型的常量和变量都集中到数据段。对于超过 64KB 的数据可采用下列两种覆盖方法:

(1) 用动态变量实现覆盖。把程序或单元中占用内存多或不需同时驻留在内存的那些变量说明成动态变量,在它们需要活动时,用过程

传递。

Turbo Pascal 5.0版增加了过程类型, 用户在类型说明部分可以定义自己所需要的过程类型标识符。过程类型一旦被定义, 就可象普通类型那样来使用; 可用它来定义具有过程类型的变量和形式参数。允许把过程和函数当作能赋给过程变量、传递给参数的对象; 它仅要求被传递的过程和函数在 {SF+} 状态下编译, 且与过程类型标识符在参数的个数和对应的类型上相同, 对于函数结果类型需一致。后面例 2 是利用过程类型实现函数作为参数传递的演示程序。程序的类型说明中定义了过程类型标识符 Fun, 函数 ITE 中设了一个类型为 Fun 的参数 fn, 程序中用参数个数及类型和结果类型都与 Fun 相同的函数 f, fl 和 fs 作实参与 fn 结合, 实现了函数作参数的传递。请注意: 要作参数传递的过程和函数必须在 {SF+} 状态下编译, (即在它们前面要加 {SF+} 编译指令, 如例程中所示); 嵌入式过程和函数, 包含过程和中断过程不能作为参数传递。

例 2. 用辛普森积分法计算定积分:

$$\int_0^{\pi} \sin(x) dx, \int_1^{10} \frac{1}{x} dx \text{ 和 } \frac{1}{\pi} \int_0^{\pi} \cos(t -$$

$\sin(t)) dt$ 的程序清单和运行结果如下:

```
program products3;
(SF+)
uses printer;
type
  fun=function(x:real):real;
var s:real;
(SF+)
function f(x:real):real;
begin f:=sin(x) end;
function fl(y:real):real;
begin
  if y < 0 then fl:=1/y
  else writeln('parameter error')
end;
function fs(t:real):real;
begin
  fs:=cos(t-2.5*sin(t))
end;
(SF-)
function ITE(a,b,eps:real;fn:fun):real;
var
  s,h,old,new,two,four,ends:real;
  n,i:integer;
begin
  h:=abs(b-a)/2.0; n:=1; two:=0;
  ends:=fn(a)+fn(b); four:=fn(h);
  new:=(ends+4.0*four)*h/3.0;
```

```
repeat
  old:=new; h:=h/2.0; n:=n*2;
  two:=two+four; four:=0; s:=a+h;
  for i:=1 to n do
  begin
    four:=four+fn(s); s:=s+h*h;
  end;
  new:=(ends+2.0*two+4.0*four)*h/3.0;
until (abs(new-old)<eps) or (n>5000);
ITE:=new;
end;
BEGIN (main)
  writeln(lst,'the results of sin(x) : ',ITE(0,pi,1.0e-5,f):10:6);
  writeln(lst,'the result of 1/x : ',ITE(1,10.0,1.2e-5,fl):10:6);
  s:=ITE(0,pi,1.0e-5,fs)/pi;
  writeln(lst,'the result of cos(t-2.5*sin(t)) : ',s:10:6)
END.
```

```
the results of f(x) : 2.000000
the result of 1/x : 2.302500
the result of cos(t-2.5*sin(t)) : 0.497094
```

三、TURBO PASCAL 中可调数组的实现方法

所谓可调数组, 就是维数和数组元素类型固定, 但每维的上下界可变的数组。可调数组是编写各种数组运算的通用程序中必不可少的, 但 PASCAL 和 TURBO PASCAL 均不允许定义可调数组。虽扩充后的标准 PASCAL 允许形参是可调数组, 但各机器上的编译程序均尚未实现。经探索, 用无类型形式参数与变量强制类型转换和单元功能与限定标识符结合均可实现可调数组。

1. 用无类型变参与变量强制类型转换相结合实现可调数组

Turbo pascal 4.0 以上版本允许过程和函数的形式变量参数是无类型的, 在调用时它可以与任何类型的实变量相结合; 于是可用无类型变参来传递任何类型的数组。遗憾的是用这种方法传递数组, 在过程中对该无类型变参的引用只能引用整个数组, 而不能直接引用数组元素。Turbo pascal 的变量强制类型转换功能允许把一种类型的变量强制转换成另一种类型来引用。所以, 我们可以用无类型形式变量参数传递数组, 并在过程和函数中把该参数强制转换成一维数组来引用它的元素。为了使多维数组 $A [i_1, \dots, u_1, i_2, \dots, u_2, \dots, i_n, u_n]$ 的元素 $A [i_1, i_2, \dots, i_n]$ 与一维数组 sa 的元素 $sa [K]$ 一一对应, 必须建立下标间的一一映射函数 $K = f(i_1, i_2, \dots, i_n)$ 。我们知道 PASCAL 语言中多维数组

Turbo Pascal 图形数组及应用

中国科学院成都生物研究所 沈国坤 左尧凤 杨丽

摘 要 本文主要分析Turbo Pascal图形数组的结构并进行其变换,以实现人面肖像、动画图片、汉字字模、广告牌等块图的多重显示方式、打印机制图、图形放大、超显示屏图形和图形的快速存取等功能。

Turbo Pascal 提供大量实用的图形功能过程,4版又在3版基础上作了改进,因此为用户设计计算机图形程序带来极大方便。对于一些无规则的块图,例如人面肖像,标本照片,动画图片,汉字字模,了解图形数组的结构和变换方

法,将有助于实现屏幕的多种显示方式,打印机制图、图形放大,汉字字模和图形的快速存取等功能。

一、图形数组Bitmap的结构

IBM PC/XT机CGA型显示屏上的像素

的元素在内存是以行主次序存放成一维向量的,因此一一映射的函数 f 具有下列形式:

$$f(i_1, i_2, \dots, i_n) = \sum_{R=1}^{n-1} ((i_R - l_R) * \prod_{j=R+1}^n (u_j - l_j + 1)) + (i_n - l_n + 1)$$

其中 u_j, l_j 是第 j 维下标的上下界,只要把它们用值参传递到过程中,就可以实现可调数组功能了。例3以二维数组相乘演示了这种方法,过程中的无类型形式变参 y 在第一、二次调用时,分别与实数组 b 和 d 相结合,数组 b 和 d 每维的上下界均不同,这就实现了可调数组功能。一维数组类型标识符 sa 在主程序中被定义,是为了便于定义它的类型和上下界;让它占用64KB内存是保证任何一个多维数组的元素都能映射成一维数组 sa 的元素。由于一一映射的函数关系,使得 $sa(y)[(K-l_2)*(u_3-l_3+1)+(j-l_3+1)]$ 间接引用了数组元素 $b[K, j]$ 或 $d[K, j]$ 。

例3的程序清单和运行结果见右边:

四、结束语

本文介绍的方法,除程序代码段的覆盖和过程与函数作为参数传递两者必须在5.0版才能实现外,其余的均能在4.0版上实现。文中各实例程序只是作演示用,未顾及程序设计的结构和技巧问题,但都在PC机上运行过。由于篇幅所限,使用单元功能与限定标识符相结合,实现可

```

program arrayex;
uses printer;
const max=10000;
type sa=array[1..max] of real;
var a:array[-2..2,3..6] of real;
    b:array[3..6,0..3] of real;
    c:array[-2..2,0..3] of real;
    d:array[0..3,-1..1] of real;
    e:array[-2..2,-1..1] of real;
    i,j,k:integer;
procedure mult(var x,y,z:l1,u1,l2,u2,l3,u3:integer);
var i,j,k,dexm,dexy:integer; s:real;
begin dexm:=u2-l2+1; dexy:=u3-l3+1;
  for i:=l1 to u1 do
    for j:=l3 to u3 do
      begin s:=0;
        for k:=l2 to u2 do { s:=x[i,k]*y[k,j] }
          s:=s+a[(i-l1)*dexm+(k-l2)*dexy+(j-l3)*dexy];
        sa[(i-l1)*dexm+(j-l3)*dexy] := s { z[i,j]:=s }
      end;
    end;
  BEGIN { main }
  for i:=2 to 2 do for j:=3 to 6 do a[i,j]:=i+j;
  for i:=3 to 6 do for j:=0 to 3 do b[i,j]:=i+j;
  for i:=0 to 3 do for j:=1 to 1 do d[i,j]:=i+j;
  mult(a,b,c,-2,2,3,6,0,3);
  mult(c,d,e,-2,2,0,3,-1,1);
  write('the results of multiplication a*b:');
  write('the results of multiplication c*d:');
  for i:=2 to 2 do
    begin
      for j:=0 to 3 do
        if j=0 then write('1:');
        else write('2:');
        write('1:');
        for j:=1 to 1 do write('1:');
        write('1:');
      end;
      write('1:');
    end;
  END.

```

the results of multiplication a*b:				the results of multiplication c*d:			
0.00	50.00	100.00	150.00	1000.000	200.000	300.000	400.000
0.00	60.00	120.00	180.00	1200.000	240.000	360.000	480.000
0.00	70.00	140.00	210.00	1400.000	280.000	420.000	560.000
0.00	80.00	160.00	240.00	1600.000	320.000	480.000	640.000
0.00	90.00	180.00	270.00	1800.000	360.000	540.000	720.000
0.00	100.00	200.00	300.00	2000.000	400.000	600.000	800.000

调数组和过程与函数作为参数传递的方法不再赘述。

参考文献

- [1] 《TURBO PASCAL 5.0参考手册》, 中国科学院希望高级电脑技术公司, 1990: 5

与内存地址一一对应, 偶行 (0, 2..., 198) 和奇行 (1, 3..., 199) 的内存地址分别为 B8000H~B9F3FH 和 BA000H~BBF3FH。在 200×640 黑白模式下, 每个字节对应 8 个像素, 二进制 1 为亮, 0 为暗; 200×320 彩色模式下, 每个字节对应 4 个彩色像素。下述均以 4 版本, 黑白模式为例。

在显示屏上截取宽 N , 高 m 的矩形图形窗口, 在窗口内各行的每 8 个像素组成一个字节, 行末字节如有不完整截取, 则窗外像素用 0 替代, 行向有 $n = (N+7) \div 8$ 字节, 由此组成一个 $m \times n$ 元素的数组。

为了确定图形的高与宽, 在数组前加 4 个字节, 第 1, 2 字节为宽度 n 值, 第 3, 4 字节为高度 m 值。因此图形数组大小为 $4+n \times m$ 个字节, 取名为 BitMap。

二、图形数组的建立

1. 手工编制

如要在宽 10, 高 10 的窗口内放一个字母 m , 计算得 BitMap 大小为 $4+2 \times 10$ 。其图形的亮为 1, 暗为 0, 组成 m 字形。对于制图中常用的符号, 块图可预先编制好, 需要时直接调用。

2. 汉字字模可从 CCLIB 字库中读取

CCDOS 启动后, 汉字库的首地址的段值放在 00042H 单元中, 偏移值放在 00075H 单元中。由区位码 $code$ 计算该码汉字模的首地址。参见 FHead 和 ReadZimo 过程。

也可将 CCLIB 作为一个文件, 由 Seek 过程查找, 参见 HanSeek 过程。这对于 24×24 或更高密度点阵的汉字模适用, 以实现高精度, 多字体的汉字模的显示。

3. 从显示屏上读取

利用 Turbo Pascal 提供的点, 线, 圆, 弧, 矩, 饼等过程在显示屏上作图; 利用图形扫描器直接对各种图像进行扫描后输入计算机在显示屏上显示。

读取方法有三种:

① 软中断读取

利用 PC 机的 BIOS, 通过对寄存器的设置, 进行软中断, 可实现屏幕控制, 参见 DP 过程。

Color 的返回值是在 x, y 坐标上的像素值。因此每行逐个读取像素值, 每 8 个二进制像素值转换成一个字节, 最终组成一个图形数组。

② 从显示器内存地址读取

因为显示屏上的像素与内存地址中的单元一一对应, 当截取一个矩形窗口, 坐标左上为 x_1y_1 , 右下为 x_2y_2 , 对应内存地址计算如下:

1) 由 y 计算行地址 C

$C = y \bmod 2$, $C=0$ 为偶行, 行首地址 = $B8000H + 80 \cdot (y \div 2)$; $C=1$ 为奇行, 行首地址 = $BA000H + 80 \cdot (y \div 2)$ 。

2) 由 x_1 计算每行列首偏移地址 B , 由 x_2 计算列末偏移地址 E

$B_1 = x_1 \div 8$, $B_2 = x_1 \bmod 8$; $E_1 = x_2 \div 8$; $E_2 = x_2 \bmod 8$ 。

从每行的行首地址 $C+B_1$ 字节的第 B_2+1 位开始, 在该行首地址 $C+E_1$ 字节的第 E_2+1 位终止。

由于字节的不完整截取, 内存地址字节值应左移 B_2 个位和下一个地址的字节值右移 $8-B_2$ 个位进行或 (or) 操作, 组成一个新的值作为图形数组中对应元素的值。

对于每行末字节, 通过右移和左移 $7-E_2$ 个位, 使低 $7-E_2$ 个位置 0, 然后按上方法计算。

3) 通过两个 for 循环, 由 y_1 到 y_2 , $x_1 \div 8$ 到 $x_2 \div 8$ 读取图形数组值。过程略。

③ 调用 Get Image 过程

GetImage 过程直接读取指定窗口的位图, 并传送到某一缓冲区 BitMap 中。BitMap 是无类型参数, 当以字节类型计算, BitMap 大小 = $[(宽+7) \div 8] \cdot 高 + 4$, 其结构与本文描述的图形数组结构一样。显然直接调用 GetImage 比上述②的方法更方便、迅速。

在用字 word 类型作 BitMap 数组元素时, 由于先读写 word 的低八位, 后读写高八位, 因此在图形变换时应注意高低八位的次序。

三、图形数组的显示

与二节 3 段类同, 写与读的顺序相反, 有软中断方法, 写入显示器内存地址法和调用 Put Image 过程方法等, 参考上述介绍。PutIma-

ge的第4个参数BitBlt可对Bitmap进行与、或、非和异或的变换,为作图带来方便。有关逻辑运算和应用见四节1段。

四、图形数组变换

1. 逻辑运算

对图形数组中某部分元素进行逻辑运算变换,可有选择性地执行图形显示方式,使画面丰富多彩。下面以某个字节的二进制举例。

① 与and操作

结果是二个图形共同亮的像素为亮。适合用来屏蔽高位或低位显示,a为屏蔽高位例。

```
a
  1 0 1 1 0 1 0 1
与 0 0 0 0 1 1 1 1
-----
  0 0 0 0 0 1 0 1
```

② 或or操作

结果是二个图形重叠。例b。

```
b
  1 0 1 1 0 1 0 1
与 1 1 0 0 1 0 0 0
-----
  1 1 1 1 1 1 0 1
```

③ 非not操作

结果是图形反转。例c。

```
c
  1 0 1 1 0 1 0 1
非 0 1 0 0 1 0 1 0
-----
  0 1 0 0 1 0 1 0
```

④ 异或xor操作

图形自身异或后,图形变为暗空,GetImage与PutImage同一窗口,将使此窗口变暗空,例d。与全1异或相当于非操作,与全0异或结果不变。

```
d
  1 0 1 1 0 1 0 1
异或 1 0 1 1 0 1 0 1
-----
  0 0 0 0 0 0 0 0
```

2. 放大

简单的放大是将一个像素按放大倍数 $m \times n$ 变成一个 $m \times n$ 个像素组成的大点。参见TwoBit过程,先横向放大2倍,纵向放大只要将上行数值赋于下行即可。此过程中采用word类型,注意高八位与低八位的交换。

五、大型图形数组

一帧图象有 200×640 个像素,但在实际应用中需要更大尺寸的图形,以实现图形的上下、左右滚动。由于图形数组结构简单,规范,可简便地把各子图形数组镶成大图数组。对于不完整截取的子图,镶接时的计算参考二节3段②2。当需要时,可从大图数组中截取任一子图数组供显示。只要内存容量允许,图形数组可设置得相当大。如利用硬盘存取,则图形数组可极其大。

六、图形的硬拷贝输出

点阵式打印机(设为24针)是以列向24个像素从左向右进行打印;屏幕显示则逐行以横向8个像素为一个字节从左向右进行显示,因此需纵横变换。以打印机一列24个像素为例,步骤如下:

① 调用DP过程,读取xy坐标像素值。

② y从 y_1 增至 $y_1 + 24$,每8位组成一个字节,共3个字节。

③ x从 x_1 增至 x_2 ,扫描 $24 \times (x_2 - x_1 + 1)$ 区域内的像素。

④ $y_1 = y_1 + 24$,准备扫描下一个区域,转接②,再次扫描。

如在②和④增量过程中,y大于 y_2 ,则表示已超出窗口区域,大于 y_2 部分用0代替,执行此扫描区域后,中止打印。

七、图形数组的存取

可以在变量说明中预置Bitmap,但这样需常占一块内存空间,且缺少灵活性。当有多个且大小不定的图形需频繁的调用,宜采用指针变量,只要堆栈空间允许,对图形的调用就很自由。

为了达到快速存取,且少占用内外存空间,采用块写(BlockWrite)和块读(BlockRead),参见LoadPict和SavePict过程。

八、结束语

以上分析的图形数组的结构,介绍了一些实用的变换方法,读者可以将本文的介绍应用到肖像照片,动画图片,图文广告牌等领域。本文附上经精选的程序H.PAS供读者参考。

ETYP E A:H.PAS

```

Program H; { 图形数组演示程序 }
uses dos,crt,graph;
Type Mo=array[1..16] of word; { 16*16 字模 }
      Zi=array[1..18] of word; { 字模数组 }
Var Hanzi,Hanzicc: Zi;
      Hanzilib: Mo;
      Twohanzi: array [1..66] of word; { 2*2 倍模数组 }
      cclibfile: file of Mo; { CCLIB 字模库文件 }
      code,base: word; { 区位码,字库首地址 }
      i1,i2,n,l: word; { 区,位,区差 }
      codestr: string[4];
{----- 图形模式初始化 -----}
Procedure graphinit; { 见 Turbo Pascal 4.0 说明 }
Var graphdriver,graphmode,errorcode: integer;
Begin
  graphdriver:=detect;
  initgraph(graphdriver,graphmode,'drivers');
  errorcode:=graphresult;
  if errorcode<>grOk then halt(1);
  setgraphmode(CG4hi); { 黑白, 200*640 }
End;
{----- 查找汉字库内存首地址 -----}
Procedure Fhead;
Var i: word;
Begin
  i:=new[$0000:$0042]; { ccbios 存放汉字库首地址段值 }
  base:=new[i:$0075]; { ccbios 存放汉字库首地址单元 }
End;
{----- 由 quweima 的值计算区位地址 -----}
Procedure Codei_j(Var quweima,i,j,n: word);
Begin { 因 8-15 区空缺, 由 Case 调整 base 和 n 的值 }
  case quweima of { base=字库首地址, n=01区或16区值 }
    0101..0794: begin n:=1; end;
    1601..8794: begin n:=9; end;
    else begin n:=9; quweima:=2077; end; { 2077='错' }
  end; { case }
  i1:=quweima div 100; { 按每区 96 个字模 }
  i2:=quweima mod 100; { 计算区位码的地址 }
End;
{----- 由区位码, 计算内存地址, 读入字模 -----}
Procedure Zimo(quweima: word); { quweima 的值等于区位码 }
Var basel,i: word;
Begin
  codei_j(quweima,i1,i2,n); { 调用 codei_j 过程 }
  basel:=base+2*((i1-n)*94+i2-1); { 计算字模内存地址 }
  hanzi[i1]:=i5; { 16*16 的点阵图像 }
  hanzi[i2]:=i5;
  for i:=0 to 15 do { 由字库内存地址读 }
    hanzi[i*3]:=new[basel+i*2]; { 入数组 hanzi. }
End;
{----- 由区位码, 计算 CCLIB 文件, 读入字模 -----}
Procedure HanSeek(quweima: word); { quweima 值等于区位码 }
Var base2,i: word;
Begin
  codei_j(quweima,i1,i2,n); { 调用 codei_j 过程 }
  base2:=(i1-n)*94+i2-1; { 计算字库字模地址 }
  hanzicc[i1]:=i5; { 16*16 的点阵图像 }
  hanzicc[i2]:=i5;
  seek(cclibfile,base2); { 查找 CCLIB 字库 }
  read(cclibfile,hanzilib); { 读入数组 hanzilib }
  for i:=3 to 18 do
    hanzicc[i]:=hanzilib[i-2]; { 赋予数组 hanzicc }
End;
{----- 向显示器内存写入字模 -----}
Procedure HanzitoSCR(x,y,quweima: word);
Const basel=$B800; { 显示器内存偶行首地址 }
      base2=$B8A0; { 显示器内存奇行首地址 }
Var i,j,xy: word;
Begin
  zimo(quweima); { 调用 zimo 过程 }
  xy:=(y div 2)*80+(x div 8); { 内存地址分 2 区, 每行 }
  for i:=1 to 8 do { 80 字节, 计算 xy 的偏 }
    begin j:=2*i+1; { 恒, 直接写入屏幕内存 }
      new[basel+xy]:=hanzi[j]; { 地址, hanzi 数组 1,2 }
      new[base2+xy]:=hanzi[j+1]; { 是图像点阵大小值, 从 }
      xy:=xy+80; { j=3 开始. }
    end; { for }
End;
{----- 用 putimage 向屏幕输出汉字字模 -----}
Procedure OutHanY1(x,y,quweima,n: word);
Begin
  zimo(quweima); { 调用 zimo 过程 }
  putimage(x,y,hanzi,n);
End;
Procedure OutHanY2(x,y,quweima,n: word);
Begin
  hanseek(quweima); { 调用 hanseek 过程 }
  putimage(x,y,hanzicc,n);
End;
{----- 放大 2*2 倍 -----}
Procedure OutHan2Y(x,y,quweima,n: word);
Var a1,a2,i,j: word;
Function Twobit(b: word): word; { 位的计算函数 }
Var t,b1,b2,b3,b4,b5,b6,b7,b8: word;
Begin
  b1:=b and 1; { 和 00000001 与操作 }
  b1:=b1 or (b1 shl 1); { 对应 1 位建立 b1 的 1,2 位值 }
  b2:=b and 2; { 和 00000010 与操作 }
  b2:=(b2 shl 2) or (b2 shl 1); { 对应 2 位建立 b2 的 3,4 位值 }
  b3:=b and 4; { 和 00000100 与操作 }
  b3:=(b3 shl 3) or (b3 shl 2); { 对应 3 位建立 b3 的 5,6 位值 }
  b4:=b and 8; { 以下相同 }
  b4:=(b4 shl 4) or (b4 shl 3);
  b5:=b and 16;
  b5:=(b5 shl 5) or (b5 shl 4);
  b6:=b and 32;
  b6:=(b6 shl 6) or (b6 shl 5);
  b7:=b and 64; { b1-b8 或操作计算点值 }
  b7:=(b7 shl 7) or (b7 shl 6); { 用 word 作 putimage 的图 }
  b8:=b and 128; { 形数取值时, 低字节在左, }
  b8:=(b8 shl 8) or (b8 shl 7); { 高字节在右, 用 Swap 交换. }
  t:=b1 or b2 or b3 or b4 or b5 or b6 or b7 or b8;
  Twobit:=Swap(t);
End; { function }
Begin
  zimo(quweima); { 调用 zimo 过程 }
  twohanzi[i1]:=31; { 32*32 的点阵图像 }
  twohanzi[i2]:=31;
  for i:=3 to 18 do

```

用高级语言编制过程控制软件

机电部第214研究所 阎建明

从事过程控制软件编程的工程技术人员中,偏爱用 Basic 的人居多。然而传统的 Basic, 由于运行速度慢、没有良好的结构控制语句、程序大小受64K 限制,很难编制出运行效率高、结构良好的、大型复杂的控制软件。尤其是做实时多任务处理时,几乎是一筹莫展。而一些专用的实时 Basic, 虽然能处理多任务,但除了本身不是结构化语言之外,使用范围也有一定的限制。

近年来推出的新型 Basic, 如 Turbo

Basic (还有 Quick Basic, 但笔者觉得 Turbo Basic 好用), 有一个编程极其方便的集成化程序设计环境,它增加了结构化的控制语句,程序可大于 64K, 其运行速度是解释 Basic 的 4 至 100 倍。

由于 Turbo Basic 具有许多新的特点,用这一通用高级语言,可以很方便地编制出结构良好、便于维护的过程控制软件和做多任务处理。

下面针对实时过程控制编程中的主要问题,

```
begin
j:=i-3;
a1:=hi(hanzi[i]);      { 同 Swap 说明, 低 }
a2:=lo(hanzi[i]);      { 字节在前, 高字节 }
twohanzi[3*4+j]:=twobit(a2); { 低, 下行与上行同 }
twohanzi[3*4+j+1]:=twobit(a1); { 低, 0=2, 1=3 的值 }
twohanzi[3*4+j+2]:=twohanzi[3*4+j];
twohanzi[3*4+j+3]:=twohanzi[3*4+j+1];
end;
putimage(x,y,twohanzi,a);
End;
{----- 软中断读写点 -----}
Procedure DP;
Var reg: registers;
x,y,co,i: word;
Procedure Point(x,y,rw:word;Var color:word);
Begin
reg.dx:=x; { 写入 Y 值 }
reg.cx:=x; { 读列 X 值 }
reg.ah:=rw; { 读写点 }
reg.al:=color; { 点值 }
intr(16,reg); { 软中断 16 }
color:=reg.al; { 返回点值 }
End;
{----- 演示 -----}
BEGIN
co:=1;
for i:=1 to 3000 do
begin { 随机数写点 }
x:=random(839);
y:=random(199);
point(x,y,12,co);
end;
END;
{----- 快速存取图形文件 -----}
Procedure Graphpic;
var pictfile: file;
p,pp: pointer;
count,size,i: word;
Procedure loadpic(filename:string);
Begin
assign(pictfile,filename);
(SI-) reset(pictfile); (SI+)
count:=filesize(pictfile);
size:=count*128;
```

```
getmem(pp,size);
(SI-) blockread(pictfile,pp,count); (SI+)
close(pictfile);
End;
Procedure savepic(x1,y1,x2,y2:word;filename:string);
Begin
size:=imageize(x1,y1,x2,y2);
count:=(size+127) div 128;
size:=count*128;
getmem(p,size);
setimage(x1,y1,x2,y2,p);
assign(pictfile,filename);
(SI-) rewrite(pictfile); (SI+)
(SI-) blockwrite(pictfile,p,count); (SI+)
close(pictfile);
End;
Begin {----- 演示 -----}
savepic(240,100,310,190,'graph1.pic');
loadpic('graph1.pic');
for i:=1 to 10 do
begin
putimage(1400,10,pp,2);
delay(1000);
end;
End;
{----- 主程序 -----}
BEGIN { main program }
assign(cclibfile,'C:\cclib'); { 打开 CCLIB 字库库 }
(SI-) reset(cclibfile); (SI+)
graphinit; { 图形模式初始化 }
fread; { 读表头地址程序 }
hanziToSCR(250,100,2638); { 显示程序 1 帧 }
OutHanXY1(270,100,4367,0); { 显示程序 2 帧 }
OutHanXY2(290,100,2780,0); { 显示程序 3 帧 }
OutHan2XY(240,120,4528,0); { 显示程序 4 帧 }
OutHan2XY(275,120,4848,0); { 显示程序 4 帧 }
DP;
graphpic;
readin;
closegraph;
textmode(3);
close(cclibfile);
END.
```

介绍笔者在实践中总结的一些编程方法。

一、实时中断编程

实时过程控制编程,必然要遇到实时中断问题。Basic本身具有实时中断功能。但由于传统Basic运行速度慢,实际意义不是很大。Turbo Basic运行速度快,因而使得这些功能具有实用性。

实时中断涉及到实时钟,一般使用的是外部硬时钟,如8253。这里介绍软时钟。

(1) 利用ON TIMER(n) GOSUB(子程序名)语句

语句中的n,表示等待多少秒的整型表达式,值为1~86400秒。用这个语句,可以设计出精巧的软实时钟,见下例。

```
例 1.  clock% = 0
        on timer(1) gosub clocks
        timer on
        do
            if (某键盘中断发生) then exit loop
            (主模块)
        loop
    end
    clocks:
        incr clock%
        if clock% = (温度采样周期) then
            call Temp(口地址)
        end if
    return
```

(2) 利用call interrupt语句

Turbo Basic 有个中断调用语句,形式为: call interrupt (n). n为整型表达式,为调用的中断号,取值范围0~255。它既可以调用DOS和BIOS中断处理,也可用于用户自己定义的软中断处理。

中断10H到1FH是BIOS调用。BIOS的1CH号中断,为计时信号中断,由8号中断服务程序TIMER-INT产生。但是BIOS使1CH号中断向量指向一条IRET指令,地址在DUMMY-RETURN (F000: FF53)。因此1CH号中断什么也不做。如果我们改变这个指针,用1CH号中断(用call interrupt &H1C),来检查时间计数器的内容,在达到某时刻时,执行你所要的实时操作,这就实现了实时中断。这里就不详细举例了。

二、“进程”的编制

实时过程控制中,往往通过“进程”,来完成任务的调度和运行,进而实现多任务处理。传统Basic很难完成这一功能。而用Turbo Basic中的“过程”,可以很方便的实现编程。

用过程表示任务,通过过程间的调用和参数传递,即可完成各种“进程”,实现多任务处理。

```
例 2.  REM *** 主程序 ***
        .....
        if (某任务运行标志为1) and (clock%(n) = "时标") then
            call queue (任务号,"时标",....)
        end if
        .....
    end
    *** 过程 queue ***
    sub queue(参数1,参数2,...)
        变量说明
        .....
        select case(某参数)
            case <某值>
                call swithing(口地址,动作参数)
                call pressure(口地址,动作参数)
            case <某值>
                .....
        end select
    end sub
```

过程queue通过任务号、“时标”等参数传递,选择相应的任务,然后调用过程swithing(开关量输出)、pressure(压力调节),并传递口地址、动作参数等参数,来完成对某被控对象的开关量控制、压力控制等“进程”。

三、数据采集程序和控制算法程序的编制

用传统Basic,往往是把采集、算法做为子程序,由gosub语句来调用。对于子程序多的程序,往往不得不使用goto语句,来控制程序流向,使得程序难维护。

Turbo Basic提供了多行函数。因此可以将采集、控制算法等,设计成一个个函数,通过参数传递来调用。和过程一样,函数中除了全局变量外,对外界是独立的,仅通过参数传递来与外部发生联系。这样除了结构良好之外,而且安全、可靠,便于维护。

```
例 3.  sub temp(参数)
        变量说明
        .....
        temp1 = FNTempc(口地址 1)
        temp2 = FNTempc(口地址 2)
```


利用软件黑盒子对PC机文本文件加密的原理与方法

西南交通大学 于功第 路 枝

摘要 本文从国外普遍采用的硬件黑盒子加密方法出发,引出了软件黑盒子的概念,并叙述了利用软件黑盒子对PC机文本文件加密的原理、方法和优点。最后给出了建立软件黑盒子文件、加密和解密的实用程序清单和通过实例。

一、引言

文本文件即标准ASCII码文件。用高级语言建立的FORTRAN、BASIC、PASCAL、LISP和C语言等文件都是文本文件。随着微机的普遍应用和软件专利的逐步建立,许多程序设计者希望自己在微机中的文本文件不被别人窃取。为此,如何对这类文本文件方便、可靠的加

密、解密是一个很现实的问题。

目前对这类文件软件加密的方法较多,但也存在着一些问题:

(1) 有些加密技术方法复杂,要求有较高的专业知识和设备条件。

(2) 有些加密方法简单,但容易破译。

例如,编制一个高级BASIC语言程序,将

```
.....
end sub
' *** 温度采集函数 ***
DEF FNTempc(参数)
  变量说明
  启动 A/D
  采集,处理
  .....
END DEF
```

四、运行速度有特殊要求的编程

这种情况只有使用汇编。用传统Basic调用汇编语言程序,不是很方便。

Turbo Basic 为汇编语言接口,提供了三种形式的调用语句。

- (1) 绝对调用语句call absolute.
- (2) 中断调用语句call interrupt.
- (3) 行间调用语句

```
call <程序名> [(参数表)]
sub <程序名>
  $inline <byte list>
  $inline "COM File Name"
end sub
```

第一种方法,子程序在被调用前,用BLOAD语句装入,或本身是程序数据空间中的一部分。第二种方法,用来调用DOS和BIOS中断处理。第三种方法,是Turbo Basic中最有力的汇编语言程序设计方式,所以通常用该方式调用汇编子程序。

例4.将一批变量,譬如510个,同时置为零

或置为某个初始常数。如果用循环语句for i=1 to 510: a(i)=0: next i, 一个个送入数组,则太慢了。这时就要用汇编。

```
new 主程序
dim a%(510)
call fillasm(a%<0>)
end
' *** 汇编子程序 ***
sub fillasm inline
  $inline &h55
  $inline &h89,&h06
  $inline &hc4,&h7e,&h06
  $inline &hb8,&h64,&h00
  $inline &hb9,&h1fe,&h00
  $inline &hfc
  $inline &hf3
  $inline &hab
  $inline &h5d
end sub
```

```
' push BP
' mov SP,SP
' les DI,[BP + 6]
' mov AX,&4E
' mov CX,1FEH
' cld
' rep
' stosw
' pop BP
```

参 考 文 献

- [1] 阎建明:“在过程控制中运用ON TIMER 语句”,《微型机与应用》2, 1989.
- [2] 阎建明:“实时多任务控制软件设计的一种实用方法”,《1989年全国计算机应用研究学术交流会议论文集》,《计算机应用研究》杂志社.
- [3] 《软件工程手册》,General Electric Compang著,科海培训中心1987年8月编,

文本文件中字符的ASCII码按等值进行位移变换,以实现加密。由于是等值,即文件中的所有字符的ASCII码加1或加X进行位移($0 < X < 255$),故破译很容易,只要编一个对应程序从0~255中取值进行复位变换,最多重复执行255次就能破译加密文件。

作者在实践的基础上,从对软件加密的硬件黑盒子的方法中得到启发,研制了软件黑盒子的加密方法。通过大量应用表明,此法简单、可靠、方便和实用。现介绍如下:

二、软件黑盒子加密的原理和方法

目前,随着电子技术的发展,国外普遍采用了黑盒子(blackbox)这一硬件对软件程序加密。其方法是:利用逻辑门阵列芯片(如GAL-16V8-25LP或其它阵列)产生硬件黑盒子。然后利用它任意可编程,固定下一组序列地址,序列地址中装入唯一一组数据。用户利用此盒对程序进行加密,别人想解密文件读取这组地址必须按原加密序列进行,如果颠倒次序或越序都不会得到正确数据。假设黑盒子内有十个地址,那么在不预知序列数据的情况下,想得到唯一加密序列的可能性是亿万分之一。可见硬件黑盒子加密的方法是非常先进和可靠的。由于这种方法需增购硬件设备,给其广泛运用带来了局限性。为了解决这个问题,笔者提出了软件黑盒子的加密原理和方法:预先设置一个数据文件,内存用户设置的任意一组数据序列(相当于一组密码)代替硬件黑盒子。笔者把这个数据文件叫做软件黑盒子。然后用高级BASIC语言编制一个文件,其功能是对任何文本文件中所有字符的ASCII码按软件黑盒子中的数据序列一组一组对应循环反复进行位移变换,直到文件中所有字符变换完为止。变换后此文件就会面目皆非、极难辨识,从而达到加密的目的。解密时,则按软件黑盒子的数据序列一组一组对应循环反复进行复位变换即可。

原理描述如下:

设:任意文本文件的字符集合为A.

$$A = \{a_1, a_2, a_3, \dots, a_n\}$$

软件黑盒子文件的数据序列集合为B.

$$B = \{b_1, b_2, b_3, \dots, b_m\}$$

$$(m \leq n) \quad (0 \leq b_i \leq 9) \quad (1 \leq i \leq m)$$

位移变换后加密的文本文件的字符集合为C.

$$C = \{c_1, c_2, c_3, \dots, c_n\}$$

又设: f 为求任意字符ASCII码的函数.

g 为求任意ASCII码对应字符的函数.

则得到:集合A、B、C间的关系为:

$$\text{加密关系: } c_1 = g(f(a_1) + b_1)$$

$$c_2 = g(f(a_2) + b_2)$$

$$\vdots$$

$$c_m = g(f(a_m) + b_m)$$

$$c_{m+1} = g(f(a_{m+1}) + b_1)$$

$$\vdots$$

$$(1 \leq k \leq m) \quad c_n = g(f(a_n) + b_k)$$

$$\text{解密关系: } a_1 = g(f(c_1) - b_1)$$

$$a_2 = g(f(c_2) - b_2)$$

$$\vdots$$

$$a_m = g(f(c_m) - b_m)$$

$$a_{m+1} = g(f(c_{m+1}) - b_1)$$

$$\vdots$$

$$a_n = g(f(c_n) - b_k)$$

从上述原理可知,由于整个文本文件中所有字符不是按一个固定增值位移改变,而是按 m 个数据序列一组一组的对应位移变换。假设 $m=10$,如果破译者在不知这组数据序列的情况下,想将已加密文件所有字符复位破译的可能性仅有亿万分之一。如果软件黑盒子中的数据序列增长到 $m=20$ 位以上,则想破译文件几乎是不可能的。

具体作法:

(1) 预建立一个软件黑盒子文件,内存一组密码数据序列,例如1, 3, 2, 4, 8, 9, 5, 6, 7, 1等。

(2) 运行加密程序调用软件黑盒子文件中的数据序列,对任意文本文件加密。

(3) 加密后,用户必须唯一保存软件黑盒子文件(类似保存保险柜的密码和钥匙)。

(4) 解密时,装入软件黑盒子文件,运行解密程序对文件进行解密。

此法优点:

(1) 软件黑盒子文件容易建立,不存在成本或其它硬件问题,易推广运用。

(2) 软件黑盒子文件中的数据序列可任选长度, 任意编码排序, 故破译困难, 加密可靠性高。

(3) 加密、解密程序简单、易编制, 操作方便, 仅输入一些数据即可建立软件黑盒子文件, 对文本文件加密或解密。

三、程序清单及说明

1. 说明

(1) 此程序可完成建立软件黑盒子文件、加密和解密文本文件的工作 (已成为实用软件工具)。

(2) 选择1, 输入数据序列 (必须是数字), 建立软件黑盒子文件 (数据序列一个数字输入一次)。

(3) 选择2或3输入加密或要解密的文本文件名, 实现文件的加密或解密。

注意: 1) 解密时, 不进行选择1的操作。否则会破坏软件黑盒子中的内容, 造成文件无法解密。故每个文件加密后的黑盒子文件必须有副本保存。

2) 加密后软件黑盒子文件必须拷到软盘保存, 同时删去C区中的原本, 以防泄密。

2. 程序清单

注: 此程序和实例均在PC/XT和386机上通过。

```

1  rem 对任何 ASCII 码正文文件加密 / 解密软件工具 于功第编
10  input "产生软件黑盒文件: 1,加密: 2,解密: 3,0: 退出?"; c
20  if c <> 1 then 50
30  gosub 100
40  goto 10
50  if c <> 2 then 80
60  gosub 200
70  goto 10
80  if c <> 3 then 95
85  gosub 300
90  goto 10
95  system
98  rem 产生软件黑盒文件
100  input "请输入要建立的黑盒子文件名"; f$
110  open g$ for output as #2
120  input "请一个一个输入密码序列,0 结束"; x
130  if x = 0 then 160
140  write #2,x
150  goto 120
160  close #2
170  return
190  rem 加密文件
200  input "请输入要加密的文件名"; f$
205  input "请输入软件黑盒子文件名"; g$
210  open "r",#1,f$,1
220  field #1,1 as a$
230  bot = lof(1)
240  open g$ for input as #2
250  for i = 1 to bot

```

```

255  if cof(2) then 285
260  input #2,k
265  get #1,i
270  lset a$ = chr$(asc(a$)-k)
275  put #1,i
280  goto 290
285  close #2
287  open g$ for input as #2
290  next
291  close #1
292  close #2
294  return
295  rem 解密文件
300  input "请输入要解密的文件名"; f$
305  input "请输入软件黑盒子文件名"; g$
310  open "r",#1,f$,1
320  field #1,1 as a$
330  bot = lof(1)
340  open g$ for input as #2
350  for i = 1 to bot
355  if cof(2) then 385
360  input #2,k
365  get #1,i
370  lset a$ = chr$(asc(a$)-k)
375  put #1,i
380  goto 387
385  close #2
386  open g$ for input as #2
387  next
388  close #1
389  close #2
390  return

```

四、实例

此实例给 YED · BAS 文件加密和解密。

```

c> basic yj.bas
产生软件黑盒文件:1,加密:2,解密:3,0:退出?? 1
请输入要建立的黑盒子文件名? luzhi.dat
请一个一个输入密码序列,0 结束? 1
请一个一个输入密码序列,0 结束? 2
请一个一个输入密码序列,0 结束? 3
请一个一个输入密码序列,0 结束? 4
请一个一个输入密码序列,0 结束? 5
请一个一个输入密码序列,0 结束? 6
请一个一个输入密码序列,0 结束? 7
请一个一个输入密码序列,0 结束? 8
请一个一个输入密码序列,0 结束? 9
请一个一个输入密码序列,0 结束? 0
产生软件黑盒文件:1,加密:2,解密:3,0:退出?? 2
请输入要加密的文件名? ygd.bas
请输入软件黑盒子文件名? luzhi.dat
产生软件黑盒文件:1,加密:2,解密:3,0:退出?? 0

```

```
c> type ygd.bas
```

```
22
```

```

) "gkoisglm+;g76wyco"/%V2*95f%.5
97oifng(73):at"d(8bpv#A%rvn11*
8:-zrtf"csy=9gqu nC8(olrnx%y{my 3
:5nm} 3 / m = = wskwi%g+71p5% C'u:
"4>7weyv>6kuotg667rjvh r7:11v}xzlu
!"#1

```

```
c>
```

硬盘加锁技术及实现方法

中国人民解放军五九一九〇部队自动化站 王晓武

摘要 本文结合分析PC-DOS的硬盘启动过程DOS分区引导记录块, 介绍一种用软件硬盘加锁的实用方法, 帮助广大PC机用户保护硬盘程序和数据免遭意外破坏。

在我们的实际工作中, 常常有用户提出这样的要求, 即: 能不能对硬盘进行加锁, 也就是说, 一个单位的计算机只限于有关操作人员才能启动硬盘上的操作系统, 运行程序, 而其他无关人员均无法使系统启动运行, 防止数据意外丢失或被破坏。对于这个问题笔者经过对PC DOS的分析与改进, 探讨了一种简便而有效的方法。

一、硬盘启动过程分析

一台IBM PC系列微机系统, 一般都装有一个硬盘适配器和驱动器, 当硬盘已格式化并驻留了PC DOS系统后, 每次冷或热启动时, 只要软盘A驱动器上没有插入PC-DOS系统盘, 系统就会自动从硬盘中读出并引导PC DOS系统, 进入系统开工状态。

从硬盘自举引导DOS的过程有以下步骤:

- (1) 上电自诊断程序对硬盘进行测试;
- (2) 调用BIOS中的INT 19H 中断服务程

序, 此中断负责读出硬盘物理0磁头0柱面1扇区上的主引导记录, 并将其装入内存000:7C00H处。

(3) 运行主引导记录检查硬盘分区表的有效性, 然后将可自举的DOS分区(硬盘逻辑0头0柱1扇区)的引导记录读入内存0000:7C00H(复盖原主引导程序)处, 并将控制转交给它, 由它负责启动DOS, 进入系统开工状态。

从上述硬盘的自举引导过程分析, 整个系统启动过程中关键的地方是读取和运行DOS分区的引导记录程序。当DOS分区的引导记录有效并运行成功, PC DOS系统就能启动。试想, 如果我们在DOS分区引导记录块中, 增加(插入)一段密码判断程序, 用ROM BIOS的INT 16H中断读取键盘密码, 若密码相符则正常运行引导记录并加载DOS系统。否则, 使机器死循环, 这样就能达到对硬盘加锁的目的。

五、结束语:

此法是笔者在应用需要的基础上得出的, 已进行过许多成功的运用。从原理上讲此法可移植用于VAX和其它系列机的文本程序的加密, 仅需把工具软件中有关语句按其BASIC版本规定作些修改即可。

微机测试婴幼儿智能状况程序问世

江苏省中医院和南京市儿童心理卫生中心共同研制成功微电脑婴幼儿智能状况测试程序, 该程序使用IBM机, 由检测评估、查询、统计、打印、修改5部分组成, 可测试不同年龄婴幼儿智能状况, 编程依据是美国一位专家的智能训练理论和方法, 内容是可能影响婴幼儿发育的一些因素, 包括家庭情况、社会环境、教育状况等300多个项目, 通过人机对话的方式, 即能检测评估出婴幼儿成熟的年龄和发育智商, 临床诊断符合率达99%, 该项目已举办全国学习班推广。(李相彬)

```
c> basic yj.bas
产生软件黑盒文件:1,加密:2,解密:3,0:退出 ?? 3
请输入要解密的文件名? ygd.bas
请输入软件黑盒子文件名? luzhi.dat
ok
system
c> type ygd.bas
1   m1 = 0
10  input "filename";f$
20  open "R",#1,f$,2
30  field #1,2 as a$
40  bot = lof(1)
45  write bot
50  for i = 1 to bot step 2
60  get #1,i
65  write a$
70  m1 = m1 + 1
80  next i
90  close
100 write m1
110 system
c>
```

二、DOS分区引导记录分析

DOS分区引导记录块一般由以下几个部分组成:

- (1) 磁盘I/O参数表
- (2) 软盘基本参数表
- (3) 引导程序代码
- (4) 错误提示数据区

引导程序的执行过程可简述如下:

首先将引导程序装入内存0000:7C00H处, 然后从此地址开始执行, 初始化中断向量1EH, 将磁盘I/O基本参数装入BIOS数据区, 然后加载DOS的两个系统隐含文件 (IBMBIO.COM 和IBMDOS.COM) 最后把控制权交给 IBM-BIO.COM, 启动PC-DOS。在上述过程中如遭到操作失败, 即显示出错信息, 正常则进入系统开工状态。

三、硬盘加锁方法

通过对PC-DOS从硬盘启动的过程分析, 以及DOS分区引导记录块的分析, 我们可以找到硬盘加锁的有效方法。

通过用DEBUG程序读取硬盘逻辑0扇区, 我们知道引导程序的第一条指令为JMP 012E, 我们将012E改为02A2, 在CS:02A2单元开始处有一段错误提示数据可复盖, 在这个地方, 我们可以插入一段用ROM BIOS INT16H中断来判断键盘密码的程序, 如果用户输入的密码正确再跳转012E处开始引导系统, 否则使系统处于死循环状态, 具体操作步骤如下:

```
C>DEBUG          ; 运行调试程序
-L CS:100 2, 0, 1 ; 读硬盘 DOS 引导扇区
-E 02A1 00        ; 改变提示信息长度
-A CS:100          ; 修改引导记录第一条指令
XXXX:1000 JMP 02A2 ; 转新增密码键判断程序
XXXX:0103
-A CS:02A2        ; 修改数据提示区
XXXX:02A2 MOV AH, 02 ; 调用16H中断2号功能
XXXX:02A4 INT 16
XXXX:02A6 CMP AL, 03 ; 是否按下左、右SHIFT键
```

```
XXXX:02A8 JNZ不是, 转CS:02A2
死循环
XXXX:02AA JMP 012E ; 是, 转原引导记录块运行并启动系统
XXXX:02AD NOP
XXXX:02AE
-W CS:100 2, 0, 1 ; 写修改后的硬盘引导记录
```

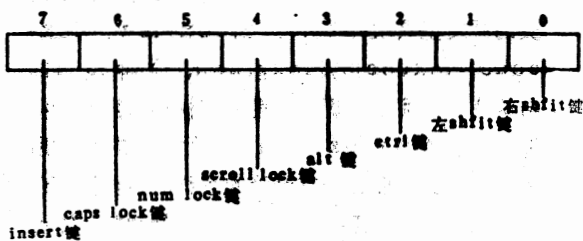
-Q
C>

注: 上述XXXX表示段地址, 不同系统地址不同。此方法是在DOS 2.1版本下实现的, 对于其它的DOS版本, 只是程序代码的转移地址有所不同, 而方法步骤相同。

从上述操作过程及插入的程序可知, 此处我们利用了INT 16H中的2号功能, 即读取特殊键盘状态, 此处AL=03表示为左、右Shift键同时按下, 也就是说, 只有当系统复位(冷、热启动)时, 用户同时按下左、右Shift键, 硬盘上的PC-DOS才能启动, 这种特殊的组合键入就是硬盘加锁的钥匙, 不知道的人就无法启动硬盘。当然有很多种特殊键的组合方式, 可根据用户的需要设置(用户也可利用INT 16H的4号功能设置ASCII密码加锁硬盘)。下面是INT 16H的2号功能的入、出口参数, 用户从中可知详情:

入口参数: AH=2

出口参数: AL=特殊键状态(每个二进制位含意如下)



四、结束语

用户按照上述方法即可实现IBM PC系列微机的硬盘加锁功能。但是常言道: “锁君子不锁小人”, 一旦内行人用与硬盘上相应的DOS系统盘从软盘启动后, 仍然可以进入硬盘运行, 但是与微机开发和操作无关的人, 一般不会有相应的DOS系统盘, 因此, 我们说本文所述方法是避免无关人员损坏硬盘及数据的有效方法。读者不妨一试。

崭新的FOXBASE + 2.10和它的开发工具

云南大学计算中心

杨剑波

云南人民出版社

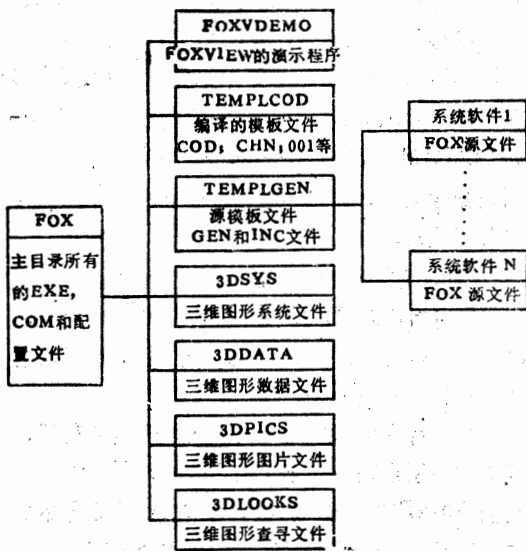
西捷

摘要 本文就fox软件公司1988年推出的foxbase + 2.10作了较全面的介绍, 分析了新版foxbase的新特点和发展, 并且重点介绍了[foxbase 开发工具中的 foxview 和 foxcode, 希望为 foxbase 的广泛应用“抛砖引玉”。

自1987年6月美国sco公司发行 fox 公司的 foxbase+后, 全世界都为它的高速和与dbase的完全兼容所吸引。而在为维护dbase的“全球霸主”地位而研制已久的dbase III 和dbase IV 相继推出不久, fox软件公司迅速推出了崭新的 foxbase+2.10 版本和与之配套的软件包, 又一次为众多的用户所垂青。

新的foxbase+在继承前任所有优点的基础上, 又有了新的飞跃。它的功能介于 dbase III plus 和 dbase IV 之间, 它仍然与工业标准的 dbase III plus 完全兼容。当然, 新版foxbase+的速度又有了进步, 它仍然是目前最快的dbase IV plus 兼容系统。特别值得一提的是 2.10 版的 foxbase+ 为用户提供了一个具有 dbase IV 控制中心大多数功能的非编程用户接口 fox-control 和一组通过 foxcontrol 使用的近乎完美的开发工具软件包, 包括 foxview、fox-code、foxdoc 以及 foxgraph。这种良好的结构使得任何未受训练的用户同样可以随心所欲地使用菜单通过 foxcontrol 进行对数据库的所有操作维护。而那些有经验的系统程序员原可以轻松地利用开发工具完成他们的“梦想”。需要指出的是, 他们将为此节省80%左右的时间, 新的 foxbase+ 将使他们从繁琐的屏幕设计、常规的数据库操作及格式打印等工作中解放出来, 更加自如地发挥自己的才能。

由于 foxbase+2.10 系统比较复杂, 加上各类开发工具的频繁调用, 因此, 一个标准实用的目录结构的建立将有助于你更好地工作。这个标准的目录设置(见图一)将 foxbase 系统文件一起严格地保持在已知的位置, 所以 foxbase+



(图一) FOXBASE + 2.10 装配标准目录

能容易地找到它们。当然, 熟悉了 foxbase+2.10 的系统 and 它的开发工具后, 你也可以自己建立一些新的目录以容纳你的程序。

新版的 foxbase+ 在保留了 dbase III plus 所有文件类型的同时, 也增加了一些新的文件类型, 它们大多用于 foxbase+ 的各类开发工具, 这些新的文件类型是:

• 编译后的模板文件 (.cod 文件); • 源模板文件 (.gen 文件); • 屏幕表格文件 (.fv 文件); • 模板子模板 (.inc 文件); • 链接模板 (.chn 文件); • 文档文件 (.doc 文件); • 浏览文件 (.vne 文件)

foxbase+2.10 有较大的改进和提高, 与先前的版本相比较, 更趋于完美, 如增加了新的 sys (2000), sys (2001) 和 sys (2003) 函数,

分别用于从磁盘恢复文件名、控制光标开/关和测定当前工作目录。

新的 **soundex** 函数返回一个字符表达式的语音表示, 以使你确定两个字是否语音类似 (如果它们声音相象)。

在一个索引数据库中的不成功 **seek** 后, 用户可以采用 **recno(0)** 函数“软寻找” (**softseek**) 逻辑的优点去确定记录号返回。该函数返回关于这个索引的先于 **seek** 表达式的记录号。

当然, 新的 **foxbase+** 还有许多特点, 例如其开发环境已转换到 **Microsoft+** 5.0 等等。由于这方面文章不少, 这里就不一一列举了。

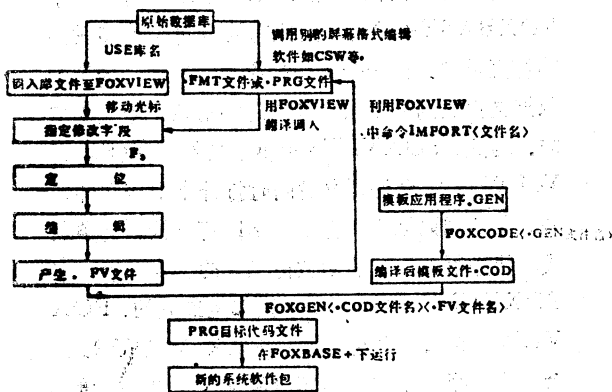
谈到 **foxbase+** 2.10, 就不能不提到它强有力的一系列开发工具。它们包括 **foxview**、**foxcode**、**foxdoc** 和 **foxgraph**。当然还有众多的标准。cod 模块程序和那个为 **foxview** 及 **foxcode** 设计的专用模块系统 **ProtempTH**。正是这些令人眼花缭乱的开发工具, 使新版的 **foxbase+** 得以发挥无比的威力。

foxview 与 **foxcode** 是一个良好的结合。用户可以通过控制中心 **foxcontral** 激发它们, 也可以在 **Dos** 下分别驱动它们。一旦用户决定了工作并开始着手编码后, **foxview** 与 **foxcode** 的强大功能便开始显露。它几乎可以使一个熟练的程序员节省 80% 左右的时间。而且, 令人吃惊的是, 利用 **foxcontral** 的帮助, 一个丝毫不懂数据库复杂操作命令的人也能利用 **foxview** 和 **foxcode** 编出完整实用的系统软件包, 简直是得心应手!

图二所列举的, 正是一个常规系统软件包的生成流程, 这个软件包包括了常规的数据库录入、修改、查寻、观察输出打印等功能。在某种程度上说, 它已经是一个很完善的软件包了。而这一切要你动脑筋的只是建立你自己的数据库和你自己的屏幕格式, 其余的部分均由机器完成!

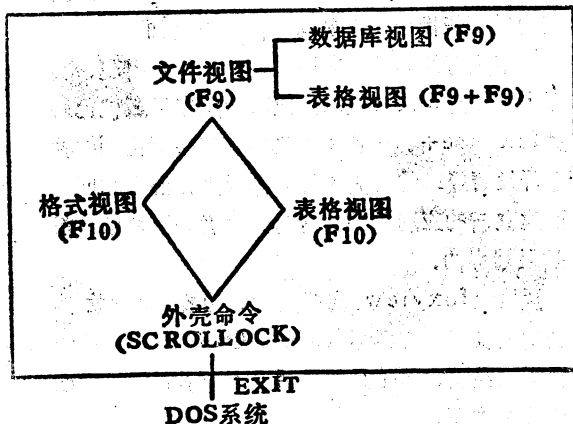
既然 **foxview** 与 **foxcode** 有如此大的魅力, 那么就让我们再回过头来看看。

先说 **Foxview**。它是一个高级的交互式环境, 让你能够控制显示, 以用来在 **foxbase+** 中输入和存取数据。它的作用有些象一个解释器



(图二) 常规系统软件包的生成流程图

——将你看得到的格式翻译成一个钻石样子 (如图3)。中央是菜单系统, 四个基本模块——**forms view**、**table view**、**files view** 以及 **shell**, 则可形成四个顶点, 它们可相互存取。图中括号内是该部分的激发键。



(图三) FOXVIEW钻石菜单屏

当你进入 **foxview** 后, 在 **shell** 命令环境下打开你欲操作的数据库, 然后在格式视图中调用各种盒子、方框和调色板上的色彩, 利用光标移动键设计你自己的屏幕输入格式。当然, 你随时可以进入表格视图, 浏览一下字段客体的所有特性、位置、功能、颜色、大小等等的屏幕图象基本特性, 完成了这些在字段是的设计操作后, 你可以用 **save** 命令将它们生成 **foxbase** 的目标代码。

对于每一位程序员来说,也许没有一个FOXBASE+的特征能比它的多文件和多关系功能更重要。在一个大型的应用程序中,繁杂的数据文件之间的联结与频繁的调用总是令人头疼的。为此,FOXVIEW提供了文件视图(FILE VIEW)。它允许开发者同时打开十个数据库文件,并且在不同数据库文件中的字段可以根据其类别或功能进行分组,然后放在相应的工作区中,在FOXVIEW中,每个字段都是独立的,FOXVIEW的独特之处在于它并未装入实际的记录,而是将一个字段的环境信息(标题、位置、宽度、颜色、类型)装入,这样,作为一个独立的实体,它可以利用文件视图中的两个子视图——数据库视图(DATABASEVIEW)和别名视图(ALIASVIEW)将打开的十个数据库中的任意字段取出进行重新组合、交换,形成新的数据库文件和报表文件,而这一系列工作,FOXVIEW会自动书写代码,你所要做的是决定要做那些操作,真是简单得难以复加。

事先写好的模板应用程序已经开始出现。实际上,在开始2.10版的修改时,共享模板就已装进foxbase+。这使得终端用户可以由别的独立开发者提供的事先写好的模板进行混合和匹配而建立新的应用程序。这对开发者和用户来说都是很精彩的。

为使foxview发挥更大的功效,fox软件公司提供了一套由MARK DOUGLAS设计的模板工具箱。该工具箱包括了十个不同功能标准的模板程序。

- **holdfile.cod**: 利用该模块用户可以生成一个程序,用于重新建立一个应用软件包中的所有文件或其中某一个文件。

- **foxindex.cod**: 提供foxview用户一种快速、简单、易学的方法,生成以单条件或复合条件重新索引一个或多个文件的程序。

- **foxmain.cod**: 提供给用户一个类似foxbase+演示程序appsi的下拉式菜单。而用户可不必生成一个新的foxview表格文件。

- **genenv.cod**: 提供用户一个容易观察的环境。

- **gensel.cod**: 提供用户改变选择设置命

令。

- **genstart.cod**: 提供用户观察当前被选文件的状态。

- **litebar.cod**: 提供foxview用户一个快速、简便、易学的设计直方图的菜单项。

- **helpmain.cod**: 提供一个详细的辅助屏幕,它是为appsi.cod的主模板而设计的。

- **polldown.cod**: 设计一个下拉式菜单。

- **popmenu.cod**: 设计一个上弹式菜单。

这些模板应用程序与foxview相结合,能够生成更具威力的foxbase目标代码。总之,foxview好比一块“画板”,用户可以在上面绘制自己的屏幕格式。而其它的事就交给foxbase+去做。

下面谈谈foxcode,foxcode是帮助书写foxbase+程序的冗长乏味过程自动化的唯一工具。它利用“填充空格”这样一个复杂的模板语言处理过程,foxcode包括三个主要成分。

1. **foxcode语言**。它是命令、符号和语法规则的列表,模板将根据这些语法规则来构造。模板语言本身由一些熟悉的命令构成,这些命令被插进你的foxbase+应用程序中。

2. **模板文本编译器(foxcode)**,它对foxbase+关键字做标记,并且压缩foxcode+命令。

3. **应用程序产生器(foxgen)**。它从被编译过的模板和用foxview建立起来的表格中提取信息,构造模板所规定的foxbase+程序。

所谓“模板”,是描述了你想要的foxbase+程序结构的一个蓝本。它包含foxbase+的命令和foxbase+语句。其中foxcode语句由《 》括起,或由《# #》、《#pragma #》来区分。foxcode语言原型是modul-2、dbase、VI programmer语言。具体的语言规范请参考foxbase+手册。

用户可以用任意一种编辑器写自己的模板,也可用标准的编译后的模板,这样,开发一个系统的工作就变得协调而轻松了。

总之,无论是新版的foxbase+2.10,还是

dBASEIII-PLUS下的窗口操作的设计

新疆电子计算中心

周步祥 阮树驊

一、概述

窗口是为完成某种特殊用途而在屏幕上开辟的某一区域,通常这块区域是方形的,一个屏幕上可同时出现好几个窗口,用户可在各个窗口中进行工作。窗口的理论基础是比较简单的,当程序执行时,程序中的各个工作分别使用各自的窗口,当工作开始时,它所使用的窗口就被启动,而在工作完成之后,该窗口便会消失,假如某项工作在执行期间被别的工作中断,则此工作将暂时停止执行和使用其窗口,但此窗口并不消失,新的工作便启动它的窗口,它将盖在其它窗口的上面。

窗口可以使用户集中注意力,屏幕显示清楚,不容易引起混淆,为人机会话提供良好的界面。

dBASE III-PLUS 没有提供窗口操作的功能,但我们发现在编制应用软件的过程中,利用窗口操作的一些性质,会使人机界面变得十分友好,因此我们进行了窗口操作的设计的尝试。

二、设计思想

我们这里的窗口是为提供dBASE III-PLUS应用系统的良好人机界面而设计的,并不是从形成一种完整的窗口操作的软件这一角度来考虑的,事实上要形成一种完整的窗口操作软件,要求是比较高的,它应该包含有:建立和初始化窗口的视框,启动和消失窗口,更改窗口的大小和移动窗口,专门的窗口输入输出函数,保存和恢复窗口的原内容。而利用dBASE III-PLUS有些工作是难以完成的(如迅速地保存和恢复屏幕原内容),有些工作在应用系统中又不是很必要的(如窗口大小的更改及移动),对于难以做到但必须的功能则通过一定的途径加以克服,对于不是很必要的东西则予以省略。

它的开发工具,都会使用户的开发工作变得更加快速、简便!

本文旨在为foxbase+2.10的广泛应用抛

对于保存和恢复屏幕原内容,我们是这样考虑的,窗口显示只安排在屏幕的空区,那么就没有必要再做保存和恢复原屏幕内容这样的操作了。

对于窗口大小的更改及移动的操作,由于实现起来比较困难,加上在应用系统并不是很必要,因此我们没有考虑这一部分的设计。

其它部分的窗口操作的设计,通过下面对它们的各种定义可以很清楚地反应出来。

必须强调的是 dBASE III-PLUS 的一些操作都是对数据库的操作,因此我们所设计的窗口操作的对象将分别对应于不同的数据库,每个窗口内所要完成的功能将分别是对不同数据库进行操作所要完成的功能。

三、基本定义

根据应用系统要求完成的功能及窗口操作的一些基本要求,在进行窗口操作的软件设计时,我们做了如下一些定义:

- (1) 窗口的视框用虚线矩形框;
- (2) ↑键有效,上移一行;
- (3) ↓键有效,下移一行;
- (4) →键有效,右移一个项目或在需要编辑时,对光标所在行进行编辑;
- (5) ←键有效,左移一个项目;
- (6) PGDN 键有效,向下翻一页(一页的长度由窗口的大小决定);
- (7) PGUP键有效,向上翻一页(一页的长度由窗口的大小决定);
- (8) Y键有效,增加窗口中的内容;
- (9) N键有效,删除光标所在行;
- (10) P键有效,打印输出窗口中的内容;
- (11) 起始光标定义于每一行的最左端;
- (12) ESC键退出窗口操作。

对于一些键的基本定义都作为我们窗口操作的操作命令。

四、程序设计

有了上面的这些定义,程序的设计并不十分

砖引玉,希望这一有力的工具能够得到更多的垂青,祝君好运!

困难,图一给出窗口操作软件的基本框图。在设计程序时主要是利用了 dBASE II-PLUS 提供的从键盘接收信息的函数 (READKEY()), 然后根据对一些键的不同定义提供不同的操作。

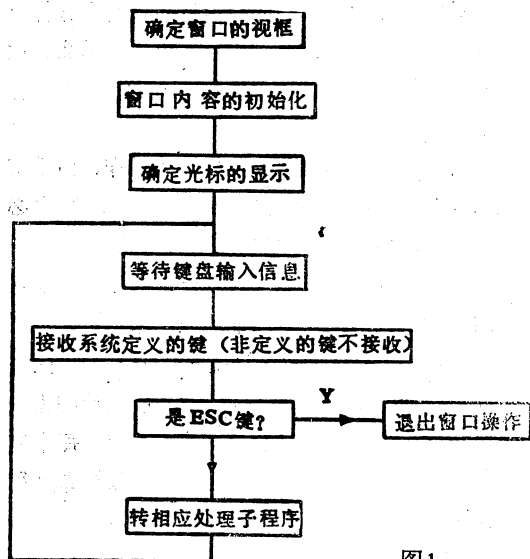


图1

确定窗口的视框比较容易,可只用一条SAY语句;窗口内容的初始化,是指显示窗口中各个项目的名称,按照用户前面工作的要求将库中满足要求的记录显示于窗口中(显示的量将受窗口定义的大小限制),这里的光标显示定义了一个空格变量,用GET语句从键盘接收信息,这样既可当做光标,又可接收键盘信息;对于前面所定义的非字符键的判别是利用READKEY()函数来进行的;至于窗口操作命令对应的处理子程序实现起来就比较简单了,都是一些常规的dBASE II程序的设计。

必须说明的一点是,当在窗口显示内容时,

由于受到屏幕宽度的限制,往往一行无法显示一条记录中的全部内容,特别是当每个字段内容都比较长的时候,这时我们采用截断显示加底行补充显示的办法来解决。截断显示即在窗口中按屏幕宽度(或窗口宽度)和字段的实际需要分配给每一字段一定的长度,在窗口中只能显示分配到的长度的内容,其他内容被隐含掉(在dBASE II-PLUS中可在SAY语句中加一定的标志完成此功能);底行补充显示,即定义紧挨窗口下的一行也为窗口操作服务,在这一行上显示光标所在字段的全部内容。一般是将被截断的字段放在第一个位置,光标首先就在此位置上,如果被截断的字段比较多,可用一键向右移动而达到查看全部内容的目的。

另外,在做截断显示时,虽然一个汉字占两个字符位置,但截断必须以单数为截断单位,这样有利于在编辑时录入或查看字段后面被截断的内容,否则将在操作过程中出现一些非字符的符号,影响操作过程。

五、结束语

采用窗口操作设计方法设计的应用系统许多地方都需要使用窗口操作,由于应用系统的窗口操作要求完成的功能基本相同,因此,不同的窗口实际上只是改变了操作的对象(数据库)或字段的名称,而整个程序的结构并没有发生变化。程序中只要做一些适当的变动就可以满足要求了。

窗口操作的设计在新疆维吾尔自治区科委办公室办公自动化辅助支撑系统的开发和研制的过程中得到了良好的运用,使用表明这种设计方法还有一定的潜力可挖。

测试异步通讯适配器

用于IBM PC/XT微机的异步通讯适配器以Intel 8250芯片为核心,具体电路可分为命令译码电路、内部数据缓冲器和与RS-232标准接口相关的控制口、状态口、串行数据输入输出电路等两部份。一般测试程序(如加电自检程序)都只能对其第一部份电路作检测。显然,即使测试通过,也不能说明适配器完全无故障。本文介绍一种测试方法,可以判别适配器能否正常工作。如有故障,还能根据测试程序执行的情况,对照适配器的电路图,确定出故障所在。

测试之前将一个25芯D型连接器插在适配器上,连接器的引脚应作如下连接:2—3,4—5—22,6—8—20,其目的是让经DTR,RTS和Sout线输出的信号能从CTS,DSR,RI,RLSD和Sin线上读回,用Debug程序,执行测试程序Test232,比较输入和输出的数值是否一致,从而可以很快诊断出有问题的芯片。检查故障时,应按先第一部份,后第二部份,先状态、控制口,后数据口的步骤进行。例如,状态、控制口的输入和输出数值不一致,则芯片U₁₂或U₁₄有错。如测试通过,则表示适配器完全能正常工作。需程序清单者,请同作者联系。

(重庆建筑工程学院计算中心 冉春林)

在数据库中如何快速处理多层次合计的表格

武汉水利电力学院计算中心 蒋仁兴

摘要: 在数据库中经常遇到要处理多层次合计的表格, 在计算合计时一般比较麻烦, 而且影响运算速度。本文介绍一种利用编码, 将它化成多叉树, 用高级语言进行计算的一种方法, 既简洁, 又快速。

一、引言

目前, 用计算机进行信息管理在国内已很受欢迎, 在数据库设计、程序设计时, 我们经常要处理多层次合计的表格(例图1), 碰到这样的表格, 在计算各层的小计或合计时, 非常麻烦, 而且严重影响运行速度, 当层次一多, 简直无法下手。为了解决这一问题, 本人摸索出一种利用编码, 将表格化成多叉树, 用高级语言计算的一种方法, 既方便设计, 又能快速计算。下面介绍这种方法的编码、数据库设计、程序设计。

二、编码方法

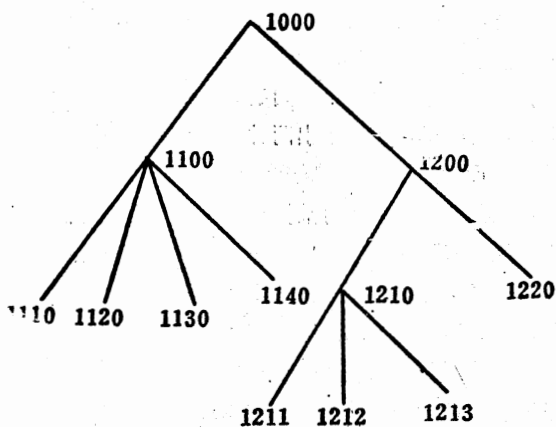
对于一个多层次合计的表格, 首先将其化成一棵多叉树, 然后对这棵树进行编码, 编码法则:

- 如果层次为 n , 那么我们就用 n 位编码。
- 编码中各个子节点依次排序(从 $1 \approx 9 \approx A \approx Z$, 不能重复)。
- 编码中非零位的个数就是节点的层数。

例图1、图2。

(图1) 表格

编码	项 目 名	用电量 (万度)
1000	全行业用电总计	100
1100	一、农、林、牧、渔、水利业合计	40
1110	1. 农业	10
1120	2. 林业	10
1130	3. 畜牧业	10
1140	4. 其它	10
1200	二、工业合计	60
1210	1. 采掘业合计	30
1211	① 矿业	10
1212	② 木材及竹材采运业	10
1213	③ 自来水生产及供应业	10
1220	2. 制造业	30



(图2) 多叉树的编码

三、处理思想

由于这种编码含有层次信息, 因此可以根据编码进行计算。计算时根据编码从后向前逐位进行, 进行到某位时, 各记录的编码如果满足该位为0而前位不为0, 则该记录便是此次的合计项。它是满足编码前面位与该记录前面位相同, 编码该位不为0且没有被计算过的各记录的和, 这样便可以计算出各个合计项的值。

四、程序设计

根据这种编码方便可以进行合算, 下面是一个使用FORTRAN语言编制的合算子程序。这个子程序提供三个哑元:

N——层次数

A——为所要合计的数组

NUMBER——为对应项的编码

防断电可中断的登帐程序设计

浙江师范大学计算中心

杨传斌

摘要 在管理信息系统中,有许多需要长时间连续工作的登帐程序,常常由于断电和需要中断等引起麻烦,本文通过分析后提出了两种妥善的解决方法。

在使用关系型的数据库管理系统 FOXBASE+ (或dBASE) 的过程中,由于断电造成数据损失,有时甚至引起整个数据库的破坏,从而造成信息系统的瘫痪,大大降低了系统的可靠性。本文在分析造成上述问题的内部原因的基础上,针对有无 UPS 提出两种预防断电且可中断的程序设计方法。

数据库管理系统对数据操作是对内存缓冲区进行的,若缓冲区中没有要操作的数据才从外存调入,当缓冲区满或正常关闭数据库时数据才全部存盘。当新增或改变的数据还在缓冲区中时就断电,该部分数据将会丢失;还有一种数据丢失是所谓的结构丢失,往往引起数据库的混乱,这种丢失中数据库的数据已存盘,但因文件未关闭就断电使数据库的结构没能及时更新,例文件长度、记录数等,造成所谓的空记录等现象,给数

据恢复工作增加更大困难。

通过以上分析表明只要及时清除数据缓冲区和更新库结构,就完全可以避免数据丢失现象。在 FOXBASE+ 中有一条命令 FLUSH, 它的作用是在不关闭数据库的前提下清除缓冲区的内容,我们可以利用它及时把改变的数据和更新的结构存盘,保持盘中的数据 and 结构一致,如果部分数据因断电丢失,只要记下前一状态,再次运行时恢复这部分数据,若这部分数据是由键盘录入的则需要重新录入,但最多也只有一条记录的数据,录入的工作量可以说是微不足道的。

在管理信息系统设计中,为了数据处理的方便和系统的可靠,在数据正式进入系统前一般先放在一个暂存库中,确认无误后再通过登帐程序正式进入系统,在实际应用中,往往由于暂存库中的数据较多或处理复杂,使整个登帐过程需很

```

SUBROUTINE STA(N,NUMBER,A)
INTEGER N,M
REAL A(500)
CHARACTER*10 NUMBER(500)
CHARACTER*1 ID(500)
! N 为层次数,M 为记录个数
! A 为所要合计的数组
! NUMBER 为对应项的编码
! 标识每位是否计算过

101 DO 5 J=1,M
ID(J)='0'
5 CONTINUE

DO 3 I=N-1,1,-1
DO 4 J=1,M
IF (NUMBER(J)(I+1:I+1).EQ.'0'.AND.NUMBER(J)(I:I).NE.'0') THEN
L=0
DO 6 K=1,M
IF (NUMBER(J)(1:I).EQ.NUMBER(K)(1:I).AND.
NUMBER(K)(I+1:I+1).NE.'0'.AND.ID(K).EQ.'0') THEN
1 A(J)=A(J)+A(K)
ID(K)='1'
L=L+1
ENDIF
6 CONTINUE
4 CONTINUE
3 CONTINUE

RETURN
END

```

好消息

为了保证本刊订户技术资料档案的完整性和系统性,并从中受益,本刊特决定将《89年全国计算机应用研究学术交流会议论文集》以半价优惠提供给本刊订户,每册仅收20元(含邮费),款到发书,售完为止。

本刊

“地震活动信息熵与分数维”研究的软件设计

新疆电子计算中心 龚宇清

新疆地震局 周仕勇

摘要 本文较为详尽地论述了地震活动信息熵与分数维系统的主要特点、使用规则、基本结构、使用要求,并给出了软件设计方法。

地震预报是世界上公认的一大科学难题。随着地震预报日益广泛的实践和日趋深入的研究,人们愈来愈认识到地震前兆是复杂的现象,引入现代物理学和系统科学的最新进展,给地震预报

科学的研究以有力的武器,从地震活动信息熵和分数维两个方面的研究,探索从总体上把握复杂异常信息的途径和方法,是一项有意义的工作。

长时间,而对操作员来说这个时间很难事先估计,在当前一些地区电力供应不正常情况下因断电或断电后UPS无法支持长时间的工作,还可能到了下班时间登帐还未结束等,造成许多麻烦。所以登帐程序应设计成防断电的又可中断的,下面举例分析具体实现方法:

其中有二个数据库,合同库HTK.DBF,到货单暂存库DHD.DBF,现要用到货单暂存库的数据来更新合同库的数据,为了方便,在DHD.DBF中加了一个逻辑字段YDZ,表示是否已登帐,登帐前全是.F.,忽略细节后的程序如下:

```

SELECT 1
USE HTK
SELECT 2
USE DHD
LOCATE FOR .NOT. YDZ
DO WHILE .NOT. EOF()
    .....
    SELECT 2
    REPL YDZ WITH .T.
    FLUSH
    IF UPPER(CHR(INKEY()))="Q"
        CLOSE DATABASE
        RETURN
    ENDIF
    SKIP
ENDDO
.....

```

&& 具体计算部分忽略

&& 清除缓冲区

上述程序中INKEY()函数功能是取键盘缓冲区的第一个键值,若没有按键盘则取到一个空串,程序继续运行。加了该语句后几乎不影响整体速度,但由于使用了FLUSH命令,每登完一条记录后就清除一次缓冲区,使速度大大下降,

最高可达100%以上,若有UPS支持应删去该行,以加快速度。不论何种方式均可按Q键退出运行,下次进入时自动继续。为了验证程序的可行性,在程序中多个用WAIT语句设置的断点处进行复位试验(相当于断电),结果数据和库结构均未出错,下次进入后继续运行正常。

若用户不使用附加字段,也可以用内存变量文件来实现上述功能,但需有UPS的支持,具体方法参见下列程序段:

```

SELECT 1
USE HTK
SELECT 2
USE DHD
IF FILE("HTDZ.MEM")
    RESTORE FROM HTDZ
    GOTO DHDREC
ENDIF
DO WHILE .NOT. EOF()
    .....
    SELECT DHD
    SKIP
    IF UPPER(CHR(INKEY()))="Q"
        DHDREC=RECNO()
        SAVE TO HTDZ
        CLOSE DATABASE
        RETURN
    ENDIF
ENDDO
ZAP
IF FILE("HTDZ.MEM")
    DELE FILE HTDZ.MEM
ENDIF
CLOSE DATABASE
RETURN

```

&& 具体计算

上述程序加断点并进行复位试验后,证明同样可以达到要求。

一、系统主要特点及使用规则

(一) 主要特点

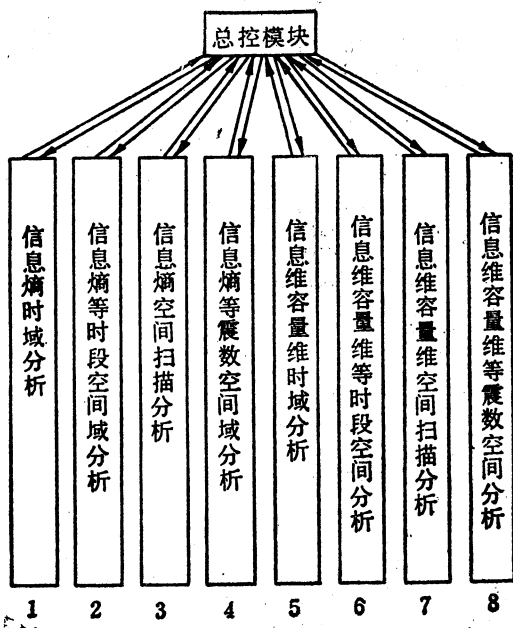
本软件的主要特点在于适应地震活动信息熵与分数维计算时多种方法、多变量的经常调整,尤其是时间域、空间域上多尺度嵌套扫描滑动,在计算空间分维时更为突出。

系统以BASIC语言编写,在PC系列机上均可运行,操作十分简便灵活,按照屏幕提示输入相应信息即可进行地震活动信息熵与分数维的时间域、空间域及空间扫描分析,并可连续对若干地震有选择地进行上述各种分析。各种参量设定自由,使用者可灵活选定。在时域、空间域上的滑动均依设定参数自动完成。系统按功能划分模块,复盖运行。程序总量约50KB。

(二) 基本结构

系统基本结构如上图所示。图中第7号模块是“信息维、容量维空间扫描分析”。由于地震定位存在误差,当研究小区值选得较小时(如 $2^\circ \times 2^\circ$),会使尺度分隔次数不足而导致无标度区难以确定,从而使分析失去意义。

总控模块实现对各种分析工作的选择控制,完成各种分析的切换,全局参量的初始设定以及结束处理。



(三) 使用要求

1. 数据文件格式

(1) 用于分析的地震数据文件可从地震数据库中摘取,并作为本系统的输入文件,其文件名作为全局参数。一次地震的七个数据项组成一个记录,格式为:

年,月,日,时,纬度,经度,震级
 $\times \times \times \times \times \times \times \times \times \times$ 度度分
 度度度分 $\times \times$

例 1970 5 21 10 43°42' 110°03' 4.2

经纬度和时值由程序自动转化为十进制数参与计算。

(2) 计算结果组成输出文件存于磁盘,其文件名由输入文件名后缀以“...DAT”组成,其中“...”具体为:

S1——信息熵时域分析; S2——信息熵等时段空间域分析; S3——信息熵空间扫描分析; S4——信息熵等震数空间域分析; W1——信息维、容量维时域分析; W2——信息维、容量维等时段空间域分析; W3——信息维、容量维空间扫描分析; W4——信息维、容量维等震数空间域分析。

每个文件的首记录为该次研究的名称、时间、经纬度范围、计算日期等信息。从第2个记录开始存放结果值,包括滑动序号、时域、空间域数值,地震次数及各种分析结果。每一记录存放一次滑动的分析结果。

2. 输入参数

一律按屏幕提示进行,包括地震数据文件名、研究的起止时间、经纬度范围,时域尺度滑动步长,扫描分析时小区及子区间的选定值,分维参数、输出方式等。

3. 在BASIC状态下,键入 RUN $\times \times$ SW ZK 命令启动本系统,而后按屏幕提示键入相应信息即可完成所需计算。

二、基本公式

(一) 缺信量与信息熵

1. 合成缺信量

$$I_0 = 1 / \sum_{i=1}^K \log \frac{1}{p(n_i)}$$

式中 $p(ni) = \frac{C^{ni}}{N} p^{ni} (1-p)^{N-ni}$
($N \leq 30$)

当 $N > 30$ 时

$$p(ni) = \frac{1}{\sqrt{2\pi} \cdot \sqrt{p(1-p)N}} \cdot e^{-\frac{t^2}{2}}$$

$$t = \frac{ni - Np}{\sqrt{Np(1-p)}}$$

2. 合成信息熵

$$H_1 = -\sum p(ni) \log p(ni) + ((1-p(ni)) \cdot \log (1-p(ni)))$$

$p(ni)$ 计算同上

3. 访问概率信息熵

$$H_{2.1} = -\sum p(i) \log p(i)$$

$$\text{式中 } p(i) = \frac{ni}{\sum ni} = \frac{ni}{N}$$

$$H_{2.2} = -\sum p(i) \log p(i)$$

式中 $p(ni)$ 计算同 1 中。

以上各种值再进一步归一化处理, 处理方法这里略去。

(二) 信息维与容量维

1. 容量维

$$D_0 = \frac{d \log N(r)}{d \log r}$$

$$\text{式中 } N(r) = \sum_{i=1}^K C_i$$

$$C_i = \begin{cases} 1 & \text{第 } i \text{ 小区有震} \\ 0 & \text{第 } i \text{ 小区无震} \end{cases}$$

对时间分形: $K = L/r$

对空间分形: $K = (L/r)^2$

2. 信息维

$$D_1 = -\frac{d \sum p_i \log p_i}{d \log r}$$

$$\text{式中 } p_i \text{ 为访问概率 } p_i = \frac{N_i}{\sum N_i}$$

三、具体设计

(一) 信息熵时域分析

取 K 个月为时间研究尺度并划定一定区域, 将时间尺度按月分成 K 个时段, 求出该区域该时段内的地震总次数 N 及发生在第 i 月的次数 $N(i)$, 依基本公式求出四种方法的熵值, 最大熵及归一化熵, 打印结果。而后按月进行滑动。

1. 处理步骤

(1) 设定研究的经纬度范围, 起止年月及时间尺度 K 值。

(2) 以研究的起始年月值作为时间尺度的起始年月, 求出时间尺度的末年月。

(3) 从地震数据文件中找出发生在本时间尺度的落在区域内的地震总次数 N , 并记录下每次地震的时间值。

(4) 按月分为 K 个子时段, 从 N 次地震中分离出发生在各子时段的地震次数 $N(i)$ ($i=1, 2, \dots, K$)。

(5) 由二所示的基本公式, 逐步算出各种方法的熵值, 并记录结果 (打印、存盘)。

(6) 根据设定的研究起止时间, 判定是否滑动。如果已到终止时间, 即转结束本次工作处理, 而后转总控选择其它工作; 否则, 往下进行。

(7) 按月进行滑动。确定新的时间尺度, 转

(3) 重复以上过程。

2. 滑动处理方法

(1) 时间尺度的始末月份各加 1, 经过年月关系的处理, 确定新的时间尺度的起止年月值。

(2) 从总次数 N 值中减去第一子时段值 $N(1) \Rightarrow N(0)$, 并将 $N(i)$ 值依次前移。

(3) 从数据文件中读出新的第 K 子时段的地震数 $N(K)$ 。

(4) 求出新的总次数值: $N(0) + N(K) \rightarrow N$ 。

以上过程避免了重复读取数据文件求总次数及分离 $N(i)$ 的工作。

(二) 信息熵等时段空间域分析

取一定时间长度为研究时段 (以月为单位), 划定研究区域, 将该区域分为大小相等的 K 个

“正方形”子区域, 求出研究区域内落在该时段的地震总次数及落在第 i 个子区域的地震数 $N(i)$ 值。依基本公式求出四种方法的熵值, 最大的熵及归一化熵, 保留结果。然后按预定的滑动步长进行滑动计算。

(三) 信息熵等震数空间域分析

划定研究区域, 取某时间值 (年月日时) 为起点时间, 以限定的次数 N 为总次数, 将该区域划分为 K 个大小相等的“正方形”子区域, 求出落在子区域内地震次数 $N(i)$, 依基本公式逐步求出四种方法的熵值、最大熵及归一化熵。而后按给定的地震次数或时间为步长进行滑动。

滑动处理方法:

按时间滑动: 预先指定以月为单位的步长值 E , 每次滑动时, 只将上一时段的起始值加上 E , 经年月关系处理后确定新的起始值, 再根据限定次数, 按上述过程处理即可。

按地震次数为步长滑动: 根据限定次数在数据文件中查找所需的地震时, 应按设定的次数步长值记录下相应的那次地震的年月日时值, 作为下一次滑动时地震计数的初值。年月日时的值唯一确定某次地震。此外, 确定是否滑动的标志是数据文件是否结束或者取出的地震次数是否达到限定值。

(四) 信息熵空间扫描分析

取一定的时间范围作为研究时段, 划定研究区域, 从中选取一个研究小区, 再将小区分为 K 个大小相等的“正方形”子区域。求出落在小区的地震总次数 N 及各个子区域的震数 $N(i)$, 逐步计算四种熵值, 输出结果后, 再让小区在研究区域中扫描, 区间扫描完后, 再以预定的时间步长进行滑动。

(五) 信息维与容量维的时域分析

1. 设定研究的区域范围、起止的年月日时值、时域尺度 E (以天计, 值为 2 的方幂), 时域滑动步长 (以月计)。

2. 确定分析的最小时段间隔 (1天/12小时/6小时)

3. 求出本次时间尺度的起始年月日时值及每次滑动应分隔时域的次数。

4. 求出落在研究范围内、本时域尺度的地震总次数 N 。

5. 由分隔次数控制, 逐次对时域进行分隔, 每次分隔后, 求出落在子时段内的地震数 $N(i)$, 进而求出该次分隔的信息维、容量维, 结果存盘打印。

时域分隔方法: 按时间 Δt 将研究时段分为 K 个子时段。

例 $E=512$ 天 最小 Δt 取 6 小时, 则

Δt	256	128	64	32	16	8	4	2	1	0.5
K	2	4	8	16	32	64	128	256	512	1024
	0.25									
	2048									

系统进行了 11 次分隔, 由此得到 11 组计算数据。

6. 按照预定的滑动步长, 进行滑动 (转 3 重复以上过程, 直至停止滑动为止)。

(六) 信息维与容量维等时段空间分析

将空间上选定的区域按“正方形”分成 K^2 个子区间, 求出落在选定的时间尺度内及该区域的地震总次数 N , 并分离出落在第 i 个子区间的震数 $N(i)$ 。 K^2 分别取 4、16、64、256、1024、4096。求出结果后, 再以设定的步长滑动。

(七) 信息维与容量维等震数空间域分析

根据限定的震数 N , 求出相应的时域范围, 具体计算过程同 (六), 最后或者选择时间步长滑动, 或者按震数步长滑动。

(八) 信息维与容量维空间扫描分析

在选定的研究区域内确定一个小区, 并将它分为 K^2 个等面积的子区间。 K^2 分别取 4、16、64、256、……, 求出小区内的信息维与容量维, 然后让小区在研究区域内以 1° 的步长滑动扫描, 每扫描一次, 求出一组信息维与容量维。最后再以选定的时间步长在给定的时域范围内进行滑动。

四、结束语

我们用以上软件对十一个地区十四次地震进行了计算, 结果表明是比较理想的。由于最初没有考虑研究区域取成“长方形”, 所以程序中一直按“正方形”设计。但稍作改动, 即可实现“长方形”区域的分析计算。

一个多机环境下的集成CAD系统

贵州省科委计算中心 张小平 巫晓龙

摘要 本文对多机环境下的集成CAD系统进行了系统环境分析,提出了三种CAD软件的图形文件相互自动转换方法,从而实现了资源共享和功能互补。

本文将介绍的系统是利用目前国内流行的auto CAD辅助绘图工具软件和smartwork电路板辅助设计软件以及最近国内外流行的ME10机械工程图形设计CAD软件的功能,经过我们自己研制的转换程序包change,异种机,异种操作系统通信软件HCOSC (Heterogeneous Computer and Operating System Communication System)组成的一个既有auto CAD的通用功能,又有smartwork的电子线路设计功能和ME10机械设计专用功能的多机组合CAD系统。该系统合理地利用了这三种独立软、硬件环境下的CAD软件的图形文件相互自动转换,通过HCOSC通信软件共享各环境下的软、硬资源,弥补各CAD软件的不足,使其共同完成更复杂的CAD任务。

一、系统环境分析

1. 三种CAD子系统的分析

auto CAD软件是一种通用CAD软件,目前国内已有多种高级版本流行(2.90版以上者支持三维图形设计),同时,汉化版已普遍活跃在各应用领域中。绘图仪上用的矢量汉字法也基本完善。

smartwork是一种交互式双面印刷电路板图形辅助设计软件,是目前国内应用最为广泛的电子工程CAD软件工具之一。具有快速、准确、高效的特点。但是smartwork有明显的缺陷。首先图形输出没有无级放大功能,输出图形只能按1:1和1:2两种。其次,在图形设计中不便编辑各种字符和辅助设计符号。第三,支持的绘图仪,打印机型号有限等。随着CAD/CAM的发展,smartwork已不能满足用户的要求。

进入90年代后,国外CAD/CAM已广泛

采用具有高速cpu,大容量内、外存的图形工作站,国内使用图形工作站的用户也越来越多。Hp9000计算机300, 800系列工作站上的ME10是世界上较为流行的一种通用机械设计CAD系统。它的支撑环境是与unix system V兼容的Hp-ux操作系统,采用高分辨率大屏幕显示器,数字化图形输入板,触笔(styles)以及各种绘图仪,打印机。由于这些优良的软、硬件支撑环境,使其有方便,友好的人机界面,完善的图形输出,输入设备。但是ME10也有它的不足,例如未提供三维图形设计功能,没有汉字处理能力等。

2. 异种机,异种操作系统的通信软件HCOSC和转换程序包change简介。

HCOSC是我们最近研制开发的专用通信软件,它主要支持运行XENIX (unix)系统的各种中小型计算机,高档微机与运行PC-DOS的IBMPC及兼容机之间互连的通信。HCOSC系统使用方便,在两种操作系统环境下有很好的用户界面。HCOSC不仅支持直接连接的近程方式,而且支持通过modem经电话线的远程方式。HCOSC提供了主机向PC机,PC机向主机的相互文件目录查询,文件查看,文件传输,以及键盘数据会话(即电话会议方式)等服务。

change是将三种CAD图形文件相互转换的程序包,它由PCB-ME, ME-aut, PCB-aut, aut-ME四个模块组成。各模块均用C语言编写,各模块功能将在以下转换方法中介绍。

本集成系统就是利用了HCOSC和change的功能,将三种CAD在不同软、硬环境下有机地连接起来,达到各子CAD系统既有高度的独立性,又有子CAD系统之间的相互协作。共享系统内各种软、硬资源,形成松散偶合的多机集成CAD系统。

二、转换方法

1. 将smartwork图形文件转换成ME10宏文件

将smartwork的图形文件转换成ME10宏文件使得在电路板设计中,可以充分享用ME10的图形编辑功能,方便地进行电路图的修改,审查对电路元件进行尺寸,正文标准。另外还可利用Hp-ux上的多窗口软件,同时编辑多个印刷电路板。

实现方法:将smartwork中图形输出模块plot产生的绘图文件*.PCB,通过HCOSC从PC机送入Hp9000计算机300或800系列工作站上,经格式转换程序包change的PCB-ME模块处理成为ME10可以接受的宏定义文件形式。ME10自动输入处理之。见图(1)

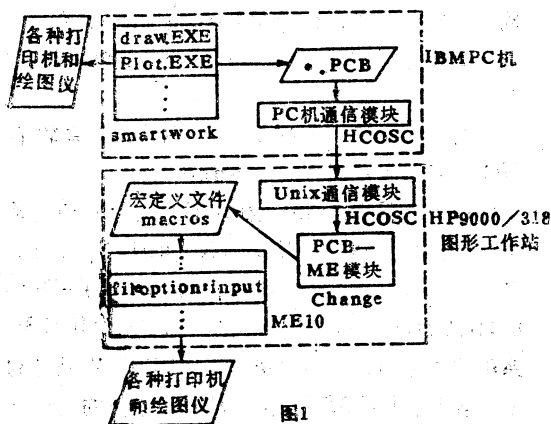


图1

2. 将ME10图形文件转换成auto CAD的绘图文件

由以上分析可知,ME10未提供三维图形设计功能,也没有汉字处理能力,这是它的明显不足。我们的工作是将ME10的功能间接移植于汉化auto CAD上,使以上不足得以克服。

实现方法:将ME10输出的绘图文件经change的ME-aut模块,转换成auto CAD的绘图交换文件*.SCR格式,通过HCOSC从Hp9000计算传入PC机,利用auto CAD的script命令装入auto CAD图形编辑窗口。见图(2)。

3. 同样我们利用HCOSC和change的

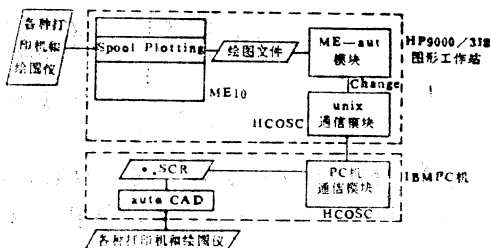


图2

PCB-aut模块,将smartwork的*.PCB文件,转换成auto CAD的*.SCR文件格式,实现了PC机上smartwork功能到汉化auto CAD的间接移植。见图(3)

change中的aut-ME模块是ME-aut模块的反转换模块。

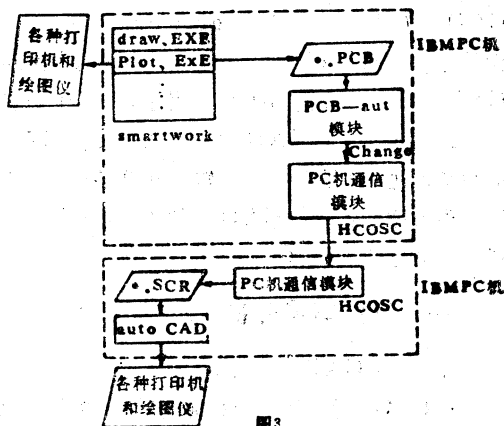


图3

三、结束语

经过一段时间的使用表明,我们的设计是行之有效的。目前,贵州省科委计算中心已经普遍采用了这个集成系统。使用引进设备Hp9000计算机318系列工作站环境编辑smartwork产生的印刷电路板,以及在auto CAD环境下对ME10工作进行汉字处理。从而使各子CAD软件和相应环境扬长避短,相互补充。随着计算机CAD技术在中国的普及,这方面可能做的工作还很多。在目前,我国外汇紧缺,软件开发人力不足的情况下,为了充分利用现有设备和软件,发挥其经济效益,该系统的设计方法无疑是一条节省人力,财力的捷径。

印制电路板图形的分块输出

西南交通大学 计算机系 方旭明 徐 锐

摘 要 在使用Smartwork 软件包时, 要用绘图仪或打印机输出印制电路板2:1图形。由于受绘图仪或打印机有效绘图尺寸的限制, 用户难以输出较大的图形。本文提出了一种在涉及较大图形时分块输出的有效方法。

Smartwork 印制电路板绘图软件包以其操作使用方便, 支持多种输出设备而具有广泛的用户。对于拥有绘图仪的用户, 可以将绘制的2:1图纸直接交生产厂家照像制板。由于一般用户所具有的平板绘图仪(如HP7475A, DXY880等)和打印机绘图尺寸较小, 在绘制2:1图形时, 即使不太大的电路板也极易超界, 从而限制了用户对于Smartwork软件包的使用。为了能够扩大绘图范围, 笔者编制了下述程序, 此程序可将edit方式下制作的图形分块处理, 各块图形单独绘制, 绘完之后, 将各块仔细地粘贴在一起, 即得到一幅完整的图纸。

一、图形文件的存贮格式

Smartwork图形的编辑是在一个直角坐标平面内完成的。进入edit状态之后, CRT屏幕上出现绘图平面的左下角窗口。左下角为坐标原点, 坐标单位为图形基元数。坐标形式如图1所示。

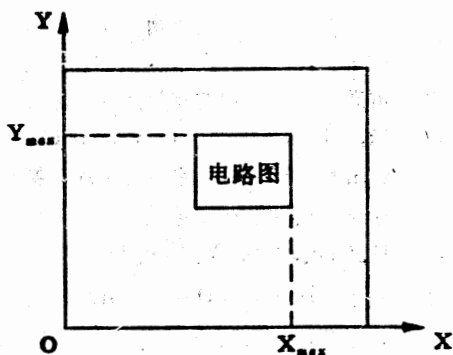


图1

Smartwork有16种图形基元, 各种图元分别用一个4bit代码表示。图元与图形代码对应关系如表1所示。

代码	0000	0001	0010	0011	0100	0101	0110	0111
图元								

代码	1000	1001	1010	1011	1100	1101	1110	1111
图元								

表 1

在存贮图形时, 以X, Y方向最大有效坐标值为X, Y方向图形边界。从绘图坐标原点开始以每两个图元代码组成一个字节逐行存放。为了便于按字节存贮图形代码, 若X方向坐标边界值为奇数, 则增加一个空图元代码(0000)凑足行代码数为整数字节。在组成图形文件时, 文件的第11、12字节为图形在X方向的边界值 X_{max} , 第13、14字节为图形在Y方向的边界值 Y_{max} (均为低字节在前, 高字节在后), 第15~16字节中的其余数据为固定的辅助信息。从第17字节开始顺序存放图形代码。图形正反两面, 图形正面存贮完之后, 再按相同的方式存贮图形反面。

二、图形的分块处理

根据用户所绘图形的形状, 可作两种方式的分块处理。

1. 水平分块

如图2所示, 若图形在Y方向超过绘图仪器的界限, 则可将图形作水平分块处理。

假设图形在坐标平面内Y方向最大有效行数为 ΔY , 则有:

- ① 若 ΔY 为偶数, 则子图1和子图2有效行数均取 $\Delta Y/2$;
- ② 若 ΔY 为奇数, 则子图1有效行数取为 $\Delta Y-1/2$, 子图2有效行数取为 $\Delta Y+1/2$;
- ③ 若用户自行选择分割位置或不希望分割

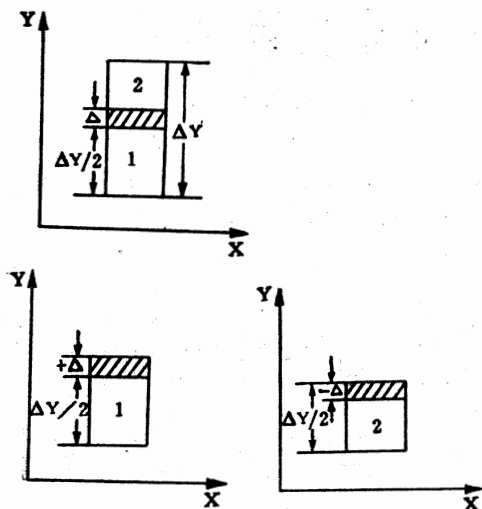


图 2

线正好经过一排焊盘的根部,则可加入修正系数,使分割线偏离上述分割线一个 Δ 值,取 $|\Delta| \leq \text{INT}(\Delta Y/2)$;

- ④ 若分块图形大小仍不符合要求,可重复①②③,直到满意为止。

2. 垂直分块

如图3所示,若图形在X方向超过绘图仪器的界限,则可将图形作垂直分块处理。

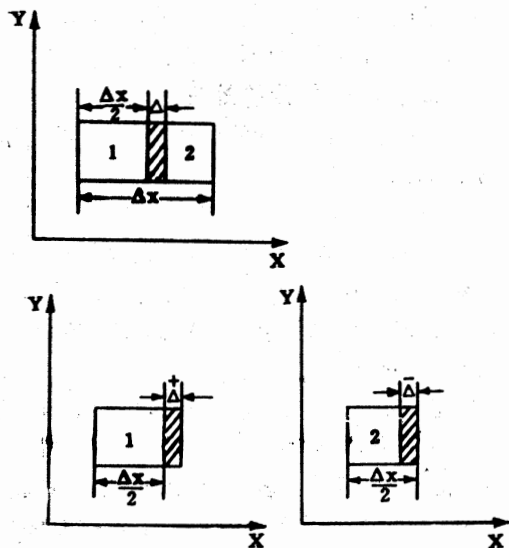


图 3

从图形存贮格式可知,图形文件中各行图元代码为偶数。

- ① 子图1和子图2在X方向有效图元数均取为 $\Delta X/2$;
- ② 同水平分块一样,垂直分块也可加入修正系数 Δ ,此时, $|\Delta| \leq \Delta X/2$;
- ③ 若分块图形大小仍不符合要求,可重复①②,直到满意为止。

上述两种方式下分块的子图形仍取原图的左边界和下边界。从原理上说,用户可以对一个图形作任意的水平和垂直分块。例如,若原图在X、Y方向均超界,则可先水平分成两块,再对两个子图各垂直分成两块。从实际经验来看,图形不宜分块太多,否则会影响图纸的质量和图形的精度,尤其是集成电路较多的电路板。此外,适当地选择分割位置将有利于提高图形的精度。

三、分块程序

分块程序由 Turbo Pascal 语言写成,运行环境为 IBM-PC/AT。程序经编译后即可运行。假设源文件名为 CUT.PAS, 则执行文件名为 CUT.EXE。程序操作方法如下(带有下列线部分为输入示例):

- ① CUT↵
- ② 提示: Input the file name of containing the circuit board,
(输入含电路板的文件名:)
输入: ABC↵ (按 Turbo Pascal 要求, 文件名长度不能超过 8 个字符, 下同)
- ③ 屏幕显示电路板的有效尺寸, 然后, 提示: Horizontal or vertical cutting? (H/V) (水平或垂直分割?)
用户根据需要选择输入:
H↵ (h↵) 或 V↵ (v↵)
- ④ 提示: Input sub-file name1: (输入子图1文件名:)
输入: ABC1↵
- ⑤ 提示 Input sub-file name2: (输入子图2文件名:)
输入: ABC2↵

- ⑥ 提示: Alter parameter delta
Y=? 或 Alter parameter
delta X=?

(修正系数 ΔY =? 或修正系数 ΔX =?)

输入: 0 ↵

- ⑦ 程序开始运行, 分块完成之后, 程序回到DOS状态。

- ⑧ 若用户要继续对子图ABC1和ABC2分块, 可重复执行①~⑦。

图形分块处理之后, 用户即可用 Smartwork软件包中的 plot 程序分块绘制图形, 绘完之后仔细地将各块图纸粘贴在一起, 绘图工作即告完成。本程序经笔者使用效果令人满意。

程序清单如下。

```

Program CutCircuitBoard (Input,Output);
Type
  FileType=File of Byte;
Var
  FileName, FileName1, FileName2: String[8];
  Source, Dest1, Dest2: FileType;
  cell, x1, x2, y1, y2, z1, z2, halfbyte: Byte;
  Xmax, Ymax, Xmin, Ymin, X1max, X2max, Y1max, Y2max, X,
  delta, deltaX, deltaY, line, col, layer, k1, k2, uni: Integer;
  ch: Char;
Procedure WriteByte1 (min, max: Integer);
Begin
  For col:=min To max Do
  Begin
    Read(Source, cell);
    Write(Dest1, cell); Write(Dest2, cell);
  End;
End;
Procedure WriteByte2 (Var Dest: FileType; min, max: Integer);
Begin
  For col:=min To max Do
  Begin
    Read(Source, cell);
    Write(Dest, cell);
  End;
End;
Begin
  Write('Input the file name of containing the circuit board:');
  Readln(FileName);
  Assign(Source, FileName);
  Reset(Source);
  Seek(Source, 10);
  Read(Source, x1, x2, y1, y2, cell, cell);
  Xmax:=x1+x2*256; Ymax:=y1+y2*256;
  Xmin:=maxint; Ymin:=0;
  k1:=0;
  For line:=1 To Ymax*2 Do
  Begin
    k2:=0; X:=0;
    For col:=1 To (Xmax Div 2) Do
    Begin
      Read(Source, cell);
      If (k1=0) And (cell<0) Then k1:=1;
      If k2=0 Then Begin If cell=0 Then X:=X*2 Else k2:=1 End;
      End;
      If X=Xmin Then Xmin:=X;
      If (k1=0) And (k2=0) Then Ymin:=Ymin+1;
    End;
  End;
  Close(Source);
  Writeln;
  Write('The approximate size of given circuit board:');
  Writeln((Xmax-Xmin)*0.05:1:2, ' by ', (Ymax-Ymin)*0.05:1:2, ' inches');
  Repeat
    Write('Horizontal or vertical cutting?(H/V)?');
  Until (ch='H') Or (ch='h') Or (ch='V') Or (ch='v');
  Writeln;
  Write('Input the sub-file name 1:');
  Readln(FileName1);
  Write('Input the sub-file name 2:');
  Readln(FileName2);
  Writeln;

```

```

Assign(Dest1, FileName1);
Assign(Dest2, FileName2);
Rewrite(Dest1);
Rewrite(Dest2);
Reset(Source);
WriteByte1(1, 10);
Read(Source, x1, x2, y1, y2, z1, z2);
Case ch of
  'H', 'h': Begin
    Write('Alter parameter delta Y:');
    Readln(deltaY);
    delta:=Ymax-Ymin;
    Y1max:=Ymin+(delta Div 2)+deltaY;
    Y2max:=Ymin+delta-(delta Div 2)-deltaY;
    y1:=Y1max Mod 256; y2:=Y2max Div 256;
    Write(Dest1, x1, x2, y1, y2, z1, z2);
    y1:=Y2max Mod 256; y2:=Y2max Div 256;
    Write(Dest2, x1, x2, y1, y2, z1, z2);
    Xmax:=Xmax Div 2;
    For layer:=1 To 2 Do
    Begin
      For line:=1 To Ymin Do WriteByte1(1, Xmax);
      For line:=Ymin+1 To Y1max Do WriteByte2(Dest1, 1, Xmax);
      For line:=Y1max+1 To Ymax Do WriteByte2(Dest2, 1, Xmax);
    End;
  End;
  'V', 'v': Begin
    Write('Alter parameter delta X:');
    Readln(deltaX);
    delta:=(Xmax-Xmin) Div 2;
    uni:=(delta+deltaX) Mod 2;
    X1max:=Xmin+delta+deltaX+uni;
    x1:=X1max Mod 256; x2:=X1max Div 256;
    Write(Dest1, x1, x2, y1, y2, z1, z2);
    X2max:=Xmin+delta-deltaX+uni;
    x1:=X2max Mod 256; x2:=X2max Div 256;
    Write(Dest2, x1, x2, y1, y2, z1, z2);
    Xmin:=Xmin Div 2;
    X1max:=X1max Div 2;
    Xmax:=Xmax Div 2;
    For line:=1 To Ymax*2 Do
    Begin
      WriteByte1(1, Xmin);
      WriteByte2(Dest1, Xmin+1, X1max-uni);
      If uni=1 Then
      Begin
        Read(Source, cell);
        halfbyte:=cell And Y0F;
        Write(Dest1, halfbyte);
        halfbyte:=cell And Y0F;
        Write(Dest2, halfbyte);
      End;
      WriteByte2(Dest2, X1max+1, Xmax);
    End;
  End;
End;
Close(Source);
Close(Dest1);
Close(Dest2);
End. ( main )

```


电视动态字幕的微机实现

空军第一航空技术专科学校计算机教研室 李志伟

利用动画技术可使计算机屏幕上显示的画面或画面中的一部分,能够按照一定规则或要求在屏幕上进行活动,这一般是通过下述方法实现的:先显示出活动对象的图形;再进行变换,然后擦去原来的图形,最后再显示出变换后所得到的图形。利用这个技术不但可以实现画面的动画,而且也可以实现字符的动画,但是由于字符在屏幕上的坐标与图形坐标不一样,每两个字符之间相差距离太大(每个西文字符占8个图形坐标),这样在每次擦去和重写字符时就会产生字符的跳跃现象,并出现闪烁。为解决此问题,我们可以采用一些高级语言提供的存图取图技术,在图形工作方式下预先把将要移动的一串字符显示在某一区域中,用存图语句将这一区域中的字符作为图形符号保存起来,然后再用取图语句将保存下来的这幅字符图象在屏幕的不同地方显示,连续改变显示位置的坐标,即可出现一幅连续移动的字符画面,从而实现电视动态字幕的显示效果。

下面一段程序演示了这个效果。该程序在Turbo Pascal 4.0系统上运行通过。

过程Move English将显示于屏幕最右端的英文字符串S缓缓向左移动,最后定位于图形坐标(X, Y)。

过程Move Chinese将显示于屏幕最右端的中文字符串S缓缓向左移动,最后定位于字符坐标(X, Y)。修改该过程,还可将字符串S定位于任一图形位置(X, Y)。而Delay Chinese过程则很方便地实现了中文字符串S从字符坐标(X, Y)处逐个向右输出的动态效果。

利用这个技术,用户可以根据需要编制一些过程,实现字符在屏幕上的各种动画,编制具有动态效果的电视字幕。

```

Program move;
uses crt, graph3, graph;
type string40=string[40];
var graphdriver, graphmode: integer;

procedure DelayChinese(x, y: integer; s: string40);
var i: integer;
begin
  for i:=1 to length(s) do
    begin
      gotoxy(x+i-1, y);
      write(s[i]);
      delay(60);
    end;
end;

procedure MoveEnglish(x, y: integer; s: string40);
var i, j: integer;
    p: pointer;
    size: word;
begin
  i:=320-TextWidth(s);
  outtextxy(i, y, s);
  size:=ImageSize(i, y, i+TextWidth(s),
    y+4*TextHeight('A'));
  GetMem(p, size);
  GetImage(i, y, i+TextWidth(s),
    y+5*TextHeight('A'), p^);
  for j:=i downto x do
    PutImage(j, y, p^, NormalPut);
end;

procedure MoveChinese(x, y: integer; s: string40);
var i, j: integer;
    p: pointer;
    size: word;
begin
  i:=312-length(s)÷(320 div 40);
  j:=((200 div 11)÷(y-1));
  gotoxy(40-length(s), y);
  write(s);
  y:=(200 div 11)÷y-1;
  size:=ImageSize(i, j, 320, y);
  GetMem(p, size);

```

MCS-51单片机多外中断扩展方法

安徽机电学院 骆德汉

摘 要 本文在介绍MCS-51单片机中断系统结构的基础上,给出一种MCS-51单片机响应多外中断请求源的扩展方法,并指出在应用扩展方法时所要注意的某些问题。

一、引言

MCS-51单片机由于其功能强、体积小,使用方便等特点,已在控制、仪表等领域中得到了广泛应用。但是MCS-51只有两个外部中断请求源输入端,它能响应外部中断源的个数由此而受到限制,难以满足有两个以上中断请求源的应用系统要求。为了扩展MCS-51响应外部中断源个数,扩大它的应用范围,在实践中,我们结合MCS-51中断系统结构特点,进行MCS-51单片机多外中断扩展设计,并取得了较好地应用效果。

二、MCS-51中断系统

MCS-51中断系统的结构如图1所示,它有五个中断请求,提供两个中断优先级,可实现二级中断服务程序嵌套。中断系统的编程可通过对内部中断优先级控制寄存器IP,中断允许控制寄存器IE,中断源寄存器TCON,SCON的有关位进行操作。

1. 外部中断请求

$\overline{\text{INT}_0}$, $\overline{\text{INT}_1}$ 是MCS-51的两个外部中断请求,通过对TCON寄存器有关位编程来选择它们的工作方式。当外部中断请求信号加到输入

端时,外中断请求标志 IE_0 , IE_1 分别被置“1”,由此向CPU申请中断。但是CPU是否能响应它们的中断请求则取决于IP,IE寄存器有关位的编程。

2. 计数器/定时器中断请求

MCS-51内部有两个16位计数器/定时器 CTC_0 和 CTC_1 ,通过对TMOD寄存器有关位编程,可选择 CTC_0 , CTC_1 的工作方式和启动它们的工作。当它们被启动工作后,内部计数器便从初值开始对外部输入脉冲或对内部机器周期进行加1计数。当计数器计数满时,最高位产生溢出,分别将中断请求标志位 TF_0 , TF_1 置为“1”,所以 CTC_0 , CTC_1 的中断请求源传递到中断系统的信息是 TF_0 和 TF_1 ,但 TF_0 , TF_1 传递到中断系统后,CPU能否响应它们请求同样受IP,IE两寄存编程控制。

3. 串行口中断请求源

MCS-51内部有一个全双工串行接口,具有地址相同,物理上独立的发送缓冲器和接收缓冲器,串行口由SCON,PCON两特殊功能寄存器控制其工作方式及波特率。当CPU启动发送缓冲器发送完一个数据或置串行接收允许位REN为“1”,接收缓冲器收到一个数据后,发送中

```
GetImage(i,j,320,y,p^);
x:=(x-1)*320 div 40;
y:=j;
for j:=i downto x do
  PutImage(j,y,p^,NormalPut);
end;

BEGIN
  graphdriver:=cga;
  graphmode:=cgac0;
  graph:=graph(graphdriver,graphmode,'');
  setgraphmode(cgac0);
```

```
graphcolormode;
graphbackground(9);
palette(3);setcolor(1);
settextstyle(triplexfont,horizdir,3);
DelayChinese(3,3,'空军第一航空技术专科学校');
MoveChinese(9,5,'计算机教研室');
outtextxy(80,100,'Moving');
MoveEnglish(80,130,'Moving');
repeat until keypressed;
clrscr;
END.
```

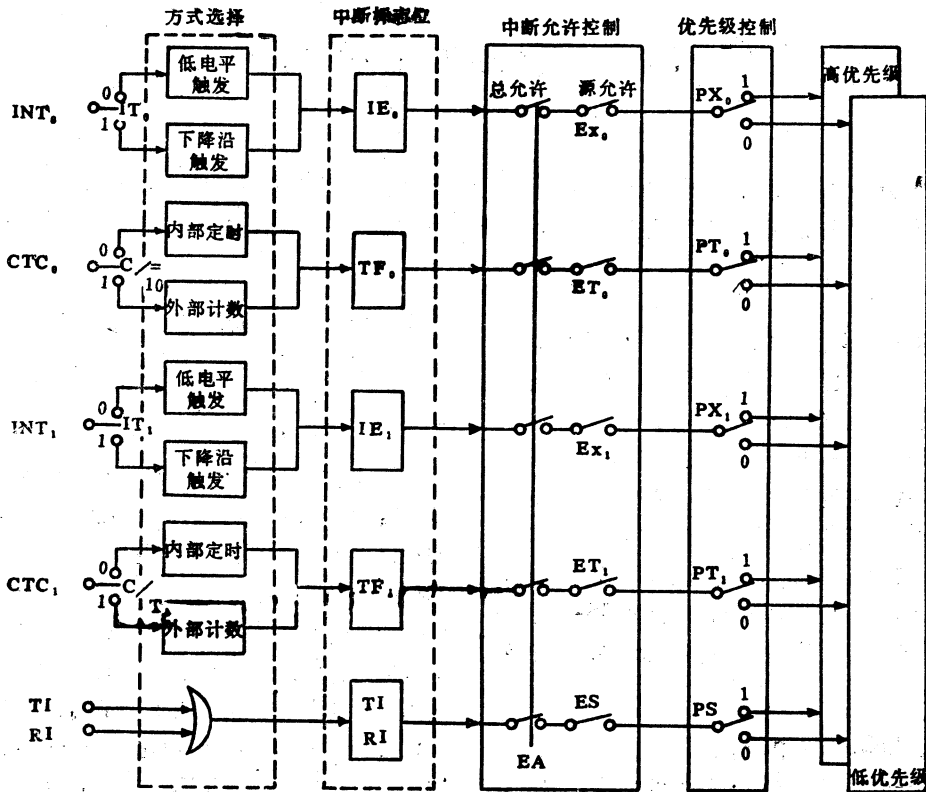


图1 MCS-51中断系统结构

断标志位TI或接收中断标志位RI分别由硬件置为“1”，即串行口中断请求传递到中断系统的信息是TI和RI，但cpu能否响应串行中断仍取决于IE和IP有关位的编程。

4. IE、IP特殊功能寄存器

MCS-51的五个中断源向cpu申请中断时，cpu能否响应它们都受IE、IP寄存器初始化编程控制。IE与中断有关的位如下所示：

EA	/	/	ES	ET ₁	EX ₁	ET ₀	EX ₀
----	---	---	----	-----------------	-----------------	-----------------	-----------------

EA是cpu开放中断标志位，当它程控为“1”时，cpu开放所有中断，EA程控为“0”时，屏蔽所有中断请求。ES、ET₁、EX₁、ET₀、EX₀分别是串行中断、CTC₁、INT₁、CTC₀、INT₀中断允许控制位。当它们程控为“1”时，各中断分别被允许，当它们程控为“0”时，各中断分别被禁止。

IP是中断优先级控制寄存器，对它的有关

位操作可使某个中断请求程控为高优先级或低优先级。它与中断有关位如下所示：

/	/	/	PS	PT ₁	PX ₁	PT ₀	PX ₀
---	---	---	----	-----------------	-----------------	-----------------	-----------------

PS、PT₁、PX₁、PT₀、PX₀分别是串行中断、CTC₁、INT₁、CTC₀、INT₀中断优先级控制位。当它们程控为“1”时，各中断源分别定义为高优先级，若它们程控为“0”时，各中断源分别被定义为低优先级。

三、响应多外中断请求源的扩展设计

根据MCS-51中断系统结构特点，在它所具有五个中断基础上进行扩展设计，可以扩充它响应外中断请求源的个数。

1. 用CTC₀、CTC₁中断扩展外中断

MCS-51内部CTC₀和CTC₁在某些应用系统中可能不被使用，此时可将它们的中断作为外部中断请求使用。当CTC₀或CTC₁程控为计数器方式工作，则它们计数输入端P_{3.4} (CTC₀)，

$P_{3..4}$ (CTC_1) 引脚有负跳变信号时, CTC_0 或 CTC_1 便加1。利用这个特性, $P_{3..4}$, $P_{3..8}$ 即可作为扩展的外中断请求输入, 而 CTC_0 、 CTC_1 的溢出中断请求标志 TF_0 、 TF_1 就是相应扩展的外中断请求标志, cpu 响应它们中断请求实际就是响应扩展的外中断请求。例如将 CTC_0 程控为计数器状态下方式2工作(自动恢复常数), 将 TH_0 、 TL_0 初值设置为 $0FFH$, 当接在 $P_{3..4}$ 引脚的外部中断请求信号发生负跳变时, TL_0 内容加1而溢出, TF_0 被置“1”向 cpu 发出中断请求, 同时 TL_0 初值又自动恢复为 $0FFH$ 。这样 $P_{3..4}$ 上每一次出现负跳变信号将 TF_0 置“1”, 向 cpu 申请中断, cpu 响应该中断请求时, 可执行扩展的外中断服务程序。所以 $P_{3..4}$ 相当于边沿触发方式的外中断请求源输入端。

2. 用串行中断扩展外中断

MCS-51的串行接口被程控为方式0工作时, 它被定义为移位寄存器, 此时可将它用做外中断请求输入。当串行口程控为方式0并置串行接收允许位 REN 为“1”, 串行口便被启动开始接收数据, $RXD(P_{3..0})$ 为串行数据输入端, $TXD(P_{3..1})$ 为同步脉冲信号输出端, 接收缓冲器以振荡器频率的十二分之一(一个机器周期)的波特率接收 RXD 端输入的数据信息。当接收到8位数据时, 接收中断标志位 RI 被置“1”, 向 cpu 申请中断, cpu 响应此中断时即可执行扩展的外中断服务程序。用串行口扩展外中断的硬件结构如图2所示。74LS165并行输入, 串行输出移位寄存器的并行输入端接固定电平, “移/置”控制端接固定高电平, 外中断请求信号接“时钟禁止”输入端并设为低电平有效。当外中断请求源为高电平时, 移位寄存器被禁止串行移位输出, 当外设申请中断, 请求源为低电平时, 74LS165开始移位输出, MCS-51便接收数据, 并在8位数据接收完后向 cpu 申请中断, 此时中断服务程序即可设计为扩展的外中断服务程序。

3. 用一个外中断扩展多个外中断

上述方法最多只能扩展三个外中断, 若应用系统有更多的中断源时, 可利用MCS-51的一个外部中断 $\overline{INT_0}$ 或 $\overline{INT_1}$, 结合软件查询来扩

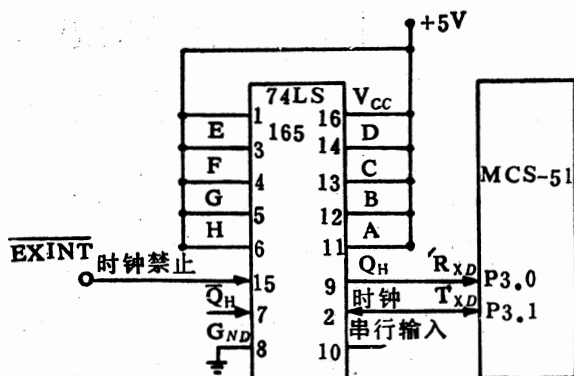


图2. 串行口扩展外中断硬件结构

展, 硬件结构如图3所示。当某外中断申请中断服务时, $\overline{INT_0}$ 就有中断请求信号, 向 cpu 申请中断, cpu 响应中断执行服务程序时, 首先依次查询 P_1 口的中断源输入端状态, 辨识中断请求源, 然后转入相应的中断服务程序。

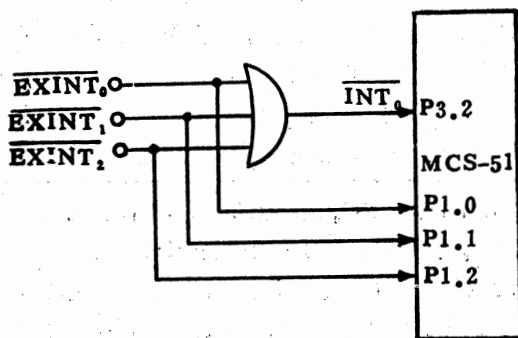


图3. 一个外中断扩展多个外中断硬件结构

四、外中断扩展需注意的问题

在应用系统中, 用上述方法扩展MCS-51响应外部中断请求源个数时需注意以下问题:

1. MCS-51中断系统只有五个中断, 两个优先级及两级服务程序嵌套, 虽然系统能扩展响应多个外中断源, 但它们只能依赖于这五个中断中的某个或某几个来响应外扩展中断请求, 它们不能实现超过两个优先级或两级服务程序嵌套而工作。

2. MCS-51的五个中断入口地址是固定的, 由内部电路产生, 它们分别为 $\overline{INT_0}$ 、

贵州省科技管理信息系统数据库的设计与实现

贵州省科委计算中心 贾洪钧

摘要 本文以贵州省科技管理信息系统数据库为原型,较全面地介绍了对管理信息系统的性能和效益都起到决定性作用的系统数据库的设计与实现的方法和步骤。文中所建立的科技管理信息的数据模型、以扩充关系数据库管理系统来实现对网络(层次)数据库的管理、通过对用户访问权限的控制来克服或降低数据库并发操作异常以及实现远程通信、联机检索的安全性控制等办法,对于大型数据库的设计与实现都具有实际的推广和参考价值。

一、概述

建立贵州省科技管理信息系统 GZSTMIS 是贵州省科委为迅速实现贵州省科技管理现代化、科学化、信息化的有卓识的决策,是改革传统的科技管理模式的必由之路。

GZSTMIS 是一个基于区域性计算机网络,覆盖全省科技管理部门的大型管理信息系统。它由科技信息管理中心 SIMC,数据处理子系统 SDPS,数据采集与传递子系统 SDCT 三部分所组成,其结构示意如下:

SIMC 实现全省科技管理信息数据库的存储、管理、维护,提供进行科技管理所需的信息

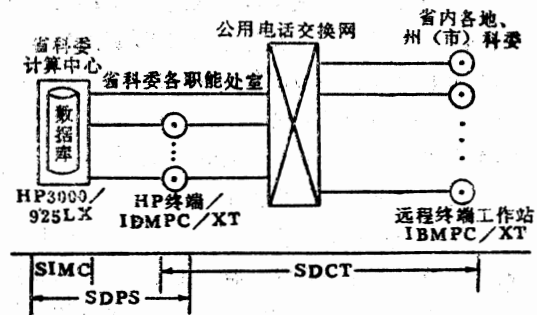


图1-1 GZSTMIS系统结构示意图

服务,并负担与国家科技信息系统及省内其他信息系统联网运行。

(0003H), CTC₀(000BH), INT₁(0013H), CTC₁(001BH), TI和 RI(0023H), 串行中断的接收中断 RI 和发送中断 TI 是同一个入口地址。cpu响应某个中断时,则硬件将相应入口地址送入程序计数器PC,执行中断服务程序。在扩展外中断时,扩展的外中断入口只能和被用来扩展的中断入口一致。

3. 用串行中断扩展外中断时,相当于将外中断请求源设置为低电平触发方式,请求源为高电平时,移位寄存器输入端固定电平被置入计数器,请求源为低电平时,移位寄存器产生串行移位输出,向cpu申请中断。cpu响应串行中断时可以去读接收缓冲器的数据而直接执行中断服务程序,并在中断服务程序执行完之前禁止接收串行口输入的数据,同时需保证在中断服务程序执行完之前外中断请求信号无效;避免重复响应中断的错误。

4. 当 INT₀、INT₁被设置为电平触发方式时,中断请求信号应是低电平有效,并一直保持到该中断被响应,同时在该中断服务程序执行完之前,请求信号又必须清除,否则将产生另一次中断;当它们被设置为边沿触发方式时,由于cpu每个机器周期采样一次外中断输入电平,所以中断请求源的高电平和低电平持续时间都必须大于一个机器周期(12个振荡周期),才能保证cpu正确检测外中断源从高电平到低电平的负跳变请求信号。所以用 INT₀或 INT₁扩展多外中断时,应用系统外设所产生的中断请求信号应满足上述要求,否则外设中断请求得不到正确响应。

五、结束语

本文介绍的多外中断扩展技术是作者的实践总结,其特点是结构简单,使用方便,大大扩展了MCS-51的应用范围。

SDPS为多用户计算机系统,它可按各终端用户所确定的信息处理需求,以终端多道程序执行方式或工作站独立工作方式进行集中式或分散式数据处理,产生管理、决策所需的各种结果信息。

SDCT以公用电话交模网为通信介质,实现SIMC、SDPS各工作站与采用相同数据模式、相似管理模式的各地、州(市)科委微机工作站间的管理信息(包括图文真迹信息)的远程联机检索、联机通信。

鉴于篇幅所限,本文仅对GZSTMIS系统

数据库的设计与实现的方法、步骤作一简要说明。

二、系统数据要求

通过对贵州省科委及其各管理职能部门实施科技管理的各个环节、组织机构、管理范围、管理流程、数据来源、数据处理方法、数据流向及流量等情况,包括其历史、现状和发展情况的深入调查、讨论、分析、研究,并考虑到可投入研制的资金、设备、技术力量等实际,GZSTMIS系统数据库必须支持如下所示的应用软件体系(图2-1)。

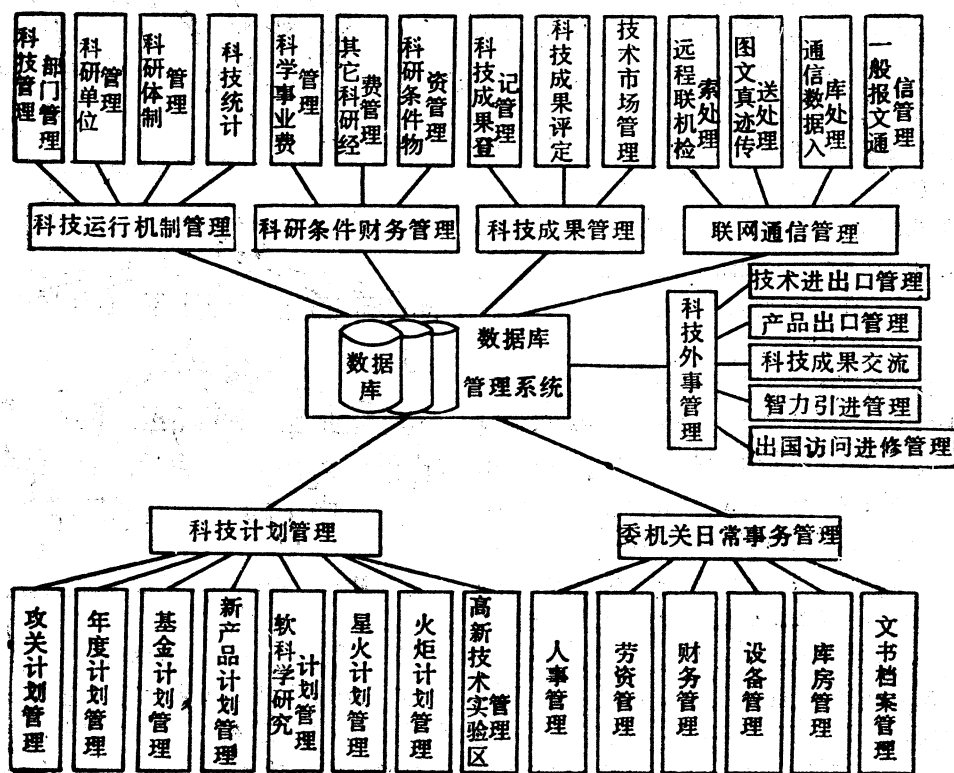


图2-1 GZSTMIS的应用软件体系

因此,本系统至少应有下列数据:

1. 科技管理机构数据:指描述各个科技管理机构情况如人员、活动、经费等的的数据;
2. 科研单位数据:指描述各个科研单位情况如建立、人员、设备、经费、效益等的的数据;
3. 计划管理数据:指描述各种科技研究开

发应用项目计划如年度、星火计划等及其管理活动如申报、审批、验收等的的数据;

4. 财务管理数据:指描述进行科技活动的各种经费收支、核算等情况的数据;

5. 条件管理数据:指描述科研条件、物资及其管理如分配、投入等情况的数据;

6. 成果管理数据: 指描述各个科技成果状况及其管理活动如登记、评定、交易等情况的数据;

7. 外事管理数据: 指描述各种科技外事活动及其管理活动的数据;

8. 文档管理数据: 指描述各种科技文件及其管理如收发、借用、处理等情况的数据;

9. 委机关日常事务管理数据: 指描述省科委机关日常事务如人事、劳资、库房、设备等情况及其管理活动如进出、借还等情况的数据。

三、建立数据模型

本系统的数据模型是在充分认识和深刻理解各有关管理职能实施方式的基础上, 按照 E-R 法来建立的, 具体步骤如下:

1. 确认实体

将描述各管理对象的所有数据项按其描述的不同方面划分成数据项集, 再按对各方面数据的处理要求和数据项间的相互依赖性来决定这些集合的再划分或合并。如此而形成的每一集合则可作为实体的属性集。[注]。

2. 确认联接

根据各实体在描述管理对象上的联系(如“明细”, “分类”等)及数据处理活动(如“累计”, “调用”)来确定实体的属性间及实体之间的联接方式及该联接的属性集(见图4-1)。

3. 绘制信息模型——E-R图

将所确认的实体及联接按 E-R 图要求绘制出来, 则形成系统数据库的信息模型(图4-1)。

4. 转换出数据模型

按 E-R 图转换数据模型的规则, 分别将实体和联接转换为相应的记录型、系, 联接——记录型, 注意处理好可能出现的实体分解或归并, 从而生成系统数据模型。从图4-2可见, GZSTMIS 的系统数据模型是一个以层次模型为基调的网络模型。

5. 选取数据库管理系统

出于本数据模型和应用程序要利于在只具有关系数据库系统 dBASE II 的各终端工作站, 特

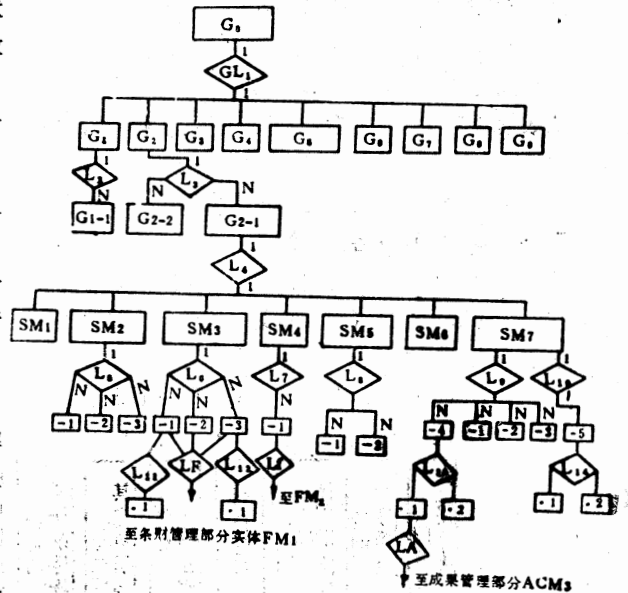


图4-1 科技运行机制管理部分的信息模型

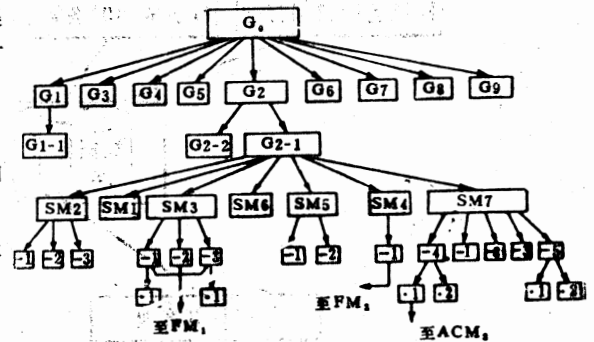


图4-2 科技运行机制管理部分的数据子模型

别是远程终端工作站(地区级科委)上的推广使用, 降低远程通信数据入库时的格式检查和编排的开销, 以及系统造价所限等因素, 我们采用了 Hp3000/925Lx 上的关系数据库 ALLBASE/xL 和结构化查询语言 SQL 来完成上述网络模型的数据模式建造。

6. 确定记录格式

本步是以规范关系和实用形式相结合来造就模式, 其步骤如下:

(1) 建立关系集: 在将各联接下方记录型

[注]: 鉴于篇幅所限及所用数据的规范化、标准化考虑, 本文中的例子均只举出科技运行机制管理部分。

的主关键字（机内记录号）列为该联接上方记录型的数据项之后，所得到的新的记录型集合就是网络模型的关系集，每一记录型则成为一个关系式。

(2) 规范化处理: 仔细确认各数据项在语义上的确切含义和和管理中所起的作用, 确定出各关系之间及关系的数据项间的依赖性, 通过关系的等价分解, 将所有关系化成 3NF 或 BCNF 型式, 形成 3NF 规范关系集。

(3) 关系模型的网状处理：由于科技管理所需的数据文件又多又大，为提高查询速度而设置的索引、例排等文件不仅使数据大大冗余，而且对完整性、一致性极为不利。为此，我们对联接两头的记录采用全部保留下方记录，消去上方记录中的冗余项并设置一个指示下方记录位置指针的办法，重新修正各记录，完成对关系库的网络化处理，从而构成网络模型。

(4) 鉴于本系统的大量数据主要用于统计分析, 查询统计操作频繁, 为提高处理速度, 我们将需经常查统的数据预先处理而形成相应的统计结果文件, 使数中增加了一些必要的文件以支持查统操作——变之为查询操作一次即可。

经上述步骤处理后, GZSTMIS 的记录结构如图6—1所示.

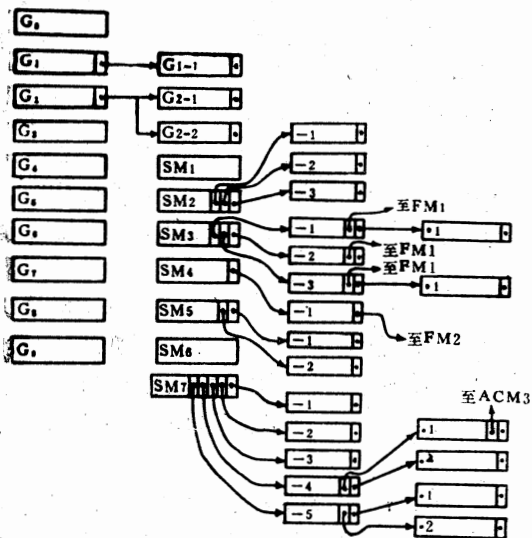


图6-1 科技运行机制管理部分数据结构

四、关于完整性、安全性的考虑

为保证数据库的合理使用、正常运转,我们对其完整性、安全性有如下考虑。

1. 系统对所有入库数据, 包括远程通信的入库数据均需进行格式、类型、名称、值域、编码等检查, 其任一检查项与定义不符, 不准入库。
2. 按科技管理组织层次, 上级可以“读出”下级的任何数据, 但不能作“写入”操作。
3. 各职能部门只能操作自己的数据、执行自己的应用程序; 在获得其他部门许可后, 方可“读取”被允许使用的数据, 但不得执行“写入”操作。
4. 每个用户都可以申请对自己所占有的数据作任意操作(称全权登录), 但某用户已作全权登录后, 该用户的其他登录则只能对数据作“读操作”。
5. 系统将自动检查远程通信、检索请求的用户标志, 若标志不符则系统拒绝该请求。
6. 系统自动对欲作远程传输的数据加密传输。
7. 任何终端用户的库维护请求均只作登记并通知系统管理员执行, 终端用户的库维护操作无效。

2. 按科技管理组织层次, 上级可以“读出”下级的任何数据, 但不能作“写入”操作。

3. 各职能部门只能操作自己的数据、执行自己的应用程序；在获得其他部门许可后，方可“读取”被允许使用的数据，但不得执行“写入”操作。

4. 每个用户都可以申请对自己所占有的数据作任意操作(称全权登录), 但某用户已作全权登录后, 该用户的其他登录则只能对数据作“读操作”。

5. 系统将自动检查远程通信、检索请求的用户标志,若标志不符则系统拒绝该请求。

6. 系统自动对欲作远程传输的数据加密传输。

7. 任何终端用户的库维护请求均只作登记并通知系统管理员执行, 终端用户的库维护操作无效。

以上各条, 均在系统主程序内予以处理.

五、支撑环境

- 1. 硬件支持:
 - Hp3000/925Lx小型计算机系统; 主存24MB; 硬盘1240MB; 速度3.2MIPS, 通信口22个, 其中远程modem口6个;
 - Hp Vectra 终端及 IBM PC 等微机系统 28套。
 - 主存 $\geq 640\text{KB}$; 硬盘 $\geq 20\text{MB}$;
 - 调制解调器 (modem) CCITT V.22/V.23; 全双工; 速度 ≥ 1200 波特;
- 2. 软件环境
 - 主机软件: 操作系统HpMPE/XL; 编程语言C/xL, PASCAL/xL, SQL/xL; 关系数据库ALLBASE/xL。

• Hp3000/925Lx小型计算机系统:

主存24MB;硬盘1240MB,速度3.2MIPS,
通信口22个,其中远程modem口6个;

· Hp Vectra 终端及 IBM PC 等微机系统
28套。

主存 $\geq 640\text{KB}$; 硬盘 $\geq 20\text{MB}$;

• 调制解调器 (modem)

集散型空调计算机控制系统

安徽省计算中心 叶定治 李淑莲

摘要 本文在论述了集散型空调控制的特点及其系统方案设计中的主要依据、目标基础上,提出了关键性的实施方法,对中小型控制系统的设计具有一定参考价值。

一、前言

空调对实现“四化”有着重要作用,无论是精密加工、电子器件生产、轻纺印染、烘烤卷烟,还是温室栽培、宾馆剧院、库房商场等都有空调通风要求。

空调控制的特点是:检测点数量多而分散;设备数量大而种类繁多,给控制系统设计带来一定困难。按照现代控制理论构成的集散型计算机控制系统,能够较好地解决空调控制问题,既可用于综合管理,又可实现空调过程最佳工况运

行。

阜阳卷烟厂新建厂房占地面积贰万平方米,厂房内部工作区划分见表一。整个空调工程有二套共六台PB型空调机组,其中一套三台PB-160型机对底层集中供气;另一套三台PB-120型机对楼层供气,二套机组彼此独立工作。

影响空调运行工况的因素较复杂,如日照量变化,室外空气温、湿度变化,室内设备的散发(吸收)湿、热量以及新风量大小等。因此具体分析当地气象资料,综合考虑上述因素,对确定

终端工作站软件:CCDOS2.1以上;
dBASE III, C, PASCAL; 近程工作站联
机程序Hp Advancelink; 远程工作站通
信检索程序(自制)SIE, GFRT, RLR;

六、数据操纵命令扩充

鉴于SOL/xL的数据操纵语句尚不能完全满足对GZSTMIS的网状数据库的操作,我们使用其他语言编程实现下述操作例行程序:

1. 数据录入

本例程包括结构指针生成(在有关文件内定位一个空记录并将其记录号填入相应指针字段)、原始业务数据装入、自动统计有关业务数据生成相应文件等模块。

2. 数据查询

本例程包括识别访问权限、确认查询路径、鉴别查询条件、输出查询结果等模块。

3. 数据修改

本例程包括数据查询例程及填写新数据等模块。

4. 数据增、删

本例程包括访问权限识别、增删路径及位置的建立与确认、结构指针生成与修改、数据输入与清除、统计结果修正与填入等模块。

5. 远程通讯数据入库

本例程包括识别通信标志,检索和重布数据格式、数据增添例程等模块。

鉴于篇幅所限,详细的建库步骤及程序不再赘述。

七、结束语

由于我们是首次进行大型信息系统的建设,技术水平和实际经验不足,在设计和实施GZSTMIS数据库工作中难免有缺陷和错误;数据库中所用数据也并非全是标准化、规范化的;数据模型、文件结构、数据项选择也不尽合理;许多指标尚留待试运行加以调整和修正;本文的叙述说明亦难免挂一漏万,还望同行赐教。

尽管本系统尚存在许多不足,但GZSTMIS的实现和运转,必定会有力地推动贵州的科技管理工作,促进其他MIS的建设。我们相信,在我们继续努力下,GZSTMIS一定能趋于完善。

本文撰写过程中,得到计算中心有关同志的支持和协助,黎蓉同志校对了全文,在此一并致谢!

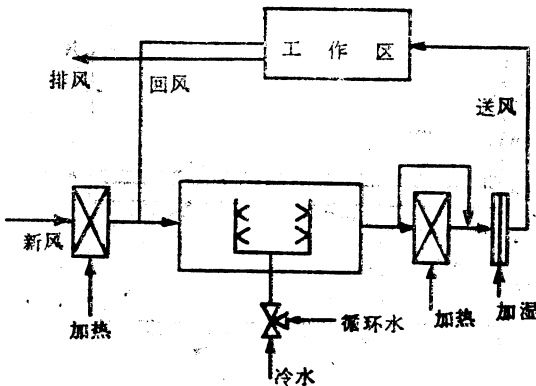
表一 室内工作区空气参数

工作区	参数	夏日 温度 $t_{s,x}^{\circ}\text{C}$	冬日 温度 $t_{s,d}^{\circ}\text{C}$	相对 湿度 $\phi\%$	送风量 M^3/h	换气 次数 次/h
底层车间		25~28	20	72	400800	12
贮叶、梗房		35~40	22~27	75~78	22400	13
二楼车间		22~30	22	58~65	252000	10
贮丝房		25~30	25	62~68	27150	8
进口机		22~25	22	58~65	29000	29
烘房		50	50	60		

一个在满足生产工艺所需的人工气候条件下,达到最大限度地节省能源的空调控制设计方案尤为重要。

二、空调控制方案确定

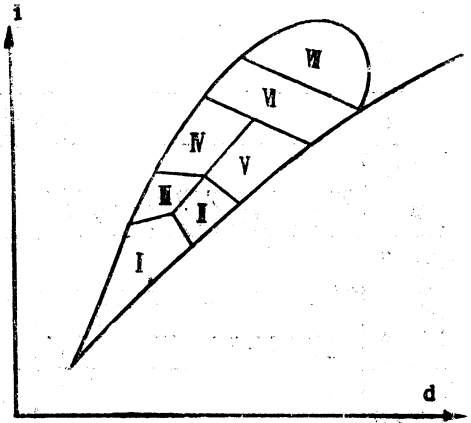
图一为PB型空调系统示意图。由图知:为一次送风型。工作区温、湿度参数受加热器(加温)、加湿器(加湿)和淋水室(降温、去湿)控制,显然二者是相关函数。



图一 空调工程示意图

根据气象资料综合分析,按工况分区绘制出阜阳地区的*i-d*焓湿图如图二所示。由此可得到全年工况自动转换条件见表二。

从节能考虑,按全年工况划分,确定系统最佳运行工况的全年能耗图如图三,作为控制系统方案设计的依据。

图二*i-d*图工况分区

由表一知:各工作区参数控制为恒湿、温度有偏差系统。从节能出发,结合室外空气参数全年变化规律,在满足工艺要求前提下,系统设计确定各工作区全年参数如图四、五示。

三、控制方案设计

据上述控制原则,系统设计方案主要内容:

(一)新风、回风混合比控制:考虑工作环境和运行经济性,冬夏季新风量应选取 $\pm 20\%$ 为宜。全年新风、回风量按图三变化,实现控制框图如图六。图中 T_{10} 、 ϕ_{10} 检测室外空气参数,按表二确定运行工况I~VII,六台风机的对应风伐由计算机同步控制。混合比控制算法见表三。

(二)各工作区参数底制:对于图四、五中参数控制,仅以底层为例,其控制框图见图七。

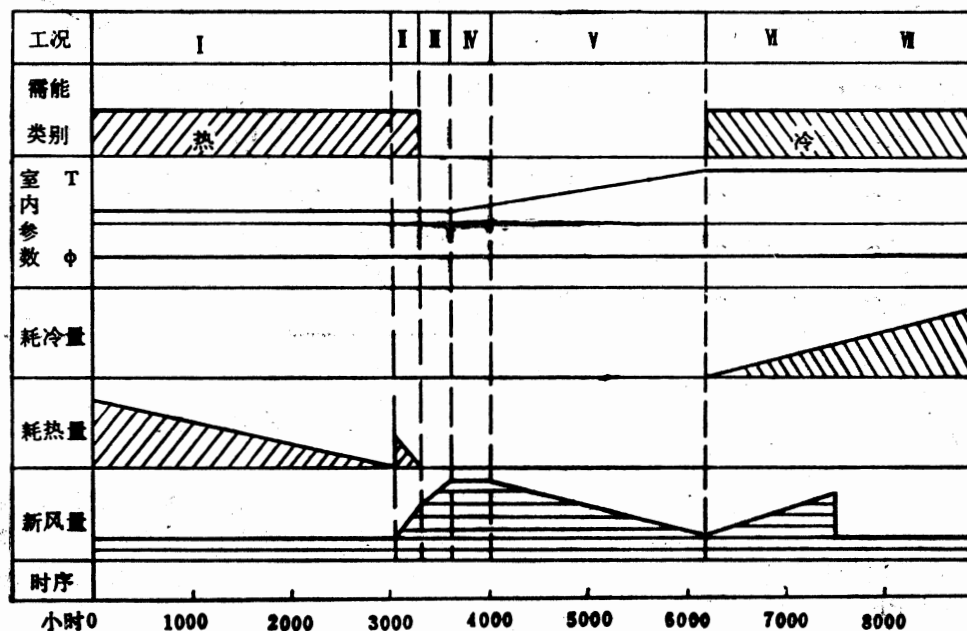
图中各风阀开度由图六新、回风混合比控制。车间温、湿度主要由加湿器 ϕ_k 、加热器 Q_k 、冷水阀MD来实现控制。其控制算法见表四,表中冬季送风量为夏季的70%,其它工作区略。

四、计算机系统设计

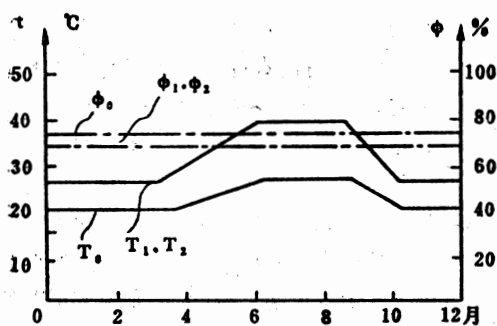
根据集散型控制原则,为实现上述控制要求,计算机系统设计框图如图八示。

(一)主微机系统:IBM PC/XT

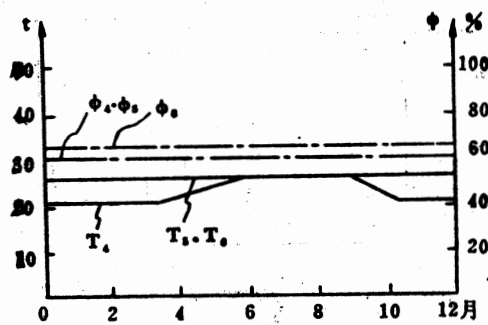
主微机有联机 and 脱机二种工作方式。在联机方式下,主机可对空调系统进行实时检测和控



图三 全年能耗图

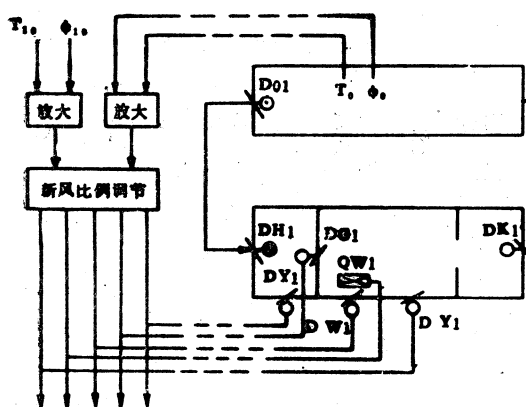


图四底层工作区参数

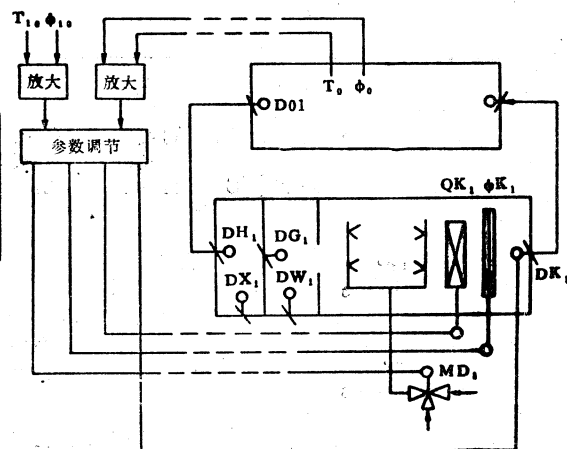


图五楼层工作区参数

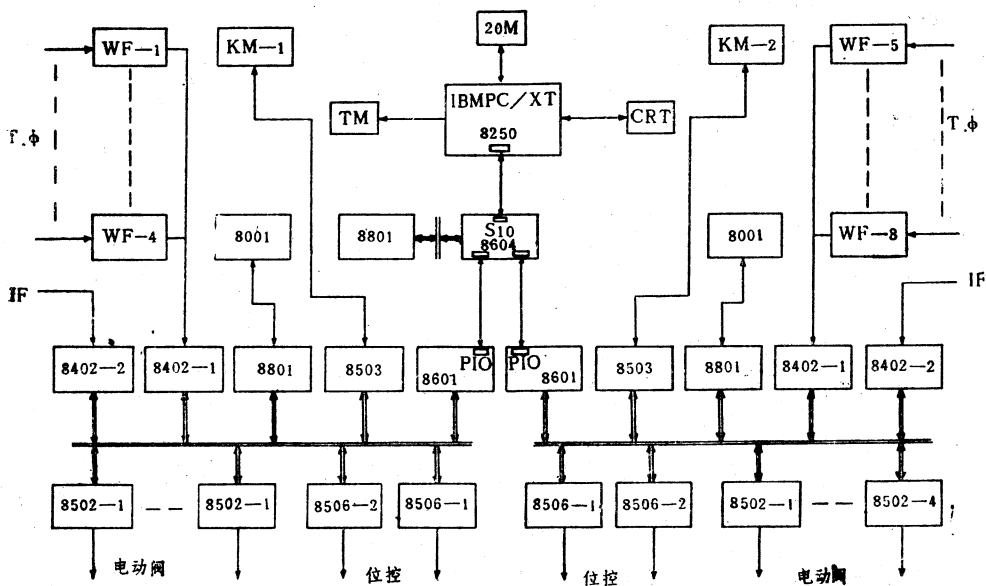
图中： ϕ_0 、 T_0 ——底层车间湿、温度参数； ϕ_1 、 T_1 ——贮叶房湿、温度参数； ϕ_2 、 T_2 ——贮梗房湿、温度参数； ϕ_4 、 T_4 ——楼层车间湿、温度参数； ϕ_5 、 T_5 ——进口机房湿、温度参数； ϕ_6 、 T_6 ——贮丝房湿、温度参数。



图六新、图风比控制



图七 温、湿度控制框图



图八系统逻辑框图

表二 全年工况自动转换条件

工 况		I	II	III	IV	V	VI	VII
特 征		干 冷	湿 冷	温度>I 湿度低	室内外相近	高温、高湿	高温、低湿	超高温
季 节		冬 季	春初秋末	春秋初末	春秋季	黄 霉 季	夏 季	盛 夏
条 件	T_{10}	$\leq 10^{\circ}\text{C}$	$10 \sim 25^{\circ}\text{C}$	$10 \sim 20^{\circ}\text{C}$	$20 \sim 28^{\circ}\text{C}$	$20 \approx 28^{\circ}\text{C}$	$28 < T_{10} < 30^{\circ}\text{C}$	$T_{10} \geq 36^{\circ}\text{C}$
	ϕ_{10}	$\geq 70\%$	$> 70\%$	$\leq 70\%$	$\leq 70\%$	$> 70\%$		

表三 底层车间混合风比例控制

工 况		I	II	III	IV	V	VI	VII
自动转换条件		$T_{10} \leq 10^{\circ}\text{C}$ 且 $\phi_{10} \leq 70\%$	$10^{\circ}\text{C} < T_{10} < 20^{\circ}\text{C}$ $\leq 70\%$	$10^{\circ}\text{C} < T_{10} < 20^{\circ}\text{C}$ $\leq 70\%$	$20^{\circ}\text{C} \leq T_{10} \leq 28^{\circ}\text{C}$ $\leq 70\%$	$20^{\circ}\text{C} < T_{10} < 28^{\circ}\text{C}$ $> 70\%$	$28^{\circ}\text{C} < T_{10} < 36^{\circ}\text{C}$	$T_{10} \geq 36^{\circ}\text{C}$
执 行 机 构 控 制 方 式	新 风 $DW_1 \sim DW_3$	20%开度	随室外湿球 温度增加而 增大开度	随室外温度 线性增大开 度	100%开度	随室外湿球 温度增加而 减小开度	随室外温度 减小至20% 开度	20%开度
	新 风 $DY_1 \sim DY_3$	关 闭	关 闭	随室外温度 线性增大开 度	100%开度	随室外湿度 增加而 减小开度	随室外温度 升高线性减 小开度	关 闭
	回 风 $DG_1 \sim DG_3$	100%开度	随室外湿球 温度增加而 减小开度	随室外温度 线性减小开 度	关 闭	随室外湿球 温度增加而 加大开度	随室外温度 线性增大开 度	100%开度
	排 风 $Dx_1 \sim Dx_3$	10%开度	随室外湿球 温度增加而 增大开度	随室外温度 线性增大开 度	100%开度	随室外湿球 温度增加而 减小开度	随室外温度 线性减小开 度	20%开度
	预 热 $QW_1 \sim QW_3$	加 热	加 热	不加热	不加热	不加热	不加热	不加热

表四 底层车间控制算法

方 式 工 况	Do和Dk	MD	Qk	ϕk
I II III	70%开度	若 ϕk 全关,室内相对湿度 $\geq 70\%$ 上偏差,MD按温度偏差控制,否则流循环水	室温偏差控制位式控制	湿度 $\geq 70\%$, ϕk 关闭; 湿度 $< 70\%$, ϕk 按湿度偏差控制
IV V	100%开度	工况IV, MD按室温偏差控制;工况V, MD按湿度偏差控制	冷却水=0,且室温低于给定值,室温受Qk控制	湿度 $< 70\%$, ϕk 按湿度偏差控制;湿度 $\geq 70\%$, ϕk 关闭
VI VII	100%开度	按室温偏差控制当MD $> 70\%$,转为按湿度偏差控制	湿度 $> 70\%$,且室温低于下阀值时启动Qk加热	湿度 $< 70\%$, ϕk 按湿度偏差控制;湿度 $\geq 70\%$, ϕk 关闭

制。可以显示和打印各工作区检测点参数值、设备运行工况、故障记录的月、年报表及运行曲线。在脱机方式下,主机可从事与运行系统无关的其它管理工作。

(二) 通讯机

通讯机由STD总线模板组成。由图八知:STD8604板的SIO芯片通过RS-232和主机的8250芯片进行双向串行通讯,为提高主机运行效率,通讯约定为主机命令方式。

通讯机的STD-8904板上有二片PIO芯片,分别和二台控制机的STD-8601板PIO芯片构成并行双向数据通讯,约定为中断方式。

为提高控制机的实时控制能力和主机利用率,通讯机对现场来的实时数据进行预处理和暂存,定期向主机传送。

(三) 控制机

二台控制机除控制软件有差异外,其硬件配置完全相同,各部分功能如下:

1. STD-8601板除完成并行通讯外,同时把控制机内存扩充到16K(12K ROM和4K RAM),板内24 I/O完成部分面板控制和操作。

2. 现场64路温湿度传感器信号经8块TC板放大送ST-8402板进行32路A/D转换,所测温度差经机内自动换算成相对湿度值。

3. 二块STD-8402板分别检测风阀开度值,作为反馈输入信号。

4. 设备运行工况监视和报警由STC-8503板32 I/O来完成。

5. STD-8502板完成所有电动阀、加湿器、冷水三通阀等控制;STD-8506板完成电机及电磁阀控制。

在系统设计中,为提高抗干扰能力,输入或

输出均加有二级光电隔离,控制信号经JTX功率继电器隔离输出。

五、软件设计

(一)主微机管理系统:有脱机和联机两种方式。

在联机工作方式,主机具有下列功能:

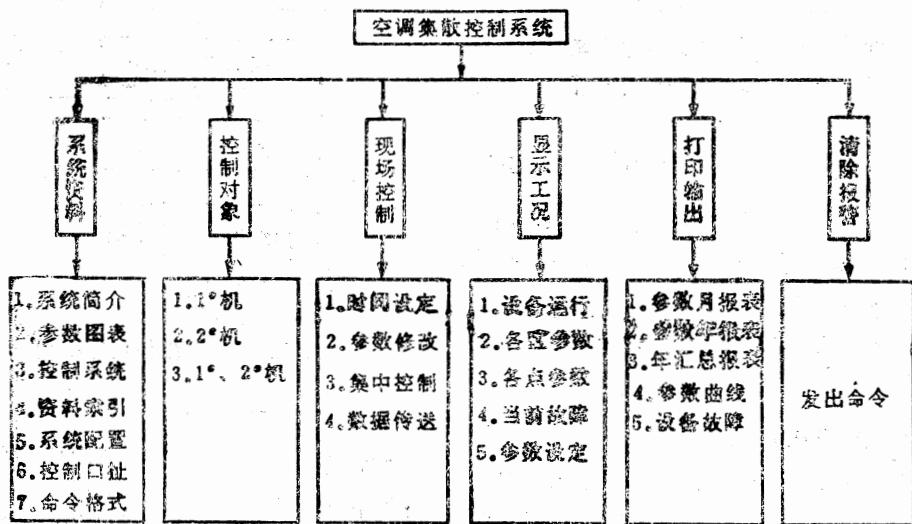
1. 系统资料查阅:系统性能及操作使用说明。

2. 系统监视:实时检测系统运行参数及设备工况。

3. 系统控制:系统运行状态的设定和改变。

4. 统计报表打印:系统运行参数及设备运行工况的月、年统计报表和曲线。

主机管理系统软件框图见图九。



图九 主机管理系统

(二) 通讯软件研制:

1. 串行通讯:对主机串行口8250芯片控制寄存器进行不同设置,使PC机能在不同文件模式下,对通讯机的SIO接口状态寄存器进行设定,实现串行通讯。

2. 并行通讯:通讯机和控制机PIO芯片经由40芯扁平电缆联接,构成并行双向数据通讯,提高数据传输速度。

(三) 控制软件研制:

1. 采样周期选择:经调试确定:

温、湿度采样周期:QT=20秒

阀门开度采样周期:QK=2秒

2. 数字滤波:

(1) 温、湿度参数:采用中值滤波法。

(2) 阀门开度参数:采用逆推平均滤波法。

$$\bar{y}(k) = \frac{1}{N} \sum_{i=0}^{N-1} y(k-i)$$

式中: $\bar{y}(k)$ ——第 k 次 N 项递推平均值

$y(k-i)$ ——往前推第 i 项测量值

N ——递推平均的项数, 取 $N=12$

3. PI控制算式: 空调系统中各类风阀, 加湿器, 冷水阀主要用于控制流量。而流量是个快变化参数, 选用PI调节, 其增量算式如下:

$$\Delta u(k) = K_c [y(k-1) - y(k)] + \frac{K_c Q}{T_i} [r - y(k)]$$

式中: K_c 、 T_i ——放大倍数和积分时间;

Q ——采样周期;

r ——给定值;

$y(k)$ ——第 k 次采样值。

4. 开关量时间比例算式: 偏差时间比例控制可减少开关量控制的波动。其原理是一个采样周期内, 正(或负)偏差大则阀门开启时间短(或长)。即

$$t = \frac{|e_i|}{e_N} Q \begin{cases} \text{开启时间 } t & \text{当 } e_i \leq 0 \text{ 时} \\ \text{关闭时间 } t & \text{当 } e_i > 0 \text{ 时} \end{cases}$$

式中: t ——阀门开启时间 Q ——采样周期

e_i ——实时偏差 e_N ——允许偏差

5. 控制软件框图

控制机控制软件总框图如图十

六、抗干扰措施

空调设备的大功率马达、各执行机构和强电电器等构成一个典型的干扰网。加上上千根从现场来的连线, 有的长达几百米。为确保系统可靠、稳定运行, 必须采取全面抗干扰措施。

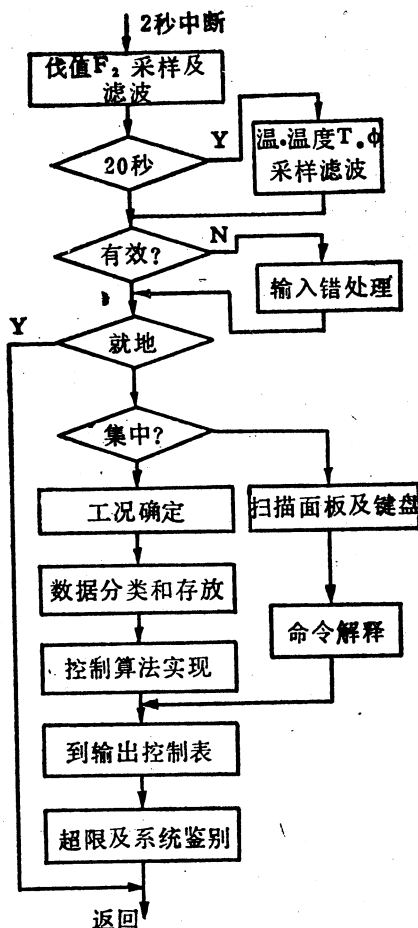
(一) 系统电源设计:

系统电源设计如图十一, 设计原则是主机、计算机控制系统经UPS电源接电网; 功率驱动电源和强电控制电源直接接电网。

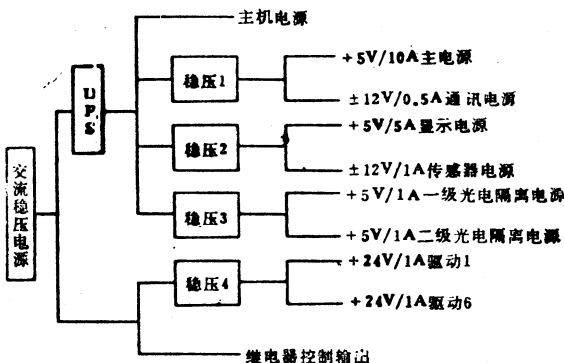
系统控制柜外壳经专用地线接地。图中主机、+5V主电源、+5V显示电源和±12V通讯电源悬浮共地, 其它电源地均单独悬浮不互联。

(二) 光电隔离措施:

选用的STD总线接口板到现场均采用二级



图十 控制程序框图



图十一 系统电源

光电隔离, 以保证系统在复杂环境下工作稳定可靠。

(三) 数字滤波: 温湿度参数采用中值滤波, 阀门开度采用递推滤波。实践证明其对提高

超长重载列车制动和缓解时间采集与处理

西南交通大学 隗永安

【摘要】 超长重载列车以150辆为一列，每辆车有制动和缓解信号各1，计有300个数字信号，由CMC-80单板机PIO口输入，以多路选择器74SL 150组成扩展接口，扩展输入信号到320个。一旦输入制动或缓解信号给出，看表（CMC—80之CTC。定时）计时，并按车号顺序存贮制动或缓解时间。最后以CMC—80之串行口（或并行口）与主机联成系统，作进一步数据处理。

一、引言

超长重载列车已经在我国铁路线上运行，为确保运输安全，提高运输效益，对超长重载列车各车辆制动或缓解设备动作时间的调整是至关重要的。本文介绍的制动和缓解时间采集和处理设备就是在车辆生产中为调整制动缓解机械动作时间而设计的。本机已在西南交通大学车辆实验室长期运行，效果良好。

二、硬件设计考虑

以单板机加扩展接口组成的智能数据采集设备，可以借单板机直接对其接口编程，独立地完成数据采集，暂存，计算和处理。并可通过接口与主机组成系统，缓解了外设对主机接口的占用。还可借单板机的显示器对工作过程进行监视。本机为开关量动作时间的采集，借助单板机现有的CTC定时组成时钟。以司机的制动缓解操作手柄给出操作命令作为时间起点，看表计时。单板机造价低兼，与扩展接口相联，简化了硬件设计，降低了造价。

CMC—80单板机有两个并行口，既PIO₁和PIO₂，较之其它单板机多一个PIO口，有利于接口扩展。有以Z—80 SIO组成的串行口，可以与主机串行口联机。CMC—80各端口总线均以印制板接插件的形式与外部设备相联，尤其适合于本机的设计。

抗干扰能力有明显效果。

（四）布线：机内走线原则：信号线和交流控制线垂直交叉分布布线；走线枋和转接插头分开，控制继电器线圈加反向二极管。现场来线分

三、设计方法及电路框图

设备框图如图3—1

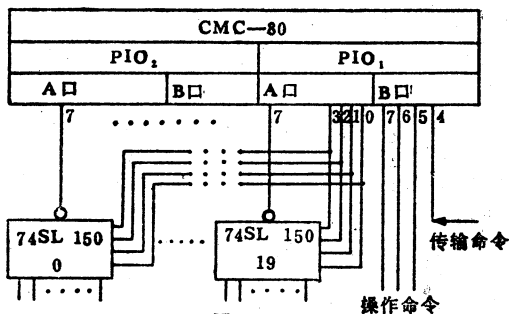


图3—1

PIO₂A口 位P_{A7}—P_{A0}

PIO₂B口 位P_{B7}—P_{B0} 计10位，制动信号输入。

PIO₂B口 位P_{B8}—P_{B0}

PIO₁A口 位P_{A7}—P_{A4} 计10位，缓解信号输入。

PIO的20位输入端，由20片多路选择器（74SL150）的输出供给。每片74SL150有信号输入端16（既16选1），由此PIO的20位输入端扩展成20×16=320个输入端，满足150辆车，每辆车制动缓解信号各1，计有300个输入信号的要求。

PIO₁A口位P_{A3}—P_{A0} 计4位，作为选择输出，与20片选择器的4位选择输入端并联。4位输入端的16种状态选择每片的16个输入信号到输出端，并加到PIO的输入口。选择表如下：

别穿管，信号线采用金属屏蔽线。

该系统于89年12月通过厂级验收，90年8月7日通过专家鉴定，系统设计先进，运行稳定，抗干扰能力强，年节能效益为30~40%。

用软件方法查找硬盘适配卡故障

长沙铁路分局娄底机务段微机室 梁军

在IBM-PC/XT及其各类兼容机中,每当进行冷启动时,都有一个上电自检程序对整个系统的主要部件进行测试,如果硬盘系统出现故障则是显示1701,但我们无从知道是何处发生了故障。笔者通过对上电自检程序进行了分析,发现在以下几个方面有错时都会出现1701:复位控制器、适配器RAM、适配器内部、驱动器准备好否、驱动器重校准等,其中前面三部分是适配卡出故障,后面二部分是硬盘有关磁道出故障或控制线路出故障,因此笔者就用8088汇编语言编写了一个类似的测试程序,当出现1701时,无论是硬盘系统何处有故障都可以迅速查出来。同时本程序还具有一些测试硬盘适配卡的基本功能,如寻道、读数据和写数据,并且还具有重复操作的功能,对一般的维修人员就可以根据有关提示对有关的器件测试出有关数据,从而就可以提高修复的效率。为了避免损失硬盘上的数据,笔者采用了读和写都在一个磁道上进行,把读取的数据放在buffer中,写操作时再重复进去。

该程序已在IBM-PC/XT及其兼容机上调试通过。

```
data segment
yy0 db '现在开始检查硬盘和适配器1701错误发生处',15q,12q,'$'
xx1 db '复位控制器有故障.....',15q,12q,'$'
aa1 db '复位控制器无故障.....',15q,12q,'$'
xx2 db '控制器RAM有故障.....',15q,12q,'$'
aa2 db '控制器RAM无故障.....',15q,12q,'$'
xx3 db '控制器内部有故障.....',15q,12q,'$'
aa3 db '控制器内部无故障.....',15q,12q,'$'
xx4 db '驱动器没有准备好.....',15q,12q,'$'
aa4 db '驱动器已经准备好.....',15q,12q,'$'
xx5 db '驱动器重校准失败.....',15q,12q,'$'
aa5 db '驱动器重校准成功.....',15q,12q,'$'
yy6 db '硬盘和适配器没有1701错误.....',15q,12q,'$'
yy7 db '硬盘有1701错误,可能为0磁道损坏.....',15q,12q,'$'
xx8 db '硬盘寻道失败.....',15q,12q,'$'
aa8 db '硬盘寻道成功.....',15q,12q,'$'
xx9 db '硬盘读数据失败.....',15q,12q,'$'
aa9 db '硬盘读数据成功.....',15q,12q,'$'
xx10 db '硬盘写数据失败.....',15q,12q,'$'
yy10 db '硬盘写数据可能会丢失数据,你是否真要写入(Y/N)?$'
aa10 db '硬盘写数据成功.....',15q,12q,'$'
err1 db '硬盘有问题.....',15q,12q,'$'
con db '连续测试请按Y(或y):$'
orlf db 15q,12q,'$'
```

```
buffer db 512 dup(?)
mz1 dw ?
mz2 dw ?
data ends
stack segment para stack 'stack'
stapn dw 50 dup (?)
top equ length stapn
stack ends
code segment
start proc far
assume cs:code,ss:stack,ds:data,es:data
org 100h
push ds
sub ax,ax
push ax
cli
mov ax,data
mov ds,ax
mov es,ax
mov ax,stack
mov ss,ax
mov ax,top
mov sp,ax
call blank
mov dx,offset yy0
mov ah,9h
int 21h
mov cx,0025h
mov mz2,0
test1:call reset0 ;测试复位控制器
jnc a0
loop test1
mov dx,offset xx1
mov ah,9h
int 21h
mov mz2,1
jmp test2
z0:mov dx,offset aa1
mov ah,9h
int 21h
test2:mov cx,1 ;测试控制器RAM
mov dx,00h
mov ax,1200h
int 13h
jnc b0
mov dx,offset xx2
mov ah,9h
int 21h
mov mz2,1
jmp test3
b0:mov dx,offset aa2
mov ah,09h
int 21h
test3:mov cx,1 ;测试控制器内部
mov dx,00h
mov ax,1400h
int 13h
```

```

jnc c0
mov dx,offset xx3
mov ah,09h
int 21h
mov mz2,1
jmp test4
c0:mov dx,offset aa3
mov ah,09h
int 21h
test4:mov mz1,410dh ;测试驱动器准备好否
d0:in al,021h
and al,0feh
out 021h,al
d1:call reset0
jc d3
mov cx,1
mov dx,80h
mov ax,1000h
int 13h
jnc d4
d3:mov ax,mz1
cmp ax,446dh
jb d1
jmp test6
d4:
mov dx,offset aa4
mov ah,9h
int 21h
jmp test5
test5:mov cx,1 ;重校准
mov dx,80h
mov ax,1100h
int 13h
jc e0
mov ax,0900h
int 13h
jc e0
mov dx,offset aa5
mov ah,09h
int 21h
jmp disk
a0:mov dx,offset xx5
mov ah,09h
int 21h
mov mz2,1
jmp disk
test6:mov dx,offset xx4 ;测寻道
mov ah,9h
int 21h
mov dx,offset yy7
mov ah,9h
int 21h
mov mz2,2
disk:mov ah,0ch
mov dh,0
mov dl,80h
mov ch,61h
mov cl,11h
int 13h

jnc d6
mov dx,offset xx8
mov ah,9h
int 21h
mov mz2,2
jmp test7
d6:mov dx,offset aa8
mov ah,9h
int 21h
test7:push ds ;测读数据
mov ah,02h
mov al,01h
mov bx,offset buffer
mov ch,61h
mov cl,11h
mov dh,0
mov dl,80h
int 13h
test ah,0ffh
jz d7
mov dx,offset xx9
mov ah,9h
int 21h
mov mz2,2
jmp test8
d7:mov dx,offset aa9
mov ah,9h
int 21h
test8:mov dx,offset yy10 ;测写数据
mov ah,09h
int 21h
mov al,0
mov ah,1
int 21h
push ax
mov dx,offset crlf
mov ah,09h
int 21h
pop ax
cmp al,59h
jz al
cmp al,79h
jnz el
a1:mov ah,03h
mov al,01h
mov bx,offset buffer
mov ch,61h
mov cl,11h
mov dh,0
mov dl,80h
int 13h
pop ds
test ah,0ffh

jz d8
mov mz2,2
mov dx,offset xx10
mov ah,9h
int 21h
jmp el
d8:mov dx,offset aa10
mov ah,9h
int 21h
succ:mov dx,offset yy6
mov ah,09h
int 21h
jmp exit
el:stc
mov ax,0
cmp ax,mz2
jz succ
mov ax,2
cmp ax,mz2
jnz exit
mov dx,offset err1
mov ah,9h
int 21h
exit:mov dx,offset con
mov ah,09h
int 21h
mov al,0
mov ah,1
int 21h
cmp al,59h
jz loop
cmp al,79h
jz loop
ret
loop:jmp start
start endp
blank proc near
mov cx,0
mov dx,184fh
mov bh,7
mov ax,600h
int 10h
mov dx,0
mov bh,0
mov ah,2
int 10h
ret
blank endp
port0 proc near
mov dx,320h
push ax
sub ah,ah
mov al,0

```

PC/XT 故障 维修 两 例

——兼附：M2024打印机维修一例

湘潭大学计算中心 戴 永

摘要：例1. 介绍开机无反映现象的一种鲜为人知的RAM等原因及其诊断处理方法。

例2. 介绍主机电源+5V电压偏离输出规格造成的一种现象及其分析方法。

例3. 介绍主机使用打印机而死机的一种打印机本身接口电路故障及其维修方法。

文中PC/XT均指以8088为CPU的个人微机。

例1. 故障现象：开机后电源工作正常，主机无反映。

分析与维修：开机屏幕不给出任何信息，对此用户是很头痛的，常有不知所措之感。造成这种现象的原因很多。针对这种现象，我们可以先采用最小系统法，初步判别引起该现象的故障是在主机板上还是在接口卡或有关外设中。本例故障发生在主机板上。

目前PC/XT主机板上CPU、ROM及RAM芯片一般采用插座连接，它们出故障比起锡焊芯片要容易处理得多。而这几种芯片损坏都可导致至开机无反映现象。三种芯片损坏后的情景分析如下：CPU无需多述；BIOS ROM损坏（包括ROM芯片坏，BIOS程序坏），则BIOS程序不能运行，显然将出现上述现象；若第一排RAM有芯片损坏，也会产生开机无反映情形。PC/XT常以\$0040H~\$004FFH为BIOS工作单元，或者说BIOS工作单元多使用第一排RAM。因

而当此段存储单元出故障，BIOS同样无法运行。以上分析表明，遇到所述现象，如果断定毛病在主机板上，依据“先简后繁”的故障消除原则，此三种芯片可作为最先考虑的故障怀疑对象。

该例由后两种原因所至。我单位有数十台PC/XT（CB0520B、IBM-PC/XT等），其中有两台为BIOS ROM坏，有四台是第一排RAM坏。故障诊断排除方法采用芯片替代法。可以先更换CPU、BIOS ROM，后掉换RAM，反之也行。若只有一台，可按反之顺序工作。没有备用RAM芯片的，可在另外几排RAM中取下同容量芯片，与第一排RAM进行替换诊断。

例2. 故障现象：开机时好时坏。非正常时，屏幕显示自检内容正确，但歪曲、晃动、滚动，同时软驱小车定位不稳、振动、发出“嘟嘟”的声响。

分析与维修：自检内容显示正确说明开机时主机板工作正常。从电路结构来说，显示器

```

add dx,ax      16:call port1
pop ax         out dx,al
ret           call port1
port0 endp     in al,dx
port1 proc near and al,2
call port0    jz r3
inc dx        loop 16
ret           stc
port1 endp     r3:pop dx
reset0 proc near pop cx
push cx       ret
push dx       reset0 endp
clc           code ends
mov cx,0100h  end start

```

（上接56页）

16. 2 到16进制之间任二种数制的相互转换。
17. 汉字的横向打印，如同古装书籍。
18. 用户欢迎程序（HELLO），它可直接删除加锁文件。
19. DOS出错信息的汉化程序。
20. 主菜单程序及说明书等。

需《中华学习机系统盘》的用户，请同本刊联系。

与软驱没有直接联系,而是通过主机协调工作,所以在硬件上二者不会互相影响。现它们同时工作不正常,可怀疑主机电源出毛病。特别是软驱内有一个步进电机和一个磁盘驱动电机,其对电源的质量要求较高,因此软驱出现上述现象更应该查一查电源的好坏。用万用表测量主机电源电压输出,测得+5V输出电压在+4V~+5V范围内变化。当开机好时,+5V电源输出电压为+4.7V~+5V;当开机出现所述故障现象时,+5V电源电压输出为+4V,明显低于输出规格(+5V输出规格:额定+5V 最小+4.8V 最大+5.25V)。调换电源实验,机器完全恢复正常工作。

例3. 故障现象:打印机自检正常,主机内打印接口卡工作正常。但打印机与主机联用时,主机执行任何与打印机有关的操作命令,打印机响应之后主机死机。

分析与维修:自检正常,接口卡好,由此可以断定该故障发生在打印机内部对外的接口电路上。M2024对外接口采用标准36线并行口,与主机连通的控制、状态讯号有9个,分别是:

①STROBE(数据选通)、②ACKNLG(回答)、③BUSY(忙)、④PNE(纸尽)、⑤SLCT(选中)、⑥INPUTPRIME(输入初始化)、⑦FAULT(故障)、⑧LD(小车状态)、⑨EXPRIME(外部初始化)。其中①、⑥、⑨是主机送往打印机的控制讯号,另外的均为打印机送往主机的状态信息。主机需打印时,通过⑥告知打印机的CPU,然后不断地访问③。当③由高变低后,发出②,主机接收②,便将欲打印的内容送入打印机。③的变化与

除②外的其它状态信号有关。打印机有响应能力,证明对应于⑥的电路是好的。如果③恒定于高电平,则主机将总处在访问③的运行状态,进而造成死机现象。至此我们可以认为毛病出在对应于③及其相关状态讯号的电路上。

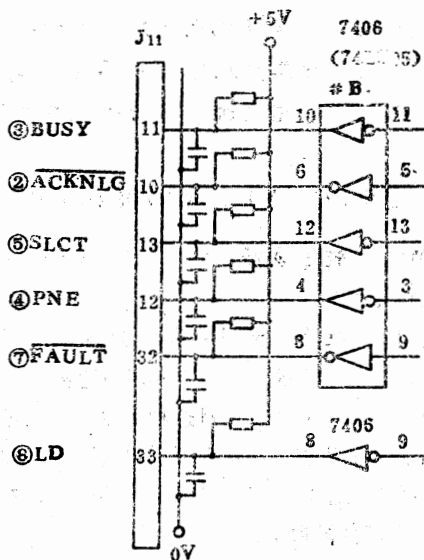


图1

图1所示为M2024状态讯号输出电路的一部分。将打印机脱离主机,置于ON LINE状态。正常情况下应有②=⑦=1,③=④=⑤=⑧=0。用万用表检测J11(36线标准插座),查得第11、12线插座电位为非正常状态,即③=④=1。打开机箱进一步检查#B₁,发现V_{B1-3}=V_{B1-4}, V_{B1-10}=V_{B1-11},无论#B₁之外的电路如何,就其逻辑功能而言#B₁已损坏。掉换#B₁机器恢复正常。图1中#B₁的括弧型号为新换上的芯片型号。

PC—1500 两套供电系统

PC—1500系列机与CE—150打印机配接口后,可用EA—150电源转换器的9伏小插头插入CE—1500接口(ADAPTOR)箱镍镉电池充电,但不是对主机的电池仓内的五号干电池充电;如单独使用PC—1500系列机,EA—150的小插头可直接插入主机的电源接口(ADAPTOR)供电。若15天左右不使用打印机,有可能按[ON]键开机时,夹笔器转动到极左端被卡住而不到位,屏显示“CHECK: 6”的提示,如多次强行复位都失效,这时误认为是打印机系统或齿轮的问题。一旦了解到主机是由两套供电系统组成的话,即可关机充电若干小时,再开机时CE—150系统就可恢复正常。

《中华学习机系统盘》(CEC-DOS) 简介

郑州市技工学校 邵爱柏

使用《中华学习机系统盘》(CEC-DOS) 是中华学习机(CEC-I) 用户最明智的选择。她比苹果机系统盘DOS 3.3要好得多。其特点有: 兼容性好, 调文件速度快, 磁盘容量大, DOS功能全, 系统盘工具文件多, 能进行加密, 汉化程度高, 操作简单方便等。

一、系统功能改造与扩充

1. 格式化磁道总数为40轨, 如果用户驱动器质量好, 尚可利用《EDIT DOS》文件格式化42轨。
2. 调文件速度平均比DOS 3.3快5倍。
3. 扩充了RUN命令功能, 欢迎程序可使用: A、B、T档文件(无需用户修改DOS内容)。
4. 改造了列目录功能: 显示文件名中控制字符, 列长目录时随时退出等。
5. 为CEC-1增加了功能键(不占用用户内存):

&H=HOME, &C=CATALOG, &L=LIST, &R=RUN, &J=CALL-151, &I=INVERSE, &N=NORMAL, &T=TRACE, &U=NOTRACE, &M=MUSIC, &Sn(将十进制数n转换成十六进制数), &\$n(将十六进制数n转换成十进制数)

6. 增加BASIC程序连接功能: CALL 46771 保护和隐藏内存中程序, CALL 46821 连接和恢复内存程序。
7. 有GOTO N(N为变量行号)功能。
8. 有READ指定的变量行号所决定的DATA语句行数据功能。
9. 有程序加密功能: 封锁CTRL-RESET和CTRL-C。
10. 有磁盘加密功能: 格式化加密子盘, 防止非法拷贝。
11. 有恢复内存中被NEW掉的BASIC程序功能。
12. 提供自动行号功能。
13. 文件名允许使用除数字开头的任意字符。

14. DOS命令允许使用小写字母。

15. 有缩写文件名功能(如FID)。

二、系统盘文件功能

1. 除保留了经过改造的DOS 3.3工具文件FID(文件拷贝)、CHAIN(程序间数据传递)、RENUMBER(更改行号和程序连接)等外, 新增20多个工具性、游戏性实用文件。
2. 40道快速拷贝(DISK MUNCHER), 有简单解密功能。
3. 磁盘医生(DISK DOCTOR), 有打印、修改扇区资料及恢复被删文件名和删除DOS功能。
4. 修改DOS(EDIT DOS), 有更改格式化磁道总数, 更改DOS命令名, 更改目录磁道、目录保护, RESET自举等功能。
5. 能格式化STC 2.0软汉字系统(CEC-DOS 盘背面录有STC 2.0字库)。
6. B档文件自动改址(MOVE)。
7. 各种汉字系统程序的相互转换。(CEC-STC, HZ-CEC等)
8. 图形显示技巧, 图形压缩存贮与恢复。
9. 图形加字、图形打印。
10. 汉字显示、打印技巧, 可制作软件封面字或打印各种艺术字。
11. 音乐与电脑琴。
12. 打印CEC-I区位码字符、命令及其入口地址。
13. 打印袖珍万年历。
14. 打印人体生物节律。
15. 打印青少年性格测试结果。(下转54页)

千里之行,始于足下

——SMD—6410系统捷足先登

SMD-6410三维制鞋辅助测量辅助设计系统是为制鞋业应用计算机辅助鞋楦测量、辅助帮底样设计开发的实用系统。该系统由主机(XT、AT/286、386)、高分辨率图形显示系统、三维楦面测量仪、平面数字化仪、XY绘图仪及软件系统构成,其提供的功能覆盖了制鞋设计全过程各个阶段,是国内功能最完整、最实用、性能价格比最优良的一套系统,适用于制鞋生产、科研及教学部门。

该系统属国内首创,达到了国际八十年代同类产品水平。

该系统完成如下功能:三维楦面测量、计算特征数据、全楦面展平、平面或立体款式设计、样片分解、放帮帮裕度、放加工余量、标注分针眼、帮样扩缩、底样设计、底样扩缩、成鞋彩色造型、楦面样片展平、优化排料、样片面积计算、料片利用率计算、设计数据的数据库维护。

系统具有优良的技术指标:

1. 楦面数据测量精度达0.1mm,测量点二万余个
2. 设计数据计算精度达0.1mm
3. 帮底样多级扩缩精度超过手工,完全符合部颁等差标准,效率超过人工五十余倍
4. 全楦的展平图符楦效果好
5. 辅助设计效率是人工设计的20~40倍
6. 设计工时节省80%以上
7. 单位排料利用率提高5%以上
8. 样鞋试制量可节省70%以上

这套系统的投入运用,可以创造极大的经济价值并带来极大的社会效益。从国外引进同类系统,一般达数十万美元一套,而SMD-6410不同配置在几万元到十几万元人民币之间,具有良好的性能价格比。(马在强、刘营)

《Z80汇编语言BCD码长浮点常用函数运算符程序库》简介

该子程序库是为各种Z80系列的单板机进行数学运算和数据处理而编写的。全部程序占2K字节内存,均固化于一片2716EPROM芯片内。

该子程序库采用模块化程序设计方法,层次清晰,各种功能均为子程序,调用方便,易于功能扩展、程序的数

据运算和数据的输入输出均采用BCD码(4bt表示1位十进制数)的数据方式。

该子程序库中的操作数和计算结果均采用九个字节表示,其中第一个字节为阶码和数符,其余八个字节为尾数。ZLP主要内容:

四则运算,开方,平均值,标准差,三角函数,对数,线性回归,方差,指数函数,双曲函数,幂级数,大小排列,递推法求平均值和方差等。

销售价格:

ZLP的EPROM 2716/2732 固化芯片一片及使用说明书一份 100.00元

ZLP的汇编语言原程序清单一份 50.00元

ZLP的加法子程序分析一份 30.00元

通讯地址:北京复兴门内大街45号 商业部信息中心

邮政编码:100801

联系人: 张铁钢

电话: 601.2893 66.8581--2893

《计算机应用研究》第7卷 (1990年度)

总 目 录

软 件 篇

用C程序调用中的参量传递	贾洪钧 (1.1)
用BASIC语言直接读取DBASE—Ⅲ数据库数据的方法	唐珍等 (1.4)
负荷传感器的动态测试	郑国基等 (1.6)
单回路智能仪表应用程序设计	阎立恒等 (1.8)
单纯形加速法与计算机控制系统的PID最优化仿真设计	李天健 (1.10)
传输控制及网络管理	张鸿鸣等 (1.14)
微机医疗专家系统	张鸿鸣等 (1.16)
高精度多字体显示器画面构造程序	李卫元 (1.18)
一个简易画空、实心框图程序	龙兵生 (1.21)
能够在微机上快速实现CRC校验的一种方法	袁兴志等 (1.23)
中心提升——从数据流图到模块结构图的直接转换方法	王治宇 (1.26)
一个微机“病毒”的感染、激活原理及其防治办法	马小宏 (1.32)
计算机圆点“病毒”的诊断及排除	王庆庆 (1.34)
字库编减及造字	雷学武 (1.36)
浅谈利用高级字库3KTZ 打印特大汉字	龙兵生 (1.37)
关于《利用BASIC产生屏幕美术字的编程技巧》之我见	李辉宏 (1.38)
丰富3070的打印机实践打印	魏旭 (1.39)
SAP6软件顾问系统概述	林丕等 (1.40)
代数多项式的Horner—差商混合算法	徐继锋 (2.1)
数据库设计最小复盖法	党正科 (2.2)
一个不易破解的文件名加密法	瞿波 (2.5)
窗口软件的编写	刘宇菁 (2.6)
一个用户友好的文件装载窗口	唐常杰 (2.7)
自动生成dBASEⅢ菜单源程序的方法	周步祥 (2.12)
汉字下拉式菜单	刘宇菁 (2.15)
用BASIC语言实现全屏幕数字编辑	段宝珠 (2.16)
屏幕划到与翻页技术	王渤 (2.18)
交互式任意大小屏幕图形的存贮与再现	李晖 (2.20)
大型汉字配上倒影的初步实验	夏国华 (2.23)
最优箭线网络图的绘制算法	周明 (2.25)
并行处理语言Occam及其在图象处理中的应用	诸昌铃 (2.26)
Turbo BASIC语言与汇编程序的连接方法	辛华 李谊瑞 (2.22)
快速高效编制报表的简易方法	方荣耀 (2.33)
MIGHTYPRESS打印接口卡特殊用法	唐汉雄 (2.36)
居民身份证底卡的微机打印程序	苏亚华 (2.40)
一个简单的标准文稿格式打印程序	周明 (3.1)
灵活的文件打印程序	刘建福 (3.2)
一个实用的方格纸格式文稿打印程序	徐雪斌 (3.5)
硬盘分区与DOS版本	彭禾 (3.10)
DEBUG对硬盘分区的浅析	陈一侃 (3.12)
保护硬盘的DOS外部命令	王志中 (3.14)
逻辑硬盘“丢失”的预防及处理	彭禾 (3.14)
再谈多种操作系统共享硬盘的方法	徐雪斌 (3.15)
抢占硬盘空间的有效方法	茹建平 (3.16)
长城0520-CH高中分辨率汉字系统同从硬盘自举的新方法	刘立强 (3.18)

提高IBM PC/XT硬盘空间利用效率的一种有效方法	许传国 (3.19)
硬盘保护的一种方法	宋俊山 (3.21)
用BASIC语言调用汇编子程序的方法修复硬盘初始引导扇区	肖浩廷 郭松亮 (3.22)
程序动态加密	刘士军 (3.24)
在绘图机上绘写汉字的新方法	胡培民 (3.26)
道路选择辅助决策系统的功能与设计	王雄基 曹光初 (3.29)
一个自动生成表格的FoxBASE程序	樊金生 (3.31)
介绍一例简易实用的键控绘图程序	瞿波 (3.33)
单缝衍射实验的计算机模拟分析	李志伟 (3.34)
APPLE—II上超长程序动态调试的实现	李勇 (3.36)
软件汉化及一个辅助工具	杨传斌 (3.37)
农业科技管理数据库系统绘图软件	唐珍等 (4.1)
黑盒法调试程序	郭继展 (4.2)
适用于微机文件的通用加密程序	段凡丁 (4.5)
单片机实时控制软件抗干扰编程方法的探讨	周鸣章 (4.7)
从根本上解决EPSON LQ—系列打印机的控制问题	杨森等 (4.9)
不用图形卡在单显上作图的方法	朱庆生 (4.11)
键控电路图的绘制	方斌 (4.14)
IBM-PC/XT兼容机软盘故障诊断实用程序的设计	王耀南 (4.15)
C-DBASE II 的有趣屏幕设计技巧	许伟彬 (4.18)
英汉电子小词典	陈阿林 (4.19)
用PC FORTRAN实现的下拉式菜单	黄心湖 (5.1)
Turbo Pascal 4.0下图形驱动程序和字体文件的连接技术	李志伟 (5.3)
IBM-PC C语言复盖式程序设计方法	徐涛 宋成刚 (5.6)
“停电宝”——上机过程记录程序	杨宁 (5.8)
一个用于汉字字幕的图形编程程序	张昶 (5.10)
最少操作通用报表系统	瓮正科 (5.13)
IBM Token—Ring网的结构特点	殷巍 李雄飞 陈恩国 (5.18)
透明环境的特征	陈庆章 (5.18)
微电脑人体心肌梗塞预测方法	张鸿鸣 田卫星 刘翠华 胡庆贤 (5.20)
计算机在粒度分析数据处理中的应用	马晓东 张亚玲 (5.22)
IBM-PC上的程序流程图编辑/打印软件	肖俊良 (5.26)
PROLOG语言简介	李景贵 (5.27)
数据处理语言的数据流模型	叶高英 (5.29)
Turbo Prolog 程序设计技巧	李春葆 章启俊 马玉枫 (6.1)
TURBO PASCAL 程序设计中几个难题的最新解决方法	唐福益 (6.4)
Turbo Pascal 图形数组及应用	沈国坤 左尧凤 杨丽 (6.8)
用高级语言编制过程控制软件	阎建明 (6.12)
利用软件黑盒子对PC机文本文件加密的原理与方法	于功第 路枝 (6.14)
硬盘加锁技术及实现方法	王晓武 (6.17)
崭新的FOXBASE+2.10和它的开发工具	杨剑波 西捷 (6.19)
dBASE III—PLUS下的窗口操作的设计	周步祥 阮树骅 (6.22)
在数据库中如何快速处理多层次合计的表格	蒋仁兴 (6.24)
防断电可中断的登帐程序设计	杨传斌 (6.25)
“地震活动信息熵与分维”研究的软件设计	梁宇清 周仕勇 (6.26)
一个多机环境下的集成CAD系统	张小平 巫晓龙 (6.30)
印制电路板图形的分块输出	方旭明 徐锐 (6.32)
电视动态字幕的微机实现	李志伟 (6.35)

硬 件 篇

单回路智能仪表硬件系统设计	阎立恒等 (1.42)
APPLE IIe微机体感诱发电位检测系统硬件接口设计	彭利安等 (1.45)
一种实时中断键盘接口电路	何杰 (1.46)
IBM-PC机并行口简易扩展器	韩知思等 (1.49)

计算器应用一例	华金保 (1.51)
一个实用的微机共享存储器通讯接口	刘宏生 (2.42)
气功信息模拟方法	吴居雄等 (2.43)
体感诱发电位检测系统的设计与应用	彭利安等 (2.46)
Z80微机系统中大量扩展 I/O 接口和存储器的巧妙方法	方旭明 (2.50)
如何用2716 EPROM代替16FDC板中的 74918 ROM	叶富乐 (3.39)
一种宽范围模——数转换与显示方法	李天健 (4.21)
紧急情况磁头缩回自动保护电路	陈兰英 (4.23)
IBM-PC总线扩展器的设计	王砚方 庞文宁 (5.32)
Mc68488和Intel 8048实现 IEEE488接口	吴剑勇 (5.36)
高抗干扰精密V/F式A/D转换芯片与APPLE II微机接口技术	杨兆庆 (5.37)
MCS-51单片机多外中断扩展方法	骆德汉 (6.36)
PC-1500两套供电系统	曹来发 (6.55)

系 统 篇

APPLE II内外存储扩充技术和提高运行速度方法研究及实例分析	沈欣等 (1.58)
多路数据采集器与微机的软硬件接口	陈启录 (1.58)
微机在小氮肥厂中的应用综述	彭献彬 (1.64)
E2PROM 2864的使用体会	张向春 (1.69)
实现MC-51单片机单步操作的简易方法	周鸣争 (1.70)
六十五吨玻璃窑炉微机控制系统	阎立恒 刘曙光 (2.52)
多级系统中IBM-PC/XT和STD总线通讯	周树杰等 (2.57)
计算机决策支持系统在铁路站场中的应用	朱怀芳等 (3.41)
一种简单易行的多机互连技术	龙卫红 (3.48)
微机控制X光测厚仪	胡士甬 (3.51)
数据通信网络系统设计及技巧	张鸿鸣 (4.35)
机械设计专家系统工具的结构与设计	殷国富等 (4.39)
西门子计算机IOP系统诊断与分析	周 林 (4.46)
工业炉窑智能仪表控制系统	阎立恒 (4.51)
TP801简易温度测量和控制系统	曹文方等 (4.55)
地震预报专家系统 (ESEP/PC)	庄昆元等 (4.58)
磁盘子系统诊断分析与调整	周 林 (4.61)
单片机两种启动方式的应用	荆子今 (4.87)
不间断电源的原理特性和选择方法	李长喜 (4.69)
DRNEST 专家系统构造工具的设计与实现	杨润生 张小衡 (5.39)
单片机中断源和中断优先级的扩展	张友生 (5.43)
长途自动交换机微机计费系统	郝国基 徐世坤 姚学英 (5.46)
热水瓶温度检验过程中的自动控制	张晓云 王水清 (5.48)
高速数字信号单片机处理机——TMS320C25 硬软件及应用	汪亚南 (5.52)
微机“五防”系统	陈骏林 王 涌 李林琪 (5.56)
甘肃省省级科技项目与科技成果管理系统	吕兴寿 吕林霞 雪志军 (5.58)
贵州省科技管理信息系统数据库的设计与实现	贾洪钧 (6.39)
集散型空调计算机控制系统	叶定治 李淑莲 (6.43)
超长重载列车制动和缓解时间采集与处理	魏永安 (6.50)

维 修 篇

外设常见故障与排除	陈兰英 (1.52)
使用FTU调整磁头的方法	陈兰英 (1.53)
0520CH—II 计算机电源故障一例	华金保 (1.54)
GW0520—CH II 硬盘维修一例	叶志斌 (1.55)
3070打印机打印针磨损的处理方法	周维寿 (1.56)
3070打印机断针的简易修复	苏亚华 (1.56)
用注射法使PP40打印机笔更新	沈友敏 (1.57)

VAX计算机接口板DHV11的故障分析	高济平 蔡新光 (2.59)
IBM-PC/XT 软盘驱动器故障两例	赵明生 (2.61)
IBM-PC微型计算机及兼容机故障检修	王 锋 (2.62)
3070打印机断针定位方法及8088汇编程序	龙兵生 (3.56)
IBM PC/XT硬盘维修及00磁道的调整	陈永青 (3.58)
长城 0520-CH 硬盘故障两例	梁 军 (3.59)
RAM 模块地址线和数据线固定故障的诊断	马永强 (3.60)
打印机断针自动测试程序	叶志斌 (3.65)
FX-100打印机不走纸故障检修一例	张学云 (3.66)
IBM-PC/XT 主机故障快速判断检修	李志刚 (4.24)
IBM微型机彩色适配器的维修	王 威 (4.25)
PC机的维护保养	龚 欣 (4.26)
SUPER AT系统维修实例	林康敏 (4.29)
中华学习机故障基本检测法	任 强 (4.32)
IBM-PC/XT及其兼容机系统板维修方法探讨	杨剑波 郭永莉 (5.60)
IBM-PC/XT微机死机故障维修一例	邵发森 (5.63)
Super-CH显示器扫描电路的代换与维修	刘亮生 (5.64)
2200VS系统死锁故障之一的排除	张晓云 (5.66)
硬盘维护的良好工具—File-Cleaner	刘里鹏 (5.67)
用软件方法查找硬盘适配卡故障	梁 军 (6.52)
PC/XT 故障维修两例	戴 永 (6.54)
测试异步通讯适配器	冉春林 (6.23)

信 息 篇

人民银行卫星通信网络加紧施工	桑小军等 (1.71)
我国用微机控制电力系统取得突破——南京地调自动化系统通过部级验收	徐 洁 (1.72)
仓库基础数据管理系统通过部级鉴定	阮高华 (1.72)
安徽省计算中心部分科研成果简介	顾福良 (1.72)
信息五则	(1.9, 44, 50, 封三)
消息十则	(2.封二, 13, 39, 64, 65)
IBM PC/XT及其兼容机故障诊断仪问世	李相彬 (3.33)
电力工程建设信息管理系统制成	李相彬 (3.40)
介绍一个围棋软件	顾建林 (3.47)
微机计算计数器大批投产	李相彬 (3.54)
消息二则	(3.70)
信息八则	(4.封二, 13, 15, 21, 66, 68)
MASP-P电能计量遥测处理系统达到国际先进水平	(5.19)
介绍一种高级工具软件	(5.7)
信息四则	(5.5, 15, 31, 45)
《中华学习机系统盘》(CEC-DOS)简介	邵爱楠 (6.56)
SMD-6410系统捷足先登	马在强 (6.57)
《Z80汇编语言BCD 码长浮点常用函数运算符程序库》简介	张铁钢 (6.57)
微机测试婴幼儿智能状况程序问世	李相彬 (6.17)