

微小型计算机



# 开发与应用

MICRO-MINICOMPUTER  
DEVELOPMENTS & APPLICATIONS



6

1989

微小型计算机开发与应用编辑部



微小型计算机开发与应用（公开发行）

---

编辑：《微小型计算机开发与应用编辑部》

出版：天津市电子计算机研究所

地址：天津市河西区友谊路爱民道5号

发行：天津市邮局

印刷：天津武清县长宏印刷厂

订购处：全国各地邮局

定价：0.95

---

邮局代号6—87

津工商广字0146号

国内统一刊号CN12—1122

《微小型计算机开发与应用》编辑委员会

顾问	郭平欣		
主编	黄侃		
副主编	王治宝	邹秀凤	
委员	(以下按姓氏笔划排列)		
	于万源	于清汶	王治宝
	王镭	王士禧	王寿松
	付国明	许镇宇	朱植松
	曲庭维	李凤祥	刘连棟
	陈力为	邹秀凤	吴锦声
	房家国	夏纪寅	夏业勋
	袁维本	曹东启	黄侃
	黄宝良	章渭臣	梅克定
	查宣明	裴少峰	薛大中

编辑:《微小型计算机开发与应用》编辑部  
出版:天津市电子计算机研究所  
天津市电子计算机学会  
地址:天津市河西区友谊路爱民道5号  
发行:天津市邮局  
印刷:天津市武清县长宏印刷厂  
订购处:全国各地邮局  
定价:0.95  
邮局代号6—87 津工商广字0146号  
国内统一刊号 CN12—1122

1989年第6期目录  
(总第44期)

微小型计算机应用实例

- 以PC机为主机的新一代国产化微型计算机  
开发工具……………机电部第46所(1)  
在IBM—PC机上实现两个任务并发运行的一  
种具体做法……………王祥 马宁(7)  
单板机显示曲线图象的应用……………  
……………鲍可进(9)  
用于实时处理的主从式多微处理机系统……  
……………方滨(11)

计算机软件

- 用主语言提高数据库软件的效率、可移植性  
和分布性……………刘长江(14)  
结构系统分析法在数据库设计需求分析中的  
应用……………匡文学(18)  
为FORTRAN语言增加系统、图形功能……  
……………叶天彤(22)  
IBMPC/AT机硬盘文件及剩余空间的加密  
与保护方法……………于甫(24)  
基于 Smalltalk 语言的面向对象的程序设计  
……………朱海滨(26)  
使用dBASE III编程的几点技巧……………  
……………罗伟其(32)

部件与接口

- 通道数可变的数据采集和过程控制接口……  
……………杨宪泽(36)  
介绍一种新型的存储器—电擦式可编程只读  
存储器……………施文济(38)  
利用APPLE II微机控制DMS—1型64路数  
据采集装置的技术……………宋晓东(40)

经验点滴

- 方便、快速地修订CCDOS拼音首尾码表……  
……………吴家富(43)  
打印机控制命令的中西兼容……………  
……………陈万年(45)  
PC机C88语言I/O接口程序的汉化……………  
……………崔迎大(47)  
89年总目录……………(48)



## 《征订启事》

《微小型计算机开发与应用》适于计算机界广大科技人员,企管人员,教学人员,大专院校学生及各行业的计算机用户阅读。

为了向广大读者提供信息,本刊还将开辟一定版面为新产品刊登广告。

本刊为双月刊，逢单月末出版，16开本，48页，每期定价0.95元。由天津市邮局向全国发行，全国各地邮局均可订阅。邮局代号：6—87

本刊编辑部地址：天津市河西区友谊路爱民道5号

本刊为国内外公开发行人，投稿时，作者姓名和文章题目请译成英文附在文后。

**欢迎订阅，欢迎投稿**

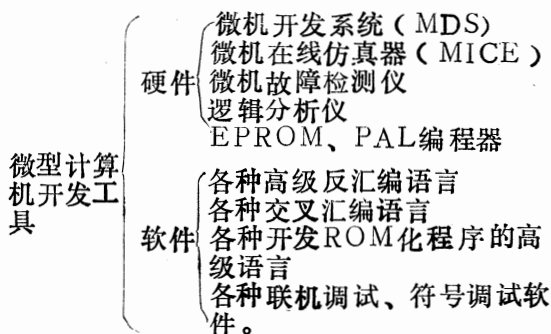
# 以PC机为主机的新一代国产化微型 计算机开发工具

机电部第46所智能仪器研究室

**摘要** 随着微处理器在国内的广泛应用,对各种高效、实用的微机开发工具有了广泛和迫切的需求。以国内主流计算机长城系列机做为主机(包括IBMPC/X T1286/386等兼容机),设计、研究新一代国产化开发工具,成了十分现实和迫切的课题。机电部天津46所近年来致力于这方面的研究工作。开发、设计并投放市场一大批以PC机为主机的国产化开发工具。受到用户的广泛欢迎。本文就这方面的情况,作较详细的介绍。

## 一、以PC机为主机的开发工具的结构及特点

微型计算机开发工具,包括了在设计、开发、调试各种通用、专用微机系统,智能仪器仪表的过程中所需要的各类工具,其大致分类如下所示。



以PC机为主机的开发工具结构如图1。

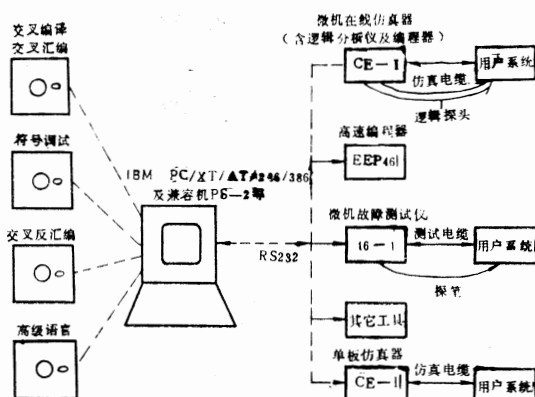


图1 以PC机为主机的微机开发工具结构框图

由图1可以看出,以PC机为主机的微机开发工具,基本复盖了开发工具需求的空间。然而,由于这些开发工具具有下列特点:

1.均采用微机技术设计。不论是各种仿真器编程器,还是故障分析仪等开发工具,其实质均是一台内含微处理器的智能仪器。其每台仪器内部,均有中央处理器(CPU),存储器(RAM),含有大量监控软件的程序存储器(EPROM)。从而构成一个可以独立工作的微机系统。

2.全部采用RS232接口与主机(PC机)通讯,用户需要使用其中任何一种工具,只要将该仪器与PC机RS232口插头插上,再放入特定的软件,即可使用。不用改变PC机内部任何硬件即可实现将PC机升级为高档开发系统、编程系统或故障分析系统。

3.全部开发系统的工作均在PC DOS (MSDOS)下运行。这样不仅大大减少了用户使用上的第二次重新学习,只要熟悉PC机的DOS系统,就能方便的使用各类开发工具,而且大大扩展了这类开发工具的使用范围。

4.由于全部采用积木式的硬件结构,并可以独立工作。所以大大提高了系统的可靠性和可维护性。主机或开发工具的故障一目了然。同时,外接仪器的联接方式上与插卡方式比较,外接方式大大减少了主机(PC机)的硬件负担,对提高主机的可靠性,减少主机故障,保证主机安全十分有益。这一点对高档微机(PC/386、PS-2)等尤为重要。

统灵活性。

综上所述,以PC机做为主机的各类开发工具,由于价格便宜、通用性强、一机多用、使用方便等特点,在国际上已成为微机开发工具的主要潮流。这一代开发工具的国产化及在国内的广泛应用,必将对我国微型计算机应用,各类智能仪器仪表的开发及传统产业改造产生巨大作用。

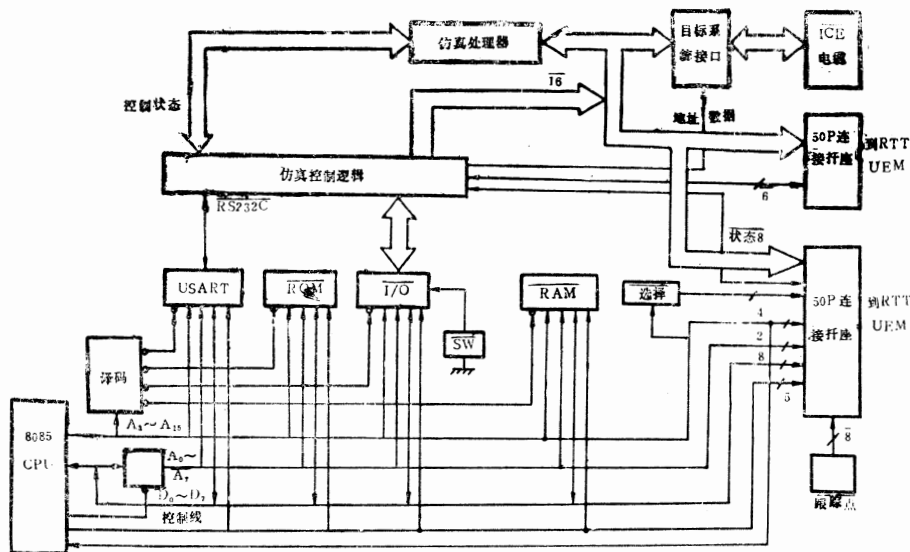
### 1. 在线仿真器 (CE—II)

[illegible]

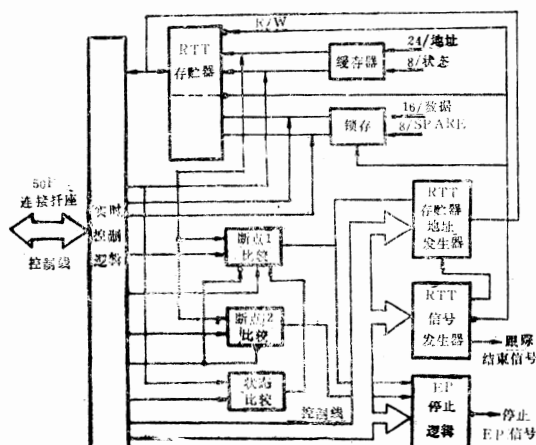
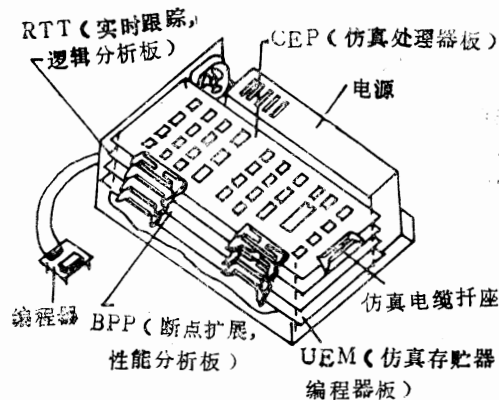
CE—II 仿真器电气原理框图如图3所示。

由图2看出,CE—Ⅰ仿真器,由4块电路板组成。但作为基本系统,仅需2块板(CEP+RTT板)即可运行。当然,存储器仿真板(UEM)一般也是必要的。新设计的断点扩展板,不仅实现了扩展4个硬件断点,而且实现了程序时间性能分析等高级功能。4块电路板采用标准50线插座及电缆联接。这种构造便于电路板增减及更换,增加了系

构,使CE系列仿真器,完全区别于一般单板机扩展方式的简易仿真器。成为高级的开发工具。



CE-1 仿真器由于配备了BPP扩展板, 所以, 除了增加 4 个硬件断点外, 还新增了程序性能分析的功能。该功能目前国外少数极高档的开发系统才配备。使用该功能, 用



户不仅能测试程序运行的实际时间,还能对编写的程序进行性能评估,以直方图方式列出执行地址,执行周期等。图6是性能分析显示的例子。

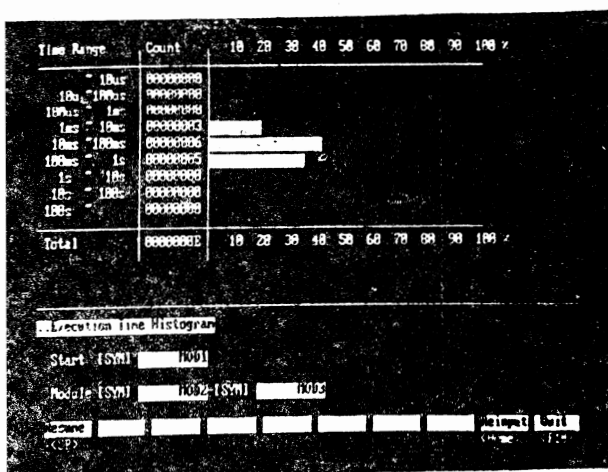


图6 CE-Ⅱ程序性能分析示例

CE-Ⅱ仿真器使用的是单字节命令,具有很强的扩展功能。不仅容易记忆,而且容易使用。其指令表如表1所示。

这一套命令,不论你更换何种CEP板,或更换何种处理器,均基本相同。这样,可以大大提高用户的工作效率。

为了将CE-Ⅱ仿真器配备成一个完整的开发系统,还在PC机上配备了相应仿真开发软件。

操作系统: MSDOS 2.0以上版本均可。

交叉汇编: 包括各种仿真处理器的交叉汇编生成目标代码,并具有宏汇编、条件汇编、多重嵌套和重新定位等功能。

交叉编译: C语言、FORTRAN语言, PASCAL语言等。

通讯软件: 通过PC机控制仿真器,并且两者能相互传送HEX文件。

仿真联机软件: 能控制仿真、跟踪、排错。

表1 CE Ⅱ 仿真器命令一览表

ASSEMBLY	A [[CS:DS:SS:ES:seq: [addr]
BACKWARD TRACE	B [R]addx[c[q]]
CYCLE STEP	C [W]c]
DISABLE	D [N: I: H: C]
ENABLE	E [N: I: H: C]
FORWARD TRACE	F [R]addx[c[q]]
EXECUTION	G [[seq: ]addr
BREAKPOINT	H [o[1:2]: i[addx[c[q]]]: 2[addr]]
INPUT	I [W]port[c[time]]
JUMP	J [seq: ]addr
LIST TRACE	L [step[a1[a2[q...]]]: S/step]: Z[step]: N]
MEMORY	M [[W] [CS:DSS]ES: seq: ]a1[a2[d1[...?d8]]][S]]
OUTPUT	O [W]port d1[...d8]
REGISTER	R [AX: BX: CX: DX: SP: BP: SI: DI: DS: ES: SS: CS: IP: FS]
INSTRUCTION STEP	S [S[R][c]: [Z]
TRANSFER/TEST	T [CS: DS: SS: ES: seq: ]a1 a2 S: M: [CS: DS: SS: ES: seq: ]a3[ ]
RESET	X [seq: addr]
DISASSEMBLY	Z [[CS: DS: SS: ES: seq: ]a1[a2]]
HELP	? [B] ?
ATTENTION	! !

符号化调试: CE-Ⅱ配套了功能很强的符号化调试程序(USD),能用符号进行排

错调试,并能处理各种系统与用户定义的符号与符号表。



## 2. 普及型仿真器CE-I

CE-Ⅱ型仿真器功能十分完善,价格虽较引进的同类产品低80%,但每台也近万元。对于国内一些中小企业及学生教学实验等,价格还是偏高。为此机电部46所还设计了新型的CE-I型普及型仿真器。其基本仿真调试功能均保留,但价格仅为CE-Ⅱ的1/3。CE-I仿真器内部结构如图7所示。

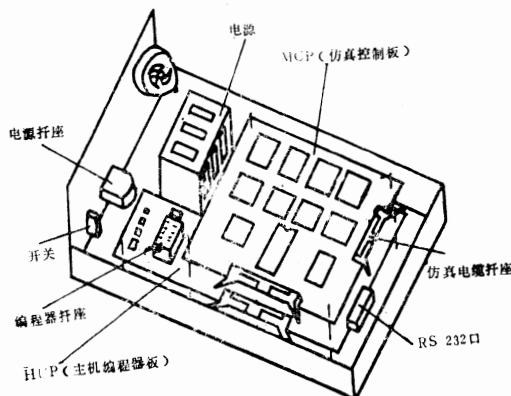


图7 CEI仿真器内部结构框图

CE-I仿真器具有一个较小的机箱,全部仿真器由二块电路板组成。采用双处理器原理设计,MCP板是可以更换的,以适应不同处理器仿真。目前已能提供Z80/6502/8085/8048/8051/8086/8088等MCP板,并很快提供新的MCP板,在MCP板上还配备了8K~64K仿真存储器。

HUP板由主机(主CPU、RAM、ROM等)及高速编程器组成。16K的固化仿真监控程序,包括3行汇编、反汇编、单步、断点等20多条双字节命令表如表2所示,并能将在目标系统调试运行好的程序直接通过编程器固化到EPROM式单片机中,该仿真器能固化EPROM范围为2716~27512及8748、8751等单片机。

## 3. EEP46型高速通用编程器

EEP46型高速通用编程器,是一种内含微处理器,能直接联接任何显示终端独立工作或联接PC机工作的新一代编程工具。

表2 CE-I指令表

1>设置出借仿真RAM.....SE	2>撤消出借仿真RAM.....RE
3>开放控制信号线.....EC	4>封闭控制信号线.....DC
5>显示内存单元内容.....DM	6>修改内存单元内容.....XM
7>显示寄存器内容.....DR	8>修改寄存器内容.....XR
9>改变PC指针.....JP	10>口输入.....IP
11>口输出.....OP	12>单周期执行.....SC
13>单指令执行.....SI	14>条件中断.....HA
15>连续单步非实时运行.....RU	16>实时运行.....RR
17>正向跟踪.....FT	18>反向跟踪.....BT
19>显示跟踪结果.....LT	20>RAM,ROM测试.....TS
21>内存数据块移动.....MV	22>帮助命令.....HE
23>仿真CPU复位.....RT	24>行汇编.....AS
25>行反汇编.....DS	

EEP46编程器,采用高速编程方法,编程速度为普通方法的10~20倍,大大节约了编程时间。EEP46编程器有2个无插拔力插座。该机不仅能读/写、校验、复制各种

EPROM,还具有奇偶字节写入,单片机写入功能。

EEP46编程器命令如表3所示。

**表3 EEP46高速编程器命令集**

DASM;	行反汇编。
L ;	内存数据移动。
RAM ;	测缓冲RAM区范围。
TEST;	测缓冲RAM是否正常, ROM求校验和。
V ;	内存数据块之间校验。
P ;	显示缓冲RAM中数据及对应的ASCII码。
B ;	检查芯片是否为空。
R ;	读EPROM中的内容到缓冲RAM内。
W ;	普通速度写EPROM芯片
H ;	高速写EPROM芯片
U ;	奇/偶字节读EPROM。(可根据用户要求增加该功能)
D ;	奇/偶字节写EPROM。(同上)
C ;	检查比较EPROM中内容是否与缓冲RAM中内容相同。

由内部原理可知,编程器内部具有固化的监控软件及大容量用户存储器(64K),这样,用户可以将软件从PC机磁盘中,直接通过RS232接口传送到编程器内存中,经修改、分配等处理后,固化于EPROM。也可以将样片插在编程插座上,然后读入编程器内存中,再换上一片写入,实现复制器功能。EEP46编程器,不仅有优良的性能,还具有一系列安全保护装置,以确保芯片不被损坏。安全写入,更主要是具备独特的编程电压软件调整技术。采用软件程序控制各种高压。编程电压可在5~23V可连续设定。大大提高了写入安全性,并方便了用户。

目前,EEP46编程器已在国内各科研所,电子生产企业得到广泛的应用。

#### 4. 微机故障分析仪

微机故障分析仪,其外部安装联接等,类似仿真器。但其实质是大不一样的。微机故障分析仪主要是通过总线检查及探测,实现微机故障的可编程自动定位,连续检测。这些功能,过去只有昂贵的引进设备上才具

备,目前的43I型微机故障分析仪,通过更接引出插板,能实现对常用8位处理器(Z80/8085/6502/6800)的自动连续测试。

该故障分析仪,其核心也是一套微机系统,通过测试探头的探笔,对被诊断仪器或微机系统进行故障分析。

该机有一套类似BASIC的小型测试语言。允许用户编程,一般而言,对一个特定系统,内存,I/O及外部A/D、D/A等均是固定的,用户可以一次性编制诊断程序存入磁盘中,当系统诊断时,插入探头运行程序,则可自动检查出开路,断路、总线及I/O故障等。这对工厂微电脑产品的成批测试具有重要作用。

另外,使用特制的探笔,还可以无须电路图,识别出目标板的信号名称、译码范围等,功能十分完善。

该故障分析仪,配有PC机即可运行,价格低、适用范围广,是数控机床等电脑设备必备的新型修理、调试工具。

### 三、微机开发工具的应用

微型计算机开发工具在产品开发的各个环节,都有着广泛应用,其应用分布图如图8所示。

由图8看出,开发工具在设计、生产、维修的每一环节,均有重要的作用。随着各种专用电脑设备,智能仪器仪表及传统产业微电子技术改造的深入,各种微机系统的设计难度更大,开发对工具的依赖性将进一步增加。所以,国产化微机开发工具的推广,必将对我国微机应用、特别是各种单片机应用、提高系统设计水平及系统设计、大大提高效益方面,产生深远影响。

另外,我国引进的大量采用微机电脑的先进设备,仪器的维护、维修及分析、解剖,

# 在IBM-PC机上实现两个任务并发运行的一种具体做法

南开大学 王 祥 马 宁

**摘要** 我们通过引入一个小的单用户/多任务处理系统doubletasks,在IBM-PC普通DOS环境下实现了打印磁盘文件任务P和另一个任务T并发运行,这不仅提高了硬件资源的利用率,而且也为用户提供了方便。本文介绍doubletasks的具体实现情况。

## 一、调度策略

doubletasks采用等时间片分时轮转法执行任务调度功能,时间片长度选为

$5 \times \frac{1}{18.2} \times 1000 \approx 275 \text{ms}$  (式中的18.2是每秒钟产生时钟中断的次数)。

在该系统中,每一个任务有两种调度状态:执行状态和等待状态。当任务T处于执

都需要从现有的凭经验的方式过渡到依靠新型工具进行工作的方式。我国现行主流机长城0520及大量进口IBM PC机的维护、维修,也可用仿真器来完成。而且采用仿真器进行故障维修,具有可编程,方式灵活等显著优点,值得提倡、推广。

总之,以PC为主机的新型国产化微机开发工具,不仅为我们扩大PC机应用范围,加快我国微机产业的产品化、商品化,提供了新的手段,而且,已开始向国外出口,开拓了高技术产品出口的新渠道。

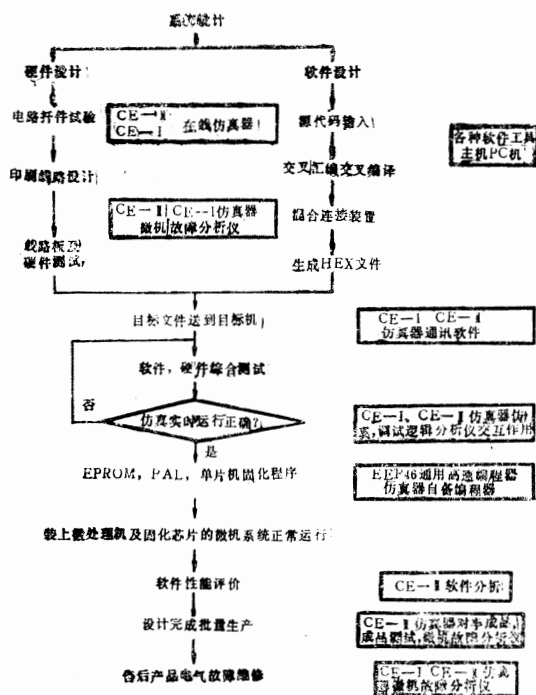


图8 微机开发工具在微机开发过程中的应用简图

## 更正声明

天津机电部46所在上期封面刊登的8/16/32位通用微机在线仿真器广告中,单位电话号码有误。应为31.8342。望广大客户注意。



行状态时,任务P处于等待状态;当任务P处于执行状态时,任务T处于等待状态。每过一个时间片状态转换一次,原来处于执行状态的任务变为等待状态原来处于等待状态的任务变为执行状态。状态转换过程中的全部调度工作均由doubletasks的核心部分自动完成。

## 二、系统结构

整个doubletasks系统由数据段、栈段和代码段三部分组成,其中代码段为主体。

代码段可分执行和驻留内存两个子段。

执行子段位于代码段的前53行,顺次完成以下三种操作:(1)借助DOS功能调用25H重新设置打印屏幕中断INT5H和时钟中断INT1CH的中断向量;(2)模拟进入INT1CH前的系统栈情况;进入INT1CH前的系统栈如图1所示。

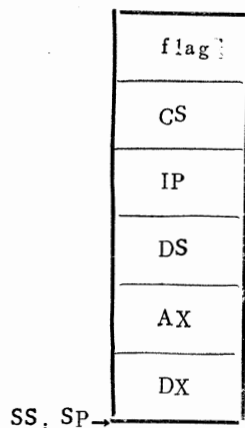


图1

栈中所保存的寄存器内容是发生时钟中断前处于执行状态的那个任务的环境信息。修改时钟中断后,系统栈保存处于等待状态的任务的相应寄存器内容(除这些栈保存的信息外,还有一些寄存器的内容被放在存储器的某些特定单元中);(3)发INT27H,终止执行子段且使doubletasks驻留内存(为了避免覆盖,我们对DX增加了1000H)。

驻留内存子段包括INT5H,INT1CH和PRINT三个子程序。其中

INT5H是我们自己设置的键盘中断服务程序,它完成的功能是:(1)清屏幕第一、二行;(2)显示插入打印任务的提示信息;(3)回显原屏幕第一、二行的内容;(4)将标识是否有等待任务的变量wait置1。

INT1CH是doubletasks的核心部分,它按照(一)中给出的调度策略执行任务调度,工作流程如图2所示。

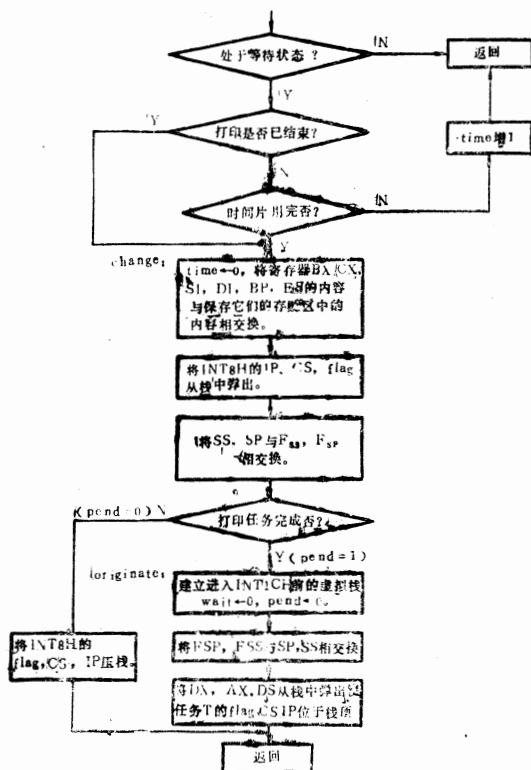


图2

PRINT子程序完成插入打印磁盘文件任务的功能。它根据从键盘输入的驱动器号和文件名,将指定的文件调入内存并送至打印机进行打印,但不在屏幕上显示所要打印的文件内容(它的执行特征与MS-DOS3.30的Print命令相类似)。

doubletasks系统用IBM-PC宏汇编语言编写,源程序清单略,(需要清单可与作者联系)。

## 三、使用方法

# 单板机显示曲线图象的应用

江苏工学院 鲍可进

**摘要** 本文提出了用单板机处理单值函数曲线的方法。把曲线图象数据压缩到最小限度以适应单板机的小容量内存。

在一些工业控制系统中,需要按一定规律动态跟踪或监视某一函数,用显示器把这函数曲线显示出来,或在显示器上移动曲线。如果处理的函数是单值的,则可将一幅图象的数据压缩为显示分辨力的列的点数,如图象显示分辨力为 $192 \times 256$ ,则一幅曲线图象只需256个数据,也即只记录曲线上每点的纵坐标。

## 一、曲线数据的读入

单板机配的显示接口,每次从屏幕上读得一个字节的数,即8个光点,一行256个点,共32个字节,一幅图象有192行,就是说可把图象分成32列,每列有192字节。从屏幕上逐列逐列地读入数据,如全要保存

则为6k字节,但是单值函数曲线,对应于曲线每一点,只有一个纵坐标Y值,而横坐标X可不记录,是固定的。这样一幅图象许多字节是全0的,我们只需考虑不是0的字节,那么用256个单元记录非0点的纵坐标就够了,其程序清单如下:

```
READ:  LD      IX,(AD)
        LD      L,0
        LD      H,0
        PUSH    IX
LOOP1:  CALL    0C012H,读屏到A中
        POP     IX
        CP      0
        CALL    NZ,RECORD
LOOP3:  INC     H
        LD      A,H
        CP      191
```

在DOS状态B>下执行doubletasks (不妨假设doubletasks的源文件和运行文件均在B盘上)系统即进入可并发执行T,P两个任务的状态;按下prtscr键,屏幕显示出提示信息;根据提示键入驱动器号和文件名(它们均被显示在屏幕第二行);按回车键,打印机开始打印指定文件并回显屏幕第一、二行内容。至此,打印任务P已完全建立起来。由于在打印磁盘文件过程中不在屏幕上显示所打印的信息,所以用户仍可执行其它的任务(任务T)。

要特别注意的是,在键入驱动器号及文件名过程中出错时不能借助退格键进行修改,而必须:(1)按回车键;(2)打印出

错信息后重新按Prtscr键;(3)重新键入驱动器号和文件名;...

本系统尚有一点不足之处,这就是当任务T也需要访问磁盘时,常会因磁盘驱动器的速度不适应T、P两个任务的竞争而导致死锁。这一点务请使用者注意。

### 主要参考文献

- [1] Multi-Tasking in Micros, Ian S. Turnbull The October Press, 1983.
- [2] PC-DOS分析报告,电子工业部第六研究所, 1986.2.
- [3] IBM-PC宏汇编语言程序设计,朱耀庭,于春凡 1986.2

```

JR      C, LOOP1-2
LD      A, 8
ADD     A, L
LD      L, A
LD      BC, 8
ADD     IX, BC
LD      A, L
CP      0
JR      NZ, LOOP1-4
LD      (AD), IX
RET
RECORD: LD      C, 8
        ERA
        JR      NC, LOOP2
        LD      B, 0
        PUSH   IX
        ADD     IX, BC
        DEC     IX
        LD      (IX), H
        POP     IX
LOOP2:  DEC     C
        JR      NZ, RECORD+2
        RET
AD:     DW      3000H
        END

```

## 二、曲线的显示

要显示某一曲线，只要从该曲线数据首址开始，依次取出数据作为纵坐标，而横坐标从左至右递增送显示器显示，则可显示该屏曲线，其程序清单如下：

```

DIS:    LD      L, 0
        LD      IY, (AD)
BBA:    LD      H, (IY)
        CALL    0C018H; 在屏上
                显示一点
        INC     L
        RET Z
        INC     IY
        JR      BBA

```

## 三、曲线的移动

如曲线需从右到左平滑连续移动，则可以把曲线每点向左移动一个单位，最右边一点取下一个曲线数据的纵坐标，横坐标为255，循环此过程，则可使曲线从右往左平滑连续地移动，直到规定的曲线结束点为止。其程序清单如下：

```

MOVE:   LD      E, 0
NEXT:   LD      IY, (AD)
        LD      H, (IY)
        LD      L, E
        LD      D, 01
        CALL    0C018H
        LD      H, (IY+1)
        LD      L, E
        LD      D, 0
        CALL    0C18H
        INC     IY
        INC     E
        JR      NZ, NEXT
        LD      EC, (LIMIT)
        PUSH   IY
        POP     HL
        OR      A
        SBC     HL, EC
        JR      NZ, MOVE
        RET
LIMIT:  DW      4000H
        RET

```

这种曲线的存贮、显示、移动的方法，成功地应用在汽车转鼓试验台的司机助手微机系统中，用来跟踪汽车的动态速度。以解决单板机内存容量小的问题，所有曲线数据可放在EPROM中直接插在单板机上，不需外存来送入数据。



# 用于实时处理的主从式多微处理机系统

北京工业大学 方 滨

**摘要** 本文介绍一种主从式多微处理机系统。它摒弃了传统主从式多微处理机系统中的公共存储器结构,取而代之的是一种“考场”式存储器结构,从而削弱了各子机之间的冲突竞争,简化了仲裁电路的设计,同时增加或减少子机(“考生”)的数目非常容易。由于不使用公共存储器结构,也使软件设计更加简单可靠,易于掌握。这种主从式多微处理机特别适用于任务可以主从化的实时工业控制场合。

## 一、前言

随着工业的发展,各种应用场合对计算机响应速度的要求越来越高。特别是在实时处理场合,单处理机系统已经显得有些力不从心了。然而,应运而生的多处理机系统由于某些技术原因,尚未能在实际中得到广泛的应用。在这些技术原因中的一个主要原因是公共存储器的竞争问题。

在主从式多微处理机系统中,公共存储器及与之相联的公共总线是各子机之间相互交换信息的主要通道。多微处理机系统正是利用公共存储器来实现信息共享,从而达到提高系统吞吐能力的目的。显然,这可能会产生两个或两个以上的子处理机同时要求访问公共存储器的情况。即所谓的公共存储器访问竞争。

为解决这种竞争,需要由仲裁电路判定谁先可以访问公共存储器。根据不同的需要,可采用不同的仲裁策略(固定优先级式,时间优先排队式,损失竞争式等)。每种仲裁策略的实现都需要一套复杂的硬件电路。由于逻辑上和时序上的复杂性,多微处理机的设计者往往会在仲裁电路的设计和调试上花费相当多的时间和精力。

在软件方面,信息的高度共享及程序的高度并发性也给软件的设计带来了许多麻烦。如何分解任务,如何将子任务分配给子处

理机,如何管理共享信息以及如何实现进程之间的同步等,一直是多处理机程序设计中的棘手问题。管理不好,非但不能发挥多微处理机的长处,反而会发生由于仲裁电路工作不可靠而带来的许多难以跟踪查找的毛病,像死锁、共享数据被破坏等。多微处理机之所以至今在实际中没有得到广泛的应用,这是一个很重要的原因。显然,问题是由引入公共存储器产生的。

## 二、基于“考场”式存储器结构的主从式多微处理机系统

这种系统由一个主机和多个子机通过一条总线联接而成。系统设有公共存储器,但主机和子机都有各自的私有存储器。所有子机都有一部分(或全部)内存与主机的一部分内存重叠。或者说,主机可以将每个子机的存储器作为自己内存的一部分来访问,每个子机也可以将主机内存的一部分作为自己的内存来访问。但各子机的内存之间是相互独立的。

在大多数工业控制场合中,控制算法保证信息可以按树形结构传递,也就是说各子

机是在主机的控制下工作的。主机除了对子机的工作进行监督管理外，还负责任务分解、各子机任务的分配及整个系统的输入输出操作等。

在实时工业控制场合中，为了提高速度，任务的分解通常是由程序员离线进行的，即事先由人工将算法转化成等价的并行算法。使算法分解成许多可以并行计算的子算法。并为每个子算法编写出相应的程序，这些程序将来要在各个子机上并行运行。系统工作时，先由主机将各子机的工作程序（子算法）发送到各机的程序存储器中，也可以事先固化在各子机的程序存储器中，然后，主机将各子机所需的局部数据送到“考场”式存储器里。这些数据一般是由主机从输入设备（如A/D变换器）中取得的。对于各子机都需要的共享信息，由主机采用“广播”方式写到“考场”式存储器中。在为各子机准备好数据后，就可以通知子机开始工作，子机对主机发来的数据进行处理。在各子机并行工作的同时，主机可以作输入输出或其它的操作。待子机运算完成后，子机将结果放入“考场”式存储器中，然后通知主机，并停止子机的工作。主机接到子机的通知后，从“考场”式存储器中取回数据，然后再放入新的数据以开始下一个计算循环。由于工业控制算法大多是按一定的采样频率工作的，所以计算具有间断性，因此，数据的传输也是阵发性的。

按照此要求设计的存储器是一种多存储器结构。在这种结构中，不存在这样一种存储器实体，它能同时作为主机和子机内存的一部分，并由主机或各子机来访问。这是“考场”式存储器结构与公共存储器结构的关键不同之处。除此之外，它还应满足下面的树形条件：

- 1) 主机端口可以通过存储器和其它任何一个子机端口直接交换信息。
- 2) 任何两个子机端口不能直接通过存储

器交换信息。

- 3) 主机端口可以通过存储器向全部或部分子机端口同时发送信息。

第2条使得增加或减少子机数目变得很容易。与公共存储器结构不同，系统中子机的数目不受仲裁电路的硬件限制，它仅仅受限于软件实现的算法效率。

第3条是缓解冲突竞争的关键。它将共享信息以“广播”的方式同时发送到各个子机，使各子机同时拥有本来需要通过竞争才能得到的信息，省去了复杂的仲裁逻辑，简化了电路设计，提高了系统的可靠性。

由于去掉了公共存储器，不存在共享信息的竞争问题，因而使软件的设计得到了简化，要注意，此结构只缓解了共享信息的竞争，但仍没有简化进程同步问题。

### 三、实现实例

按照这种想法，作者设计并实现了一种主从式多微处理机系统。系统的构成如图1所示。

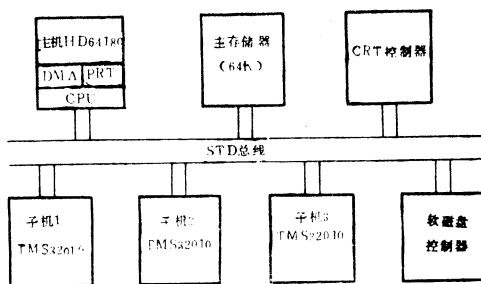


图1 主从式多处理机系统

该系统的总线采用STD工业控制总线标准。主机选用高性能八位CPU HD64180。这种CPU的指令系统与Z80兼容，芯片内部还有两个DMA控制器，两个定时器，两个异步通讯口及一个同步通讯口。其中定时器用来产生采样信号，DMA用来实现主机

与子机之间的信息传递。串行口可用于与其它系统联接,构成更复杂的分布式系统。

子机采用高速十六位信号处理机 TMS 32010 作为 CPU。每个子机有 8k 字节的内存。当子机工作时,这 8k 字节的内存作为 TMS 32010 的  $4k \times 16$  程序存储器。当主机访问时,这 8k 字节的内存点占据了主机 64k 内存中的 8k。注意,系统中所有子机都占用主机中的同一段内存地址如图 2 所示。

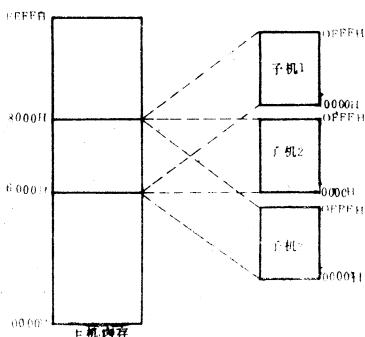


图 2 内存分配

这样才能实现“广播”式发送。为了使主机能够选择访问哪一个子机的内存,每个子机上都装有一个存储器门锁电路。[3]当某子机的存储器门锁电路起作用时,该子机的内存就不能由主机访问。所以主机可以通过控制子机上的存储器门锁电路来选择要访问的子机。子机运算成后,用中断来通知主机,从而可以用已经成熟的中断判优电路来代替仲裁电路,提高了系统的可靠性,简化了程序设计的步骤。主机和每个子机都设计在 STD 总线标准的电路板上,增加或减少子机板的数目都非常容易。

在这个系统中增加所需的外设接口便可构成一个实时信息处理系统。作者在这个系统之上增加了光码盘接口电路、D/A 变换电路和驱动电路,构成了一个多微处理器、多变量、电机控制系统。该系统现在用作工业机器人的控制器,来计算关节型工业机器人的动力学方程。该方程是一个多变量的非线性方程组 [4]。算式很长、计算量大(在

五自由度情况下,即便采用符号解,也有 156 次单精度浮点乘法,99 次单精度浮点加法、12 个超越函数),时间要求又很紧(一般要求 10ms),所以单处理器系统唯一胜任。例如,用一个 8088 来计算需要 300ms;若 8088 + 8087,需要 50ms,还是不能达到要求 [5]。采用这种主从式多微处理器系统,在四个子机的情况下,全部动力学方程的计算可在 4ms 内完成 [2]。

除了提高系统的处理速度外,本系统在简化程序设计方面亦有明显的改进。该系统的其它性能还在进一步的评价中。

#### 参考文献

- [1] Ravi Nigam and C.S.G. Lee:  
"A Multiprocessor-Based  
Controller for the Control of  
Mechanical Manipulators"  
Proc 1985 Int Conf. on Robotics  
& Automation, PP815
- [2] Ray Simar, Jr.  
"Floating-Point Arithmetic  
With the TMS32010"  
Digital Signal Processing  
Applications with the TMS320  
Family, PP213 Texas Instruments,  
1986
- [3] 方力:  
"微型机高速运算处理单元的研制"  
北京工业大学自动化系毕业论文 1987, 6
- [4] M. Vukobratovic & N. Kircanski,  
Real-Time Dynamics of Manipu  
lation Robots  
Springer-Verlag, 1985
- [5] 方滨:  
"五自由度关节型工业机器人动力学方程的  
符号算法"  
北京工业大学自动化系技术报告, 1988 \*5



# 用主语言提高数据库软件的效率、 可移植性和分布性

新疆独山子乙烯工程指挥部 刘长江

**摘要** 目前,凡是使用计算机的人,几乎都知道关系数据库。目前国内微机最流行的象dbase、ingres、informix等都是典型的关系数据库。本文将介绍我们在使用关系数据库的同时,采用主语言的方法,为提高其运行效率,软件移植性和分布性上所作的努力和探讨。

## 一、问题的提出

所谓关系数据库管理系统是以表(如ingres)、库(如dbase)、文件(如informix)来管理数据。它们的共同特点就是都由列组成的成行的数据的集合,其中列是数据字段,而行是一个独立的记录值,这个概念是很容易理解的。

与dbase不同,informix和ingres除此之外还有数据库的概念,即由数据库管理着若干表或文件,显然后两者更具有数据库的特点,本文重点对其进行讨论,当然,其中的思想和方法,对其它数据库管理系统也是适用的。

凡是学习和使用过数据库语言的人都知道,数据库系统较之文件系统具有安全、可靠、数据独立性、减少冗余、完整性、并发控制等特点。数据库语言与高级语言相比,具有简单易学、使用灵活方便,编程效率高等优点。因此,受到了计算机软件人员,尤其是应用软件人员的欢迎,在很多场合下取代了高级语言而成为软件开发的重要工具,以至于有些软件人员认为只要掌握了数据库语言就可以不再学习高级语言。

在实践中,我们体会到,数据库语言的确是强有力的软件开发工具,但美中不足是数据库语言开发的软件相对来说运行效率较低,并且移植性差,对数据的分布处理功能也较差。

我厂从1985年开始与中科院新疆物理所联合开发独山子炼油厂企业自动化微机网络系统,1988年上半年通过部级鉴定,并获得石油部计算机应用一等奖。在该系统中采用了主语言与数据库语言相结合的方式,较好地解决了上述问题,并以此开发了该网络支撑软件中的分布式数据库系统和计划、调度、财务、销售、人事等十几个应用子系统。下面对提高数据库软件的效率、移植性、分布性方面分别加以讨论。

## 二、采用主语言提高程序效率

在我厂各应用子系统开发初期,大量使用了如ingres和dbase等数据库语言进行编程,在使用过程中,我们发现数据库语言虽然编程效率较高,但其运行效率相当低,尤其是运行计算量较大的程序时矛盾更加突出。如计划子系统的成品台账输入和处理,使用ingres数据库语言编程时运行一次约需2小时,这显然不能被用户所接受。又如销售子系统每天数以千计的票证处理,采用dbase数据库语言处理时约需80分钟,这对用户来说也是难以忍受的。

数据库软件运行速度慢的原因一方面是很多动态的数据结构难以实现,另一方面就是在于数据库进行检索时采用的逐行扫描的方式进行,每一个语句均需对数据库进行一遍扫描,以获取所需数据,这一方面使数据库具有了强大的检索功能,另一方面使其效

率大大降低了。如果一个库表有 $m$ 个记录，一个针对该库表进行操作的程序有 $n$ 行，假设程序中每个语句均需访问数据库，则运行时需对数据库进行 $m \times n$ 次扫描，如果 $m$ 和 $n$ 的值都比较大时，程序执行的时间就会很长，并要进行频繁的盘交换。当然，我们可以对某些关键字段建立索引或散列函数，根据条件决定是否扫描全部库记录，采用这些措施，一定程度上可以加快运行速度，但令人遗憾的是，实际应用过程中往往有很多数据是处于非关键字段中的。

为此，我们提出了这样的设想，让数据库负责管理和保存数据，有时也可担负一部分输入/输出工作，以充分发挥DBMS安全可靠的特点和其强大的检索功能及良好的用户界面，而主要的计算工作均采用主语言（可以是c、basic、pascal、fortran等各种高级语言）来完成，发挥高级语言运算速度快的优点。

对于ingres和informix数据库系统来说，本身就具有与宿主语言如c语言的接口，象ingres数据库，有EQUEL/C语言，也就是常说的扩展c语言，对informix数据库，有一大批c语言例行程序供用户使用。即便对dbase这样较简单的数据库来说，也可以用高级语言读取数据库文件，这方面在国内已有许多成功的先例。对于初学者来说，了解库文件的结构可能比较困难，但也可以用copy命令生成txt正文文件，然后再用高级语言进行处理。采用主语言的方式，就是将数据经一次扫描从数据库中全部或部分读取出来，在内存中进行计算和处理，其结果或输出，或存回数据库中，从而大大提高了程序的运行速度，

以前面介绍过的成品台账处理程序为

例，运行时间由原来的两小时缩短到8分钟，销售数据处理由原来的80分钟缩短到10分钟，其效果是十分显著的。

### 三、采用主语言提高软件的可移植性

前面我们介绍了可以采用主语言的方法提高软件效率，与此同时，我们发现采用主语言方式编程后，可以大大提高软件的可移植性。也就是说当改换机型或数据库时，只需对原有程序稍加改动即可运行。

众所周知，数据库应用软件的可移植性差，这主要是由于每一个DBMS都有自己的一套命令或语言，这一点是很容易理解的，很难设想一个用dbase数据库语言编写的程序能不加修改或只加很少修改就可以在ingres或informix数据库环境下运行。

采用主语言方式后，由于程序的主体是高级语言，而高级语言的可移植性是比较好的，所以一旦数据库系统或机型更换，只需对程序中有关数据库部分加以修改，重新编译后即可运行。

请比较以下用EQUEL/C语言写的ingres数据库的数据处理程序和用扩展c语言写的informix数据库的数据处理程序。

#### 程序一：

```
# include <stdio.h>
## 数据类型及变量定义
main ( )
{
## ingres database /*打开数据库*/
## range of e is data file
```

```

/*为表格定义域变量*/
** retrieve( 变量1=e。字段1,
            变量2=e。字段2, ..... )
/*读表格中数据*/
** {
    用c语言写出的数据处理部分;
}
    用c语言写出的结果处理部分;
** }

```

#### 程序二:

```

#include <stdio.h>
#include <dbio.h>
变量类型及定义;
main( )
{
    dbselect( DBOPEN, "dat
abase" );
/*打开数据库*/
dbselect( FILEOPEN,
    "datafile" );
/*打开数据文件*/
cc=dbfind( "datafile", 参
数及变量表 );
/*读数据文件*/
while ( cc == SUCCESS )
{
    用c语言写成的数据处理部分;
    c=dbfind( "datafile", 参
数及变量表 );
}
    用c语言写成的结果处理部分;
}

```

从上述两个程序我们已可以很明显的看出,两者在很多地方是类似的,当我厂的机理由原来的BCM——68000改为ALTOS——3068和浪潮0530D之后,数据库也由原来的ingres关系数据库改为informix关系数据库,由于在软件开发过程中大量使用主语言方式,使得软件移植的工作量大大减少,很容易的完成了这一重大转换。如果以前的软

件开发均采用数据库语言的话,很难想象这种因为数据库和机型的改变而引起的软件修改量有多么巨大。

最近,新疆石油管理局综合经济信息系统总体设计方案通过评审,该方案决定全局的数据库系统统一选用ORACLE关系数据库,这一决定给几十个二级厂处带来了巨大的冲击,这意味着所有的软件系统的重新开发,而对我厂来说,由于较早地采用了主语言方式开发软件,程序的可移植性较好,从而避免了一次大手术。我们在软件开发的方向和策略上的成功,受到了与会专家的一致好评。

## 四、采用主语言提高软件的分布性

目前,关系数据库系统正在由早期的集中式数据库向着分布式数据库的方向上发展,虽然分布式数据库目前正处于发展阶段,但随着计算机网络技术的发展,这项技术显得越来越重要。

对于分布式数据库,目前定义不一,但其中较有影响的说法为:“分布式数据库是分布在计算机网络中不同计算机上数据的集合。网络上的每一个节点都具有自主处理能力,即能执行局部应用,并通过通信系统,至少参加一个全局应用。”这里的关键在于数据库在物理上的分布,对于最简单的情况,可以是多用户系统中不同用户、不同盘区上的数据库,也可以是不同机器上的数据库系统,这些数据库可以是同构的,也可以是异构的。如果我们采用数据库语言的话,即使在多用户系统中,实现分布式存取也是有困难的,更不用说在局域网或广域网中了。

我们首先讨论如何在一个多用户系统利用主语言如何实现数据库数据的分布式存取,这个问题清楚后,就可以说明独山子炼油厂微机网络中是如何对数据库中的数据实现分布式存取的。

对于一台使用UNIX操作系统的计算



机, 如果配有ingres或informix数据库, 凡用过的人都知道, 对于这种类型的数据库, 不论是用命令方式还是用程序方式, 在同一时刻内只能打开一个数据库, 可以对该库中的若干表格或文件进行操作, 即使是同一用户建立的不同数据库, 也不允许同时打开。更不要说不同用户建立的数据库还存在查询限定和授权等一系列问题。

正因为数据库具有这种特性, 所以分布式的数据存取(其实这还是在同一机器上, 严格的说并不是真正的分布存取)用数据库语言是很困难的, 如果使用主语言的方式, 首先它支持全路径名, 也就是说只要在同一机器上, 如果不考虑用户权限, 那么整个文件系统上所有的数据库都可能被程序访问, 但同一时刻只能打开一个数据库, 但由于是在高级语言中运行, 我们可以在内存中开辟缓冲区保存对前一数据库中一个或若干库表中查询或处理的结果, 然后再打开另一数据库, 根据需要对其进行操作, 而内存缓冲区则成为联系两个数据库的桥梁, 这时, 第二个数据库的操作是直接对库表进行的, 而对第一个数据库的操作实际上是在缓冲区进行的, 如果必要, 关闭第二个数据库后可重新打开第一个库进行必要的操作, 对于多个库也可以按此方法进行处理。

对于用户来说, 整个过程是透明的, 他自己并不知道自己的操作是直接针对库表还是针对缓冲区, 他的直观感觉是可以对不同数据库的数据进行分布处理。

对于独山子炼油厂微机局域网来说, 网络硬件选用的是日本的TERRANET网络,

网络软件自行研制, 其核心为分布式 unix 操作系统, 主要采用命名技术来取得分布性, 用于修改原来 unix 文件系统的命名和ingres数据库的命名, 其技术上的关键在于在原各机器的文件系统之上再加上一层, 形成一个新的根, 也就是我们所说的虚根, 以此构成一个新的更大的文件系统, 数据库系统其实也就是一个特殊的文件系统, 因此也可以联到虚文件系统中进行处理, 这些文件名, 数据库名将通过分布式软件进行解释, 或变成网络地址, 或变成本地unix可以识别的地址或名字, 如果是网络地址, 则通过一套算法来实现分布性。前面已经介绍过同一机器同一文件系统中数据的分布存取问题, 通过网络和分布式unix的支持, 网上的分布处理方式就和前面使用的方法十分类似了。对分布式ingres数据库系统, 分为命令级和程序级两部分, 仍采用原来的命令和操作方式。

在我厂网络系统应用软件中, 最复杂的一个就是分布式查询系统, 它连接全厂十几个应用子系统, 也只有通过它, 才能使分布的数据库系统的系统化变为可能, 全厂所有的数据, 通过查询系统成为一个整体, 从而成为辅助领导决策的一个重要手段。整个分布式查询系统均采用主语言的方式写成, 充分发挥了微机网络的作用。

综上所述, 我们认为采用主语言方式, 可以较好的解决数据库系统软件运行效率、可移植性、分布处理等方面的问题, 随着计算机应用的不断发展, 主语言嵌套的编程方式将越来越受到计算机用户的欢迎。

## 关中信息报

## 长期免费赠送

亲爱的读者, 您想引进先进设备吗? 您想了解购销行情吗? 您想寻求贸易伙伴吗? 您想走上成才之道吗? 本报愿为您提供全方位、高密度、大容量的信息。欢迎来函索取。

编辑部地址: 陕西省大荔体育路一号二楼  
邮政编码: 715100

# 结构系统分析法在数据库设计需求分析中的应用

海军工程学院 匡文学

**摘要** 本文介绍了一种将结构系统分析法应用于数据库设计需求分析中的方法。详细分析和证明了如何将结构系统分析法中的工具经过适当的修改,简化和提取,来帮助设计人员进行应用单位的需求分析,并录取数据信息,从而确定所要设计的数据库的数据信息内涵。

## 一、引言

现行的数据库设计方法分为需求分析、概念设计、逻辑设计和物理设计四个步骤。其中系统的需求分析是进一步设计工作的基础,它要求设计人员必须充分地理解应用环境的各方面要求和约束条件,工作量大且较繁琐,如果没有一个实用的方法可循,则有可能在投入了巨大的人力后仍然达不到目的,而一般的数据库设计方法对这一环节的阐述不全面,有的则是笼统地一述而过。

需求分析强调的是设计者和用户的协同工作,来完备地定义系统的需求。这步工作的主要困难如下:

1. 具有计算机和数据库知识又精通业务知识的人很少,而数据库设计要求设计人员对于这两方面的知识都必须有相当的了解。

2. 对于某一企业或组织的数据库系统的目标需求是什么缺少明确的规定,上面的方针没有确定,下面的设计很难进行。而确定方针的行政人员一般不精通计算机知识,很难提出具体的、恰当的要求。

因此有效的需求分析方法必须使用户和设计人员能充分地相互交流,必须要一种恰当的沟通两者的工具,即需要一个用户和设计人员都可以设计、理解、修改的中间媒介。

信息管理系统的分析与设计给了我们启示:用结构系统分析法(简称SSA方法)可以满足需求分析中的这一要求,它最大的特点就是结构化,这种结构化方法的应用使得需求分析变得简单、直观、完备可靠。虽然结构系统分析法是一般计算机化的电子数据处理的工具[2],但是通过适当地修改,简化和提取,可以用于数据库需求分析,并贯穿于设计的全过程[5]。

## 二、结构系统分析法(SSA法)

结构化系统分析法可以帮助用户了解将要建立系统的逻辑功能,从而判别是否满足要求,它的详细介绍可参阅文献[2],本文只介绍其基本特点和在需求分析中的应用。SSA法有以下特点:(1)用画图的方法,(2)自顶向下地分解、逐步求精,(3)强调是逻辑而不是物理,(4)没有重复性。用该方法来帮助设计数据库有以下优点:(1)SSA用的画图而不是繁琐的语言来简明地表达系统,用户能直观地理解系统的概貌,设计人员由此图的概念模式的设计打下了基础。(2)由于使用了自顶向下技术,把复杂的系统逐级地分解成尽可能独立的子系统、模块和子模块,这样用户对具体部分易

于理解,能提出修改意见,设计人员也可以了解数据的静态特性和动态处理特性。(3) SSA法着重逻辑级,而不是物理实现,这正好体现了数据库设计多级方法的思想,可以脱离物理级而独立地完成。

SSA法的工具有:(1)数据流程图DFD,(2)数据字典,(3)处理逻辑的表达方法,(4)数据存储结构规范化,(5)数据立即存取图。在数据库设计需求分析中,本着充分地了解用户的数据要求和处理要求这一原则,工具(3)(4)就不必要了、工具(3)描述的具体的处理细节。对数据库设计的影响不大,是开发应用程序应考虑的问题。工具(4)中的规范化问题的解决在数据库设计中有自己的一套完善的处理办法。工具(5)立即存取图对数据库的设计是有影响的,但它是物理实现时,结合具体的DBMS所应考虑的。因此在设计中,特别是逻辑设计的需求分析中,SSA法中应用最主要的是工具(1)和(2)。在建立数据字典时,必须考虑数据之间的依赖关系。

1. 数据流程图 DFD。它由四种基本单位,外部项、处理逻辑、数据流和数据存储。我们以[2]中的流程图为模式来说明它在数据库设计中的作用。数据流和数据存储可以反映出数据的静态特性、而处理逻辑可以引伸出数据的动态特性。

在画数据流程图时,自顶向下扩展到对处理逻辑有一个基本了解的程度就可以了,不必对处理逻辑扩展到能用程序清晰实现的程度,因为数据库设计中,要了解的是用户的数据要求和处理要求,而不是如何用计算机具体实现用户的要求。由此[2]中的汽车配件公司的一、二级流程图如下,而不必再扩展下去。一级数据流程图如图1所示,二级数据流程图如图2所示。

2. 数据字典数据字典最初用于数据库管理系统,现在又用来进行分析用户的数据

需求,其目的都一样:为用户、数据库设计人员以及数据库管理员提供数据的综合信息。

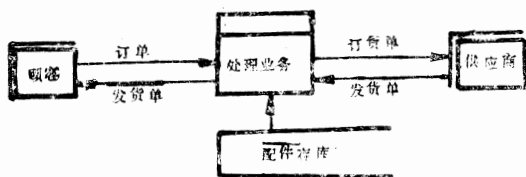


图1 配件公司的第一级数据流程图

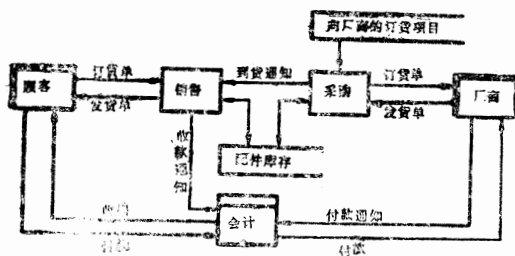


图2 配件公司的第二级数据流程图

用于数据库设计分析的数据字典由两部分组成:数据元素和数据结构。数据结构是由若干数据元素组成,数据元素是数据的最小单位,不可再分,它相当于关系数据模型中的域或字段,数据字典中对数据元素的定义包括四项内容,(1)名称或标识符,(2)别名,(3)取值范围,(4)数据元素的长度。字典中把逻辑上组合在一起的数据元素定义成一个数据结构,数据结构也可以由数据结构和数据元素的混合组成。定义数据结构对下一步进行数据依赖的分析是有帮助的,可以肯定逻辑上没有联系的数据之间是不会有依赖关系的。因此在分析数据依赖关系时,只要在某一数据结构内部的元素之间进行就可以了。数据结构的定义包括以下内容(1)数据结构的名称(2)数据结构的组成。有了数据元素和数据结构的定义,则数据流程图中的数据流和数据存储所含的内容就清楚了,在数据字典中定义它们由那些元素和结构组成。建立数据字典是需求分析中的重要任务。

### 三、确定数据库的数据内涵

用自顶向下所得的数据流程图中，特别是DFD到达某一层次后所含的数据流和数据存储是较多的，但所建数据库的数据内涵肯定包含在它们之中，数据内涵相当于关系模型中表头所含的内容。下一步的问题是如何用简明的方法来确定数据库的内容。

#### 1. 系统的输入数据

定义：在DFD中，从外部项到系统的数据流和只是某一处理逻辑的输入数据存储所含有的数据叫系统的输入数据。由定义可知，输入数据可以容易地从DFD中找到。图3中的输入数据是顾客订货单、厂商的发货单、向厂商的订货项目、顾客付款、厂商的付款通知。后面将证明输入数据是数据库内涵的组成部分。

#### 2. 系统的更新数据

定义：在数据流程图DFD中，如果某一数据存储所连的数据流线都是双箭头的，则该数据存储叫系统的更新数据。更新数据也是所建数据库的内涵部分。图2中的更新数据为配件库存。

#### 3. 数据库数据内涵的确定

定理：数据库的数据内涵一定包含在系统的输入和更新数据中，是它们并集的一个子集。

证明：用排他法来证明该定理。在DFD中，除了系统的输入数据和更新数据之外，还有的数据客体如下：

(1) 数据存储：它既是某一处理逻辑的输入又是某一处理逻辑的输出，这样的数据存储只能是业务处理过程中的中间数据结果，显然它不能作为原始数据库的内容，体现了数据库设计无冗余性这一特点。

(2) 数据流：它是某一处理逻辑的输出且到另一处理逻辑作为输入，它的作用和(1)中的数据存储类似，反映组织内部业务部门处理的顺序，也不能作为数据库设计的内容。

(3) 数据存储或数据流：它仅是某一处理逻辑的输出，反映系统的处理结果，故不能作为原始数据库的数据内容。

排除了上述三种情况的数据存储和数据流后，在DFD中剩下的只是系统的输入数据和更新数据，因此系统的输入和更新数据是数据库设计内涵的一个超集。事实上(1)(2)(3)中的数据都是由系统的输入数据和更新数据经一定的处理过程后得到的，如果把它们作为数据库的内容，则数据之间存在冗余，是设计要求所不允许的。原定理得证。

### 四、数据对象的分析

我们把上述的系统输入数据和更新数据统称为语义实体。为了进行下一步的概念模式的设计，必须要对语义实体作进一步的分析。我们以图2中的语义实体订货单为例来说明数据对象的分析。由数据字典得到订货单数据结构如下：

订货单标识  
订货单编号order-no  
日期data  
顾客详情  
    顾客名称CName  
    顾客地址caddress  
    电话ctele  
    开户银行Cbank  
    帐号Cbank-no  
配件详情  
    配件编号part-no



配件名称part—name

规格part—speci

订货数量part—amount

显然这种结构是不能作为数据库中的独立数据存在的,因连最起码的范式要求都不能满足,须对其数据元素作相关分析,重点是各数据元素之间的依赖关系。

(1) 找出语义实体的语义关键字。如果语义实体中的某一数据元素或某些数据元素的组合决定了实体中的所有数据元素,即决定了该语义实体,则此元素或多个元素的组合就叫语义关键字。如订货单语义实体中的语义关键字是订货单编号和配件编号的组合,其他一切数据元素都依赖于它们的组合。

(2) 找出数据元素之间的语义依赖关系。寻找语义实体中数据结构内部数据元素之间和数据结构与元素之间的依赖关系,必须针对应用环境作具体的分析。语义实体的容量有限,用户和设计人员能列出这些依赖关系。如订货单语义实体中的此类依赖关系为:

```
cname——caddress    cname——ctele  
cname——cbank    cname, cbank——cbank  
                    —no  
part—no——part—name    part—no——  
                        part—speci  
cname part—no——part—amount
```

对于每个语义实体作一个表,分别列出其语义关键字及所有非关键字的元素依赖。至此我们认为需求分析特别是用户的数据需求分析已经完成,积累了文字和图型资料:DFD流程图,数据字典、语义实体集,语义实体内的依赖关系等,为以后的设计工作做了充分的准备。

## 五、结束语

本文讨论了结构系统分析法在需求分析中的应用,对设计数据库有一定的帮助。用此方法得到的语义实体,可以直接用到概念模式的设计中,具体的方法可以详见文献[1]。利用一定的算法将每一语义实体转换成对应的局部概念模型[5],进一步的设计工作及分析可见参考文献[5]。需要指出的是由本方法得到的需求分析的结果,只反映数据的静态特性,如果要了解数据的行为特性,则必须对数据流程图中的处理逻辑作详尽的分析,即使如此,也不能对行为特性做完备的分析,必须进一步了解用户的业务结构,是一件很复杂的任务。

总之,用SSA法进行需求分析,具有直观、完备和易于交流的特点,它是沟通用户和设计人员良好的中间模型,保证了数据库的设计具有坚实的前提条件。

### 参考文献

1. 姚卿达,《数据库设计》,高等教育出版社,1987.4
2. 王勇领,《计算机数据处理系统分析与设计》,清华大学出版社,1986
3. P.P.Chen. A Logical Design Methodology for Relational Database using the Extend Entity—Relationship model, Computer survey, vol.18, no.20 June 1986, P197-222
4. 陈自安,一种简单有效的数据库设计方法综观,《第四届全国数据库会议录
5. 匡文学,数据库逻辑设计方法的研究和CADEER工具的实现,海军工程学院硕士论文,1989.2

# 为FORTRAN语言增加系统、图形功能

西北工业大学 叶天彤

**摘要** 本文提出了增强FORTRAN语言的系统操作和图形操作功能的一些实用的方法和编制技术。主要详述其原理并略举几例,以便用户掌握后可对FORTRAN语言或其它高级语言自由地增加一些用户需要的功能。

## 一、FORTRAN语言与汇

### 编语言的接口

FORTRAN语言提供了CALL语句可实现与外部模块文件的联接,而汇编语言具有各种系统操作功能和屏幕控制能力,因此,只需编制出这些需要的功能子模块,并将它们与FORTRAN程序连接即可实现所需功能。

FORTRAN语言与汇编语言的参数通讯是通过堆栈来实现的,最低两字节用于保存BP的值,其次四个字节为返回到调用程序的地址,再在其上按逆序存放参数地址。例子有子模块dot,参数有x、y、c,即在(x,y)点画颜色值为c的点。它们的地址各自在栈中的位置如图1。由于FORTRAN语言不允许传递值参,所有参数都必须用变量,所以每个参数固定地在栈中占有四个字节。因此,在汇编语言子模块中只要到栈中的相应单元取出参数即可。后附的程序一为一个让FORTRAN实现BASICA的INKEY函数的程序,程序二为画点程序(中分辨率图形方式),程序三为设置显示器显示方式程序。

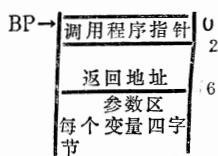


图1

## 二、实现过程

按所需功能用汇编语言写出子程序,其

中过程名必须被说明成PUBLIC型,程序被说明为FAR型。然后用汇编语言编译程序编译成目标文件,并用LIB·EXE文件将其扩充到库文件中。例如:LIB FORTRAN·LIB+INKEY·OBJ,若想仍保留该目标文件,则应把加号改为乘号,即:LIB FORTRAN·LIB\*INKEY·OBJ,若想取消库中的一个模块,如POINT·OBJ,可输入LIB FORTRAN·LIB-POINT·OBJ命令即可。一个功能子模块扩充到库后,在以后的程序中可用CALL随意调用。

## 三、功能子模块编制技

### 术及应注意的问题

**1. 寄存器保护:** 在汇编子程序编制时,应注意对要用到的寄存器加以保护,在程序开始部分应把将会影响到的寄存器的值压入堆栈,在退出前恢复。

**2. 参数返回:** 返回参数(出口参数)可象入口参数一样看待,在汇编语言中把要返回的值放到栈中与该变量对应的位置就行了。

**3. 速度影响:** 由于作图时计算点位置过程占了很大的比重,要注意改进算法以提高作图速度。例如在画斜线时可用半值法,画图时正弦、余弦可用泰勒展开式的前三项代替等。

**4. 中断的避免:** 在子模块编程过程中,应尽量避免引用DOS中断(尤其是INT 21)这是因为:首先现在许多软件都可常驻内存。并对一些相应的中断进行了修改,为确保程

序能在以后长期安全地使用，故最好少用中断；其次，这是一个更重要的原因，即DOS中断的执行速度太慢，这一点可以通过比较用中断10和直接向屏幕显示缓冲区写字符两种显示方法明显地看出。第四部分提供了几个控制单元及其意义，可供参考。附例程序4即为改进的MODE程序。

## 四、几个重要的内存控制单元

IBM—PC/XT内存中有一些固定的单元用于存放机器运行所需的一些重要信息，地址0000—0400H用于存放中断向量地址，从400H开始的一些单元存放机器状态，现列出几个重要的单元如下：

40:17H该字节用于检测几个功能链状态，0位至3位分别对应右shift键、左shift键、ctrl键、Alt键，按下为1，放开为零，第4位为scroll Lock键（滚动为1），第5位为Num Lock键（数字方式为1），第6位为Caps键（大写为1），第7位为ins键（插入为1）。

40:57H该单元记录屏幕显示方式，其中存放的值对应于各种显示方式号。

40:1AH该字存放键盘缓冲区的以40H为偏移量的起始地址。

40:4EH该字用于存放图形缓冲区的起始偏移地址。

40:66H该单元低四位控制中分辨率方式下的背景颜色。

用户可以在程序中直接对这些单元读写以实现对机器的控制。

## 五、增加中断功能

由于每添加一个模块都必须重新编一个汇编程序，这对懂汇编语言的用户当然不成问题，但对于不知道的用户就有点困难了。为解决这一问题，可考虑为FORTRAN增添一个实现直接控制中断的语句，由于程序中

中断地址可根据中断号直接计算得出，故可避免修改中断的干扰。假设模块各为Intr，调用命令为Intr(IntNo, Regs)；其中IntNo为中断号，Regs为一数组，记录着中断的入口参数、出口参数。中断入口地址为：中断号\*4—1，占四个字节，前二字节为低址，后二字节为偏移地址，直接进入运行即可。

以上仅为本人在应用上的几点经验体会，提出供用户参考。利用这些方法，本人已对FORTRAN、PASCAL库进行了扩充，增加了七十多个模块，经实践证明这些方法是有效可靠的。

```

程序一: INKEY.ASM    读键盘
调用格式: CALL INKEY(CHAR)
public inkey 说明为PUBLIC型
kbd
segment byte public
assume cs, kbd

inkey
proc far
push bp          ; 必需的栈保护
mov bp, sp
mov ah, 08h
int 21h          ; 读键盘, 不显示
mov si, [bp+6]; 返回ASCII码
mov [si], al
mov sp, bp
pop bp
ret 4             ; 1*4

inkey
endp
kbd
ends
end

程序二: POINT.ASM    作点
调用格式: CALL POINT(X, Y, C)
public point
setpoint
segment
assume cs, setpoint

point
proc far
push bp
mov bp, sp
mov si, [bp+14]; x 座标
mov cx, [si]
mov si, [bp+10]; Y座标
mov dx, [si]
mov si, [bp+6]; 颜色值
mov al, [si]

```

# IBMPC/AT机硬盘文件及剩余空间的加密与保护方法

中国科学院上海生物化学研究所 于 甫

**摘要** PC/AT机硬盘的文件和剩余空间是可以被任意读写和删除的,为了防止其他用户对其任意读写,本文介绍一种对硬盘子目录和剩余空间的加密保护方法。

## 一、DOS文件在硬盘的存储结构

不管是硬盘还是软盘DOS对文件目录的管理方法都是一样的,每个文件的目录都是占用32个字节。每个32字节的字段可分解:

字节: {0-7}	文件名
8-10	文件扩展名
11	文件属性
12-21	(保留)
22-23	上次存取文件的时间
24-25	上次存取文件的日期
26-27	起始簇
28-31	文件的长度

```

mov ah, 12
int 10h          ; 作点
mov sp, bp
pop bp
ret 12           ; 3*4
point endp
setpoint ends
end

程序三: MODE*ASM 置显示方式
调用格式: CALL MODE(I)
text segment byte public
assume cs, text
public mode
mode: proc far
push bp
mov bp, sp
mov si, [bp+6]   ; 取得方式值
mov ch, [si]
mov al, ch
mov ah, 00h
int 10h          ; 显示方式
mov sp, bp
pop bp
ret 4            ; 1*4
mode endp
text ends
end

程序四: MODE*ASM 置显示方式
调用格式: CALL MODE(I)
text segment byte public
assume cs, text, es, nothing,
ds, text
mov ax, 12
public mode
jmp mode
mess db 2ch, 28h, 2dh, 29h, 2ah,
2eh, 1eh;
对应0-6方式控制值
proc far
push bp
mov bp, sp
mov ax, cs
mov ds, ax
mov ax, 0000
mov cs ax        ; 0段
mov ax, 1097     ; 控制显示方式单元地址
mov di, ax
mov si, [bp+6]   ; 取方式值
mov al, [si]
cmp al, 06h      ; 大于6退出
ja stop
mov ah, 00h
mov si, ax
lea dx, mess
mov bp, dx
mov ah, ds, [bp+si]
mov es, [di], ah; 置方式值
mov sp, bp
pop bp
ret 4            ; 1*4
mode: endp
text ends
end mode

```

, 注意: 用此方法改变显示方式前应清屏



其中文件的属性字节每位是这样定义的，第七、第六位不用，第五位是更改位，第四位是子目录位，第三位是卷标位，第二位是系统位。第一位是隐含位，零位是只读位。DOS使用文件分配表FAT来给文件分配其在磁盘上的存贮空间，并跟踪其空白扇区。文件分配表中的每一项相对于盘上某一特定的簇，文件的范围通常跨过很多簇，一个文件的目录项包含起始簇和文件大小，通过对FAT的查找DOS找到对应于起始簇的位置并找到对应的下一簇的簇号，一直找下去直到此文件簇的结束标记FFFF。为可靠起见所有磁盘都保存两份FAT，在PC/AT的20M硬盘内FAT数据项的长度不同于软盘，它采用16位。每个簇代表4个扇区，其硬盘的逻辑扇区的分配是这样的：

第一个FAT在硬盘中位置 1H—29H  
第二个FAT在硬盘中位置 2AH—52H  
根目录的位置 53H—73H

## 二、对硬盘的特殊管理方法

由于PCDOS的文件管理对用户来说是透明，任何用户只要知道文件的名称就可以对文件进行读写操作。那种改文件名称为其它字符（ASCII码值大于127的字符）和改文件属性为隐含只读等方法是不够的。为了对用户的文件安全进行保护以及对你的硬盘剩余空间进行保护，笔者设想了两种方法并编写了两个简单的实现程序。

### 1. 对子目录的加密法

对子目录的加密都是为了使一般用户不

能对子目录下的文件进行正确操作。笔者采用的方法是不让其正确移到子目录。我们知道子目录文件的目录项28—31字节是文件的长度，其值都为00，26—27字节为文件起始簇号，12—21字节也都为00不用，如果我们编写一个程序将起始簇号改掉，将密文写在文件长度或备用字节上。这样其他用户只要不知道子目录的起始簇号就不能进入其目录对文件进行操作，从而达到加密的目的，当你要使用时，只要通过解密程序就可正确使用了，使用完可再运行加密程序重对子目录上锁。

笔者所编的加密程序与解密程序为同一个程序，只是通过判定起始簇是否为0000，而对子目录进行解密或加密处理，简单实用。具体作法加密时将文件长度的前两个字节与起始簇的两个字节对换，然后文件长度的前两个字节与文件名称的前二个字符进行逻辑异或运算。结果放在文件长度的前两个字节中。解密过程略，框图如图1。

### 2. 对硬盘剩余空间的加密

在公用的计算机里，计算机硬盘存有很多大家公用的软件，用户在使用这些程序的时候。将运行中产生的中间结果或者其他自己的数据拷贝到硬盘，使用完后又不删除，这样日积月累，硬盘的空间就会被很多小文件分割成许多的小块，将硬盘空间占光，并且很多小文件即使不到一簇也占用一簇，不利于空间的有效利用，如果由我们定时删掉这些文件，对有用与无用的文件进行筛选将耗费我们很多时间与精力。为此笔者设想将所有还没有使用的文件分配链表的簇

# 基于Smalltalk语言的面向对象程序设计

国防科技大学计算机系 朱海滨

**摘要** 面向对象的概念形成于七十年代末期,十年来,这一概念从计算机语言、数据库管理到操作系统的组织和系统结构的研究已经几乎渗透到了计算机领域中的各个方面,特别是在程序设计和软件设计方法学中,面向对象的设计方法甚为深入和广泛。仅已经开发的面向对象的程序设计语言就有十几种之多。所以,有人认为,面向对象的设计方法在八十年代程序设计领域中的地位 and 影响将超过七十年代该领域中的结构程序设计。

Smalltalk—80系统是一个基于图形的面向对象交互式程序设计环境它以Smalltalk语言为核心,配以支持图形处理的语言工具和多窗口系统及弹出型菜单,从而成了一个具有友好界面的面向对象交互式程序设计环境。为了学习和跟踪国外先进技术,我们在VAX/VMS系统上实现了Smalltalk—80系统并进行了一些必要的删节和扩充。本环境曾在武汉大学软件所国家软件工程实验室验收会议上展示,受到验收委员会的好评。本文在描述Smalltalk语言基本功能的基础上非形式地给出了基于Smalltalk语言的面向对象程序设计的定义和描述。

## 一、基本思想及对语言的要求

### 1. 基本思想

面向对象的程序设计方法基于Parnas的信息隐蔽和Cuttag的抽象数据类型概念,被认为是八十年代的结构程序设计。它把系统中所有资源如,数据,模块以及系统都看成

对象,每个对象把一个数据类型和一组过程封装在一起,使得这组过程了解对这一数据类型的处理,并在定义对象时可以规定外界在其上运算的权限,使用这一方法,设计人员可以依照自己的意图创建自己的对象,并将问题映射到该对象上。这一方法直接,自然,可以使设计人员把主要精力放在系统一

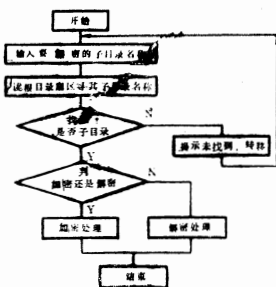


图1

链接起来,并产生其文件的目录存于根目录下,使得硬盘的可用空间为零,或者为一个比较适当的数目,这样其他用户用机器时就没有很多硬盘空间为其使用,只好用软盘或

删掉无用的文件,这样我们对于硬盘文件的管理就比较方便了,也有利于我们有时装入大容量的应用软件。具体实现方法是模拟子目录的生成,程序的流程框图如图2,实现程序解密程序略。

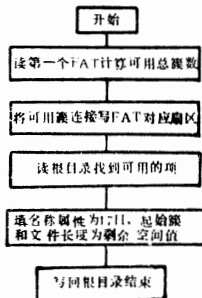


图2

级上，而对细节问题可以较少地关心。

## 2. 对语言的基本要求

要支持这种思想，面向对象的语言必须具备以下特征：

(1) 信息隐蔽：以受限的外部接口，限制外部与对象内部的联系；

(2) 模块及封装功能：为信息隐蔽提供具体的实现手段；

(3) 抽象：为问题定义提供良好的支持；

(4) 代码共享：避免重复的代码设计，真正使面向对象的方法成为有效；

(5) 灵活性：支持对象的主体特征，使得对象可以根据自身的特点进行功能实现。

Smalltalk语言正是以此为目的而设计的，它以其对象、类、继承机制和动态汇集等分别支持了以上特征；（详见第四）

## 二、Smalltalk语言的功能分析

作为运行环境，Smalltalk语言由两部分组成：即，虚拟机(Virtual Machine)和虚拟象(Virtual Image)、虚拟机是一个虚设的用软件实现的面向Smalltalk语言的“硬设备”，用于支持虚拟象的存在并解释虚拟象的功能实现，Smalltalk语言的基本功能全部反映在其虚拟象上。

类在语言中形成功能模块，如果将Smalltalk语言比作一幢楼房，那么类就是建造这座楼房的砖石，笼统地说，Smalltalk语言的虚拟象就是一个Object类，其具体语言功能由Object及其子类支持，下面就分别阐述这些子类的功能设计（各种功能相关的类由图1给出）。

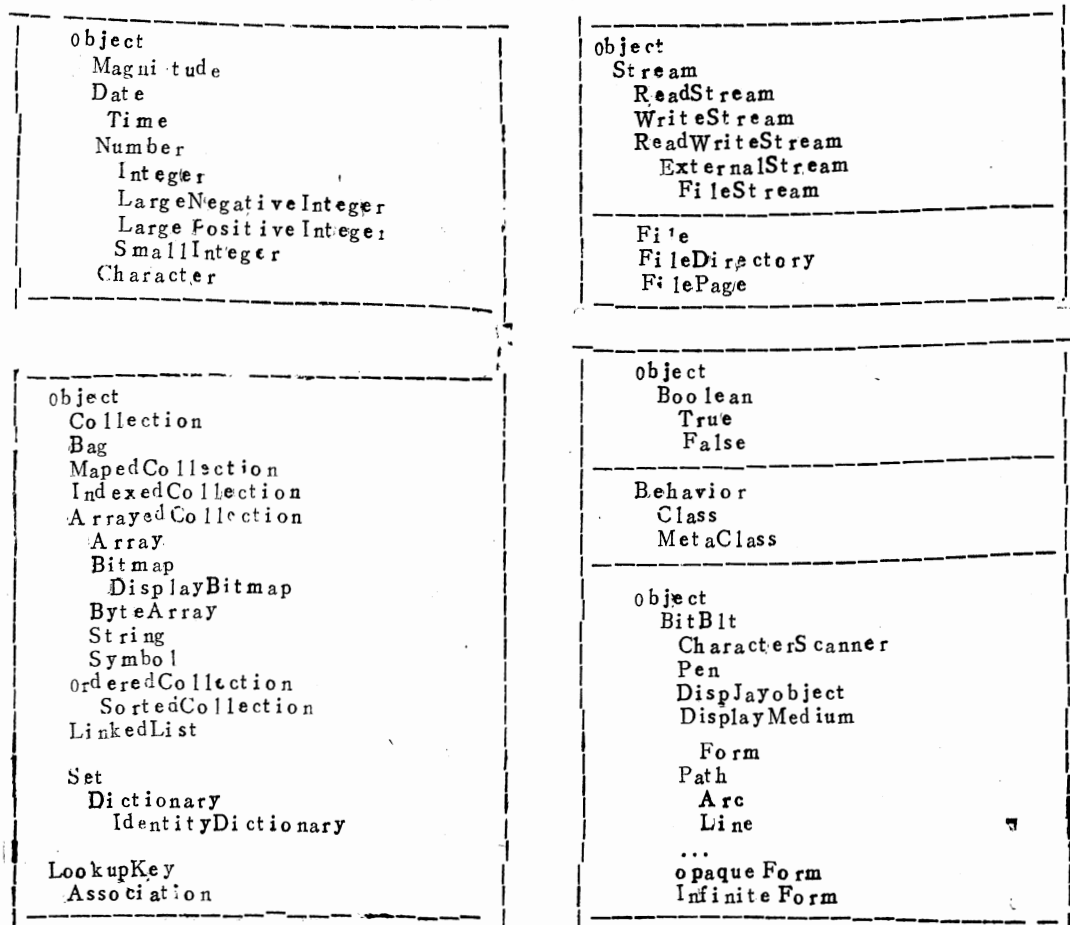


图1

## 1. 系统的总根基Object

Object类集中了所有对象的共同功能。该类所提供的功能可为系统中每一个对象所共享,从而减少了系统的信息冗余。Object提供以下几种功能:

(1) 功能测试:用于解决每个对象可能遇到的功能测试问题,比如,查询一对象是否响应某一消息、查询某一对象属于哪一类等等。

(2) 对象处理:主要支持对象的拷贝,对象的比较等功能。

(3) 错误处理:用于支持当对象在接收或处理消息时产生错误的处理,如通告错误信息,使程序可以清晰地结束等等。

(4) 消息变量的处理:通过消息变量,可以使程序设计更加灵活,方便。

## 2. 数学运算

任何高级程序语言都对数学运算给以支持,Smalltalk语言也是如此。它通过Object的一个子类Magnitude实现了这一支持。Magnitude类支持人们处理事物时所遇到的具有量化特征的问题处理。如时间量、空间量及数量等。由于各种不同量都有自己的特点,所以,Magnitude类也实现了这些量的共同功能,各量的特殊功能仍由具体的子类支持,

- 时间类和日期类:用于支持一般管理工作中所遇到的时间和日期的查询、计算、和变换等功能。

- 数量类:提供完整的数学运算功能,其中包括算术运算和函数运算。

- 字符类:提供完整的字符比较、运算、串操作等功能。

## 3. 数据结构

对数据结构支持的好坏直接影响着它的能力和实用性,Smalltalk语言以其独特的方式为丰富的数据结构提供了强有力的支持,它既支持了简单数组,又支持了诸如集合、队列、栈等复杂的结构。之所以将这些数据结构都集中于Collection类中,是因为

各种数据结构无论如何变化,都可以看作是许多数据类型的收集,Smalltalk语言将它们进行了统一,具体的数据结构只要在Collection的基础上加以特定的约束和精化描述就可以获得。

Collection类抽象了收集的统一功能和特征,它其中的元素既可以是有序的,也可以是无序的,它的大小既可以是固定的,也可以是动态可变的。

Collection的四个子类SortedCollection, OrderedCollection, LinkedList, 和ArrayedCollection支持了其元素有序的收集。其中,SortedCollection表示那些其元素按内定的或明确指出的函数进行排序的有序收集,OrderedCollection表示那些只能从其两端存取元素的收集,由它可以支持如栈、队列等数据结构的实现;LinkedList表示那些元素靠指针组织在一起的收集,支持链表的实现,ArrayedCollection表示那些可以用整数定位存取其元素的收集,可以支持一般的数组实现,并在功能上有所扩充,它是一种不要求其元素是同一类型的动态数组。

无序的收集主要由Bag、Set及其子类Dictionary给以描述和支持,其中,Bag表示一组任意事物的收集,并且其元素可以重复存在;Set表示的是一组其元素不允许重复存在的任意对象的集合;Dictionary表示一组二元关系的集合,其中一个二元关系由键字和值两项组成,它提供了根据键字或值查询、修改其二元关系的功能。

另外,在Collection的其它子类中还提供了串、集合等结构的模式匹配功能,这使该语言的能力得到了增强。

## 4. 文件组织

Smalltalk利用Stream和File类分别支持内、外部文件的实现、“流(stream)”的概念取自水流,人们知道,水流中的某一点在流动过程中其位置是不断变化的,语言中的“流”则是指在一组数据的检索中,每一



时刻所检索的数据是不一样的。Smalltalk语言的基本数据结构是收集Collection，Collection提供了直接存取和枚举两种方式，在一个收集上建立一个流就将两者结合成位置访问，也就是为每个流结构设立一个动态索引，每进行一次访问，该索引就变动一个单位。

流概念的建立支持了文件的实现，一个流便形成了一个内部文件，传统语言中文件的读写窗口可以用其索引实现。Smalltalk语言中的流结构提供了三种索引方法，第一种是计数，用于顺序存取，第二种是指针，既可以顺序存取，又可以跳跃存取；第三种是利用随机数进行存取，直接支持对随机事件的模拟。

### 5. 类的组织

在Smalltalk语言中，语言功能是以类为基本单位的。那么，如何来组织这些类呢？Smalltalk语言用Behavior及其子类Class和MetaClass来完成这一任务，它们将类作为结构一种特殊的对象，支持类的创建、修改、删除等功能。

其中，Behavior描述了类的最小行为状态，它既定义了类的基本结构，和对类所执行的基本操作。Class和MetaClass区分了两种不同的类，Class是对一般对象的抽象，而MetaClass则又是对类的进一步抽象。

### 6. 图形处理

支持图形的处理是Smalltalk语言的一特色，这也是一个好的软件环境所应具有的功能，Smalltalk语言的BitBlt、Bit-map、Displayobject等及其子类给出了图形处理的完整支持。

(1) 图形的表示和存贮：表示图形的基本机制是像素，在Smalltalk语言中，像素是一定大小的点阵，像素越小，所表示的图形越精确，利用对象的概念存贮一个图形非常方便，Smalltalk语言中利用像素位图Bitmap来示和存贮图形。

(2) 图形的操作：一个图形及其中一块区域在Smalltalk语言中都作为form看待，图形操作的基点是“位块传送”BitBlt (Bit Block Transfer)，Form和BitBlt支持了图形的位置变动、重叠、覆盖和拷贝等功能。

(3) 图形的显示：DisplayObject及其子类实现这一功能。

(4) 其它支持图形的工具：为了使图形处理更为方便，我们还实现了许多支持特定图形的类，如Point、Rectangle、Pen、和Curve等，这些类可以使程序员在程序中更方便灵活地建立和操纵图形。

## 三、Smalltalk语言的结 构及其功能组织

类的设置使得Smalltalk语言的实现非常简单，易行。Smalltalk语言其实就是其所有类功能的有效集中。这些类按以下原则组织：

(1) 语言中所有的类在概念上按其继承关系形成树形层次结构（见图1），Object类代表整个系统的所有可用对象，Object类的直接子类代表系统的各种语言功能。

(2) 每个类可作为一种特殊的对象来看待，这类对象的共同特征由Class类来描述。

(3) 每个类都是Object类的子类

(4) 每个元类对应于一个类，是用于支持这个类作为对象应具有的行为实现；

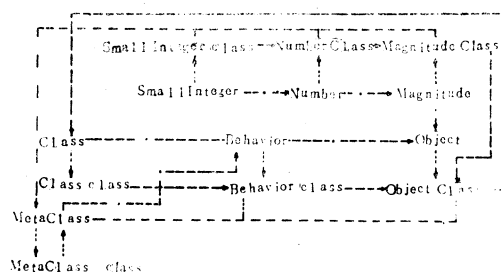


图2

其中“—→”表示类的直接继承关系，A—→B表示A直接继承B；

“ $\rightarrow$ ”表示元类的直接继承关系,  $A \rightarrow B$ 表示元类A直接继承元类B;

“ $\rightarrow$ ”表示实例关系,  $A \rightarrow B$ 表示A是B的实例

- (5) 每个类都是其元类的实例;
- (6) 每个元类都是Class类的子类;
- (7) 每个元类都是MetaClass的实例;
- (8) Class类和其超类支持所有类的共同行为;

(语言的组织如图2所示, 以Smalltalk-1k类为例)、

## 四、基于Smalltalk语言的面向对象程序设计

Smalltalk语言可以支持面向对象的程序设计与其功能和组织结构分不开的。基于以上的Smalltalk语言的基本功能, 以该语言为基础的面向对象程序可以如下定义:

程序:  $=$  程序段 | 类创建. 程序段 | 程序. 类创建. 程序段

程序段:  $=$  对象创建. 消息序列 | 程序段. 对象创建. 消息序列

对象创建:  $=$  类 创建消息

消息序列:  $=$  对象 消息 | 消息序列 对象 消息

对象:  $=$   $\langle MS, DS, MI \rangle$

其中MS是对象受理的操作集合;

DS 是对象的存贮或数据结构;

MI 是对象受理的操作名称集(即消息名集, 也称对外接口)。

在实现Smalltalk语言时, 为了减少冗余并实现代码共享, Ms和MI两部分对于一个对象是虚设的, 它们由其所属类中得到实际的支持。

消息:  $=$   $\langle$  消息名, 变元组  $\rangle$

其中, 消息名  $\in \cap MI$ ;

变元组:  $= \langle$  变元, 变元, ..., 变元  $\rangle$ ,

变元:  $=$  对象。

类:  $= \langle INH, DD, OI, ITF \rangle$

其中 INH是类的继承性描述;

DD 是数据结构描述;

OI 是操作集合的具体实现;

ITF是统一的对外接口。

从上面的定义可以看出, 对象是对操作表示和数据表示两种功能的抽象。当DS $\rightarrow$ 1时, 对象 $\rightarrow$ 一组纯过程; 当MS $\rightarrow$ 1时, 对象 $\rightarrow$ 一个纯数据结构, 如果真是这样, 对象就成为废品一堆, 无任何作用。所以, 对象使得数据及相应的操作不可分, 实现了数据封装的功效。

另外, 每个对象的功能实现是独立的, 其具体功能实现直到执行时才可确定, 即动态汇集。支持了程序设计的灵活性。

类是对一种对象的抽象, 它将该种对象所具有的共同特征(包括操作特征和存贮特征)集中起来, 由该种对象所共享, 在系统构成上, 则形成了一个具有特定功能的模块和一种代码共享的手段。

类实现中的继承机制是一种独特的性质, 它使得子类可以继承其前辈类的特征和能力, 类的继承具有传递性: IF (C2 INH C1) & (C3 INH C2) THEN C3 INH C1。其中INH表示“继承于”。

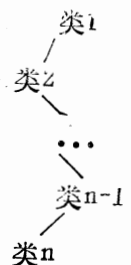
为了清晰起见, 类可以重新定义如下:

类n:  $= \langle \bigcup_{i=1}^n$  类i本身的数据结构描述,

$\bigcup_{i=1}^n$  类i本身的操作实现,

$\bigcup_{i=1}^n$  类i本身的对外

接口  $\rangle$



从以上的定义可以看出,Smalltalk语言支持了三层抽象级别:即,对象是对一般事物的功能及物理状态的抽象;类是对同一类对象的抽象;前辈类是对子孙类的抽象。

根据以上的程序定义,用Smalltalk语言进行程序设计过程如下:

1. 问题定义;
2. 解决该问题所需要的类是否全部存在?若是,则转4;
3. 创建(定义)对象;
4. 创建不存在的类;
5. 编制消息序列(这一步的工作要比一般的程序设计简单得多,它的工作主要是建立对象间的联系);
6. 调试运行程序。

#### 五、结束语

以上给出了面向对象程序设计的基本概念和基本过程,可以看到,这一方法具有其新颖独特的风格,具有强大的生命力。目前,世界各国都在热心这种程序设计风格的研究,国外有的大学已经将面向对象的程序设计作为计算机专业学生的必修课程并制订了专门的教学计划。

以上对利用Smalltalk语言进行面向对象的程序设计给出了非形式的描述,以期给出了一个初步的认识,以此为基础,可以在整个软件生存周期中引入这一风格,进一步提高软件生产率。另外,数据库管理、知识表示、新型语言的设计、操作系统组织和体系结构研究都在寻求面向对象这一风格的支持,由上可见,学习、使用、开发Smalltalk语言是非常有意义的且有成效的。

本文源于笔者硕士课题的工程实现,在

此谨向我的导师陈火旺教授致以诚挚的谢意,并向给予合作的其它同志致谢。

#### 主要参考资料

1. Adele Goldberg and David Rodson, Smalltalk-80: The Language and Its Implementation, Addison-Wesley, Reading, Mass, 1983.
2. Stephen T. Pope, Adele Goldberg And L. Peter Deutsch, "Object-Oriented Approaches To the Software Lifecycle Using the Smalltalk-80 System as a Case Toolkit," 1987.
3. Bhaskar K.S., J.K. Pocol and J.L. Beug, "Virtual Instruments, Object-Oriented Program synthesis," Proceedings of the ACM Conference on Object-Oriented Program Systems, Languages and Architectures(OOPSLA), 1986.
4. Jim anderson and Barry Fishman, "The Smalltalk Programing Language", BYTE, MAY, 1985
5. Geoffrey A. Pascoe, "The Elements of Object-Oriented programing," BYTE, Feb., 1986
6. Alan Snyder. "Encapsulation and Inheritance in Object-Oriented Programing Languages," OOPSLA, 86 Proceedings.
7. Stephen S. Yan, Jeffery P. Tsai, 软件设计技术之概观, 计算机科学, 4, 1987, PP10—18.
8. 朱海滨, 对Smalltalk-80系统的分析及其核心环境的实现, 硕士学位论文, 国防科技大学, 4, 1988.

# 使用dBASE III 编程的几点技巧

暨南大学 罗伟其

**摘要：**在应用dBASE III编写程序过程中以及有关的教学工作中，笔者曾对遇到的某些问题加以分析研究后，作了技巧方面的处理和改进。现将把有关方法介绍如下。

## 一、对使用TOTAL命令引起的不足之处的改进

TOTAL命令的格式是：

```
TOTAL ON <关键字段> TO  
<目标数据库>[<范围>][FIELDS<字段表>][FOR/WHILE<条件>]
```

本命令是针对已经按关键字段索引过或做了排序的源数据库进行的，它把源数据库文件中关键字段具有相同键值的所有记录的数值字段的内容进行求分类和，并将求和结果产生一目标数据库，其中非数值型字段的内容，就是源数据库中与该关键字段值相同的那一组记录中的第一个记录的对应字段的内容。由于它具有快速方便，容易实现的优点，大部分人在做分类求和时，都普遍使用本命令来实现。

但因为由TOTAL命令产生的目标数据库的结构与源数据库结构完全相同，所以必然会引起下面两点不足。

1. 对源数据库中的数值型字段为避免在求和时出现数据溢出，通常的办法是在建立结构时各数值型字段的宽度均按最大可能发生的求和结果需要的宽度设定。这样做虽解决了溢出问题，但造成了存储空间浪费，而且数值型字段越多，浪费就越大；记录越多，浪费也越多。

2. 在许多情况下，对源数据库做分类求和时，希望在目标数据库中只保留部分有关的字段。但也因目标数据库与源数据库的结构相同，而不能实现。这样不仅因字段多余造成存储空间浪费，而且降低了执行的速

度。

由上述两方面造成的存储空间浪费之和，在许多应用系统中是相当大的。笔者在开发财务管理系统时深有体会。

如何解决这些不足？可考虑采用下面的方法来实现。

我们知道APPEND FROM<文件名>命令是从一个标识的<文件名>数据库文件中复制全部记录到当前数据库，而且只复制这两个数据库中字段名和字段类型相同的那些字段内容，不要求字段的宽度相同。我们可以利用这一功能特点来帮助解决问题，具体的做法如下：

首先，为了避免存储空间浪费，对源数据库结构按记录实际所需的字段宽度来设定，无需考虑求和目标数据库所需的宽度。为了说明方便，给出了一个例，在例中设源数据库文件名为ABC.DBF，其结构如图1所示。

第二，建立一个用来做TOTAL分类求和的中间数据库文件结构。这个结构按目标数据库所需的字段以及求和所需要的最大宽度来设定。在本例中设中间数据库文件名为XYZ.DBF，其结构见图2，它比ABC.DBF数据库少了目标库中不需要GH和XM两个字段，数值型字段的宽度按实际需要加宽了。

```
Structure for database , b, ABC.dbf
Number of data records:      11
Date of last update         : 01/01/80
Field Field name Type      Width Dec
  1  BMH      Character      3
  2  GH       Character      6
  3  XM       Character      8
```

4	A1	Numeric	6	2
5	A2	Numeric	5	2
6	A3	Numeric	5	2
7	A4	Numeric	6	2
8	A5	Numeric	5	2
9	A6	Numeric	5	2
10	A7	Numeric	6	2
11	A8	Numeric	5	2
12	A9	Numeric	6	2
13	A10	Numeric	5	2
14	A11	Numeric	7	2
••Total••			79	2

图1

Structure for database, b, XYZ.dbf  
Number of data records, 0  
Date of last update, 01/01/80

Field	Field name	Type	Width	Dec
1	BMH	Character	3	
2	A1	Numeric	8	2
3	A2	Numeric	7	2
4	A3	Numeric	7	2
5	A4	Numeric	8	2
6	A5	Numeric	7	2
7	A6	Numeric	7	2
8	A7	Numeric	8	2
9	A8	Numeric	7	2
10	A9	Numeric	8	2
11	A10	Numeric	7	2
12	A11	Numeric	9	2
••Total••			87	

图2

第三,在求分类和之前,利用APPEND FROM<文件名>命令把源数据库的记录复制到中间数据库,然后对中间数据库做分类求和产生目标数据库。在本例中设目标数据库文件名为TABC.DBF。

最后,把中间数据库中的全部记录删除掉,只留下结构,以便下次需要做分类求和时用。

上述步骤的实现程序段如图3所示,同时由图4给出这一方法的关系图。

```
USE XYZ
APPEND FROM ABC
INDEX ON BMH TO IXYZ
SET INDEX TO IXYZ
```

```
TOTAL ON BMH TO TABC
ZAP
USE
```

图3

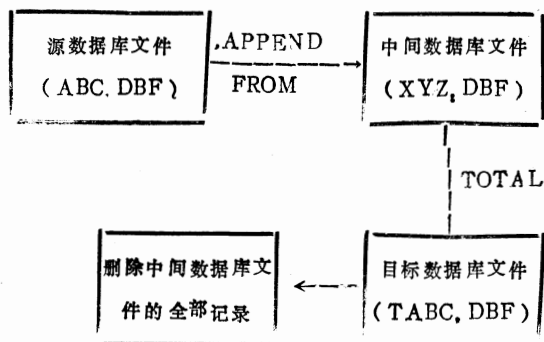


图4

## 二、SUBSTR函数在编程

### 中的一些应用技巧

SUBSTR函数的功能是从字符串中取出一子串,并可通过指定取子串的起点和长度而得到满足各种需要的子串。它的格式为:

SUBSTR(<'字符串'>,<起点>  
[,<子串长度>])

如果在编程中能灵活地利用SUBSTR函数的功能,将会使程序不仅变得简练,而且可以提高一些运行效率。下面作一些应用的介绍。

#### 1.在设计实线表格中的应用

在设计实线表格时,要用到一些制表符。这些制表符一般情况下都从区位码中取出。因为区位码的字符不能恢复输入,如需要由'-'符组成一直线段时,只好反复地键入这个制表符的区位码。这样做既使人感到厌烦,又使输入效率低,而且会因数错输入的次数,表格不整齐,需要调整修改,给程序调试带来点小麻烦。克服这个问题的通常做法是用循环语句来帮助组合线段。但因制表的线段并非全是一条直线,常常是带有'Γ'、'⊥'、'+'等线段,这时,用循环语句来组



合线段就感到不方便了。

用SUBSTR函数来组合各种线段是一种较好的方法。因为所有实线表格都要用到制表符，所以把制表符及一小段直线先建立一个内存文件，需要时调用，然后利用SUBSTR函数与内存文件取出的制表符一起组合成各种规格的线段。现用下面的例题给出这种方法的应用。

例。若需打印如图5的表格，可用下面图6给出的小程序来实现。其中BGFH是内存文件，它的内容见图7，其中L1是由10个'—'符组成，总长度为20个字符。

南方实业有限公司生产月报表


图5

#### • 用SUBSTR函数制表程序例

```

RESTORE FROM BGFH
L2=F4+SUBS(L1,1,10)+F6+SUBS(L1,1,6)+F6+SUBS(L1,1,8)+F6+SUBS(L1,1,8)+F6+SUBS(L1,1,10)+F3
L3=F9+SUBS(L1,1,10)+F7+SUBS(L1,1,6)+F7+SUBS(L1,1,8)+F7+SUBS(L1,1,8)+F7+SUBS(L1,1,10)+F5
L4=F1+SUBS(L1,1,10)+F2+SUBS(L1,1,6)+F2+SUBS(L1,1,8)+F2+SUBS(L1,1,8)+F2+SUBS(L1,1,10)+F8
SET PRINT ON
SET DEVICE TO PRINT
@ 1, 10 SAY '南方实业有限公司生产月报表'
@ 2, 10 SAY L1+SUBSTR(L1,1,18)
@ 4, 2 SAY L2
N=5
DO WHILE N<9
@ N, 2 SAY L3
N=N+1
ENDDO
@ N, 2 SAY L4
@ N+1, 2 SAY ''

```

```

SET PRINT OFF
SET DEVICE TO SCREEN
RETURN

```

图6

RESTORE FROM BGFH

LIST MEMORY TO

```

L1      pub  C  "  _____"
F1      pub  C  "[ "
F2      pub  C  "  _  "
F3      pub  C  "  ]  "
F4      pub  C  "  [  "
F5      pub  C  "  _  "
F6      pub  C  "  _  "
F7      pub  C  "  +  "
F8      pub  C  "  ]  "
F9      pub  C  "  _  "
F10     pub  C  "  |  "

11 variables defined,      62 bytes used
245 variables available,
5938 bytes available

```

图7

2. 恰当使用SUBSTR函数来代替一些条件或情况语句实现分支控制

在通常的情况下，如果程序中出现分支选择，则肯定会考虑到使用IF或CASE语句来实现，大家知道分支越多，程序越长，因判断而花费的运行时间也就越多。事实上，在某些情况下，出现分支不一定需要使用条件或情况语句，只要灵活使用SUBSTR函数就可以达到分支的控制。这样做不仅使程序变得简短，而且因不用判断而提高了运行速度。下面给出两个应用SUBSTR函数代替条件语句或情况语句来实现分支控制的例子就可以说明这一点。

例1 在给数据库的某些字段输入汉字时，为了提高输入速度，常采用代码的方式输入，然后由程序进行汉字的转换。但如果用IF语句来做就会不仅使程序变得冗长，而且在判断过程中多花了时间。现假设要给某数据库文件中的文化程度字段输入数据，为了进行对比，下面分别给出了用IF语句和SUBSTR函数编写的程序段，分别见图8和图9。



# 通道数可变的数据采集和过程控制接口

西南民族学院 杨宪泽

**摘要** 本文介绍一个设计构思与一般方法不同的在APPLE—Ⅱ微机上使用的数据采集和过程控制接口,其特点是不使用I/O接口芯片,器件少,经济实用,简单而易于制作,且对外有8个通道可任意作为输入通道或输出通道,

## 一、引言

按一般方法在APPLE—Ⅱ微机一个扩展插槽中设计D/A和A/D转换电路的原理框图如图1所示。

图1中,由于一般一个I/O接口芯片仅有两个并行口,要带A/D和D/A及两个多路器引脚不够,所以只能采用一个I/O接口芯片带A/D和多路器,另一个带D/A和多路器。译码器保证只有一个接口芯片导通,要么是模拟信号输入,要么是控制信号输出。

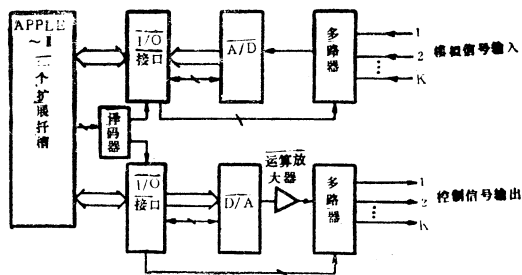


图1 A/D和D/A转换电路一般方法框图

这种设计方法所用器件多,符合微机接口设计规范,但从经济实用,制作难度上看并不是最佳选择。一般地讲,在A/D、D/A与微机总线之间使用接口芯片,可以使外设与主机隔离,不致于引起逻辑混乱;同时,接口芯片有一些特殊功能(诸如中断、计时计数等)使程序编制灵活。如果我们并不使用接口芯片的特殊功能,设计中保证逻辑关系正确,就能省去I/O芯片及一些辅助器件,使电路简单易制,既经济实用,又有

利于控制系统的稳定。

## 二、接口设计的电路原理

按上述构思设计的接口电路如图2所示。图2中,电路包括多路开关CD4051 ( $IC_5$ ), A/D转换器AD7574 ( $IC_2$ )、D/A转换器DAC0832 ( $IC_3$ )、集成运算放大器F005 ( $IC_4$ ), 隔离二极管2AP1、三态输出总线缓冲器74LS126 ( $IC_1$ )等器件。A/D输入和D/A输出量程都在0~10V。图2中“( )”号码表示接APPLE—Ⅱ扩展插槽中的引脚。

图2电路中,要使逻辑关系正确,必须使主机不同A/D和D/A打交道而作其它工作时,出现于 $A_0 \sim A_3$ 的地址信号和 $D_0 \sim D_7$ 的数据信号与我们这个接口无关。这时必须封锁A/D芯片和D/A芯片。在需要从接口输入模拟信号时,仅打开A/D芯片;输出控制信号时,仅打开D/A芯片。由于多路开关 $IC_5$ 两用,既作为模拟信号输入,又作为控制信号输出,模拟通道数和控制通道数根据需要可在0~7通道上任意变化,具体选择由 $A_0 \sim A_2$ 译码决定。为了不使模拟信号输入时因与F005运算放大器输出端相接而受影响,加入了隔离二极管;反之,虽然控制信号作用于A/D输入端,但由于A/D不导通,输入阻抗高,无关紧要。

A/D、D/A芯片被封锁,打开由三态缓冲器、3DK7构成的电路和 $A_3$ 地址信号,  $\overline{DEVICE}$ 、 $\overline{SELECT}$ 接口开关信号执行。其原理如图3所示。这个三态缓冲器若1或4端为低电平时,输出断开。

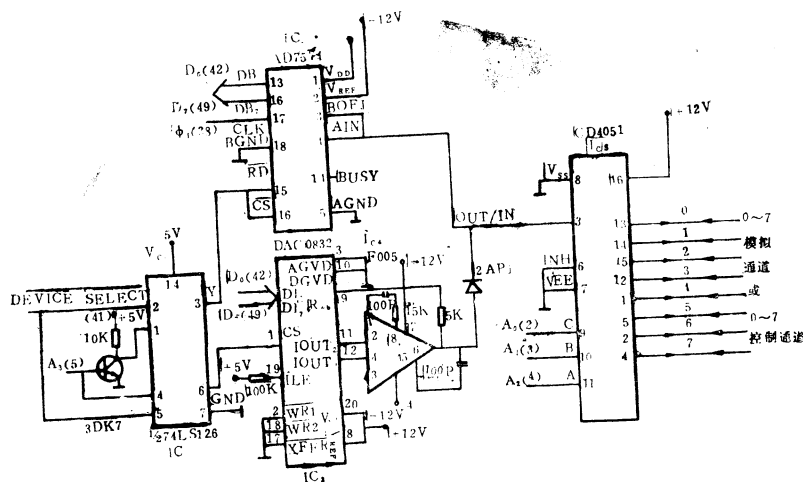


图2 A/D、D/A转换电路原理

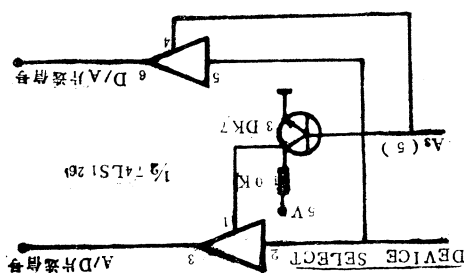


图3 A/D D/A 芯片导通、不导通原理图

如果接口插在APPLE—II微机5号扩展插槽上,需要A/D输入信号时,发出地址选择信号\$C0D0~\$C0D7,其中0~7用于A<sub>0</sub>~A<sub>2</sub>译码,任选一通道。地址信号到来,DEVICE SELECT信号到来。由于A<sub>3</sub>为0,继续封锁D/A芯片,而三态缓冲器1端为1,打开了A/D芯片,通道上模拟信号即送A/D开始转换,需要从D/A发出控制信号时,地址信号为\$C0D8~\$C0DF,其中8~F用于A<sub>0</sub>~A<sub>2</sub>译码,任选一通道,同时A<sub>3</sub>为1,三态缓冲器1端为0,封锁了A/D芯片。4端为1,DEVICE SELECT信号到来,打开了D/A芯片,控制信号即可发送。一般情况下,我们不会使用地址信号\$C0D0~\$C0DF,在5号扩展插槽中不产生

DEVICE SELECT信号, A/D、D/A芯片均被封锁, 不干扰主机总线。

### 三、编程示例

编程中应考虑这种硬件结构的特性。例如,从通道0采集数据,从通道7输出5V控制信号,用6502汇编语言编制的程序如下:

```

LDX    # $00    ; X寄存器送初值
STA     $C0D0    ; 启动通道0, 开始转换
LOOP:   INX       ; 延时、等待转换结束
        CPX      # $20
        BNE      LOOP
        LDA       $C0D0    ; 采集数据
        STA       $1000    ; 送数据区
        LDA       # $8A    ; 取5V信号值
        STA       $C0DF    ; 从通道7送出控制信号
        NOP
        NOP            ; 延时
        NOP
        RTS          ; 返回调用程序

```

#### 四、讨论

本文介绍的接口设计构思,不仅实用于APPLE—II微机,根据需要,这种方法也

# 介绍一种新型的存储器

## ——电擦式可编程只读存储器

济南自动化仪表厂 施文济

### 摘要

本文介绍一种新型的电擦式可编程只读存储器E<sup>2</sup>PROM的特性和运用情况以及其指令功能,以供国内微机爱好者参考。

### 一、前言

最近在国际市场上推出一种新型的电擦式可编程只读存储器E<sup>2</sup>PROM,其价格低廉,适应性广,编程容易,在许多运用场合是一种较理想的器件。特别用来准永久性存储仪表配置数据时是很理想的。

### 二、E<sup>2</sup>PROM的特性与应用情况

本文以松下NMC9306型256比特电擦式可编程只读存储器E<sup>2</sup>PROM为例来介绍其特性和运用情况。

E<sup>2</sup>PROM存储器是将通常的EPROM存储器的永久性存储的特性和RAM存储器

的随机性存储的特性结合起来。在这一点上来说,其功能类似于带备用电池的RAM存储器或零功率RAM存储器(例如48Z02)。E<sup>2</sup>PROM是一种只读存储器,一般来说只能从中读出。然而,它可以用电来擦除其内部的专用程序,并在正常操作期间又可对其重新装入程序。对于传统的EPROM来说,要擦除内部程序时,必需用紫外线光照射。编程时必需施加较高的编程电压(一般为+25V)。而NMC9306E<sup>2</sup>FRM是用5V单电源供电,并在芯片中装有升压转换器,以提供编程电压。它含有16个16位寄存器。每个寄存器都能独自被擦除。与常规的RAM相比,其最重要的区别是写入一个寄存器所需的时间。对于NMC9306芯片来说,每个寄存器的编程周期至少10ms。每个寄存器的写操作数被限止在10000左右,保证数据保

可在其它机型上实现,但必须注意几点:

1.该方法由于节约了器件,A/D芯片转换期间采用的是程序等待方法,一般不易设计出转换结束的中断请求方式。

2.该方法如果控制信号发出,一旦撤出地址信号,D/A芯片即被封锁,因此也应加一定延时保证控制信号能送给被控对象。

3.D/A输出时由于要通过2AP1,必须

考虑二极管正向压降。例如上例程序中送出的控制电压为5.4伏,0.4伏作为二极管正向压降补偿。

实践证明,这种方法是一种经济实用,简单而切实可行的方法,完全可以在一定应用范围内代替按一般方法设计的较贵的A/D、D/A转换接口。

存的最长时间是10年。

如上所述，对于设备配置数据的准永久性存储来说，E<sup>2</sup>PROM是最理想的器件。例如飞利浦测试仪器上装有许多带E<sup>2</sup>PROM的高级频率表。在这些E<sup>2</sup>PROM中存有对应于温度补偿炉内所装的中心石英晶体温度响应曲线的数据。在这些仪器中将想用的每块石英晶体的温度系数分别作为一条特性曲线记录下，然后将它数字化并装入E<sup>2</sup>PROM中。控制仪器测量炉温的微处理器从对照表中装入有关的温度系数，并校正中心时钟频率，以保证最小偏差。

E<sup>2</sup>PROM 芯片和其他存储器芯片之间的最本质的区别可从图1的方框图中看出来。数据经过串行接口送入 E<sup>2</sup>PROM，并从E<sup>2</sup>PROM中读出。这不仅使E<sup>2</sup>PROM有可能以8脚双列直插芯片型式来封装，而且还使它的使用不依赖于数据总线和地址总线的配置。从这点来说，E<sup>2</sup>PROM 类似于一种小型外围器件。

表1 NMC9306 16×16位E<sup>2</sup>PROM指令表

指 令	最高位	操作码	地 址	数 据	注 释
READ	1	10××	A3A2A1A0		读寄存器A3A2A1A0
WRITE	1	01××	A3A2A1A0	D15~D0	写寄存器A3A2A1A0
ERASE	1	11××	A3A2A1A0		擦除寄存器A3A2A1A0
EWEN	1	0011	××××		允许擦除/写入
EWDS	1	0000	××××		禁止擦除/写入
ERAL	1	0010	××××		擦除所有寄存器
WRAL	1	0001	××××	D15~D0	写入所有寄存器

注：× 是任意状态

位寄存器用。

E<sup>2</sup>PROM控制指令的功能概括如下：

Read（读）数据首先装入移位寄存器，然后经串行输出DO移出。通过加到SK输入的信号高低变换来计算移出操作的时间。

空位（逻辑零）放在16位数据输出串之前。仅是读指令使串行数据经DO线输出。

Erase register（擦除存储器）E<sup>2</sup>PROM跟RAM不一样，其寄存器在装入新数据之前应该先清洗。

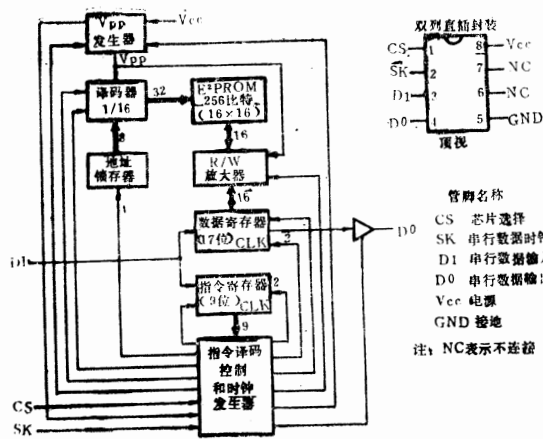


图1 NMC9306型E<sup>2</sup>PROM存储器的方框接线图

串行输入、输出插脚（DI和DO）可以通过分隔串行的格式来控制，这种串行接口还可以用来从主微处理器中接收E<sup>2</sup>PROM控制指令。对NMC9306E<sup>2</sup>PROM来说，有7条指令，它有9位串行数据字组成，每条指令的最高位是“1”，并看作为接口程序的起始位。接下来的4位组成操作码（见表1）。另外余下4位组成寄存器的地址，供1至16个16



# 利用APPLE II 微型机控制DMS-1型 64路数据采集装置的技术

紫金信息公司 电脑分公司 宋晓东

**摘要** 本文介绍一种用微型机系统来控制数据采集装置对热管表面温度进行测试的技术,着重叙述了APPLE II 微机与DMS-1型采集器接口的硬件及软件设计情况。

DMS-1 型仪表是一种国产的64路数据采集装置,通过微型机对它的控制,可实现多路、多种数据(如温度、水压等)的采集和处理,有着广泛的应用。下面就具体介绍一下这种仪表与微型机连接的硬件和软件的有关技术问题和解决方法。

## 一、硬件设计

DMS-1型数据采集装置与微型机方面采用的是可编程并行接口器件8255,而以ROCKWELL公司的6502芯片为CPU的APPLE II微型计算机在与这种仪表连接时,存在着时序不匹配问题,经过分析并反复实践,解决了这一问题,其方法是:在8255的片选信号线和数据线上增加反相器,作适当

\*\*\*\*\*  
■Erase all register(擦除所有寄存器)这条指令类似于上面的指令,所不同的仅是它对整个芯片起作用,而不是对个别寄存器起作用。

■Write(写入)将数据写入预先清除过的寄存器中。

■write all register(写入所有寄存器)这条指令是将同一数据写入所有的寄存器中。

■Erase/write disable(擦除/写不起作用)这条指令可以防止寄存器被无意地擦除或重复写入。

在E<sup>2</sup>PROM上有二条控制来排列时标,高低时钟脉冲在SK(串行数据时钟)线上传输,控制数据和指令的移入和移出,其最大时钟频率为250KHz。CS(芯片选择)线为有效高电位,并能允许或禁止全部数据和I/O操作指令的传输。它还用来抹掉时标,产生一个10~30ms宽度的编程脉冲,装入

擦除指令或写入指令后,当CS变成低电平时,有关的脉冲周期就开始,直到CS换向到逻辑1才维持编程。在指定的时标里,SK的输入被禁止、编程完成后,CS可以保持逻辑高电平,允许装入新的指令。当CS变成逻辑低电平时,E<sup>2</sup>PROM转换到低电源方式。在指令之间,CS的最小低电平时间为1μs。

当然在实际使用时,大容量的存储器芯片是十分引人注目的。然而象E<sup>2</sup>PROM那样的低容量存储器,由于其价格低廉,适应性广、编程容易,因此在许多应用场合是一种较理想的器件。目前在国外的一些仪器上已经使用这种存储器。

## 参考文献

1. Microcontroller-driven power supply—ELECTRONICS May 1988, June 1988.
2. Background to E<sup>2</sup>PROM—ELECTRONICS, December 1988

延时,可以实现正确的读写应答时序关系,从而在硬件电路上保证了用APPLE II对DMS—1型仪表的控制。图1给出了DMS—1型仪表与APPLE II微型机连接的接口电路示意图。

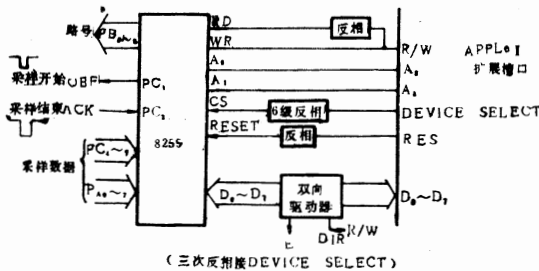


图1 DMS—1与APPLE II接口电路示意图

Intel 8255是一种功能很强的并行输入输出接口芯片,它共有三个端口PA、PB和PC,每个端口8位,可以作为数据通道,也可以作为控制信号线。通过编程,8255可以设定三种工作方式。

图2给出了8255B口输出的时序图。

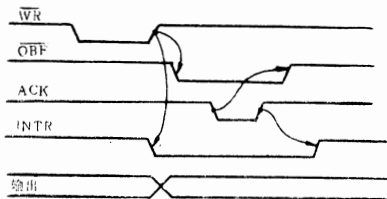


图2 8255 B口输出时序图

由图1可见,在这个接口电路中,8255的A组被设置为基本输入输出方式,作为采样数据的输入通道,B组被设置为通道输入输出方式,其中B口作为选择采集路号的输出通道,PC1作为启动采样的信号(OBF),负脉冲有效,由微型机发出,PC2作为采样结束的回答信号(ACK)仍然是负脉冲有效,由采集装置返回。

## 二、软件设计

8255方式字定义的格式如图3所示。方式1输出结构状态字格式如图4所示。

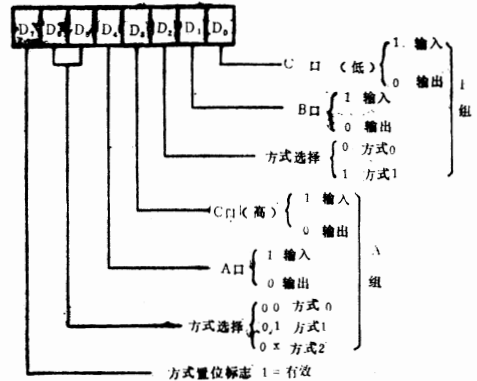


图3

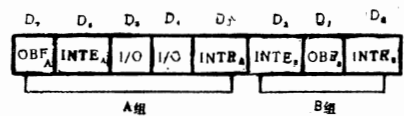


图4

硬件设计规定的地址为:

\$C0B0: A口(采集数据低8位的通道)。

\$C0B1: B口(选择采集数据路号的通道)。

\$C0B2: C口(其中PC4~7为采集数据高4位通道)。

\$C0B3: 写入为控制字,读出为状态字。

根据时序图(图2)和上面的叙述,设计出采集一个数据的程序流程图如图4所示。

我们在利用APPLE II控制DMS—1型数据采集装置对热管表面各点温度的测试和数据处理中,获得了满意的结果。热管表面温度经过热电偶转换接入DMS—1型仪表,APPLE II对DMS—1实施控制,将采集的数据校正和打印。图5、图6是完成这一工作的流程图。

下面给出了利用APPLE II控制DMS—1型数据采集装置对热管表面温度采集并处理的程序清单。

注意:程序规定连接DMS—1的接口板应插在APPLE II的3号扩展槽上。

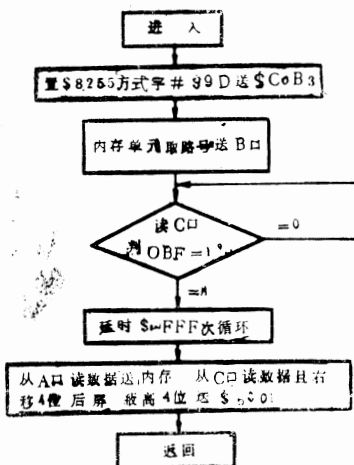


图5 采集子程序流程图

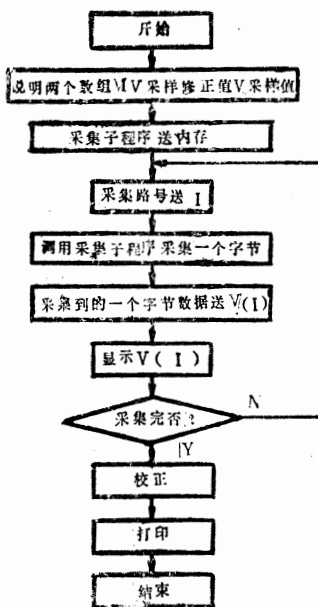


图6 BASIC采样程序流程

# LIST

```

100 DIM MV(63)
110 DIM V(64)
120 LI = 3 * 256
130 FOR I = 0 TO 62
140 READ X; POKE LI+1, X
150 NEXT I
160 AD = 5 * 16 * 256
170 FOR I = 0 TO 63
180 POKE AD, I

```

```

190 CALL LI
200 V(I) = PEEK (AD)+PEEK (AD+1)
    * 256
210 PRINT I; " "; V(I); " ";
220 NEXT I
230 PRINT
240 V=30.26
250 FOR I=2 TO 63
260 K=V/(V(0)-V(1))
270 MV(I)=K*(V(I)-V(1))
280 NEXT I
290 GOTO 370
300 FOR I=2 TO 63
310 PRINT "MV(" ; I ; ")= ";
    MV(I); " ";
320 NEXT I
330 DATA 169, 157, 141, 179, 192,
    173, 0, 80, 141, 177, 192, 173, 178,
    192, 41, 2, 240
340 DATA 249, 32, 40, 3, 173, 176,
    192, 141, 0, 80, 173, 178, 192, 106,
    106, 106, 41, 15, 141, 1, 80,
    96, 72, 138
350 DATA 72, 162, 255, 138, 72, 162,
    255, 169, 255, 202
360 DATA 208, 251, 104, 170, 202,
    208, 242, 104, 170, 104, 96
370 END
300LL
0300— A9 9D LDA #$9D
0302— 8D B3 C0 STA $C0B3
0305— AD 00 50 LDA $5000
0308— 8D B1 C0 STA $C0B1
030B— AD B2 C0 LDA $C0B2
030E— 29 02 AND #$02
0310— F0 F9 BEQ $030B
0312— 00 28 03 JSR $0328
0315— AD B0 C0 LDA $C0B0
0318— 8D 00 50 STA $5000
031B— AD B2 C0 LDA $C0B2
031E— 6A ROR
031F— 6A ROR
0320— 6A ROR
0321— 6A ROR
0322— 29 0F AND #$0F

```

## 方便、快速地修订CCDOS拼音首尾码表

在IBM PC/XT及其兼容机上普遍使用的CCDOS中,拼音与首尾输入法共存,极大的方便了汉字的输入。但是,在具体使用过程中,我们发现了几百个汉字的拼音码或首尾码不很恰当、甚至完全错误;按照取码规则很难找到这些汉字,例如“泽”字的首尾码应该为“HM”竟编成了拼音码“ze”...,还有部分汉字的拼音、首尾码全部错误,这些字只能靠区位法输入,严重地影响了工作。

在工作实践中,有不少同志都在探求对输入码表的修改。有些同志利用BASIC语言编制一小程序查出难找汉字的编码,然后利用DEBUG或PCTOOLS对CCCC.EXE文件进行修订…。虽然这些方法都能起到对输入码表进行修改的作用,但是,效率甚低且由于人为因素的影响而很易出现差错,麻烦程度更不用说。笔者通过探索,总结其他同行的经验,寻求了一种比较方便而又快速、省力的方法,减少了人的因素的影响,提高了修订的准确性和高效性,而且可以对

内存、磁盘文件同时进行修订,使得修订后的输入码表既可以为当时输入汉字服务,又有利于以后重新启动系统时使用,使修订工作真正自动化,达到一劳永逸的功效,经过多次试用,效果令人满意。

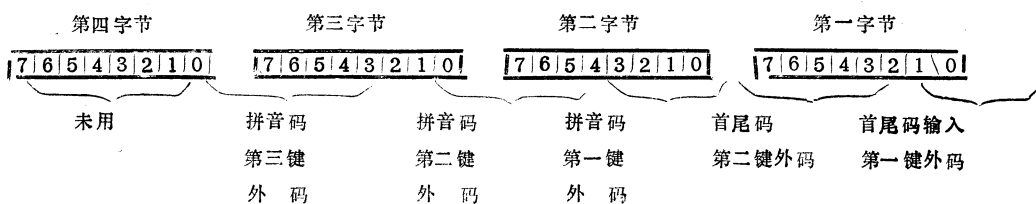
首先,我们通过对CCCC.EXE文件的分析,找出了拼音首尾输入码表的起始地址:对于CCDOS V2.0版而言,其起始地址为2C71H(十进制数11377);对于CCDOS V2.1版而言,其起始地址为2BD6H(十进制数11222)。考虑到目前还有很大一部分同志仍在用2.0V/2.1V CCDOS,因此,本程序兼顾了这两个版本(区别在于NO初值的不同)。

由于输入码表采用了拼音首尾混合的方式存贮,因此,对输入码必须进行一定的转换。

每个汉字的拼音首尾码的表项各占据4个字节,各字节与拼音/首尾码的键位关系如图1所示:

\*\*\*\*\*

0324—	8D 01 50	STA	\$5001	0336—	68	PLA -
0327—	60	RTS		0337—	AA	TAX
0328—	48	PHA		0338—	CA	DEX,
0329—	8A	TXA		0339—	DO - F2	BNE \$032D
032A—	48	PHA		033B—	68	PLA
032B—	A2 FF	LDX	#\$FF	033C—	AA	TAX
032D—	8A	TXA		033D—	68	PLA
032E—	48	PHA		033E—	60	RTS
032F—	A2 FF	LDX	#\$FF	033F—	00	BRK
0331—	A9 FF	LDA	井\$FF	0340—	00	BRK
0333—	CA	DEX		0341—	00	BRK
0334—	DO FB	BNE	\$0331	0342—	00	BRK



根据上图可以很容易找到“外码 $\Rightarrow$ 字节 2 30及2200~2230行)。  
内容”间的转换关系(参见程序清单2120~ 程序清单如下:

```

2000 DIM T(4), R$(5), K(5)
2001 CLS
2002 LOCATE 9, 10, PRINT "[0]-CCBIOS V2.0 [1]-CCBIOS V2.1";
    ., WW$=INPUT$(1); IF VAL(WW$)<0 OR VAL(WW$)>1 THEN 2002 ELSE PRINT
    WW$
2003 LOCATE 9, 10, PRINT "[1]-仅改内存编码 [2]-仅修改CCCC.EXE [3]-两者同时修改 ";
    , WX$=INPUT$(1); IF VAL(WX$)<1 OR VAL(WX$)>3 THEN 2003 ELSE PRINT WX$
2004 IF VAL(WW$)=0 THEN WW=0 ELSE WW=1
2005 IF VAL(WX$)=1 THEN 2010
2007 LOCATE 9, 10, PRINT "CCCC.EXE 所在盘号"; SPACE$(45); , X$=INPUT$(1); IF
    (X$<"A" OR X$>"C") AND (X$<"a" OR X$>"c") THEN 2001 ELSE PRINT
    X$
2009 OPEN "R", #1, X$+"", "CCCC.exe", 1, FIELD#1, 1 AS A$; IF LOF(1)=0 THEN
    BEEP; CLOSE; KILL X$+"", "CCCC.EXE"; SYSTEM
2010 IF VAL(WX$)=2 THEN 2035
2020 DEF SEG=0; D=PEEK(127)*16+PEEK(126)
2030 DEF SEG=D
2035 CLS; LOCATE 4, 15, PRINT STRING$(44, "."); LOCATE 8, 15, PRINT STRING$(
    (44, " .")); LOCATE 9, 1, PRINT SPACE$(23); "请使用小写英文字母!"; SPACE$(30)
2036 LOCATE 1, 12, PRINT "请选择[1]-单个 [2]-连续 [E]-结束";
2037 C=ASC(INPUT$(1)); IF C=69 OR C=101 THEN CLOSE, SYSTEM
2038 IF C=50 THEN LL=1; LOCATE 2, 18, PRINT " 注意: 退出连续请按ESC键" ELSE
    PRINT CHR$(C)
2050 IF C<>49 AND C<>50 THEN GOTO 2036
2060 LOCATE 5, 18, INPUT "请输入您所要修改的汉字: ", HZ$; IF HZ$="" THEN 2036
2070 QH=ASC(LEFT$(HZ$, 1))-160; WH=ASC(RIGHT$(HZ$, 1))-160; N0=11377; IF
    WW=1 THEN N0=11222
2075 QH$=RIGHT$(STR$(QH), 2)
2077 IF WH<10 THEN WH$="0"+RIGHT$(STR$(WH), 1) ELSE WH$=RIGHT$(STR$(
    WH), 2)
2080 LOCATE 5, 48, PRINT "区位码: "; QH$; WH$; N=N0+376*(QH-16)+4*(WH-1)
2090 IF VAL(WX$)=2 THEN FOR I=1 TO 4: GET #1, N+I+512; T(I)=ASC(A$); NEXT;
    GOTO 2120
2110 FOR I=1 TO 4; T(I)=PEEK(N+I-1); NEXT
2120 R$(1)=CHR$(96+(T(1) AND &H1F)); R$(2)=CHR$(96+(T(1) AND &HE0)\2+5+(T(2)
    AND &H3)*2+3)
2130 R$(3)=CHR$(96+(T(2) AND &H7C)\2+2); R$(4)=CHR$(96+(T(2) AND &H80)\2+7
    +(T(3) AND &HF)*2); R$(5)=CHR$(96+(T(3) AND &HF0)\2+4+(T(4) AND &H1)*2
    +4)

```

# 打印机控制命令的中西兼容

**摘要** 在CCDOS下,不认打印机的控制命令,这样就不能充分发挥打印机的功能。本文在不修改打印驱动程序的情况下,就解决了中西兼容的问题。

## 一、设计思想

打印机控制命令不能中西兼容的原因是:进驻CCDOS时,修改了有关中断矢量,从ROM—BIOS指向了CC—BIOS。而汉字

\*\*\*\*\*

```
2140 LOCATE 6,18: PRINT HZ$; " 的原首尾码为: "; R$(1); R$(2); " 请改成: ";  
    , INPUT " ", R$: IF LEN(R$)<>0 THEN R$(1)=MID$(R$,1,1): R$(2)=MID$(  
    (R$,2,1)  
2150 LOCATE 7,18: PRINT HZ$; " 的原拼音码为: "; R$(3); R$(4); R$(5); " 请改成  
    : "; , INPUT " ", R$: IF LEN(R$)<>0 THEN R$(3)=MID$(R$,1,1): R$(4)=  
    MID$(R$,2,1): R$(5)=MID$(R$,3,1)  
2170 FOR I=1 TO 5: IF LEN(R$(I))=0 THEN R$(I)=" { "  
2180 NEXT  
2190 FOR I=1 TO 5: K(I)=ASC(R$(I))-96: NEXT  
2200 T(1)=(K(1) AND &H1F)+(K(2) AND &H7)*2^5  
2210 T(2)=(K(2) AND &H18)\2^3+K(3)*2^2+(K(4) AND &H1)*2^7  
2220 T(3)=(K(4) AND &H1E)\2+(K(5) AND &HF)*2^4  
2230 T(4)=(K(5) AND &H10)\2^4  
2240 IF VAL(WX$)=2 THEN 2260  
2250 FOR I=1 TO 4: POKE N+I-1, T(I): NEXT  
2260 IF VAL(WX$)=1 THEN 2280  
2270 FOR I=1 TO 4: LSET A$=CHR$(T(I)): PUT #1, N+I+512: NEXT  
2280 C=ASC(INPUT$(1)): IF C=27 THEN LL=0: PRINT: GOTO 2035  
2290 IF LL=0 THEN CLS: GOTO 2035 ELSE WH=WH+1: IF WH=95 THEN WH=1: QH  
    =QH+1: IF QH=88 THEN 2035  
2300 LOCATE 5,18: PRINT " 您所修改的汉字: "CHR$(160+QH)CHR$(160+WH)SPACE  
    $(8): , HZ$=CHR$(160+QH)+CHR$(160+WH): GOTO 2075
```

对程序使用中的几点说明:

①本程序可以在单个修改或连续修改两种状态下进行工作。若工作在连续修改状态,则除了第一次必须送一个汉字以确定修订起点外,对后续部分汉字仅需提供新的首尾码及拼音码;如果对原码不进行修改,则可直接按回车键以保留之)。

②本程序可以三种方式工作:仅仅对内存中输入码表进行修订;或仅仅对磁盘文件CCCC.EXE中输入码表进行修订;或二者同时进行。

③如果欲对CCCC.EXE进行修订,可事先指定盘号,使程序更加灵活、方便使用。

湖北省水利厅水利学校 吴家富



打印驱动程序,把大部分控制命令,当做字符处理了。这样使得打印机的功能不能充分发挥,打印时,也显得不够灵活。改写汉字打印驱动程序,非专业人员办不了。而且,改用不同型号的打印机,还得改写相对应的汉字打印驱动程序,相当麻烦。其实,只要反其道而行之,在发送打印机控制命令时,使有关中断矢量复原,发送完毕后,返回指向CC-BIOS。这样就解决了中西兼容的问题,也不影响汉字的输出。

## 二、实现方法

通过分析,发送打印机控制命令只与INT17H中断有关。它的入口地址放在0:005C-0:005F这四个字节中。不管用什么方法,只要在发送打印机控制命令之前,在这四个字节里,重新写入PC-DOS INT17H中断矢量。发送完毕后,再返回CCBIOS,这样就达到了中西兼容的目的。

方法是多种多样的。可以用汇编语言,编写一段子程序;也可利用有些高级语言,直接访问内存的功能,读写这四个字节。

例如,BASIC中就有POKE,PEEK和DEF SEG,可以直接访问内存。在dBASE-Ⅲ中,现也发现有三条类似的命令和函数。下面就以dBASE-Ⅲ为例。(程序清单附后)本程序已在PC-DOS3.3 CC-DOS2.1和dBASE-Ⅲ1.0支持下通过。

SET SEGMENT TO 0是设置POKE和PEEK可访问的段地址。本语句是定在0段。注意一点的是,POKE命令的参数,不可为表达式。这样,就得先用PEEK函数或其它方法,读出中西INT17H中断矢量,再写入POKE命令。POKE命令中,前一个

数字是地址偏移量(十进制数),后一个数字是写进的十进制数据。前四个POKE命令是恢复PC-DOS INT17H中断矢量。后四个POKE命令是返回CCDOS INT17H中断入口。在调用该程序时,只要在主程序中把打印机控制命令,以字符的形式,赋给变量QQ就行了。

例如,在主程序中,加入这样两条指令:

```
QQ="CHR(28)+'*t' "
```

```
DO INT17
```

就激活了M-1724打印机的下划线功能。在打印数据的同时,打出了下划线。加快了打印速度,又节约了纸张,也使得表格更加优美。

```
type INT17.PRG
SET SEGMENT TO 0
POKE 92, 210
POKE 93, 239
POKE 94, 0
POKE 95, 240
? &QQ
POKE 92, 131
POKE 93, 0
POKE 94, 209
POKE 95, 83
```

### 主要参考文献

- 李玉林编 《PC-DOS2.0系统汉化和CCDOS2.0/2.1分析与使用》中科院北京科海培训中心 1986  
江明富 《dBASE Ⅲ的几个直接内存访问命令和函数及其应用》“新浪潮” 1989.1

山东潍坊造锁总厂 陈万年

## PC机C88语言I/O接口程序的汉化

在PC机上用C88语言做一些具有屏编功能的软件时,发现若在汉字操作系统下运行这些软件,有些编辑功能键和函数功能键不好用,或输入汉字时执行编辑功能键( /函数功能键)功能。如PgUp, PgDn, Ins, Del等。其原因是C88的I/O接口程序“PC-IO.A”为编辑功能键和函数功能键所设置的ASCII码有些是大于160的,所以,若在汉字操作系统下使用这些软件,系统就把编辑功

能键( /函数功能键)与汉字的区位码混淆了。

只要把“PCIO.A”文件中为函数功能键和编辑功能键所设置的ASCII码做些改动,使其适应汉字操作系统,就实现了C88语言I/O接口程序的汉化。下面给出一种改动,“PCIO.A”的方案,用修改后的“PCIO.A”作接口程序时,编辑功能键和函数功能键就可运用自如了。

原PCIO.A 中应修改的程序段		对应修改后的 PCIO.A 的程序段	对应键
• • •		• • •	
; /* control key translations: */		; /* control key translations: */	
up_char equ 30		up_char equ 5	↑
down_char equ 31		down_char equ 24	↓
left_char equ 29		left_char equ 19	←
right_char equ 28		right_char equ 4	→
bol_char equ 200		bol_char equ 1	Home
eol_char equ 201		eol_char equ 6	End
Pageup_char equ 202		pageup_char equ 18	PgUp
pagedown_char equ 203		pagedown_char equ 3	PgDn
bof_char equ 204		bof_char equ 30	^Home
eof_char equ 205		eof_char equ 31	^End
Ins_char equ 206		Ins_char equ 22	Ins
Del_char equ 207		Del_char equ 7	Del
NextWord_char equ 208		NextWord_char equ 2	^→
PrevWord_char equ 209		PrevWord_char equ 26	^←
M1 equ 210		M1 equ 8	F1
M2 equ 211		M2 equ 9	F2
M3 equ 212		M3 equ 10	F3
M4 equ 213		M4 equ 11	F4
M5 equ 214		M5 equ 12	F5
M6 equ 215		M6 equ 15	F6
M7 equ 216		M7 equ 16	F7
M8 equ 217		M8 equ 21	F8
M9 equ 218		M9 equ 28	F9
M10 equ 219		M10 equ 29	F10
• • •		• • •	

# 微小型计算机开发与应用

## 一九八九年总目录

### 一期

微机在潮汐模型试验中的应用  
辛普森法在离散控制系统中的应用  
一个实用的计算机选票统计系统  
胎儿微机中心监护系统  
编译方法在经济领域中的应用  
双级微机数据采集存贮系统  
关于第四代语言的讨论  
一类瓶颈问题的最优化解  
PC-DOS文件屏幕显示的改进  
Intel 82385单片高速缓存控制器  
以单片微型机为基础的汉字小键盘的研制  
汉字终端的新技术及其实现  
微型计算机工业控制可靠性实践

### 二期

减压器微机测试台  
TP801控制的测井信息实时显示器  
在汉字终端上开发“自动译报”功能  
单片微机火灾报警控制器  
0520CH作VAX系统图形终端的实现  
MCS-48多回路可编程调节仪软件抗冲程措施  
使用dBASE III的几个技巧  
在微型机上计算大规模线性规划模型  
IBM个人计算机虚拟盘  
高分辨率图形的存贮和恢复  
GATEWAY实现原理  
一种基于IBM PC机的语音输入输出接口设计  
巧妙的接口技术  
NM9400打印机断针的维修  
LA210打印机绘图故障分析和排除

### 三期

单板机在汽车动态测试系统中的应用  
万能材料试验机中的微机测量系统  
APPLE II微机和钙铁分析仪及水泥生料配料系统的  
联机应用  
一个用单板机研制的电源“浪涌”监测仪  
浅论计算机图形显示技术  
层次分析法软件AHP  
微型机开发系统设计新技术  
客观性通用出题系统——OGCTS

远动系统的主控机与通讯机的互连设计  
微机在在线控制系统中的应用  
小型DDNC系统的实现  
IBM-PC/X T738K内存的扩充  
数字语音识别  
个人计算机与普通电视机的接口  
单板机内存的扩充和监控的改写

### 四期

一个变电站的微机监测系统  
政府机关公文管理系统的设计与实施  
以三次样条函数逼近非线性、用微机实现的方法  
微电脑计价秤的接口电路设计  
ROBOT-1机械手控制器剖析  
不添加任何硬件扩充内存的一种方法  
信息系统开发的几种有效方法  
微机上矩阵求秩的一种快速算法  
对于稀疏型数据表的压缩方法  
在应用软件中执行DOS命令的方法  
微型机信号处理系统中软接口的研究  
一种实现UNIX系统与VAX/VMS系统间异步  
通讯的软件  
新一代关系数据库—FoxBASE+  
STE总线介绍

### 五期

长城0520-CH的屏幕作图  
单片机8031在STD智能打印模块中的应用  
在IBMPC机上实现心电图动态显示  
智能型心率测试仪  
人群膳食结构及营养状况分析评价系统的设计与实现  
APPLE IIe微机系统在数控加工中的应用  
UNIX时钟的夏令时机制及改为我国夏令时的方法  
市级经济信息管理系统分析  
dBASE III屏幕窗口设计  
一种用于绘制集成电路图的交谈式计算机绘图系统  
局部网络传输介质比较  
磁带机接口设计  
六所CCDOS打印模块到五笔字型打印模块的转换

### 六期(见本期目录)