

# 计算机时代

COMPUTER ERA



1991



# 首届浙江省CAD/CAM学术研讨会征文通知

浙江省计算机学会

浙江省计算机学会CAD/CAM专业委员会将于1991年9月下旬在杭州市主办首届浙江省CAD/CAM学术研讨会，会期一天。

## 征 文 内 容

会议以CAD/CAM应用为主，包括CAD/CAM的基本方法在电子、建筑、轻工（纺织、服装、美工装潢）等领域的应用及各种通用图形支撑软件。

请于1991年4月30日前将论文全文（不超过4000字）寄浙江大学人工智能研究所，截止日期以当日邮戳为准。要求论文没有在其它期刊或全国性学术会议上发表过。

会议程序委员会将于6月上旬发出论文录用通知。论文的作者可列席随即召开的第四届全国CAD/CAM学术会议。凡是全文录取的论文都将录入正规印刷的第四届全国CAD/CAM会议论文集。会议将评选少量优秀论文推荐到全国性计算机专业期刊上发表。

## CAD/CAM学术研讨会联系处

杭州浙江大学人工智能研究所 段健平 曹学军

邮政编码 310027 电 话 572244转2578

注 稿件恕不退回，请自留稿底

计 算 机 时 代

（季 刊）

1991年 第1期 （总第29期）

编辑出版 《计算机时代》编辑部  
通讯地址 杭州环城西路新5号  
邮政编码 310006  
主办单位 浙江省计算技术研究所  
浙江省计算机学会  
印 刷 余杭县文教印刷厂

总 发 行 杭州市邮局报刊发行科  
订 购 处 全国各地邮电局  
国内统一刊号 CN 33—1094  
邮 发 代 号 32—81  
出 刊 日 期 1991年2月20日  
每 期 定 价 1.00元



# 计算机时代目次

1991年第1期

(总第29期)

## 应用与实践

地区级电业局MIS的设计与开发..... (1)

于功弟 张秋玲

INFORMIX多用户关系数据库应用实例..... (5)

王敬农 郭磊 陈晓亮 蔡文起 孙波

在VAX/VMS应用程序中控制打印机的一种方法..... (9)

王勇

如何使五笔字型汉字适用多种打印机..... (11)

阎立伟

## 软件技术

计算机图形学中的背景移动算法浅析..... (13)

柏林森

交互图形屏幕编辑系统..... (16)

罗国明

自动生成编译程序的测试实例..... (21)

冯斌

实时中断通信软件包的设计与实现..... (27)

张卫兵

解决内存不足的有效途径..... (31)

胡维华

## 专题报告

智能化信息系统中的离散结构..... (33)

陈有刚

应用软件防错措施的设计..... (36)

马勇

## 维修经验

1BM BIO.COM、IBMDOS.COM丢失后硬盘系统的修复..... (38)

郭和生

24针打印机更换打印针的简单方法..... (39)

瞿明德

## 知识园地

XENIX操作系统存贮管理特点..... (40)

魏勇 潘磊

MCS—51与MCS—96之间数据传输..... (44)

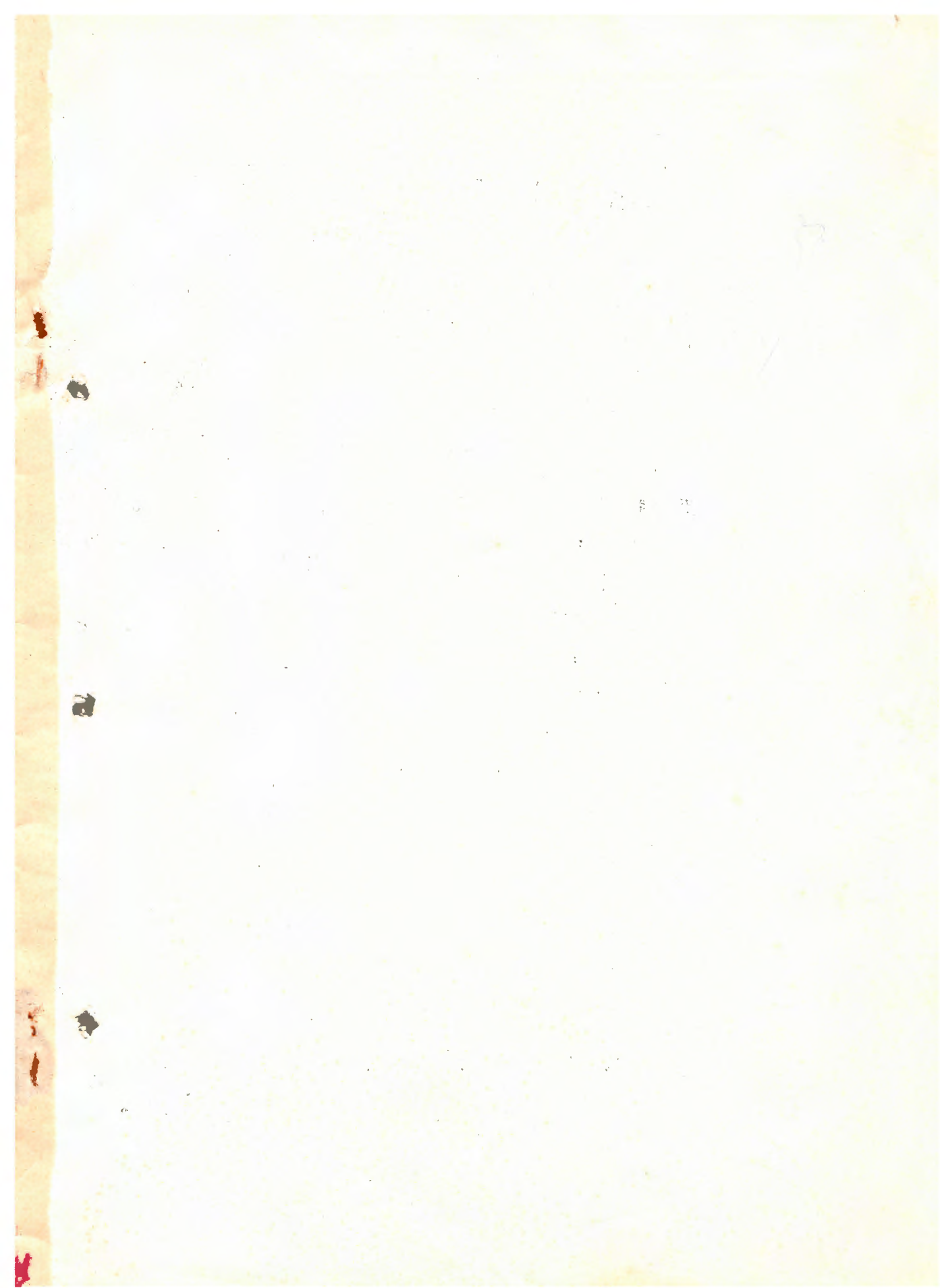
李令戈

1990年分类总目次..... (46)

编辑部

首届浙江省CAD/CAM学术研讨会征文通知..... (封底)

浙江省计算机学会





# 地区级电业局MIS的设计与开发

西南交通大学 于功弟 张秋玲

**摘要** 本文对地区级电业局管理信息系统(MIS)的结构、功能作了探讨,并叙述了利用信息系统工程方法的设计过程,采用综合与分析相结合的设计方法, CODASYL 建库原则,演绎数据库、程序设计风格及结构设计方法和系统的黑盒、白盒测试技术的运用。最后介绍了系统的特点。

## 一、系统介绍

### 1. 系统的构成

本系统从实用的角度出发,根据用户的要求,通过反复调研修改,运用信息系统工

程和软件工程相结合的方法,利用 MU/FO 多用户数据库系统软件环境,建立了整个 MIS 系统结构。

整个系统分为:用电管理系统、财务管理系统、局长查询辅助决策系统,向下再分为各子目标系统,MIS 总体结构见图 1。

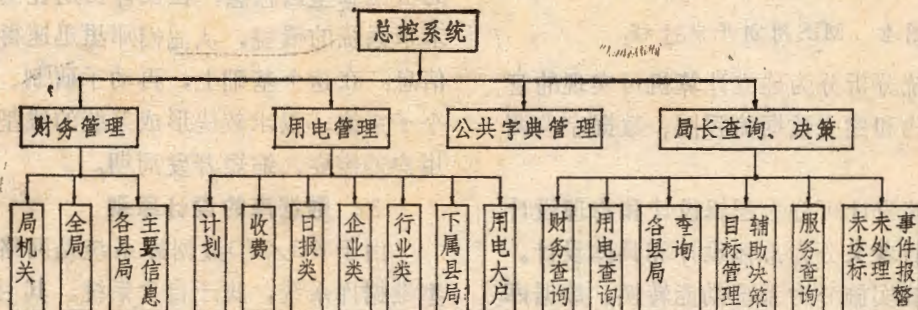


图 1 地区级电业局MIS总体结构

### 2. 系统功能

①完成地区电业局全局主要经济信息的分析、调度和管理。可对统计与经济汇总的大量数据资料进行后期分析处理,并将处理后的数据按一定的表格形式存放。系统在设计上着重加强了对汉字表格的处理功能,各种数据处理方法,全部配有相应表格输出形式。汉字以编码方式储存,由系统建立的统一汉字字典库进行输出和输入转换。

②完成下属 14 个县电业局上报的近 60 多种报表的统计、汇总,存储、计算和分析,并形成各类地区级统计汇总报表上报省局。

③制订每年的供电计划,并对实施进行监控。

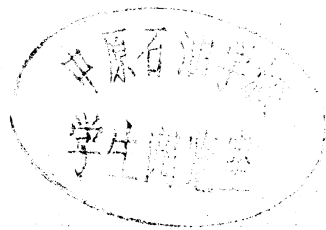
④对下属县局的用电抄表卡进行收费管理,自动完成打印发票,汇总统计各种数据,并将结果数据直接汇总到财务报表中。

⑤为局长及时提供各种信息,以便进行目标管理和辅助决策,并设有未达标及未处理事件的报警系统。

## 二、系统设计

### 1. 系统设计过程

采用系统和软件工程相结合的 开 发 方



# 地区级电业局MIS的设计与开发

西南交通大学 于功弟 张秋玲

**摘要** 本文对地区级电业局管理信息系统(MIS)的结构、功能作了探讨,并叙述了利用信息系统工程方法的设计过程,采用综合与分析相结合的设计方法,CODASYL 建库原则、演绎数据库、程序设计风格及结构设计方法和系统的黑盒、白盒测试技术的运用。最后介绍了系统的特点。

## 一、系统介绍

### 1. 系统的构成

本系统从实用的角度出发,根据用户的要求,通过反复调研修改,运用信息系统工

程和软件工程相结合的方法,利用 MU/FO 多用户数据库系统软件环境,建立了整个 MIS 系统结构。

整个系统分为:用电管理系统、财务管理系统、局长查询辅助决策系统,向下再分为各子目标系统,MIS 总体结构见图 1。

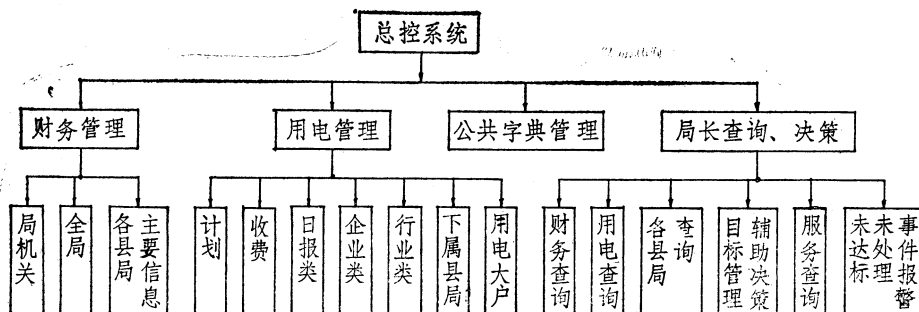


图 1 地区级电业局MIS总体结构

### 2. 系统功能

①完成地区电业局全局主要经济信息的分析、调度和管理。可对统计与经济汇总的大量数据资料进行后期分析处理,并将处理后的数据按一定的表格形式存放。系统在设计上着重加强了对汉字表格的处理功能,各种数据处理方法,全部配有相应表格输出形式。汉字以编码方式储存,由系统建立的统一汉字字典库进行输出和输入转换。

②完成下属 14 个县电业局上报的近 60 多种报表的统计、汇总,存储、计算和分析,并形成各类地区级统计汇总报表上报省局。

③制订每年的供电计划,并对实施进行监控。

④对下属县局的用户抄表卡进行收费管理,自动完成打印发票,汇总统计各种数据,并将结果数据直接汇总到财务报表中。

⑤为局长及时提供各种信息,以便进行目标管理和辅助决策,并设有未达标及未处理事件的报警系统。

## 二、系统设计

### 1. 系统设计过程

采用系统和软件工程相结合的开发方

法,分为系统分析、系统设计、系统实施和系统评价4个阶段,进行周期开发,见图2。

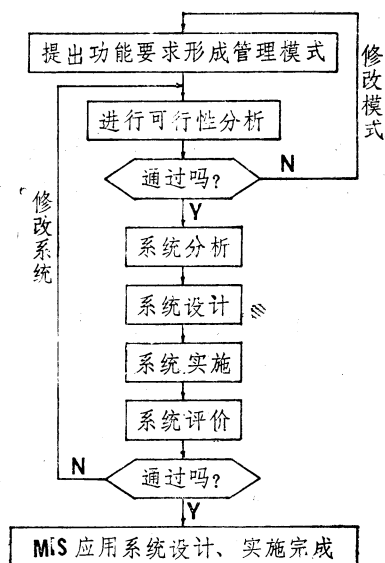


图2 MIS周期开发过程

其中,系统分析分为建立计算机可实现的管理模式结构和建立数据流程图、数据字典两个阶段。

系统设计可分为逻辑设计和物理设计两部分,后者又分为总体设计和具体设计。

系统实施分为系统功能转换、数据库设计实施、程序设计实施和代码设计四部分。

系统评价分为培训管理人员、试运行和鉴定技术文件两部分工作。

从调研到正式投入使用完成上述整套MIS系统,用了一年半时间。

## 2. 系统设计方法

整个系统设计采用综合与分析相结合的方法。综合法是从每个子系统的实际需要或从已有的经验知识出发,自下而上地进行设计,综合成一个子系统。分析法是从总体目标出发,自上而下进行设计,将总目标分解成子目标,变换成所需要的功能,再将第一功能分解为一系列子功能,然后再编制程序

任务书和程序。分析法注重系统的控制流或控制结构比较严谨,容易达到整体最优,可是工作量大,开发周期长。而综合法较能照顾整体要求,又容易开发。在实际应用时,我们将两种方法有机地结合起来。初期设计先采用分析法从总体目标出发,建立整个系统模式,将它分为三个系统,若干个子系统,自上而下进行初设计。然后再将子系统分解成子目标,变换成系统功能,并进行各种约定、规范。最后再从每个子系统的实现目标需要出发,利用MU/FO软件开发工具,自下而上地进行具体设计,综合实现一个总系统。接着再进行整体调试、运行和分析优化,使系统既具有较严谨的系统结构、整体较优又可减少开发工作量。此外,为了加快系统的开发,我们还采用了国际上较新的快速雏型法思想,在设计初期让用户了解未来系统的概貌,从他们那里迅速得到反馈信息。在这个基础上,再动手研制、修改各个子系统,以求较快形成系统的雏型,满足用户的需要,缩短开发周期。

## 3. 数据库的设计原则

由于MU/FO数据库系统是网络、关系型数据库系统,属于自含系统。其主要特点是用网络数据模型描述数据库结构。故可以依照CODASYL (Conference On Data System Language) 组织于1971年4月通过DBT报告中确定的CODASYL表示法原则建库<sup>[1]</sup>。

CODASYL表示网络的办法是把一个网络分解成若干个二级树,并按下列原则建立级间的关系:

①事件间存在一对一关系,两者合并建成一个主子库。

②事件间存在一对多关系,建立事件主子二级库。

③事件间存在多对多关系,分别建几个主子库,直到满足要求。



④采用适当冗余法,以提高系统的响应速度。

⑤单个记录类型的自回路。

采用此原则设立的数据库有下列特点:

- A. 一记录是子又是主。
- B. 一记录可属于多个主记录。
- C. 允许交叉索引记录。

我们在设计数据库的过程中,充分利用上述原则,采用网络与关系相结合的建库设计方法。注意在库的数据设置、功能设置、更新同步和查询分解上采用集中式数据库的优点进行设计。将该建几个大库的数据按应用范围、使用频度分解成不相交或无重叠的子集,对应建立许多小数据库,它们是相互关联的又是独立的。这样一来既提高了系统的响应速度,又解决了数据更新同步问题,还提高了库中数据的安全性。(整个系统共建71个网状主子二、三级库)。

为了减少信息存储量,我们利用演绎数据库的思想设置部分数据库的功能。

演绎数据库 = 实数据 + 规则

演绎数据库能在不占用更多的空间资源下提供远远多于实数据的数据。它还具有易维护、易扩充、冗余度更小和数据录入量小的特点。如我们在设计财务系统时,将一、二级科目下的实金额数据存入一个库,然后再建一个关系规则库,将一、二级科目间的关系和运算规则存入此库。在对数据进行处理时,从两库中分别取出实数据和规则,进行统计、计算、汇总形成各种报表结果数据。以求为用户提供超数据存储量的数据,弥补微机外存容量小的不足。

#### 4. 程序设计方法

##### ①程序设计风格

我们在设计各个表达式文件程序时,按软件工程中的 YOURDON 程序风格<sup>[2]</sup>指导原则约定:

A. 使用结构化编码技术。

B. 不滥用语言的特色。

C. 不写修改自编的程序。

D. 除非绝对必要,避免多重任务。

E. 避免不必要的复杂算法和逻辑表达式。

F. 增加括号以避免含混。

G. 代码应简单而直接。

并约定程序名标准化、含义化,(例 T 开头表示财务程序,第二个字母为 P 表示控制程序, F 和 R 分别表示表格和报表处理程序)。以利调用和检查。

##### ②程序设计方法

我们约定系统所有程序按下述结构化方法设计:

A. 自顶向下逐步求精的来编写程序。

B. 尽量用顺序、选择和循环 3 种基本结构来构成程序,使程序是线状或树型。

C. 限制运用 GOTO 语句,以减少错误,使程序便于调试和查错。

D. 一般限制每个程序不多于 100 条语句。

在程序设计过程中,我们还注意使用软件工程新思想再用性方法。即模仿硬件的组装方式,设计新程序时尽可能使用原有程序作为基本构件,再增添必须语句组成新的程序。同时在设计新程序时,又尽可能使新设计的程序可用于其它程序。即多设计公共模块程序,使其易扩展、多用化。将整个程序软件系统象积木一样堆积起来,组成整个应用系统的软件结构。例如,在设计各系统、子系统的主控程序时,我们先设计一个具有公共功能的模块化程序,其它控制程序都由它再用扩展而成。此外,在设计中我们还注意运用防错性程序设计<sup>[3]</sup>的思想。在设计处理输入数据的重要程序时,采用被动的防错性技术,设立数据检查点,以检查防止此程序的处理错误。在设计数据库数据处理程序时,我们也注意运用上述手段防错,整个



应用系统共编制了大约两万多条语句。

### 5. 系统的测试

在整个系统的调试过程中,我们采用了部分软件测试技巧<sup>[4]</sup>。首先对系统进行系统审查会、人工运行及复查,纠正了部分错误,并修改了系统与用户要求不一致的部分。进一步还利用了黑盒原理测试系统。即在完全不考虑系统内部结构特性和内部特性的条件下,通过输入选择的数据,看是否能得到预期的结果。把输入条件按等价类划分,并进行边值分析,以求覆盖尽可能多的输入情况。例如,在测试财务系统时,将输入凭证按5种(银收、银付、现收、现付、转帐)分成等价类,并考虑一种无效等价类作为输入条件对系统进行测试。另外,还利用白盒测试(也称逻辑驱动测试)从检查系统的内部逻辑结构着手,得到测试数据,对系统边分析边测试,以求达到预期结果。根据最后粗略统计,大约56%的错误在系统审查、人工运行时发现,40%左右的错误在黑盒白盒测试中发现,4%由猜错发现。通过上述各种测试,使系统较快地投入了试运行和正式运行。

## 三、系统特点

### 1. 系统具有积木式、模块化

主子系统是积木式的,子系统内子目标功能实现是模块化的。整个系统便于维护和扩展。

### 2. 程序设计采用结构化设计

各个模块程序具有线状或树型顺序结构,便于理解和维护。由于采用公共功能程序模块规范化调用,简化了程序,最大限度地减少了程序占用的存储空间。

### 3. 采取较为有效的数据设置

划分数据集(建立不相关数据小库,设计具有演绎思想的数据库等),压缩了数据冗余,为用户提供了超数据库存储数据的大

量数据。使整个系统数据库具有较高的存储空间利用率。

### 4. 具有较强的运行安全环境

所有访问库程序都设有锁库语句,并设计了软件锁,以解决并发更新问题。同时还确定了数据库读写级别,设定专用帐号才能更新重要数据库的数据,以提高数据的安全性。

### 5. 具有较好的用户友好性

汉字输入具有编码和汉字两种输入方式。为减少信息录入量,部分输入表格具有重复数据自动传递功能。另外,用户界面全部采用国内、外流行的层次菜单结构,并设有汉字提示,直观易操作。考虑用户查询方便,还吸收了中、大型机中的方法,将整个系统的用户操作手册分层次存于系统中,供用户随用随查。

## 四、结束语

这套MIS系统是在一台微机带四个终端(20M外存,640K内存)的硬件环境下实现较大规模管理系统的一次实践。由于微机运行速度较低,内、外存空间较小,限制了应用系统的功能、规模和扩展。尽管如此,笔者认为仍可作为一个可借鉴的例子。该系统已在山西临汾地区电业局正式运行,收到了较好的经济效益和社会效益。

### 参 考 文 献

- [1] 美T威廉·奥著,诸葛成有译,网状数据库管理系统,浙江教育出版社,1987,1~7。
- [2] 龚世生编著,软件工程引论,成都电讯工程学院出版社,1987。
- [3] M·L 舒曼著,朱兆堂等译,软件工程—设计、可靠性和管理,上海翻译出版公司,1987,87~89。
- [4] 美G·J·Myers著,周之英等译,计算机软件测试技巧,清华大学出版社,1985。

# INFORMIX多用户关系数据库应用实例

浙江省计算技术研究所 王敬农 郭磊 陈晓亮 蔡文起 孙波

## 一、前言

INFORMIX多用户关系数据库由于其数据模式清晰、数据独立性强、查询功能齐全、使用方便、易于扩充等优点,并为用户提供了屏幕格式定义(PERFORM)、报表书写语言(ACE)及C语言扩充接口等工具,因此应用范围比较广泛<sup>[1]</sup>,特别适用于企事业的中、小型数据库管理。但使用该数据库也会遇到一些问题,我们根据实际工作经验,介绍几个问题的解决方法实例。系统的工作环境:硬件为PC/AT系列微机,硬盘在40MB以上;软件为XENIX操作系统,INFORMIX数据库管理系统(版本为1.10)。

## 二、用C语言的结构定义

### 使用INFORMIX的记录字段

INFORMIX数据库管理系统提供了很强的C语言接口(ALL-II)功能,它提供了一个结构视图dbview用于说明INFORMIX的记录字段作为C语言变量的信息<sup>[2]</sup>,但是在INFORMIX版本1.10中,系统是将每一个字段作为简单变量在C语言接口中定义的,要将INFORMIX的记录字段作为C语言的结构乃至结构数组变量来定义,则由于C语言的编译系统和ALL-II之间存在着差异,相应的变量在内存中分配的实际地址往往会发生差异,作为结构数组使用时就不能保证得到正确的结果。两者的差异见图1。

由图1可见,ALL-II定义的字符串出现奇数位时,则补上一个空bit,以保持字符

串的起始地址始终为偶数,其目的是为了加快程序的执行速度(详见8086CPU汇编手册)。而C语言的结构定义字符串变量没有这种要求,因此不能直接使用C语言结构,否则取出的数据将不正确。若不采用C语言的结构来定义变量,那将给编制程序带来许多不便,如不能使用循环语句来处理变量,则程序量将大大增加。为了解决这个问题,我们采用先用ALL-II定义记录缓存区(即按简单变量定义相应的记录字段),然后再定义一批指针变量,这些指针按需要指向所定义的简单变量,而这些指针本身是采用结构定义的,这样就使简单变量转化成间接的结构数组,这在以后的程序处理中就可通过结构数组处理来实现对简单变量的处理,给程序编制带来许多方便。定义方法如下:

```
struct
{
    char jiedai0 [3];
    char kemu0 [13];
    double jine0;
    double shuliang0;
    char jiedai1 [3];
    char kemu1 [13];
    double jine1;
    :
    double jine3;
    double shuliang3;
} rec - pz;
struct
{
    char * jiedui;
    char * kemu;
    double * jine;
    double * shuliang;
```



```

} item [4];
item [0].jiedai = rec - pz.jiedai0;
item [0].kemu = rec - pz.kemu0;
item [0].jine = &rec - pz.jine0;

```

```

item[0].shuliang = &rec - pz.shuliang0;
:
item [3].jine = &rec - pz.jine3;
item[3].shuliang = &rec - pz.shuliang3;

```

定 义	内 存 实 际 存 放 情 况
<p>C语言结构定义</p> <pre> struct {     struct item [4]; } rec - pz; struct item {     char jiedai [3];     char kemu [13];     double jine;     double shuliang; }; </pre>	
<p>ALL - II 定义</p> <pre> struct {     char jiedai0 [3];     char kemu0 [13];     double jine0;     double shuliang0;     char jiedai1 [3];     char kemu1 [13];     double jine1;     double shuliang1;      double jine3;     double shuliang3; }; </pre>	

注：\表示字符串的结束标记

图 1 C语言结构定义和ALL - II 定义的差异

### 三、ACE报表处理中 C语言接口程序

#### 1. 问题的提出

在报表处理中，经常会遇到一些ACE报表生成语言无法解决的问题。这时就需要

利用C语言接口来扩充ACE的功能。现有这样一个时间运算问题：如甲用时1小时50分，乙用时3小时40分，丙用时8小时29分，合计用时13小时59分。要求按格式《13：59》的形式输出，并且要具有通用性，可在不同的报表中被引用。

#### 2. 问题的分析

INFORM IX中提供的INFORMER查询语言和ACE报表生成语言中有简单的四则运算,但没有六十进制运算和模运算,这个问题的关键在于多进制运算和将运算结果转化成字符串输出。另外还要考虑调用格式通用。

### 3. 问题的解决

首先我们在建立数据库时分别为时和分建立两个字段:时字段为time,合计时字段各为t-time,分字段为minute,合计分字段为t-minute。分别将时和分送入time, minute字段中,然后在INFORMER中将time字段累计存入t-time中,minute字段累计存入t-minute中。这样对于C语言来讲有了传输的参数。再利用C语言编制程序(get-t.c)<sup>[4]</sup>,对t-minute字段作六十进制运算,将六十的倍数加到t-time字段中,余数存入t-minute字段中,再将两个字段合成字符串,并在时和分之间插入冒号以待输出。C语言程序见图2。

```
# include "ace.h"
extern acevalue get-t();
struct ufunc userfuncs[] =
{
    "get-t", get-t,
    0, 0
};
acevalue get-t(f1, f2, time)
acevalue f1, f2, time;
{
    long t1, t2, t3;
    int i;
    t1 = f1->v-double;
    t2 = f2->v-double;
    t1 = t2/60;
    t2% = 60;
    for(t3 = t1, i = 4; i >= 0; i--)
    {
        if(t3 = 0 && i < 4)
            time->v-charp[i] = ' ';
        else
            time->v-charp[i] = t3%10+'0';
    }
}
```

```
t3/ = 10;
}
time->v-charp[5] = ':';
for(t3 = t2, i = 7; i > 5; i--)
{
    time->v-charp[i] = t3%10+'0';
    t3/ = 10;
}
}
```

图2 get-t.c源程序

编译链接方法为《CC get-t.C-lace-lldb-0 get-t》在执行ACE程序时用get-t替代acego命令即可。也可用《ln get-t acego》处理,执行ACE程序时,仍用acego命令,具体调用格式如下:

```
database 数据库名 end
define
    function get-t
        variable time type character 8
    end
    :
format
    :
call get-t ( t-time, t-minute, time )
print time
    :
end
```

## 四、INFORMIX中各数据库的数据共享

在数据库管理中,经常会遇到各数据库之间的数据共享和数据传输问题。这也是多用户系统比单用户系统优越之一。但在实际应用中往往不容易掌握。

现有这样一个问题:在用户user1和用户user2下有两个不同的数据库A和B。在A数据库中有库文件file,现要求用户user2下的B数据库可共享file文件中的数据,有两种共享情况:

① file文件是用户user1下的数据库A的库文件,用户user1可对其数据有增、删、改和查的权限,现要求用户user2可读file文件的数据,但不具备增、删、改的权限。



② 要求用户user2对file文件的数据享受与用户user1同样的权限。

根据上述要求,解决的步骤分为4步。

第1步,将user1和user2两用户建在同一用户组下,方法为将etc目录下的group文件调出,通过vi编辑手段,将user1和user2划为同一组<sup>[3]</sup>。

第2步,在用户user2下的B数据库中建立一个与file文件结构完全一样的文件file1(文件名可不一样)。通过编译后在B数据库下可产生数据文件file1.dat和索引文件file1.idx

第3步,用ln命令进行逻辑链接。命令格式如下:

```
ln /usr/user1/ file.dat /usr/user2/
file1.dat
```

```
ln /usr/user1/ file.idx /usr/user2/
file1.idx
```

当链接成功后,对用户user2来讲file文件从逻辑上看搬到了自己的数据库B下的file1文件中来了,但实际上物理位置并没有改变,它只是建立了一条库文件之间的链。

第4步,对file.dat和file.idx文件进行权限限制。对第一种共享要求,可限制file.dat, file.idx文件只有当前用户有读、写权,同组用户有读权,其它用户无读、写权,命令为chmod 0640 file.dat、chmod 0640 file.idx。第二种共享要求,可限制file.dat、file.idx文件为同组用户具有读写权。命令为chmod 0660 file.dat、chmod 0660 file.idx。

上述解决数据库数据共享的方法是非常简便有效的。

## 五、INFORMIX数据库管理系统中的char字段

在INFORMIX系统中,char类型字段是较为常用的一种数据类型。也是较易产生问

题的一种字段类型。在查询过程中,当判定条件处于临界状态时,往往出现用户所不希望的结果。因此我们设计了一个测试用的数据库及一些典型数据,对数据库进行测试以求对char型数据有一个较深入的了解。

例如 数据库定义

```
database test
file test
field test1 type char length 10 index
field test2 type char length 10
```

测试用数据

record No.	field1	field2
1	aaaaaaaaaa	aaaaaaaaaa
2	▯bbbbbbbbb	▯bbbbbbbbb
3	▯▯cccccccc	▯▯cccccccc

测试方法

在INFORMER查询系统下,用以下步骤进行测试:

step1 read into x test

step2 用print x 语句,条件分别为:

- ① where test1 ( test2 ) = " " ;
- ② where test1 ( test2 ) = "▯" ;
- ③ where test1 ( test2 ) = "a" ;
- ④ where test1 ( test2 ) = "▯a" ;
- ⑤ where test1 ( test2 ) = "▯b" ;
- ⑥ where test1 ( test2 ) = "▯c" ;
- ⑦ where test1 ( test2 ) = "▯▯c" ;

显示结果如下:

	test1 ( 关键字 )	test2 ( 非关键字 )
①	▯bbbbbbbbb	▯bbbbbbbbb
	▯▯cccccccc	▯▯cccccccc
②	▯bbbbbbbbb	▯bbbbbbbbb
	▯▯cccccccc	▯▯cccccccc
③	aaaaaaaaaa	aaaaaaaaaa
④		
⑤	▯bbbbbbbbb	▯bbbbbbbbb
⑥		
⑦	▯▯cccccccc	▯▯cccccccc

根据测试结果,可得出如下结论:

在INFORMIX数据库管理系统中,用SQL语言对数据库进行查询时,字符型数据有效字符后面都用通配符“.”填充。字

# 在 VAX/VMS 应用程序中 控制打印机的一种方法

兰州电机厂计算机中心 王 勇

## 一、前 言

VAX/VMS 8350 是美国DEC公司生产的超级小型机,它所带的打印机有两种工作方式:终端打印机(联机打印)和系统打印机(脱机打印)。使用终端打印机,必须用setup键进入屏幕设定状态,选择打印功能,然后键入命令实现打印;对于系统打印机,起动队列、打印文件和删除作业等操作须在终端上键入命令(设备分配和初始化队列一般在系统开工文件 systart.com 中),以上操作均须在操作系统下执行。本文给出了在PASCAL应用程序中控制打印机的简便方法,并附有程序实例。

## 二、联机与打印机功能选择

对于终端打印机,首先应联机并选择打印功能,这是通过终端设置实现的。下表是终端控制打印机的几条命令,CSI 是一个控制序列,可以表示为ESCI。

终端打印机控制命令

方式	序列	动 作
自动打	CSI?5i	设置终端成自动打印方式
印方式	CSI?4i	退出自动打印方式
控制器	CSI 5i	设置终端成控制器方式
方 式	CSI 4i	退出控制器方式

因此,用PASCAL程序联机并选择自动打印方式的语句如下:

```
writeln (chr (27), '[?', '5i');
```

其中 chr ( ) 为将ASCII码转换为字符的内部函数。退出自动打印方式的语句为:

```
writeln (chr (27), '[?', '4i');
```

执行后,终端状态为正常打印方式。打印机控制器方式选择与此相同。

## 三、运行时间库过程LIB\$SPAWN 的调用技术

不论对终端打印机还是系统打印机,打印文件都可以通过在程序中派生一子进程来实现。运行时间库过程LIB\$SPAWN为我们提供了这一功能。LIB\$SPAWN请求调用进程的命令行解释程序(CLI)为执行CLI命令生成一子进程。该子进程继承了调用进程环境中的下述属性:

- 进程逻辑名
- 全局和局部CLI符号
- 缺省设备和目录
- 进程特权
- 进程不可扣除的限额
- 当前命令校验设置

要从PASCAL程序中请求运行时间库子程序,必须先将其说明为一个外部函数,然后再用函数说明符调用它。下面列出了PASCAL对LIB\$SPAWN的调用接口和传参技术。变元仅讨论本文所用到的,其余的可查阅有关资料。

格式

```
ret - status := lib$spawn (command -  
string, input - file, flags);
```

变元

command - string: 要由生成的子进程执行的一条CLI命令。通过描述符传递。

input - file: 与该子进程的逻辑名表中



的逻辑名sys\$input 相关联的等效名, 缺省时为调用者的 sys\$input。通过描述符传递。

flags: 指定选择行为的标志位的长字。通过引用传递。flags的定义是:

位0: 置位,父进程与子进程并发执行;复位,父进程冬眠,直到子进程完成。

位1: 置位,生成的子进程不继承从它的调用者来的CLI符号;复位,子进程继承所有当前定义的CLI符号。

位2: 置位,生成的子进程不继承从它的调用者来的进程逻辑名;复位,子进程继承所有当前定义的进程逻辑名。

位3至位31: 系统保留作扩充用,必须为零。

ret-status: 返回一整型常数,程序员可根据返回状态码编写条件处理程序。

#### 四、程序举例

例1 应用于终端自动打印的程序

```
program termprn(input, output, p);
var
  p: text;
  sign, ret-status: integer;
  fn: varying[10] of char;
  str: varying[10] of char;
[external] function lib$spawn (cs:
varying[a] of char; inf: varying[b] of
char; flags: integer): integer; extern;
procedure create-input;
begin
  open (p, 'infile.com', new);
rewrite (p);
  writeln (p, '$ TYPE ', fn);
  close (p);
  fn := 'infile.com';
end;
begin (* main *)
  write ('打印文件名: ');
```

```
  readln (fn);
  str := ''; sign := 0;
  create-input;
  writeln (chr(27), '[?', '5i');
  ret-status := lib$spawn (str,
fn, sign);
  writeln (chr(27), '[?', '4i');
end.
```

变元 str 赋值为空, lib\$spawn 生成的子进程将以 infile.com 的内容为命令取出执行, 这样使得打印过程成为非交互式。create-input 为产生 input-file 的过程, input-file 文件名为 infile.com, 内容为 "\$ TYPE 打印文件名"。语句 sign := 0, 使得调用进程冬眠, 直到子进程执行完打印功能。writeln (chr(27), '[?', '5i'); 和 writeln (chr(27), '[?', '4i'); 分别为终端自动打印方式的选择和退出语句。打印完成后, 控制返回调用进程。

例2 应用于系统打印机的程序, 变量说明与上例相似, 下面列出主程序部分。

```
program sysprn (input, output);
  变量说明部分;
begin (* main *)
  str := '$start/queue txb7: ';
  fn := 'sys$input';
  sign := 0;
  ret-status := lib$spawn (str,
fn, sign);
  str := '$print/queue = txb7: ';
  ret-status := lib$spawn (str, fn,
sign);
end.
```

语句str := '\$start/queue txb7: '的作用是启动系统打印队列, txb7为打印队列的系统逻辑名。sign := 0保证了首先启动打印队列。str := '\$print/queue = txb7: '为打印命令, 执行该语句时, 系统将提示输  
(下转第37页)

# 如何使五笔字型汉字适用多种打印机

石家庄市第一建筑工程公司计算机室 阎立伟

## 一、概 述

五笔字型汉字系统，因其独到的汉字输入方式，而被广大计算机工作者所采用，各种PC机均配有该系统。但是，该汉字系统配备的打印机驱动程序种类很少，因此，其应用受到限制；另外，其汉字显示字库与其它汉字系统相比，也不够美观。而CCDOS 2.13汉字系统，显示字形美观，且支持几乎所有类型的打印机，具有强大的汉字输出功能。能否将二者结合在一起使用呢？笔者认为影响汉字系统不兼容的一个重要原因是汉字库的结构不同，若五笔字型使用CCDOS 2.13的汉字库，也就可以使用CCDOS 2.13的打印机驱动程序了，这样五笔字型的应用范围将大大扩展。现将实现方法简述如下。

## 二、五笔字型的引导过程

五笔字型的引导主要依靠DAO·EXE、ZHENG·EXE两个文件来完成。

### 1. DAO·EXE的主要功能

检查汉字库WBK的完好性，以及为汉字库申请内存空间，并保留其申请到的内存空间首地址。图1为DAO·EXE程序框图，该程序的最后一步用了系统提供的“驻留、退出运行”的功能，这就是说，在该程序结束时，不释放它占有的内存空间。因为该程序仅仅替汉字库申请到了内存空间，并没有将汉字库调入内存。把汉字调入内存的工作由ZHENG·EXE来完成。

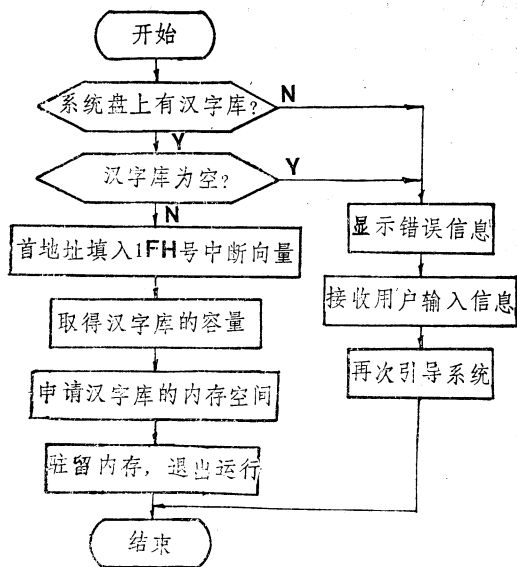


图1 DAO·EXE程序框图

### 2. ZHENG·EXE的主要功能

将汉字库调入内存指定区域。首先取1FH号中断向量的值，这里存放的是内存中汉字库空间的起始地址（由DAO·EXE填入），然后将其值保存在0075单元中，这是因为1FH号中断向量要放入新的内容（字符字模库地址），以便今后访问汉字库时能正确取到字模；接着把汉字库（WBK）调入内存，由3FH号系统功能调用来完成这项工作。每次读入8000H个字节，查到3FH号系统调用返回的读入字节为0，即为汉字库读完。装入工作结束。

## 三、汉字库与机内码转换

### 1. 汉字库分析

CCDOS 2.13汉字系统的汉字库（HZK

16)容量为261396Byte,而五笔字型汉字系统的汉字库(WZK)的容量仅为240640Byte。所以要利用CCDOS 2.13的汉字库,DAO·EXE 申请内存时的程序必须修改,以适应HZK16 汉字库。同时修改调用的汉字库名字,见表1。

表1 HZK16和WZK汉字库内容

区 号	HZK16字库	WZK字库
1—5	符号	符号
6	希腊字母表	希腊字母表
7	俄文字母表	俄文字母表
8	空区	制表符
9	汉字制表符	汉字制表符
10	斜体英文字符	汉字
11	上标符号	汉字
12	下标符号	汉字
13—15	空区	汉字

HZK16字库16-87,WZK字库16-80为汉字区

从表1知两种汉字库的结构是不尽相同的,13区以后的内容是一致的,HZK16字库共87区,而WZK字库有80区。要顺利使用HZK16就得对汉字机内码转换程序进行修改。

## 2. 机内码转换

这一子程序在ZHENG·EXE程序中,起始地址29F2。框图及转换程序见图2、图3。

入口(DX)=汉字机内码,出口参数:(DX)=字模地址(以段地址形式表示),图2子程序框图,这个结构与汉字库有着密切的联系,从表1可知,HZK16比WZK多7个区,就要修改此段程序。

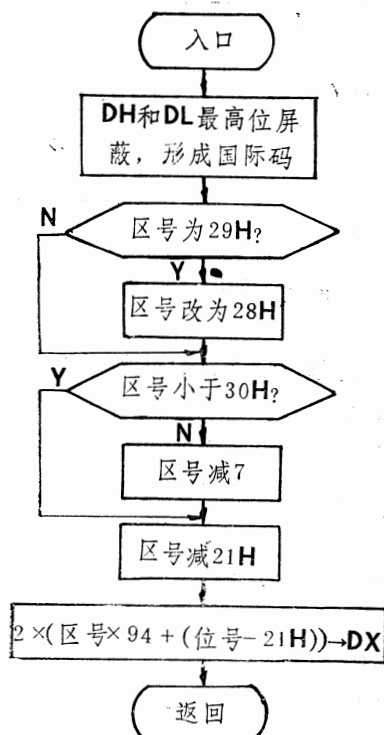


图2 机内码转换子程序框图

```

429F: 29F2 81E27F7F AND DX,7F7F
429F: 29F6 80FE29 CMP DH,29
429F: 29F9 7502 JNZ 29FD
429F: 29FB B628 MOV DH,28
429F: 29FD 80FE30 CMP DH,30
429F: 2A00 7203 JB 2A05
429F: 2A02 80EE07 SUB DH,07
429F: 2A05 80EE21 SUB DH,21
429F: 2A08 B05E MOV AL,5E
429F: 2A0A F6E6 MUL DH
429F: 2A0C 80EA21 SUB DL,21
429F: 2A0F 32F6 XOR DH,DH
429F: 2A11 03C2 ADD AX,DX
429F: 2A13 D1E0 SHL AX,1
429F: 2A15 1F POP DS
429F: 2A16 2E CS:
429F: 2A17 03067500 ADD AX,[0075]
429F: 2A1B 8BD0 MOV DX,AX
429F: 2A1D 59 POP CX
429F: 2A1E 5B POP BX
429F: 2A1F 5F POP DI
429F: 2A20 C3 RET
  
```

图3 机内码转换子程序

# 计算机图形学中的背景移动算法浅析

中国科学技术大学 柏林森

## 一、前言

在计算机动态图形程序设计中,主要有两种技术,一是前景移动,背景不动;二是背景移动,前景不动。前景移动的优点是算法简单,处理速度快,但当前景活动范围较大时,背景完全不动就不能满足要求了。本文讨论背景移动的算法,考虑到算法的实现环境和各语言及数据结构表示的差异,算法用类pascal的伪语言给出,略去说明语句。

将背景所在的坐标系称为世界坐标系,把显示屏幕看成取景框,背景的移动可看成取景框在世界坐标系中的反向移动。取景框的坐标系称为取景坐标系,取左下角为坐标原点 $(0, 0)$ 。设此原点在世界坐标系中的坐标为 $(x_0, y_0)$ ,则某点的世界坐标 $(x, y)$ 和取景坐标 $(x', y')$ 的转换关系为

$$x' = x - x_0 \quad y' = y - y_0$$

实际的作图是在取景坐标系中进行的,为描述方便,将取景框取为长方形,四个端点坐标为: $(0, 0)$ ,  $(x_s, 0)$ ,  $(0, y_s)$ ,  $(x_s, y_s)$ 。

## 二、背景移动算法的初步设计

背景中的图形都看成由直线构成,曲线亦是由直线段构成。构造一张直线端点表 $s$ ,表的每一项存放一条直线段两个端点的坐标。再构造一个显示档案,其中存放当前在取景框中的,即需要显示的背景直线段。当取景框原点处于 $(x_1, y_1)$ 点,查上述端点表,某一直线段端点 $(x_2, y_2)$ 如满足: $0 \leq x_2 - x_1 \leq x_s$ ,  $0 \leq y_2 - y_1 \leq y_s$ ,

则此端点在取景框内。若一条线段的两个端点都在取景框内,此线段必全部可见,挂入显示档案。若一条线段只有一个端点在框内,则是部分可见。求此线段与取景框边界交点,将可见点和交点挂入显示档案。若两端点均不可见,则此线段不一定完全不可见,这要求它和框边线的交点。这实际上是取景框对直线段的裁剪,可采用Sutherland—Cohen算法,将裁剪结果挂入显示档案。查表结束后,转显示子程序,擦去取景框中原有内容,将显示档案中的线段画入取景框,也就是屏幕上。取景框作 $dx, dy$ 移动,令 $x_1 = x_1 + dx$ ,  $y_1 = y_1 + dy$ ,重复以上查表、显示过程,算法如下:

```
    / 直线端点表放于数组s(n, 4)中。
s(n, 1)、s(n, 2)及s(n, 3)、
s(n, 4)分别是两端点坐标 /
    procedure backmove(x1, y1);
    / (x1, y1)为取景框原点在世界
    坐标系中的坐标 /
    Begin
    for n := 1 to all do
    begin
    cx1 := s(n, 1) - x1;
    cy1 := s(n, 2) - y1;
    cx2 := s(n, 3) - x1;
    cy2 := s(n, 4) - y1;
    sutherland(cx1, cy1, cx2,
    cy2, dis);
    / 调 Sutherland—Cohen 算法
    裁剪, 结果挂入显示档案 dis /
    end;
    clear; / 清取景框 /
    display(dis) / 显示子程序 /
    end;
```



上述背景移动算法实现较为简单,在背景图形不复杂的情形下是可用的。但有三个弱点:

①此算法要查遍整个背景直线段表,当背景面积远大于取景框面积,直线段相当多时,其运算量是很可观的,不能满足动画设计的实时要求。

②在显示当前档案时,需要擦去以前的显示内容,此算法需要涂抹整个取景框,耗时较长,容易出现闪烁。

③背景如主要由填充的色块图形和不规则点阵,如字符等构成,要细分成直线段处理虽然也可以,但过于麻烦,需要扫描转换等预处理。

### 三、提高动态图形显示的实时性

为了解决第一个弱点,设此背景中直线的最大长度为 $a$ ,如图1。

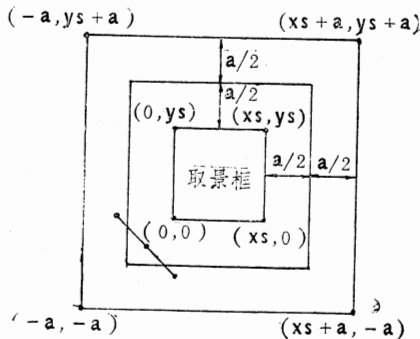


图 1

易见,凡和取景框相交的线段,必落在 $(-a, -a)$ ,  $(x_s+a, -a)$ ,  $(-a, y_s+a)$ ,  $(x_s+a, y_s+a)$  四边形范围内,其中点必落在在四边形 $(-a/2, -a/2)$ ,  $(x_s+a/2, -a/2)$ ,  $(-a/2, y_s+a/2)$ ,  $(x_s+a/2, y_s+a/2)$  内,此四边形称为检索四边形。

这样我们就只需处理中点落在检索四边形内的线段,大大缩减了处理范围。算法变成了:

①选定适当的 $a$ ,将长度大于 $a$ 的线段分成多段长度小于或等于 $a$ 的线段;

②建表;

③对于取景框的每一个位置,确定中点落在检索四边形内的线段;

④对这些线段用 Sutherland—Cohen 算法裁剪;

⑤显示。

其中,第三步是关键。如果用检索全背景线段表的办法确定四边形内的线段,与原算法相比,本质上并无提高,可采用牺牲存储量来换取速度的办法。

将背景的纵横坐标分别分成宽度为 $b$ ,  $c$ 的窄带,如图2。

给纵横交错的小块编上坐标 $(m, n)$ ,表示它是横第 $m$ ,纵第 $n$ 块。建立表 $ss(m, n)$ ,  $ss(m, n)$ 中放的是中点落在 $(m, n)$ 块中的背景直线段。此表可由计算机自动生成,见图3。

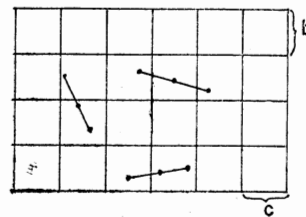


图 2

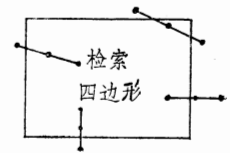


图 3

为了确定中点落于四边形内的线段,只需查 $ss$ 表中与四边形有相交部分的小块就行了。对于四边形的每一移动 $dx, dy$ ,中点落在四边形中的大部分线段是不变的。例如向左移动一步,变化的只是从左边框移进的新线段和从右边框移出的老线段。设 $check$ 中放的是中点落于四边形中的线段,四边形的每一移动就对 $check$ 执行一次增删,只需检查和左、右边框相交的小块。这些小块的确是容易的:令 $m_1 = \text{INT}((x_1 - a/2)/b)$ ,  $m_2 = \text{INT}((x_1 + x_s + a/2)/b)$ ,  $n_1 = \text{INT}((y_1 - a/2)/c)$ ,  $n_2 = \text{INT}$

$((y_1 + y_s + a/2)/c)$ ,  $(x_1, y_1)$  是取景框原点的世界坐标, INT是取整函数。需检查的即  $(m_1, n_1) - (m_1, n_2)$  的  $n_2 - n_1 + 1$  个和  $(m_2, n_1) - (m_2, n_2)$  的  $n_2 - n_1 + 1$  个小块。以上讨论是对四边形在一步移动中没有跨越小块进行的。如跨越了小块, 检索两组小块即可。y 向的移动类似, 斜向移动可分解成 x 向和 y 向的移动。a, b, c 的确定在处理时间和存储量之间作一权衡。

改进后的背景移动算法如下:

- ①由s表建立ss表, 确定check表初始内容;
- ②对每一步移动对check表进行增删;
- ③调 Sutherland—Cohen 算法对 check 表中的直线进行裁剪, 结果挂入显示档案 dis;
- ④显示dis;
- ⑤转②步。

改进算法的检索时间大大减少, 如适当选取a, b, c, 其时间复杂度与背景大小, 线段多少基本无关。可收到较好的效果。

#### 四、避免闪烁的技术措施

解决第二个弱点的办法可以采用双显示档案, 一档案为当前显示档案, 另一档案为前一次显示档案。在显示过程中, 先应用老档案以底色擦去原来图形, 再利用新档案绘出当前图形。背景作一新的移动时, 将新的显示内容写入原来的老档案, 原来的当前显示档案就变成了现在的老档案。由于每次只需擦去原来显示线段, 不需要涂抹整个取景框, 速度可以提高, 避免闪烁的出现。算法如下:

```

/dis 1 和 dis 2 为两显示档案 /
creat (dis); /产生当前档案 /
clear (dis 2);
display (dis 1);

```

mov (dis 1, dis 2) /将当前档案变成下一次老档案 /

#### 五、对不规则点阵构成的背景处理

背景由不规则点阵构成时, 应将每个点阵作为一个显示单元, 显示档案 dis 和显示过程均作一相应修改。首先, 用逼近的多边形将点阵包围起来, 多边形的每一条边象其它背景线段一样写入ss表, 显示档案中如有此多边形的边, 显示内容则为此点阵。在显示过程中, 判断此点阵中的每一个点, 若在取景框中则显示, 否则隐去。

#### 六、两种算法的比较

我们从理论上对新、老算法的运算时间作一比较。设背景共有 n 条线段, 背景面积为  $s_0$ , 不失一般性, 设在整个背景上线段的分布是均匀的, 则一块面积内包含的线段数和面积大小成正比。设一条线段作裁剪耗时  $t_1$ , 确定一条线段是否落于裁剪框内需时  $t_2$ 。对于老算法, 一次移动检索裁剪需时  $t = nt_1$ 。对于新算法, 一次移动最多有  $2 \left( \frac{x_s + a}{b} + \frac{y_s + a}{c} \right)$  个小块被检索, 需时  $2 \left( \frac{x_s + a}{b} + \frac{y_s + a}{c} \right) \left( \frac{bc}{s_0} \right) nt_2$ , 裁剪框面积为  $(x_s + a)(y_s + a)$ , 裁剪需时  $\frac{(x_s + a)(y_s + a)}{s_0} nt_1$ , 一次移动新算法总需时

$$t' = 2 \left( \frac{x_s + a}{b} + \frac{y_s + a}{c} \right) \left( \frac{bc}{s_0} \right) nt_2 + \frac{(x_s + a)(y_s + a)}{s_0} nt_1$$

新旧算法时间比

$$\frac{t'}{t} = \frac{(x_s + a)(y_s + a)}{s_0} + 2 \left( \frac{x_s + a}{b} + \frac{y_s + a}{c} \right) \left( \frac{bc}{s_0} \right) \frac{t_2}{t_1}$$

一般说来, 检索时间  $t_2 \ll t_1$ ,  $bc \ll$

# 交互图形屏幕编辑系统

浙江大学 CAD&CG 国家重点实验室 罗国明

**提要** 本文提出的交互图形编辑技术,完全采用交互化和动态化的技术,这种方法使得图形在屏幕上显示具有动感,并且速度快,交互性好,给用户提供了一个良好的界面。

## 一、概 述

交互图形编辑是计算机图形学重要组成之一,在 CAD/CAM 中起着极其重要的作用。它给设计者带来极大的方便,它能合理地安排图形在屏幕上的位置,修改在设计过程中发生的错误,或者随心所欲地改变图形的形状。此外,交互图形编辑还具有使用简单、直观、人机对话、子图形拖动等功能,所以广为用户所欢迎。

## 二、交互技术

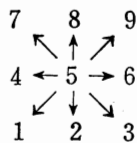
### 1. 实时光标

实时光标是交互图形编辑中使用得最基本的一种手段,也是极其重要,几乎在图形编辑过程中每做一步,都要采用到光标的移动,所以设计好光标是一个重要的环节。我们采用了+字光标 $30 \times 30$ 的长度,每走一步,前面的光标就自动消失,留下后面的光标,并采用了显示终端现有的右边键盘的功能,可以任意8个方向自由移动,操作非常方便和简单。当进入光标子程序时,终端屏幕显示

Please input start point of cursor(x,y)!

如果用户按回车键,光标就出现在屏幕中间,否则就出现在你输入点的位置。输入完光标的起始点坐标后,屏幕上出现

Please use function key!



其中 1、2、3、4、6、7、8、9 这 8 个方向用户可以随意选择,如果按

RETURN 键,则表示上次输入的数据不变,即光标移动的方向不变;如果按 5,则表示这点的坐标已被记录下来,并退出光标功能。

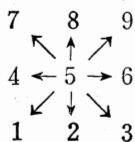
### 2. 橡皮筋

橡皮筋功能作为一种交互手段,输入任意形状的图形非常方便,它在图形上的功能就象橡皮筋一样,可以随便伸缩和任意方向移动,用户可以利用键盘在屏幕上自由作图。当进入橡皮筋子程序时,终端上就显示出

Please input a start point  
of ruband (x, y)!

当起始点输入后,图形显示器上就出现+字光标,终端屏幕上显示

Please use function key!



这时用户可以利用键盘 8 个方向随意移动,当按 3 时,橡皮带向 $-45^\circ$ 方

QUIT; Please input q 向直线移动;如果这条线太长了,就可以按 7,这时直线向 $135^\circ$ 方向移动,直线缩短到满意为止。按下 5,这条直线就固定下来,并把这条线的坐标存贮起来。按下又可作下一条直线,直到输入 q,退出橡皮带功能为止。

### 3. 调色板

画面编辑时颜色选择非常重要，一幅漂亮的图形必须用各种不同色彩表示，特别在隐藏面消除和真实感图形中更是如此。我们采用调色板技术是在图形显示器屏幕的下方显示出一组各种颜色的方块，见图1，下排是显示器的基本颜色，上排是用户选择和下排对应的颜色。

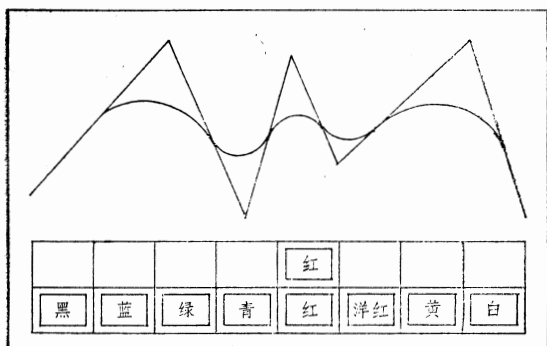


图1 调色板和橡皮筋产生的自由曲线

当用户进入调色板子程序时，终端屏幕上显示

Please select color board; default

[RETURN] !

当按下RETURN键时，表示缺省状态，就是说从上排第一块颜色开始选择颜色，一块一块地显示过去，直到找到满意的颜色为止，每显示一块，前面一块的颜色就自动消失。如果按数字键，则调色板就显示所需要的第n块颜色，当调色好后，调色板就自动消失，并且屏上显示出的颜色就是刚才调好的颜色。如果要改变颜色，只要再进入调色板，就可以得到你所满意的颜色。

图1中小方块表示各种颜色，例如写着“红”字表示屏幕上这个小方块显示出全部红色。

#### 4. 更换整幅图形颜色

当整幅图形显示后，若觉得这幅图形的颜色不够满意，可利用更换颜色程序来任意选择所需要的颜色。当进入这个子程序后，图形显示器上显示出调色板，终端屏幕上显示

Do you want to change use—color,

(Y/N)?

按N键说明不需要更换这幅图形的颜色；按Y键则说明要更换颜色。这时图形又显示出

Please select color board; default

[RETURN] !

用户就可以从第一个方块颜色开始。改变整幅图形的颜色，也可以直接按数字指定某一块颜色，直到找到满意的颜色，这时显示器的图形全部改变为当前指定的颜色。

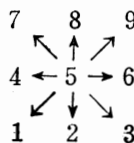
改变整幅图形颜色的方法主要是寻找帧存贮器中的颜色查找表，我们的显示器每个像素的颜色对应4位，即该图形显示器可以显示16种颜色。如果要改变显示器上的某颜色，只要把帧存贮器的颜色查找表中相应的值改为所需要的颜色值，则图形也就改变为所指定的颜色。

### 三、输入技术

二维图形输入，只要通过光标或橡皮筋的移动就能确定位置，把整幅图形输入到计算机，也可在屏幕上自由作图。而三维数据在屏幕上直接输入，在当前还是一个难题，人们往往对图形的产生和显示感兴趣，而往往忽视输入问题，尤其是三维输入。以下讨论的三维输入，实际上是工程上常用的二维半问题。

#### 1. 二维半圆形的输入

对于二维半的物体，可以利用一个物体平面的一些性质，用橡皮筋来作出立体图。屏幕的水平方向定为x轴，垂直方向定为y轴，再以适当的角度用橡皮筋移动作为z轴，具体移动如下：



QUIT; Please input q

其中 4, 6 表示x方向移动。



2, 8表示y方向移动。

1, 9表示z方向移动。

7表示 $x-1, y+1, z+1$ 移动。

3表示 $x+1, y-1, z-1$ 移动。

5表示把一个面的坐标点存贮起来。

q表示退出输入。

按以上方法移动橡皮带, 逐面输入就可以把一个立体图输入到计算机中。

## 2. 复杂物体图形的输入

按以上方法只能输入比较简单的物体, 复杂的物体这种输入有些困难, 关键在于不知道哪一些面已经输入, 哪一些面还没有输入, 使得图形混淆起来, 而且输入的时间也很长。对于复杂的物体, 经分割后还是二维半的物体, 但分割一个复杂的物体并不那么容易。我们的办法是利用Sweeping技术, 首先输入一个面, 把这个面向z轴(或x, y轴)平移一个 $\Delta z$ , 然后由计算机自动找出平移后物体的每个面来产生一个基本体, 如果是相同的体, 可由这个基本体进行仿射变换就可产生出一个较为复杂的物体了。

例如要产生桌子这样的复杂体, 先产生一个长方形面, 经过sweeping平移技术得到一个长方体, 再通过仿射变换产生出桌子, 然后利用连贯性扫描线隐藏面消去, 即可得到一张有色彩的漂亮桌子。这种方法对输入是十分有用的, 只要输入一个面, 并不需要输入物体的其它面, 即可产生一个物体, 在输入技术中是值得一提的。

## 四、图形编辑

### 1. 菜单的内部表示

菜单的形式在屏幕的右边显示出来, 提供用户用光标或键盘自由选择, 并以多级子菜单的方式出现。子菜单的任意级都可返回到主菜单或上一级子菜单, 而且用户可加入自己设计的菜单。当用户一进入这个系统, 屏幕右边就显示出一幅主菜单, 提供用户选

择。如选择好子菜单, 主菜单马上自动清除, 并显示出子菜单。如果用户选择好了图形, 要退出这一层子菜单, 用户可以回到上一层菜单或主菜单上, 非常灵活方便, 菜单的表示如下:

### 主菜单

<menu> :: = <基本曲线> | <基本体素>  
| <曲线曲面> | <平面和交  
差运算> | <自由作图> | <局部  
放大> | <变换> | <sweep> |  
<消隐> | <橡皮带> | <调  
色板> | <更换画面色彩> |  
<图形打印> | <图形保存>  
| <重显图形> | <绘图输出>  
| <多边形>

### 子菜单

<基本曲线> :: = <直线> | <圆弧> | <椭圆  
弧> | <抛物线> | <双曲  
线>

<直线> :: = <实线> | <点划线> | <虚线> |  
<双点线> | <波浪线>

<圆弧> :: = <实圆弧> | <点划线圆弧> |  
<虚线圆弧> | <双点线圆弧>

<椭圆弧> :: = <实椭圆弧> | <虚线椭圆弧>  
| <点划线椭圆弧> | <双点  
线椭圆弧>

<基本体素> :: = <n面体> | <圆柱体> | <楔  
体> | <球> | <环> | <圆  
锥>

<曲线曲面> :: = <Bezier曲线> | <B样条  
曲线> | <Beta曲线> |  
<Bezier曲面> | <B样条曲  
面>

<平面和交差运算> :: = <和运算> | <交运  
算> | <差运算>

<sweep> :: = <拉伸sweep> | <旋转sweep>  
| <广义sweep>

<汉字及字符> :: = <汉字显示> | <汉字旋

转> | <汉字放大> | <字  
符显示> | <字符旋转>  
| <字符放大>

<消隐> :: = <隐藏线消去> | <隐藏面消去>

<变换> :: = <二维旋转> | <二维放大> |  
<图形拖动> | <二维平移> |  
<三维旋转> | <三维放大> |  
<三维平移>

<多边形> :: = <多边形裁剪> | <填色> |  
<画剖面线> | <显示多边形>

## 2. 图形的插入、删除和修改

图形的插入、删除和修改在各级子菜单中都可以进行。有时一幅图形设计好显示出来才发现某些地方有缺陷要修改——去掉某条线、移动某个点、去掉某幅子图形、加上一幅图形等，象这些情况就需要利用光标的功能移动对图形进行识别，找出图形档案中的有关信息进行修改，同时也修改好屏幕上的图形。如去掉一条直线段，只要找出两端点的坐标就行，并且在图形档案中也删去相应的信息，插入恰与删除相反，把插入屏幕的信息在图形档案中同样的记录下来，而修改本质上就是插入和删除的综合。

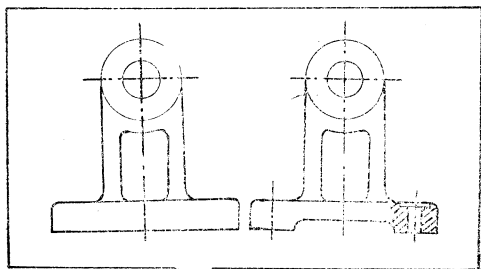


图 2 图形修改

## 五、图形拖动和布尔运算

### 1. 图形拖动

对于菜单上的子图形或屏幕上的子图形，采用图形拖动的方法安排在你所指定的

屏幕位置上，即把光标移到你需要拖动的菜单的子图形上或屏幕子图形。当然也可用键盘指定那个子图形，不管光标点到那个子图形都要能够把子图形在屏幕上进行拖动。拖动的方式和光标一样，后面的子图形自动消失，只显示当前的子图形。当移到一定的位置以后，可以根据需要进行旋转和缩放子图形，使整幅图形安排得更加合理。位置排好后就可以把整幅图形保存到图形档案中，也可以把这幅图形放到菜单中，供以后使用。键盘具体操作如下：

7 8 9 其中 4, 6 表示图形向  
x方向移动。  
4 ← 5 → 6  
2, 8 表示图形向  
y方向移动。  
1 2 3

7 表示图形向逆时针旋转。

9 表示图形向顺时针旋转。

1 表示图形缩小。

3 表示图形放大。

### 2. 平面图形布尔运算

屏幕上有n个子图形拖动到一起以后，有时需进行布尔运算后才能产生出复杂的图形。当光标点到菜单中的布尔运算时，通过字符终端指定两个运算对象，指定后，字符终端显示出

Do you want to union intersection  
or different (U, I, D)?

输入I表示对两个子图形进行求交运算，运算后自动把屏幕上两个图形清除掉，同时在图形档案中也删去，然后显示出求交后的子图形；输入u表示求和运算；输入D是求差运算<sup>[8]</sup>。

## 六、局部放大

用户要显示一幅比较大而复杂的图形时，由于屏幕大小有限，不能清晰地全部显示出来，为此必须建立一个视区和窗口，局部地显示所需要观察的图形，以便进行图形

编辑。我们在屏幕左上角建立一个视区，其位置可变；并且在屏幕的任意位置建立一个窗口，这个窗口完全是动态进行的，就象一个长方形的光标移动一样。这个长方形窗口一边移动，视区就显示出对应窗口内放大的图形，窗口外的图形利用梁友栋教授的裁剪方法就自动裁剪掉了。当窗口小于视区时，图形就放大了；反之，则图形就缩小。窗口移动和光标移动完全一样，可按8个方向任意连续运动，非常方便，而视区也对应于窗口动态地显示出。

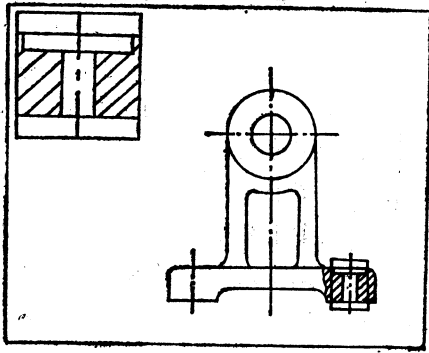


图3 局部放大

## 七、后 语

交互图形编辑系统已成功地在宇宙68000超级微机上实现。系统软件均由C语言编写，并在UNOS操作系统（类UNIX操作系统）支持下工作。整个系统均可移植到有图

形底层——图形指令或图形子程序的计算机上运行，如PC机等。对硬件没有特殊要求，一般计算机配备图形显示器、绘图机和打印机，内外存容量256KB以上就能正常工作。本系统当图形编辑全部完成后，只要把图形档案转换为绘图档案，并且转送到绘图机上，即可绘出图形。整个编辑过程，交互性强，使用方便灵活，是CAD/CAM的一个良好的支撑环境。

## 参 考 文 献

- [1] JAMES D. FOLEY, Fundamentals of Interactive Computer Graphics, Addison—Wesley, 1982.
- [2] William M. Newman Robert F. Sproull, Principles of Interactive Computer Graphics, Mc Graw—Hill 1979.
- [3] 金廷赞著，计算机图形学，浙江大学出版社，1988。
- [4] 冯星，微机图形编辑，《计算机研究与发展》，1989，8。
- [5] 应道宁 罗国明，GPSUZ 图形处理系统，《计算机绘图与辅助设计》，1986，4。
- [6] 周明伟等，微机交互式图形编辑系统，《机电工程》，1987，3。

（上接第15页）

$s_0$ ， $\frac{t'}{t}$ 的第二部分可以略去。

$$\frac{t'}{t} = \frac{(x_s + a)(y_s + a)}{s_0}$$

由此可知，背景面积  $s_0$  越大， $\frac{t'}{t}$  越

小，越能显出新算法的优越性。在实用中，曾取  $x_s = 100\gamma u$ （光栅单位）， $y_s = 100\gamma u$ ， $a = 50\gamma u$ ， $s_0 = 500 \times 500\gamma u^2$ ，此时  $t'/t =$

0.09，需时只有老算法的  $1/10$ 。用老算法，背景线段超过数十条，动画即出现明显停顿。采用新算法，背景有数百图形单元，依然能收到良好效果。

## 参 考 书

D. F. 罗杰斯著，梁友栋 石教英 彭群生译，计算机图形学的算法基础，北京，科学出版社，1987。

# 自动生成编译程序的测试实例

杭州市电信局规划设计所 冯 斌

**摘要** 软件测试是软件工程的重要研究课题之一,本文提出了一种自动生成测试实例的方法,利用属性文法,引入属性分配的思想,简化了属性计算,使自动生成上下文有关的源程序成为可能。

## 一、前 言

### 1.问题的提出

在软件开发过程中,为了保证软件的质量,人们广泛使用软件测试的方法,它包括生成有代表性的测试实例,使用测试数据执行程序,并将运行结果与预测结果进行比较,其目标是尽可能多地发现程序中存在的错误。

与其它软件相比,保证编译程序的可靠性尤为重要,因为编译程序是保证其它采用源语言描述的软件正确执行的基础。

本文介绍的是chill编译器测试系统的实例产生部分,其内容包括chill语言属性文法的输入输出,属性计算及根据属性文法随机生成chill源程序文件和预期结果文件。

### 2.目的及意义

本系统要达到的目的是:将chill语言属性文法的语法规义描述输入到计算机中,由计算机随机生成一段chill语言源程序,并同时生成预期结果文件,随后将这些生成的源程序送交chill语言编译程序进行编译,将结果与预期结果文件比较,以检验编译程序的正确性。

它的主要意义是:

- ①对属性文法中属性计算这个难题提出了简洁、优化的新算法。
- ②对自动生成上下文有关的源程序作了

有益的尝试。

## 二、总体设计思想

完整的测试系统包括实例产生部分和环境支撑部分。本文所涉及的是实例产生部分,这部分需要完成以下的工作。

- 1.将属性文法(AG)存入合理的数据结构,便于规则添加、输出、查找。
- 2.能完成对属性的初始化工作。
- 3.设置一个属性计算器完成属性计算的工作,它包括多个辅助函数的实现。
- 4.设置一个条件计算器,能完成AG中条件计算的功能。
- 5.设置合理的数据结构生成属性树。
- 6.对于生成的正确源程序,给出“预期结果”文件。
- 7.对于生成的错误源程序,给出“所含错误”文件。
- 8.利用随机数,使生成的源程序尽可能地覆盖AG子集。

## 三、基本理论

### 1.BNF文法

BNF文法定义的语言属于上下文无关语言。

$$G = (V_N, V_T, P, Z)$$

其中  $V_N$  是非终接符号集,

$V_T$  是终接符号集,



P是产生式集,

Z是开始符号。

P中的产生式为:

$P: X_{p,0} \rightarrow X_{p,1} \dots X_{p,n_p}$

$n_p \geq 1$ , 表示右部所含符号的长度;

$X_{p,0} \in VN, X_{p,i} \in VN \cup VT, 1 \leq i \leq n_p$ ,

其中  $VT$  是主语言的标记符号。

每一个程序中的语句, 如在语法上符合BNF文法, 则在句法上是正确的。

但是, BNF仅能表示上、下文无关文法, 而实际上, 一个程序的上下文是密切相关的, 所以, 通常编译程序要分析某一源程序语句串时, 首先辨认语法结构, 然后调用一语义程序, 语法分析与语义分析分别独立进行。而下面介绍的属性文法, 则是在进行语法分析的同时进行语义分析, 语法分析结束时, 语义分析也就完成了相应的工作。

## 2. 属性文法

属性文法AG是在一个上下文无关文法G上附加上下文有关的属性及其语义规则。

例如, 一个将二进制正整数转化为十进制数的例子, 含语法及语义规则如下:

语法规则

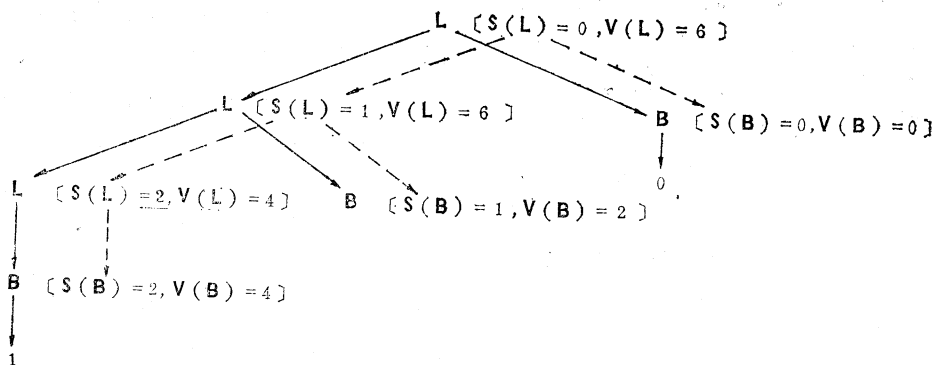
语义规则

- |                           |   |
|---------------------------|---|
| ① $L_0 \rightarrow L_1 B$ | $S(L_0) + 1 \rightarrow S(L_1)$<br>$S(L_0) \rightarrow S(B)$<br>$V(L_0) \leftarrow V(L_1) + V(B)$ |
| ② $L \rightarrow B$       | $S(L) \rightarrow S(B)$<br>$V(L) \leftarrow V(B)$   |
| ③ $B \rightarrow 0$       | $V(B) \leftarrow 0$   |
| ④ $B \rightarrow 1$       | $V(B) \leftarrow 2S(B)$   |

其中  $V$  为综合属性, 是一个正整数。

$S$  为继承属性, 是一个整数。

下面是二进制数110转化为10进制数的例子, 即一棵属性树。



从上面这棵属性树可以看到, 继承属性S是从根结点自上而下流动的, 综合属性V是从叶结点自下而上流动的, 继承属性的计算依赖其祖先, 而综合属性依赖其后代。这样, 使原来上、下文无关的文法成了上下文有关的AG文法。其精确定义见参考文献〔2〕。

## 四、问题的分析

### 1. 编译程序的测试方法

现有的软件测试方法主要有: 穷举测

试、黑箱测试、白箱测试。

穷举测试要遍历程序的所有路径, 不太可能。

白箱测试也称逻辑驱动测试, 利用程序的内部结构信息设计测试实例, 理论上也应当是穷举的路径测试, 这也不太可能。

黑箱测试又称功能测试, 即把程序作为黑箱, 测试实例的设计仅从要求, 说明书等开发阶段的说明或功能方面考虑, 没有利用程序内部结构方面的信息。但是, 测试的目

的是尽可能多地发现程序中存在的错误,并不能保证程序完全正确。

编译程序测试普遍采用黑箱测试。采用这种方法的关键是选择有代表性的测试实例,其覆盖面要尽可能大,所带来的复杂问题是如何生成上下文依赖的程序。

一种方法是以purdom算法作为程序自动生成的基础,加上处理上下文相关的信息。purdom算法的生成策略有以下两个特点。

- ①生成最短句子的最小集合;
- ②保证每个语法式至少使用一遍。

这个算法输入的是高级语言的上下文无关的语法产生式集,输出的是满足语法规范的符号串或语言。

另一种是开发一些上下文有关的文法,作为测试实例产生的依据,可以一次性地生成满足语法和语义规则的源程序。属性文法是其中一种。

西北工业大学计算机系开发的新一代测试环境——基于知识的编译自动测试系统kcats,是由一个知识库和一个知识库支持环境组成。知识库是可变的,用户可添加知识;而知识库支持环境是不变的,是kcats的推理机制,包括测试实例的生成方法和生成策略,它的生成策略是先生成语句体后生成说明部分,以达到上下文相关的目的。这是个通用的系统。

迄今,还没有一个完全自动生成随机程序测试实例生成系统。

## 2. 系统的构成

本系统以属性文法为基础,对属性文法加以修改,使综合属性都转化为继承属性,采用“属性分配器”代替“属性计算器”,从而简化了繁琐的计算,减少属性树的遍历次数,能在遍历一遍属性树后就产生一个chill源程序。

### ①属性计算器

对于每个属性文法AG,必须建立相应的属性计算器来加工处理语言L(AG)的语义属性,属性的计算不仅反映属性文法的结构和特性也准确地描述了属性文法的实际表达能力。

由前面的叙述表明:继承属性信息是沿着树自顶向下流;综合属性信息是沿树自底向上流。一个属性计算的成功是依赖于其兄弟、父亲,甚至自己的属性计算的完成。因此,属性计算的次序确定是一大难点。

一般的计算策略可用如下递归过程描述:

```
procedure    evaluate (n0)
begin
    计算n0的某些继承属性
    for i : = 1 to m DO   (设n0有m个儿子, V1~Vm)
        evaluate (Vi);
    计算n0的某些综合属性
end;
```

属性计算器通常有两种基本确定次序的方式:静态计算次序、动态计算次序。前者在构造属性计算器时确定属性计算次序,静态定义计算的策略;后者是根据每个分析对象的结构动态地确定计算策略。

### ②属性分配器

属性计算是一个十分复杂的问题,如何减少传递信息的数量,减少每个属性的依赖性,是简化这个问题的关键。

为此,本文引入“属性分配器”的概念,属性分配源于属性计算,又比属性计算简洁明了,适合于测试实例产生过程的运用。在“属性分配器”中,只有继承属性,无综合属性,同一结点的属性计算次序是随意的,它们仅仅依赖于自己父亲结点的信息,与兄弟结点或自己的其它属性无关。

下例一棵属性树是chill语言源程序的一个整型数生成过程。

为简洁起见,所含规则中未用到的可选项不列出;语义规则中有三个属性,val是

值的大小；class是值的类型；tag是值的最终标记；含两个辅助函数，Re是截取一个整型数中除末尾数外的数字，dig是截取一个整型数的末尾数。

所用规则如下：

A.  $\langle \text{literal} \rangle ::= \langle \text{integer literal} \rangle$

属性规则 1:  $\text{val}(\langle \text{integer literal} \rangle) := \text{val}(\langle \text{literal} \rangle)$

属性规则 2:  $\text{class}(\langle \text{integer literal} \rangle) := \text{class}(\langle \text{literal} \rangle)$

B.  $\langle \text{integer literal} \rangle ::= \langle \text{integer literal} \rangle \langle \text{number} \rangle$  (可选项 1)

属性规则 1:  $\text{val}(\langle \text{integer} \rangle) := \text{Reval}(\langle \text{integer literal} \rangle)$

属性规则 2:  $\text{val}(\langle \text{number} \rangle) := \text{DIG lval}(\langle \text{integer literal} \rangle)$   
|  $\langle \text{number} \rangle$  (可选项 2)

属性规则 1:  $\text{val}(\langle \text{number} \rangle) := \text{val}(\langle \text{integer literal} \rangle)$

C.  $\langle \text{number} \rangle := \langle \text{digit} \rangle$

属性规则 1:  $\text{tag}(\langle \text{digit} \rangle) := \text{val}(\langle \text{number} \rangle)$

D.  $\langle \text{digit} \rangle := 1$

属性规则:  $\text{tag}(\langle \text{digit} \rangle) := 1$

| 2

:

| 0

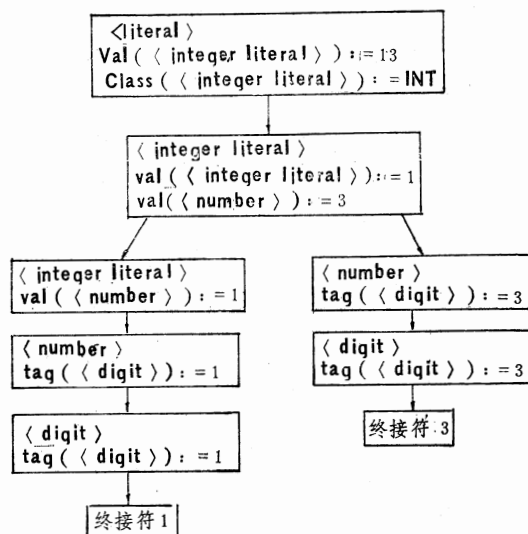
属性规则:  $\text{tag}(\langle \text{digit} \rangle) := 0$

设  $\langle \text{integer literal} \rangle$  的父亲  $\langle \text{literal} \rangle$  的属性为

$\text{val}(\langle \text{literal} \rangle) := 13, \text{class}(\langle \text{literal} \rangle) := \text{INT}$

则属性树如下：

从上面的属性树可以看到，属性分配是以已知的数据驱动的，以分配的方式代替繁琐的计算，十分简洁。它避免了属性计算中相互依赖，有时甚至出现循环计算这样的问题。



但是，为了实现这样的设想，需要把原有的chill语言属性文法改造成为适合于属性分配的文法，也就是把综合属性转化为继承属性。

### ③对chill语言属性文法的修改

为了实现属性分配，对chill的属性文法作了修改：

A. 对非终接符，将综合属性改为继承属性。

B. 对终接符，以属性匹配的方式赋予其一定的属性值。

C. 对公有的属性以表格形式存放，以节省内存。

首先，是非终接符的属性修改。

对于属于传递类的属性，修改十分方便，例如：

规则  $\langle \text{integer literal} \rangle := \langle \text{number} \rangle$   
原有的属性计算规则为：

$\text{val}(\langle \text{integer literal} \rangle) := \text{val}(\langle \text{number} \rangle)$

现在，只需改为  $\text{val}(\langle \text{number} \rangle) := \text{val}(\langle \text{integer literal} \rangle)$  即可，即改变了传递的方向。

对属于计算类或连接类的属性，修改略为复杂。

如规则:  $\langle \text{expression} \rangle := \langle \text{expression} \rangle$   
+  $\langle \text{operator} - 1 \rangle$

原来的属性规则如下:

$\text{val}(\langle \text{expression} \rangle) := \text{val}(\langle \text{expression}_1 \rangle) + \text{val}(\langle \text{operator} - 1 \rangle)$

修改后的属性规则为

$\text{val}(\langle \text{expression}_1 \rangle) = \text{RNDA1}(\text{val}(\langle \text{expression} \rangle))$

$\text{val}(\langle \text{operator} - 1 \rangle) = \text{RNDA2}$

其中RNDA1是根据 $\text{val}(\langle \text{expression} \rangle)$ 的值随机产生的一个小于等于 $\text{val}(\langle \text{expression} \rangle)$ 的值, RNDA2是 $\text{val}(\langle \text{expression} \rangle)$ 与RNDA1的差值, 它俩相加和仍符合等式。

对于连接类的属性, 也一样采用分解的方式。

经过上述修改, 消除了原来属性文法中的综合属性, 使之成为适合于属性分配的属性文法。

其次, 对于终接符, 不存在属性分配的问题, 而是让它自身带一个确定的属性值, 当分配下来的属性值正好与它自身所带的值相等, 则说明此终接符为符合属性要求的终接符。

例如 A:  $\langle \text{digit} \rangle ::= 1$

$\text{tag}(\langle \text{digit} \rangle) := 1$

| 2

:

| 0

$\text{tag}(\langle \text{digit} \rangle) := 0$

B:  $\langle \text{number} \rangle ::= \langle \text{digit} \rangle$

$\text{tag}(\langle \text{digit} \rangle) := \text{val}(\langle \text{number} \rangle)$

B是A的父亲, 若B所带的值 $\text{tag}(\langle \text{digit} \rangle) := 2$ , 当B传递给A后, 进行比较, 当相等时, 即判断 $\langle \text{digit} \rangle ::= 2$ , “2”是符合属性要求的终接符。

再者, 对于公有属性, 即仅仅起传递作用的, 可以把它归出来, 成为一张表, 表中存的是一些三元组(即变量名、类型、值),

然后, 即以这张表为基础, 驱动属性分配, 随机地生成源程序。

经过上述修改, 使属性分配的实现有了可靠的基础, 使计算机自动生成上下文有关的程序有了必要的保证。

④如何生成一个正确的chill源程序。

首先是进行初始化, 产生一张表, 内容即为一些三元组(变量名、值、类型), 然后将这张表赋给初始属性Semantics和Decs, 其中Semantics属性是用于产生动作语句的, 而Decs属性是用于产生说明语句的, 两者源于同一张表, 必然是相关的。

初始化时还需对其他一些属性赋随机初始值, 如模块名、过程名等等。并把AG从磁盘文件上以一定的数据结构读入内存中, 以便需要时查找。

有了初始属性, 即可进行属性树的生成过程了, 它的根结点是Program(即标识符号), 叶子结点是终接符, 也就是构成源程序的一些字符串。

然后, 将属性数从左至右遍历, 读出各叶子结点的值存入磁盘上的一个文件, 即为一个Chill语言源程序。

⑤如何生成一个含错误的源程序

主要的错误可分两类:

A. 语法错误, 即终接符的错误, 一般是关键字的错误, 如begin, end, if等字符串的错误, 或标点符号的误用、漏用。

B. 语义错误, 如重复定义一个变量, 表达式类型不匹配等等。

第一类错误的自动生成较为简单, 在遇到一个属于关键字或标点的符号串时, 调用一个随机出错过程, 将改写后的终接符存入源程序即导致了错误。

第二类错误的自动生成要靠属性分配时的条件计算, 属性分配的成功与否依赖于所含条件是否成立, 如条件不成立时仍进行属性分配, 必然会出现语义的错误。



例如  $\langle \text{expression} \rangle ::= \langle \text{operator} - 1 \rangle$   
 $| \langle \text{operator} - 1 \rangle \text{AND} \langle \text{expression} \rangle$   
 条件X:  $\text{class}(\langle \text{operator} - 1 \rangle) = \text{class}$   
 $(\langle \text{expression} \rangle) = \text{bool}$   
 $| \langle \text{operator} - 1 \rangle + \langle \text{expression} \rangle$   
 条件Y:  $\text{class}(\langle \text{operator} - 1 \rangle) = \text{class}$   
 $(\langle \text{expression} \rangle) = \text{int}$   
 $| \langle \text{operator} - 1 \rangle // \langle \text{expression} \rangle$   
 条件Z:  $\text{class}(\langle \text{operator} - 1 \rangle) = \text{class}$   
 $(\langle \text{expression} \rangle) = \text{char}$

如果随机选中了定义项  $\langle \text{operator} - 1 \rangle$  AND  $\langle \text{expression} \rangle$  而条件X恰好不满足, 就会出现类型不匹配的现象, 在生成正确程序时, 若条件不满足, 则让它选下一个可选项目尝试。若要产生错误的程序, 则可以人为地改变属性值, 使条件不满足, 则产生的源程序中即会包含此类的语义错误。

基于以上的论述, 可以设计出相应的软件予以实现。

## 五、本系统的实现

在实际设计时选择了chill语言中28条规

则作为一个子集予以实现。首先, 对原有的属性文法加以修改, 使之适合于测试实例生成的需要; 第二, 建立数据文件及设计数据结构存贮AG, 便于查找修改; 第三, 选用Turbo Pascal4.0作为设计语言, 具有调试方便, 可读性好的特点; 第四, 对系统的结构予以模块化并逐步细化。最后, 在IBM—PC机上将本系统调试、运行, 结果达到设计要求, 能自动生成chill源程序, 并正确地给出“预期结果文件”及“所含错误文件”。

## 参 考 文 献

- [1] 周继鹏、顾元祥、华庆一, 编译程序自动测试中的知识表示, 《计算机科学》, 1989.5。
- [2] 顾元祥、李支仁, 属性文法的理论、应用与实现, 《微电子技术 with 计算机》, 1987.12。
- [3] 李支仁, 软件工程与软件质量分析, 电子工业出版社, 1987.10。

(上接第8页)

符串相匹配时按自左而右的顺序进行。因此, INFORMER的查询功能本身就提供了“模糊”查找的功能。但是, 用户在查询过程中要注意, 由这种“模糊”性而引起查询到并非使用者本意要求的记录项, 特别是在以字符型字段作关键字进行数据汇总时, 尤

其要注意, 以免把不需要的数据汇到一起。

另外, 当一个字符型字段的内容为空时, 用SQL语言判定条件“ ”与“ ”具有相同的效果。其结果相当于一个空格后加上通配符“.”。但是, 若字段内容非空时, “ ”与“ ”的意义是不同的, “ ”表示无字符, 而“ ”表示有一个空格字符。

## 参 考 文 献

- [1] 裴广生等, 汉字INFORMIX关系数据库管理系统, 高档微机协会、中国科学院计算技术研究所、地质矿产部北京计算中心, 1985年9月。
- [2] 浪潮系列微机资料XENIX操作系统INFORMIX数据库管理系统, 山东电子设备厂、山东计算机服务公司, 1987年8月。
- [3] 刘日升等, UNIX操作系统原理与应用, 能源出版社。
- [4] 孙玉方等, C语言及其程序设计, 中国科学院计算技术研究所《C语言》编辑部。

# 实时中断通信软件包的设计与实现

武汉二炮指挥学院四大队 张卫兵

## 一、概 述

多数微机通信系统在运行前要通过一定的方法或约定(如电话通知,定时起动)使通信双方同时运行通信程序,进入通信状态后才可开始通信。本实时中断通信系统,不论计算机正在运行何种程序,只要申请通信请求一到,即可中断现行作业,自动进入通信状态。完成通信任务后,返回原程序的断点继续原来的作业。

## 二、硬件结构

同大多数通信系统一样,本通信软件包采用随机设置的异步串行通信接口板。通过RS—232C接口用通信电缆互相连接,形成点对点通信的星形网络结构。本系统的GW—286微机具有4个异步串行通信接口板,对应的RS—232C接口为COM<sub>1</sub>、COM<sub>2</sub>、COM<sub>3</sub>、COM<sub>4</sub>。GW—CH和GW—DH微机均为两个异步串行通信接口板,对应的RS—232C异步接口为COM<sub>1</sub>、COM<sub>2</sub>。具体连接如图1所示。

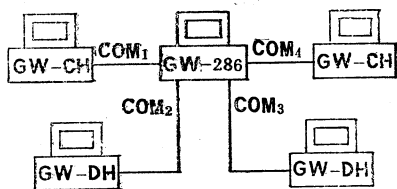


图1 实时中断通信系统的网络结构

RS—232C接口之间的连接是通过一个零调制解调器实现,图2给出了它们之间的具体连接方法。由这种零调制解调器连接的计算机通信的最大距离可达300米,信息传输最高速率为9600波特。

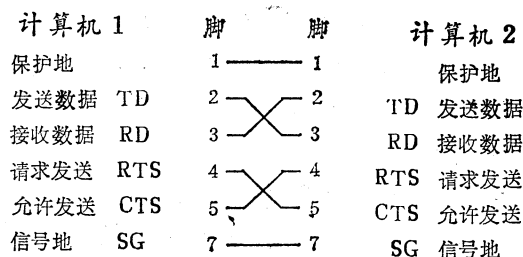


图2 两个RS—232C接口间的连接

## 三、系统的主要功能

### 1. 文件的接收与发送

微机间磁盘文件的转移通过文件的发送与接收完成。本系统实现文件传送有三种方式。

#### ① 传送ASCII码文件

在发送上载和接收下载时产生本地回声校验,发送与接收双方都可以从屏上看到发送以及接收文件的情况。

#### ② 传送非ASCII码文件

用于发送二进制数据文件和其它的非ASCII码文件,也可用于发送ASCII码文件。发送与接收时均不产生本地回声校验。发送与接收双方无法从屏上看到发送和接收文件的情况。

#### ③ 传送XMODEM协议式文件

XMODEM协议式文件的传送支持各种文件的发送与接收,并且可以使用户从支持XMODEM文件协议传送的其它系统上下载文件。

### 2. 键盘对话与快报传递

通信双方通过键盘输入,在自己屏上显示的同时发送到对方微机并在屏上显示,就如同自己的显示器连接在对方的微机上一

样。

### 3. 远地目录的索取

无论对方微机处于什么状态(读写磁盘或与其它联网的微机正在通信除外),与之连接的本地微机均可通过中断方式,在自己的屏上显示对方硬盘的文件目录。

### 4. 用户的程序接口与选单接口

① 对于GW—286和GW—DH微机用户可通过热键“SysReq”激活驻留的通信系统主体。对于GW—CH微机,则由“Shift + Prtsc”激活驻留的通信系统主体。

② 用户也可在程序中通过软中断直接调用INT 15H中断进入通信状态。

### 5. 本站有关的文件操作

① 显示本站微机的硬盘文件目录。

② 显示本站微机的硬盘及软盘的ASCII码文件内容。

③ 打印本站微机的硬盘及软盘的ASCII码文件内容。

### 6. 共享打印机

本系统只实现对网络中有关打印机的控制使用,而不涉及任何有关打印内容以及有关的打印机。除微机对自身所连的打印机是直接使用外,凡申请使用与异地机直接相连的打印机均采用中断方式来申请使用。

### 7. 通信口与通信参数的选择

本系统可以显示当前使用的通信口(对GW—286有COM<sub>1</sub>、COM<sub>2</sub>、COM<sub>3</sub>、COM<sub>4</sub>,对GW—CH和GW—DH只有COM<sub>1</sub>、COM<sub>2</sub>)及通信参数。通信参数包括波特率、数据位数、奇偶校验类型、停止位数。本系统允许用户修改当前使用的通信口及通信参数,而且允许保留修改后的通信口和通信参数使之成为下次通信的缺省值。此后这些参数就一直起作用除非用户重新修改这些参数。

## 四、实时中断通信的原理和方法

### 1. 中断请求信号的引入

IBM PC/XT—AT的RAM中从绝对地址0开始的1k字节里,存放着中断向量表,这就为通信中断请求信号提供了必要的条件。

中断请求信号由异步串行通信接口板ASYN作为外部硬件中断向系统发出。GW-286微机的四个串卡的中断信号对应8259芯片的中断请求线分别是:COM<sub>1</sub>-IRQ<sub>4</sub>、COM<sub>2</sub>-IRQ<sub>3</sub>、COM<sub>3</sub>-IRQ<sub>7</sub>、COM<sub>4</sub>-IRQ<sub>5</sub>。GW-CH和GW—DH两个串卡的中断信号均是:COM<sub>1</sub>-IRQ<sub>4</sub>、COM<sub>2</sub>-IRQ<sub>3</sub>。但是,异步接口板ASYN对应的中断请求线IRQ<sub>3</sub>、IRQ<sub>4</sub>、IRQ<sub>5</sub>、IRQ<sub>7</sub>所产生的中断请求并非时时能向系统发出。因为异步接口板产生的中断在DOS运行后被关闭。有些软件如BASICA、dBASEⅢ系统等对此中断也有处理。如进入BASICA时关闭,而退出dBASEⅢ时也关闭。但事实上这些中断打开后也不影响系统运行和软件的执行。

为了使中断请求信号在任何时刻都能向系统发出,我们采取截取操作系统计时中断并打开通信口的方法来实现。因为ROM BIOS中包含一特殊的伪中断INT 1CH,这一中断在你给它提供子程序之前不起任何作用。启动时中断矢量指向IRET指令。调用INT 1CH中断程序只是简单地返回。INT 1CH的特别之处在于它是在BIOS定时器中断修正日历计数之后由该中断调用的。也就是说这是每秒18.2次自动发生的硬件中断。我们可以改变它的中断矢量使之指向自己编制的对通信口进行检查和处理的子程序段。从而这段子程序就每秒钟自动执行18.2次,就可以防止其它软件将通信口中断关闭。这样就能保证异步串行接口板的中断请求信号IRQ<sub>3</sub>、IRQ<sub>4</sub>、IRQ<sub>5</sub>、IRQ<sub>7</sub>时刻起作用。具体的处理流程如图3。

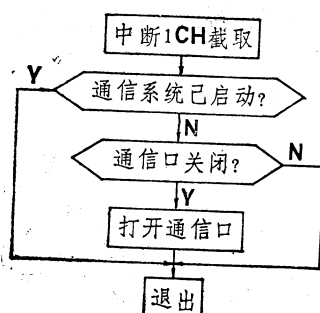


图3 检查通信口的处理流程

## 2. 通信中断的处置

当系统接收到某异步串行接口板的中断请求信号后将自动进入对应通信口的通信中断服务程序，接着系统将做以下处理：

①设置通信系统激活的标志  $CMC_0$ 。 $CMC_0$ 为通信系统已启动的标志， $CMC_0=1$ 表示原先驻留内存的通信程序被激活并调入前台运行。这时有关的计时中断截取及其它几个异步接口板的中断均被关闭，整个系统硬件已被该通信口对应的中断通信程序所控制。

②保护中断现场并且建立新的堆栈。保护现场应该是全面的，因为中断是随机发生的。为了保证完成通信后用户能正确地继续自己的工作，保护现场包括三个方面：寄存器、堆栈区、视频显示窗口以及DOS参数。重新建立通信过程中所需要的堆栈区是显而易见的。

③进入通信状态，具体处理各种不同的通信中断请求。如接收ASCII码文件或接收非ASCII码文件。

④完成通信处理后恢复中断前的机器状态，使用户继续自己的工作。

⑤清  $CMC_0$ ，即  $CMC_0=0$ ，退出中断。

## 3. 中断通信系统的装入

### 与用户的调用接口

在DOS操作系统装入后，即可装入本中断通信系统。在装入过程中主要完成下列工作：

①检查是否已驻留该中断通信系统，若已驻留则不再装入。

②读取缺省的通信参数，并以此对系统的硬件进行初始化。

③对要使用的中断矢量进行截取修改，使之指向我们自己编制的中断服务程序。

④初始化有关标志，如置  $CMC_0$  为0。

⑤最后调用  $INT\ 21H$  的子功能  $31H$  退出，将中断通信系统的主体驻留于内存中，返回到DOS系统下。

通信系统驻留后即可响应其它工作站的通信请求。如果本站用户不主动进行通信，则可随意进行其它工作；若本站用户要主动进行通信，有两种方式供选择。

①在任何时候均可按下“热键”而激活通信系统的主体。GW—286和GW—DH的热键为键盘左下角的“Sys Req”键，GW—CH微机为“Shift + Prtsc”。

②用户可在自己的程序中通过软中断调用  $INT\ 15H$  即可激活通信系统。

## 4. 系统实现中的问题

### ①通信中断发生的时机

因为通信中断请求是随机发生的，当本站正在进行磁盘读写操作时发生了通信中断，中断服务程序就要立即中断现时的磁盘操作，而响应来自异地工作站的通信请求。然而在通信过程中又要进行磁盘操作，这样就势必将原来的磁盘机械状态破坏，从而造成出错甚至于死锁。所以必须禁止在磁盘读写操作时发生通信中断。为此，我们截取了驱动磁盘中断服务程序  $INT\ 13H$  并对其进行修改。方法是进入子程序后，先判断是否在通信程序中的磁盘操作。是，则执行正常的  $INT\ 13H$ ，反之则先关闭DTR使得通信中断请求进不来。然后再执行正常的  $INT\ 13H$ 。这样就保证了在磁盘操作期间不接受其它工作站的通信中断请求。

此外当对方微机正在与其它联网机进行

通信时,本站对该微机的通信中断请求也将无效,直到它们之间的通信结束后,本站的中断请求才有效。并且各从机对主机的中断请求优先级相等,且遵守先到先服务的规定。

### ②DOS系统调用的重入

因为当前的DOS代码是“不可重入”的。也就是说当前台程序正在使用系统调用,常驻程序就不能再发出DOS系统调用。否则将可能破坏系统内部的堆栈而导致系统死锁。但通信中断请求的发生是随机的,所以为了避免当前台正在系统调用时,而其它工作站发出通信中断请求而造成系统死锁。我们在编写通信程序时大量使用ROM BIOS的初级I/O来代替INT 21H中的调用。对于有关文件操作中非使用INT 21H的调用时,也是进行了谨慎的处理,尽量避免死锁的发生。

## 五、系统的主要特点

### 1.通用性

凡是IBM—PC系列微机及兼容机均可使用本系统软件将有关的微机联网使用。

### 2.实用性

采用中断方式进行通信及通信申请的处理工作。当异地机发出中断申请时,只要本站不是正在进行读写磁盘或正处于和其它联网机通信状态,即可中断当前的工作进入通信状态,处理异地机对本机的中断申请。

### 3.联网简便,费用低

微机间联机是将RS—232C接口的有关

引脚直接相互连接而成。因此只需使用与接口相配套的25芯插头,所用的连线是普通计算机通信电缆。

### 4.联网结构灵活

微机联网可选用星形、树形或互连形结构联网,也可以同时采用多种结构联网。本系统采用以GW—286为主机, GW—CH、GW—DH为从机的星形网络结构。

### 5.可靠性

由于网络中各微机均可以不依赖其它微机而独立运行,即使网络中个别微机出了故障,也不会影响整个系统的正常工作。星形结构中主机坏的情况例外。

用上述方法设计的实时中断通信系统全部用8086汇编语言编写,源程序量约138k。现已经过各种测试检验。证明它的实时中断特性是可靠而实用的,基本上达到了设计的要求。

## 参 考 文 献

- [1] Robert Tourdain, Programmer's Problem Solver For The IBM PC, XT/AT.
- [2] L. E. Jordan C. Churchill, Communications And Networking For The IBM PC.
- [3] 夏东涛, MS—DOS高水平程序设计, 电子工业出版社, 1989年。
- [4] 张载鸿, 属部网操作系统DOS高级技术分析, 国防工业出版社, 1988年。

(上接第38页)

③两文件丢失,第一、二根目录已被其他文件占领,且第一根目录的起始族号不为2,则首先必须寻找起始族号为2的根目录,如果在根目录区中找不到,则还得到子目录文件的内容区中去寻找。因子目录在系统中是作为文件来管理的,子目录中的文件

的目录则作为子目录的内容来保存。具体做法可先在根目录区中找到子目录对应的起始族号,然后在该族号对应的文件内容区中去寻找。找到起始族号为2的文件后,把它同第一、二根目录对应文件一同复制后再删除,再用sys命令同样传输系统文件,之后恢复被删除的文件,系统便可完全恢复。

# 解决内存不足的有效途径

杭州电子工业学院 胡维华

## 一、问题的提出

在基于微机的管理信息系统的开发与运行中,经常发生由于内存不足而引起出错。分析其原因,主要是国内大部分微机均采用CCDOS操作系统,该系统采用软字库,字库要占235KB,操作系统本身占40多KB,同时常驻内存的还有一些驱动程序、DOS通讯区等。因此,使用dBASE III (或FOXBASE)就已经很勉强了,如果剩下的内存空间不足200KB,那么dBASE III就不能正常运行。而且在开发与运行管理信息系统时,为了弥补dBASE III (或FOXBASE)的不足,常常想调用其他高级语言程序,如调FORTRAN程序作复杂计算,调BASIC程序绘图或演奏音乐等,这就需要更多的内存空间。

为了解决内存不足的问题,必须压缩系统常驻内存部分的容量。笔者经过分析与实践,提出以下三个有效途径。

## 二、适当设置操作系统参数

操作系统参数的设置是编写一个Config . sys 文件,每次启动DOS, DOS都会从驱动器的根目录寻找该文件,若找到则由Command . com解释并执行。在Config . sys中,有两条命令与内存有关。

Buffers = xx

Files = xx

**1. Buffers命令** 该命令规定DOS在内存中分配的磁盘缓冲器的个数。磁盘缓冲器是DOS用来保存向磁盘读或写数据的存储块,所有缓冲器都包含最近使用过的数据。缓冲器数目的变化范围是1~99,缺席值是2。

DOS拥有的缓冲器越多,则内存存储的数据也越多。对于按随机方式读写的应用程序,如果DOS参与工作的缓冲器越多,则在缓冲器中找到所需记录的可能性就越大,这意味着读取数据的速度可加快。但缓冲器数目太大,可能会使系统速度反而慢下来,这是因为庞大的缓冲器数目,迫使DOS从所有的缓冲器中寻找记录,这可能比它从磁盘读取此记录所化的时间更长。

决定分配多少缓冲器的最终考虑还是计算机的内存大小。一个缓冲器是528字节,缓冲器由DOS管理,每增加一个缓冲器就使DOS驻留内存的容量增加528字节,而相应地,应用程序的可用内存空间就减少528字节。

以下是用Pctools . EXE测试的数据:

当Files = 10时

缓冲器数	用户可用内存空间(字节)
10	208496
11	207968
12	207440
13	206912
14	206384
15	205856

当应用程序可用内存空间过小时,将导致更加频繁地读和写,从而导致系统速度减慢,甚至应用程序或某些操作系统命令无法运行。

大多数教科书与dBASE III手册上,将缓冲器的数目定为24。事实上,当Buffers = 24时,在dBASE III程序中不能以RUN命令



调用 BASIC 程序, 因为进入 dBASE III 状态后, 用 CHSROSK 测试, 内存只剩 80 多 K。

因此, 在开发微机管理信息系统时, 要对实际应用程序以不同的缓冲器数目进行运行时间的测试, 并保证有足够的内存空间调用需调用的其它语言程序, 从而确定一个最佳的缓冲器数目。

**2. Files 命令** Files 命令用来指定 DOS 允许同时打开的文件个数, 缺席值是 8。当应用程序打开一个文件时, DOS 就在其通讯区内构造一个文件控制块 FCB。一个 FCB 占 32 字节, 扩展的 FCB 在 FCB 前面加上 7 字节的前缀, 共占 39 字节。当 DOS 启动时, 即设置好这个区域, 这个区域的大小由 Files 命令指定。因此, 每多一个文件, 这个区域就增大 40 字节, 用户可用内存空间就减少约 40 字节。以下是一组测试数据。

当 Buffers = 10 时

文件数	用户可用内存空间(字节)
15	208288
14	208336
13	208368
12	208416
11	208448
10	208496

dBASE III 手册上 Files = 20, 对具体的系统, 这个数目显然是不定的, 应根据应用程序的需要, 合理地设置 Files 数目, 力争多节省一些字节。

### 三、适当设置 DBMS 参数

除操作系统的设置文件 Config.sys 外, dBASE III 也有自己的设置文件 Config.DB, 该文件与 dBASE III 系统文件放在一起。当启动 dBASE III 时, 将按 Config.DB 的内容设置系统。Config.DB 中与内存有关的命令有两条。

Maxmem = xxx

Mvarsiz = xx

**1. Maxmem 命令** 该命令指定整个 dBASE III 系统将占用的最大内存, 变化范围是 200K~720K。若不加说明, dBASE III 系统将最大占用到 256K 内存。

**2. Mvarsiz 命令** 该命令指定在 dBASE III 命令文件执行期间内存变量将占用的内存最大字节数, 缺席值是 6K, 变化范围是 1~31K。在实际应用中, 由于 dBASE III 系统约定, 非全程变量在遇到 Return 命令时, 所占据的内存空间自动释放, 只要在编程时注意及时释放内存变量, 或使用相同名称的内存变量来覆盖, xx 的值完全可以设置得小一些。

据笔者所知, 一般的应用系统以上两个参数都没有指定, 实际上浪费了内存空间。如遇到内存不够的情况, 建议合理设置这两个参数。

### 四、适当压缩汉字库

如果用户的应用程序特别大, 采用上述方法仍不能满足对内存的要求, 而硬件配置一时不能改善, 要运行 dBASE III 或 FOX-BASE 程序, 还可以在汉字库想办法。

实际上, 一般管理信息系统中使用的汉字大都是一级字库中的汉字, 二级字库(56区到87区)很少使用, 特别是70区以后的汉字更少使用。这部分不常使用的汉字所占用的内存空间就可以省出来给应用程序使用。方法如下。

先用西文 PC-DOS 启动机器, 然后调用动态调试程序 Debug.Com, 删除无用字库的点阵。

```
DEBUG    CCLIB
-R  BX:  0003↵
          0002↵
-W ↵
-Q ↵
```

重新启动汉字系统, 内存可用空间已多出 64K 字节。

# 智能化信息系统中的离散结构

东北财经大学人工智能研究所 陈有刚

**摘要** 本文以离散结构的观点, 讨论智能化信息系统的抽象模型和功能。

## 一、智能化信息系统的离散结构

信息系统是对数据、信息、知识进行获取、表示、存贮、查询、传送和交换的计算机系统, 一般分为数据管理, 信息管理, 知识管理三个阶段, 有的分为信息管理, 办公自动化, 辅助决策三个层次。具有知识库、推理机和知识处理功能, 可以用于辅助决策的信息系统称为智能化信息系统, 这里简称为信息系统。从本质而言, 信息系统从处理对象到处理方法, 从信息结构到系统结构, 都是离散结构。

信息系统的离散结构, 主要包括信息系统中的系统结构, 语言结构, 信息结构, 关系结构, 逻辑结构, 功能结构等。

### 1. 系统结构

系统结构是指组成该系统的各个组成部分及其功能分配和工作机制等。

信息系统的系统结构, 通常由信息库(数据库、事实库、知识库、算法库、模型库等), 推理机, 管理系统, 人一机接口, 计算机通信网等部分组成, 见图1。

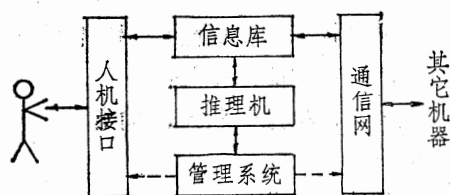


图1 信息系统的结构框图

信息库通常包括数据库、模型库、算

法库、知识库等等。数据是事物和信息的单个属性的数量或特征。模型是事物和过程的抽象, 通常是指数学模型、逻辑模型、结构模型、人工智能模型等。算法是一个问题或一个过程的计算方法或解法, 通常有数值算法、非数值算法、人工智能算法等。知识是规则、常识性知识和专业性知识的总称。信息库相当于计算机的内外存贮器, 由信息库存贮各种信息, 例如数据、事实、知识、规则、算法和模型等。

推理机相当于计算机的运算器, 由推理机进行信息库中信息的删除、增加、修改和变换以及数据的运算, 事实的查询, 知识的推导, 规则的调用, 算法的生成, 模型的建立。

人一机接口相当于计算机的输入输出设备, 它能将人所给出的信息, 例如自然语言、语音、图象、程序等转换为机器语言, 它还能将机器语言转换为人可以接收的信息。

通信网能实现信息在各个系统、各个机器之间接收、发送和传输, 以实现信息的共享。

管理系统相当于计算机的控制器和操作系统。由管理系统对整个信息系统进行管理和控制, 解释并执行人的命令, 使上述各部分组合成一个完整的有机的整体。

### 2. 语言结构

语言结构是指信息系统中所使用的自然语言和程序设计语言的表示、形式和语法结

构。程序设计语言可以是代码语言、汇编语言和高级语言。高级程序设计语言可以是命令型（例如BASIC, PASCAL），函数型（例如LISP），逻辑型（例如PROLOG），数据类型（例如dBASE）等。高级程序设计语言还可以分为块状结构（例如BASIC, FORTRAN），嵌套结构（例如ALGOL, PASCAL），表结构（例如LISP）等。

描述一个语言的语法结构，通常可以使用以Backus表示法所给出的语法公式，语法图等。

### 3. 信息结构

信息结构是指信息的表示和存贮方式。在信息系统中，关于信息的表示，主要有自然语言（或受限的自然语言），命题和谓词，特性表，状态图，语义网络，框架表示法，与/或图，产生式等。关于信息的存贮方式，主要有顺序存贮，分块存贮，拉链存贮，随机存贮等。

### 4. 关系结构

关系结构是指信息间的关系，信息系统中各个组成部分之间的关系。信息间的关系可能有恒等或等价关系，相邻关系，蕴涵关系，转换关系，后继或前趋关系等。信息系统中各个组成部分之间的关系可能是控制和被控制的关系，信息流动和传递的关系等。信息系统网通常有星型结构，环型结构，总线式结构，树型结构，图结构等。

### 5. 逻辑结构

逻辑结构是指信息系统进行处理和推理的逻辑方法和规则等。就硬件设计而言，由与门、或门、非门、触发器等基本逻辑元件组成寄存器、移位寄存器、译码器、计数器、加法器等逻辑部件，进而由各种逻辑部件组成运算器、控制器、存贮器、输入输出

设备。就软件设计而言，由算术表达式、关系式、逻辑值、或、与、非等组成逻辑表达式，根据逻辑表达式判断条件，选择方向，进行推理。

## 二、智能化信息系统的抽象模型

具有离散结构的智能化信息系统的抽象模型是有限自动机和产生式系统。

有限自动机通常定义为

$$M = (Q, V_T, W, f, g)$$

其中  $Q$ ——非空有穷状态字母表，

$V_T$ ——非空有穷输入字母表，

$W$ ——非空有穷输出字母表，

$f$ ——状态转换函数，

$g$ ——输出函数，

而且  $f: Q \times V_T \rightarrow Q, Q^{t+1} = f(Q^t, V_T^t)$

$$g: Q \times V_T \rightarrow W, W^{t+1} = g(Q^t, V_T^t)$$

$f$ 和 $g$ 的意义是： $t$ 时刻的状态和 $t$ 时刻的输入决定 $t+1$ 时刻的状态和 $t+1$ 时刻的输出。根据 $f$ 和 $g$ 是不是单值的全函数，可以将有限自动机分为确定的和非确定的。

由上所述可以看出，不仅整个信息系统可以看作一个有限自动机，每个组成部分可以看作一个有限自动机，而且在词法分析、语法分析、分析和识别程序设计语言和受限的自然语言的句子中也有重要的应用。在使用产生式进行推理时，其实质也是个有限自动机。对于其中每一个非最终状态，都问一个问题，或判断一个条件，或输入一个回答，然后根据条件或输入，分别转换为相应的后继状态，直至到达目标状态为止。

如果将状态转换函数 $f$ 和输出函数 $g$ 简称为功能，可将功能简化为：

输入，状态→状态，输出

再简化为：

条件→动作

前提→结论

输入→输出

这便是一组有序偶，可称之为产生式系统。

### 三、智能化信息系统的功能

智能化信息系统的功能是将输入信息和信息库信息转换为信息库信息和输出信息。

信息库包括数据库、事实库、知识库、文本库、算法库、模型库等。在信息库中，存贮建库时已给的信息，以及以后系统从外界所获得的信息和自身推导出的信息，见图2。

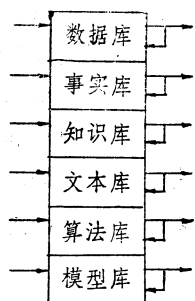


图2 信息库的组成

就信息库的类型而言，其输入可以是新数据、条件事实、前提知识、输入文本、基本算法、原始模型；其输出可以是结果数据、结论事实、推理结果、输出文本、组合算法、应用模型。

信息库的输入和输出，就形式而言，不仅有一般信息系统的基本量（数、逻辑值、

符号、字符串等），而且还有表达式和过程。例如

INPUT X

在一般信息系统中，X只是一个数、一个逻辑值、一个符号、一个字符串。但在智能化信息系统中，X还可以是一个算术表达式、布尔表达式、集合运算表达式，甚至可以是一个程序、一个文本、一组规则等。

例如，在执行INPUT语句时，当机器问“此算法所使用的表达式是什么”时，用户可以键入任意的表达式，如

$$A + B \times \sin(X) - C(5)^{-7}$$

当机器问“在推理时所使用的规则是什么”时，用户可以键入任意的规则集，如

$a, b(1), abc \rightarrow F$

$\sim a, b(2) \rightarrow F$

$\sim a, b(3) \rightarrow F$

$F, r, c \rightarrow G$

这就是说，用户告诉计算机的不是“怎么做”而是“做什么”。这不仅增加了算法和程序的通用性和标准化，而且提高了信息系统的智能化程度。

（上接第12页）

### 四、修改步骤

#### 1. 修改DAO.EXE

C>KENDAO.DAO.EXE DAO

C>DEBUG DAO

-E 35A 11

-E 303 'HZK16'

-W

-Q

C>REN DAO DAO.EXE

#### 2. 修改ZHENG.EXE

C>REN ZHENG.EXE ZHENG

C>DEBUG ZHENG

-E 2A04 00

-E 50E3 'HZK16'

-W

-Q

C>REN ZHENG ZHENG.EXE

至此修改完成，五笔字型汉字系统，可以使用CCDOS 2.13汉字库了。同时使用CCDOS 2.13的打印程序，可以输出多种字形24点阵汉字。

此方法在IBM PC及286、386多种机型上通过。

# 应用软件防错措施的设计

建设银行上海石化专业支行 马 勇

## 一、问题的提出

随着办公自动化、决策科学化的需要，有越来越多的应用软件投入使用，不仅改善了人们的工作环境，降低了工作强度，同时也提高了工作效率。

对于使用者来说，软件只是一个“黑盒子”，他不必了解软件的内部结构及工作过程，他只需在终端上输入自己要处理的各种数据，相应的软件将会按照要求对数据进行加工、处理，最后，用户在终端上将得到经过一系列综合处理后的信息。根据需要，可能是图表、文字，也可能是音响、图象等等。

只有输入了正确的数据，才能期望得到正确的处理结果。所以，我们就应确保输入数据的准确性或叫做可靠性。

确保输入数据的准确，有时会变得很困难，因此如何防止错误的干扰，是软件人员必须高度重视的问题。一方面，要提高计算机操作人员的素质以及对工作的责任感，不断提高其业务技能；另一方面，软件人员也应研究如何使应用软件更安全、更可靠。这其实也是对软件质量的一种评价标准，同时对提高软件在用户心目中的可信程度也有相当大的作用。

## 二、产生错误的原因及分类

在软件的应用过程中所产生的错误，大致可分为三类：

第一类错误是操作员的故意行为。由于种种原因，操作员不希望继续运行这个软件或者是无法运行，但又不按照正确的操作步

骤退出操作，而是以强行中止运行的方法结束操作。比较多见的有按Ctrl-C，Ctrl-Z，Ctrl-Break，Ctrl-Alt-Del键等，甚至中断工作电源重新启动。这样所产生的结果往往是很恶劣的，有时将会严重地破坏数据库文件。因此，这一类错误是一种人为故障。

第二类错误是操作人员的误操作。由于环境、气候、心情等诸多因素的原因，造成操作人员体力和智力上的暂时性幻觉。有时也可能由于操作员长时间连续工作所引起的疲倦。在这种情形中，多数是数据的“位”错误，比如看错了符号、键盘位置的记忆错或者是数据不清晰等等。

第三类错误是由于操作员的业务不精通或工作不认真造成的。诸如数据类型、数据格式、数据内容上的错误，一般都出于此。这类错误是数据的“段”错误。

## 三、防错措施的设计及实现

所谓防错措施，就是利用软件手段对操作人员的错误信息的干扰作出的反应，这种反应的效果是最大限度地降低数据错误对软件及数据库的威胁。

对于不同类型错误，在软件设计上采取的手段与技巧是不相同的。但是，在具体实现时所产生的效果以及设计思想是相同的，即，一是简便操作；二是保护数据；三是保证屏幕信息的完整。

对于第一类错误，一般可采用错误“陷阱”的方法加以控制，利用“前台、后台”操作技巧，使后台处于经常的键盘监测状

态。一旦发现操作人员的非正常操作,便立即作出反应,从而使这种企图不能得逞。

比如,当操作人员按下Ctrl-Break,或按下Ctrl-C, Ctrl-Z,即“中止”键,企图中止操作时,就可以使喇叭发出警报声以提醒操作员。

如果操作人员按下的是Ctrl-Alt-Del,即热启动键,作为对操作人员的错误操作所提出的严重警告,可以在喇叭发警报声响的同时,在显示屏幕上以闪烁的文字形式给出警告。

对于处理重要信息的软件包,在有条件的情况下都应设法加上这一类错误的防错手段,以确保数据库文件的安全。

大多数错误属于第三类。采用的防错措施一般有:

1. 输入数据时,同时给出数据的示范格式和数据内容提示。如,输入日期时,可以要求按月、日、年的格式或年、月、日的格式输入。

2. 接收数据后,按数据要求进行检测,如有不符,则取消此次输入,同时返回到入口处,要求操作员重新输入正确的数据。如,输入日期时,出现了类似13月或2月30日这样的错误等。

3. 对于系统无法确认的数据,则在数据输入后,提示操作员加以确认。

(上接第10页)

入打印文件名并打印该文件,这与上例的方法不同,是交互式的。打印结束后,子进程被撤消,调用进程成为当前进程。

最后说明几点:

- ①程序中不宜过多使用lib\$spawn,否则将导致页故障率增加,处理机工作时间和进程切换时间过长,I/O增多,影响执行速度和系统性能。

- ②用类似方法还可实现系统的其它功能,但需考虑到用户进程的优先数,如启动系统打印队列须有OPER特权。嵌套使用

4. 对于能确定范围的输入数据,一是采取部分地封锁键盘的方法限制数据的按键输入,比如,对于数字型的数据,就可以把键盘的使用仅限制在数码键的范围中,而封锁其他如字符、字母键、功能键、控制键等;二是提供选择性的代码输入,比如,可用代码1、2分别代表男、女性别。

5. 全部数据接收完毕,则提供数据重显和部分修改的机会。

以上五种措施可以单独使用,如果结合音响和色彩的效果综合使用,则效果会更好。第二类错误较难处理。因为很可能操作员在疲倦的状态下“确认”的是一个错误数据,而这种错误是软件所难以发现且无法纠正的,此时,可以采用后期处理的方法。比如,通过打印机打印,输出文字材料供校对复核就是较常用的一种方法。

#### 四、结 论

一个高质量的应用软件,应具备强有力的防错措施以保证各种信息的有效性。

软件人员在设计应用软件时,当然首先要考虑软件在结构上、逻辑上的合理正确,用以保证对正确的数据的处理能得到正确的结果;同时,也要考虑软件应该如何有效地防止错误信息的入侵。只有这样,这个软件才是高质量的,才能得到用户的高度信赖。

时,还需考虑子进程创建限额PRCLM,以免引起调用失败。

- ③调用lib\$spawn,系统的命令语言解释程序(CLI)必须是DCL,否则将产生不正确的结果。

#### 参 考 文 献

- [1] “VAX-11 运行时间库参考手册”,沈阳计算技术研究所
- [2] “GW 220 终端用户手册”,长城微型机开发公司



IBMBIO.COM、IBMDOS.COM

## 丢失后硬盘系统的修复

航空航天部航天质量可靠性信息中心 郭和生

在AST286微机上遇到硬盘不能启动且屏幕无任何提示信息,使用软盘系统却可以启动,且可以转到硬盘,在硬盘中进行操作。

使用 ptools,发现硬盘系统中缺少 IBMBIO.COM、IBMDOS.COM两文件。于是,使用“sys”命令,从A盘往C盘传输系统文件,结果传送失败并显示

“no room for system on destination disk”

分析分区引导程序,它在执行过程中将读入根目录区第一扇,并查看第一个根目录是否为 IBMBIO.COM,第二个根目录是否为 IBMDOS.COM,如果不对,则引导失败并显示

non-system disk or disk error

replace and strike any key when ready

如果正确,则读入IBMBIO.COM并运行之。

由此可见,IBMBIO.COM、IBMDOS.COM 必须占领根目录区的第一、二个根目录位置。

而现在通过 ptools 发现,根目录区的第一、二个根目录被其他文件占用了,而 sys.com 在传输系统时必须将 IBMBIO.COM、IBMDOS.COM传送到第一、二个根目录位置上,所以传输失败。

把第一、二根目录对应的文件先复制后删除,然后用sys命令传输,结果传输成功,之后恢复被删文件。

但是,硬盘启动仍然失败。

再次用 ptools观察了硬盘根目录区,每个根目录的第27、28字节记录了文件存放在硬盘文件内容区的起始族号。刚传送的IBMBIO.COM的起始族号为1ED5,而平常所见到的 IBMBIO.COM的起始族号为2,即整个文件内容区的起始族号,这也是造成硬盘启动失败,且屏幕无任何提示信息的原因。

于是,在根目录区中寻找起始族号为2的根目录,找到后先把它相应的文件复制出来,然后把它和刚传送的两系统文件同时删除,再用sys命令从软盘中传输系统给硬盘,同时恢复刚复制出来的文件,结果,硬盘系统得到完全修复。

由此可见,对于丢失 IBMBIO.COM、IBMDOS.COM的硬盘系统的恢复,必须做到:

①保证恢复后的系统中,根目录区的第一、二根目录为IBMBIO.COM、IBMDOS.COM。

②保证恢复后的 IBMBIO.COM是从文件内容区的首族存放,即第2族开始存放的。

具体修复方法:

①如果两文件丢失,根目录区中第一、二根目录空,则直接用sys命令传输系统。

②两文件丢失,且第一、二根目录已被其他文件占领,但第一根目录的起始族号为2,则可先复制、后删除第一、二根目录对应的文件,然后用 sys 命令从软盘给硬盘传输系统,再复原被删文件。

(下转第30页)

# 24针打印机更换打印针的简单方法

西北机器厂计算机中心 翟明德

在打印过程中,由于种种原因常常使打印机出现断针,致使打印出来的字符少点缺划。特别是使用频率较高的针(如打印制表横线的针)磨损或断针后,就无法打印表格,严重影响正常的打印输出。造成打印机断针的原因很多,如经常打印腊纸,又不及及时清洗打印头,使打印机针头端面与出针孔被干枯的污垢堵塞,严重磨损打印针,越磨越细,最后因打击强度不够而断裂;再就是不及时更换已打印烂的色带,使运动着的色带将正在打击的针头拖断等。若送往维修部门修理,不仅价格昂贵,而且维修周期长,影响打印机的正常使用。

实践证明更换打印针的工作,用户可以自行完成。建议按下列步骤进行。

1. 准备好十字螺丝刀、镊子、放大镜、小锉刀、小刷子和少量清洗剂。

2. 切断电源、取下打印机防尘盖和色带盒。

3. 用螺丝刀拧开紧固针头的两个螺钉,拔出连接针头的电缆插头(请记住插头插入插槽的位置,防止安装错误),轻轻取下针头部件。

4. 将针头部件头部放在清洗剂中浸泡一会儿,然后用小刷子刷掉沾在针头上的油墨污垢,直到清洗干净为止。

5. 拆开针头小后盖,可以看见圆状排列的24根打印针根部,再用小镊子逐个轻压

打印针根部,边压边借助放大镜观看打印头端面(端面有两列针孔,每列12根打印针)。找出断针,在其根部作好标记。

6. 卸开后盖周边的螺钉,打开大后盖,(注意将针头方向朝下,否则打印针会自动从后端抽脱掉落,再装入就很麻烦)。用镊子夹住作过标记的打印针的根部轻轻抽出。

7. 若有新的打印针或可以从已报废的打印头中找出好针来,就直接换入;若手头没有完好的打印针,可用步骤5和步骤6取出打印头顶部或底部任一针来代替。(因为这些针,不常用,不会对打印输出有多大的影响)。用镊子将完好的打印针从打印头后端的断针孔轻轻插入,此时要使打印针穿过四道定位孔槽,这一步要仔细小心,反复多次,直到针头从打印头端面中相应缺针孔中出来为止。

8. 若换过来的新针比其它针稍长一些,可用小锉刀将其锉修到与其它针平齐为止。

9. 按原来位置将打印头装好即可使用。

以上方法是针对3070打印机进行的,其它24针打印机的换针方法类似。我们用以上方法对多台24针打印机进行维修,既经济又方便,效果良好。

# XENIX操作系统存贮管理特点

电子科技大学计算机系 魏 勇

西南技术物理研究所 潘 磊

**摘要** XENIX操作系统是UNIX在80/86系列计算机上的变种,除功能有较大扩充外,与UNIX系统比较相似。但在与硬件相关部分,却同原来基于PDP-11、68000系列的UNIX操作系统有较大区别。本文在扼要介绍80286/386存贮管理体系结构的基础上,通过操作系统的分析、阐明XENIX存贮管理的特点。

## 一、80286/386的存贮管理体系

### 1. 逻辑地址与物理地址映射

80286/386的物理地址就是存贮管理硬件发送给存贮器的地址。逻辑地址是由指令根据寻址方式产生的地址。80286/386的存贮管理是通过硬件动态地进行逻辑地址和物理地址之间的转换的。每访问一次内存,都要把CPU所发出的逻辑地址转换成物理地址。当一条指令在物理存贮空间找不到所需的虚拟地址,就要通知操作系统,把所需的内容从辅助存贮器调到物理存贮器。用户是在逻辑地址空间(也称虚拟地址空间)进行程序设计的。

### 2. 80286/386分段存贮的实现

80286/386采用说明存贮状况的描述符来实现虚拟存贮管理。在描述符表中有RAM内当前所有各段的有关信息的拷贝。每个存贮段都配备一个描述符,描述符中包含物理存贮器的基地址。用户以选择符的方式进行编程,当系统把一个选择符装入段寄存器,它就自动保存旧的描述符而把新的描述符从存贮器装入80286/386的CPU。

以80286为例,它把虚拟地址分成两个16位的段。其中前半为段选择符,用来指定使用的段,后半是段内偏移量。这样就允许对64k字节地址空间中的任何一个字节

进行访问。它的虚拟地址格式如图1示。

15	2	1	0	15	0
变址		T	RPL	段内偏移量	

图1 80286虚拟地址格式

各字段的意义:

变址——由T位指定被选中描述符表中段描述符的变址。

T——表指示符,说明是全局描述符还是局部地址空间。

RPL——所请求的特权级。

偏移量——指定段内的偏移量。

80286的描述符是一个8字节存贮表,

见图2。它是从虚拟地址空间向实地址空间

7	0 7 0						
+7	Intel 保留						+6
+5	P	DPL	S	TYPE	A	基地址23...16	+4
+3	基地址15... 0						+2
+1	界 限15... 0						0

图2 80286描述符格式

转换的地址转换表的集合,也包含实现虚拟地址空间和存贮保护的一些信息。它的各个字节都具有特定的意义,除有段的基地址以外,还包括有如段长界限、访问权(只读、读/写、仅执行和执行/读)、虚拟存贮器是否存在、描述符特权级等。任何违反描述符规定的段标志属性的使用,都将停止存

贮周期的执行，并产生异常或中断信息。

流程图见图6。

## 二、XENIX操作系统 对描述符的操作

由于80286/386用描述符来反映某段存储空间，因而在XENIX操作系统中，对描述符中各字段的读、写操作是必不可少的。

### 1. 描述符各字段的读出程序

在XENIX操作系统中，对界限，地址低位、地址高位以及访问权字段的访问是分别通过子程序dscrlimit、dscraddr、dscracc和dscrsw来实现的。分别说明如下：

dscrlimit

功能：在AL中返回描述符的界限字段。

流程图见图3。

dscraddr

功能：在AX中返回描述符的地址低位字。

流程图见图4。

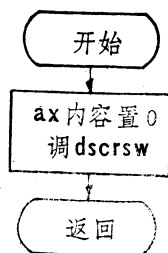


图3 dscrlimit流程

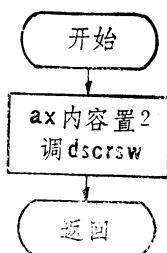


图4 dscraddr流程

dscracc

功能：在AL中返回描述符的地址高位字节字段。

流程图见图5。



图5 dscracc流程

dscrsw

功能：在AL中返回描述符的访问数字段。

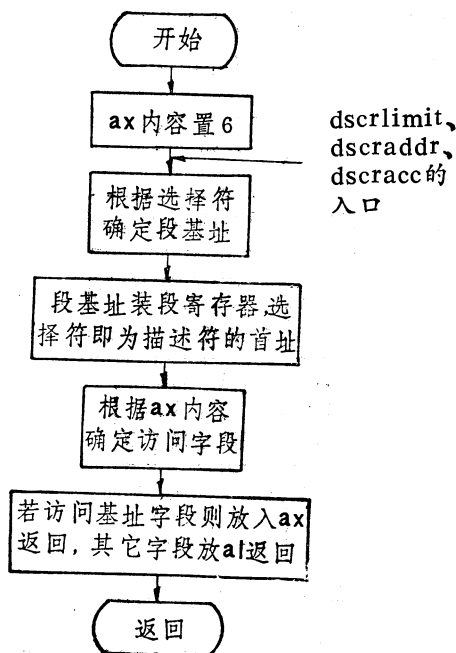


图6 dscrsw流程

参数说明：该子程序仅一个参数，它是所访问描述符的选择符。

在XENIX操作系统中，Proc结构新增添了P.fcse1、P.fdsel、P.ltsel成员，分别表示第一个代码段描述符的选择符、第一个数据段描述符和最后一个数据段描述符的选择符。所以可以根据这些参数，找到进程某段存储空间的描述符，并根据要求读出该描述符的某一字段。

### 2. 描述符字段的写入

根据某进程的Proc结构，找到所要写入的描述符，由子程序mmudescr完成。

mmudescr

功能：实现对某描述符各字段的写入。

流程图见图7。

参数说明：

参数1——描述符的选择符。

参数2——将写入的地址低位字段。

参数 3 —— 将写入的基地址高位字段。  
 参数 4 —— 将写入的访问权字段。  
 参数 5 —— 将写入的界限字段。

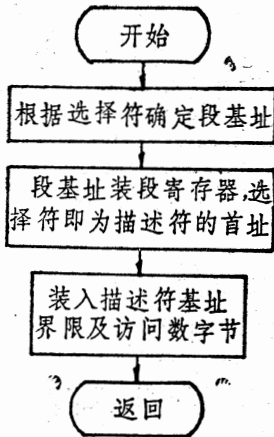


图 7 mmudescr 流程

### 三、XENIX 操作系统进程 映象存贮申请与释放

#### 1. 存贮资源管理

XENIX 对内存或盘交换区两部分存贮资源的分配和释放使用的算法同 UNIX V6 类似。它们分别由 malloc 和 mfree 子程序实现。为了便于管理, XENIX 也设置了内存可用存贮区表和盘交换区可用存贮区表, 这与 UNIX V6 略有区别。它们分别是:

Struct map coremap [CMAPSIZ]

Struct map swapmap [SMAPSIZ]

数组中, 每个分量又由两个元素构成的表目项组成, 表目项的数据如下:

struct map

```

{
    short m—size;
    unsigned short m—addr;
};
    
```

m—size 是相应空闲存贮区的长度, 其单位对内存而言是内存字符块, 对盘交换区

则外存字符块, m—addr 是相应空闲区起始地址。

#### 2. 进程映象内存的申请与释放

进程映象内存申请与释放分别由子程序 mmupget 和 mmupfree 实现。系统对存贮空间是按段式管理的, 对每一段内存的申请与释放是由子程序 mmusegget 和 mmusegfree 实现的。而各段内的存贮申请与释放又通过 mmuget 和 mmufree 分别调用 malloc 和 mfree 来实现的, mmuget 和 mmufree 子程序很简单, 直接给出内存字符块大小和 mp 指针后就调用 malloc 和 mfree。

mmupget

功能: 进程映象内存的申请。

流程图见图 8。

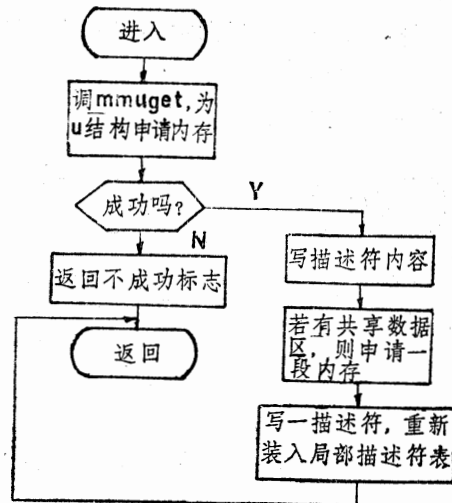


图 8 mmupget 流程

说明: 程序中, 首先要为 u 结构申请 2k 的存贮区。如果有共享数据区, 除了为本进程申请内存外, 还要为共享数据区申请内存。若申请成功, 立即填写描述符, 并装入局部描述表中。

mmupfree

功能: 进程映象内存的释放。

流程图见图 9。

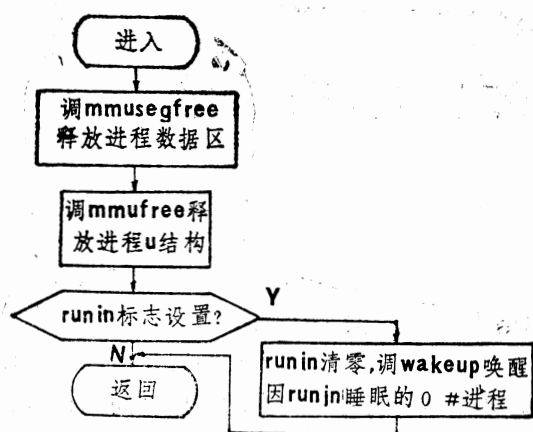


图9 mmupfree流程

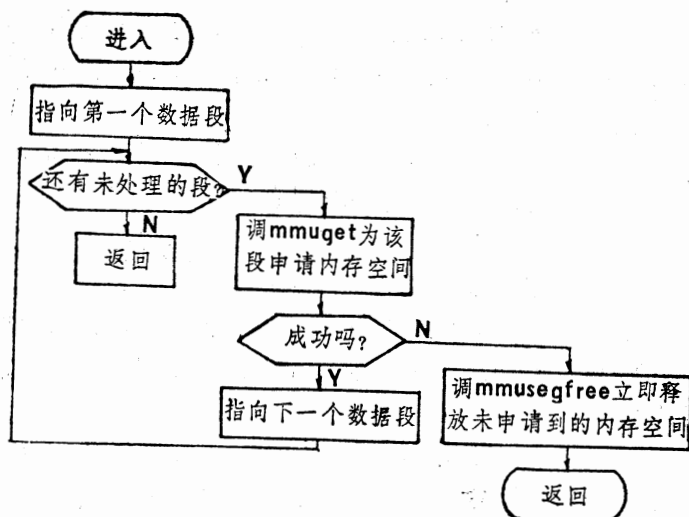


图10 mmusegget流程

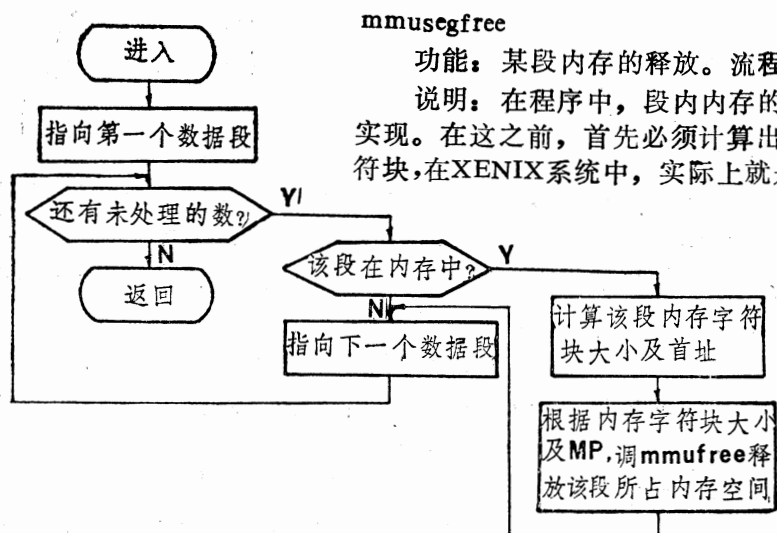


图11 mmusegfree流程

说明: 程序中 runin 标志与进程图象在内存和盘交换区之间的传送有关, 如若 runin 标志已经设置则表示在盘交换区上有准备就绪进程图象需要调入内存, 但是在内存中无足够大的空闲区, 而且也设有合适的进程可以调出。当一个进程进入低优先权睡眠状态后, 因为要等待的事件一般进行得比较缓慢, 因此可以将其图象送到盘交换区上。于是, 就可以唤醒因runin而睡眠的 0 # 进程, 使它再次试图进行进程图象在内存和盘交换区之间的传送操作。

#### mmusegget

功能: 某段内存的申请。

流程图见图10。

说明: 根据该段的大小和 mp, 调mmuget 申请一块内存, 若不成功, 必须立即调用 mmusegfree 释放这块未申请成功的内存区。

#### mmusegfree

功能: 某段内存的释放。流程图见图11。

说明: 在程序中, 段内内存的释放具体通过 mmusegfree 来实现。在这之前, 首先必须计算出该段的大小, 单位是内存字符块, 在XENIX系统中, 实际上就是512字节。

#### 参考文献

- [1] 尤晋元, UNIX 操作系统教程, 西北电讯工程学院出版社, 1985。
- [2] 艾德才, 长城286/386体系结构与汇编程序设计, 天津大学出版社。
- [3] 雷英杰, XENIX内核的破译与分析,《微型计算机》, 1988年, 8卷3期。



# MCS-51与MCS-96之间数据传输

西安邮电学院 李令戈

## 一、前言

随着微电子技术的发展,近年来出现了各种不同类型的单片微型计算机(简称单片机)。由于单片机具有体积小、功耗低、功能强和价格便宜,稍加一定的外围设备,就可方便地构成一个应用系统。因而在分布式控制系统中被广泛采用。

Intel公司生产的MCS系列产品是目前国内应用比较广泛的产品。本文介绍单片机MCS-51与MCS-96之间实现异步通信的方案及编程要求。

## 二、串行口简介

MCS-51系列采用的是8031单片机, MCS-96是8096单片机, 8031和8096的串行口均为全双工数据通信通道, 都能处理异步通信协议。

8031串行口具有接收、发送两个数据缓冲器, 并带有两个控制寄存器SCON和PCON。

PCON是数据传输波特率选择寄存器, 用来确定串行数据传送时的波特率, 但它无位寻址功能, 编程时不须进行位操作。

SCON是串行I/O口控制寄存器。该寄存器设有位地址, 编程时须进行位操作。各位分别用来确定串行口工作方式、是否允许串行接收、发送、接收中断标志等参数。

8031单片机编程要点:

①确定数据传送速率。通过对16位的可编程定时器/计数器编程来确定波特率。

②确定工作方式。对SCON中SM0、SM1工作方式选择位设置参数。

③确定是否采用中断工作方式、对SCON中TI、RI位设置参数。

8096串行口有4个控制寄存器。这些寄存器是串行口操作与驱动程序之间的接口。CPU通过向IOC1、BAUD-RATE、SP-CON寄存器写入控制信息, 有效地控制串行口电路的操作过程。此外, CPU通过读取SP-STAT寄存器的状态信息, 及时地了解通道的工作情况。

IOC1是I/O控制寄存器, 它可以控制多功能P2口的各引脚功能。

BAUD-RATE是16位寄存器, 它的作用是确定串行通信的波特率。

SP-CON是控制寄存器, 它被用于确定工作方式; 设置发送、接收中断标志; 允许接收及奇偶校验。

当串行口采用中断方式进行数据传输时, 还需对中断寄存器INT-MASK、INT-PENDING设置参数。

8096单片机编程要点:

①确定数据传送波特率。即设置BAUD-RATE寄存器参数。

②设置多功能P2口的引脚功能。置TXD/P2.0和RXD/P2.1引脚为TXD和RXD端, 即设置IOC1.5 = 1和REN = 1。

③确定串行口工作方式。设置SP-CON中M<sub>1</sub>、M<sub>0</sub>的方式字。

④确定是否开中断工作方式。对SP-CON中RI、TI位设置参数。

## 三、程序设计

发送、接收程序流程图见图1。MCS-

51机的发送程序见图2；接收程序见图3。8096机的发送程序见图4；接收程序见图5。

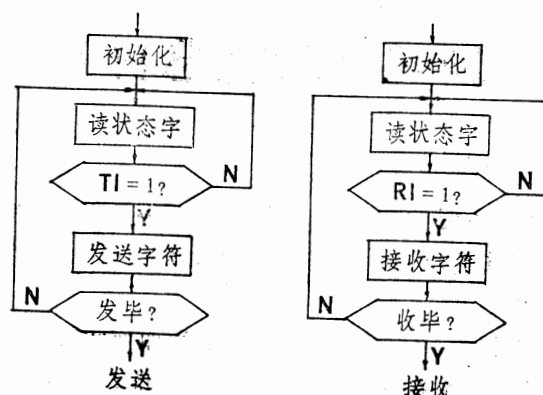


图1 程序流程图

```

ORG 1000
MOV TMOD #20H
MOV TH1 #0F3H
MOV TH1 #0F3H
SETB EA
MOV IEC #90H
MOV SCON #40H
MOV R0 #XXH
MOV R1 #64H
LOOP MOV A @R0
MOV SBUF A
TAN JNB TI TAN
CLR TI
DEC R0
DJNZ R1 LOOP

```

图2 MCS—51机发送程序

```

ORG 1000
MOV TMOD #20H
MOV TH1 #0F3H
MOV TH1 #0F3H
SETB EA
SETB ES
MOV SCON #50H
MOV R0 #XXH
MOV R1 #64H
REVE AJMP REVE
MOV A SBUF

```

```

MOV @R0 A
CLR R1
DEC R0
DJNZ R1 REVE

```

图3 MCS—51机接收程序

```

SP EQU 18H SBUF EQU 07H
SPCON EQU 11H A EQU 66H
SPSTAT EQU 11H IOC1 EQU 16H
REG1 EQU 90H REG2 EQU 80H
ORG 4200
LD SP #0C0H
ORB IOC1 #20H
LDB 0EH #XXH
LDB 0EH #XXH
LDB A #20H
LDB SPCON #09H
LD REG1 #1200H
LD REG2 #64H
TANS ORB A SPSTAT
JBC A 5 TANS
LDB SBUF [REG1]
ANDB A #0DFH
INC REG1
DEC REG2
CMP REG2 #0H
JNE TANS
END

```

图4 8096机发送程序  
(下转第48页)

# 1990年分类总目次

## 人 工 智 能

- 一种抽油机井故障智能诊断系统  
王中秋(黑龙江电脑应用技术公司) N01,1
- 喷漆机器人关节精确  
——模糊集成智能控制研究  
王昌银 王应建 林建亚 路甬祥  
(浙江大学流体传动及控制研究所) N02,1
- 智能化信息系统中的算法库设计  
陈有刚(东北财经大学信息系) N02,12
- 面向C语言的专家系统程序设计  
吴朝晖 应 晶 何志钧 俞瑞钊  
(浙江大学人工智能研究所) N04,15

## CAD、CAM

- Auto CAD与高级语言的联结  
茹志富(杭州电子工业学院) N01,13
- Auto CAD与FORTRAN接口  
——DWG文件型接口  
朱水令(浙江冶金设计院) N04,23

## 智 能 仪 器

- MDC—1多功能数据通信器  
张厥标 顾天行 劳丽群  
(浙江省计算技术研究所) N04, 5

## 计 算 机 网 络

- 高速光环网络  
——应用于灵活的多重服务一体化  
徐 加编译(浙江省财政厅) N01,40
- CSMA/CD总线令牌总线令牌环局网的互连  
李腊元(武汉水运工程学院) N02,28
- 本地网公共传输介质动态时分多路化协议  
陈恭璋(核工业天津理化工程研究院)  
N04,39

## 数 据 通 信

- IBM-PC/XT和单片机数据通信系统的设计  
谢西就(中科院广州电子技术研究所)  
N04, 1
- C语言如何实现串行通信  
赵 伟(大连海运学院电子系) N04,19

## 检 测 与 控 制

- 半连续碳化硅烧结炉微机控制系统  
褚士连 屠晓燕 曹巧英  
(嘉兴市计算机应用研究所) N02, 4
- 可编程实时多任务集散型控制系统  
张林平(浙江省计算技术研究所) N03,11
- 洗衣机流水线可编程自动控制系统  
林方靖(浙江省计算技术研究所)  
陈流潭(杭州洗衣机总厂) N03,15
- 产生式系统的并行处理  
郑德玲(北京科技大学) N04, 8

## 图 形 显 示 与 处 理

- 直接在操作系统下设置彩色屏幕的方法  
张 红(湖北省公路局科研所) N01,10
- 一种用于胎儿心电QRS波识别的属性文法  
黄 政 熊 新  
(浙江省计算技术研究所) N01,18
- 屏幕图形拷贝存贮和显示  
罗国明  
(浙江大学CAD/CAM研究中心) N03,21

## 操 作 系 统

- XENIX操作系统的安全保密性  
郑启心(农业银行武汉市分行) N02,36
- 面向综合化的UNIX-SUN OS 4.0的特点  
秦学礼(衢州化学工业公司) N03,30

## 计算机语言与数据库系统

### 4 GL DataFlex - DOS数据库语言的特点

沈嘉麟(浙江省计算技术研究所) No2,15  
提高dBASE III运行速度的有效方法

王钱华(临安县科委计算机站) No2,22  
一种更好的C——C的衍生体在兼容性和可移植性上超过了母体

龚欣译(成都军区司令部计算站) No2,45  
一个以K-NET网络为基础的  
分布式数据库KNDDBS系统

刘冬喜(成都机车车辆工厂) No3,1  
关系型数据库管理系统选型分析

徐跃强 王光明  
(杭州商学院管理信息系) No4,30

## 计算机管理预测和辅助决策

### 渔业管理预测决策计算机辅助系统

赵丽仙 徐金林  
(浙江省计算技术研究所) No1,3

### 临安县政府微机管理信息系统

黄文龙(临安县科委计算机站) No1,7  
数量遗传和计算机在畜禽育种方面的应用

骆权(浙江省计算技术研究所) No2,8  
物资帐务微机辅助管理系统

付国华 黄乃扬  
(浙江省计算技术研究所) No2,10

### 一个宏观产品质量信息管理系统

宋如顺 王必友(南京师范大学)  
舒平(江苏省计经委质量处) No3,4

### 集装箱动态跟踪管理系统

苏建岛(山东外运公司计算机室) No3,7  
用灰色系统预测模型预测

### 浙江省粮食产量和消费量

周洪祥 郭佳南  
(浙江省计算技术研究所) No3,47

计算机在辅助厂内银行管理及决策中的应用  
朱沈华 王培永(浙江省计算所)

蔡建华 潘伟宏(湖州永昌丝织厂) No4,11

## 程序设计

### PC BASIC程序的自生成技术

顾景文(同济大学) No1,21

### 对PASCAL动态内存管理的改进

潘凌云(杭州大学计算机系) No1,24

### 专家系统程序设计环境

汤庸(长沙铁道学院电子工程系) No1,32  
24×24点阵通用高级汉字打印驱动软件

王磊 刘凤荣(郑州高炮学院) No2,18  
一种自动确定模块覆盖结构的方法

潘卫东(江苏广播电视大学) No3,26  
IBM硬盘失踪加密法分析与实现

陈恭璋(核工业天津理化工程研究院) No3,33

### dBASE III宏代换应用技巧

章寿发(嘉兴民丰造纸厂) No3,38

### 提高dBASE数据库建库时的数据录入速度

高清泉(济南石化经济学校) No3,41

### 在XENIX系统下复制DOS环境下

### dBASE III文件应注意的问题

刘曙光(浙江省电力试验研究所) No4,34

## 故障诊断

### 模板的调试及故障诊断

翁玉贞(杭州自动化研究所) No3,18

### 软件测试的进展和国内外现状

廖焕然(浙江省计算技术研究所) No4,35

## 计算机发展动态

### 80486和68040的基本特性

庄国选(浙江省计算技术研究所) No2,39

## 维修经验

### 3070打印机断针的应急措施

吴达(无锡市港务管理处) No1,26

### 软盘驱动器及软盘的使用和维修

燕军(西南石油学院) No2,25

### APPLE II机故障维修十例

徐有刚(核工业九院四所) No2,26

## 键盘引起字符错乱故障及排除方法

胡建人(杭州电子工业学院) No3,46

## IBM PC/XT微机故障检修

王峰(烟台师范学院) No4,46

## 试谈TM100软磁盘驱动器的故障

## ——零磁道定位开关引起的读/写故障处理

钱希承 马晓琪

(中科院近代物理研究所) No4,47

## 产 品 介 绍

## ZJ-STD集散式工业控制机

梁加权 朱宁

(浙江省计算技术研究所) No1,28

## ZJDG-2打印机共享器介绍

郁德金(浙江省计算技术研究所) No1,封底

## SPC-II共享打印机转接器介绍

黄瑞国(河北机电学院电子工程系)

No2,封底

## 其 它

## 如何选购小型或微型计算机

陈国翠(杭州金融管理干部学院) No1,35

## 1989年分类目录索引

编辑部 No1,47

## 《汉字dBASEⅢ及其程序设计技巧》书再

## 次发行

浙江省计算机学会办公室 No1,封三

## 九宫图之算法研究

贡可荣(海军工程学院) No2,41

## 简讯 期刊盛会、智能型的SPC-II、

## 价廉实用的ZJDG-2

特约通讯员 No3,封底

## 答读者问

编辑部 No4,封三

## 征稿启事

编辑部 No4,封底

(上接第45页)

```

SP      EQU 18H   SBUF EQU 07H
SPCON   EQU 11H   A      EQU 66H
SPSTAT  EQU 11H   IOC1   EQU 16H
REG1    EQU 90H   REG2   EQU 80H
        ORG 4300
        LD SP     #0C0H
        ORB IOC1  #20H
        LDB 0EH   #XXH
        LDB 0EH   #XXH
        LDB A     #20H
        LDB SPCON #09H
        LD  REG1  #1200H
    
```

```

        LD  REG2  #64H
REVE    ORB  A     SPSTAT
        JBC  A     5 REVE
        LDB [REG1] SBUF
        ANDB A     #0BFH
        INC  REG1
        DEC  REG2
        CMP  REG2  #0H
        JNE  REVE
        END
    
```

图5 8096机接收程序