

# 电脑

(月刊)

1994 年第 7 期

总第 73 期

主 办: 电子工业部中国软件行业协会

编 辑: 《电脑》编辑部

出 版: 电脑杂志社

地 址: 广州市石牌华南师范大学内

邮政编码: 510631

电 话: (020) 5514304

总发行处: 韶关市邮电局

国外发行: 中国国际图书贸易总公司

(北京 399 信箱 邮政编码: 100044)

国外发行代号: M4190

印 刷: 韶关曲江印刷厂

定 阅 处: 全国各地邮电局、所

定 价: 3.00 元

出版日期: 1994 年 7 月 15 日

刊 号: ISSN1002-9613  
CN44-1188TP

邮发代号: 46-115

广告经营许可证: 粤工商广字 01090 号

主 编: 吴 军

副主编: 林 林

编 辑: 卢成彪 陈卫康

# 奖

## 有奖征名

为了把《电脑》办得更生动活泼, 内容既充实又具现代气息, 《电脑》今年开展栏目有奖征名活动, 对现有的栏目, 你有更贴切更醒目的名称取代, 请来信。一经采用, 奖励 50 元/个, 并在杂志上公布。

### ♥ 多媒体

2\ 声霸卡后台语音播放功能的实现

5\ 计算机软件技术的现状与发展趋势

### ♥ 电脑与法律

7\ 积极推动软件保护, 促进广东经济、技术的发展

### ♥ 应用与发展

10\ 面向产品的工程数据库管理模式探讨

13\ 新制度财务会计计算机管理系统的设计

### ♥ 软件纵横

16\ 介绍一种功能强大的软件——WORKS

17\ 实时接收程序的设计方法

### ♥ 网络与通信

21\ NetWare v3.11→NetWare SFT II 3.11

23\ 高级 UNIX 连网技术讲座

第六讲 远程文件系统 RFS (三)

27\ IBM-PC 机的异步通讯

### ♥ 使用与维修

29\ M-1724 打印机常见故障排除三例

30\ AR-3240 打印机断针的修复

### ♥ 新天地

31\ 妙笔生花——蒙恬中英文手写辨识系统

### ♥ 奇思妙想

33\ 上下文相关联机帮助机制在 Foxpro 上的实现

36\ C 语言程序对成批数据库文件的加密和解密

37\ 在 Windows 环境中运行汉字系统

38\ 2.13H 五笔字型内部词汇的查阅

### ♥ 中文信息处理

39\ 基于 WINDOWS 的中文输入

41\ WPS 打印蜡纸一法

42\ WPS 非正常退出遗失文件的找回

### ♥ ABC

43\ 通用菜单信息控制技术的设计与实现

44\ 如何有选择地学习汉字输入法

45\ 几种拷贝工具的比较

46\ 浅谈 C 程序中对 Ctrl-Break 的处理

### ♥ 简讯

47\ 中国软件行业协会新举措 《世界软件总汇》编辑出版工作正式启动

### ♥ 游戏乐园

48\ 谈谈“Syndicate (极道枭雄)”

48\ 巧改《快打至尊》人物谱

49\ 凯兰迪亚传奇详解

### ♥ 病毒防治

51\ INTEL 网络疫苗撒下企业网络防毒网

52\ INOC 病毒的解密与消除

## ♥ 软件廊

55\ 中外软件廊软件报价

## ♥ 摩卡

57\ 功能强大的 VESA Local Bus 三合一显示卡

## ♥ 工控天地

59\ 单片机与模糊控制讲座

第十讲 用 MC6805R3 控制的模糊洗衣机(下)

## ♥ 说长道短

61\ 《巧用 FoxBASE 中的宏命令》一文中存在的漏洞

## ♥ 用户园地

62\ 图形用户接口中图标(ICON)的设计

64\ FoxBASE+ 动态索引技术的应用

65\ EMS 与 C 的文件型接口系统

66\ 实现图形局部放大的简单有效方法

68\ 在中国龙汉字系统中使用九针打印机

71\ 直接读写 EGA/VGA 卡 VRAM 区显示汉字

72\ 实用文件分割存储程序

74\ TVGA 1024×768 模式下屏幕换段的实现

75\ 背景音乐

76\ 空心汉字的显示

## ♥ 新辞典

77\ 新辞典

78\ 广告索引

## CONTENTS

5\ On the situation and trend of software

7\ Promote software protection to help the progress of  
Guangdong economy

10\ Product oriented engineering database

13\ A MIS for China new financial system

16\ WORKS — a powerful software

17\ A way for making a real time input program

21\ Netware v3.11 to Netware SFT Ⅲ 3.11

23\ Advanced UNIX networking techniques (6) RFS

27\ Asynchronous communication in IBM-PC

31\ A Chinese/English hand writing recognition system —  
PenPower

33\ Context-sensitive on-line helping based on Foxpro

36\ Data base files encipher and decipher in C

37\ Chinese system running based upon Windows

45\ A comparison among severel copy-tools

47\ Decipher and elimination of viruse INOC

62\ The icon design of graphical user interface

64\ Dynamic index in FoxBASE+

65\ File type interfaces in EMS and C

68\ using 9 pins printer in Software China Dragon

72\ File partition stored programs

# 中外软件廊

——国内首家正版软件交易市场

国内最大的正版软件集散地

需要找软件客户请与我们联系

需要找软件开发请与我们联系

需要推广原版软件请与我们联系

需要购买原版软件请与我们联系

地址:广州市流花路 119 号锦汉大厦 11 楼

广州国际电脑电子博览中心内

电话:6689457、6689452、6689447、6689456

邮政编码:510015

# 声霸卡后台语音播放功能的实现

四川大学工程技术学院 陈星

【摘要】本文通过实例,说明了实现声霸卡后台放音的方法,并给出了具体的实现步骤。用户可根据其原理,用于自己的应用程序中,使自己的应用程序变得声色并茂,更加引人入胜。

自 1989 年秋季,新加坡 Creative 公司在美国 COMDEX 展示会上推出了震撼计算机界的 Sound Blaster Card (声霸卡)之后。由于其性能卓越,因此在短短时间,即以压倒性优势占领了整个语音介面卡的市场,至今应用愈来愈广泛。但随卡配备的手册及软盘上都仅给出了声霸卡前台语音播放的范例。而在实际应用中,常常要求实现后台语音播放的功能。

前台语音播放因语音播放过程是相对独立的,故可编成一个通用程序,亦可进一步编译 EXE 文件,用户程序可直接调用实现语音的播放。而后台语音播放无法编成一个 EXE 文件序,因为一旦被调用完毕,随着放音程序在内存中退出,语音播放会自动停止。故只能以函数或子程序形式,让用户的应用程序调用。同时,除首次调用外,在每次调用后台放音程序时,必须先终止前次的语音播放,并关闭前次打开的语音文件。

实现后台语音播放的一般步骤如图 1 所示:

以上涉及的各类语音函数详见声霸卡的手册。这里给出一个实现后台放音的实例。使用的声霸卡为 SB Pro 版,采用 TURBO C 作为编程语言。这里仅以此例说明实现后台语音播放功能的方法和步骤,用户可根据自己应用程序的需要稍加修改,也可采用其它为声霸卡允许的编程语言进行编程。

程序清单如下:

```
/*-----*/
/*@@ Source Documentation      ** TC Version *** */
/* DESCRIPTION:
/* 程序在屏幕上依次从 0 开始显示数字,每隔 1 秒钟,
/* 该数字加 1 并显示。
/* 同时用后台放音的方式将磁盘上的语音文件 F1.VOC 调入内存并播放声音。
/* 放音 20 秒后,结束 F1.VOC 的放音,开始 F2.VOC 的放音,20 秒后结束本程序。
/* ENTRY:
/* None
/*-----*/
```

```
/* EXIT:
/* None
/*-----*/
#include <dos.h>
#include <bios.h>
#include <stdio.h>
#include <fcntl.h>
#include <sbcb.h>
#include <sbvoice.h>
#include <loaddrv.c>

int Handle; /* 语音文件指针 */
int voice-num=0; /* 放音次数变量 */
extern char far *near ctvdsk-driv;
main()
{
    int t=0; /* 存放数字的变量 */

    voice-out-background("f1.voc"); /* f1.voc 的后台放音 */
    /* 从数字 0 开始,每隔 1 秒钟数字加 1 并显示 */
    while(t<20){
        printf("%d",t); /* 显示数字 */ sleep(1); /* 延时 1 秒 */
        t++;
    }
    voice-out-background("f2.voc"); /* f2.voc 的放音 */
    /* 从数字 20 开始,每隔 1 秒钟数字加 1 并显示 */
    while(t<40){
        printf("%d",t);
        sleep(1);
        t++;
    }
}

/*-----*/
/* voice-out-background(char *voice-name)
/* DESCRIPTION:
/* 对声霸卡进行检测和初始化,并用后台放音方式输出文件名为
voice-name
/* 的语音文件。
/*
/* ENTRY:
/* voice-name: filename to be output.
/* EXIT:
/* None.
/*-----*/
voice-out-background(char *voice-name)
{
    /* 判断是否是第一次放音,若前面已有过放音,则结束上次放音 */
    if(voice-num>0){
        -close(Handle);
```

**正大数据修复**

地址:五山路华附商铺 22 号(市团校对面)  
电话:(020)7570626, 7570627, 5515961-5931

```

ctvd—terminate();
}
/* 对声霸卡进行各种检测及初始化设置 */
if (!GetEnvSetting()) /* 获得 BLASTER 环境变量 */
{
    if (sbc—check—card() & 4)
        /* 侦测声霸卡及设定基本 I/O 地址 */
    {
        if (sbc—test—int()) /* 侦测声霸卡中断设定 */
        {
            if (sbc—test—dma() >= 0) /* 测试声霸卡 DMA 的设定 */
            {
                /* 如果为首次放音,则载入 CTVDISK 驱动程序到内存缓冲区 */
                if (voice—num == 0) {
                    ctvdsk—drv = LoadDriver("CTVDISK.DRV");
                }
                if (ctvdsk—drv != 0)
                {
                    if (!ctvd—init(16)) /* 初始化驱动程序 */
                    {
                        OutputFile(voice—name); /* 播放语音 */
                    }
                    else
                        ShowError(); /* 显示错误代码 */
                } else {
                    printf("driver load error\n");
                    exit(-1);
                }
            }
            else
                printf("Error on DMA channel.\n");
        }
        else
            printf("Error on interrupt.\n");
    }
    else
        printf("Sound Blaster Card not found or wrong I/O settings.\n");
}
else
    printf("BLASTER environment not set or incomplete or invalid.\n");
return(0);
}
/* ----- */
/* OutputFile (char *szFilename) */
/* DESCRIPTION: */
/* 播放语音文件 SxFilename */
/* ENTRY: */
/* szFilename: filename to be output. */
/* EXIT: */
/* None. */
/* ----- */
OutputFile (char *szFilename)
{
    /* 打开语音文件并播放语音 */
    if ((Handle = —open(szFilename,O—RDONLY)) != —1)
        /* 打开语音文件 */
    {
        ctvd—speaker(1); /* 将 DAC 喇叭打开 */
        if(ctvd—output(Handle) == NO—ERROR) /* 播放语音 */
        {
            voice—num++; /* 放音次数变量加 1 */
            if(ctvd—drv—error())
                ShowError();
        }
    }
}

```

```

else
    printf("Voice output ended.\n");
}
else
    ShowError();
}
else
    printf("Open %s file error\n",szFilename);
return(0);
}
/* ----- */
/* ShowError (void) */
/* DESCRIPTION: */
/* 显示在语音输出过程中出现的错误的代码 */
/* ENTRY: */
/* None. */
/* EXIT: */
/* None. */
/* ----- */
int ShowError (void)
{
    int Err;
    Err = ctvd—drv—error(); /* 取得驱动程序错误码 */
    Printf("Driver error = %2d\n",Err);
    Err = ctvd—ext—error(); /* 取得 DOS 扩充错误码 */
    if(Err != 0)
        printf("DOS error = %2d\n",Err);
    return(1);
}

```

#### 附录:

LoadDriver.c 程序清单,在声霸卡的手册上可查到。

```

/* ----- */
/* char far *LoadDriver(char *szDrvName) */
/* DESCRIPTION: */
/* 将文件名为 szDrvName 的驱动程序调入内存 */
/* ENTRY: */
/* szdrvName: Driver name to be loaded. */
/* EXIT: */
/* Pointer to the loaded driver if successful, else returns NULL */
/* ----- */
#include <dir.h>
#include <stdlib.h>
#include <io.h>
#include <fcntl.h>
#include <dos.h>
#include <string.h>

char far *LoadDriver(char *szDrvName)
{
    char far *lpDrvPtr=0;
    char far *lpPtr;
    char szDrvFile[100];
    char *pPtr;
    int Handle=1,NotDone=1;
    unsigned wDrvSize,wTemp,wDrvSeg;
}

```

广州白云山电源设备厂

**CWY 系列**

**高抗干扰稳压电源**

地址:510515 广州市沙河同和 电话:7714403 FAX:7705761

```

struct ffbk stFile;

—fmode=O—BINARY;
if ((pPtr=getenv("SOUND"))!=0)
{
    strcat(strcpy(szDrvFile,pPtr),"\\DRV\\");
    strcat(szDrvFile,szDrvName);
    NotDone=findfirst(szDrvFile,&stFile,0);
}
if(NotDone)
{
    strcpy(szDrvFile,szDrvName);
    NotDone=findfirst(szDrvFile,&stFile,0);
}if(NotDone)
{
    strcpy(szDrvFile,szDrvName);
    NotDone=findfirst(szDrvFile,&stFile,0);
}
if(NotDone)
printf("Driver file does not exist. \n");
else
{
    if((Handle=-open(szDrvFile,O—RDONLY))==—1)
    printf("Open %3 error, \n",szDrvFile);
    else
    {
        wDrvSize=(unsigned) filelength(Handle);
        if (allocmem((unsigned)((wDrvSize+15)>>4),&wDrvSeg)==—1)
        {
            lpDrvPtr=(char far *)((unsigned long)wDrvSeg<<16);
            lpPtr=lpDrvPtr;
            if(DosReadDrv(Handle,lpPtr,wDrvSize,&wTemp)==0)
            {
                freemem(wDrvSeg);
                lpDrvPtr=0;
            }
        }
        else
        printf("Memory allocation error. \n");
        —close (Handle);
    }
}
return(lpDrvPtr);
}

DosReadDrv(int Handle,char far * Buffer,unsigned wLen,unsigned *
wByteRead)
{
    union REGS regs;
    struct SREGS segregs;
    regs.h.ah=0x3f;
    regs.x.bx=Handle;
    regs.x.dx=FP—OFF(Buffer);
    regs.x.cx=wLen;
    segregs.ds=FP—SEG(Buffer);

    intdosx(&regs,&regs,&segregs);
    if(regs.x.cflag)
    *wByteRead=0;
    else
    *wByteRead=regs.x.ax;

    return(*wByteRead);
}

```

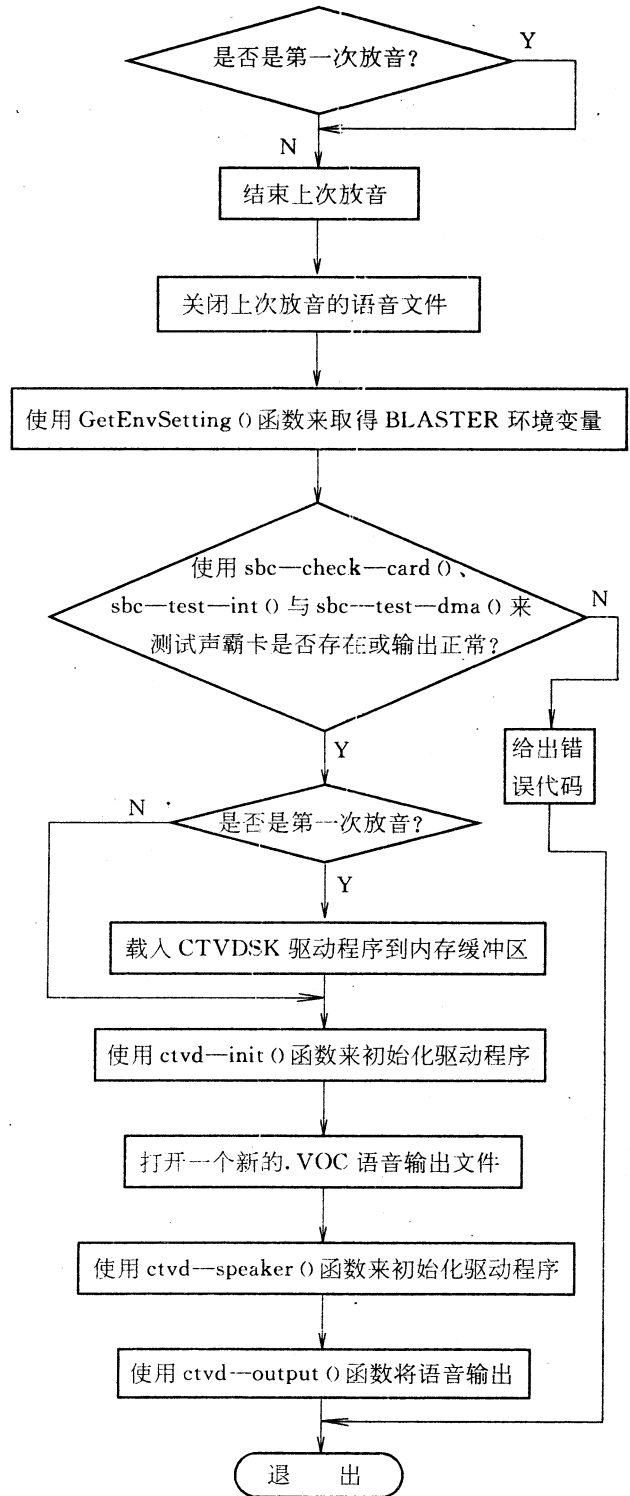


图 1

#### 参考文献:

- 1.《声霸卡万用参考手册》孙高华 台湾第三波文化事业股份有限公司
- 2.《Turbo C 使用大全》徐金梧、刘冶钢等 北京科海培训中心
- 3.《实用 C 语言编程技术》韩仲清、唐先余等 四川大学出版社

# 计算机软件技术的现状与发展趋势

华南师范大学计算机科学系 王小铭

根据国家 1991 年发布的《计算机科学技术学科发展战略》中的“计算机软件分支学科发展战略报告”指出,“计算机软件的概念,广义而言,包括系统软件、支撑软件和应用软件。随着计算机科学技术的迅速发展和日趋成熟,计算机软件分支学科当前所包含的内容可以概括为:软件基础理论和算法,构造计算机软件的方法学和软件工具以及与此相关的各种软件技术”。计算机软件技术在整个计算机科学技术领域中,占有极其重要的地位。国际上许多专家认为,软件技术有可能是未来科学技术最大的突破领域。我国国家计委、科委及经贸委近日联合发布了《九十年代中国经济发展的关键技术》,其中计算机软件被列为信息与通信领域的 9 项关键技术之一。因此,认识计算机软件技术的现状和发展趋势,进而对其基础、方法、技术和发展方向进行深入的研究和探索是非常必要的。

## 一、计算机软件技术的主要范畴

按照计算机软件分支学科的内容划分,计算机软件技术相应有如下的七个领域。

### 1、软件工程技术

- 软件开发的原则与策略
- 软件开发方法与软件过程模型——传统的瀑布模型法、JSP 方法、JSD 方法和速成原型法、面向对象(OO)方法与实用形式化技术。

- 软件标准与软件质量的度量
- 软件开发的组织工作与项目管理
- 软件版权

### 2、程序设计技术

- 程序的结构与算法设计
- 程序设计风格
- 程序设计语言
- 程序设计方法
- 程序设计的自动化
- 程序的正确性证明
- 程序的变换

### 3、软件工具环境技术

- 人机接口技术
- 软件表现工具
- 软件系统生成
- 软件工具的集成与软件开发环境
- 软件的复用
- 逆向工程

### 4、系统软件技术

- 操作系统

- 编译方法
- 分布式系统的分布处理与并行计算
- 并行处理技术
- 多媒体软件处理技术

### 5、数据库技术

- 数据模型
- 数据库与数据库管理系统
- 分布式数据库
- 面向对象的数据库技术
- 工程数据库
- 多媒体数据库

### 6、实时软件技术

- 嵌入式实时软件技术

### 7、网络软件技术

- 协议工程
- 网络管理
- 局域网技术
- 网络互连
- 智能网络

软件技术还渗透到计算机科学技术的其它领域,如人工智能、CAD、CAT 和计算机图形学、办公室自动化、MIS 与 DSS,计算机仿真等。计算机软件技术的水平对整个计算机科学技术的发展影响极大,是计算机工作者必须十分关注的研究领域之一。

## 二、计算机软件技术的现状

从近年分布的资料来看,国外计算机软件技术的发展主要在系统软件、数据库和软件工具环境三个方面,在并行分布式处理、操作系统(UNIX, WINDOWS),网络软件,系统互连与集成、分布式数据库、软件开发支撑环境、软件开发自动化计划(美国国防部的 STARS,欧洲共同体的 ESPRIT,日本的 Σ)等方面都取得了重大成果。在国内,在改革开放大好形势的推动下,软件技术有了很大的发展。特别是在软件工程和软件工具环境,并行处理算法,软件形式化研究方面。例如,北京大学研制开发的具有综合特征的软件工程核心支撑环境 β-85 系统,南京大学研制的软件开发工具和方法系统 SGG,武汉大学提出的 WUDA 规范,中科院软件所研制的基于时序逻辑的 XYZ CASE 环境系统、软件复用系统 DRS-1,中科院属下六个单位协同研制的集成化软件工程环境 MTIPS,北京大学、华东师范大学等单位联合开发的管理信息系统生成系统 MISGS,以及全国各高校、科研单位自行研制的用于软件开发,特别是信息系统开发用的众多工具与辅助开发系统等。分布式数据库



的并行算法研究已取得不少成果,对引入系统软件、数据库管理系统的汉化,汉字系统的升级与研制,面向对象程序设计方法的推广应用,网络互连技术,多媒体技术,文字处理与排版,图形绘画等实用软件在我国全面开花,呈现出一派欣欣向荣的景象。

从软件技术的现状看,有几个值得正视的问题。

1、尽管软件技术有很大的发展,但“软件危机”的问题至今尚未很好解决,原型法和面向对象方法的出现在一定程度上缓解了这种危机。但软件构成过程在本质上仍然采用手工或半手工方式,已有的软件工程方法或模型还未能软件开发过程中切实发挥作用,相应的软件工具仍有待进一步的研制。

2、在我国,应用软件的开发和应用是软件技术现状的主流。相比之下,系统软件和支撑软件的研究需要引起足够的重视。系统软件的汉化、国产化和系列化亟待解决,特别是在操作系统、编译系统和网络软件上,我们的技术队伍还非常薄弱。

3、软件技术标准、软件版权和软件的价值评估等问题在我国尚未得到重视,因而大大地影响了软件技术的发展。

4、软件技术的基础理论研究和人才队伍的建设,是摆在我们面前的严峻问题。新一代的软件技术人员必须在吸收国外先进技术和国内开创应用的同时,认真重视软件技术各个范畴里的理论研究、方法论研究和实现技术研究,才能从根本上促进软件技术的发展,缓解或解除“软件危机”的威胁。

### 三、计算机软件技术的发展趋势

从软件技术的现状和必须正视的几个问题来看,结合当前国内外软件发展的最新动向,我们认为,在今后的研究中,应该注意软件技术各个范畴的发展趋势。

在软件工程技术方面,各种软件开发方法的评价、比较、条理化剪裁,新思想和策略,软件过程模型的规范化,形式化的软件组合技术,软件的定性评估和定量测量,软件开发工具的实用性、通用性等,使软件工程方法真正成为人们构造软件过程中接受的理论、原则、规范和方法。

在程序设计技术方面,面向对象的方法及其与面向过程和逻辑的方法的结合。

在软件工具环境相关的技术方面,统一规范的接口标准,支持软件开发全过程及各阶段的各种工具的研制,并将它们统一集成到CASE环境之中。

在系统软件技术方面,原来的个人机操作系统向高性能发展,大型机操作系统则朝着更高档次的性能和个人化两极发展,各种操作系统的互补,操作系统与编译的并行化以及应用软件环境的集成和优化等。

在数据库技术方面,数据库设计方法与软件设计方法的结合,面向非结构数据的声象图多媒体数据库,面向对象的数据库,分布式数据库的设计及人工智能与数据库技术的结合等。

随着计算机科学基础理论和计算机硬件技术的发展,计算机应用的有力推动,计算机软件技术还会不断提出新的问题、新的方向。根据我国计算机科学技术发展的战略,今后一段时间我国计算机软件技术发展应当优先考虑如下课题与研究领域。

- 1、新的软件开发方法和软件过程模型;
- 2、实用的软件工程形式化技术;
- 3、面向对象的软件技术;
- 4、软件复用技术;
- 5、程序变换技术;
- 6、逆向工程技术;
- 7、软件工具集成;
- 8、人机接口技术;
- 9、并行操作系统的典型技术;
- 10、并行语言和并行编译及优化技术;
- 11、分布式计算;
- 12、新的数据库技术;
- 13、嵌入式实时软件技术。

根据笔者近年来从事教学、科研与科技开发的实践体会,从应用的角度上看,对发展我国软件技术提几点建议。

1、统一软件标准和规范并且尽快与国家标准接轨,是确立我国自己的软件产业,使软件技术健康发展的前提和保证,因此需要花大力气对系统软件、支撑软件特别是应用软件;制定一套权威的而又可操作的标准和规范。

2、要重视软件工程方法论和软件过程模型的研究,使各种软件开发方法真正成为软件构成行为的准则,并大力研制与各种开发方法和策略相应的软件工具,并且逐渐将这些工具集成为商品化的CASE环境。

3、加强应用系统集成支撑环境的研究与实践。尤其是网络环境下从网络软件、操作系统、编译系统、汉字系统、语言系统到数据库管理系统的软件平台的集成,将会受到广大计算机软件技术人员的关注。

4、面向对象的方法是当前软件技术的一个热点,必须加强面向对象方法的理论研究,和面向对象方法在程序设计、数据库设计的运用,使面向对象的思想和方法渗透到软件技术的传统领域中。

5、多媒体技术是信息综合、处理和通信发展的必由之路,如何适应多媒体大容量存储和高速处理的需要,是摆在软件技术工作者的迫切课题。多媒体技术个人化、家庭化也是值得研究的问题。

6、分布式系统技术是软件技术的另一个热点,对提高资源的利用率和信息处理的效率有重大的意义,必须开展有关并行算法,并行操作系统,并行编译和分布式数据库技术的研究。软件技术的发展有赖于软件技术工作者的辛勤劳动和不断开创的精神,让我们为我国计算机科学技术的发展,现代化建设突飞猛进而共同努力。

## 一、软件是经济 技术发展的动力之一

计算机软件法律保护,是知识产权保护的主要内容之一,这一工作做得如何,对一个国家(或地区)的经济、技术发展,有着深远的影响。“经济发展得快一点,必须依靠科技和教育”,“高科技领域的一个突破,带动一批产业的发展”(邓小平)。计算机软件是当前世界上最突出

的,能带动其它产业发展的高新科技之一,它的作用表现在下述几个方面:

### 1. 促进有形的“硬”生产工具与产品的改进

软件的应用,将使许许多多第一、二产业的生产工具和消费产品从执行“怎样做”指令变为执行“做什么”指令。真正实现生产过程的自动化和产品智能化,从而大大提高生产率和产品质量,缩短产品改进、升级、换代的周期。电脑用于生产过程控制、计算机辅助设计、辅助生产、辅助测试,电子类、机械类产品的电脑化等等都是这方面的应用的结果。

### 2. 促进带信息时代特征的第三产业的发展和形成

由于软件凝聚了大量的脑力劳动,本身便具有很高的产值(90年代初期美国商用软件每行为10美元;而宇航专用软件则每行高达1000美元),而当软件渗透到智力密集的如金融、证券、银行、财税、海关、外贸、情报、教育、计划、统计、咨询、技术服务等行业时,这些行业的服务水平年在时间和空间上都有飞跃的改进,产值迅速提高。以软件为核心的信息服务业,全球产值估计2~3年后将高达近5000亿美元。这一产业在下一世纪的地位,将会变得更为突出。

### 3. 促进社会的变革

软件用于研究、教育、学习(培训)、管理(决策与指挥)、文化、娱乐方面,形成了一系列改变人们生活和社会关系的技术和产品。如办公室自动化(OA),管理信息系统(MIS),仿真技术,文字处理技术、电子数据交换(EDI),多媒质技术(MT)、人机界面技术等等,使人类社会出现新的文化现象、管理关系和科研手段,引起重大的社会变革。

可以说,软件技术的存在和发展与一个国家(或地区)的科技、济状况有着密切的依存关系。凡是经济技术发达的国家,软件产业都相应发达,而经过软件的带动,便进入具有明显效益的良性循环;反之,没有真正利用软件技术的国家,其经济、科技都会明显地落后,停滞不前。从下面的数据可以看出经济发达国家软件发展的简况与在软件产业中占有的地位。

据权威的国际数据公司 IGC 估计,1991 年全球软件市

# 积极推动软件保护 促进广东经济、技术的发展

华南师范大学 王桂海  
广东省版权局 刘建珩 郭秀文  
电脑杂志社 吴军

场约为 526 亿美元。美国占有 35.4%(186 亿),日本占 18.6%(98 亿),欧洲占 35.5%(186 亿多)。到 1996 年,全世界软件业产值将上升到近 1000 亿元。美、日、欧洲所占的份额大体保持不变,分别达到 339 亿、186 亿、335 亿。以软件为核心,以计算机和通信设备为依托而形成的信息服务业,1991 年达到 3374 亿美元,其中美国为 1194

亿,日本 630 亿,欧洲 1997 亿;到 1996 年将达到 4799 亿,美国为 1641 亿,日本 907 亿,欧洲 1631 亿,所占份额与软件产业的状况接近。其它如 Forrest 公司,安达信公司,INPUT 公司所作的估计,其趋向与 IGC 公司的预测是一致的。至于亚洲地区,软件产业在台湾、新加坡、韩国等地,每年都以大于 10% 的速度递增,产值也相当的高(表 1)。台湾 1992 年软件产值到达 10.34 亿美元。

表 1 1991 年东南亚地区软件产值比较

国家或地区	台湾	新加坡	韩国	广东
软件从业人员(万人)	1.7	—	0.4	0.5
软件产值(亿美元)	8.7	9	2	0.15

资料来源:据 1991 年中国计算机报,计算机世界、国际电子报及有关传媒公布的数字整理

总的来说,近几年全球信息产业或软件产业的规模越来越大,而美国、日本、欧洲等经济发达国家总计约占 90%,其余地区仅为 10%。东南亚近几年经济发展迅速,到 1996 年,其份额可能提高到占 13%,约挤掉经济发达国家与地区 5~6 个百分点。这都说明,软件自身发展的迅速及其与经济发展的密切关系。

广东省的情况也与此相似:

1. 广东省一批传统工业已经用上计算机技术,而且继续需要软件技术带动、支持其发展。1992 年,广东完成的工业总产值为 2269 亿人民币。计算机应用开展的较好的传统工业在这当中占有一半以上产值(共计 1135 亿);其中,电子、纺织、机械等行业,计算机应用历史都在 10 年以上,其工业总产值则占全省总产值的近 40%(860 亿)。其它几类工业,计算机应用的广度和深度,也在逐年增加。

2. 广东为我国软件发展的三个重点基地之一,计划在深圳、珠海建立软件园。目前,深圳市软件开发和销售的企业达 200 家,软件从业人员超过 2000 人,近几年来,软件行业的产值每年以 50% 的增长率递增。1991 年软件出口达 700 万美元。1992 年,软件产业突破 1 亿元人民币,其规模与上海接近(图 1)。



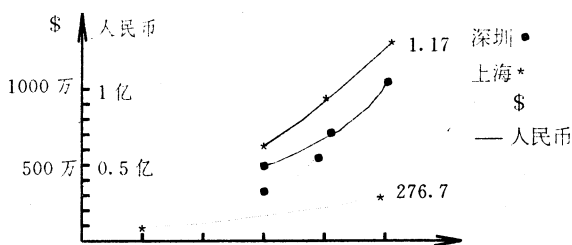


图1 上海和深圳近年软件产值增长趋势

资料来源:1993年《计算机世界》周刊

广州市从事计算机软件开发和销售的厂商超过300家,软件开发人员超过5000人。一批大学、研究所、中外合资的高技术开发公司扮演着重要的角色。由此可见,软件无论是间接(支持和带动其它产业)或直接(自成强大的产业)方面,对广东经济与技术都有着不容忽略的影响。应当抓住软件技术,求取更新、更全面的成果,作为广东经济与高新技术成长壮大的强大推动力。

## 二、广东有条件发展软件产业

广东省要形成拥有相当规模的软件产业,必须创造一定条件;从另一方面看,如果这些条件不具备,就成为当前的某些制约发展的因素。对条件和制约的问题,我们从内外环境;技术与资金;法律的保障这三方面探讨。

### 1. 发展广东软件产值的环境

市场:我国软件人员的水平较高、工资低,开发的软件具有一定竞争能力。近几年来,我省在接受软件订单、开发商品化软件,为消费类产品(大部分为家用电器)开发的配套软件等,不论对外对内都有较大的市场。

政策与认识:广东省从上(各级决策层)到下(经营者)对软件的重要性都已有一定认识。广东省科研项目中,软件开发也占有了一定份量。各类计算机企业,也相当重视软件的上市和推广使用。

综合环境:

要发展软件要有相应的综合力量。根据国内外经济专家研究,要求人均生产总值一般要达1000美元左右。从全国来看,当然与这个要求距离甚远,但从某个局部(省、市)来看,这却不是可望而不可即的条件。如上海市1990年已接近这一标准,而且事实也表明,从90年代开始,其信息服务业产值即迅速上升,1990年为人民币1.92亿元,1995年预计提高到5.27亿元。广东省人均产值1992年为人民币3400元,远远未到这要求,但珠海(人均总产值人民币1.75万)、广州(1.71万)、深圳(1.5万),则是到达了这一要求的。也就是说,具备可以较快地发展信息产业的综合环境。

所以,从主、客观方面看,广东软件发展是有一定条件的。当然,还需要有一系列的政策跟上去。

### 2. 技术与资金

广东软件开发和销售的力量,尤其是前者,与发达国家相比,差距很大,但与台湾、韩国等地相比,则相当接近。资金的投入和软件产值则小得多。(表2)

表2

地	厂家公司数	软件人员	软件开发资金 (亿美元)	91年软件产值 (亿美元)
台湾	100多	1.6	1.7	8.7
韩国	100多	0.4	0.4	2
新加坡	—	0.2	1.8	9
广东省	300多	0.6	0.03	0.15

资料来源:根据1991年有关传媒综合。开发资金一栏按公布数字表示是占GNP的百分数,台湾为0.036%、韩国为0.064%、广东为0.016%,本表是按1991年估算,东南亚地区软件产值的投入产出比为1/5反推算

所以,广东如果选择适当的重点作为突破口。技术先行带动产值提高,取得效益再反过来促进软件的创业,则在几年内、可以使软件产业迅速成长。资金尽管不太充分,但是我们可以以技术来吸取外资,以补不足。

### 3. 法律保护

法律保护是使软件形成产业的必要的条件,但又是我们非常薄弱的环节。如果说,广东从上到下对软件的重要性已有一定认识的话,那末,对软件保护,则不能给以同样的结论了。对软件法律保护重要性的承认、了解,并采取有效的落实措施,广东省有很大的不足。这个工作不抓好,势必成为广东软件发展的严重障碍。对此,下面将作进一步的讨论。

## 三、加强对软件保护的认识

软件是知识与技术的成果,作为一种产品,它与其它工业产品有许多不同之处:它的生产周期,产品形式都有很大的“弹性”,在有形的介质上固定了极为悬殊的价值,软件开发过程是艰苦的脑力劳动过程,集中了个人独特的创造性的智力结晶,在这方面,它接近于文学作品。但是,它又是实用的工具,在应用中可以直接创造价值,一个成功的软件可以带来高额的利润,这方面,它接近于工具性产品。软件很易被复制、改进,对内容进行更新和改装。这样做的时候,它因为继承了原作成果而花费成本甚小,却比原作有更大的竞争力。这样的“制品”(Ware)很易被盗窃者不劳而获或“少劳”而获。如果没有足够的法律保护,软件的劳动价值就得不到承认,软件人员缺乏积极性,盗版产品横行,也就难以实现商品化和开拓出正常的、大规模的市场。还在软件产业形成的初期,国外早已注意到要把软件纳入知识产权的范围。后来,又更明确主张用著作权法来保护软件。而专利法、商标法、反不正当竞争法等,也从不同的方面给软件提供保护。美国、日本、亚洲一些国家、欧共体,都先后这样做了。

中国在这个问题上的认识,比国际上要晚10年左右。这使得我国软件市场一度比较混乱。1990年我国公布了著作权法,1992年颁发了计算机软件保护条例,并先后加入了《伯尔尼公约》和《世界版权公约》,为贯彻实施计算机软件保护建立了司法、行政的机构。在国务院颁布的《实施国

际著作权条约的规定》(1992年9月)中,又专门对计算机软件作出相应的规定;1993年2月北京审理了全国首起软件著作权侵权案。用这样的事实,改变了中国软件保护无法可依的形象,向全世界表明,在中国,软件是得到法律保护的。这样就促使一批外商看好中国软件市场,踊跃前来投资合作。

但是,就整个软件保护工作来看,中国仍然严重落后于早期立法的国家。在国内比较,广东省又严重落后于北京和上海。

不可否认,广东在贯彻我国软件保护条例方面,做了一定的工作。如广东省版权局1992年承办的国际著作权知识训练班,召开了全省软件保护的专家座谈会。1994年又召开了全省的软件研讨会。广东省科委把软件保护研究定为“八五攻关”重点项目和软科学研究专题。机电部在广东省主办的全国性的《电脑》杂志开辟了“电脑与法律”的专栏。广东省计算机用户协会、省计算机学会都举行过多次软件保护的学术专题报告。

但是,目前广东地区的软件工作者的法律意识,仍显得相当薄弱。从广州天河高科技区、广州电脑城、广州万利广场电脑经营部等地的电脑软件经营者的抽样调查来看,掌握我国计算机软件保护条例基本知识的人并不多。某些高层的学者、专家对此了解、关心也很不够。有人甚至认为软件保护条例“解决不了实质问题”,“不相信条例能行得通”。广东软件市场盗版的情况很严重,从广州、深圳高科技产品集中的市场上来看,公开出售各类盗版的、来历不明的、自行复制的软件十分普遍。有些软件权利人明知被侵权,但亦只是感到束手无策。不少人坚持软件只能靠技术保护,而不寄希望于法律保护。广东的司法部门、软件保护管理部门也还未就广东地区的软件保护事项作过详细、认真的共同研究。反观北京所采取的措施,上海已经专门培养出1000名懂得软件保护的人员,陕西、江苏、广西等地的法学界、计算机界多次进行软件版保护的研讨活动等事实,都说明这些地区已走在我省前面。

#### 四、运用法律的武器

由于我国已有了著作权法和计算机软件保护条例,广东现在的实质问题是如何把“法”用好。我们认为,解决软件保护的现阶段对策应是“以法治乱、依法纠错、按法论罪”,充分显示出法律的作用。这个“法”包括了国际法律和我国的法律。

##### 具体的措施是:

1. 加紧普法宣传,让软件的开发人员、经销人员都了解软件已纳入有关法律保护,并知道各种法律、法规的内容及其保护对象。懂得对法律来说“知者得其利,妄者受其制”的道理。

2. 抓紧案例的办理

在普法宣传的基础上,要抓住典型的软件纠纷案件,认真依法处理。我们不主张“不教而诛”,但不能对“教而不善”

者置之不理。法律工作者,行政管理部门,软件工作者都要关注这类事态,运用典型,推动我省软件保护。同时也要运用、宣传外国案例。

3. 加强交流、树立形象

北京、上海多次举行软件保护的国际学术交流。让国际上了解我们,树立起中国软件工作者知法、守法、护法的形象。这一做法很值得学习。只有在有法治的环境,国外才放心把软件技术输入到我国。而通过国际学术交流,则可以及时了解国外的法律和案例,掌握住软件保护的主流动向。

4. 深入研究软件保护中的斗争策略。不论对国外还是在国内,都要在专利法、版权法、商标法、反不正当竞争法当中,学会如何运用法律和运用哪一种法律能最有效地保护自己,最大限度地制约竞争对手。同时也要研究对方采用的手法,他们将会如何抓住我们的弱点提出指控,我们又应如何反诉,等等。要在理论上、方法上、资料上、证据上,精神上都有所准备。

5. 一加强软件保护技术的研究

软件如何取得保护、怎样就会引起侵权、如何鉴别、如何判断、如何确定赔偿等,是目前诉讼等中的突出问题。这里有相当复杂的技术工作要我们去研究。例如:软件相似(包括实质相似性)判断的准则及计量方法研究,软件原创性的证明研究,软件“无接触”证明的法律手续研究,软件的形式与表达的分界线研究,人机接口(界面)的表达形式范围,逆向工程的合理性探讨,各类性质软件(系统软件,应用软件,数据库等)的特殊地位及其就享有的保护范围研究。被侵权后的利益赔偿计算,雇主与雇员在软件开发中的关系……等等。

6. 加强管理

建立我省的软件纠纷仲裁机构,鉴于纠纷的复杂性,有些问题可以通过仲裁机构而不必一定诉诸法律。例如:美国仲裁协会(American Arbitration Association,简称A.A.A)便在解决日美之间的软件纠纷中起到很好的作用。

初步设想,仲裁机构的早期任务是:判断告诉的软件与原作相比,是否存在“实质相似性”;判断提供分析的软件是否具有原创性的特征;软件开发过程是否真正“无接触”;判断争议双方要求的合理性。

仲裁机构的作用是调解、裁决或者向法律或其它权力部门提供可供参考的意见,它应当具有相当的权威性,其权威性的基础是仲裁的科学性、准确性、合理性,而不是行政权力。

7. 推动广东软件保护联盟“这样的民间社团的成立。

总的来说,广东省的软件产业大有前途,但这一产业迫切需要法律的保护。我们在这方面要做的事情很多,但现在必须立即从最基础的工作上认真去做,这样,才不会错过时机。

# 面向产品的 工程数据库管理模式探讨

长沙国防科学技术大学计算机系 秦 清 唐罗生 广州军区中南人防实业开发集团 谭清生

**【摘要】**本文简要介绍了工程数据库和面向对象数据库管理的主要功能和优势,针对 CAD 系统中工程数据管理的特点,提出了一种面向对象的工程数据库管理模式。即在系统中建立样板库和标准库来唯一存贮设计结果,既提高设计效率,又能有效地实现数据共享和一致性。文中以青岛电冰箱总厂 CAD 系统为实例,说明了该思想的基于 EUCLID-IS 软件的实现方法。此管理模式适用于各种面向产品的 CAD 设计系统,是一种通用的工程数据库管理模式。

## 1. 工程数据库简介

从 70 年代开始,随着 CAD/CAM 及 CIMS 技术的发展,工程应用领域的数据管理变得极为重要,由于工程应用领域对数据管理的要求与商业应用领域存在极大差异,而传统的数据库管理系统和技术是针对商用数据处理发展起来的,因而难以直接胜任工程数据的管理。各国研究机构于是竞相开展研究,虽然尚无一个公认满意的工程数据库管理系统问世,但已取得不少研究成果。

一般认为,一个通用的工程数据库管理系统应满足支持复杂实体的表示和处理,支持快速查询和良好的查询接口,支持分布式设计环境,维护数据一致性、完整性和安全性等要求。

当前公认的工程数据库技术的发展趋势有以下四个方向:

- (1) 分布式工程数据库管理
- (2) 面向对象的工程数据库管理系统(OODBMS)
- (3) 多媒体数据库管理系统
- (4) 智能化工程数据库管理

其中面向对象的工程数据库被认为是 90 年代工程数据库研究的必然方向。目前,对什么是面向对象的系统还没有一个明确一致的意见,因而对面向对象的数据库系统还没有统一、规范的定义,还缺乏强有力的理论框架支持,但面向对象的工程数据库已显示出其优越性:

(1) 用简单的概念一对象(object)一描述所有实体,一个对象有与其相关的属性集和方法。简化了设计人员的任

务和应用软件的开发

(2) 允许把任意复杂对象表示成一个循环递归的对象

(3) 提供类层次概念和伴随类层次的特性继承概念

由此,面向对象的数据库系统可以支持丰富的语义和复杂的数据模型,可以对广泛的数据库应用领域的要求提供支持,用面向对象方法的丰富语义建立的数据模型能更真实地描述现实世界。

至今为止,面向对象数据库核心系统的概念逐步取得了共识,一些技术问题得以基本解决,外围工具正在开发之中,并出现了一批商品化软件:ORION/ITASCA、GEMSTONE、ONTOS、ObjectStore、VERSANT 等。随着面向对象技术的发展,把面向对象的工程数据库管理系统应用于 CAD/CAM 中已是当务之急。

## 2. 数据组织模式

一个大型工程设计,通常要由多个设计人员紧密地合作并动态地共享设计数据库信息。要求数据库管理系统提供有效的存储管理机制、查询检索手段及交互通讯服务。另外,各种工程设计中不同程度地存在重复性工作,CAD 系统应使设计人员尽可能少地纠缠于这些重复性工作,而把时间和精力集中于创造性设计工作中。

传统的 CAD 系统对于共享数据往往采用拷贝复制的方法,造成数据多次重复,使设计数据量过大,这一点在装配时尤为突出。例如,假设一辆汽车有 300 个零件,每个零件的数据量为 1MB,在装配时,按传统方法拷贝复制零件,整车数据量往往超过 600MB。这样容易造成数据恶性膨胀影响系统功能。同时,传统的数据管理使设计的各个部分之间变得不透明,数据间的关联是靠拷贝转换建立起来的,修改某一部分时,与此相关的其它数据不会自动更新,这样使数据失去一致性。另外,按照一般 CAD 系统管理方式各个普通用户都对系统数据库进行读写操作,相互间可能干

**买正版软件到  
中外软件廊**

地址:广州流花路 119 号锦汉大厦 11 楼 电话:6689457

扰,因“人多手杂”容易危害数据安全性。

我们针对这些问题,对一种新的数据管理模式进行了尝试。整个系统中的数据在逻辑上分为三大部分,分属三个数据库:样板库、标准库和用户库。其中用户库可以是多个。如图1所示:

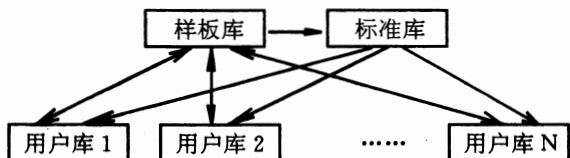


图1 数据逻辑模型

(1) 样板库:样板库中存储着系统中所有已设计完成并经审核确认的设计结果。所谓样板,是指这些数据可被各用户共享,作为后续设计的参考和依据。设计人员从其中直接引用相同的零件或引用相似零件修改形成新零件,而不必再另起炉灶从头设计起,大大节约了时间和劳动量。

(2) 标准库:标准库中存放着与工程设计有关的“标准”。如标准、紧固件、外购件、自定义标准等。为使设计标准化、系列化,对已有规定的型号系列的零件,要求设计人员不要自行设计而应从标准库中选取,这样设计结果标准、规范,也便于加工生产。

(3) 用户库:用户当前工程设计工作的临时结果及未经审核确认的最终结果都存放于各自的用户库中,用户可以对自已的库进行读写操作。用户库是用户的工作空间,一个系统中可存在多个相互独立的用户库。

上述各数据库之间的数据传递转换关系是:

(1) 用户库从样板库或标准库中引用数据

样板库和标准对一般用户是“只读”的,系统中所有用户都可共享这两库中的资源,但无权对它们进行修改,只有系统管理员才能对它们进行写操作。

(2) 用户之间的通讯

不同的用户之间可以进行数据交流,某用户可以向其它用户申请,得到允许后,可以引用其它用户中的数据,但不允许对其它用户库进行写操作。

(3) 归档

某用户的工程设计结束,设计结果经审核确认后,由系统管理员归档到样板库中,用户库中不再保留这些信息。对样板库中数据的直接引用通过指针实现,所有零、部件的数据只存贮一次,这就是数据存贮唯一性。由此,保证数据一致性。

(4) 样板库中某些零部件的“标准化”

系统中不断进行着工程设计,有些零、部件的形状,尺寸规格可能会逐渐固定下来,极少变化,经主管部门认可,把这些零、部件相关的数据转移到标准库中,要求后续设计选用这些标准系列而不再另行设计。样板库中也不再保存

这些信息。

### 3. 数据库管理系统功能及实现

#### 3.1. 基础软件简介

上文中提到的数据管理模式已应用于青岛电冰箱总厂CAD系统QBZ-CAD系统中,下面就以此系统为例说明其功能与实现。

该系统选用EUCLID-IS为基础软件。机械CAD、CAM、CAE集成软件系统EUCLID-IS是法国MATRA公司的产品,它包括了通用机械设计所需的绝大多数软件,如二维作图、参数化设计、三维实体设计、复杂曲面设计、动态机械仿真分析、有限元分析、钣金设计、2.5轴到5轴数控加工刀迹设计、仿真分析、模具设计、塑料成形分析、工厂管道设计、建筑设计等。所有这些模块都建立在EUCLID-IS系统的三维的面向对象的数据库上,是一个CAD、CAM、CAE高度集成化的设计环境。

EUCLID-IS数据库的组织结构允许信息分布在计算机网络上不同节点和不同存贮装置上,无论如何安装数据库,都保证不会失数据的完整性,在整个组织机构中能有效地共享数据。面向对象的数据库可以存贮和管理单个物体一级的信息,而不是象传统的那样管理和查询包含信息的文件。EUCLID-IS数据库管理系统能够满足对一般工程数据库管理系统的功能要求。

QBZ-CAD系统是在EUCLID-IS基础上进行的二次开发,在不改变EUCLID-IS数据库底层数据组织、存储管理的基础上,利用其提供的FORTRAN语言实现的,调用EUCLID-IS提供的标准过程,只是在逻辑层上嵌入了新开发的数据库管理系统。

#### 3.2. 各模块功能及实现

EUCLID-IS数据库的特点是逻辑组织与物理组织相对独立,逻辑组织对用户是透明的,是用户对数据库进行操作的窗口,物理组织对用户是不透明的,是系统中数据的实际存贮,一般用户不必了解到物理层。逻辑组织与物理组织之间可以建立起一一对应关系,这种对应可在系统中动态地改变。以下各模块都是在此基础上开发的。

(1) 建库模块:

系统中样板库和标准库在系统开始运行时一次性建立,然后不断动态扩充。每当一个用户申请开始一个新的工程设计任务时,动态地建立起一个用户库,并设置该库的使用权限。系统中同时可建立多个用户库,它们相对独立,互不干扰。

所谓建库,EUCLID-IS环境下,就是建立起一些逻辑

**科达电源**  
急您所急 想你所想

地址:519000 珠海翠香二路 34 号红海工业楼三楼  
电话:(0756)220324 FAX:(0756) 231980

辑结构与物理结构的连接,使用户能通过逻辑层对数据进行操作。在基于其它 CAD 软件的系统中,可以用其它合适的方法建库。

#### (2)数据库查询、检索模块:

EUCLID-IS 数据库是统一的,各库都以单个实体为管理对象,因而对各个库的查询、检索操作是一致的,只是查询范围、查询依据不同而已。QBZ-CAD 系统中有对样板库、对标准库、对用户库三种查询,只有对某库具有使用权限的用户才能查到所需结果。使用权限作为每一个实体的一个属性、可动态改变。

产品在数据库中是按树状层次结构存贮的,如图 2 所示:

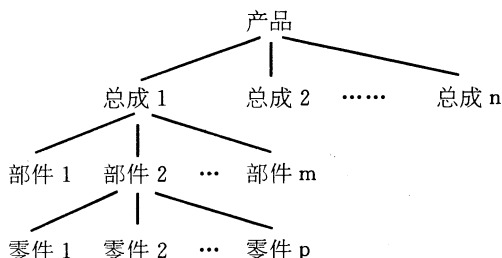


图 2 产品结构图

按照这种结构我们为其编制了一套代码来唯一标识系统中各个零件。此代码为八位长的字符串,各位含义如下:

<u>XX</u>	<u>X</u>	<u>XX</u>	<u>XXX</u>
冰	总	部	零
箱	成	件	件
型	代	代	序
号	号	号	号
(0~99)	(0~9~A~Z)	(00~99)	(000~999)

#### (3)数据引用模块:

QBZ-CAD 系统对库中数据的引用有两种方式,直接引用和引用修改。

① 直接引用是从样板库,标准库或其它用户数据库检索到某零件,不需修改就直接利用,只要建立起一个指针即可。本系统中每一实体有一个属性为“子类”,即用于记录树形结构中该节点所有的子节点。建立指针实际上未对物理存贮的数据本身作何操作。

② 引用修改是从库中检索出某零件后,对其进行编辑修改。本系统中是对三维实体进行修改,形成一个零件。被引用的零件与这一新零件已断开了联系。在后续设计中,某零件如果被修改,直接引用它的各处自动更新,而引用修改后的零件已与它无关,不受影响。

#### (4)设计结果归档模块:

某工程设计完成并确认有效后,用户库中所有的设计结果数据归档到样板库中,并从用户中删除这些数,空的用用户库可以供其它设计任务之用或被删除。这种归档不同于一般 CAD 系统中数据的拷贝复制,而是断开物理存贮的数据与逻辑框架中用户库的联系,把它们重新连接到样板库上,并改变使用权限。

#### (5)标准库更新模块:

标准库的扩充有两个途径,一是按国标、企标等标准直接设计出标准件实体存入库中;另一途径是样板库中的某些零件形成了标准系列,作为厂自定标准件转移到标准库中。这里的数据转移含义及操作与上一模块中相同,只是操作对象不同。

#### (6)样板、标准数据归档模块:

随着系统运行时间的增加,样板库和标准库中的数据量不断增多,一部分过时数据已极少或不再使用。为节省存贮空间,提高检索效率,有选择地把一部分数据归档到磁带、磁盘等存贮介质上,系统中只记录下这些数据被归档到何处,必要时,再让它们返回系统。

### 4. 优点

采用上述数据管理模式有以下优点:

(1) 多用户分布式管理系统,各用户可以充分共享系统数据资源,通过引用样板库、标准库中已有的设计成果,缩短设计周期,减轻劳动量,同时也减少设计错误。

(2) 整个系统中数据信息只唯一存贮一次,用指针实现引用,某零件修改后相关各处自动更新。具有最小的数据重复,减少了数据冗余,既保证数据一致性又大大节省了硬件资源和系统开销。

(3) 用户当前工作的数据库与存贮设计结果的数据分开,用户的操作不会影响以前的设计成果。各库、各实体都设置了使用权限并可动态改变,保证数据安全性。

(4) 用户设计工作从检索样板库、标准库开始,对已“有据可查”的零部件,尽可能利用以前成果,可使设计结果更符合系列化、标准化的要求。

(5) 本方法具有一定通用性,可方便地在多种 CAD 软件基础上实现。

### 5. 结束语

本文中提出的数据库管理模式在 QBZ-CAD 系统中实现并已投入青岛电冰箱总厂实际设计任务中,运行效果良好。

#### 参考文献:

- [1]《计算机辅助设计与制造》,M. P. Groover. E. W. Zimmers
- [2]《面向对象的工程数据库管理系统》,赵致格
- [3]《工程数据库原理》,陈其明
- [4] MATRA DATAVISION 公司,EUCLID-IS 用户手册等

人算不如电算  
EASY IS EASY  
拓展财务网络软件 EASY  
电话:(020)7501451

# 新制度财务会计计算机管理系统的设计

★万宝电器集团公司★

钟恩光

从一九九三年七月一日起在我国开始全面实施新会计制度,寻求一个满意的计算机系统去辅助实施新会计制度已是当务之急。

## 一、对计算机会计软件的分析

计算机会计管理系统(电算化),真正适合新会计制度的软件和具有与整个企业计算机信息管理系统集成化的并不多见:

1. 一些即使现在宣传“已改成适合新制度”的商业化很强的会计软件,由于它要保护原有用户的应用,由于它必然是从老会计系统的软件为基础修修补补,因此无论从它的整个系统结构、设计思想、运作模式、处理方式、语言描述、个中名词等方面都遗留较浓厚的传统会计制度的色彩和较深的旧制度的痕迹。

2. 现在一些会计软件完全是以单一的、封闭的、纯会计软件的面目出售的,把会计业务这个属于整体管理的一部分割裂出来进行管理,当企业发展到要求进行全面计算机管理时,企业将十分难堪或不知所措。因此,企业在选择会计系统时(如果打算仅从会计管理做起的话),考虑是否具有一体化的企业管理软件为支持的会计子系统是重要的。

3. 从使用的计算机语言来说,许多会计软件采用的是 DBASE III、FOXBASE、CLIPER 等数据库管理系统,有的也参插了一些 C 语言,对于使用该类语言的会计软件来说,和最近发表的 FoxPro2.5 来说其系统处理、运行速度、用户界面、帮助形式等就显然会天生不足了。

基于上述分析,这里想通过一个不太成熟的《MACS 通用会计系统》的设计过程来探讨新制度会计系统的设计方法及集成的企业计算机信息管理大系统怎样支持作为子系统的新制度会计管理系统的。

## 二、对新制度会计计算机管理系统的要求

一个成功的真正符合新会计制度的计算机系统,作者认为应考虑满足以下几个方面:

1. 不但要符合新会计制度而且要符合改革后的会计准则:

面向市场经济、和国际接轨的会计准则是这次实质性变革的标志,所谓新会计制度就是指在会计准则统驭下会计制度,因此在计算机系统设计时必须考虑诸如会计假设、会计主体、会计原则、会计要素、会计报告等带根本性的问题。

有些会计软件,所谓“改成了新制度”,实际上主要仅体现在一个平衡原理(公式)的变化,即由“资金占用=资金来源”,改为“资产=权益”,实际上远不止这么简单。而应该整个计算机系统的设计思想、软件结构上均要贯穿会计准则,而整个计算机处理过程则要始终贯彻新会计制度。

2. 必须有一个集成的企业经营管理与控制系统为平台,与其他企业经营业务的计算机管理子系统具有信息共享、联机处理的接口。

会计准则十分强调正常会计反映、历史成本计价及真实性、相关性、及时性、权责发生制等,这些都要求一个好的会计系统必须是一个集成系统的子系统,而不应是一个孤立的单纯的会计管理系统,而 MACS 通用会计系统就是基于 MACS 企业经营管理与控制系统为平台(该系统受到全国多个省市、自治区有关人士的肯定,并见《消费时报》93.5.5 日的报导、《电脑》杂志 93 年第二期文章),在库存、采购、销售、工资、固定资产管理等子系统的经营管理的控制的动态信息能及时和会计系统相衔接,而不是将库存、采购等信息事后再由会计系统录入、处理,同样,会计系统的信息如已收账款、已付账款、已预付账款等发生额应及时的传递到相关系统,使有关的业务人员能及时得到动态信息,以指导、控制他们的管理业务如发货、追款、催货等。

3. 不但具有完善的账务处理功能,更重要的要具有管理与控制功能:

4. 应充分体现新制度的规范化、标准化的精神:

新会计制度一个显著特点是在严谨的规范化及应有的灵活性,在会计方法、记账方法、科目代码、会计业务处理(会计分录)到最后报表等方面都有规范要求,如:

在财务分析方面,传统的会计系统强调图形分析的手段,而忽视分析指标的优化和规范化,新制度下,财务分析的指标准则已较为规范,因此,在财务分析计算机处理方面亦应向标准化靠拢。

从上可见,计算机会计系统如何和新制度接轨的问题还是有許多值得探讨的问题。

## 三、MACS 通用会计系统的功能

### (一) 词典管理:

本系统的会计账务处理,是通过一系列会计词典为基础管理的,正是这词典将会计准则和新制度的规范、标准体现出来,贯穿到账务处理中去。

### (二) 科目管理:

会计科目是设置帐户和进行会计核算的基础,新会计制度在会计科目体系上作了重大的改革,它按会计报表要素分为五大类,充分反映了企业资产、负债、权益、费用、损益等要素的全面情况。

本系统的处理以科目词典的定义为基础,自动进行其它相关的会计处理,科目词典的内容包括:

1. 科目代码体系:代码及名称、科目分类、科目标识、科目级别



2. 帐户标识体系: 帐户类别、借贷标识(记账方向)、转帐关系标识

3. 帐簿打印体系: 帐簿类别(三栏式、金额数量式、多栏式), 打印时段定义(按月、年打印)

4. 初始数据体系: 上年结转余额、期初余额, 累计发生额等

### (三) 凭证管理:

凭证类型: 本系统采用收、付、转凭证(即收款凭证、付款凭证和转帐凭证), 另外还有辅助凭证(或称附加凭证, 其作用是作为新财务报表的自动生成服务的)

凭证格式: 本系统凭证录入的屏幕格式、凭证打印的格式均与通常使用的凭证格式一致, 在任何涉及到凭证构成的帐务查询时(如明细帐、日记帐、凭证清单等)均可即时显示凭证整单, 直观清楚。

凭证摘要: 采用代码化录入方式, 凭证录入时允许输入摘要代码、自动代换出汉字摘要, 可通过即时提示查询摘要, 具有即时增添摘要代码功能, 当摘要库中不存在新摘要条目时, 系统将自动增添摘要条目并自动编码, 并即时生效, 在下一张凭证录入时即可调用新摘要条目。

摘要词典: 摘要词典是预先定义的一系列摘要的代码与内容, 摘要词典可在摘要维护模块中得到维护。

凭证签名: 凭证制单会计、审核等人员的机上签名, 均以代码输入, 系统自动代换出汉字姓名, 方便。

凭证录入: 凭证录入自动化程度高, 方便快捷, 对收、付、转三种凭证均按其会计记帐特征作各自的相应处理, 输入科目代码后, 可对科目的合法性、科目属性、科目级别自动判别、自动代换各级科目名称、自动确定发生额的借贷方的位置、对科目附属项目的自动确定(如属银行存款科目, 系统要求录入银行帐号代号、结算凭证类型及号码、以供银行对帐用), 能方便的增加、复制、删除凭证纪录, 转帐凭证录入时, 可复制摘要相同而记帐方向相反的凭证纪录。

凭证管理功能还包括: 自动分录、计数器功能、凭证校验、凭证审核、凭证汇总等会计功能。

自动分录: 当在凭证摘要中输入了自动分录的代码时, 系统将从分录词典库中自动生成若干条有摘要、有科目代码及名称的会计分录;

计算器: 在任何金额、数量录入时, 可调出计算器进行运算, 结果自动放入正在录入的字段中;

凭证效验: 提供多种凭证效验功能,

### (四) 记帐、结帐管理:

记帐管理包括: 会计记帐管理及出纳序时记帐管理, 适应不同岗位财务人员的工作需要;

本系统采取各会计人员各自独立制单、各自校验、集中审核、集中自动记帐的处理方式。记帐时 明细帐、日记帐、总帐等一次性完成。

可分期、分批多次记帐, 记帐时系统将自动再次进行凭证校验、试算平衡、凭证号连续性检验等合法性校验, 如不符合记帐条件系统将拒绝登帐。

结帐管理分为: 科目汇总, 冻结本会计期登帐作业, 月

终结帐, 年终结帐。

### (五) 帐簿管理:

帐簿管理包括帐簿生成、帐簿查询、帐簿打印等

### (六) 对帐管理:

系统提供银行对帐功能, 只要按规定输入收、付凭证中与银行存款科目有关的信息及银行对帐单, 本系统即可进行自动对帐, 并自动生成规范的银行存款余额调节表, 可输出打印调节表; 银行对帐采用先自动对帐、而后由用户机上人工对帐, 经用户人工调整、确认后, 再正式生成、打印银行存款余额调节表;

### (七) 报表管理:

本系统提供两种报表生成、输出方式; 统一规范报表的管理;

规范报表仍然提供用户自定义方式, 即当用户的报表与财政部规定表式有所增减时, 系统提供栏目的增删及计算公式的编辑功能, 因此报表处理灵活、适应性强。

系统提供自定义报表的表式编辑、计算公式编辑等功能

### (八) 自动转帐管理:

本系统, 自动转帐功能包括:

1. 自动转帐;
2. 自动分录;
3. 自动结转;
4. 跨系统自动转帐;

### (九) 出纳作业管理:

出纳管理包括: 出纳序时帐记帐、查询、打印, 出纳日记帐登记、生成、打印, 银行对帐管理等; 出纳与会计作业的衔接可在系统维护中设定。

### (十) 安全保密管理:

1. 工作人员的注册、权限管理;
2. 工作人员的密码管理;
3. 作业密码管理;
4. 系统工作日志管理;
5. 会计数据安全;

### (十一) 系统教学与实习管理:

本系统提供具有直观、全面的教学实习管理系统, 可对用户进行全系统的培训;

本系统提供本软件的实习系统, 用户可在实习系统下运行任何模块、任何功能, 而对用户的在用系统的任何数据、会计帐本无任何影响, 对于用户中的新手, 尽快熟悉、使用本系统、减少差错, 具有重要意义。

### (十二) 桌面办公系统:

本系统还提估面办公系统, 其功能有计算器功能、日历、记事本、万能表、文件编辑等, 会计人员可调用这些功能, 以辅助他们的桌面办公。

## 四、MACS 通用会计系统的技术设计

1. 在系统设计上采用模块化、标准化、网络化技术, 程序适应性广, 可改性强、维护性好。屏幕格式、操作方法、代码方案、数据结构都采用一致的标准, 具有一致的风格。

2. 人机界面友好, 联机帮助功能强, 在各种查询、录

入、修改等功能下,均可得到明确的帮助信息和提示信息,在有限的屏幕中,给出多库、多业务的相关信息,在所有运行过程中,系统的运行状态、出错信息、操作提示等都不断地给出用户,系统透明度好。

3. 灵活性强。MACS 系统既可运行于微机网络多用户环境,又可单机运行,各功能模块的信息既可联机、多模块、交叉实时地动态处理,又可不联机、进行信息的集中处理。

4. 本系统采用 FoxPro2.5 数据库管理系统编程:

FoxPro2.5 符合最优秀的 XBASE 工业标准,具有开放性,由它设计出来的程序可以在 DOS,WINDOWS,U-NIX 等多种操作系统下不加修改的运行,对于用户今后的管理程序升级换代奠定了可靠的基础,保护了用户的软件

资源;

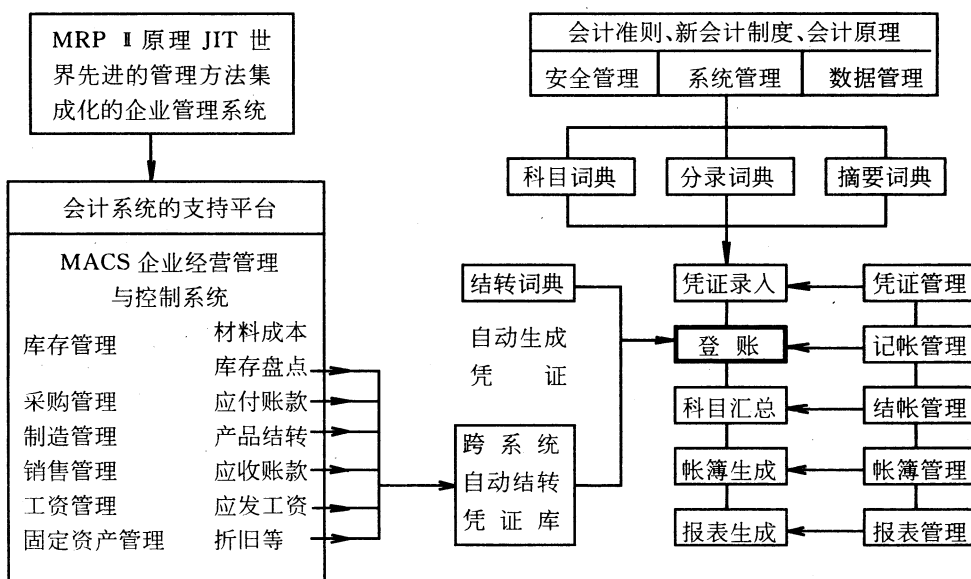
FoxPro2.5 具有极佳的工作效率、极快的速度和完备的功能,由它设计的程序比 Foxbase2.1 快 25 倍,大大地提高了工作效率;

FoxPro2.5 具有良好的多用户性能,支持很大的数据库记录,即使超过 2000 条记录,其运行速度亦不会显著降低。

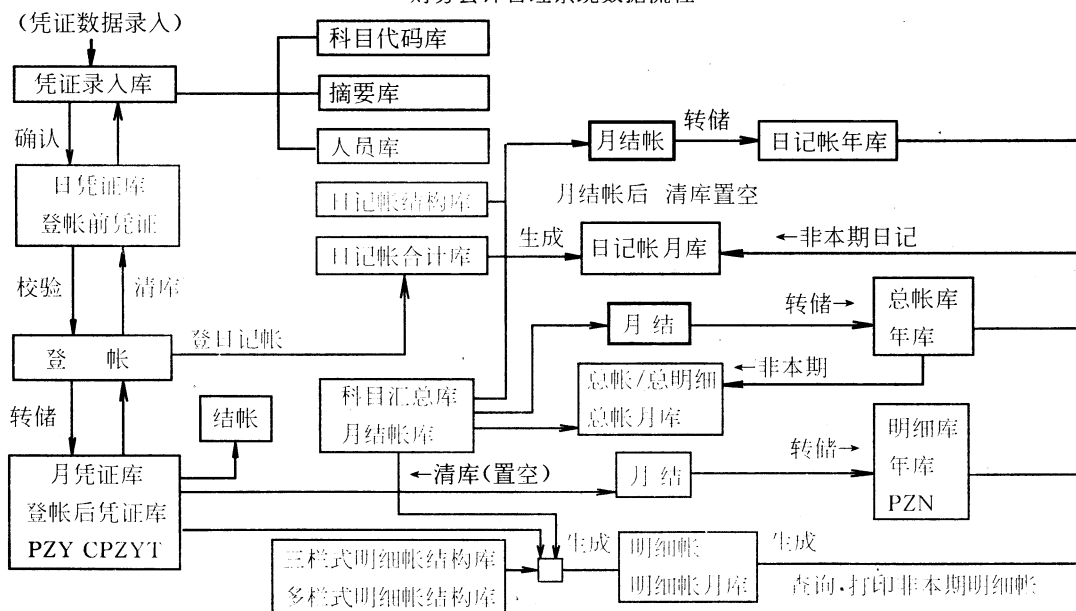
除此之外,本系统还最大限度地采用了数据词典技术、优化查询(Rushmore)技术、直观示例查询(RQBE)及结构化查询语言(SQL)等,使用户可以多种方式高速检索数据库记录。

附系统流程图如下:

MACS 通用会计软件系统结构示意图



财务会计管理系统数据流程



# 介绍一种功能强大的软件

## WORKS

● 杨 峰

Microsoft 公司出版的 Works 是一个功能非常强大的办公室用软件,它的主要功能模块有四个:文字处理(Word Processor),电子表格(Spreadsheet),数据库(Database)和计算机通讯(Communication)。Works 2.0 有两个版本,一个是为 DOS 环境开发的 Worksfor Dos,另一个是为 Windows 环境开发的 Works for Windows。如果完全安装,前者约占 5M 硬盘空间,后者约占 7.5M 硬盘空间。

Works 与许多常用软件兼容:文字处理与 Word Perfect, Windows Words 及 DOS 编辑器兼容;电子表格与 Lotus 1-2-3, Symphony, Multiplan 等兼容;数据库与 dBASE III、dBASE IV 兼容。

这是一个全屏幕操作,下拉式菜单驱动的程序,并有在线帮助及辅助教学。在内存允许的情况下,可同时打开多个文件。虽然一次只能对一个文件操作,但打开的文件间可交换信息,非常方便。

在 Works 的四个功能模块中,设计最精巧的是数据库,给人耳目一新的感觉。下面,我依次加以介绍。

### Word Processor (文字编辑器)

Works 中的编辑器非常方便适用,可用与编辑信件、通告、论文等。

它可以自动换行,分层缩进;对刚刚进行的一次操作,有取消(Undo)的功能。对欲编辑的文件块,可按 F8 后移动箭头将其变亮,然后可进行拷贝、搬移、删除等操作;或者改变其字体;或者加上斜体、粗体、下划线等属性;或者改变其排版格式:左对齐、右对齐、中间对齐、左右同时对齐;还可设置打印,并有模拟显示打印结果的功能(Preview)。

在编辑的文件中可插入图画,电子表格图表,特殊字符,甚至数据库或电子表格的部分内容。系统中带有十多个图案,在 Worksfor Windows 中还可作出新图案。

如果与数据库同时使用,可打印出格式信件(Form Letters),例如,可给数百人打印内容基本相同的信件。

此外,它还有检查拼写错误、查找同义词、单词计数等实用功能。

### SpreadSheet (电子表格)

电子表格是一个非常方便的计算、分析工具,可用于财务计划、预算控制、记账、科学计算等方面。

象 Lotus 1-2-3、Symphony 中一样,表格中可输入各种各样的信息:字符串、数字、日期、公式等。Works 中的公式总是以“=”开始,公式既可进行数学运算,也可进行逻辑判断。

如果要对多个格子进行操作,可按 F8 后移动箭头将

其变亮,然后可进行拷贝、搬移、删除、改变显示方式等操作。

可用 FillDown 或 FillRight 命令对多个格子填充相同的数据或公式。

还可用 FillSeries 命令在多个格子填充有规律变化的数据,比如递增的数、递减的数,甚至日期。

在表格中填充好了数据之后,可根据这些数据作出需要的图表(chart),比如柱状图,饼图,线图等这些图表可让你从不同的角度直观地观察和分析你的数据。比如,你将公司在十二个月里的销售额输入表格中,作柱状图可直观地看出各月份间增减情况,作饼图可直观地看出各月份销售额在年度中所占份额。

### Database (数据库)

数据库是 Works 中设计最富有特色的部分。它可以帮助你建立资料档案,快速找出所需资料,打印出满意的报告。

数据库有 3 种状态:Form, List 和 Report。其中 Form 状态就好像在 dBASE 中通过 format 文件来观察数据一样。但 Works 中允许你全屏幕地移动任意一个域(field),也可在任何地方增加或删除一个域。List 状态就好像在 dBASE 中用 Drow 命令查看数据库一样,每一列是数据库中一个域,每一行是一个记录。而 Report 是专为向打印机的输出设计的。在 dBASE 中如要打印出一个报告,需要很麻烦地计算每一行打印多少字,空多少格。但 Report 中可让你很方便地进行全屏幕编辑,画出边框并对齐每一行,并且几乎不用担心纸张大小,因为可以选择不同字体及大小。打印机将准确地打印出你设计的报告。比如,某公司建立职员资料档案库,有年龄,姓名,号码,性别,电话等资料,现在需要打印出职员的以上资料,则只须如图 1,在屏幕上打入所需信息,然后选择打印,将得到如图 2 报告。其中最后一行是求平均年龄。

Title

Title

Headings

Headings

Headings

Record

Record

Summary

Summary

Summary

NAME	NO	AGE	SEX	TELEPHONE
=NAME	=NO	=AGE	=SEX	=TELEPHONE

AVERAGE AGE = AVG (AGE)

图 1

# 实时接收程序的设计方法

大连旅顺 81841 部队 王伟

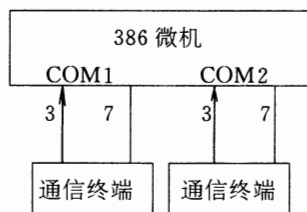
**【摘要】**本文介绍在 UNIX 和 DOS 环境下用 C 语言和汇编语言完成对通信终端机输出信息的接收与处理的方法。并给出 386 微型机加装多用户扩充卡的步骤和在此基础上进行多路实时接收的实例。

## 一、通信终端的数据规格

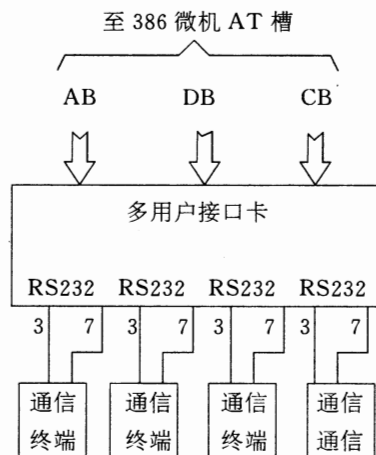
现代数字通信越来越依靠计算机进行数据加工处理。通信终端设备与计算机的接口日益规范。特别是在卫星和短波数字通信中,代码信息经过数据终端解调处理后相继续转换成五~八单位的数据,传输速率范围在 50~9600 波特,这给计算机接收创造了条件。通信终端将数据以字、组、块的方式弹出,受消息的影响,信息流有不定长的间隔时间,使完全实时接收、记录成为可能。

## 二、计算机接收方式设定

计算机一般具有由 8250 芯片构成的 RS232 通信接口,设备端口 COM1 口地址 3F8H,COM2 口地址 2F8H。两口均可接收五~八位字长,50~9600 波特的串行信号。其通信规格记为:ST B<sub>0</sub> B<sub>1</sub> B<sub>2</sub> B<sub>3</sub> B<sub>4</sub> B<sub>5</sub> B<sub>6</sub> B<sub>7</sub> PTY SP 其中 ST 为信号(低电平),B<sub>0</sub>~B<sub>7</sub> 为数据位,PTY 为校验位,可分为奇校验(odd)和偶校验(even)。SP 为停止位(高电平),其长度可为  $1, 1\frac{1}{2}$ , 或 2 位。在 DOS 环境下 ST B<sub>0</sub>~B<sub>7</sub> SP 至 ST B<sub>0</sub>~B<sub>7</sub> PTY SP 均可应用,只需在初始化时对 8250 设不同的参数。在 UNIX 环境下,终端和系统进行通信时一般采用 ST B<sub>0</sub>~B<sub>6</sub> mark SP 和 ST B<sub>0</sub>~B<sub>7</sub> SP, SP=



单用户微机用通信终端连接



四用户接口卡与通信终端连接

图 1 通信终端与微型机的连接

## REPORT

NAME	NO	AGE	SEX	TELEPHONE
ZHNAG SHAN	1	20	MALE	404088
LI SHI	2	22	MALE	262095
WANG WU	3	30	FEMALE	642708
ZHAO LIU	4	24	FEMALE	272404

AVERAGE AGE 24

图 2

在数据库中,各个域的数据可逐个手工输入,也可用公式代替手工输入,这是一个极大的改进。公式中可进行简单的四则运算、逻辑运算,也可包含 Works 提供的数十种函数。

在 List 状态下可进行块操作,按 F8 后移动箭头将多个记录和域变亮,然后可进行复制,移动,删除等操作。

可作 Queiy。其中功能同 dBASE 中的 Select 命令一样,找出满足某个条件或某几个条件的记录。不满足条件的

记录全部隐藏起来。此时所有的操作都只针对这些满足条件的记录。

## Communication (计算机通讯)

在 Work for DOS 中有计算机通讯的功能。首先,用于通讯的两台计算机要用装有 Modem 的电话线连接起来,或直接用通讯线连接 COM 口。两台计算机必须同时打开一个通讯文件(\*.WCM)。该文件只能存贮 Works 中通讯所需的命令,比如设置,电话号码等。

通讯方面的设置包括三个方面的内容:电话设置(电话号码,Modem 设置,音频或是脉冲),信息流动方式(COM 接口,波特率,数据位,停止位,奇偶校验等),信息显示方式(字符集,缓冲区大小)。

如果两台计算机以上设置相匹配,且接线良好,就可通过键盘在显示器上通讯了。

总之,Works 会成为你办公室自动化的好帮手,它能极大地提高你的效率。希望我的介绍能使你喜欢它。

1. 实时接收一般是盲收, 即对处理机发来的信息全部接收。图 1 给出计算机在单用户和多用户情况下两种联机方式。

在单用户下通信终端通过计算机的 RS232 接口连接, 接收程序按通信终端的信息流格式设定通信协议, 控制信息的读写。在配有多用户卡的计算机上若使用 UNIX 操作系统, 计算机终端先以用户方式登录, 进入系统, 然后调用接收程序使接收准备完毕, 随后将多用户卡上的 RXDA 接收线调至各通信终端的数据输出上 (可用微动开关控制), 此时计算机接收的是由通信终端发来的信息。信息通过计算机反弹至显示终端上, 可用来监视。多用户卡在 DOS 环境下可设成扩展的 RS232 接口, 4 片 8250 对应不同的中断线和地址, 编写对应的中断服务程序, 同样也可接收四路信息。

### 三、多用户接口卡的设定

多路数字信号的接收比较方便而直接的方法是利用多用户接口卡。目前较为流行的多用户卡大都采用 AST 四用户卡的扩展方式, 与 386 微机及 UNIX SYSTEM V 在硬软件方面兼容。采用二块卡可完成八路信息的接收。下面对四用户接口卡设定做一介绍。(多用户方式)

#### 1. 使用多用户卡。

首先正确设置卡上的跳线, 使系统能够识别该卡。设置卡上的 DIP 开关选择中断线。

在 386/AT 微型机中, 中断线由 2 片 8259 进行控制, 其中断线和中断请求的关系如下:

中断线	中断源	DIP
IRQ3	COM2	6ON
IRQ4	COM1	7ON
IRQ5	并口 2	8ON
IRQ6	软驱	
IRQ7	并口 1	10ON
IRQ10~IRQ12	保留	5,4,3(ON)
IRQ13	协处理器	
IRQ14	硬盘控制器	
IRQ15	保留	2(ON)

根据所选的中断线来设置 DIP 开关, 选中的置 DIP 为 ON 状态。本例中选 IRQ3, IRQ4, IRQ5, IRQ7 做为中断线。完成上述工作后将接口卡插入微机内的 AT 槽口。

#### 2. 在 UNIX SYSTEM V 环境下对系统文件进行部分修改。

首先利用安装软件 pkginstall 安装 unlimited Runtime system (非受限证书) 和连接工具箱 Link kit。

```
#pkginstall✓
```

按菜单提示插入盘片, 进行软件安装。

然后利用 Vi 编辑修改 /etc/atconf/modules/asy/config 文件, 将文件中的 intvec=3,4 改为 intvec=3,4,5。将修改后的 config 文件写回。

修改 /etc/atconf/modules/Lp/config 文件, 将文件中的 intvec=5,7 改为 intvec=7 将修改后的文件写回。

#### 3. 重构系统内核

```
#CONF=/etc/atconf;export CONF
```

```
#/etc/mRunix 记住系统返回的重构次数区
```

```
#cp/etc/atconf/KerneLs/unix.std 8/unix
```

```
#sync
```

```
#init Q 退出系统
```

#### 4. 修补核内的 tty 参数。

设起始 tty 号为 tty02, 中断行号 5,8250 基址 672, 在此中断行上连续的 VART 号为 4。

```
#ttypatch -t2,-i5,-a672,-n4,-v703,-mQ
```

```
#sync
```

```
#ttypatch -d -t2-n4
```

```
#sync
```

```
#init Q 退出系统
```

#### 5. 运行 sysadm 管理程序, 激活上述设定的 ttyQ2~ttyQ4 终端口。

# sysadm 出选单后选 ttymgmt, 再选 3, 将 ttyQ2~ttyQ4 终端设为 ON, 设定与通信终端相对应的波特率, 安装其参数后退出 sysadm。此时多用户接口卡安装完毕。它对应产生 /dev/ttyQ2~/dev/ttyQ5 设备。

#### 6. 建立接收程序运行的环境目录

运行 sysadm usermgmt, 建立 /usr 下的用户目录, 注册名。将用户目录分配给一个通信终端。

至此你可在 Login 提示下以注册名登录, 并调用接收程序进行数据接收。四用户卡可在不同的目录下实施对四路信息的接收。

### 四、接收程序实例

#### 1. 在 UNIX 下的接收程序

在 UNIX 环境下通过对终端设备名的读写即可实现终端与主机间的通信。接收程序根据通信的需要建立数据文件, 较常用的有四种建立方式。一是根据通信终端发送的每份消息建立相应的文件, 一般应用在有固定收发关系格式的场所。第二种是根据时间定时建立文件。文件名取自系统的时间参数, 这种方式应用在对信息进行定时加工处理的场合。第三种是根据信息量建立相应的文件, 每个文件有相等的块长度。适用于大量数据块接收。第四种是由用户定义一个文件名, 所接收的全部信息原样记录到该文件下, 用来进行存档。文件名的定义由参数传递的方式最为方便。下面给出在 UNIX 环境下用 C 语言编写的接收程序例。其功能是接收通信终端发来的数据。每日零点自动更换文件名, 在每份消息结束后追加接收时刻的日期、时间。文件名形式为 fna xxx yy, 其中 xxx 表示月份 (英文缩写), yy 表示日期。要求信息流每行 < 256 个字符, 行结束时含有换行, 回车码、0AH, 0DH。程序可直接生成 2 个文件, fna xxx yy 和 fna-data, fna-data 文件是滤掉空格、回车、换行符

以每行 40 码进行存贮的数据文件,仅保留当日所接收的信息。fna xxx yy 则是原样照印的通信终端发来的信息。以上两数据文件均可在其它终端上进行加工处理。

tele 接收程序编译后生成的目标文件拷贝到/bin F, 修改其文件方式和文件主。

```
#CL-O tele tele.C
#CP tele /bin/tele
#chmod 0777 /bin/tele
#chown bin/bin/tele
#rm tele
```

这样在所有目录中均可利用 tele 程序进行信息接收。程序在收到 DEL 信号后中断退出。在主控台上也可用 PS-e 命令得到其进程号,用 kiu 进程号取消该进程,返到 sheel 下。

附 tele.C 源程序在 AST286,SZ386,Inter386、UNIX 操作系统 V 下通过:

```
main ()
{
    int i,n,s,sg,pi,fd1,fd2,fd3,fd4,k,k1;
    char buf[1024],c,buf1[30],name[10],name2[10];

    start:
        printf(" FROM TELEVIDEO TO SZ-386 V/386 UNIX . TO\n");
        WAIT FOR INPUTING. \n");
        creat("date1",0777); /* 建立时间文件 */
        fd3=open("date1",2);
        system("date>date1");
        read(fd3,buf1,28);
        close(fd3);
        system("rm date1");
        name[0]='f'; /* 建立接收数据文件名 fna */
        name2[0]='f'; /* 建立信息过滤后的数据文件名 fna-data */
        name[1]='\n';
        name2[1]='\n';
        name[2]='a';
        name2[2]='a';
        name[3]='-';
        name2[3]='-';
        name[4]='d';
        name2[4]='d';
        name[5]='a';
        name2[5]='a';
        name[6]='t';
        name2[6]='t';
        name[7]='a';
        name2[7]='a';
        name[3]=(buf1[4]+32); /* 向文件名追加系统时间 */
        name[4]=buf1[5];
        name[5]=buf1[6];
        if(buf1[8]==32)
            name[6]='0';
        else
            name[6]=buf1[8];
        name[7]=buf1[9];
        if((fd1=open(name,2))>0)
            lseek(fd1,0,2);
        else{
            creat(name,0777);
            fd1=open(name,2);
        }
        s=0;
```

```
sg=0;
i=(-1);
    creat(name2,0777);
    fd4=open(name,0);
    fd2=open(name2,2);
    while((n=read(0,buf,512))>(-2))
/* 读取来自标准 tty 的信息 */
    {
        for(i=0;i<n;i++) /* 判断信息结束标志 */
        {
            if(buf[i]==64)
                buf[i]=' ';
            if(((buf[i-2]=='*')&&(buf[i-1]=='*'))
                ||((buf[i-3]=='e')
                    &&(buf[i-2]=='n')&&(buf[i-1]=='d'))))
            {
                sg=1; /* 向文件追加时间信息 */
                pi=i+2;
                buf[i]=32;
                buf[i+1]=32;
                mknod("date1",0777);
                system("date>date1");
                fd3=open("date1",2);
                read(fd3,buf1,28);
                close(fd3);
                system("rm date1");
                if((buf1[11]>'0')&&(buf1[12]>'0'))
                    k1=0;
                k=1;
                if(((buf1[11]=='0')&&(buf1[12]=='0'))
                    {
                        k=0;
                        k1++;
                    }
                else
                    k=1;
            }
        }
        if(sg!=1) /* 滤除空格.回车后数据写入 fna-data 文件 */
            write(fd1,buf,n);
        else
        {
            write(fd1,buf,pi);
            buf[0]='\n';
            write(fd1,buf1,28);
            write(fd1,buf,1);
            write(fd2,buf,1);
            i=(-1);
            while((n=read(fd4,buf,1))>0)
            {
                if(buf[0]=='\n' || buf[0]==' ')
                    buf[0]='\0';
                else
                    i++;
                if(i!=40)
                    write(fd2,buf,1);
                else
                {
```



```

c=buf[0];
buf[0]='\n';
write(fd2,buf,1);
buf[0]=c;
write(fd2,buf,1);
i=0;
s++;
}

printf("\n");
printf("THE NUMBERS OF CHARACTER THAT IS
SENT TO DATA FILE ARE %d.\n",40*s+i+1);
s=0;
sg=0;
if(k==0&& k1<=1){
    k1++;
    close(fd4); /* 关闭 fna 和 fna-data 文件 */
    close(fd3);
    close(fd2);
    close(fd1);
    goto start; /* 每日零时再建新文件 */
}
}
close(fd4);
close(fd3);
close(fd2);
close(fd1);
/* 关闭打开的文件 */
}

```

## 2. 在 DOS 环境下的接收程序

在 DOS 环境下一般采用中断方式进行信息接收,中断向量为  $0BH \times 4, 0CH \times 4$ 。中断程序有二种运行方式被经常使用。一种是采用驻留技术,在中断发生后自动从当前的处理机状态转入相应的中断服务程序。前台可以在命令解释程序下进行其它操作,中断返回后击任一键可返到原状态。另一种是在主程序中等待中断的发生,处理机资源全部由主程序控制。采用这种方式运行且在中断处理程序中不进行硬盘的读、写操作,程序工作稳定、可靠。若在中断处理程序中进行 DOS 功能调用,执行磁盘的读写操作就容易造成系统的死机,特别是采用驻留方式的程序。解决这一问题的方法是在中断处理程序中加设信号灯操作,运用  $int21H, 34H$  功能和  $int28H$  功能调用,检查 DOS 是否处在临界状态,若在临界状态则继续向缓冲区写字符,否则将缓冲区内容写盘。写操作的时机最好处在的一组消息的结束,以不影响信息接收的完整性。

在具有多个 RS232 接口的微机中,实现多路接收可分别对 8250 初始化,建立相对应的中断处理程序。如在 286 微机中 COM1、COM2 口地址分别为  $3F8 \sim 3FFH$  和

$2F8 \sim 2FFH$ ,中断类型为  $0CH, 0BH$ ,可实现 2 路信息流的实时接收。信息的显示可采用窗口技术,将屏幕分割。下面给出 DOS 环境下利用汇编语言实现实时接收的主要部分流程。图 2、图 3、图 4 流程附后,需要汇编源程序请与编辑部联系。

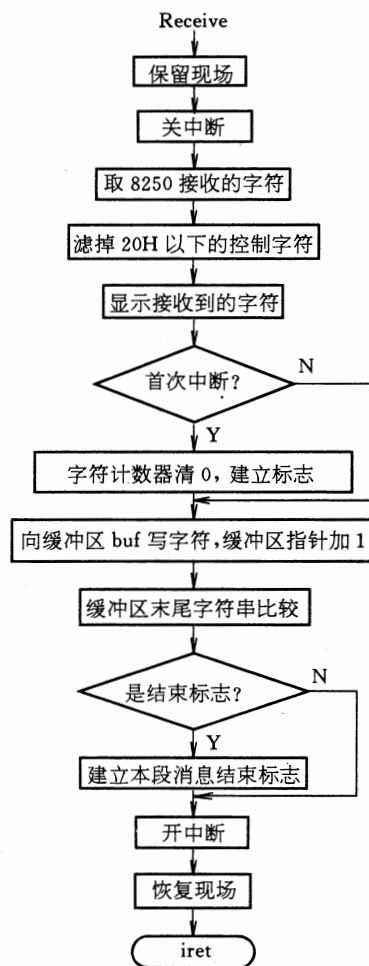


图 2 RS232 中断(0CH)服务程序流程

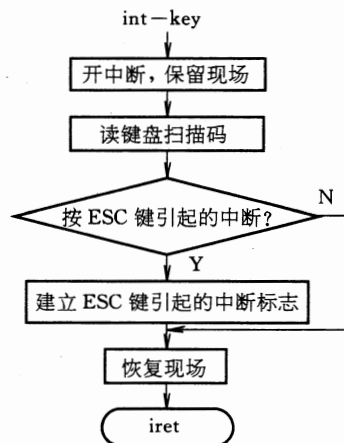
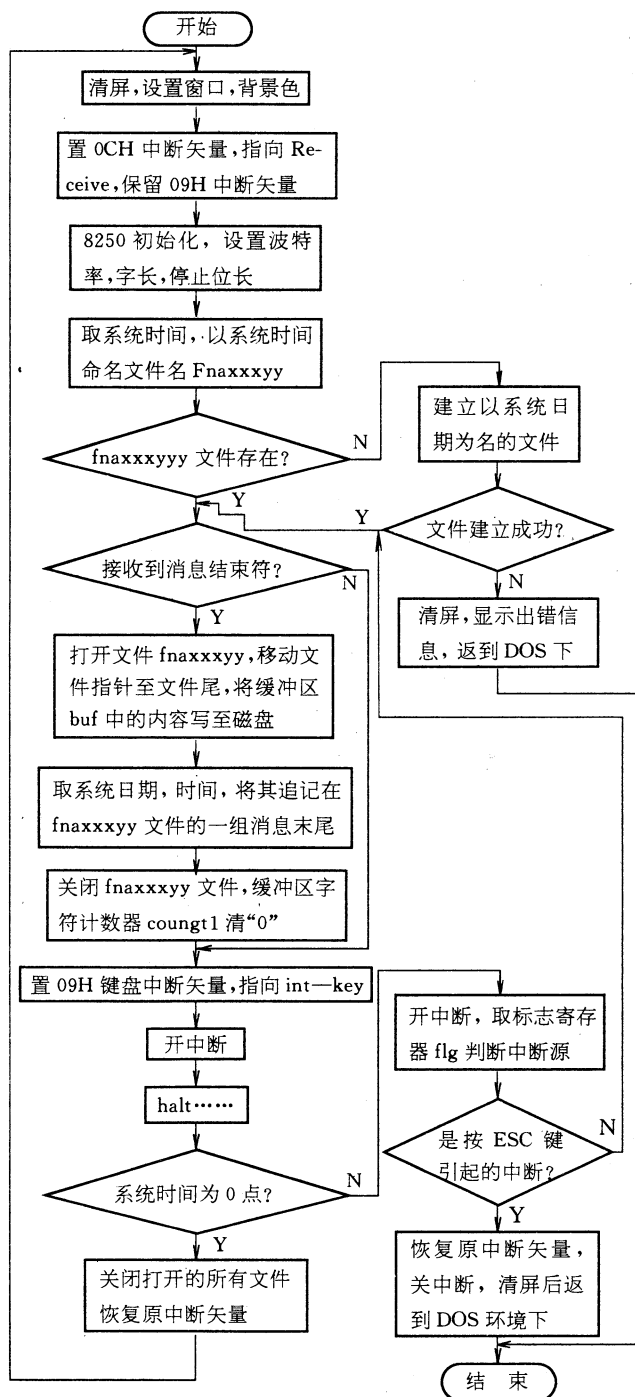


图 3 键盘中断(09H)服务程序流程

**欲免海盗软件之害  
请到中外软件廊**

地址:广州流花路 119 号锦汉大厦 11 楼 电话:6689457



#### 参考文献:

- (1) 北京计算中心 SYSTEMV/386《程序员参考手册》
- (2) ASYNC CLUSTER ADAPTER multichonneL Board proriding Four Indiridually Addressable RS-232 seival ports on IBM PC/XT/AT and compatibles User's Manval AST Research Inc
- (3) 贺志强、李昌译《DOS 磁盘操作系统高级程序员指南》北京联想计算机集团公司
- (4) 王伟《DVAL68000 83/80 微型机与 TP801 通信接口》电子技术应用 87.10.

## NetWare v3.11→

## NetWare SFT III 3.11

交通部广州信息所 涂超汉

南方四通信息研究所 楼新平

Novell 网络是美国 Novell 公司发表的世界上第一个真正独立于硬件并实现了真正文件共享的个人计算机网络系统。它的成功之处在于有一个精心设计的网络管理软件 NetWare。

自一九八九年 Novell 公司正式推出 NetWare 1.0 版本至今, NetWare 已有 20 多个正式发表的版本, 其产品有早先的 NetWare286 (包括 ELSI、ELSII、ADS、SFT) 和 NetWare386, 近年来又先后发表了 Netware4. 01, Net 4. 01 for OS/2 以及在中国市场大受欢迎的 Net Ware SFT III 3.11。

本文除了对 NetWare SFT III 3.11 作了较详细的论述外, 对如何把 NetWare v3.11 升级到 NetWare SFT III 3.11 作全面的介绍。

### 1. 三级系统容错功能

在 NetWare286 的版本中已经出现过系统容错 SFT (SystemFault Tolerant) 的功能, 但它仅仅局限于磁盘镜像, 即只能做到在磁盘这一级的容错。在 NetWare SFT3.11 版本中所提及的是系统容错功能, 指的是三级系统容错技术; 第一级 SFT 技术是围绕着硬盘表面介质随机损坏而设置的; 第二级 SFT 技术是为防止硬盘或硬盘通道失效而设置的; 第三级是防止服务器故障而设置的。可以说, 在 NetWare 家族中, NetWare SFT III 3.11 是真正做到了系统的容错而不是局部的容错。正因为如此, 所以, NetWare SFT III 3.11 在证券、银行、民航等不能中断工作的各部门受到青睐的原因所在。

三级容错技术主要的有:

- 写后读验证
- 差错检测与校正
- 表面热修理
- 冗余目录结构
- 自管目录管理
- 开机加电时目录验证
- 磁盘映象
- 磁盘表面介质失效恢复
- 磁盘通道映象

- 服务器映像

此外, SFT 还包括事务处理跟踪系统, TTS 和 UPS 监控。

由于 Novell 网络采用了上述 SFT 技术, 从而保证了 Novell 网络系统有极高的可靠性。

## 2. NetWare SFT III 3. 11

### 2.1 概述

NetWare SFT III 3. 11 是一个网络操作系统, 用于要求最高可靠性的网络计算。SFT 包括了 NetWare v3. 11 极大部分的功能, 同时又增加了当今最高水平的容错技术。

NetWare SFT III 3. 11 是通过二个在物理上彼此独立的服务器—主服务器与次服务器的集成, 时刻监视着网络是否发生故障而造成停工。二个服务器则要求有相同的内存映像和相同的磁盘容量。如果主服务器由于硬件故障而被迫挂起, 那么, 次服务器自动地成为新的主服务器, 这一处理过程对所有网络上的用户是瞬时的, 又是透明的, 即所有网络上的文件服务和应用对网络用户来说是继续有效, 现场不受任何干扰和破坏。

二个服务器不一定需要在一个网络段上, 不需要有相同的 LAN 适配器的类型。但是, 服务器必须在相同的交互网络上, 所有的客户必须能够得到二个服务器的数据报文。

### 2.2. SFT III 的特殊功能

NetWare SFT III 3. 11 具有其它网络系统无法比拟的特殊功能, 这些功能主要有:

- 双机容错, 安全可靠

由于 SFT III 采用了双服务器, 因此, 用户不必担心计算机故障, 可以放心地在网上存放重要的软件。这样也可消除因网络引起的停工而付出的昂贵的代价。

• 作为一个完整的 32 位操作系统的运行, 获得了 386, 286 的全部优点。

• 在一多客户机环境中能够共享文件应用, 打印机和其它的外设。

• 对 DOS, MS—Window, OS/2 或以后的版本以及 UnixWare 提供工作站支持。

- 提供快速隔离的解决问题的工具。

- 支持与任何平台小型机, 大中型机的连接。

- 支持各种网络硬件。

• 提供完整的后备服务和强有力网络管理功能, 包括远程管理。

- 支持磁盘的快速存取与可变迁的磁盘支持。

- 提供适当的位置和移植程序以利于更新。

• 提供一个没有危险的系统。NetWare SFT III 是一个具有模拟故障功能的软件产品, 它可模拟各种错误现象, 包括系统错误, 可观察系统是如何对各种故障进行响应的,

一般来说, 由于硬件错误是随机的, 故没有测试工具很难进行模拟的, 而 SFT III 可以靠容错能力提供了这种工具, 使用户对自身系统更加自信。

### 2.3. 硬件要求

- 二个不小于 25MHz 的 Intel386/486 服务器
- 二服务器具有等量的磁盘空间
- 每个服务器需一个网络适配器
- 显示器要求 EGA, CGA 或 VGA
- 镜像器连接(MSL) 通讯适配器及电缆
- 每个工作站需一个网络适配器
- 传输介质

这里需要说明的是服务器的 RAM 及磁盘容量取决于网络用户的多少。

### 2.4 软件要求

- 在每个服务器需装有 DOS3. 1 以上的版本
- NetWare SFT III 3. 11
- 镜像服务器连接(MSL) 驱动程序
- 双处理机(DP)驱动程序
- 网络驱动程序
- 磁盘驱动程序
- DOS 工作站要求 MS-DOS 2. X, 3. X, 4. X, 5. X 或

DR DOS 6. X

• OS/2 工作站: OS/2 标准编辑 2. 0, 2. 1; OS/2 扩充编辑 2. 0, 2. 1; OS/2 NetWare 客户机

- MS Windows 工作站: MS Windows 3. x

• UnixWare 工作站: UnixWare Personal edition 1. 0 或 1. 1; UnixApplication Server 1. 0 或 1. 1

### 3. NetWare v3. 11→NetWare SFT III 3. 11

目前, 正在运行的 NetWare v3. 11 比较多, 为了保护原先这些用户的投资, 可以使用升级的办法把 NetWare v3. 11 升级到 NetWare SFT III 3. 11。

#### 3.1. 准备 NetWare v3. 11 服务器

1、确保有一份所有服务器文件的最新备份。

2、装入 Install. 编辑 Autoexec. NCF 文件如下:

- 如 MSEngine 名代替文件服务器名
- 用 MSEngine IPX 内部网替代 IPX 内部网
- 删除文件中的其它行

3、从工作站上把 Autoexec. NCF 文件拷贝到 MSAUTO. NCF 上。

4、关闭服务器。

5、把下列文件拷贝到 DOS 分区。

- 从 SYSTEM—1 软盘上来的 MSERVER
- 从 SYSTEM—2 软盘上来的 INSTALL .NLM
- 从 SYSTEM—2 软盘上来的需要的磁盘驱动程序,

网络驱动程序及可装入模块

6、MSL 驱动程序拷贝到 DOS 分区。



# 高级 UNIX 连网技术讲座

## 第六讲 远程文件系统 RFS (三)

昂立自动化工程公司 冯家宁

### 6.11. 权限检查

RFS 有几层安全层,它与熟悉的 UNIX 相对应。安全主要是为了保护资源即文件,它包括标准 UNIX 文件的 `rw` 保护及进入 `/usr/group/other` 的范围。显然会有某些改变,因为 `/usr/group/other` 的用户标识号和组标识号都在另一节点上。新的改变在于每一节点的管理员都可指定用户标识号和组标识号与作为服务器的节点的用户标识号和组标识号的对应方式,如此确定远程安装的资源的使用权限。这种对应也在相反方向上完成以使远程用户看到权限的数字。这种对应是非常简单的,每一个远程用户对一

个在服务器上不存在用户。当然对应也可以是指定的。

在检查用户标识及组标识对应的上一层是对一客户请求安装一本地资源的检查。这一检查要求远程机器给出其口令,本地机(或代表它的域名字服务器)将此口令与机器增加到域时给域管理员的口令比较。本节后面再讨论这种检查的几个变体。

在上述两个检查之上就是 RFS 的一般结构,用 `rfmaster` 文件指定节点地址,名字服务器地址,基本名字服务器注册等。为使用任何远程资源,一个客户必须为名字服务器所认识并将只用其正式地址来联络。

7、在 DOS 分区上把 `START.NCF` 文件拷贝到 `IOSTART.NCF` 文件上。

8、编辑 `IOSTART.NCF` 文件,包括下列内容:

- IOEngine 名
- IOEngine IPX 内部网络号
- 为磁盘驱动程序装入指令
- 为网络驱动程序装入指令
- 为 MSL 驱动程序装入指令
- 早先需要的指令从第二步的 `Autoexec.NCF` 中删除老的服务器的 `IOSTART.NCF` 文件的例子:

```
ioengine name RIGHT-SFT 3
```

```
ioengine ipx internal net FABE 1945
```

```
load isadisk port=1f0 int=eload ne2000 port=1f0 int=2  
bind ipx to ne2000 net=fade2412load nmsl
```

9、`IOSTART.NCF` 文件拷贝到一软盘上,这样能把这文件放到新的 DOS 分区上。

### 3.2. 准备新的服务器

1、把下列 SFT III 文件拷贝到新的服务器的 DOS 分区:

- 从 `SYSTEM-1` 软盘上来的 `MSERVER.EXE`
- 从 `SYSTEM-2` 软盘上来的 `INSTALL.NLM`
- 必要的磁盘驱动程序,网络驱动程序及可装入模块

2、把 MSL 驱动程序拷贝到 DOS 分区。

3、把 `IOSTART.NCF` 文件拷贝到 DOS 分区。

4、编辑 `IOSTART.NCF` 文件如下:

- 用新的名替代 IOEngine 名

- 用新的号代替 IOEngine IPX 内部网络号

新的服务器硬件结构,当需要时,修改端口地址和中断号新的服务器 `IOSTART.NCF` 文件例子: `ioengine name LEFT-SFT 3ioengine ipx internal net FABE = 1946load isadisk port=1f0 int=eload ne2000 port=320 int=2bind ipx to ne2000 net=fade2412load nmsl`

### 3.3. 装入 NetWare SFT III

1、每个服务器根据 C:提示打入:

```
MSERVER <enter>
```

2、从老的服务器 IOEngine 控制台上打入:

```
ACTIVATE SERVER <enter>
```

这样装入 MEngine,自动地同步服务器,以前的 NetWare v3.11 服务器成为 SFT III 的主服务器。新的服务器还没有 NetWare 的分区,服务器不能镜像。

3、按 `<ALT>+<ESC>` 在主服务器上,根据 IOEngine 提示改变为 MEngine 提示。

4、从主服务器上的 MEngine 提示打入:

```
LOAD C:INSTALL <enter>
```

5、使用 `INSTALL` 中的 “DISKOPTIONS” 在新的服务器的硬盘上建立一个 NetWare 分区。警告:必须使 NetWare 分区在二个服务器上具有相同大小,否则服务器不能镜像。

6、使用 `INSTALL` 中的 “SYSTEMOPTIONS” 通过主服务器的软盘把 `SYSTEM` 和 `PUBLIC` 文件从 NetWare SFT III 拷贝到 MEngine 上。此时,已经完成了 NetWare v3.11 到 NetWare SFT III 3.11 的升级。

## 安装检查

当一服务器通过 `rfstart` 启动时,管理员不一定使用了 `-v`(检查)选项。

当安装一个服务器没有指定选项时,服务器上有 域名/口令 的节点就用此条目作检查,其它节点无需检查就可连接。

如服务器安装时指定了检查选项,那么只有在服务器上有 域名/口令 条目的节点才允许安装其资源。

注意是服务器上的 域名/口令 文件控制这些核查而不是基本名字服务器上的文件起上述作用。这些文件通常是一样的。本地的检查使服务器管理员的控制更为严格。如果某些节点从其 域名/口令 文件中删除,那么检查将使它们无法使用服务器资源。

如上述讨论,客户节点可能要求给自己提供口令。这口令由节点管理员在第一次启动 RFS 时提供并存在文件 `/usr/nserve/loc.passwd` 里给其它节点作检查用。如此文件损坏了,它可以在运行 `rfstop` 后删除并重新 `rfstart`。这时可以提供新的口令。

## 6.12. 授权与权限

这里主要讨论与文件 `/etc/passwd` 有关的事。

有一个 `domain/passwd` 文件,上面有每一主机的条目。这是在主机运行 `rfstart` 时建立的,它会拷贝到主机的 `loc.passwd`(`rfstart` 会提示这样做)。

当运行 `rfsetup` 时,它问域主机 `nserve` 要此口令。

以 `uid.rules` 和 `gid.rules` 为例说明权限的配对。如果想要用每台主机的名字,这可以从域主机的 `ahto.info/domain` 目录中得到。

如果想让其它成员使用自己的口令文件作为名字配对,可将它们拷贝到 `auth.info/domain/本机名/目录`,或把它们用邮件送给域管理员去做。特别是本机的 `root` 可以使用上面的目录,可以自己做 `cpio -p`。

配对规则如下:

- \* 如从不进行设置,所有远程的用户标识为 60001。
- \* 如在远程节点建立文件,规则会告诉远程用户这是 60001 建立的。
- \* 如本地用户建立文件,规则告诉远程用户这是 60002 建立的。
- \* 只有其它人能得到的我才能得到,因为我是 60001。
- \* 用户可以配对;透明意味着保持用户标识的一致。
- \* 可以单独地配对也可成组地配对到某一用户标识。
- \* 可以从设置中获得唯一的用户标识

## 6.13. 监听服务的要求

RFS 依赖于一个由选定的支撑块提供的 NLS 监听服务,因此运行 RFS 必然先运行支撑块。

当远程机器想安装一个资源,而这一资源在本地机上已发广告时,被监听程序分叉出来的服务器就被频繁地使用。RFS 的服务代码是 105,服务器程序是 `rfsetup`,不带参数运行。对于任何由监听程序分叉出来的服务器都可找到它的支撑块和导致其分叉的客户地址,在 `for` 或 `exec` 前由

服务器设定环境变量。

为确保支撑块的初始化:

1. 初始化支撑块的监听程序:

```
nlsadmin -i mqnet
```

2. 在支撑块上定义此主机的监听程序地址:

```
nlsadmin -l "mqmainserve"
```

3. 定义该支撑块的 RFS 服务器:

```
nlsadmin -a105 -c /usr/net/servers/rfs/rfsetup -y  
"rfsetup" mqnet
```

4. 启动监听程序:

```
nlsadmin -s mqnet
```

- 6.14. 运行与故障

发生故障有几种形式:

\* 域主机(名字服务器)故障。这不会影响现有的广告和安装,但不会有新的许可权和主机的出现。若有一台次级名字服务器,就不会有更多其它的事情发生。在基本机恢复后它会被告知发生过的情况。

\* 如果有一服务器故障,本地机的 `rfdaemon/rfuadmin` 会处理此事,频繁地重新安装,通知用户等。

\* 如果有一服务器要停机,本地机的 `rfdaemon/rfuadmin` 会处理此事。它们将执行 `fumount`,这相当于把一个磁盘卸下。

\* 如有一客户发生故障或停机,只是少一个要操心的顾客。

## 6.15. 恢复

有 `rfstart` 建立的内核恢复驻留监视程序处理发生各种故障时的内部恢复工作。此驻留程序并不直接与本地或远程的管理员打交道。

用户一级的恢复驻留程序叫 `rfdaemon`,它也是由 `rfstart` 启动的内核驻留程序,处理用户检测得到的事件。它从本地或远程的 RFS 或远程的用户(通常是 `root`)得到信息,它以适合于事件的参数运行 shell 程序 `rfadmin`。

如当前的机器是作为一资源 `resource` 的客户,那么事件会有:

\* `DISCONNECT`: 断接 `resource`。与资源服务器的断接由支撑块检测。

\* `FUMOUNT`: `fmount resource`。资源服务器的 RFS 内核检测 `fumount`。

\* `GETUMSG`。这可以是来自资源服务器的用户进程的任何信息,虽然目前只有 `fuwarn` 发送信息: `fuwarn resource time`。

\* `LASTUMSG`。此事件导致 `rfdaemon` 的死亡。它是非本地 停机时由 RFS 发送的。

在上述情形中(RFS 软件文本所提供的行为) `rfuadmin` 所要求的行为有:

\* `DISCONNECT`:

一用 `wall` 提醒所有用户 `resource` 不能在使用。

一用 `fuser` 找出所有 `resource` 的用户并 `kill` 他们。

一用 `umount` 拆卸所有 `resource`,以至内核可以清除其

结构。

一用 `rmount` 重装 `resource`。重复试图重装 `resource`，直至成功。

注意在此有状态系统与无状态系统有显著的不同。资源将被拆卸，用户程序对资源的调用将失败。当资源重装时，用户程序找出所有他们在资源关闭时有关的文件并把它们重新打开。对于无状态系统，有可能一直试图做一种操作直至成功。由于文件状态是由本地保存的，在远程机发生故障时它不会被丢失。

\* `FUMOUNT`：此动作与上相似。此时我们仍连接资源服务器，不同在于试图重新连接时。

\* `FUWARN`：这是发送“System going down in 60 seconds”信息。可能也发“RESOURCE will be umounted in 60 seconds”信息。

\* `rfuadmin` 不处理 `LASTMSG`。

上述活动可能导致一次强行的磁盘拆卸。对于远程的拆卸，并不会影响到数据的物理存储，而只是一次网络电缆的断接。

关于 `fuser` 有几点要注意的。首先它不找出打开本地文件的远程用户。这些用户一般由他们本机的 `fuser` 处理。其次 `fuser` 不能找出一项远程资源的所有本地用户。

## 6.16. 报告

普通的 UNIX 系统报告 `sar` 可用于报告 `RFS` 的活动，就象报告本地磁盘和缓冲区活动一样。

`sar` 的 `-Dc` 选项向本地系统调用请求有关网络的信息。

```
# sar -Dc
textsys textsys 3.11 i386 05/12/89
00:00:00 scall/s sread/s swrite/s fork/s exec/s rchar/swchar/s
10:00:00
in      7      2      4      0.00    770    680
out     6      4      2      0.00    450    600
local  180     43     18    0.91    23076  9814
10:20:00
in      0      0      0      0.00      0      0
out     0      0      0      0.00      0      0
local   4      1      1    0.05    336    203
10:40:00
in      0      0      0      0.00      0      0
out     0      0      0      0.00      0      0
local   6      2      2    0.94    691    365
11:00:00
in      4      1      2      0.00    376    322
out     3      2      1      0.00    178    202
local  132     30     11    0.78    1232  3822
Average
in      3      1      2      0.00    104     39
out     2      2      1      0.00     92     40
local   81     19      8    0.67    9812  988
```

这里 `scall` 是任意的系统调用，`sread`，`swrote`，`fork`，`exec` 是指定的系统调用，`rchar` 和 `wchar` 是由 `read` 和 `write` 处理的字符数。以此节点作服务器的调用用 `in` 描写，以此节点作客户的调用用 `out` 描写，本地调用（通常用 `sar -c` 报告）用 `local` 描写。`scalls` 的效率用 `rchars/s` 除以 `sread/s` 和 `wchars/s` 除以 `swrite/s` 来度量。这给出了平均读写量

并可指示是否要改变一个应用程序的缓冲策略或类似的调整是否值得。

如有本地的高速缓冲。外向的 `scall` 可以不要求与服务器通信。`scall` 的统计是对服务器的资源做的，`scall` 并不在服务器上执行。

远程 `exec` 显然很消耗资源，因为这要求把整个装入模块拷贝到客户机上。使用粘着位也许是合适的，但注意客户机粘着位的取消要到机器重新启动时才有效。

对 CPU、客户/服务器使用的统计可以显示资源是否被客户不适当地消耗着：

```
# sar -Du
textsys textsys 3.1 i386 05/12/89
00:00:00 %usr %sys local %sys remote %wio %idle
10:00:00 57 20 9 8 6
10:20:00 4 5 0 1 90
10:40:00 1 2 0 1 96
11:00:00 54 5 11 7 23
11:20:01 6 7 0 2 84
11:40:01 6 6 0 2 85
Average 21 10 3 2 64
```

细读这些数字可以发现应限制远程的使用，可以安装较少的资源或通过调整参数放慢对它们的使用。

下面的报告描述缓冲池的活动。`RFS` 的安装就象是一磁盘的安装。`RFS` 可能使用大量的名字服务。

```
# sar -Db
textsys textsys 3.1 i386 05/12/89
00:00:00 bread/s lread/s %rcache bwrite/s lwrit/s %wcache pread/s pwrit/s
10:00:00
local 5 72 92 2 8 77 0 0
remote 2 21 90 1 2 55
10:20:00
local 1 48 0 0 1 63 0 0
remote 0 0 0 0 0 0
10:40:00
local 1 5 83 1 1 60 0 0
remote 0 0 0 0 0 0
11:00:00
local 2 41 93 1 6 81 0 0
remote 1 12 91 1 3 64
Average
local 2 21 93 0 2 78 0 0
remote 1 7 87 1 1 61
```

`bread/s` 和 `bwrite/s` 代表缓冲区与磁盘，网络间的传输，`lread/s` 和 `lwrit/s` 代表缓冲区的使用，`%rcache` 为  $(1 - \text{bread}/\text{lread})$  的百分比。

对于只读的文件应尽可能进行高速缓冲。`%rcache` 由客户的“新”的读操作次数确定，因为每一数据块可能要服务器读一遍以上。在编辑或进行 `shell` 级的文件分页时，此统计数可能会较高，此时会一再读同一块数据。

在我们研究的例子里，可以预料在 `mcs.spool` 有较高的 `%rcache`，至少在读新闻时会这样。这不仅是在用户会在文章里前后翻页，而且用户会同时读同一篇文章。

对 `%wcache` 的解释需要作些思考，幸好对它的研究可以给我们一些有用的信息。象通常高速缓存那样，写操作总



是送到服务器, bwrite 条目意味着是“我写的, 没有作高速缓存东西”。较高的 %wcache 就是较频繁地向高速缓存写东西, 这样服务器不得不常常投掷 Cache 条目。

-Db 选项不报告 pread/s 和 pwrite/s。

加到 RFS 上有两个新的 sar 选项。-S 给出服务器活动的统计数字, -C 给出有关 cache 的浓度, 即服务器要通知客户放弃 cache 条目的频度。

```
# sar -S
textsys textsys 3.1 1 i386 05/12/89
00:00:00 serv/lo-hi request request server server
          3 - 6 %busy avg lgth %avail avg avail
10:00:00 6 100 .33 0 0
10:20:00 3 0 0 100 3
10:40:00 3 0 0 100 3
11:00:00 4 67 8 33 2
Average 3 0 0 100 3
```

这些数字会有助于内核参数 MINSERVE 和 MAXSERVE 的设定。但你也也许要咨询一下在排队论方面有训练的统计学家。serv/lo-hi 显示在一段时间内服务器运行的极端数字, 最低的数字将是 MINSERVE。%busy 统计显示由于服务器的忙一次请求要等待的次数。如果此数字高, 我们就没有足够的服务器。那么是否可以通过增加 MINSERVE 或 MAXSERVE 来改善呢?

如果等待的百分比是高的但服务器的平均数是低的, 这说明我们应增加 MINSERVE, 以使大量的服务器不断地有效。

如果等待的百分比高而服务器的平均数低, 我们应增加 MAXSERVE。这种等待是因为使用已超出 MINSERVE。

如果服务器几乎总是有效的, 我们就有过多的服务器。如果运行的服务器总数总是接近 MINSERVE, 那么降低 MAXSERVE 几乎不会有什么影响, 于是我们应降低 MINSERVE。如果可用的服务器数仍高, 我们降低 MAXSERVE。

```
# sar -C
musys musys 3.2 2 i386 05/12/89
00:00:00 snd-inv/s snd-msg/s rcv-inv/s rcv-msg/s dis-bread/s blk-inv/s
          0.6 1.1 0.2 1.5 0.0 0.0
10:00:00 0.0 0.0 0.0 0.0 0.0 0.0
10:20:00 0.0 0.0 0.0 0.0 0.0 0.0
10:40:00 0.0 0.0 0.0 0.0 0.0 0.0
11:00:00 0.3 0.6 0.0 0.6 0.0 0.1
Average 0.2 0.9 0.0 0.7 0.0 0.2
```

snd 表示本节点作服务器, snd-inv 表示对某一客户缓存的块发送投掷命令 (由于资源的改变) 的频度, snd-msg 表示向一客户写的频度。类似地 rcv 表示本节点作客户, rcv-inv 表示对一个块作不必要 (至少是无帮助) 的缓冲的频度, rcv-msg 表示从服务器收到的总的报文数。这对数字的比率告诉我们报文开销中被无效缓冲的比率。

为了解下面两个数字, 我们必须首先明白对于一项特定资源的 cache 什么时候是无效的。假定无效是在很短时间内定期发生的, 于是 cache 关闭直到某一时间过去 (内核参数值为 RCACHETIME)。

dis-bread/s 表示本节点作为客户, cache 关闭时, bread 调用的次数; 如果 cache 是打开的话, 这些数据应得到缓冲, 因而此数据表示我们由于 cache 自动关闭而受惩罚的次数。这一情况可通过增加 RACHETIME 参数而改善。

blk-inv/s 表示本节点作客户时无效发生的比率。

## 6.17. 统计

对客户一个字节数据在本地文件系统与安装的资源间的传送作统计是很有用的。fusage 给出这一报告的几种形式:

fusage 资源名

显示客户使用此资源的情况。

```
# fusage
FILE USAGE REPORT FOR textsys
/dev/dsk/0s1 / textsys 6936 KB
/dev/dsk/0s3 /usr textsys 4880 KB
/dev/dsk/0s4 /usr2 textsys 4930 KB
```

如果对某一客户的使用总大于对服务器的使用, 也许应把资源移到客户上。

df 命令对 RFS 作如下解释:

```
# df
/(dev/dsk/0s1): 36462 blocks 6700 i-nodes
/usr/(dev/dsk/0s3): 5552 blocks 4264 i-nodes
/usr2/(dev/dsk/0s4): 42026 blocks 9631 i-nodes
/news(MSCSNEWS): 29740 blocks 20850 i-nodes
/usr/lib/news(NEWSLIB): 7542 blocks 8832 i-nodes
/comptmp(COMPTMP): 57396 blocks 18910 i-nodes
/usr/mail/mscs(MSCSMAIL): 7542 blocks 8832 i-nodes
```

由于一项服务器资源通常只以一个目录表示, 并且不是本地安装点。远程的 df 系统调用执行包括资源的本地文件系统的 df。由于文件系统内有几项资源, 每一项都应列在 df 的输出中。

在我们的例子中, 资源 MSCSNEWS 和 MSCSMAIL 在服务器上是用 /usr/spool/news 和 /usr/spool/mail。它们都是文件系统 usr/spool 的子目录。

du

这只是一个普通的命令。它不象 df, 它只对目录而不是文件系统。当然如果在参数的树下有远程安装点的话, 输出是要作些解释的。

fmount

这命令也许是最有用的。它列出所有路径全名和当前安装的所有 (本地) 资源的名字 (包括客户的域名)。这可以显示当前没做广告的资源, 它们在 mount 时做过广告; 但随后被撤消。

fmount [-h] 资源

正如我们所料, 如在命令中不给出资源, 那么所有安装的资源 and 客户都将列出。-h 选项要求不印标题。(作者地址: 广州天河北路 84 号光华大厦侨顺楼 1804, 邮政编码: 510620, 电话: 7575216)

# IBM-PC 机的异步通讯

北京国家安全局 齐北魁

IBM-PC 机与外部的串行通讯是采用标准的 Rs232C 接口方式,由异步适配器完成。采用串行通讯线路的主要原因在于:降低了通讯线路的价格和简化通讯设备。

串行通讯中的控制规程很多,但 IBM-PC 机所采用的异步适配器核心是 INTEL8251,而 8251 仅支持同步(BISYNC)和异步(ASYN)两种方式。IBM-PC 机与外界的串行通讯一般都采用异步方式,因而本文也仅限于讨论异步通讯。

异步的含义是发送器和接收器不共享共同的同步信号,也不在数据中传送同步信号。它的定时方法是在一个字符的首尾放置起始符号和停止符号,供接收端用起始符号和停止符号判断一个字符。如图 1 所示。

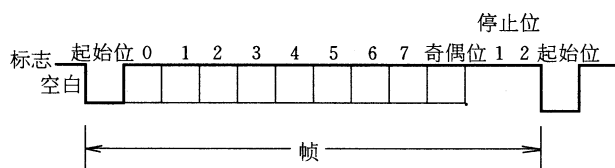


图 1 串行通讯帧格式

由上图可见,从起始位开始到停止位结束这一周期成为一帧。在停止位之后,如果有下一个字符要传送,发送器可以立即传送一个新的起始位。否则,发送器一直维持线路处在空闲标志状态。自最后一个停止位起,发送器可以在任一时刻传送一个新的起始位。也就是说从一个字符传送结束到下一个字符的传送开始,之间并没有一定的时间间隔。这些间隔称为空闲标志。通常用“1”表示。这些标志位与间隔位的多少均由串行电路根据通讯规程自动产生。

由上可见,通讯波特率的高低仅由空闲标志的多少决定。空闲位较多则意味着通讯波特率较低,反之则波特率较高。因此波特率的高低并不影响通讯的误码率。高波特率下的串行通讯与一般速度下的通讯是同样安全的。

IBM-PC 机异步串行通讯适配器的核心是 INT-EL8251 芯片。分析该芯片其大致特点为:

①通过读写“除数锁存器”可支持 50BPS~115,200BPS 的通讯速率。

②在线路上加 MODEM 可进行远程通讯。

③可提供 5,6,7,8 位不同位数的信息位。

④支持全双工通讯。

⑤具有全双缓冲机构,传输信息不需精确同步。

⑥起始位,停止位和奇偶的加入与取消可通过编程给予控制。

IBM-PC 机所采用的 INTEL8251 的上述特点提供了异步通讯的如下方式:

## 1、调用 BIOS 的通讯方式。

IBM-PC 机 ROM-BIOS 提供有异步 I/O 功能,可通过 INT 14 指令来调用。当要启动通讯端口进行数据传输时,要在 AL 中放好启动参数。如图 2 所示,并置 AH=0,调用 INT 14 指令,通讯状态返回 AL 中。如图 3、4 所示。传送字符时置 AH=1,调用 INT 14 指令,这个调用将设置 DTR(数据终端就绪)和 RTS(请求传送)两个信号。如果没有收到正确的 DSR(数据装置就绪)和 CTS(允许传送)等控制信号,则不能进行传送。在收到 DSR 和 CTS 之后,且 8251 的传送保持寄存器空,就把一个字符传给 8251。如果上述条件在某段时间内不能满足则建立超时状态,作为状态的一部分返回 AH 中。将 AH 的第 7 位置“1”。

7	6	5	4	3	2	1	0	
						0	0	代码长度 5 位
						1	1	6 位
						1	0	7 位
						1	1	8 位
						0	1	0=1
						1	1	1=1.5 如果代码长为 5 位
								=2 如果代码长为 6,7 或 8 位
								奇偶校验
								01 为奇校验
								11 为偶校验
								波特率
0	0	0	0	0	0	0	0	=110
0	0	0	0	1	0	0	0	=150
0	0	1	0	0	0	0	0	=300
0	1	1	0	0	0	0	0	=600
1	0	0	0	0	0	0	0	=1200
1	0	1	0	0	0	0	0	=2400
1	1	0	0	0	0	0	0	=4800
1	1	1	0	0	0	0	0	=9600

图 2 AL=启动参数

不论何时接收一个字符,都要设置 AH=2,调用 INT 14 指令,设置 DTR(数据终端就绪)控制信号,并且要一直

到收到正确的 DSR 信号之后,才能读取所接收到的字符。如果在一个预定的时间内,没有收到这个信号则 BIOS 就将超时错置入 AH 而返回;否则, BIOS 要一直在 8251 接收到字符并读取 AL 之后才把控制权交回,返回到 AH 的状态除了 5 位、6 位为“1”外,其余如图 3 所示。

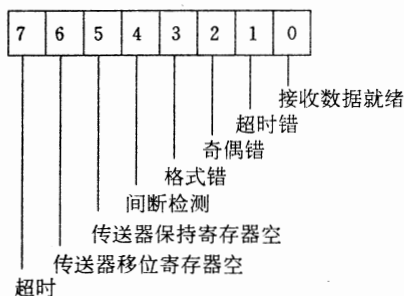


图 3

通讯线状态和调制解调的状态可以设置 AH=3, 调用 INT 14 指令来实现。AH 中返回的是串行通讯线的状态。如图 3 所示。AL 中返回的是调制解调的状态。如图 4。

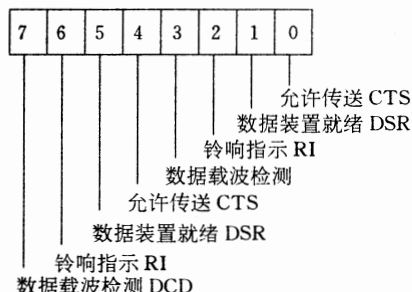


图 4

利用 INT 14 指令调用 BIOS 服务程序可以实现异步通讯,但这种通讯方式功能单一,有一定的局限性。正如图 2 所示它不能支持 9600 波特以上的通讯。

2、直接操作异步通讯控制器的通讯方式正如以上所述 BIOS 提供的中断服务不能支持高于 9600 波特的串行通讯,因此,要实现高波特率的通讯必须绕过中断 14H 直接操作 8251。

IBN-PC 机是 CPU 通过 I/O 指令操作 8251 的。MS-DOS 定义的 COM1 的基址为 3F8H, COM2 的基址为 2F8H,各端口地址及功能如下表。

串行口通讯的所有动作都是直接操作上述端口来实现的。其中,CPU 可读写的通讯线控制寄存器(LCR)用以指

定串行通讯的数据格式,也即用于控制通讯时的数据位,停止位的多少以及是否进行奇偶校验。CPU 可读写的线路状态寄存器(LSR)提供串行数据传送、接收时的状态。也即检测是否接收到数据,上一发送数据是否发送完毕。

当 LCR 的第 7 位为“0”时,3F8H 或 2F8H 则是作为串行数据读写端口使用,此时向其中写入字符将被发送出去,而读出的则是最近一次接收到的字符。但是此端口的读写必须参考 LSR 的当前状态。LSR 的第“0”位是“接收数据就绪”位。若此位为“1”则前一字符已经发送完毕。可以开始下一次传送。此时,向“数据读写端口”中写入的数据才会被正确送出。同样,当 LSR 的第 6 位(传送器移位寄存器空)为“1”时,从“数据读写端口”中读出的才是正确的接收信息。可见 LSR 查询方式通讯时必不可少的。

通讯的波特率是由除数锁存器中的波特率因子决定的。向其中写入不同的波特率因子就可以产生不同的波特率。见表 d。其中 MSB 是除数锁存器的高 8 位。LSB 是除数锁存器的低 8 位。由表 d 可以看出 MSB 和 LSB 取不同的值即可确定传输的波特率。如:MSB=09H, LSB=00H 则对应最低的 50BPS。当 MSB=00H, LSB=0CH 则对应 9600BPS。同样,若令 MSB=00H, LSB=01H 那么波特率则最高为 115,200BPS。

由于 8251 使用的频率为 1.8432MHZ 的基准时钟输入信号,所以需要用分频的办法产生所需要的波特率。8251 传送或接收数据时,使用的时钟信号频率是数据传送波特率的 16 倍。因此,除数锁存器的除数值可以用下式计算:除数=1.8432MHZ÷(16×波特率)。

MSB、LSB 寄存器填写的步骤如下:

可见这种形式下的异步通讯突破了调用 BIOS 的通讯方式的某些局限,大大提高了信息的传输速率。

3、中断方式控制下的异步通讯异步通讯时,信息的发出是由 CPU 控制的。但信息的接收 CPU 则无法预测信息的到达。为保证计算机正确及时的接收每一个信息,那么 CPU 就需不断的查询 LSR(通信线状态寄存器)的第 0 位(接收数据就绪)。显而易见这种方式 CPU 的很多时间处于无效等待状态。为了解决这一不合理状态而采用中断方式进行异步通讯。在此方式下中断允许寄存器(IE)被置“1”,当有效字符到达时,8251 自动发出中断申请。请求 CPU 处理。对于到达 COM1 的字符 8251 发出 IRQ4 请求,触发 OCH 号硬件中断对于到达 COM2 的字符则发出 IRQ3 的请求,触发 OBH 号硬件中断。用户只需在 OCH 和 OBH 中断向量中放入自己的程序的起始地址就可进入自己的中断处理程序。就可响应 8251 的中断申请。

在中断服务程序中用户可定义自己的缓冲区,每当数据到达时,中断服务程序被激活,程序从“数据读写端口”中读入接收数据,并将其存入缓冲区。而后修改缓冲区的有关标志,清中断,返回主控程序。而后用户再进入缓冲区的数据进行处理。这种中断方式的异步通讯十分合理方便。

**万利科技广场**

广州市国际路 25 号(中央酒店侧)

## M-1724 打印机常见故障排除三例

湖南桑植县 王琳峰

### 例一：

**故障现象：**接通电源后，打印机面板指示灯一闪即灭，第二次开启打印机电源时，打印机面板指示灯再无任何反应。

**现象分析及维修：**根据上述故障现象可判断故障在打印机的电源部分。拆开打印机用万用表测打印机电源变压器初级绕组，结果呈开路状态。说明 M-1724 打印机电源变压器初级绕组上所串的一个热敏保险已熔断。维修时需拆开电源变压器修复。首先从机架上卸下变压器，然后用刀顺着图 1 所示的虚线部分划开绝缘青壳纸及聚脂薄膜层（注意：千万不要划破变压器的漆包线，否则将难以恢复）。待变压器初、次级绕组线全部露出后，可以看到所有引线抽头是从一些绝缘层中抽出的，剥开变压器反面（正面青壳纸上印有参数，反面无任何文字）的两绕组间的绝缘腊管，该管内包的就是热敏保险。上面印有 220V、130℃ 等字样，用万用表测其两端阻值已呈开路，想找一个同样的热敏保险替换是比较困难的。应急办法有两个：一是把热敏保险原来位置直接短接；二是用一只 3.15A 的保险丝管接上。重新包上绝缘青壳纸，装上金属架，恢复变压器原来的位置。

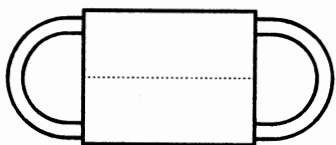


图 1 电源变压器外形示意图

### 例二：

**故障现象：**打印机装有打印纸，开机后打印头复位，但面板 ON LINE 灯不亮，CHECK 灯亮，按亮 ON LINE 联机打印一行后，ONLINE 灯灭，CHECK 灯亮，打印停止。

**现象分析及维修：**根据上述现象可知，这是 M-1724 打印机常发生故障之一，即纸尽检测光电耦合器造成的。

M-1724 打印机字辊下装有纸尽检测光电耦合器。其中的发光二极管与光敏三极管是装在同一侧的，工作时是利用打印机对纸的反射作用来决定光敏三极管的导通，装有打印纸时，光被反射，光敏三极管导通；不装打印纸时，光无法反射，光敏三极管不导通。在开机复位后打印机要对打印纸的安装与否进行检测，有打印纸时面板上 ON LINE 灯亮，CHECK 灯灭，打印机进入联机打印准备就绪状态；

无打印纸时面板上 ON LINE 灯灭，CHECK 灯亮，打印机进入脱机状态。联机打印时打印完一行后打印机也要对打印纸的有与无进行检测监视。

如图 2，纸尽检测电路图中，当有打印纸时电压比较器 A1 第 7 脚送出一个低电平给 M50734，否则送出一个高电平。本例是打印机装有打印纸，但检测结果无纸，显然这是错误的。在纸尽检测电路中纸尽检测光电耦合器造成故障的情况较多，通常是光电耦合器界面的灰尘过多所致。

拆下打印机的机械部分，翻过来放置于工作台上，在靠近左边的地方可以看见一黑色护套线；套内有棕、绿、蓝、黄四色线，就是纸尽检测光电耦合器的信号线。用手握住套管慢慢抽出光电耦合器，然后用无水酒精棉球擦净光电耦合器表面，重新安装后故障排除。

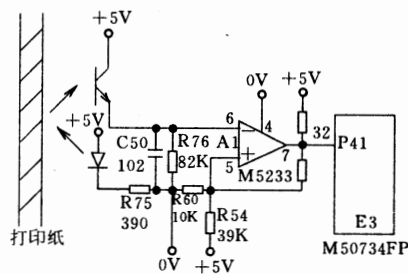


图 2 纸尽检测电路图

### 例三：

**故障现象：**打印机加电后字车能复位，但在初始位置处字车抖动不停。现象分析与维修：根据上述故障现象可知打印机初始位置检测电路有问题。M-1724 打印机初始位置检测元件采用的是机械式开关，并采用弹簧片式触键。由于长时间的机械动作，弹簧片会发生弹性形变以及两簧片之间会有灰尘介入，于是两簧片在字车挡板碰撞时，并不能正常接触，导致 CPU 控制字车一直左移直到初始位置为止，无休止的左移就会使字车在初始化位置抖动。解决办法是对检测开关进行调整、清洁。

282

北大方正系列产品代理

**广州市大恒科技**

地址：广州市越秀北路 133 路二楼 电话：3327850

## AR—3240 打印机 断针的修复

建设银行开平市支行 张春新

在针式打印机系列中,AR—3240 打印机以其较高的性能价格比,得到计算机应用部门较广泛的应用,该机虽已停止生产,但在用户中仍有相当的数量在使用。由于使用年深日久,难免会出现打印针磨损的现象,使得打印的字符缺笔少划,在打印表格较多的场合,个别打印针的磨损程度更为严重,造成打印质量下降。当然,更换打印头是一种快速修复的途径,但是,目前一个打印头的零售价高达一千七、八百元,相当于整机价格的三分之一还多,实在很不合算。而更换个别短缺的“断”针,则是一种既可行又经济的办法。现将换针的方法和步骤介绍如下,以供同行参考。

### 一、检查打印针的短缺情况

- 1、关闭打印机电源;
- 2、取下色带架;
- 3、从打印机内拔下打印头扁平电缆;
- 4、将字车上固定打印头的两个螺丝卸下,手持打印头,向上用力,将打印头从字车上取下;
- 5、用放大镜观察检查两排打印针的出针状态,将会发现一些针有短缺现象。记下短缺针的位置(一般可将左右两排的各 12 根针从上至下排序为 1、2...12)。

### 二、更换打印针

- 1、让出针孔向上,沿打印头周边卸下固定针座的 20 个螺丝(注意,图示的 A1、A2 两个螺丝不要拧动,以保持针架和针柄定位槽板的整体性);
- 2、用“—”字起子插入永磁体(线圈)与针座间的缝隙中,轻轻用力将针座与永磁体分离;
- 3、根据记下短缺针的位置,对应找到其针柄(如图示,出针孔的针 1 对应针柄 1,针 1' 对应针柄 1'...)。卸下固定针柄螺丝,用镊子夹住针柄,轻轻拉动该针,同时观察出针孔,以确认此针柄就是短缺的针根;
- 4、拔出短缺针、换上一根新针,拧紧针柄固定螺丝。这里要特别注意,因为每一根针要穿越针架圆形定位孔、椭圆形定位孔、垂直定位孔和出针孔四道“关”,所以穿针时一定要耐心、细致。如果要更换多根短缺针,最好是逐根进行,换好一根再换另一根,不要一次将全部短缺针拔出,否则,穿新针时容易插错定位孔,增加操作难度;
- 5、换完针后,用放大镜观察出针孔、检查 24 根针排列是否整齐划一。如果发现新换上的针突出较多,可用细油石加一点机油小心打磨,直到各针长短一致、排列整齐为止;
- 6、用无水酒精将全部针清洗一遍,晾干后注几滴润滑

油;

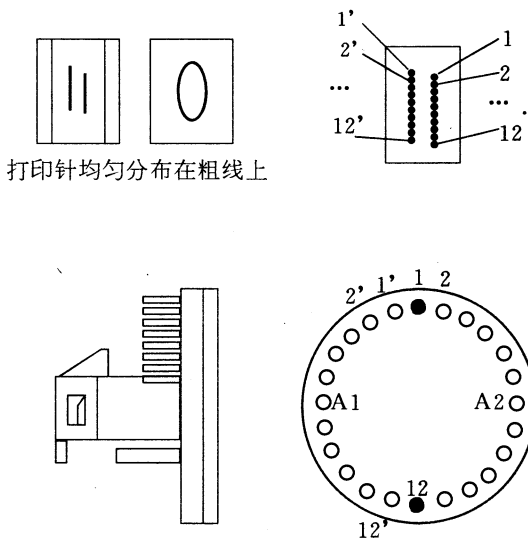
7、将针座与永磁体复位。安装时要注意针座上的两个定位销应与永磁体上的两个定位孔吻合。否则,要重新安装;

8、装回 20 个固定针座螺丝,打印头整体复位。最后按照拆卸打印有的反顺序,将打印头及扁平电缆复原即告成功。

### 三、几点体会

- 1、更换打印针是一件精细工作,穿针力度要适当、轻巧,尤其是穿最后一道出针孔时,可以用磨尖的钢针插入两排针的间隙,适当拨动,使新针顺利出孔;
- 2、整个穿针过程最好借助放大镜、手电筒进行,反复观察针的走向是否正确;
- 3、如果必须重穿 24 根针时,则工作较为艰巨,更需要耐心、细致。据笔者经验,最好先按顺序号穿好一排,然后再穿另一排。在穿越椭圆定位孔时,不要跨孔斜穿,一定要按顺序一针一孔排列,每穿好一根,就用螺丝将针柄固定,全部针穿完后,认真检查一遍,确认无误,便可复原;
- 4、在日常使用中,一定要保持打印机清洁干净。当出现卡纸时,应先关机,退出打印头(即将打印头调节杆拨离与卷纸胶棍最大距离),小心排除卡纸后,再将打印头调回适当位置使用;当出现色带变形或断裂时,要及时换色带,保证打印机处于良好的工作状态,延长使用寿命。

附图



打印针均匀分布在粗线上

# 妙笔生花

## ——蒙恬中英文手写辨识系统

电脑的发明和应用对人类社会的发展产生了巨大的影响和推进作用。在我国，近年来随着办公室自动化的兴起，电脑网络的普及，要求每个人都必须是电脑的使用者，而不能只把电脑作为打字员的打字工具，我们每个工作人员都应该能够方便熟练地使用电脑来提高工作质量和工作效率。但在我国普及电脑遇到的最大问题就是汉字的输入，各种汉字编码输入方法无疑对电脑的中文化起了很大的作用，但终究是需要花大量时间和精力专门学习，对于广大管理人员来说，要记住那些拆字组码方法，真是“目瞪口呆”。几十年来，人们一直期望能有一种方便易用的方法来

输入汉字，操作电脑，最简单而又符合人们习惯的莫过于直接用笔书写输入、绘图、甚至控制整个电脑环境。这也是多年来各国科研机构专家学者投入大量人力和物力研究的课题——中英文手写辨识输入即笔式输入。

手写辨识研究起源于五十年代，开始是对英文和数字进行辨识，一直没有重大突破，到八十年代，随着 CPU 处理能力的增强和辨识算法的改善，文字辨识研究也有了快速发展，九十年代始进入产品化阶段。1991 年台湾蒙恬科技推出第一套中国人自行开发的中英文手写辨识系统——蒙恬中国笔，深获海内外华人的支持与使用，此后软件不

### 单色显示器

#### ASCII 码抖动的维修经验

哈尔滨建筑大学 董礼文

用户从美国买了一台型号为 AdvanTech 的 286 主机，由于经费不足，显示器买的是型号为 SAMSUG 的单色显示器，SAMSUG 单色显示器要求用户采用 120V, 60HZ 的交流电。回国后，用户找有关的部门进行改造，改造后出现以下的现象：

故障现象：只打开单色显示器的电源，屏幕的光栅呈黄色，有扫描线，扫描线的间距上密下疏，光栅的两边不能充满屏幕。再打开主机的电源，屏幕显示的 ASCII 码上下抖动（ASCII 码上下抖动的范围是两个 ASCII 码大小的距离，频率 1 秒左右），行与行之间的 ASCII 码大小不同。

故障的分析和维修：由于修理显示器的部门，只将 SAMSUG 显示器的变压器进行改造，未对其它器件进行调整（用户反映），这样的改造有可能出现这些情况：电源部

分的桥整流二极管部分损坏；电源部分的滤波电容不匹配；帧频过高（两方国家显示器的帧频为 60HZ，我国显示器帧频为 50HZ）。

先检查电源部分：在电源的滤波电容上并联一个 0.01HF 的电容器，电源输出的直流电压来发生改变，ASCII 码抖动的现象并未消除，说明滤波电容的性能良好。用示波器检查输出电源电压的波形，排除桥式整流电路中有一只二极管损坏，而使 ASCII 码有上下抖动的可能性。

对有关的电位器进行调节：只调节帧电位器，ASCII 码抖动现象有所减轻（但抖动现象并未消除），ASCII 码抖动的频率降低，幅度未变。同时调节帧频电位器、帧幅电位器、行幅电位器，ASCII 码上下抖动的现象消除，行与行之间的 ASCII 码大小相同。关掉主机电源观察显示器，显示器的光栅充满屏幕，扫描线的间距相同。重新打开主机电源，工作十二小时，ASCII 码未发生抖动现象。三个多月过去了（用户反映）工作良好。

通过修理得出以下经验：

对国外的单色显示器的改造，不仅要换变压器（用国产单色计算机显示器的变压器取代西方国家供本国使用的显示器的变压器），而且要调整帧频电位器、帧幅电位器、行幅电位器，还需检查电源部分的滤波电容、桥式整流电路中的二极管是否匹配，以防隐患。



断改进和完善,与多家硬件厂家配合推出多种手写 OEM 产品,其软件被美国、日本多家世界著名公司采用,至 93 年蒙恬科技以其最新研究成果推出自有品牌产品“蒙恬笔式环境”,为了与其 OEM 产品作市场区隔,“蒙恬笔式环境”不论在软件和硬件上都有很大程度的改善,这一产品的推出标志着真正实用成熟的笔式产品的诞生,也标志着电脑笔输入时代的来临,彻底结束了几十年来人与电脑的隔阂,人们从此可以用一支驾轻就熟的笔轻松自在地写入汉字、绘图、以及控制整个电脑环境。

笔式环境的推出,备受广大用户的青睐。该产品在我国的推广应用必将对我国的计算机普及和现代化建设产生巨大的作用。“蒙恬笔式环境”1994 年 2 月被《PC WORLD》评为中文手写产品第一名。在五月份举行的北京电脑交易会上,众“笔”云集,“蒙恬笔式环境”又以其先进的技术、完善的性能独领风骚。

“蒙恬笔式环境”是一套完全以笔为设计中心的软件系统,使用者只需一支笔就可以完全驾驭电脑。可直接书写辨识国标二级汉字 6763 个(繁体 13052 个)、英文数字符号等,除辨识率高外,还可自动学习使用者笔迹,不论行书、草书、甚至速记简写字,都可使辨识效率达到百分之百。还可以在电脑文稿中加上眉批、注角、手迹签名等,并提供有多种与人书写习惯相似的笔式符号,直接用笔画出符号就可以进行文字删改、编辑、排版及命令操作,还可以用笔作为鼠标使用,特别对于图形绘制和平面设计人员来说,用笔的感觉真是太美妙了,与在纸上画图一样方便、准确,确是鼠标所万万不可及的。

### 一、蒙恬笔式环境产品配置与应用

**硬件:** 6X6 感应式数位板及书写笔,25 至 9 芯串口转接头大小键盘连线各一组。

**软件:** DOSPEN (DOS 环境)

WINPEN & INKPEN (WINDOWS 环境)

UTILITY (辅助驱动程序)。

**系统需求:**

DOSPEN: PC 286 或以上,2MB RAM

WINPEN & INKPEN: PC 386 或以上,4MB RAM (建议)

**连接:** 无插卡,直接通过串口与电脑相连,可配备于任意微机或笔记本电脑

**应用:** 在中文系统级提供支持,驱动程序驻留内存,可完全取代键盘和鼠标,现有软件不需做任何修改即可用笔写字、画图、替代鼠标和键盘操作。

## 电脑股市接收分析卡

总经销:广州市全通计算机公司 电话:7752397 7766887

**DOS 环境:** 支持金山,中国龙,超想,UCDOS,213,倚天/震汉简体版,AUTOCAD……等系统,及其上应用软件 WINDOWS 环境:支持微软中文 WINDOWS,中文之星……等,及其上所有软件

### 二、DOSPEN

1. 全屏幕书写:可全屏幕书写,不受大小位置限制,并可书写笔式符号取代键盘上任何按键。

2. 全屏幕编辑:利用笔同时具有书写及鼠标功能,可全方位编辑文稿。。

3. 蒙恬书写盘:提供四个方格,可边写边认,连续书写,无需等待

4. 特殊符号表:提供各种特殊符号,包含图形、数学、表格、及综合符号并可自定义符号。

5. 书写环境设定:可在线设定,调整书写速度、连笔程度、笔迹颜色等。

6. 书写学习功能:可在线学习记忆各种书写习惯,辨识行书、草书、速记简写字等。

7. 书写辅助方式:提供联想输入、词组输入和候选字功能。

8. 其它输入法:提供部首笔划检索和拼音点取输入,并可外挂各种输入方法。

9. 全功能键盘:提供软键盘可直接控制各种应用软件

10. 词库设定:可将整段词句输入词库,点取使用,避免重复繁琐书写

### 三、WINPEN

**包括:**蒙恬书写盘,蒙恬手写作家,蒙恬学习对话盒,蒙恬学习浏览字库,蒙恬笔势群,蒙恬全能键盘,蒙恬特殊符号表,蒙恬小字典,蒙恬词库设定,等功能是唯一在 WINDOWS 下真正做到连续书写,边写边认的多工手写系统。提供一个集合书写编辑和排版的全能环境,具有全屏幕多格书写稿纸,供大量文字书写用;提供各种笔势符号,并可自行定义,直接控制整个 WINDOWS 环境;提供智慧型学习功能,并可与 DOS 下的学习字库合并;书写盘提供两个方格可连续书写,可与任何应用软件搭配使用,不论文字编辑还是图形处理,都可以随心所欲,达到最佳效果。

### 四、INKPEN

可直接在电脑文稿中加上眉批、注角或留下手写签名。将任意电脑变成笔式电脑,将任意软件变成笔式软件,让任何人都可以使用电脑。

正是“妙笔生花活蒙恬,笔式环境调至尊”。

本刊经营“蒙恬笔式环境”手写识别系统,有兴趣的读者,请与本刊联系。(地址:广州市天河科技东街 49 号,邮政编码:510630,电话:(020)5514304

# 上下 文相关联机帮助机制 在 Foxpro 上的实现

湖南双峰工商银行 罗辉 陈又来

**【摘要】**本文详细地介绍了 FOXPRO 数据库软件的联机帮助机制,并提供了几种上下文相关联机帮助的程序设计技巧。

Foxpro 是 90 年代推出的 FOXBASE+ 换代产品。到目前,它已有三代产品。较之 FOXBASE,它在功能、用户接口、图形窗口、跟踪调试及速度等许多方面都有彻底的改进和全新的完善,使传统的数据库软件业者耳目一新,振奋异常。除此之外,FOXPRO 还提供了强大的帮助机制,可通过编程以多种方式给用户提供尽可能多的帮助信息,这也是它较之 FOXBASE 的一个重大改进。本文基于 FOXPRO 2.0/DOS 版本,对其上下文相关联机帮助机制的设计加以讨论。

## 一、联机帮助(HELP)手册

### 1、FOXPRO 系统本身的联机帮助手册

正如 FOXBASE 一样,FOXPRO 提供一个十分有用的微型参考指南(Minireference guide),每当你想获得有关命令、函数或其它信息时,就可以直接向系统求助,这是检查语法、缺省值、功能及察看使用例子最简单的方法,除了正在对话、或 SET 选择项 HELP 被置为 OFF 之外,任何时候都可以引用 HELP。

①、引用联机 HELP。可通过如下三种方式获得帮助信息:从 SYSTEM 弹出菜单中选择 HELP;按帮助热键 F1 (亦可另外设置);从命令窗口键入 HELP 命令。

②、HELP 窗口。激活帮助后,系统弹出 HELP 窗口。HELP 窗口是一个标准的窗口,可被卷动、关闭、移动或改变尺寸,从而允许在你编写或调试运行程序时始终保留 HELP 窗口在屏幕上。HELP 窗口有两类信息:帮助主题(TOPIC)、帮助细节(DETAIL)。激活 HELP 后,首先显示 HELP 主题信息,从中选择某一需要帮助的主题后,按 ENTER 键,HELP 窗口就显示该主题的帮助细节内容。

③、帮助信息的汉化或重写。与 FOXBASE 不同的是,FOXPRO 的帮助信息的汉化或重新改写相当容易。这是因为,FOXPRO 提供的帮助信息实际上都是以记录的方式存储在 HELP 数据库文件中,系统默认的联机帮助数据库

是 FOXHELP.DBF。可以用常规的数据库记录操作命令对帮助库进行编辑修改,也可以通过命令:

SET HELP TO <帮助文件名>✓

使系统打开另一个帮助库提供其它帮助。如果要汉化系统的帮助信息,可以建立编辑一个 FOXHELP.DBF 的汉化库,并用上面命令打开它,HELP 机制将根据该库对你提供汉化的帮助信息。

④、HELP 数据库。任何数据库,如果它至少包含两个字段、而且第一个字段是字符型字段(帮助主题)、第二个字段是备注型字段(帮助细节),它即可作为 HELP 库使用。除此之外,为提供更为贴切的帮助信息以增强可用性,也可以在 HELP 数据库中任意增加字段,正如后面的例子库 USERHELP.DBF。

⑤、屏蔽 HELP 机制。如果不希望提供联机 HELP,可在命令窗口中键入 SET HELP OFF 或在 VIEW 窗口的 ON/OFF 面板上置 HELP 为 OFF 即可。

### 2、应用软件的简易联机帮助手册设计

如系统提供联机帮助手册一样,也可以为应用软件提供联机帮助手册。首先分析整理本软件须提供的帮助信息为帮助主题和详细帮助细节两部分,录入用户的帮助库(如 USERHELP.DBF)。然后在你的应用软件开始加入如下语句行:SET HELP TO USERHELP 安装用户软件联机帮助手册;在应用软件的出口处加入下面语句行:SET HELP TO FOXHELP 重新恢复系统联机帮助手册。那么在软件运行过程中随时可按 F1 激活 HELP 窗口,通过选择合适的主题即可以获得相应的帮助信息。

**AOK 彩显**

**王者之尊 傲视同侪**

广利电脑设备厂 电话:8895924,8895934 传真:8895943

## 二、应用软件中实时帮助信息的设计

联机帮助手册为我们提供了整个软件本身的帮助信息,为获得与程序运行当前操作有关的帮助,必须在此手册上翻阅找到该帮助主题后,才能得到相应的帮助细节。帮助信息不能精确定位,操作也繁琐。有时我们更希望在我们设计的应用软件中提供帮助信息,以便于在程序运行的任意时刻都可获得与正在运行的程序上下文相关的帮助信息。

利用 FOXPRO 众多命令本身可携带帮助信息的功能,可在程序中直接嵌入帮助信息。可携带帮助信息的命令主要有如下:

① @...SAY/GET ... MESSAGE '帮助信息'...

用于在格式化输入/输出各编辑字段的信息提示场合。

② @...PROMPT ... MESSAGE '帮助信息'...

用于光棒菜单上各菜单项的信息显示。

③ DEFINE MENU ... MESSAGE '帮助信息'...

用于对棒式菜单的菜单名提供帮助信息的场合。

④ DEFINE POPUP ... MESSAGE '帮助信息'...

用于对弹出式菜单名提供帮助信息。

⑤ MENU BAR <二维数组>,<数值表达式>

用于下拉式菜单中菜单棒各选择项的帮助信息提示。其中二维数组的(i,1)中内容为菜单名,(i,2)内容为相应菜单项的帮助信息。

上述命令的帮助信息提示位置由 set message to <行数>决定,即显示在行数所指定的那一行上。

用这些命令将可以提供密切的实时帮助信息,光标在不同域上移动时,相应的帮助信息自动在信息行显示,勿需热键激活。但所提供的信息十分有限(最多不能超过一行),其修改也不太方便(必须直接修改程序)。

## 三、上下文相关联机帮助的设计

FOXPRO 也提供了一些利用帮助库或有助于进行程序上下文相关的联机帮助设计的命令。其提供两条在程序中直接控制帮助主题选择的命令:

SET TOPIC TO <EXPC>

和 SET TOPIC TO <EXPL>

通过它们即可在程序运行中激活 HELP 后自动选择相应的帮助主题并直接将其帮助细节内容显示在 HELP 窗口上。用这种方式提供上下文帮助须用 ON KEY=<热键值> HELP 命令设置 HELP 陷阱,而且只对执行 READ 命令时有效,但 GET 命令中的变量既可以是某数据库字段名也可以是内存变量名。对其它全屏编辑命令如 browse、edit、change 等,仍只提供简易软件帮助手册的

帮助方式,并且它们使用的热键由 SET FUNCTION <热键值> TO HELP 设定,而不是由 ON KEY 设定,如果 ON KEY 在 SET FUNCTION TO HELP 命令之后设定而且设定的是同一个热键,那么前者将会屏蔽掉后者的作用,也就是说其后的非 READ 全屏编辑命令中将无法用该热键激活 HELP 窗口获得帮助。编程时必须注意这一点。

### 1、SET TOPIC TO <EXPC>命令方式。

SET TOPIC <EXPC>命令是指定 HELP 主题的最简单方式,在激活 HELP 后,系统首先计算字符型表达式 <EXPC>,然后在由 SET HELP TO 指定的用户帮助库中查找第一个字段(帮助主题)内容与该表达式值相匹配的记录,尤如用 LOCATE 命令定位记录一样。匹配后即 HELP 窗口显示第二个字段(帮助细节)的内容。用这种方式必须在程序中需要不同帮助时不断地用 SET TOPIC TO 修改主题,以便提供正确的帮助信息。

假设用户帮助库 USERHELP 中有如下记录,其中备注型字段 HELPTTEXT 中为相应主题 TOPIC 的帮助信息:

字段名:TOPIC	HELPTTEXT	FLDNAME	PRGNAME
NAMEHELP	memo	NAME	TMP1
MONEYHELP	meno	MONEY	TMP1
MEMOVARHELP	meno	MEMOVAR	TMP1

且有一个用户数据库 userdata,包括两个字段:NAME、MONEY。则运行下面的小程序:

C>TYPE TMP.PRG

```
set talk off
set help to userhelp && 指向你的应用程序帮助库
set function 1 to 'help' && 设置 F1 键为 BROWSE 时的 HELP 热键
use userdata
```

① brow

```
on key = 315 help && 设置 F1 键为 READ 时的 HELP 热键
memovar = 0 && memovar 是一个内存变量
set topic to 'NAMEHELP'
```

```
@ 0,20 say "领域一的值:" get NAME
read
```

```
set topic to 'MONEYHELP'
```

```
@ 2,20 say "领域二的值:" get MONEY
read
```

```
set topic to 'MEMOVARHELP'
```

```
@ 4,20 say "变量三的值:" get memovar pict '99999'
```

```
read
set help to c:\oxhelp && 恢复系统帮助库
go top
```

③ ON KEY && 开放 F1 键为 BROWSE 时的 HELP 热键

② brow  
use

这样,当你在①处按了 F1 键,可翻阅该系统的联机帮助手册;在②处,如果没有③处的 ON KEY 命令取消 READ 的 HELP 热键设置,按 F1 键就会毫无反应了;而在 READ 期间,按下 F1 键,此时光标所在域相应的帮助细节将在 HELP 窗口直接显示。

从上面的小程序可以看出,程序中必须含有许多的

# 宏图电脑

广州市文德路 84 号 邮码:510030 电话:3325843

SET TOPIC TO 命令,而且不能对同一个 READ 命令内的多个 GET 命令分别提供实时帮助,如果要分别提供帮助则必须用多个 READ 命令——将各个 GET 语句分开,不好进行全屏幕设计,程序冗余量大,不易维护。只适用于小型应用软件中。

## 2、SET TOPIC TO <EXPL>命令方式。

逻辑表达式 EXPL 首先被计算,然后提取使该表达式为真的用户帮助库第一个记录中的帮助细节内容在 HELP 窗口显示出来。它提供了一种更为通用的设计方式。因为 EXPL 能包括本身属于帮助库的字段名,所以在帮助库中设置其它辅助字段将变得十分有用。可以把任何类型的标识信息放在其中,作为帮助主题的搜索关键字搜寻相应的帮助细节,更有利于上下文相关帮助的设计。

仍以上面的用户帮助库 USERHELP 和数据库 USERDATA 为例,执行下面程序:

C>TYPE TMP.PRG

```
set talk off
set help to userhelp
set function 1 to 'help'
use userdata
browse
memovar=0
on key=315 help
set topic to FLDNAME=SYS(18)
@0,20 say "领域一的值:" get NAME
@2,20 say "领域二的值:" get MONEY
@4,20 say "变量三的值:" get memovar pict '99999'
read
on key
set help to c:oxhelp
use
```

注:因为 SYS(18)返回的 GET 域变量名都是大写,因此帮助库的 FLDNAME 值也都应大写。

执行上述程序,在 READ 命令执行期间如果你正在输入值,比如输入 NAME 的值,并且你按了 F1 键,那么将弹出 HELP 窗口显示主题 NAMEHELP 的细节内容。而如果上面设置主题的语句换成: set topic to fldname=sys(18).and. prgname=sys(16),则激活帮助时,将在帮助库中寻找 FLDNAME 和 PRGNAME 与此时编辑的 GET 变量名及正在运行的程序名相匹配的帮助主题,并将帮助细节显示出来。

因为可以在帮助库中添加任意多的字段,还可用任何形式的表达式来选择帮助主题,因而你可以设计出许多丰富得多的帮助机制来。

## 3、非标准 HELP 窗口的帮助方式。

上面各种方式设计的帮助,都是将帮助信息在标准的 HELP 窗口显示的,其在多样性、灵活性等方面要受到一定的限制,如果你想设计有自己特色(譬如对颜色、窗口大小、位置等有更灵活的控制)的 HELP 方式,可参考下面仍以 USERHELP 和 USERDATA 为例的设计实例。对 HELP 实施不同的控制,将获得不同的效果。

C>TYPE tmp1.prg

```
set talk off
set proc to tmp1
select 10
use userhelp
on key=315
do helpproc with varread(),
PROGRAM()
select 1
use userdata
memovar=0
do while .not. eof()
@ 10,20 say "领域一的值:" get NAME
@ 2,20 say "领域二的值:" get MONEY
@ 14,20 say "变量三的值:" get memovar pict '99999'
read
skip
enddo
on key
clos data
***** 帮助子过程 *****
proc helpproc
para fldname,programname
set compatible off
selectno=str(select(),2) && 保存当前工作区号
select 10
locate for ((fldname = fldname) . and. (programname = prgname))
define window help from 0,0 to 10,30 panel shadow float grow title ;
fldname+' HELP' color scheme 5 && 开自定义的 HELP 窗口
activate window help
? helptext && 在 HELP 窗口上显示帮助细节内容
wait ' '
select &.selectno.
release window help && 关闭 HELP 窗口
return
```

本设计方法需显式地打开帮助库并进行主题定位,因而在应用程序设计时必须约定:不能用 SET HELP TO 命令隐式打开该帮助库,也不能试图去占用帮助库已占用的工作区号,否则会出错。

## 四、结束语

上面主要讨论的几种联机帮助机制的设计方法,可以联合运行,以取得更为实用的效果。如果再与 READKEY()、LASTKEY()及各种陷阱命令结合起来,则设计效果更佳。

## 参考文献

- ① 《新一代 FOX 数据库及其实用工具》,海洋出版社,严晓舟等
- ② 《FOXBASE+/MAC 参考指南》,国防科大 MAC 发展中心

286

**DECpc 微机广东总代理**  
**广州方正公司**

地址: 广州市东风东路 733 号羊城晚报 19 楼  
电话: 7664962、7664963、7776211—8863,8864,8865

# C 语言程序对成批数据库文件的加密和解密

浙江省台州地区信托投资公司 林荣庆

## 尽

管能将 FOXBASE 和 DBASE 程序经编译或伪编译等措施进行加密,使他人无法通过程序修改数据库的内容,但由于数据库文件本身未加密,他人仍可直接打开数据库进行各种操作,对数据的安全性和保密性都带来问题,因此对数据库本身加密是计算机应用中必须解决的问题。

通常对数据库文件的加密采用改变文件属性,使之成为隐含,但这种加密方法对稍懂 debug 和 ptools 的人来说是不起作用的。本文介绍的是改变数据库文件头的信息代码,C 语言编制的程序对成批数据库文件进行加密和解密。

对 FOXBASE 数据库结构的分析可知,数据库文件头也称库结构描述段,它位于文件的最前面,由系统说明段和字段说明段组成。系统说明段已对整个数据库的组成情况进行了总的说明,它占 32 个字节,第一位是数据库的标志(03),第二至第四位是最后更改日期,第五位至第八位是记录总数,第九位至第十位是文件头的长度,第十一位至第十二位是记录长度,位于系统说明段之后是字段说明段,每个字段说明段都占 32 个字节。

CDJM.C 是对成批 FOXBASE 或 dBASE 数据库文件加密和解密的源程序。其中 buf2[512]是放文件头信息的数组,可存放三十一个字段的信息,一般够用,如用户数据库的字段多于三十一个字段,可以增加数组的元素个数,数据库最多字段是 128 个, tcd 是文件头的字节数, tcd1, 是相应的字个数, caf.dat 文件存放要加密的成批数据库的文件名,格式可如下。

cad.dat

ABC.dbf  
ABC1.dbf  
AB.dbf  
.....

程序先要求用户输入口令,予置口令是 dbase,口令正确则进入下面的加密或解密功能,加密时先把数据库的标志位 03 变成结束标志 1A,再对字段说明段的代码进行变换,代码变换的方法有多种,为简便说明起见,文中把源代码同某数异或,把结果写回数据库文件头,那么数据库无法打开,对其中的记录无法操作。解密时把数据库标志位从 1A 变回到 03,把字段说明再进行一次异或,代码复原。本程序进行异或的数是十六进制的 ffff,读者可指定任何其他十六进制的数,为了更加保密,可结合多种变码方式进行变换。本程序是用 XENIX 系统多用户 C 语言编写的,经编译可产生执行文件 CDJM.EXE,成为 XENIX 系统或 DOS 系统的一个命令,可供 FOXBASE 或 DBASE 程序调用。一般在 FOXBASE 主控程序执行时先进行解密,在应用程序退出主控模块时进行加密,达到有效地保护数据库文件。

程序清单如下:

```
cdjm.c
#include "stdio.h"
int buf2[512], n, tcd, tcd1, fd2, i, *p1;
char c, xz, buf1[13], kl[6];
main()
{
    FILE *fp, *fopen();
    printf("输入口令,五个字符\n");
    initser();
    ngetcho();
    scanf("%s", kl);
    echo();
    if(strcmp(kl, "dbase") != 0)
        exit(0);
    printf("选择:加密选 a,解密选 b.\n");
```

```
scanf("\n%c", &xz);
fp=fopen("cdf.dat", "r");
if(fp<0);
{
    printf("cdf.dat 文件不能打开!\n");
    exit(0);
}
while((c=getc(fp))!=EOF)
{
    for(i=0; i<13; i++)
        buf1[i]=000;
    buf1[0]=c;
    i=1;
    while((c=getc(fp))!='\n')
    {
        buf1[i]=c;
        i++;
    }
    fd2=open(buf1, 0);
    if(fd2<0)
    {
        printf("%S 文件不能打开\n", buf1);
        continue;
    }
    read(fd2, buf2, 32);
    p1=(int *)buf2;
    p1=p1+4;
    tcd=*p1;
    tcd1=(tcd-1)/2;
    close(fd2);
    fd2=open(buf1, 0);
    if(xz=='a')
    {
        read(fd2, buf2, tcd);
        buf2[0]=buf2[0]^0x0017;
        for(i=16; i<tcd1; i++)
            buf2[i]=buf2[i]^0xffff;
    }
    else
    {
        if(xz=='b')
        {
            read(fd2, buf2, tcd);
            buf2[0]=buf2[0]^0x0017;
            for(i=16; i<tcd1; i++)
                buf2[i]=buf2[i]^0xffff;
        }
        else
        {
            exit(0);
            close(fd2);
            fd2=open(buf1, 1);
            write(fd2, buf2, tcd);
            close(fd2);
        }
    }
    fclose(fp);
}
```

# 在 Windows 环境中运行汉字系统

广东汕头金山中学 何管略

一般情况下,汉字系统大都是在 Dos 环境中运行的。以 2.13 汉字系统为例,在 Dos 环境中运行时,可以通过键盘,用拼音、五笔等方法,输入单字、词组;可将 16 点阵的汉字,显示在单色、彩色(CGA,EGA,VGA)屏幕上;可以支持多种打印机,打印 40 多种大小不同、形状各异的字体;还能支持各类经过汉化的应用软件,如 FoxBase 等,提供多项服务。

如所周知,汉字系统对常规内存的开销较大。仍以 2.13 汉字系统为例:在不装载五笔、联想词组等附加程序,并将显示字库全部装入虚盘[设置在 Ext Mem 中,不占常规内存],仅装载 16,24,40 点阵汉字打印驱动程序的前提下,常规内存的分配情况,大致如下表所示。

软件名称	常规内存
MS-DOS 6.0 操作系统	26KB
2.13H 汉字系统	88KB
Foxbase 数据库系统	387KB
自由空间	139KB

由于本机已将 Dos 的部分资料、数据,装入高位内存区 HMA,并将部分设备驱动程序和驻留程序装入上位内存块 UMB 中,才使得总数为 640KB 的常规内存中,剩有 139KB 的自由空间;然而,还是无法继续运行,其它较大型的实用程序或工具软件。

这种单一任务独占全部系统资源的情况,是由 Dos 设置的实时运行模式所决定的,它对大型的、带有汉字的应用程序的编制和调试工作来说,必然增加许多重复性的繁琐操作;既耗时、又费力,往往事倍功半。

窗口软件 Windows 3.1,为 Dos 的各类应用程序[包括汉字系统],提供优越的多任务功能。在 Windows 环境下,我们可以为运行汉化的数据库系统 FoxBase,设置一个窗口;为运行汉化的文字编辑软件,设置另一个窗口,专作修改程序之用;再为运行纯西文的工具软件,如 PCShell 8.0,设置一个窗口。只要 Windows 所控制的内存不会超出,还可以继续设置更多的窗口。

在所开设的众多窗口中,只能有一个窗口是活动的,它可以接受键盘输入、运行程序;其余没有激活的窗口,全都藏在幕后待命,当需要谁的时候,可通过预先自定义的、各自独立的简捷键(Short Cut Key)快速切换出来,协助调

试工作的顺利进行,非常方便。

Windows 3.1 在 386 增强模式下运行时,能够开设多个窗口的关键,是由于它会在硬盘的自由空间中,以交换文件的形式,建立虚拟内存;这就使它所控制的内存总量(如 6MB),实际超过系统的物理内存(如 4MB)。例如,同样对 1.44MB 容量软盘的复制操作,用 Windows 的 CopyDisk 功能,可将来源盘中的资料,一次过全部读入内存,在提示换盘之后,又一次过全部写在目标盘上;而用 Dos 的 DiskCopy B: B:命令,在运行过程中,则需要读、写三个来回,前后换盘共五次。

Windows 3.1 为每一个运行 Dos 应用程序的窗口,设置一套占用内存空间 1024KB 的虚拟机;并按特定的分时机制,安排 CPU 轮流控制各个窗口的虚拟机,使它们在操作人员的指挥下,分别值班运行。

在 Windows 环境中运行汉字系统,有三件事,最为人们所关注:

1、多占常规内存。确有其事;不过它在每个虚拟机的窗口中,驻留三个程序,并自动把最长程序的大部分,装载在上位内存块 UMB 中,仅多占常规内存 7KB,为数不多,影响不大,请阅下表。

程序名称	常规内存	UMBs
Win.com	2KB	
Win386.Exe	1KB	127KB
Command.com	4KB	

2、降低运行速度。当汉字系统在带有窗口信息显示方式的屏幕中运行时,的确如此。不过,我们可以将它安排在,不带窗口信息显示方式的全屏幕中运行,同时采用独占方式。实践证明:运行速度,不会降低。主要原因在于 Windows 采用了下列两项技术措施:

(1)对硬盘上、虚拟内存中的资料、数据的存取操作,既不是按 Dos 的读、写文件的方式,也不是按 BIOS 的读、写扇区的方式,而是采用 32 位的保护模式直接进行的;运行速度比前两种方式都快。

(2)在字符显示速度方面,Windows 环境改进了显示驱动程序,比 Dos 环境要快得多。我们在 Compaq 386/20e 型机上,用 Speed 软件测试,以 chr/ms 为单位的结果如下表:

## 2.13H 五笔字型内部词汇的查阅

福建省农业区划研究所 林宜辉

2.13H 五笔字型本身带有一些内部词汇。在应用中,由于内部词汇不象外部词汇一样能用自带软件进行查阅,这使得很多人由于不了解内部词汇的内容,在很大程度上影响了汉字的输入。为此,笔者用 Turbo c 编写了一个程序。将五笔字型的内部词汇写入 WB-TEXT 文件中,以便于用各种字处理软件或直接用 TYPE 查阅,2.13H 系统的内部词汇在 wbzx.exe 中,将其拷到当前目录下并把下列 C 语言编译后运行 WB 即可运行。查阅中可发现不少内部词汇有误,大家可参照有关资料修改内部词汇,这里就不详述。程序在 AST286 机上通过。

```
#include "stdio.h"
#define SIZE 100
main()
{
    FILE *fp1, *fp2, *fopen();
    int i,n,b,c,k;
    long num;
    char str[SIZE],a[3];
    c=0x80;
    /* 打开文件 出错则返回 */
    if((fp1=fopen("wbzx.exe","rb"))==NULL)
    {
        printf("不能打开文件\n");
    }
```

```
        exit(0);
    };
    fp2=fopen("wbtext","wb");
    /* 寻找词组 */
    while(!feof(fp1))
    {
        a[0]=fgetc(fp1);
        if(a[0]==0x08)
        {
            a[1]=fgetc(fp1);
            a[2]=fgetc(fp1);
            if((a[1]==0x09 && (a[2]==0x0a))
                break;
            else {
                fseek(fp1,-2L,SEEK-CUR);
                continue;
            }
        }
        fseek(fp1,21L,SEEK-CUR);
        while(!feof(fp1))
        {
            b=fgetc(fp1);
            if(b<c) break;
        }
        /* 将词组取出后存入 wbtext 文件 */
        i=0;
        k=2;
        num=0;
        fseek(fp1,-1L,SEEK-CUR);
        while(!feof(fp1))
        {
            b=fgetc(fp1);
```

```
        if(b<c) {
            if(b==0x00) break;
            b=b|c;
            str[i]=b;
            i=i+1;
        } else
        {
            if(i<2) {
                i=0;
                k=0;
            }
            else
            {
                str[i]=b;
                str[i+1]=fgetc(fp1);
                str[i+2]=' ';
                n=i+3;
                for(i=k;i<n;i++)
                    fputc(str[i],fp2);
                if(num%10==0) {
                    fputc(0x0d,fp2);
                    fputc(0x0a,fp2);
                }
                i=0;
                k=2;
                num++;
            }
        }
        fclose(fp1);
        fclose(fp2);
        return;
    }
```

289

运行环境	字符显示速度
MS-DOS 6.0	4255
Windows	7992

3、减弱软件功能。不会,Windows 不但全部保持虚拟机上,所运行软件的各项功能,而且提供 Dos 环境下所没有的特异功能。如在运行带汉字的数据库系统时,发生死锁现象,是难以避免的。若在 Dos 环境下,我们无法解脱,只得将那些,尚未从虚盘转储入硬盘的、非常重要的资料、数据,全部放弃,重新启动系统。而在 Windows 环境下,我们则可以切换到别的窗口,调用工具软件,从容处理各项资

料、数据,然后再将死锁的窗口,强行关闭;不必重新启动系统,又可在新的窗口中,从头运行汉字数据库系统。

大多数的汉字系统,都可在 Windows 环境中顺利运行,但有个别软件不能。例如,WPS 系统的较低版本,可在 Windows 环境中运行;而带软字库的、与 WPS 3.0F 配套的 SPDos 6.0F 系统,则无法在 Windows 环境中运行。究其原因主要是:它需要 284KB 以上的、Ext Mem 自由空间,用来装载显示字库;而 Windows 却将全部 Ext Mem 的自由空间,控制起来,以便分配给各个窗口使用,彼此相互冲突所致。

288



# 基于 WINDOWS 的中文输入

广州天河智能电子技术开发公司 马 敬

WINDOWS 3.1 自问世以来,迅速获得了巨大的成功,成为计算机应用的最佳平台之一。MICROSOFT 公司在此基础上,于 93 年 10 月下旬又推出了 MS WINDOWS 3.1 中文版(以下简称中文版)。据估计,该软件将对我国的软件行业产生巨大的冲击。笔者认为,中文版不仅是 WINDOWS 界面的更新,更重要的是,它提供了一个标准的 WINDOWS 中文平台,既促进了中文软件的国际化、标准化,更有利于国际优秀软件的汉化。该软件在中文输入方面的新特点主要体现在三方面。

## 一、对传统输入法的改进

中文版提供了国标/区位、拼音两种输入法。它吸收了许多国内优秀软件的优点,对拼音输入法作了较大的改进。不妨,同以 WPS 金山汉字系统为代表的汉字系统的拼音输入法作一比较。如下表所示。

	金山 6.0 版	WINDOWS 中文版
输入方法	全拼、双拼双音	全拼、双拼字词
高频先见	✓	✓
联想功能	✓	✓
可扩充词库	✓	✓
单字/词组输入	✓	✓
动态调频	无	有
模糊音设置	无	有
自定义键盘	无	有

通过上表可以看出,中文版的拼音输入法兼容了金山 6.0 版的全部优点,又提出了动态调频、模糊音设置、自定义键盘三个新特性。所谓动态调频是在吸取了国内一些智能录入软件的特长,为专业录入人员提供极大的方便,即在高频先见的基础上,系统自动对用户输入的每个字进行计数,并建立一个频度表。在使用过程中,系统就以此频度表为基础,对重码字进行重新排序,专业高频字就被排在了前面,从而进一步提高了输入效率。同时,整个复杂的过程对用户都是透明的。模糊音输入对于那些汉语发音不准的用户非常有用。汉语拼音中许多发音,如 in 和 ing, z, c, s 和 zh, ch, sh 等的发音,对有些用户分不太清楚。此时用户就可以把自己不易分清声母和韵母归为一类。例如,在声母模糊音设置了 s 和 sh。在设置完成后,用户如果输入'si',提示行将显示发'si'和'shi'音的所有字。这样,偶然的发音错误或输入错误,输入了错码,系统仍然能够输出正确的字。拼音输入法还有一个特点就是可以由用户自定义双拼

韵母表,这对于那些以前用惯了其他汉字系统双拼韵母键盘的用户是非常有用的。这样用户就不需要改变自己的双拼输入习惯,只需要通过在 WINPY.INI 文件中加入习惯的双拼输入键,重新定义双拼键盘就可以了。

仅是一个拼音输入模块,即可以看出微软公司的 WINDOWS 中文版是独具匠心的,完全从用户的角度出发,才能对简单的拼音输入法有如此的改进。

## 二、开放的中文输入法接口

中文版在西文版的基础上,提供了一个开放的中文输入法接口模块“通用码表生成器”,为中文输入法的扩充提供了方便。实际上,码表生成器就是一种汉字输入法的自动生成系统。任何一种汉字输入法,输入过程就是从汉字编码到汉字/词(或符号)的映射过程。只要根据需要建立自己的汉字/编码关系,就可以将自己建立的汉字/编码映射装入 WINDOWS 工作平台。在中文版中,通常需要经过以下几个步骤:

- 1、首先建立码表文件;
- 2、用码表生成器编译码表文件
- 3、在控制面板的输入方法程序项内安装及选用码表输入码表文件是以 .TXT 为后缀的文本文件,具体格式如下:

```
[Description]
Name=输入法名称
MaxCodes=最大码长
UsedCodes=码元
WildChar=查询键
Sort=排序标志

[TEXT]
编码 1
编码 2
.....
编码 n
```

在生成码表文件之后,通过简单的编译和安装过程,就可以使用新的输入方法了。由于使用码表输入法具有模糊查询键的功能,对非专业人员和初学者更大大提高了效率。以往忘记某词根而无法输入的情况就不太会出现了。

使用码表生成器,关键是得到一份完整的码表文本。金山系统的五笔字型和表形码,无疑是国内用户最多的输入方法之一。但是,其编码并不是以开放的形式提供的,而是

以压缩的形式存储在可执行文件中。所以,笔者通过研究 WPS 金山系统的五笔字型 and 表形码输入模块,编写了两个程序,成功地把五笔字型和表形码移植到了中文版下。这两个程序仅仅作为一个范例,说明码表输入法的用法。同时,也提供了一份五笔字型和表形码的完整码本。具体文件格式可参见程序说明。该程序在 Compaq 386, TC V2.0 下编译通过。所用汉字系统为金山 DOS5.0 或 5.1。在程序生成了 BXN. TXT 和 WBZX. TXT 之后,通过通用码表生成器的编译,就可以安装使用了。

### 三、对应用中文程序的支持

WINDOWS 是一种国际化的软件。其软件开发包 SDK(3.1 或更高版本)已经包含了对双字节代码(DBCS)和混合多字节代码(MBCS)的支持。中文版的平台是一种基于 DBCS 的系统。在这个平台上,应用程序所需的中文输入将非常简单。同时,对一些编程规范的所谓“友好”的西文程序将同样获得中文输入的支持。这里所指的编程规范,是指尽量采用 WINDOWS 提供的函数和例程。

对一个典型的中英文输入模块的流程可以这样编写:首先,采用接收 Ansi 字符的 WM-CHAR 来接收第一个字节,然后需要用 IsDbcsLeadChar()函数来快速判断该字节是否为双字节代码的开始;若是,就可以用 PeekMessage()函数从消息队列中取下个字符。直到输入完毕。具体程序略。

一般说来,在应用程序的设计中还可以通过以下两个函数,来处理中文输入字符串。

LPSTR AnsiPrev(lpStart, lpCurrentChar)用来返回给定字符的前一个字符的长指针位置;

LPSTR AnsiNext(lpCurrentChar)用来返回给定字符的下一个字符的长指针位置;这些指针的移动都是以单字(汉字或字符)为单位,可以避免半个汉字等现象。限于篇幅,这里就不赘述了。

WINDOWS 中文版是建立在 WINDOWS 巨大成功的基础上,无疑对其他国产的中文平台是一种强烈的冲击。今后,中文版能否成为 WINDOWS 标准的中文平台?还有待时间作出结论。

附五笔字型和表形码的生成程序:

程序 1

```
/* 本程序通过读取金山汉字系统的 WBM.COM 文件,生成 WINDOWS 3.1 中文版的五笔字型编译码本文件。可编译生成 WINDOWS 下的五笔字型输入法。同时也提供了一份完整的、包括一级简码、全部单字和词组在内的码本文件,在 Compaq 386SX, Turbo C 2.0 下编译通过 */
#include <stdio.h>
#include <ctype.h>
#include <alloc.h>
#define wz-off 0x5ff2 /* 词组压缩码表位置 */
#define jm-off 0x57 /* 一级简码码表位置 */
#define offset 0x6ac /* 单字压缩码表位置 */
static char mb[]="abcdefghijklmnopqrstuvwxyz"; /* 五笔字型编码 */
```

```
int main()
{
    unsigned int cc=hi,cc=lo,i,c,c1,c2,c3,b1,b2,b3,b4;
    FILE *fpout,*fpin;
    if((fpin=fopen("c:\\wps\\wbx.com","rb"))==NULL)
        /* 读取码表文件 */
        { printf("Error Open file bxm.com!!!\n");
          exit(1);
        }
    if((fpout=fopen("wbx.txt","ba"))==NULL)
        /* 打开输出文件 */
        { printf("ERROR LOAD THE INPUT OR OUTPUT FILE!!!\n");
          exit(1);
        }
    /* 以下程序建立文件头 */
    fprintf(fpout,"[Description]\n");
    fprintf(fpout,"Name=五笔字型\n");
    fprintf(fpout,"MaxCodes=4\n");
    fprintf(fpout,"UsedCodes=%s\n",mb);
    fprintf(fpout,"WildChar=?\n");
    fprintf(fpout,"Sort=1\n");
    /* 以下程序建立一级简码码表 */
    fprintf(fpout,"[Text]\n");
    fseek(fpin,jm-off,SEEK-SET);
    for(i=0;i<25;i++)
        { fputc(fgetc(fpin),fpout);
          fputc(fgetc(fpin),fpout);
          fputc(mb[i],fpout);
          fputc('\n',fpout);
        }
    /* 以下程序建立单字码表 */
    fseek(fpin,offset,SEEK-SET);
    for(cc=hi=0xb0;cc=hi<=0xf7;cc=hi++)
        for(cc=lo=0xa1;cc=lo<=0xfe;cc=lo++)
        {
            fputc(cc=hi,fpout);
            fputc(cc=lo,fpout); /* 先生成国标码 */
            c1=fgetc(fpin); /* 取压缩码表 */
            c2=fgetc(fpin);
            c3=fgetc(fpin);
            /* 解码 */
            if((b1=(c1>>3)&0x1f)!=0) fputc(mb[b1-1],fpout);
            if((b2=(c1<<2)&0x1d|(c2>>4)&0x03)!=0)
                fputc(mb[b2-1],fpout);
            if((b3=(c2<<1)&0x1e|(c3>>5)&0x01)!=0)
                fputc(mb[b3-1],fpout);
            if((b4=c3&0x1f)!=0) fputc(mb[b4-1],fpout);
            fputc('\n',fpout);
        }
    /* 以下程序建立词组码表 */
    fseek(fpin,wz-off,SEEK-SET);
    i=0;
    c=fgetc(fpin);
    do
    {
        switch(isascii(c)) /* 最高位为 0 的字节为压缩码表开始 */
        {
            case 0:if(c==0xff){ /* FF 为词组块结束标志 */
                    c=fgetc(fpin);
                }
            }
        }
    }
```

```

c=fgetc(fpin);
++i;
break;
}
else{
do{
fputc(c,fpout);
}while(!isascii(c=fgetc(fpin)))&& c!=0xff;
/* 解码 */
fputc(mb[i],fpout);
fputc(mb[(c1>>2)& 0x1f]-1],fpout);
fputc(mb[(((c1<<3)& 0x1c)|((c2>>5)& 0x07))
-1],fpout);
fputc(mb[(c2& 0x1f)-1],fpout);
fputc('\n',fpout);
break;
}
default:{
c1=c;
c2=fgetc(fpin);
c=fgetc(fpin);
break;
}
}
while(i!=25);
fclose(fpin);
fclose(fpout);
return 0;
}

```

## 程序 2

/\* 本程序通过读取金山汉字系统的 BX.M.COM 文件,WINDOWS 3.1 中文版的表形码编译码本文件,可编译生成 WINDOWS 下的表形码输入法。同时也提供了一份完整的表形码表。在 Compaq 386SX, Turbo C 2.0 下编译通过 \*/

```

#include <stdio.h>
#include <alloc.h> #define offset 0x1aa
static char mb[]="abcdefghijklmnopqrstuvwxyz12345";
int main()
{
unsigned int cc—hi,cc—lo,c1,c2,c3,b1,b2,b3,b4;
FILE *fpout,*fpin;
if((fpin=fopen("c:\wps\bxm.com","rb"))==NULL)
{ printf("Error Open file bxm.com!!!\n");
exit(1);
}
fseek(fpin,offset,SEEK—SET);
if((fpout=fopen("bxm.txt","a+"))==NULL)
{ printf("ERROR LOAD THE INPUT OR OUTPUT FILE!!!\n");
exit(1);
}
fprintf(fpout,"[Description]\n");
fprintf(fpout,"Name=表形码 \n");
fprintf(fpout,"MaxCodes=4\n");
fprintf(fpout,"UsedCodes=%s\n",mb);
fprintf(fpout,"WilsChar=?\n");
fprintf(fpout,"Sort=1\n\n");
fprintf(fpout,"[Text]\n");
for(cc—hi=0xb0;cc—hi<=0xf7;cc—hi++)

```

```

for(cc—lo=0xa1;cc—lo<=0xfe;cc—lo++)
{
fputc(cc—hi,fpout);
fputc(cc—lo,fpout);
c1=fgetc(fpin);
c2=fgetc(fpin);
c3=fgetc(fpin);
if((b1=(c1>>2)& 0x1f)!=0) fputc(mb[b1-1],fpout);
if((b2=(c1<<3)& 0x18|(c2>>5)& 0x07)!=0)
fputc(mb[b2-1],fpout);
if((b3=c2& 0x1f)!=0) fputc(mb[b3-1],fpout);
if((b4=c3& 0x1f)!=0) fputc(mb[b4-1],fpout);
fputc('\n',fpout);
}
fclose(fpin);
fclose(fpout);
return 0;
}

```

290

# WPS 打印蜡纸一法

安徽省广德县教育局委员会 刘德旺

目前用 WPS 作排版系统,LQ—1600K 打印机作输出设备的办公系统很普及,但在用普通打字蜡纸打印文稿时,经常出现蜡纸打皱现象,用下列方法可克服:

一、在用 WPS 进行文稿排版时,按下列数据设置边界:

①在左边界设置为 13;②右边界设置为 61 或 62。然后按 ^ B 进行段落重排。

二、用 PCTOOLS 修改 WPS 打印纸的尺寸。

WPS 本身提供了一组可供选择的打印纸的尺寸,用 PCTOOLS 将 WPS 本身的宽行打印纸长度改为和普通打字蜡纸(16K)相仿,方法是:

①启动 PCTOOLS;

②选择 WPS.EXE 文件;

③用 F 命令,查找 WPS 中宽行打印纸的尺寸代码“F0 06 90 09”;

④选“E”命令进行编辑;

⑤将“F0 06 90 09”改为“F0 05 90 09”

⑥存盘,退出 PCTOOLS。

三、调整 LQ—1600K 打印机竖向导纸器左导轨,使其靠最左边放置。

四、打印选择 WPS 打印状态参数时,选宽行打印纸,并选换页时不停,即可打印出漂亮的蜡纸。

291

# WPS 非正常退出遗失文件的找回

工商银行江苏省洪泽县支行 罗 升  
江苏省洪泽县医药公司 张 群

WPS 文字编辑处理系统,以其简单易学、使用方便、功能齐全深得广大用户的喜爱。在使用 WPS 的过程中,相信大家都会有这样的体验,当刚刚编辑完的大段文字信息、材料在模拟显示或打印输出时常出现诸如内存溢出出错、非正常出错,退出到 DOS 操作系统等情况发生。当这些信息未存盘时将会造成被编辑的文件遗失,辛苦工作付之东流,这是很懊恼的事情。这些“遗失”的文件能否找回?笔者通过对 WPS 系统的研究分析,借助 DEBUG 调试工具找回了“遗失”的 WPS 文件,免除了重新录入编辑之苦。

用户在用 WPS 编辑文件时,被编辑的文件都是存放在内存当中,只有在用户键入 Ctrl+K+S 键或存盘退出 WPS 时才将已编辑的文件由内存转储于磁盘中。经过分析,被编辑的文件在内存中都是以 DS:0100H 为起始地址的连续存储单元中,且从偏移量 X100~X4FF 地址的 1024 个字节中存放的是 WPS 格式文件公共信息,用户录入的信息存放于偏移量为 X500 以后的内存储区中。据此可以利用 DEBUG 调试工具找到已编辑完的最后几个汉字或字符的机内码,即可确定文件的实际长度,将其转存于磁盘之中达到恢复的目的。下面以一个实例谈一谈具体实现的方法(如图 1)。

## 1. 确定汉字或字符机内码

汉字的机内码是由国家标准信息交换汉字编码即(国标码 GB2312-80)的每个字节最高位加一得到。如“阿”国标码 GB2312-80 为 3022H 对应二进制为 0011000000100010 高位加一为 1011000010100010 对应机内码 B0A2H。对字符则相应为 ASCII 码值。“结束”的 GB2312-80 国标码为 3D61H、4A78H 相对应机内码为 BDE1H、CAF8H。

## 2. 确定文件长度

启动 DEBUG 程序,使用搜索命令(Search Command)搜索最后几个汉字或字符。本例设最后几个汉字为“结束”,对应机内码为 BDE1H、CAF8H;具体操作为:

-S DS:00100 02000 BD E1 CA F8

显示-514E:052E

-514E:1245

-514E:18ED

这时可确定所显示的第一个地址是所要的最后录入的信息。则的出“遗失”文件的长度为 0534H-1000H=0434H(查到的地址为 052E 应加上“结”后半字节、“束”两字节以及文件结束符两字节;即为 052EH+0005H=0534H)。

## 3. 修改 CX 寄存器

使用 R 命令修改计数寄存器 CX,具体命令为:

-R CX

-CX 0000

:0434(为输入文件长度)

## 4. 存储找回的文件

使用 DEBUG 程序调试工具的 N 及 W 命令,具体操作为:

-N FILE.WPS(存储的文件名)

-W DS:00100 0 1 0434 (存到 A:软盘中)

Writing 0434 bytes (显示信息)

至此遗失的 WPS 文件以 FILE.WPS 文件名的形式被保存于 A:驱动器之中。

需要说明的是,当我们使用 WPS 录入,编辑大段信息时要养成输入,编辑了一定数量后键入 Ctrl+D+S 文件存盘命令的好习惯,以避免发生意外所带来的不必要麻烦。

```
514E:0470 00 49 6E 76 61 6C 69 64-20 70 61 72 61 6D 65 74
514E:0480 65 72 20 66 6F 72 20 57-50 53 20 21 0D 0A 24 4D
514E:0490 53 2D 44 4F 53 20 76 65-72 73 69 6F 6E 20 65 72
514E:04A0 72 6F 72 20 21 0D 0A 24-54 68 65 20 66 69 6C 65
514E:04B0 20 6F 70 65 6E 65 64 20-6E 75 6D 62 65 72 20 69
514E:04C0 73 20 6C 65 73 73 20 74-68 61 6E 20 32 30 20 69
514E:04D0 6E 20 43 4F 4E 46 49 47-2E 53 59 53 20 21 0D 0A
514E:04E0 24 20 20 20 20 20 20 20-20 20 20 20 20 20 20
514E:04F0 20 20 20 20 20 20 20 20-20 20 20 20 20 20 20
514E:0500 31 32 33 34 35 36 37 38-39 30 61 62 63 64 65 66
514E:0510 67 68 69 6A 6B 6C 6D 6E-6F 70 71 72 73 74 75 76
514E:0520 77 78 79 7A 21 40 23 24-25 5E 26 2A 28 29 BD E1
514E:0530 CA F8 1A 10 B2 4B CD 10-33 C0 50 B2 44 BE 6E 36
514E:0540 E8 27 51 33 D2 8B 0E 98-02 8B 1E 88 00 B4 40 1E
514E:0550 8E 1E 8A 00 CD 21 E8 1F-3F 1F B4 3F 8B 1E 86 00
```

图 1

292

美国惠普(HP) 美国保时(Pulse)特约代理  
海谊电子仪器实业公司

电话:4420788 转 8912、8910、8916

# 通用菜单信息控制 技术的设计与实现

华南理工大学计算中心 黄远利

## 一、定义和概述

为了方便阐述,约定用汉字 FOXBASE+语言作设计语言,并作如下定义。

定义 1,子程序:用 FOXBASE+语言编成的可以运行的单个基本程序单位。

定义 2,物理模块:按照某一内部规则,把若干个子程序组合成的过程文件。

定义 3,索引模块:按照某一处理功能,把若干个物理模块中的子程序组合成的形式模块。

定义 4,物理菜单:按照物理模块的子程序对应编排出来的层次菜单。

定义 5,索引菜单:按照索引模块的子程序对应编排出来的层次菜单。

定义 6,定位指针:能够唯一确定某层菜单位置的控制参数。

菜单信息的处理是信息管理软件设计的一个重要课题,在设计时有多种实现方法。例如,在一些教科书中或者应用软件中,有的将菜单直接嵌入应用程序之中,层次菜单依赖某个物理模块,修改菜单信息时程序也要作相应的修改;有的把菜单与程序分离,设置菜单信息库,与前一种方法相比,这种方法有了较大的进步,但层次菜单仍依赖某个物理模块,等等。总之,在菜单设计中,以下问题是需要研究的。其一,菜单设置受物理模块的约束,难以实现对不同模块和子程序的任意组合调用。其二,编程和修改工作量大,容易出错和难以维护。其三,不直观,无法综观全貌。

## 二、设计思想和特点

1、设计思想,参照索引技术和指针技术的特点,将菜单驱动程序与菜单分离,把物理菜单信息组合成索引菜单信息,建立通用的菜单信息库。根据不同的要求组合成不同层次的菜单,每条菜单信息可根据需要设置定位指针,由定位指针定位到相应的层次菜单。整个设计过程大体经历以下几个阶段:总体数据规则、通用逻辑设计、物理设计、通用菜单信息设计、程序设计,其中最重要的是通用逻辑设计,其中包括参数数据库设计和逻辑数据库设计。通过菜单信息库可以综观整个系统的模块与子程序之间的从属关系和调用关系。为了便于菜单信息的管理,还配备一个菜单信息管理模块,实现对菜单信息的录入、修改、查询、删除和打印。平时,增删修改菜单信息时,只修改菜单信息库的有关参数,不用修改菜单驱动程序,以不变的程序应付万变的菜单信息。

2、设计特点,通用性,固定不变的菜单驱动程序,适用于任意层次菜单的组合,既能调用本系统的过程,又能调用其他系统的过程,只要在相应的菜单信息中填上简单的几

个参数,就可以接口调用。灵活性,层次菜单不受某个模块的约束,根据实际需要灵活设置。同一层次菜单既可以对应于物理模块,又可以对应于索引模块。同时,还可以设置多个定位指针,这样,既可以连续递次进退定位,又可以间接跳跃进退定位,不受任何约

束,进退定位灵活多变。安全性,对于不同的操作管理人员设置不同的操作权限,对菜单的选择使用作相应的限制。易维护,使用简单快捷,菜单管理模块提供多种功能,不懂得编程的管理人员也能增删修改菜单信息。

## 三、菜单信息库的结构

以某个物资子系统为例,设置录入、修改、查询、删除、打印、维护等七个物理模块,每个物理模块有多个子程序。设置编码、物价、合同、设备、材料、仓库、公共管理等多个索引模块。为了方便起见,约定菜单信息库名为 WZCD.DBF。字段有菜单名称、层参数、控参数、子程序名、过程(模块名)、序号等六个字段(实际应用时,还可以根据需要增设参数)。对于控参数,当控参数等于层参数时为该层菜单的定位标志,处于层的首行。控参数值为 9 时是终极控参数,执行相应模块的子程序,其对应的层参数只是起序号作用,不作定位指针;值为 4 时是转移控参数,其对应的层参数是一个定位指针,控制定位到相应的层次,其他控参数本身就是一个定位指针,控制定位到相应的层次,其对应的层参数只起序号作用,不作定位指针。数据库结构如下:

菜单信息表

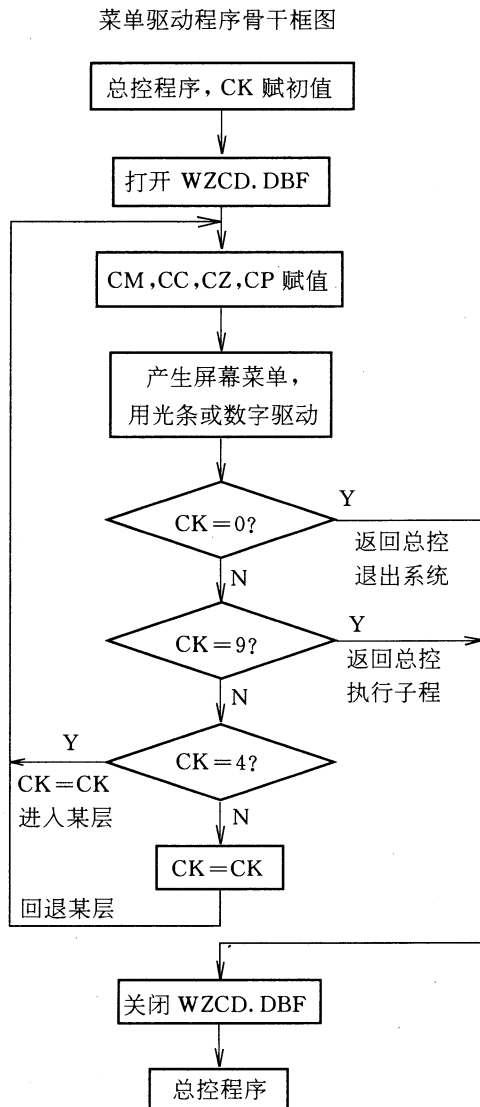
菜单名称	层参数	控参数	子程序名	过程	栏参
物资管理操作表	1	1	WGLC10		0001
0,退出顶层	10	0			0002
1,编码管理	11	4		WGLC	0003
2,物价管理	12	4		WGLC	0004
3,设备管理	13	4		WGLC	0005
4,合同管理	14	4		WGLC	0006
5,仓库管理	15	4		WGLC	0007
仓库管理操作表	15	15	WGLC15		0008
0,退回上层	150	1			0009
1,设备仓库管理	151	4	WGLC151	WGLC	0010
2,主材仓库管理	152	4		WGLC	0011
3,退回顶层	153	1			0012
设备仓库管理操作表	151	151			0013
0,退回上层	1510	15			0014
1,录入设备仓	1511	9	WLRR02	WLRC	0015
2,修改设备仓	1512	9	WXGT02	WXGC	0016
3,查询设备仓	1513	9	WCXX11	WCXC	0017
4,删除设备仓	1514	9	WSCS02	WSCC	0018
5,打印设备仓	1515	9	WDYD01	WDYC	0019
6,统计设备仓	1516	9	WTJR01	WTJC	0020
7,退回原层	1517	15			0021
8,退回顶层	1518	1			0022

.....

#### 四、菜单信息控制技术的实现

为了实现菜单信息控制技术,用 FOXBASE+ 语言编写了总控制程序和菜单驱动程序,总控制程序负责设置初始参数,根据菜单驱动程序提供的参数打开或者关闭有关过程文件,调用相应的过程和子程序。菜单驱动程序根据菜单信息库的参数,产生屏幕菜单,供用户选择数字或者光条驱动,最终为总控制程序提供打开过程和调用子程序的参数。通用菜单信息控制技术在实际工作中已经得到了应用,如笔者带领学生进行毕业设计,特别是与某企业研制 MIS,菜单信息达一千多条,响应速度快,取得了较满意的应用效果。下面是菜单驱动程序的骨干框图,约定菜单库名为 WZCD. DBF,菜单名称为 CM,层参数为 CC,控参数为 CK,子程序名为 CZ,过程名为 CP,序号为 CH。

附菜单驱动程序骨干框图:



#### 参考文献:

《汉字 FOXBASE+ 2.10 用户手册》刘藏泽。

## 如何有选择地

## 学习汉字输入法

广东广雅中学

梁立志

近年来,随着电子计算机的不断普及不少家庭都拥有了计算机。最近有一些朋友和学生向我询问学习哪一种汉字输入方法好。要较好地回答这个问题,首先就要对各种汉字输入方法的特点有所了解。

常见的汉字输入法有几十种之多,把它们分类大致有取形、取音和形音结合三种。取形分解的汉字输入法有五笔字型、表形码、首尾码、五笔划、仓吉码等,取音的汉字输入法有全拼音码、双拼音码等,形音结合的汉字输入法有自然码、首尾拼音码等等。

每一种汉字输入方法都要熟记键盘上的一些分区和代码。一般来说,取形分解的汉字输入法要记忆的键盘代码较多,以五笔字型汉字输入法为例,要记忆一百多个字根和键盘上的一些分区才能比较自如地输入汉字。而它的优点是,输入操作和书写汉字相似,而且重码很少。还是以五笔字型输入法为例,熟记字根和熟练操作以后,汉字输入的速度可达到每分钟一百个汉字以上。取音的汉字输入法要记忆的键盘代码则较少,以全拼音码为例,因为拼音的代码跟键盘上的英文符号基本上是一致的。所以,对于已经学习了汉语拼音的朋友们来说,根本不需要刻意去记忆就可学会这种汉字输入方法。但是,由于汉字的同音字多,拼音编码又没有声调的区别,因此,这种汉字输入法会因为出现较多的重码字而影响汉字的输入速度。以自然码为代表的形音结合的汉字输入法则是集中了前两种汉字输入法的优点而进行编码的。以音码减少记忆,以形码减少重码字。

认识了各种汉字输入法的一些特点以后,还要对个人的特点也要有所了解,每个人对形、音的认识和理解能力有所不同,一些人可能会对形象的东西比较容易接受,而另一些人可能会对声音比较敏感。

对于不同年龄的学习者其特点也各不相同。年青人记忆力好,而中、老年人的记忆力相对就不如年青人。

还有,对于专业的电脑操作员,他们所面对的是需要输入的稿件,首先进入大脑的是字的形象。而对于一般的创作人员,他们首先想到的应该是字的发音。

那么要选择一种适合自己学习的汉字输入法,就要根据自己的所长和不同的工作岗位来挑选,这样将会收到事半功倍的效果。

例如,前面所说的电脑操作员学习取形分解的汉字输入法就比较合适。而一些学习过汉语拼音的创作人员学习取音的汉字输入法就会比较容易掌握。

学习汉字输入操作也跟学习其他知识一样,“熟能生巧”。相信你看了本文以后,也一定能根据自己的特长找到一种适合自己学习的汉字输入方法。

# 几种拷贝工具的比较

吉林工业大学 马健

前随着配高密度软驱微机的日渐普及,专门针对各种高密度软盘的磁盘拷贝工具也日渐流行,并逐渐取代了DOS传统的DISKCOPY命令。这些工具软件的性能各有差异,对不熟悉的用户来说常常感到难于选择。本文将对几种目前比较流行的拷贝工具的性能进行比较,以供参考。本文所讨论的几种拷贝工具是:

TBH-Softworx 的 HD-COPY V1.7(简称HC);

Micro System Design, Inc. 的 Disk Dupe Pro V4.0(简称Dup);

EAWOLF N. T Group Inc. 的 Hi-Performance DISKCOPY V2.07(简称SC)。

性能数据均在 Olivetti M300-10 机上测出, CPU 为 386SX/20, 内存 4M, 硬盘 120M。

## 一、拷贝速度

种软件把整张 5 寸 1.2M 盘拷到另一张同型盘上所用时间如下表所示。

	读盘时间	只写时间	写+校验	格式化+写+校
HC	30 秒	27 秒	63 秒	105 秒
DUP	40 秒	40 秒	70 秒	110 秒
SC	42 秒	80 秒	105 秒	无格式化功能

由表中数据可知, HC 的拷贝速度最快, DUP 次之, SC 的最慢。在实际使用时, 由于 HC 能自动根据 FAT 识别出盘末空道, 并在读、写时略去这些空道, 因此实际读、写时间可能比表 1 所列更短。对内存的使用能力也会影响到拷贝速度, HC 和 DUP 都能使用 EMS, XMS, SC 只能用 EMS。

## 二、备份速度

这里所说的备份是指把整张软盘的内容转存到硬盘的某个文件上, 将来需要时再把备份文件还原到软盘上。这个功能可以帮助软盘周转不灵的用户“挤”出一些急需的软盘, 也可用来把大量软盘备份到硬盘上, 便于携带。

软盘备份功能只有 HC 和 DUP 具有, 二者的备份方式还不一样。

HC 备份时须先选 Read 菜单把软盘内容读入缓冲区, 再选 Put 菜单把缓冲区内容写入扩展名为 .IMG 的备份文件。恢复时则须先选 Get 菜单把备份文件读入缓冲区, 再选 Write 菜单写入软盘。对一张 5 寸 1.2M 软盘备份约需 50 秒, 恢复(带校验)约需 75 秒。另外 HC 在对多张软盘连续备份时能按增量自动产生编号文件名。

DUP 备份时直接把读到的软盘内容写入扩展名为 .DDI 的备份文件, 恢复时也是一次恢复, 不需要在几个菜单项之间来回折腾。对一张 5 寸 1.2M 软盘备份时间约为 40 秒, 恢复时间(含写检验)约为 66 秒。

由以上比较可知在备份、恢复方面 DUP 要比 HC 更快速、方便。

## 三、成组盘的拷贝能力

用 HC 拷贝一组软盘时, 最快捷的方法是把 Option 菜单下的 User mode 选为 RISK, 这时系统每秒测试一次驱动器, 发现源驱动器中有新盘时立刻读, 发现目标驱动器有新盘时立刻写。它的成组拷贝速度无疑是三者中最快的。但在只用一个驱动器拷贝时由于需要交替插入源盘和目标盘, 容易造成顺序混乱。当源盘未贴写保护时会被当做目标盘而造成覆盖。

DUP 成组拷贝能力最差, 每拷完一张源盘后都要按键选择才能开始拷下一张源盘。

SC 是专为成组拷贝设计的, 能一次把所有源盘读入缓冲区, 然后再顺序写入目标盘。即使只用一个软驱也不用在拷贝中途交替插入源盘和目标盘, 因此能避免一般的人为失误。

## 四、软盘格式的兼容性

HC 是专为高密度软盘设计的, 因此不支持 5 寸倍密盘的标准格式和扩容后的各种非标格式, 但对 5 寸高密盘和 3 寸高密盘则能支持 10 种标准或非标格式(5 寸从 1.2M 到 1.48M, 3 寸从 1.44M 到 1.72M), 并在一定程度上支持异种格式盘之间的相互拷贝。DUP 支持标准的 320K、360K、720K、1.2M、1.4M、2.8M 格式, 不支持异种格式相互拷贝。SC 只支持 360K、1.2M、720K、1.4M 格式, 也不支持异种格式相互拷贝。

## 五、软件的易用性与安全性

HC 的菜单项较多, 只能用键盘操纵, 不支持鼠标。有时为了完成一项功能需选多次菜单, 略显繁琐。当用一个软驱以 RISK 方式快速成组拷贝时, 容易出现人为失误。但当软盘出现故障时, HC 能越过有故障的磁道继续工作, 并在事后给出准确、详细的错误信息。DUP 和 SC 遇到软盘故障时将停止工作, 给出的错误信息有时不够准确。

DUP 的菜单项是最多的, 但它支持鼠标, 因此容易使用。防止人为失误能力比 HC 稍强。

SC 的菜单太简单, 用键盘足以操纵, 在成组拷贝时最方便、人为失误最少。

## 六、附加功能

HC 除了能对软盘拷贝、备份外, 还提供数据加密手段, 加密后的软盘无法用正常的方式读、写。此外 HC 还提供非标格式化工具, 可用于软盘扩容。

DUP 提供标准格式化工具, 和整盘删除工具。

SC 仅是一个纯粹的拷贝工具, 无任何附加功能。



# 浅谈 C 程序中对 Ctrl—Break 的处理

华南师范大学物理系研究生 许俊禹

众所周知, Ctrl—Break 为用户强行中断当前活动进程返回原进程提供了强有力的途径,但同时也可能由于操作员误操作而引起意外的中断,因而前功尽弃。因此,如何控制 Ctrl—C 或 Ctrl—Break 序列,就显得极为重要。下面将在分析其机理的同时,给出相应例程及 C 语言调用例程的实例。

当一个程序(进程)开始执行, DOS 在为它置新的 Ctrl—Break 处理程序的地址之前,先把原地址存入子进程的 PSP 中。在子进程结束之前, DOS 从该子进程的 PSP 中恢复原来进程的 Ctrl—break 处理程序的地址。因此用户程序在完成自己的处理后可不用恢复该地址,这一任务由 DOS 自己完成(为了演示,本文仍提供 release() 函数来恢复)。另外用户程序不能直接调用 INT 23H, 但可以编制自己特定的处理程序替代省缺的 DOS 处理程序。使其执行自己希望的操作。象其它中断处理程序一样,凡用户处理程序改变了的任何寄存器都应加以保护。退出处理程序的方法通常有两种,效果也各不相同,一是借用中断 IRET 指令,这条指令恢复被中断程序,使它继续执行,本文采用此方法;二是借助一个长返回,这时, DOS 通过检查进位标志位(CF)确定是否继续执行原来程序。另外,若由 DOS 中断功能调用 09H, 0AH, 检测到 Ctrl—C 组合键时,将在屏幕上显示字符 ^ C, 可截取键盘中断 09H 过滤字符 ^ C (03H), 达到不显示该字符,以保护屏幕的目的。

本文例程 BK. ASM 由 TASM 编译成. OBJ 文件,同时 C 语言调用程序 TBK. C 由 TCC 编译成. OBJ 文件,然后用 TLINK 连接(本例应采用小模式,否则应根据所用模式调整 BK. ASM 中的 args 变量)即可,若经常用到该例程则可将它连入库,用 TLIB 可实现之。

; BK. ASM

args equ 4 ;small mode

—TEXT segment byte public 'CODE'

assume cs:—TEXT

public —capture, —release

```

: static int flag;
: capture (& flag):
—capture proc near
    push bp
    mov bp, sp
    push ds
    push di
    push si
    mov ax, word ptr [bp+args]
    mov cs:flag, ax
    mov cs:flag+2, ds
    mov ax, 3523h
    int 21h
    mov cs:int23, bx
    mov cs:int23+2, es
    mov ax, 351bh
    int 21h
    mov cs:int1b, bx
    mov cs:int1b+2, es
    push cs
    pop ds
    mov dx, offset ctrlbrk
    mov ax, 02523h
    int 21h
    mov ax, 0251bh
    int 21h
    pop si
    pop di
    pop ds
    pop bp
    ret
—capture endp

: release ()
—release proc near
    push bp
    mov bp, sp
    push ds
    push di
    push si
    mov ds, cs:int1b
    mov dx, cs:int1b+2
    mov ax, 251bh
    int 21h
    mov dx, cs:int23
    mov ds, cs:int23+2

```

## 七、结论

1、HC 的综合性能最佳,不仅速度快,而且支持异种格式盘之间相互拷贝,并提供加密、扩容等附加功能。

2、DUP 由于支持鼠标,因此对习惯用鼠标的用户来说

也许更有吸引力。另外在做软盘备份时也比 HC 快速、方便。

3、在成组拷贝时 SC 是最佳选择,不仅容易使用,而且能把人为失误减至最少。

# 中国软件行业协会新举措

## 《世界软件总汇》编辑出版工作正式启动

为促进国民经济信息化建设,配合“三金”工程实施,提高中国计算机应用水平,推动软件产业和市场发展,由中国软件行业协会倡导发起,并联合英国 ITC 数据工程有限公司和(中外合资)北京其汇电子科技电子科技有限公司共同组织实施的大型系列丛书《世界软件总汇》编辑出版工作目前正式启动。

《世界软件总汇》丛书的主要内容包括:(1)广泛介绍中外主要软件厂商基本情况,专业领域和市场业绩;(2)详细介绍有特色的代表性软件产品,侧重于用户所关心的应用环境、性能、价格与服务等方面;(3)集成优秀软件产品的演示系统于光盘上,随书发行,供用户试用;(4)收录客观和权威性的软件产品评测报告,帮助读者正确认识各项软件产品。

作为中国第一本专门面向软件领域的权威性市场大全,《世界软件总汇》旨在有效地增进中国软件产业和用户对国外产业情况和产品发展状况的认识与了解,帮助中外

厂商建立积极的合作关系,为用户选购所需的软件产品提供周全的服务,进而改善目前中国软件市场产品单调重复,高水平软件缺乏,买好应用软件困难的局面。该书的正式出版发行,将有力地促进软件厂商及其软件产品品牌的宣传,强化软件的市场意识和商品概念,推动中国软件市场向正规化和国际化的方向发展。

《世界软件总汇》丛书首卷将于 1994 年 12 月正式出版,首期印刷发行 10000 册。电子部计算机推广司司长兼中国软件行业协会理事长杨天行同志担任丛书总顾问。目前,丛书编辑出版的各项工作已全面展开,由清华大学教授王尔乾等 10 余位计算机软件及应用专家负责丛书编写工作,并组织丛书首卷内容的征集工作。在海外的征集工作将委托英国 ITC 数据工程有限公司负责具体实施。

国内联系地址为:中国软件行业协会,北京 162 信箱,邮政编码:100036,联系电话:01-2576888, FAX: 01-2545613, 联系人:张健小姐

297

```

mov ax, 2523h
int 21h
pop si
pop di
pop ds
pop bp
ret
—release
endp

ctrlbrk
proc far
push bx
push ds
push ax
mov bx, cs: flag
mov ds, cs: flag + 2
mov word ptr ds: [bx], -1
pop ax
pop ds
pop bx
iret
ctrlbrk
endp
flag
dw 0, 0
int23
dw 0, 0
int1b
dw 0, 0
—TEXT
ends
end

/*      TBK.C      */
#include "conio.h"

```

```

#include "stdio.h"
main ()
{
    int hit = 0, ch;
    static int flag = 0; /* static int is a must */
    printf (" \n Testbreak.c is running. \n");
    printf ("Press Ctrl-c or Ctrl-break to test handler. \n");
    printf ("Press the Esc key to exit Testbreak. \n");
    capture (& flag);
    printf ("Testbreak has Captured interrupt vectors. \n");
    while ((ch & 127) != 27)
    {
        hit = kbhit();
        if (flag != 0)
        {
            printf (" \n Ctrl-break has been detected. \n");
            flag = 0;
        }
        if (hit != 0)      ch = getch();
    }
    release();
    printf (" \n Testbreak has Released interrupt vectors. \n");
}

```

### 参考文献:

《DOS/BIOS 使用详解》电子工业出版社 舒志勇等译  
《C 语言高级实用教程》清大出版社 尹彦芝 编著

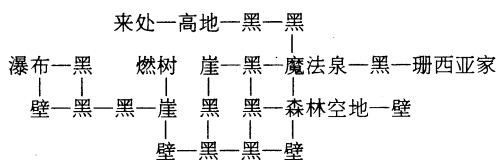
296



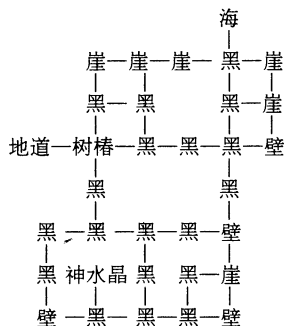


### 第三关: 珊西亚家。

这一关的主要目的是拿到皇家圣物之一的圣杯, 并要飞往凯兰迪亚城堡。通过珊西亚家的地道有一个新版面, 因此下面画两张平面图。



第三关平面图 I



第三关平面图 II

注释:

黑: 黑暗森林;

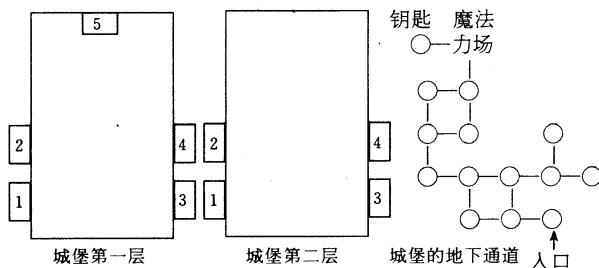
壁: 石壁;

崖: 悬崖。

此关的图像及动画最为精美, 情节也十分有趣。珊西亚家门口的小东西会让你皮命蛙口; 在美丽海岸用紫色护身符也可飞向城堡, 但飞行过程中你将葬身鱼腹。有兴趣的读者不妨试一试。这一关所用宝物最多, 具体走法请见后面的“部分宝物的获得及用法”。

### 第四关: 凯兰迪亚城堡。

这一关是游戏的最后版面, 来到这里后, 不能再回到前一关。所以在飞往凯兰迪亚城堡前, 一定要带三样东西: 火海中的钥匙、圣杯、兰花。否则无法通关。城堡又分三层, 平面图如下。



来到这一关, 你首先见到你父母的坟墓。把兰花放在墓上, 你将得到红色护身符。现在, 你有四色护身符, 你一定要

善于运用。城堡第一层的 1、2 是相通的。2 是一间书房, 在这里你应该找到皇冠, 那就是《书名第一个字的重要性》, 《逆天而行》, 《转变中的思想》, 《阴暗的天空》, 《阳光与植物》。2 还通向地下通道, 你要使用护身符穿越魔法力场, 拿到一把钥匙。第一层的 3、4 也是相通的, 4 是厨房, 这里有另一件皇家圣物——权杖。城堡第二层的房间内是你朋友的石像, 你必须战胜恶魔马尔寇才能使他们复元。在 1 中你要破解一个机关才能再拿到一把钥匙, 你必须敲出四个音符的美妙乐曲。现在, 你有两把钥匙、三件圣物。用钥匙打开第一层的 5, 再把三件圣物按一定次序放好, 就可以和马尔寇决斗了。用你的红色护身符, 再躲到镜子旁边, 让马尔寇作法自毙, 自取灭亡。

### 部分宝物的获得及用法:

#### 第一关:

泪珠: 从泪泉接到, 救活即将枯死的柳树;

羽毛: 用黄色护身符为鸣鸟疗伤得到, 交给奇怪茅屋中的达恩;

红宝石: 在红宝石树上。太阳石、郁金香在喷泉处;

魔法书卷: 把羽毛给达恩后得到, 用于火海和燃树。

#### 第三关:

魔法泉水: 从燃树处找回珍珠, 修好魔法泉, 自己饮用可得到蓝色护身符, 还要给珊西亚带一瓶;

蓝莓: 采回新鲜的蓝莓后, 珊西亚会自动离开, 这时可发现地毯下的地道。在瀑布边采得;

蓝水晶、蓝莓: 配制蓝色浓汤。蓝水晶可由蓝宝石代替;

石榴石、兰花: 配制红色浓汤。石榴石可用玛瑙代替;

黄玉、郁金得: 配制黄色浓汤。黄玉可用翡翠代替;

空瓶: 身上带着空瓶离开珊西亚家后再返回, 地上会再多一个空瓶, 你至少需要四个空瓶子;

神水晶: 配药;

紫色药剂: 红色浓汤和蓝色浓汤配制, 饮用后可使你变小, 找回被松鼠抢走的圣杯;

橘色药剂: 红色和黄色浓汤配制。在海岸饮用可变成飞马, 飞向凯兰迪亚城堡;

绿色药剂: 蓝、黄浓汤配制而成, 剧毒!

兰花: 在美丽海岸采得;

苹果: 在森林空地松鼠家交换圣杯;

### 护身符魔法的用途:

黄色护身符: 疗伤之用。为鸣鸟疗伤; 拿红宝石被蛇咬伤后, 给自己疗伤; 在第四关, 催眠赫曼。

紫色护身符: 用于飞行和照亮阴暗的地方;

蓝色护身符: 隔空取物, 第三关拿取悬在半空中的圣杯; 感应力场, 第四关突破魔法力场;

红色护身符: 隐身, 和火海钥匙共同使用进入凯兰迪亚城堡; 隐身, 和马尔寇决斗。

# INTEL 网络疫苗撒下企业网络防毒网

北京 王立峰 李路

INTEL 电脑网络疫苗(LANDesk virus Protect)是 INTEL 公司与世界著名电脑防毒厂商趋势科技公司合作开发,第一套以服务器为主的网络防毒产品。自 92 年五月份推出以来,全球销售量已逾二万套。INTEL 网络疫苗程序设计的最主要目的,在于减轻网络管理人对防毒管理的负担,远离网络中毒的梦魇。INTEL 网络疫苗为有数个和或众多网络服务器的企业网络,寻求更简单的,更具效率的网络防毒集中管理方案。

防毒措施集中管理

## INTEL 网络疫苗功能有:

### 企业网络集中管理

目前,市面上所有的网络防毒管理软件,都仅能就一台服务器加以设定或管理,虽然能达到集中管理其下所属各工作站,但对于企业内的另一个网络服务器,甚至其连线工作站的病毒状况,则显得力不从心,必须对各个网络服务器逐一检视才行。

INTEL 网络疫苗在这方面有了重大的突破,企业内的网络服务可统一设定,管理,更新,甚至做成各种合并报表。

举例来说,甲企业的网络共有二十台服务器,原先网络管理者需设定每一台服务器的病毒扫描方式,扫描时间,报表产生方式,紧急情况通知方式,设定完毕后,每天还得一一阅读每台服务器的病毒报告,遇上病毒码更新时,还要更新每台服务器,十分麻烦,费事。

有了 INTEL 网络疫苗,网络管理者可统一设定(或分组设定)企业内所有的服务器,并集中管理。每一台服务器及其连线的工作站随时在掌握之中,更方便的是,只要一个指令就可以把各个服务器的病毒记录报表加以合并,产生一个新报表,使得网络防毒管理轻松又有效率。

### 共用病毒码

INTEL 网络疫苗不但在更新服务器的病毒码时,同时自动更新工作站的病毒码档案,甚至服务器和服务器之间也能进行自动病毒码更新,值得一提的

是,新版中还提供一道功能,让使用者通过数据机定期自动上 BBS 下载最新病毒码档案,自动进行更新。

### 智慧型病毒陷阱侦毒

INTEL 网络疫苗除了使用病毒码比对的方式侦测已知病毒外,更加入智慧型病毒陷阱,抓未知的新病毒,由于是以分析病毒的特性及感染模式作为判断,使得一些变体引擎病毒,多形体病毒,隐型飞机型病毒全部无所遁形,大幅降低网络受病毒感染机会。

## GUI 介面利学习

### 图形用户介面

INTEL 网络疫苗提供 DOS 及视窗两种使用者介面,并设有在线辅助说明,便利网络管理者使用及设定。INTEL 网络疫苗的各项功能,例如:浏览任一服务器的病毒事件报表,目前的设定状态,启动扫描程序等等,此外,产品尚提供可将所有网络服务器根据特性分成不同组别(DOMAIN),分别设定,倘若组别变更,只需在荧幕上以鼠标点两下,该服务器的设定便可立即由 A 组转到 B 组。

### 加入工作站保护

以网络防病毒而言,最需要保护的是网络服务器,工作站往往是病毒最喜欢藏身的地方,伺机利用工作站作为进入服务器的最佳途径,所以 INTEL 网络疫苗加强了各个工作站的保护,即使该工作站尚未登入网络作业系统,仍在 INTEL 网络疫苗防毒保护范围之内。

INTEL 网络疫苗通过一个常驻扫描程序随时检查工作站是否受病毒感染,与服务器的监控相互呼应,达到全面性的网络防毒效果。

301

软件法保护咨询热线

(020)7504151

逢星期五下午 2:30~6:00

# INOC 病毒的解密与消除

厦门大学 许东华

本系近来遇到一种新病毒,其现象是感染“.EXE”文件和长度不大于 2048 字节的“.COM”文件(但不感染 COMMAND.COM),感染后长度增加 1787~1833 字节。用美国的 CPAV1.4 和 MCAFEESCAN113 等均无法检测出来;用冰岛大学推出的 F-PROT 软件可以检测出文件感染了名为“INOC”的病毒,但无法清除,只能把被感染文件改名(“.COM”改为“.VOM”;“.EXE”改为“.VXE”)。运行了带毒程序后,用 MEM 或 MI 命令可以看到病毒已用该程序的名字驻留内存,占用 2288 字节,并截取了 INT 2F 中断。此外该病毒还有一个特殊现象;若是被感染的文件名为 SCAN.EXE,那么运行该文件时屏幕上将会显示出如下一段信息:

```
Anti Virus
If your software has been
inflected by other viruses,
run your software,then the
virus will be cleaned!
THANK YOU!
--Mr. MeiHua--
```

初一看这似乎是一种病毒疫苗或抗体之类有益无害的东西,能自动清除感染上的其他病毒。我们用“新世纪”病毒做了试验,证明确实如此,难怪 F-PROT 要把它命名为“INOC”病毒(在英文里“INOC”是“inoculate”的缩写,意思是免疫、接种)。但是由于这个病毒感染一些文件时破坏了文件的正常结构,致使这些文件运行出错甚至死机;而且病毒在某些情况下会无限制地重复感染同一文件,使其长度越来越大,多占用了不少磁盘空间,所以我们宁可把它作为一般的病毒来看待,进行彻底剖析,找到检测、消除的办法。

病毒的开头部分设计得相当巧妙;我们经过多次跟踪分析,发现其初始化部分进行了双重加密,而解密过程则利用了 1 号单步中断。显然,病毒设计者想以此阻碍任何调试软件对病毒的跟踪。以下简述病毒的解密原理。

病毒体从偏移 0000 到 0007 是连续 8 条 PUSH 指令,

用于保存各寄存器的当前值,从 0008H 开始与解密有关,如下所示:

```
XXXX:0008 E88B04      CALL 0496
;调用偏移 0496 处的子程序,先保存住 1 号中断(即单步中断)向量当前值,而后修改这个向量使;之指向病毒体偏移 0154 处。
XXXX:000B 2E          CS;
;记下指向待解密码的初始指针值 40H。
XXXX:000C C70640054000 MOV WORD PTR,0040
XXXX:0012 9C          PUSHF
;这五条指令的作用是:打开标志寄存器的中断位和陷阱位,而保持其他位不变。
XXXX:0013 58          POP AX
XXXX:0014 0D          0003 OR AX,0300
XXXX:0017 50          PUSH AX
XXXX:0018 9D          POPF
```

从这段程序可以看出,如果病毒没有受到跟踪,那么只有执行到偏移 0018 处的 POPF 指令时才打开陷阱位,这条指令一执行完立即发生单步中断,转入病毒体偏移 0154 处的第一重解密程序。

第一重解密程序的大致思想是:修改堆栈中保存的中断现场的指令指针 IP,强制本中断程序返回到偏移 0019 处;然后解出实际要执行的下一条指令,放到 0019 处,再从中断返回,就直接返回到这条指令。执行完这条指令后又发生单步中断,于是再次进入解密程序,解出下一条指令,再返回执行。这样解出一条指令执行一条指令,一直到这一段需要动态解密执行的程序结束,才关闭陷阱位,恢复程序的正常运行。那么这段需要动态解密执行的程序是什么呢?是第二重解密程序。第二重解密机制本身倒是比较简单,只按顺序对后面的一段密码进行异或操作。但是两重加密结合起来,却足以令任何经验不足的跟踪者晕头转向。

经过对病毒的剖析,我用 Turbo C 设计了一个程序(程序清单附后),能够检测并彻底清除 INOC 病毒。程序运行时首先自动检查内存是否已有 INOC 病毒,若有则把它清除掉;如果运行本程序时命令行未带参数,则要求输入文件名,这里你不仅可以输入带通配符“\*”或“?”的文件名,也可输入路径名,诸如“C:”(C 盘当前目录)、“D:”(D 盘根目录)、“.”(当前盘当前目录)等都是合法的路径名。如果输入的是路径名,则程序自动遍历检查所指定的目录及以下各级子目录的所有后缀为“.EXE”、“.VXE”、“.COM”、“.VOM”的文件,若有病毒则将之彻底清除。由于该病毒虽然附着在原文件尾部,但并不总是紧接在原文件

自然码

最自然的汉字输入法

广东代理:电脑杂志社 电话:5516911-3273



数据后面,而是隔开 0~15 字节,破坏了原文件长度的信息,所以我的程序从病毒头相对文件头的位置来决定杀毒后文件的长度,会比未染毒之前增长 0~15 字节,不过这通常不影响染毒文件杀毒后的正常运行。本程序在 286、386 上调试通过,已在厦大计算机系发挥了一定的效用。

#### 附程序单:

```

/* program name: KI.C (to test and kill INOC virus) */
#include <stdio.h>
#include <io.h>
#include <stdlib.h>
#include <dos.h>
#include <dir.h>
FILE *f1;
unsigned char filename[20],oldname[20],vname[20],buf[0x200];
char *vptr;
long vhead;
int exe, vxe, com, vom;
int virfound=0;
union REGS r;
struct SREGS s;
main(int argc, char *argv[])
{
    int driver;
    char olddir[MAXPATH];
    char drive[3], dir[MAXDIR], name[9], ext[5];
    int flag;
    int len;

    printf("\n=== Inoc Virus Killer ===");
    printf("\n By Xu Donghua, Xiamen University. Apr,1994");
    printf("\n=====\\n");
    search—kill—memory();
    if (argc<2)
    {
        printf("Please input the infected file name: ");
        scanf("%s", filename);
    }
    else
        strcpy(filename,argv[1]);
    driver=getdisk();
    getcwd(olddir,MAXDIR);
    if (!strcmp(filename, ".") /* define "." as " *.* " */
        strcpy(filename, " *.* ");
    else
    {
        flag=fnsplit(filename, drive, dir, name, ext);
        if (flag & DRIVE)
            setdisk(drive[0] % 16 -1);
        if (flag & DIRECTORY)
        {
            len=strlen(dir);
            if (len>1)
                dir[len-1]=0;
            chdir(dir);
        }
        fnmerge(filename, "", "", name, ext);
        if ( !(flag & FILENAME) & & !(flag & EXTENSION)
            /* without name */
            strcpy (filename, " *.* ");

```

```

} /* end of else */
search—kill—file(filename);
if (!virfound) printf("\nInoc Virus not found. ");
printf("\nCompleted !\\n");
setdisk(driver);
chdir(olddir);
} /* end of main() */

search—kill—memory()
{
    unsigned hi—int2f;

    r.x.ax=0xaeff;
    int86(0x2f,&r,&r); /* to test that if virus is in memory */
    if (r.h.al==0)
    {
        printf("%cFound Inoc Virus in memory...", 7);
        hi—int2f=peek(0,0xbe);
        if (peek(hi—int2f,0)==0x5350) /* can kill */
        {
            r.h.ah=0x49;
            s.es=hi—int2f - 0x10;
            intdosx(&r,&r,&s);
            /* release memory block occupied by virus */
            poke(0,0xbc, peek(hi—int2f,0x548));
            /* restore INT2F vector */
            poke(0,0xbe, peek(hi—int2f,0x54a));
            printf("killed.\\n");
        }
    }
    else
    {
        printf("but fail to kill (INT 2F used by another TSR).\\n");
        printf("%cYou'd better reboot now!\\n",7);
    }
}

} /* end of search—kill—memory() */

search—kill—file(char *name)
{
    struct fblk file;
    char fname[4], *extptr;
    char thename[14];
    char pathname[MAXDIR];
    char wholenam[MAXPAT];

    if ( findfirst(name, &file, 0xff)==0)
    {
        getcwd(pathname,MAXDIR);
        if (pathname[strlen(pathname)-1]!='\\')
            strcat(pathname, "\\");
        do
        {
            strcpy(thename,file.ff—name);
            if (file.ff—attrib == FA—DIREC)
            { if (thename[0]!='.')
                {
                    chdir(thename);
                    strcpy(fname, " *.* ");
                    search—kill—file(fname);
                    chdir("..");
                }
            }

```

```

    }
    else
    {
        extptr=(char *)strchr(thename, '.')+1;
        vxe=exe=vom=com=0;
        if (!strcmp(extptr, "VXE"))
            vxe=1;
        else
            if (!strcmp(extptr, "EXE"))
                exe=1;
            else
                if (!strcmp(extptr, "VOM"))
                    vom=1;
                else
                    if (!strcmp(extptr, "COM"))
                        com=1;
        if (vxe || exe || vom || com)
        {
            strcpy(wholename, pathname);
            strcat(wholename, thename);
            printf("Searching -- %s ", wholename);
            gotoxy(1, wherey());
            while (findvirus(wholename))
            {
                virfound=1;
                printf("Found Inoc Virus in %s ...", wholename);
                if (vxe || vom)
                {
                    strcpy(vname, thename);
                    strcpy(oldname, vname);
                    oldname[strlen(oldname)-3]=vxe?'E':'C';
                }
                killvirus(wholename, file.ff-attrib);
            } /* end of while(findvirus()) */
        } /* end of if( strstr(thename, ".EXE") || vxe) */
    } /* end of else */
} while (findnext(&file)==0);
} /* end of if(findfirst) */
else
    printf("\nFile <%s> not found", name);
} /* end of search-kill-file() */

findvirus(char *wholename)
{
    int i;
    int found=0;
    char commark1[3]={0xe, 0x58, 0x5};
    char commark2[3]={0x50, 0x33, 0xc0};
    char exemark[2]={0x4d, 0x5a};

    fl=fopen(wholename, "rb");
    if (com || vom)
    {
        rewind(fl);
        fread(buf, 8, 1, fl);
        if (!strcmp(buf, commark1, 3) && !strcmp(buf+5, commark2, 3))
            found=1;
    }
    else /* exe or vxe */
    {
        fseek(fl, -0x200, SEEK-END);
        vhead=ftell(fl);
        fread(buf, 0x200, 1, fl);
        for (i=0; i<0x100; i++)
        {
            if (!strcmp(buf+i, exemark, 2) && !strcmp(buf+i+0xd2,
                exemark, 2))
            {
                vptr=buf+i;
                vhead=vhead+i-0x55c;
                found=1;
            }
        } /* end of for */
    } /* end of else */
    fclose(fl);
    return(found);
} /* end of findvirus() */

killvirus(char *wholename, char attrib)
{
    unsigned oldlen, oldhead;
    int handle;

    setattr(wholename, 0);
    fl=fopen(wholename, "rb+");
    if (com || vom)
    {
        oldlen=(buf[3]+(unsigned)buf[4]*256)*16-256;
        oldhead=oldlen+0x55c;
        fseek(fl, oldhead, SEEK-SET);
        fread(buf, 0x1f, 1, fl);
        rewind(fl);
        fwrite(buf, 0x1f, 1, fl);
    }
    else /* vxe or exe */
    {
        rewind(fl);
        fwrite(vptr, 1, 0x1c, fl);
        oldlen=vhead;
    }
    handle=fileno(fl);
    chsize(handle, oldlen);
    fclose(fl);
    if (vxe || vom)
    {
        if (rename(vname, oldname)==0 /* rename succceful */
            printf(" (renamed to %s) ", oldname);
        )
            setattr(wholename, attrib);
        printf("killed!\n"); } /* end of killvirus() */

    setattr(char *wholename, char attrib);
    {
        s.ds=FP-SEG(wholename);
        r.x.dx=FP-OFF(wholename);
        r.x.cx=attrib;
        r.x.ax=0x4301;
        intdosx(&r, &r, &s);
    }
}

```

# 中外软件廊软件报价

## 美国 MICROSOFT 公司软件(联系电话:6689452)

产品名称	单价
DOS6.0	1158.00 元
WINDOWS3.1	1743.00 元
WINDOWS3.1(简体中文版)	1460.00 元
WINDOWS3.1(简体中文版 UPGRADE)	807.00 元
WINDOWS3.1 中文升级版	350.00 元
用户必须有原版的中文 WINDOWS3.1 才能用此软件升级	
WINDOWS FOR WORKGROUPS	2919.00 元
WINDOWS NT 3.10	5792.00 元
WINDOWS NT 3.10ADVANCED SERVER	35042.00 元
SQL FOR NT4.20(WORKGROUP SYSTEM 10 用户)	35042.00 元
SQL FOR NT 4.20(DEPARTMENT SYSTEM 64 用户)	93542.00 元
SQL FOR NT 4.20(ENTERPRISE SYSTEM 无限用户)	175442.00 元
SQL SERVER PROGRAM TOOLKIT 4.20	8132.00 元
EXCEL 5.0 FOR WINDOWS 中文版	5792.00 元
WORD 3.0 FOR WINDOWS	5792.00 元
POWERPOINT 3.00 FOR WINDOWS	5792.00 元
OFFICE 4.00 FOR WINDOWS 英文版	8775.00 元
(包括:WORD、EXCEL、POWERPOINT、MAIL)	
OFFICE 4.00 FOR WINDOWS 中文版	8775.00 元
(包括:WORD、EXCEL、POWERPOINT、MAIL)	
OFFICE 4.00 PROFESSIONAL FOR WINDOWS	10518.00 元
PROJECT 3.00 FOR WINDOWS	8132.00 元
WORKS 2.00 FOR WINDOWS	2328.00 元
WORKS 3.00 FOR DOS	1743.00 元
VIDEO 1.00 FOR WINDOWS	2328.00 元
WINDOWS PUBLISHER 1.0(SOLUTION SERIES)	1626.00 元
ACCESS 1.10 FOR WINDOWS	5792.00 元
ACCESS 1.10 VERSION UPGRADE	175.00 元
ACCESS 1.10 DISTRIBUTION KIT	5792.00 元
FOXPRO 2.50 FOR DOS	5792.00 元
FOXPRO 2.50 FOR WINDOWS	5792.00 元
FOXPRO 2.50FOR DOS DISTRIBUTION KIT	5792.00 元
FOXPRO 2.50 FOR WINDOWS DISTRIBUTION KIT	5792.00 元
FOXPRO 2.50 LIBRARY CONSTRUCTION KIT	5792.00 元
FOXPRO 2.50 CONNETIVITY KIT	5792.00 元
FOX GRAPH 1.00	3249.00 元
C/C++ COMPILER 7.00(包括 WINDOWS SDK)	5838.00 元
MACRO ASSEMBLER 6.00	1649.00 元
WINDOWS SDK 3.10	3838.00 元
WINDOWS DDK 3.10	6903.00 元
WIN32 SDK 3.10	5792.00 元
WINDOWS NT DDK 3.10	5792.00 元
MDK 1.00 FOR WINDOWS	4343.00 元

## VISUAL BASIC 2.00 FOR WINDOWS PROTOOLKIT

VISUAL C++ PRO 1.00	5442.00 元
VISUAL C++32 BIT VERSION(FOR NT) 1.00	5838.00 元
VISUAL BASIC FOR WINDOWS STANDARD EDITION 3.00	7008.00 元
	2328.00 元
VISUAL BASIC FOR WINDOWS PROFESSIONAL EDITION 3.00	5792.00 元
VISUAL BASIC PROFFESIONAL VERSION UPGRADE 3.0	1158.00 元
TESTING TOOLS 2.00 FOR WINDOWS	5442.00 元
WINDOWS SOUND SYSTEM 3.10	3200.00 元
WINDOWS SOUND SYSTEM 2.00	
(WITH MICROPHONE & SOFTWARE)	935.00 元
WINDOWS SOUND SYSTEM 2.00	
(WITH SOUND BOARD)	2562.00 元
DOS6.2 STEP UP	150.00 元
用户必须有原版的 DOS6.0 才能用此软件升级	

## CAD/CAM 软件(联系电话:6689452)

ACAD R12(DOS,WIN,NETWORK,MAC,DEC ULTRIX	
	62000.00 元
HP/UX,RS/6000,SUN SPARC,SGI,SUN SOLARIS)	
ACAD R12 WITH AME 2.1 (DOS,WIN,NETWORK,MAC,DEC	70000.00 元
ULTRIX,HP/UX,RS/6000,SUN SPARC,SGI,SUN SOIARIS	
ACAD R12 TYADITIONAL CHINESE WITH AME(DOS)	75000.00 元
ADVANCED MODELING EXTENSION-AME R2.1	
(DOS,WIN, SUN SPARC/SOLARIS,DEC/ULTRIX,HP/UX,RS6000 SGI)	8000.00 元
ADVANCED MODELING EXTENSION-AME TYADITONAL CHINESE R2.1(DOS ONLY)	8000.00 元
ACAD IGES TRANS 5.1.01(DOS,WIN,DEC/ULTRIX,HP/UX,SGI,SPARC)	18000.00 元
DESIGN EXPERT-ACAD12+AUTOSURF(DOS,SUN SPARC,HP/UX,SGI)	83000.00 元
MANUFACTURING EXPERT-ACAD12+IGES TRAN 5.1.01+AUTOSURF+AUTOMILL & IGES (DOS,SUN SPARC,HP/UX,SGI)	158000.00 元
AUTOSURF/AUTOMILL & IGES(DOS,HP/UX,SGI,SUN SPARC)	130000.00 元
AUTOSURF R2(FOR ACAD R12 DOS ONIY)	23000.00 元
AUTOMILL(DOS,HP/UX,SGI,SUN SPARC)	95000.00 元
服装 CAD	70000.00 元
时装设计系统(AT486+彩色扫描仪+FARGO 彩色打印机+设计软件)	90050.00 元

## 国产 CAD/CAM 软件(联系电话:6689447)

DUM 大漠工程图纸理解及管理系统	11000.00 元
PICAD 参量化工程设计系统	12800.00 元
PICPD 冷冲模具 CAD 系统	7400.00 元
ALCAD 铝门窗设计系统	30000.00 元
PINC 两轴半数控自编程系统	7500.00 元
SUMS-IM2D 注塑模二维 CAD 系统	13000.00 元

CPD 冷冲模 CAD/CAPP/CAM 系统	40000.00 元
MICCIMS 集成制造系统	50000.00 元
ZH-CAPP 计算机辅助工艺过程专家系统	15000.00 元
FEA 有限元分析系统	15000.00 元
建筑室内装潢 CAD 系统	30000.00 元

#### LOTUS 专营店(联系电话:6689452)

LOTUS 1.2.3	1980.00 元
(简体中文版在中文 WINDOWS 下运行的著名电子表格)	
LOTUS AMIPRO	1980.00 元
(简体中文版在中文 WINDOWS 下运行的强大的字处理软件)	

#### 2.13 专营店(联系电话:6661810)

2.13K 办公排版系统	2800.00 元
(2.13K-I 汉字系统+2.13K 办公排版系统)	
2.13K-I 汉字系统+AUTOCAD 支持	2800.00 元
2.13K-I 汉字系统+办公排版系统+AUTOCAD 支持	3200.00 元
2.13K 网络汉字系统+网络办公排版系统(10 用户)	5000.00 元
2.13K-I 汉字系统+办公排版系统+超级数据库工具包	4300.00 元
3.13 网络汉字系统(10 用户)	5000.00 元
3.13 网络汉字系统(20 用户)	9000.00 元
3.13 网络汉字系统(50 用户)	12000.00 元
3.13 网络汉字系统(100 用户)	15000.00 元
3.13 网络汉字系统(200 用户)	18000.00 元
3.13 网络汉字系统(250 用户)	20000.00 元

#### 长城、超想系列软件(联系电话:6689452)

长城图文编排系统	500.00 元
长城 TIDE/C 多媒体数据库系统(单色图象、纯软件)	800.00 元
长城 TIDE/C 多媒体数据库系统(彩色图象、带视频卡)	6700.00 元
长城五合一办公通讯系统(带 FAX/MODEM 卡)	3380.00 元
长城 9000B 字符汉卡	1600.00 元
长城 805 图形加速卡	2800.00 元
超想挑战者套装软件	1100.00 元
(CXDOS5.0, CXED, CXWPS, 自然码, GK 数据库工具)	
中文 WINDOWS 的超想自然码	380.00 元

#### 万达系列软件(联系电话:6689446)

万达财务核算系统	2800.00 元
万达通用数据库输入输出系统	1800.00 元
万达应收款核算系统	1800.00 元
万达应付款核算系统	1800.00 元
万达物资管理系统	1800.00 元
万达现金出纳管理系统	680.00 元
万达通信录及名片管理系统	180.00 元
万达全国航班列车轮船时刻表	180.00 元
万达数据库工具软件	2800.00 元

#### 彩色平面专业设计类(联系电话:6689457)

COREL DRAW 4.0	4800.00 元
COREL DRAW 5.0	5800.00 元
COREL ARTSHOW 3.0	900.00 元

COREL ARTSHOW 4.0	1100.00 元
CORELDRAW 3.0 CD VERSION	2600.00 元
CORELDRAW 4.0 UPGRADE	4300.00 元
COREL VENTURA 4.2	3900.00 元
COREL VENTURA 4.2 UPG.	1650.00 元
COREL CD POWER PAK	1650.00 元
COREL SCSI PRO 2.0	2100.00 元
COREL SCSI PRO 2.0 UPG	1240.00 元
COREL NETWORK MANAGER	9300.00 元
COREL NETWORK MANAGER UPG	3900.00 元
PHOTO SHOP 2.5	8000.00 元
PICTURE PUBLISHER	4800.00 元
PAGE MAKER 7.0	7500.00 元

#### 彩色三维专业设计类

3DS3.0	28000.00 元
--------	------------

#### 建筑专业设计类

ARE-24	18000.00 元
RENDER STAR	18000.00 元

#### 网络类(联系电话:6689452)

NETWARE V3.12 (250 用户)	112455.00 元
NETWARE V3.12 (100 用户)	62955.00 元
NETWARE V3.12 (50 用户)	44955.00 元
NETWARE V3.12 (25 用户)	33255.00 元
NETWARE V3.12 (10 用户)	22455.00 元
NETWARE V3.12 (5 用户)	9855.00 元
NETWARE SFT III V3.11 (250 用户)	170955.00 元
NETWARE SFT III V3.11 (100 用户)	94455.00 元
NETWARE SFT III V3.11 (50 用户)	67455.00 元
NETWARE SFT III V3.11 (25 用户)	47655.00 元
NETWARE SFT III V3.11 (10 用户)	35955.00 元
NETWARE SFT III V3.11 (5 用户)	23355.00 元
NETWARE V4.01 (1000 用户)	431955.00 元
NETWARE V4.01 (500 用户)	237555.00 元
NETWARE V4.01 (250 用户)	141255.00 元
NETWARE V4.01 (100 用户)	79155.00 元
NETWARE V4.01 (50 用户)	56655.00 元
NETWARE V4.01 (25 用户)	44055.00 元
NETWARE V4.01 (10 用户)	28755.00 元
NETWARE V4.01 (5 用户)	12555.00 元
NETWARE ACCESS SERVICES V1.3	21555.00 元
NETWARE MULTIPROTOCOL ROUTER V2.0	8955.00 元
INTEL LANDESK VIRUS PROTECT	7000.00 元
(100 用户网络防病毒软件)	
INTEL LANDESK VIRUS PROTECT	15000.00 元
(无限用户网络防病毒软件)	

#### 中外软件廊软件商的天堂,用家的乐园

软件联系地址:广州市流花路 119 号锦汉大厦 11 楼

联系单位电话:各软件经营单位、电话

中外软件廊专线:6689457

# 功能强大的 VESA Local Bus

## 三合一显示卡

文伯聪

人们对电脑的要求越来越高了。三十二位微机已取代了十六位微机的位置,占据了大部分的微机市场。以前的十六位驱动电路在三十二位微机上显得“力不从心”,于是三十二位的驱动卡也就随之而诞生。我们这里介绍的 VESA LOCAL BUS 三合一显示卡就是其中产品之一。

### 一、简介

#### 1、特性

VESA Local Bus 三合一显示卡(以下简称 VL-Bus 三合一卡)虽称“三合一”但其功能却远不止三种。

①提供基于 Cirrus Logic 5426 Graphic Engine 的显示功能,基于 APPIAN ADI2 芯片的 IDE 功能,以及对于 VL-Bus 标准与 AT 总线的多个 I/O 端口。

②与 IBM VGA 显示 100%兼容。

③支持 VESA 标准的下列显示方式:

1024×768 256 颜色

800×640 256 颜色

640×480 256 颜色

④支持:

800×600/32K 64K 颜色

640×480/32K 64K, 16.7M 颜色

1024×768/64K 颜色

⑤对于隔行扫描模式,支持高达 1280×1024/256 颜色的分辨率。

⑥板上存储器标准配置 1MB,可以扩展至 2MB。

⑦适用于显示与 IDE 的 VESA Local 总线。

⑧适用于多个 I/O 端口(2 串/1 并/1 游戏接口)的 AT 总线。

⑨对多数常用软件提供相应的驱动程序,使其能够运行或者使本来已经能够运行的软件得到更好的运行效果。

#### 2、电路图。

图 1 是 VL-Bus 三合一卡的逻辑电路图。

#### 3、跳线设置。

为了方便读者使用,我们把 VL-BUS 三合一卡的跳线功能翻译如下,不同厂家生产可能有所不同,请参考相应

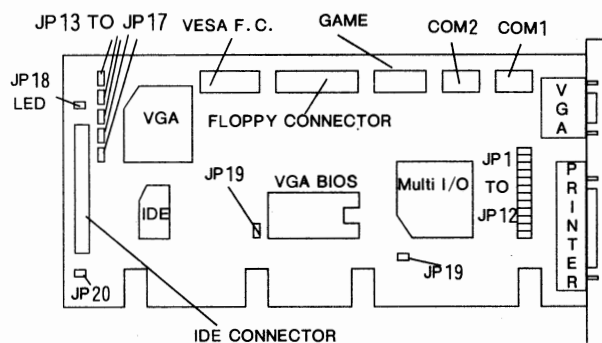


图 1

的技术手册。(注:带“\*”号的项表示缺省设置)

跳线开关	设置	功能
JP1	* 1-2 脚	FDD 允许
	2-3 脚	FDD 禁止
JP3	1-2 脚	HDD IDE 允许
	* 2-3 脚	HDD IDE 禁止
JP4	* 1-2 脚	S101 允许
	2-3 脚	S101 禁止
JP6	* 1-2 脚	S101 端口-2F8-2F (COM2)
	2-3 脚	S101 端口-2E8-2EF (COM4)
JP7	* 1-2 脚	PRT 允许
	2-3 脚	PRT 禁止
JP8	* 1-2 脚	S100 端口-3F8-3FF (COM1)
	2-3 脚	S100 端口-3E8-3EF (COM3)
JP9	* 1-2 脚	FDD 端口-3F0-3F7
	2-3 脚	FDD 端口-370-377
JP11	* 1-2 脚	PRT 端口-378-37F (LPT2)
	2-3 脚	PRT 端口-278-27F (LPT3)
JP12	* 1-2 脚	S100 允许

电子工业出版社

广州科技公司

电话: 7588476 7588494

	2-3 脚	S100 禁止
JP13	1-2 脚	Device ID=32h 或 33h
	* 2-3 脚	Device ID=30h(缺省值)或 31h
JP14	1-2 脚	Device ID=31h 或 33h
	* 2-3 脚	Device ID=30h(缺省值)或 32h
JP15	* 1-2 脚	IDE 端口-003xh
	2-3 脚	IDE 端口-00Bxh
JP16	* 1-2 脚	486 模式
JP17	* 2-3 脚	486 模式
JP18		IDE 指示灯
JP19	连接	VGA IRQ9 允许
	断开	VGA IRQ9 禁止
JP20	* 1-2 脚	VL-Bus IDE 允许
	2-3 脚	AL-Bus IDE 禁止
JP21	* 1-2 脚	并行口:IRQ7
	2-3 脚	并行口:IRQ5

## 二、安装

VL-Bus 三合一卡需要与 32 位 VL-Bus 扩充槽连接。安装完毕后,需用 SETUP 对机器系统重新设置以便使该卡能够工作。

对于 AT 以及 VL-Bus 三合一卡兼容的 SETUP 程序,应该设置成 VGA 模式,如果没有 VGA 模式,则选择 EGA 模式。

对于有些系统,VL-Bus 三合一卡与 EGA 卡相似,如果没有 EGA 选择项,可以设置成以下项目:没有监视器(No Monitor),保留(RESERVED)或特殊(SPECAIL)。

## 三、实用程序 CLMODE. EXE 的使用

VL-BUS 三合一卡所附的磁盘中,提供了一个 CLMODE 的实用程序,CLMODE. EXE 用于根据实际安装情况设置监视器的类型,以及所支持的各种显示模式和刷新频率,以产生高质量的输出效果。

CLMODE. EXE 可用两种方式调用:

第一种方式是在 CLMODE 的菜单状态下进行设置。在 DOS 提示符下打入 CLMODE<Enter>,屏幕将显示出 CLMODE 的主菜单,即可选择有关项目进行设置。

第二种方式是使用 CLMODE 的命令进行设置。在 DOS 提示符下直接打入命令行来设置监视器的类型,显示模式和刷新频率。命令行的格式如下:

CLMODE [[modenum][+ \* -][m[montype][refresh]]

其中:

[modenum]表示模式代号

[+ \* -]中,+表示选择 400 线(缺省值)

## 广州袖珍计算机技术服务中心

地址:510080 广州市东风东路 745 号

电话:(020)7662683 FAX:(020)7758117

\* 表示选择 350 线

-表示选择 200 线

[montype] 表示监视器类型

[refresh] 表示高速(+)或低速(-)VGA 刷新频率

例如:CLMODE 3+ m2+

如果打入无效的命令,将会显示命令帮助信息以供参考,而打入命令 S 则显示当前的 CLMODE 设置。

## 四、在 Windows 环境下美化显示效果

在 Windows 环境下使用 VL-Bus 三合一卡所提供的驱动程序,可以使文本或图形显示的清晰度得到提高。Windows 下显示驱动程序的安装请参考 windows 的安装手册,这里不再赘述。

显示驱动程序安装完毕后,可以使用 SetRES 实用程序,用以选择显示分辨率、颜色以及字符集(Windows3.0 不允许改变字符集)。

SetRES 的安装步骤如下:在 Windows 的“Program Manager”选择“File”中的“Run”。如果存放 Windows 驱动程序与实用程序的磁盘插在 A 驱动器,打入 A:\INSTALL,并指明把有关文件复制到 Windows 子目录。

## 五、VL-Bus IDE 磁盘接口与 Multi-I/O 端口

1、VL-Bus 三合一卡提供 32 位数据通道,在 VL-Bus 主板上能以 50Hz 与 0 等待的方式运行。为优化 IDE 硬盘的运行,VL-Bus 三合一卡配备 2 个驱动程序。下面说明它们的使用方法。

ADI2CXXX. SYS:其中 XXX 是软件版本号。将 ADI2CXXX. SYS 复制到硬盘某个子目录上,然后在 CONFIG. SYS 文件中加入命令行:

DEVICE=C:\directory\ADI2CXXX. SYS

其中 DIRECTORY 是 ADI2CXXX. SYS 所在的目录。

ADI2WXXX. 386:该文件在 Windows3.1 的操作环境下工作。要使用该文件,必须按如下所述修改 C:\WINDOWS\SYSTEM. INI 文件:

①把其中的“Device = \* wdctrl”改为“device = A-DI2W. 386”。

②加入“32BitDiskAccess=on”使系统能够进行 32 位磁盘操作。

2、VL-Bus 三合一卡还提供有以下几个接口:两个全功能通用异步接收发送器,一个软磁盘控制器,一个并行口与游戏口。它们具有如下特性:

①与 IBM PC,XT 和 AT 体系 100%兼容。

②与 16C450 通用异步接收发送器兼容。

③与 82C765B FDC 工业标准 100%兼容。

④双向并行口可用于打印机或扫描器。

# 单片机与模糊控制讲座

## 第十讲 用 MC6805R3 控制的模糊洗衣机(下)

广东工学院 余永权

### 三、洗衣机物理量检测

洗衣机在洗衣过程中起决定作用的物理量有布质、布量、浑浊度、水温等四种。这些物理量都需要采用一定的方法检测出来,并且转换成单片机 MC6805R3 能接受的形式送入单片机中,才能进行处理和执行模糊推理。

#### 1. 浑浊度的检测

衣物的脏污程度、肮脏性质和洗净程度等都需要检测,以便进行工作过程的整定的控制。浑浊度的检测是采用红外光电传感器来完成的。利用红外线在水中的透光率和时间的关系,通过模糊推理,以得出检测的结果而这个结果就可以用于控制推理。

浑浊度检测器的结构和安装情况如图 4 所示。红外发射管和红外接收管分别安装在排水管的两侧;在红外发射管中通以一定的恒定电流,使红外线的一定的强度发射,红外接收管中接收到的红外强度反映了水的浑浊程度。

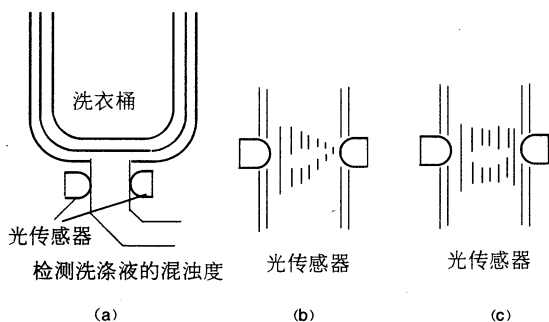


图 4

在图 4 中,(a) 表示了红外光电传感器的安装情况;(b) 表示水的浑浊度较高时的红外线透光率变低的情况;(c) 表示水的浑浊度较低时,红外线透光率较好的情况。

根据红外接收管所接收到的红外线强度,就可以得出水的浑浊度。通过实际,可知在洗涤中红外线透光率的变化情况以及有关因素的关系。这种关系如图 5 所示。在图 5 (a) 中给出了洗涤开始到漂洗结束的整个过程透光率的变化曲线。从曲线看出,随着洗涤的开始,衣物中的脏污物溶

解于水,使透光率下降;同时随着洗涤剂的投入,衣物中的污物进一步溶解和脱落于水,故透光率进一步下降,并达到一个最低值;然后,随着漂洗的进行,衣物变干净,水质也变清,从而使红外透光率渐渐升高,最后达到初始值。一般而言,当透光率再次达到初始值时,说明衣物洗涤干净,这时可以停止漂洗。

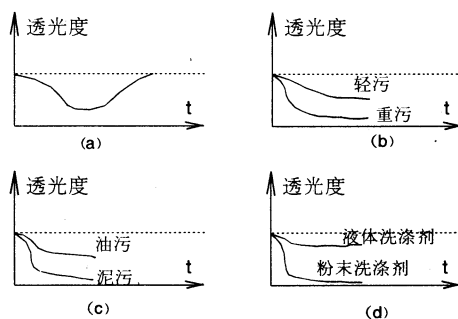


图 5

图 5(b) 表示了衣物轻度脏污和重度脏污,进行洗涤时红外线透光率的变化曲线。重污时透光率较差;轻污时,透光率较高。利用这种特别可以别衣物的脏污程度。

图 5(c) 表示了衣物的脏污性质。油污时透光率较高,泥污时透光率较低。

图 5(d) 表示了洗涤剂的类型。液体洗涤剂透光率较高,粉剂洗涤剂透光率较低。

按照图 5 给出的透光率曲线,就可以根据洗衣机中水的透光率来判别衣物的脏污程度、脏污性质以及洗涤剂的种类。从而可以按此去进行洗涤过程控制。

#### 2、布量和布质的检测

布量和布质的检测是在洗涤之前进行的。在水位为一定的时候,布量和布质的不同就会产生不同的布阻抗。通过

**华南计算机公司**

地址:广州市沙河太公路京溪桥 电话:(020)7705696

给定一定的水位,然后在这个给定水位的条件下使主电机进行间断旋转,则不同布阻抗就会使主电机制动的性能不同,利用主电机在不同布阻抗时的制动特性就可以推断出布质和布量。不同布质和布量的布阻抗如图6所示。从图中可知:硬质布的布阻抗较高,软质布的布阻抗较低。但两者有同样布阻抗的区间。

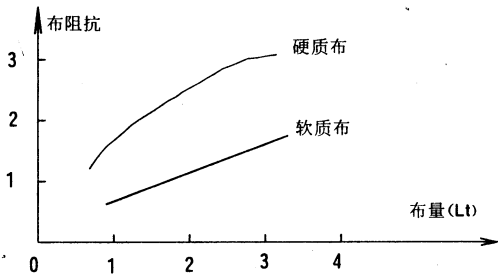


图6

在进行布质和布量检测时,首先注入一定的水位,然后启动主电机旋转,按着断电让主电机以惯性继续运转直到停止。在主电机断电的时间内,由于主电机的惯性,所以它处于发电机状态;并且会产生感应电势输出。随着布阻抗的大小不同,主电机处于发电机状态的时间长短不同。只要检测出主电机处于发电机状态的时间长短,就可以反过来推理出布阻抗的大小,当然,主电机发电时间长,布阻抗就小,主电机发电时间短,布阻抗就高。

通过去主电机的正反转控制绕组输出电势的整流和检测,由光电隔离后形成脉冲信号送入单片机,单片机只要计算出主电机在停电时产生的计数脉冲个数就可以知道布阻抗的大小,脉冲个数多,电阻抗小,反之亦然。

在得出电阻抗之后,通过模糊推理就可以产生相应的布质及布量。布质布量检测电路见图1的PA<sub>2</sub>端口所接电路。

### 3. 水温检测

水温检测由温度检测器 MTS102 执行。由于 MTS102 有线性度好,对温度敏感等特点故采用它对常温检测有较大的优点。在电路中,采用两个运算放大器对 MTS102 的输出信号进行处理,一个用于隔离阻抗,一个用于信号放大。这个电路结构如图1的AN<sub>1</sub>端所接的电路。水温一般为4℃—40℃;在一些特殊的洗衣机中,有时会加入热水,则水温较高。但水温一般不会超过60℃。因为,水温太高对衣物有一定的影响。

## 四、控制软件框图

控制软件由主程序、各种子程序和中断服务程序组成。

主控程序如图7所示。所有模糊推理在洗涤之前都基本执行完毕。所以,在程序,一般判别出是启动之后,就开始进行一系列的检测工作和推理工作。在推理工作完毕之后,就开始进入洗涤过程。

在洗涤过程中如果产生故障则系统会执行报警。

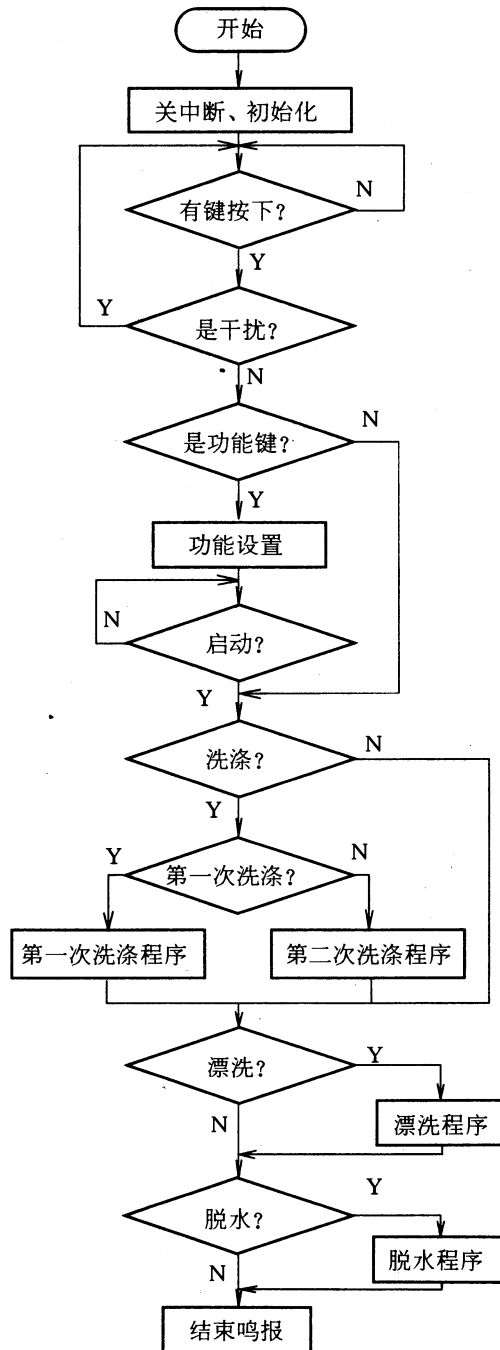


图7

人算不如电算  
EASY IS EASY  
拓展财务网络软件 EASY  
电话:(020)7501451



# 《巧用 FoxBASE 中的宏命令》 一文中存在的漏洞

贵州风华机器厂 李德文

贵刊 1993 年第 11 期 31 页发表了宋捷同志的文章《巧用 FoxBASE 中的宏命令》(以下简称为《巧》)。通过细读原文,发现了其中的不少地方存在着漏洞和不妥。为正读者尤其是初学者视听起见,特列举于后,供作者及读者诸君参阅。

## 1. 何苦自造“宏命令”

《巧》文中,把 KEYBOARD 和 SET FUNCTION 称为“宏命令”,自造了一个并不十分合适的“术语”。在 FoxBASE 中,语法结构通常有三个部分,即:命令、函数和语句。在命令中,KEYBOARD 被称为“填充键盘缓冲区命令”,SET FUNCTION 则被称为“设置功能键命令”完全没有必要自造一个“宏命令”。更何况“宏命令”一词很容易同宏代换函数 & 相混淆。

## 2. “机与机对话”并非编程成功与否的关键

《巧》文称,人们在进行程序设计时,通常把重点放在有良好的人机对话界面上,而忽略了编程成功与否的关键——机与机对话的程序设计。系统开发,之所以要设计人机对话,其原因通常在于当程序运行到对话点后,程序已经无法预知操作员要进行后续内容的哪一类操作,或者某些参数已经无法由前趋程序推知。之所以要设计良好的人机界面,往往是为了让操作员在控制程序走向的同时不感到费解和枯燥。

《巧》文所述的“机与机对话”,实际上是在说让机器自动运行而无需人工干预。诚如是,则在这些地方提供人机界面,本身已属于程序设计的不足。再者,既然对话,必有双方,试问,除了正常运行的程序外,另一对话方是谁呢?

## 3. 功能键 F1 不能设置

《巧》文称,SET FUNCTION n TO “填入缓冲区的数据”命令中 n 的取值范围为 1 至 10。

实际情况是,自 dBASE III 至 FoxBASE,功能键 F1 都是用来作为专用的帮助键。未曾看见过能对其设置的资料和试验结果。再者,这些功能键的使用和设置,通常都是在点状态下进行的,将其移至命令文件里,虽有新意,但也有自立“山头”之嫌——毕竟与别人的习惯都不相容了。

## 4. 清屏操作何必“父债子还”

《巧》文程序中有两处屏幕菜单。然而在菜单之前,都没有于循环体内清除屏幕。这样,当执行循环时,只得依靠其

子程序来清屏。姑且不论这使得每个子程序的最后都不得不增加清屏命令,仅从主程序把一个可有而更应可无的“待清屏”信息传送给子程序,也是很合理的。

## 5. 运行时间何以知

《巧》文中,程序主菜单给出了报表计算和打印时间实为不妥。诚然,作者的本意无非是想用来当作提示信息。但是,严密的程序里最好不要这样做。因为程序的运行时间与主机频率、总线结构、内存占用情况等都有关系,难以准确定量。

## 6. 变量 XZI 究竟是几位

由于程序清单经过排版,已难以看出 XZI 的初值为一个还是两个空格。然而,仔细分析程序发现,无论 XZI 的初值为几位,在程序运行过程中都可能会出现问題。

若 XZI 初值为一位,那么字符串“10”就无法输入,dy10.prg 当然也就无法运行了。若 XZI 初值为两位,那么程序中所称的“报表批处理”(实际上是程序的演示)将无法按文中所述运行。况且,输入字符“1”至“9”时,若输成了空格十数字字符,又如何能经过宏代换执行程序 dy1.prg 至 dy9.prg 呢?

## 7. note 2 部分无法执行

《巧》文称,note 2 部分与 note 1 部分具有相同的功能,其实不然。使用 SET FUNCTION 命令,能把字符串设置进入 F2 功能键中,但并不能压一下 F2 后就把相应的信息送入键盘缓冲区内。这样,“机与机对话”自然就无法实现了。不信你试试看。

专业写作,错误难免,尤其是充满自信的字子,提笔不久者和长于实践而短于推理的工程技术人员。但是,起码的这一点要做到,睁着眼睛说瞎话和想当然的毛病要克服。无论怎样说,如此之多的漏洞和不妥集中于一篇短文中,除作者外,审稿者和编辑部看来都不得不承担一点“文责”了。

305

AOK 彩显

王者之尊 傲视同侪

广利电脑设备厂 电话:8895924,8895934 传真:8895943

# 图形用户接口中国标(ICON)的设计

♠ 江西军区自动化工作站 马小蕙      南昌陆军学院 葛伟华 ♠

**【摘要】**功能强、具有吸引力的图形用户接口成功因素之一在于使用了图标,屏幕设计过程中图标的使用已经成为用户接口范围内最热门的话题之一。本文在目前流行的窗口管理程序用户接口的表现属性基础上,分析了一般图形用户接口中国标的特征和特点。并且结合X环境下程序设计的特点,给出了图标设计的四项原则和具体的实现步骤。

由 Xerox PARC 研究中心提出的图标应用技术,作为代表应用程序的一种图形标识,已经在当代图形系统中得到广泛的使用,其中最具有代表性的系统是: Xerox Star(图标首次使用), Apple Macintosh, MIT X window, SUN NeWS 和 Microsoft Windows。在这些系统中图标化的用户接口已经成为其主要的接口特性。图标和其它标识符号一样可以代表某种系统功能或者操作,因此它已经成为图形系统表现特征的一部分,是衡量一个图形用户接口好坏的重要标准。

## 一、图标的特征

图标用来在菜单型显示中代表系统功能,使用这种接口技术,无论对有经验或没经验用户,学习时间将会减少,操作变得简单。所以,在一般的图形用户接口的表现属性中,支持一系列与图标有关的操作功能,我们从中可以得出一些图标所具有的特征。

### 1. 图标的表示特征

图标的表示特征主要体现在两个方面:

★ 图标可以表示操作对象(Operate Object)

在一些图形系统中,例如 Xerox Star, Apple Macintosh, 利用图标来表示每个操作对象(如:文件,文件夹,文件柜和打印机等),以图形建立一个视觉印象来模拟用户日常的经验。系统的操作是利用光标取出对象并移动它们,使得用户可以通过借鉴现实世界中的工作方法来学会如何操作使用这些对象。例如,为了删除某一个不需要的文件,用户可以通过把代表该文件的文件夹图标移动到类似于废物箱的图标即可。

★ 图标可以表示应用窗口(Application Window)在大多数的窗口系统中,例如 MIT X Window 和 Sapphire 等,以不同的方式利用图标来表示某一类窗口或应用,使得窗口系统的屏幕显示整洁明了,便于用户更好地组织和安排自己的工作。

### 2. 图标的映射特征

图标作为窗口的映射,用户可以利用图标向所对应的消隐窗口发送命令。并且图标与其它窗口可以同时出现在屏幕上。

### 3. 图标的窗口特征

图标本身也是一个微缩的标识窗口,或者说是根窗口的子窗口,窗口的尺寸、位置和内容是可以修改的。根据这个特征,图标可以分为静态和动态两种。

### 4. 图标的接口特征

图标作为用户接口的一种表现方式,可以看作是索引和表示符号的混合体,在系统图形字典中建立了对应的条目,因此它也具有句法规则,词法规则,语义规则和实用性规则等接口属性。

### 5. 图标的菜单特征

图标作为菜单的图形表现方式比一般的弹出式菜单或下拉式菜单更为直观,使用起来更为方便。而且图标菜单使得用户容易产生功能联想,便于学习与记忆。

## 二、图标的优点

为什么图标符号在目前的计算机图形系统中非常流行,部分原因在于其实用性好。一个图标化的图形用户接口可以使一个杂乱无章的用户屏幕显示变得清晰整洁,便于用户在图形系统中更好地工作。图标具有以下特点:

1、图标是吸引人的,清晰的和有利于观看的。并且具有良好的视觉效果,学习使用容易。

2、图标作为符号相比文字所需花费的象素和占用的屏幕显示空间都很少,但是仍需要文字背景。

3、图标符号可以替代自然语言,有利于接口的协同设计和理解。

4、图标符号可以增强或辅助用户理解文字形式的信息。

5、图标符号的适应性强,适合于访问系统和命令界面的各种用户,以及计算机发起的对话。

6、图标符号语言独立性强,编程相对比较容易。

7、新的图标符号的引用必须经过研究,设计,测试和向市场及用户介绍推销四个步骤,周期太长。

8、由于缺乏新型的图标设计语言,使得图标设计的标准不易掌握,设计的规范性差。

## 三、图标设计的原则和实现步骤

图标设计的原则概括起来有四点:一致性、清晰性、简

单性和熟悉性。这要求在设计过程中代表同类操作的图标其设计风格和显示图案应保持一致；在正常的工作距离内，用户能区分出代表不同操作的图标符号；图标中的图案尽量简单，应与操作功能相关，使用户能产生联想。

如何设计一个好的图标，目前还缺乏一个统一的标准，但是有一些建议可以供用户参考：

- ★ 让用户来测试图标的含义。

- ★ 使用的图标尽可能逼真。

- ★ 图标应有一个清晰的轮廓，以利于辨别。

- ★ 当显示命令时，要给出在此命令下操作对象的具体表现。

- ★ 要避免使用符号，除非其意义是众所周知的。

在 X 窗口系统中，图标是一种单平面的像素图 (Pixmap) 或者说是位映象图 (Bitmap Image)，因此在 X 系统下，用户设计图标包括以下几个步骤：

### 1. 产生一个图标位图 (Icon Bitmap)

在 X 环境下，用户可以通过调用 `XCreatePixmap()` 函数或使用 X 窗口系统的 Client Bitmap 来产生图标的位图映象。在实际应用过程中，大多数的用户都是利用 Bitmap 来产生标准文件格式的位图，保存数据到位图文件中，然后在用户 Client 过程中调用位图文件，将其转换成像素图，即可告之窗口管理程序作为用户图标使用。

Bitmap 格式: Bitmap Filename [width×height]

利用 Bitmap 产生的位图数据是与 C 语言中定义的静态数据相对应的，请参阅第四部分 Example 1。

### 2. 位图数据转换 (生成 Pixmap)

利用 Xlib (X. 11) 库函数 `XCreateBitmapFromData()` 和 `XReadBitmapFile()` 可以在应用过程中引用位图文件，并把原始数据转换成 X 的单平面的像素图 (Pixmap) 数据。

```
Pixmap XCreateBitmapFromData (display, drawable, data, width, height)
```

```
Display *display;  
Drawable drawable  
char *data;  
unsigned int width, height;
```

```
int XReadBitmapFile (display, drawable, filename, width,  
height, pixmap, x-hotspot, y-hotspot)
```

```
Display *display;  
Drawable drawable;  
char *filename;  
unsigned int *width, height;  
pixmap *pixmap;
```

```
int *x-hotspot, y-hotspot;
```

### 3. 修改窗口的属性

一旦用户利用 `Create-included-pixmap` 过程或者 `Load-Bitmap` 过程产生了图标像素之后，就必须告之窗口管理程序使用图标作为用户应用程序的图形标识。当用户产生顶层应用窗口时，可以设置窗口 Hints 属性中的对应区域为使用图标所需的状态值，为实际使用过程中图标化应用窗口做准备。应用程序的 Hints 属性是窗口管理程序

Hints 属性 WMHints 的具体实现，其数据存储在全局 XWMHints 结构中。应用程序必须设置的 Hints 属性最小集合包括：

★ 窗口名	★ 图标名
★ 图标像素图	★ 命令名和参数
★ 参数的个数	★ 用户所偏爱的窗口尺寸

```
typedef struct {  
    long flags; /* which files are used */  
    Bool input;  
    int initial-state;  
    pixmap icon-pixmap;  
    Window icon-window;  
    int icon-x, icon-y;  
    Pixmap icon-mask;  
    XID window-group;  
} XWMHints;  
XWMHints wmhints;  
Display *display;  
Window window;  
pixmap bitmap;  
wmhints.flags = IconPixmapHint | stateHint;  
wmhints.icon-pixmap = bitmap;  
wmhints.state = IconicState;  
XSetWMHints (display, window, & wmhints);
```

### 4. 图标的使用

图标设计和使用的最后步骤是在根窗口状态下，应用程序通过 `XIconify Window()` 函数和 `XSendEvent()` 函数发送一个特殊的消息 (Client Message 事件) 给窗口管理程序，告之用户程序将使用图标来缩微应用窗口。应用程序与窗口管理程序之间传送的消息类型为 `Atoms(32bit)`，而有关函数和事件的结构如下，应用实例参阅第四部分 Example 2。

```
status XIconify Window (display, window, screen)
```

```
Display *display;  
Window window;  
int screen;
```

```
typedef struct {  
    int type; /* ClientMessage */  
    unsigned long serial;  
    Bool send-event;  
    Display *display;  
    Window window;  
    Atom message-type;  
    int format;  
    Union { char b[20];  
            short s[10];  
            long l[5];  
        } data;  
} XClientMessageEvent;
```

### 四、图标设计 and 应用实例

Example 1: Create a pixmap from include data and from data read from the bitmap file.

```
#define icon-bitmap-width 16  
#define icon-bitmap-height 16  
static char icon-bitmap-bits[] = {  
    0x0f, 0xf0, 0x07, 0xe0, 0xf0, 0xf0, 0x1d, 0xb8,  
    0x38, 0x1c, 0x70, 0x0e, 0xe0, 0x07, 0xc0, 0x03,  
    0xc0, 0x03, 0xe0, 0x07, 0x70, 0x0e, 0x38, 0x1c,
```

# FoxBASE+动态索引技术的应用

江西省宜春地区农行干校 彭福林

在我们开发应用程序时,都需要按条件检索记录,并对符合条件的记录进行统计。建立一个或几个固定索引文件,这对于此项工作无非是很有用处的,也是开发过程中必须的,这可以提高应用程序的速度和效率,但这种索引是有限度的,当我们要对不在索引字段中的字段进行检索时,特别是在此种条件下要做很多种操作,这时这种索引就显得毫无用处了。比如在“人事档案”中对“职称或职务”的统计等。建立一个临时的动态索引,就显得很有必要了。

FoxBASE+的 INDEX 语句可以为我们实现动态索引,此语句的格式如下:

**INDEX ON<KEY>TO<FILE>[FOR<EXPL>] [UNIQUE]**

<KEY>是一个字段或一组字段,它存在于当前使用的数据库中,可以用“+”运算将多重的字段作联结,在数据库中的子字符串可以用来设置成<KEY>或是<KEY>字段的一部分,数值及日期型的字段可以用 STR 或 DTOC 函数作联结,逻辑型以及备注型的字段不可以设置成<KEY>字段;<EXPL>是条件串,它是可选项,如果使用则该当前数据库索引就如同判断条件一般,这也叫条

件索引;<FILE>是索引文件名。

实现动态索引还必须将<KEY>和<EXPL>生成可用宏替换(&)调用的字符串,将此字符串前面加宏替换符放入 INDEX 语句中,一旦执行了此语句,当前数据库就按此串及条件进行了索引,这个字符串的不断变化,索引也随着在变化,这就是 FOXBASE+的动态索引技术。当此索引不再需要时,用 SET ORDER TO 0 命令就可以使此索引不再起作用。值得一提的是,在动态索引中使用了条件索引,将会在很大程度上加快处理速度,从而提高程序的执行效率。

本人应用此技术成功地开发了一个信息检索系统,它就象一本字典一样,可以随心所欲地检索出您所需要的信息,只要您能对这些信息进行科学的分类。分类类型分为四个级别,可以设置成树型或网状型等结构。它适用于日常工作的资料存储检索或档案工作等。如企、事业单位的“文件”管理、产品简介、图书资料检索等。(感兴趣的读者可与本人联系,江西宜春地区农行干校 336000)

```
0xid,0xb8,0x0f,0xf0,0x07,0xe0,0x0f,0xf0
};
create--included--pixmap(icon--pixmap,width,height)
pixmap *icon pixmap;
unsigned int *width,height;
/* Creating a pixmap from included data */
if(( *icon--pixmap=XCreateBitmapFromData(display>window,
icon--bitmap--bits,width,height))
==False)
{ return((pixmap)None); }
else
{ return(*icon--pixmap); }
} /* Create--included--pixmap */
pixmap Load--Bitmap(display>window,filename)
Display *display;
Window window;
char filename[];
/* Creating a pixmap from data read from a file */
int x--hot,y--hot;
unsigned int width,height;
pixmap bitmap;
int status;
status=XReadBitmapFile(display>window,filename,& width,
& height,& bitmap,& x--hot,& y--hot);
if(status==BitmapSuccess)
```

```
{ return(bitmap); }
else
{ return((pixmap)None); }
} /* Load--Bitmap */
Example 2: Iconifying application window
XClientMessageEvent event;
Window rootwindow;
.....
/* Get root winow */
rootwindow=RootWindow(display,DefaultScreen(display));
.....
/* Fill up our event structure */
event.type=ClientMessage;
event.xclient.display=display;
event.xclient.window=window;
event.xclient.send--event=True;
event.xclient.message--type=wm--change--state;
event.xclient.format=32;
/* 32--bit(Atoms) */
event.xclient.data.i[0]=IconicState;
Status=XSendEvent(display,rootwindow,False,
SubstructureRedirectMask,SubstructureNotifyMask,&event);
XFlush(display);
.....
```

# EMS 与 C 的文件型接口系统

中山嘉华电子科技研究中心 杨震

早期的 PC 机只能寻址 1M,而这 1M 中能给用户使用的只有低 640K,高 384K 用于多种目的,如视频适配器,硬盘控制器,ROM BIOS 等等。现在,由于硬件的升级和软件的支持,PC 机已经拥有两大内存来源:EMS(Expanded Memory Specification),XMS(Extended Memory Specification)。

XMS 是通过扩大寻址能力而得到的内存,它位于 1M 的上面,显示这是“正宗”的主存。EMS 是在 XMS 出现之前通过加装内存板来实现的,它并不占用主存的地址空间,而是通过映射来访问的。本文只介绍 EMS,因为它出现得早,是由 Lotus,Intel,Microsoft 三家公司联合开发的,有大量的软件支持,当没有 EMS 时,还可用 EMM386 将 XMS 模拟成 EMS。

EMS 是通过映射来存取的,比较复杂,这也是为它做一个接口的原因;可以使用户完全不必了解 EMS 的任何细节,只要会用文件就可以使用 EMS。

本系统包括一个头文件 EMM.H 和一个库文件 EMM.LIB,EMM.LIB 是 EMM.H 中的 7 个函数分别编译成 OBJ 文件,然后再用 TLIB.EXE 程序把这些 OBJ 文件组合成 EMM.LIB。用户只需在自己的程序中加上 #include“emm.h”,再把 EMM.LIB 加入工程文件.PRJ 中就可以使用本系统提供的函数了。

下面将逐个说明函数的使用方法:

(1)int far echeck(void);

功能:检查 EMS 是否存在,硬件和驱动程序是否正常;

输入参数:无;

返回参数:无;

函数返回值:00H=正常;

00H=EMS 不存在;

EMS 错误代码。

(2)int far esize(unsigned long far \*avail,unsigned long far \*total);

功能:取 EMS 中总的字节数和可用的字节数;

输入参数:无;

返回参数:avail=可用的字节数;

total=总的字节数;

函数返回值:00H=正常;

EMS 错误代码。

(3)int far eopen(int far \* handle, unsigned long length);

功能:打开一个文件,也就是分配一段 EMS 内存;

输入参数:length=所需的内存字节数;

返回参数:handle=文件句柄;

函数返回值:00H=正常;

EMS 错误代码。

(4)int far eclose(int handle);

功能:关闭一个文件,也就是释放已分配的 EMS 内存;

输入参数:handle=文件句柄;

返回参数:无;

函数返回值:00H=正常

EMS 错误代码。

(5)int far esseek(int handle,unsigned long address);

功能:将指针移到指定地址;

输入参数:handle=文件句柄;

address=指定指针定位的绝对地址;

返回参数:无;

函数返回值:00H=正常;

02H=指针出界;

EMS 错误代码。

(6)int far eread(int handle,unsigned long length,char far \*buf);

功能:将文件的内容读至缓冲区;

输入参数:buf=缓冲区首地址;

handle=文件句柄;

length=字节数;

返回参数:无;

函数返回值:00H=正常;

EMS 错误代码。

(7)int far ewrite(int handle,unsigned long length,char far \*buf);

功能:将缓冲区的内容写至文件;

输入参数:buf=缓冲区首地址;

handle=文件句柄;

length=字节数;

返回参数:无;

函数返回值:00H=正常;

EMS 错误代码。

还有一个函数 efind()是接口内部使用的,用户不必调用。

另外,本系统的错误代码包含了 EMS 的错误代码,现解释有关代码如下:

80H=EMS 软件内部出错;

81H=EMS 硬件故障;

83H=非法句柄;

84H=应用程序请求的功能未定义;

85H=无可调用句柄;

86H=保存或恢复映射关系时出错;

87H=请求分配的 EMS 超过系统总量;

88H=请求分配的 EMS 超过当前可用量;

为了演示本接口的功能和实际用法,本文也是提供了一个实用程序作为范例。该范例程序的功能是磁盘拷贝,它能将软盘上的内容一次性读入 Expanded Memory,然后可多次将此内容写入其它软盘。

程序首先确定软盘的容量,再根据此容量用 eopen()申请文件,若成功,则将软盘的内容全读入文件(Expanded Memory),最后将文件中的内容写入磁盘(可多次重复)。这对同时需要多份拷贝的情况尤为适用。程序很简单,只是注意一下 swapdisk()函数,这个函数独立完成磁盘和文件之间的数据交换。

# 实现图形局部放大的简单有效方法

中国矿业大学建筑系 余索

在一些 CAD(如建筑结构 CAD)程序的设计中,经常要对屏幕上显示的图形进行局部放大,使用户能够更清楚地观察图形的一些细部。

一般的图形局部放大实现流程是这样的:

- 1、确定图形局部放大的范围。
- 2、图形裁剪,以从整体图形中取得需放大的局部图形。
- 3、坐标变化,把图形从局部裁剪范围内的坐标系中变换到屏幕坐标系,从而实现图形的局部放大。

但是,用这种放法实现局部放大对于象建筑结构 CAD 等有非常复杂仔细的图形的软件来说,如果用一般的计算机图形学的裁剪算法来实现,将使程序的数据结构变得复杂,程序运行速度下降的问题。

笔者在实践中发现利用 Quick BASIC 的“WINDOW”语句能够很方便地实现图形的局部放大。

利用 WINDOW 语句,可以定自己的坐标系,从而不受物理坐标系的限制,当执行 WINDOW 语句不带任何参数时,屏幕恢复正常物理坐标系。而且,WINDOW 语句能自动地进行图形裁剪,从而在程序设计中不必进行这一项工作,大大地简化了程序。

如需要放大的范围是:(x1,y1)~(x2,y2),其中(x1,y1),(x2,y2)分别是矩形范围的左上角和右下角,执行“WINDOW SCREEN (x1,y1)~(x2,y2)”后屏幕坐标被定义为左上角为(x1,y1),右下角为(x2,y2)的逻辑坐标系了。当再执行“WINDOW”语句后,屏幕坐标恢复以(0,0)为左上角的物理坐标系。

本文给出图形局部放大的源程序,限于篇幅,子程序收集在本期的程序盘中。其中 egdraw()是图形输出子程序,其流程是:

- 1、显示十字光标并移动定位(discross(),movloccross())
- 2、获取需放大的范围(getgraph())
- 3、产生新的逻辑坐标系(makenewcoor())
- 4、设置屏幕为逻辑坐标系并输出图形(endraw())
- 5、返回 1。

当用 egdraw()和 enlarge()组成程序时,更成为一个递归程序,对图形进行逐步深化的放大。(键入 Esc 退出程序)

```
DECLARE SUB makenewcoor (length %, high %, crossx %, crossy %,

```

```
enlstartx!, enlstarty!, enlendx!, enlendy!)
DECLARE SUB getgraph (cinfor() AS INTEGER, lcinfof() AS INTE-
GER, crossx %, crossy %, high %, length %)
DECLARE SUB movloccross (infof() AS INTEGER, cinfor() AS IN-
TEGER, lcinfof() AS INTEGER, crossx %, crossy %)
DECLARE SUB discross (infof() AS INTEGER, cinfor() AS INTE-
GER, lcinfof() AS INTEGER)
DECLARE SUB egdraw ()
DECLARE SUB endraw (sx!, sy!, ex!, ey!)
DECLARE SUB setcross (crx %, cry %)
DECLARE SUB enlarge ()
DECLARE SUB pointxor (x %, y %)
DECLARE SUB linexor (x1 %, y1 %, x2 %, y2 %)
DECLARE SUB rectanglexor (xs %, ys %, xe %, ye %)
DIM SHARED pointimage(3) AS INTEGER
REM *** Main programm ***
CALL egdraw
CALL enlarge
END
```

```
SUB egdraw
SCREEN 12
LINE (100, 200)~(500, 200), 12
FOR i% = 1 TO 15
CIRCLE (150 + i% * 20, 202), 2, 14
NEXT i%
END SUB
```

```
SUB endraw (sx!, sy!, ex!, ey!)
CLS
WINDOW SCREEN (sx!, sy!)~(ex!, ey!)
CALL egdraw
WINDOW
END SUB
```

```
SUB enlarge
DIM infof(186) AS INTEGER
DIM cinfor(186) AS INTEGER
DIM lcinfof(124) AS INTEGER
STATIC enlstartx!
STATIC enlstarty!
length % = 0: high % = 0
CALL discross(infof(), cinfor(), lcinfof())
CALL movloccross(infof(), cinfor(), lcinfof(), crossx %, crossy %)
CALL getgraph(cinfof(), lcinfof(), crossx %, crossy %, high %, length %)
CALL makenewcoor(length %, high %, crossx %, crossy %, enlstartx!,
enlstarty!, enlendx!, enlendy!)
CALL endraw(enlstartx!, enlstarty!, enlendx!, enlendy!)
DO
c $ = INKEY $
LOOP WHILE c $ = ""
```

```

IF c$ = CHR$(27) THEN
  EXIT SUB
END IF
CALL enlarge
END SUB

SUB linexor (x1%, y1%, x2%, y2%)
  xs% = x2% - x1%
  ys% = y2% - y1%
  IF xs% <> 0 THEN
    stp% = ABS(xs%) / xs% * 2
    FOR j% = 0 TO xs% STEP stp%
      CALL pointxor(x1% + j%, y1%)
    NEXT j%
  END IF
  IF ys% <> 0 THEN
    stp% = ABS(ys%) / ys% * 2
    FOR j% = 0 TO ys% STEP stp%
      CALL pointxor(x1%, y1% + j%)
    NEXT j%
  END IF
END SUB

SUB makenewcoor (length%, high%, crox%, crosy%, enlstartx!,
  enlstarty!, enlendx!, enlendy!)
  STATIC times%
  STATIC lentime!
  leng% = ABS(length%); hig% = ABS(high%)
  ln% = leng% / length%; hn% = hig% / high%
  IF leng% * 3 >= hig% * 4 THEN
    entime! = 639 / leng%
    stoe! = leng% * 3
    crosy% = crosy% - (stoe! / 4 - hig%) / 2 * hn%
  ELSE
    entime! = 479 / hig%
    stoe! = hig% * 4
    crox% = crox% - (stoe! / 3 - leng%) / 2 * ln%
  END IF
  IF times% = 0 THEN
    lentime! = 1
  END IF

  enlstartx! = enlstartx! + (crox% + 10) / lentime!
  enlstarty! = enlstarty! + (crosy% + 10) / lentime!
  enlendx! = enlstartx! + stoe! / 3 / lentime! * ln%
  enlendy! = enlstarty! + stoe! / 4 / lentime! * hn%
  IF ln% < 0 THEN
    chan! = enlstartx!
    enlstartx! = enlendx!
    enlendx! = chan!
  END IF
  IF hn% < 0 THEN
    chan! = enlstarty!
    enlstarty! = enlendy!
    enlendy! = chan!
  END IF
  times% = times% + 1
  lentime! = lentime! * entime!
END SUB

SUB movloccross (infor() AS INTEGER, cinfor() AS INTEGER,
  lcinfor() AS INTEGER, crox%, crosy%)
  speed% = 12
  senlst:

```

```

DO
  q$ = INKEY$
  LOOP WHILE q$ = ""
  SELECT CASE LEN(q$)
    CASE 1
      SELECT CASE q$
        CASE CHR$(27)
          EXIT SUB
        CASE CHR$(13)
          PUT (crox%, crosy%), cinfor, XOR
          PUT (crox% + 5, crosy% + 5), lcinfor, XOR
          EXIT SUB
        CASE ELSE
          BEEP
      END SELECT
    CASE 2
      PUT (crox%, crosy%), cinfor, XOR
      SELECT CASE RIGHT$(q$, 1)
        CASE CHR$(77)
          IF (crox% + 19) + speed% < 639 THEN
            crox% = crox% + speed%
          END IF
        CASE CHR$(75)
          IF crox% - speed% > 0 THEN
            crox% = crox% - speed%
          END IF
        CASE CHR$(72)
          IF crosy% - speed% > 0 THEN
            crosy% = crosy% - speed%
          END IF
        CASE CHR$(80)
          IF (crosy% + 19) + speed% < 479 THEN
            crosy% = crosy% + speed%
          END IF
        CASE ELSE
          BEEP
      END SELECT
      PUT (crox%, crosy%), cinfor, XOR
    END SELECT
  GOTO senlst
END SUB

SUB pointxor (x%, y%)
  IF x% >= 0 AND x% <= 639 AND y% >= 0 AND y% <= 479
  THEN
    PUT (x%, y%), pointimage, XOR
  ELSE
    PRINT "pointxor data error!"
    EXIT SUB
  END IF
END SUB

SUB rectanglexor (xs%, ys%, xe%, ye%)
  CALL linexor(xs%, ys%, xe%, ys%)
  CALL linexor(xe%, ys%, xe%, ye%)
  CALL linexor(xe%, ye%, xs%, ye%)
  CALL linexor(xs%, ye%, xs%, ys%)
END SUB

SUB setcross (crx%, cry%)
  LINE (crx% - 10, cry%) - (crx% + 10, cry%), 9
  LINE (crx%, cry% - 10) - (crx%, cry% + 10), 9
END SUB

```

# 在中国龙汉字系统中使用九针 打印机

肇庆化学纤维厂 肖永江

## 近

年来,中西文兼容的新一代汉字系统不断涌现,象中国龙、超想汉字、天汇汉字等等。这些汉字系统最大的共同的优点是几乎不用汉化西文软件就可以处理中文,解决了以前汉化西文软件带来的运行速度慢、兼容性差以及版权纠纷等一系列问题,为提高国内的软件应用水平、缩短与国外的差距提供了一个必要条件。其中的中国龙汉字系统(ACIOS)做得尤其出色,它是一个纯软汉字系统,支持西文和汉字的直接写屏显示,提供全面的西文表格识别功能,它的系统内核小,可以使用扩充内存(Expanded Memory)和扩展内存(Extended Memory),在 MS DOS 5.0、6.0 下可不占用系统的基本内存,做到“零内存运行”,并且提供了一个完善的图形接口 AGI(ACIOS Graphics Interface),为制作漂亮的图形界面提供了一个方便的工具。

由于中国龙汉字系统着眼于比较高档的硬件环境,只配有 24 针打印机驱动程序,未提供在九针打印机下的驱动程序,而九针打印机在广大的电脑用户中拥有量是很大的,使得只有九针打印机的电脑用户不能使用中国龙来输出汉字文本,这不能不说是一个遗憾。作者在分析了九针打印机的命令和中国龙汉字接口后,编写了一个仿二十四针打印机的九针打印机驱动程序,不仅解决了中国龙在九针打印机上的输出,而且使打印质量接近二十四针打印机的水平,可以改变字体、行距以及横纵向放大倍数。该程序运行后驻留于内存,此后只要向打印机输出汉字和西文字符,就可以得到仿二十四针打印机效果的文本, DOS 的重定向到打印机、CTRL-P 联机打印功能都能实现。下面就简要介绍该程序思路(全部源程序附后,用 Turbo C 2.0 编写)。

九针打印机之所以打印质量差,是因为针间距离大(1/72 英寸),幸好九针打印机提供了 1/216 英寸走纸命令(ESC 3 n),可以解决这个问题。我们设想让打印机在打完一行后走纸 1/216 英寸,再打印下一行,这就可以把针间距缩小为 1/216 英寸,为原来的 1/3,接近二十四针打印机的水平。考虑到 16 点阵字库由于点阵少,字形不精美,输出到九针打印机上不能显著改善打印效果,故选用 24 点阵字库来输出汉字和西文字符,显著提高了打印质量。

输出汉字时打印机必须工作于图形方式,而九针打印机的图形方式下分辨率有 480x8 和 960x8 两种(命令分别为 ESC M n1 n2 和 ESC L n1 n2),此时九针打印机只用九根打印针中的八针,每八针对应于一个字节数据的八个

位,如下图 1。如果选用 960×8 分辨率,则一行可打印  $960 \div 24 = 40$  个汉字,和屏幕上一行的字符数相同。由于针间距缩小为 1/216 英寸,故原来打印一列 8 个点的位置可容纳 24 点,这样就使原来打印一行西文的位置也可以打印一行 24×24 点阵的汉字来,关键是如何把点阵数据送到打印机。以横纵放大倍数均为 1 的情况为例,打印一行汉字要往打印机送三次数据。第一次把汉字点阵中每一列的对 3 取模余数为 1 的点(即图 2 中的 1,4,7,...,22 点)转化成字节数据(按图 1)送往打印机,然后走纸 1/216 英寸;第二次把汉字点阵中每一列的对 3 取模余数为 2 的点(即图 2 中的 2,5,8,...,23 点)转化为字节数据送往打印机,然后走纸 1/216 英寸;第三次把汉字点阵中每一列的对 3 取模余数为 0 的点(即图 2 中的 3,6,9,...,24 点)转化成字节数据送往打印机。这样就完成一行汉字的打印,对于横纵放大倍数不为 1 的打印可以此类推。

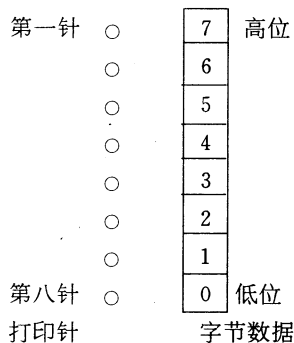


图 1

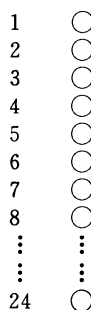


图 2

在中国龙中可通过中断 16H 的 80H 功能取出各种字体的汉字和西文字符的 24x24 点阵的字模,不过事先要运行 24 点阵字库装载程序 LOADLIB.EXE,该功能入口参数如下:

AH = 80H 功能号

CL = 1 读 24 点阵宋体字模  
2 读 24 点阵仿宋体字模  
3 读 24 点阵黑体字模  
4 读 24 点阵楷体字模

CH = 0 读正常字型

DX = 汉字内码, DH=0 时, DL=ASCII 字符码(00H-A0H)

ES:BX = 字模存放地址,应有 72 字节空间

除了用 16H 中断的方法外,还可以用直接读字库的方法得到字模,这样就不限于使用中国龙的字库,可以使用其它汉字系统的各种点阵各种字体的字库,比如金山汉



字,就可以使用四十点阵的字库。下面给出中国龙 24 点阵汉字库 ACLIB24.DAT 的结构:

偏移范围	字库内容
0—5795	161 个 ASCII 字符, 24×24 点阵, 00H—A0H
5796—107315	1—15 区图形符号, 24×24 点阵
107316—594611	16—87 区, 24×24 点阵宋体字
594612—1081907	16—87 区, 24×24 点阵仿宋体字
1081908—1569203	16—87 区, 24×24 点阵黑体字
1569204—2056499	16—87 区, 24×24 点阵楷体字

把按上述方法生成的程序设置成一个中断处理程序,并取代原中断 17H 的处理程序,即可成为一个九针打印机驱动程序,只要通过中断 17H 向打印机输出字符,都可以输出 24 点阵的汉字和西文字符来。如果要在文本某处改变字体、行距及横纵向放大倍数,则可以直接在该处加入相应的命令,在本文的程序中命令设置如下:

@c m 设置字体, m=1,2,3,4 分别对应于宋体、仿宋体、黑体、楷体

@x m 设置横向放大倍数, m=1,2

@y m 设置纵向放大倍数, m=1,2,3,4

@n m 设置换行行距, m=1,2,3,4 当然也可以另写一程序来改变这些设置。

以上介绍了驱动程序的主要思路,用户完全可以根据这个框架改进这个程序,使之具有更完善的功能,具体的源程序附后。

```
#include "stdio.h"
#include "stdlib.h"
#include "dos.h"
#include "bios.h"
void printchar();
void step();
void linefeed();
void setgraphics(int number);
void biosp(int cmd,int byte,int port);
void printch(unsigned DX,unsigned AX);
void interrupt NewPrinter(unsigned BP,unsigned DI,unsigned SI,
    unsigned DS,unsigned ES,unsigned
    DX,unsigned CX,unsigned BX,unsigned AX);
void copyrights();
char fbuffer[24][24],pbuffer[12][960],printbuffer[82];
int printlen=0,xtype=1,ytype=1,feedlen=12,chartype=1,
    maxlen=80,endsign=0;
void interrupt (*OldPrinter)();
main()
{
    copyrights();
    if(biosprint(2,0x78,0)==0xF8){ /* 检查程序是否已驻留 */
        puts("Driver already exist !\n");
        exit(1);
    }
    OldPrinter=getvect(0x17);
    setvect(0x17,NewPrinter); /* 截取 17H 中断 */
    keep(0,4000); /* 驻留退出 */
}
void printchar() /* 打印在行缓冲区 printbuffer 里的汉字和字符 */
{
    int i,j,k,n,x,y,byte,p,len;
```

```
unsigned es;
union REGS regs;
char c1,c2,buffer[72],c;
for(p=0;p<printlen;p++){
    c1=printbuffer[p];
    if(c1>0xA0){ /* 检查该字符是否是汉字 */
        c2=printbuffer[p+1];
        if(c2<0xA1){
            c2=0xA1;
            len=12;
        }
        else len=24;
    }
    else {
        c2=c1;
        c1=0;
        len=12;
    }
    regs.h.ch=0;
    regs.h.cl=chartype;
    regs.x.dx=c1*256+c2;
    regs.x.bx=FP-OFF(buffer);
    es=-ES;
    -AX=FP-SEG(buffer);
    -ES=-AX;
    regs.h.ah=0x80;
    int86(0x16,&regs,&regs); /* 取汉字和字符的点阵 */
    -ES=-ES;
    for(n=0;n<len;++n){
        /* 把汉字点阵字节转化为 24x24 点阵逻辑矩阵 */
        for(i=0;i<3;++i){
            c=buffer[n*3+i];
            for(j=0,k=0x80;j<8;++j,k=k/2){
                if((c&k)!=0)fbuffer[n][i*8+j]=1;
                else fbuffer[n][i*8+j]=0;
            }
        }
    }
    for(i=0;i<ytype;++i){ /* 生成送往打印机的图象影像字节 */
        for(j=0;j<3;++j){
            for(x=0;x<len;++x){
                byte=0;
                for(k=0,n=0x80;k<8;++k,n=n/2){
                    y=(k*3+j)/ytype+i*(24/ytype);
                    if(fbuffer[x][y]==1)byte=byte|n;
                }
                for(k=0;k<xtype;++k)pbuffer[i*3+j]
                    [(x*p*12)*xtype+k]=byte;
            }
        }
    }
    if(len==24)++p;
}
for(i=0;i<feedlen;++i)step(); /* 走行间距 */
for(i=0;i<ytype;++i){ /* 打印汉字和字符 */
    for(j=0;j<3;++j){
        setgraphics(p*12*xtype);
        for(k=0;k<p*12*xtype;++k){
            biosp(0,pbuffer[i*3+j][k],0);
        }
        step();
    }
    linefeed();
}
```

```

return;
}
void step() /* 走纸 1/216 英寸 */
{
    biosp(0,27,0);
    biosp(0,'3',0);
    biosp(0,1,0);
    biosp(0,0x0D,0);
    biosp(0,0x0A,0);
    return;
}

void linefeed() /* 走纸 7/72 英寸 */
{
    biosp(0,27,0);
    biosp(0,'A',0);
    biosp(0,7,0);
    biosp(0,0x0D,0);
    biosp(0,0x0A,0);
    return;
}

void setgraphics(int number) /* 设置打印机图形工作方式 */
{
    int low,high;
    high=number/256;
    low=number-high*256;
    biosp(0,27,0);
    biosp(0,'L',0);
    biosp(0,low,0);
    biosp(0,high,0);
    return;
}

void interrupt NewPrinter(unsigned BP,unsigned DI,unsigned SI,
                          unsigned DS,unsigned ES,unsigned DX,
                          unsigned CX,unsigned BX,unsigned AX)
/* 新的 17H 中断处理程序 */
{
    static int settype=0,next=0;
    enable();
    if((AX & 0xFF00)==0){
        if(settype==1 || AX=='@'){
            /* 判断是否是设置字体,行距和横纵向放大倍数 */
            if(settype==1){
                if(next==0)next=AX;
            }
            else {
                /* 设置横纵向放大倍数 */
                if(next=='x' && AX>='1' && AX<='2'){
                    xtype=AX-'1'+1;
                    maxlen=80/xtype;
                }
                /* 设置纵纵向放大倍数 */
                if(next=='y' && AX>='1' && AX<='4'){
                    ytype=AX-'1'+1;
                }
                /* 设置字体 */
                if(next=='c' && AX>='1' && AX<='4'){
                    chartype=AX-'1'+1;
                }
                /* 设置行距 */
                if(next=='n' && AX>='1' && AX<='8'){
                    feedlen=(AX-'1'+1)*4;
                }
                settype=0;
                next=0;
            }
        }
        else settype=1;
    }
}

```

```

}
else {
    printch(DX,AX);
}
AX=AX | 0x1000;
}
else {
    if(AX==0x0278)AX=0xF878; /* 返回已驻留标志 */
    else { /* 执行原 17H 中断处理程序 */
        --AX=AX;
        --DX=DX;
        (*OldPrinter)();
        AX=--AX;
        DX=--DX;
    }
}
return;
}

void printch(unsigned DX,unsigned AX) /* 行缓冲处理程序 */
{
    unsigned n;
    union REGS regs1;
    n=AX & 0x00FF;
    if(n==0x0A){ /* 遇换行字符 0x0A 则打印缓冲区内容 */
        if(endsign==1)endsign=0;
        else {
            printchar();
            printlen=0;
        }
    }
    else if(n!=0x0D){
        if(printlen==maxlen-1){
            /* 如缓冲区已满则打印缓冲区内容,否则送入缓冲区 */
            printbuffer[maxlen-1]=n;
            ++printlen;
            printchar();
            printlen=0;
            endsign=1;
        }
        else {
            printbuffer[printlen]=n;
            ++printlen;
        }
    }
    return;
}

void biosp(int cmd,int byte,int port)
/* 执行原 17H 中断处理程序 */
{
    unsigned AX,DX;
    AX=(cmd << 8)+byte;
    DX=port;
    --DX=DX;
    --AX=AX;
    (*OldPrinter)();
    return;
}

void copyrights()
{
    puts("RoadSigns (R) ACIOS 9 pin Printer Driver Version 1.00");
    puts("Copyright (C) ZhaoQing Chemical Fiber Factory, Y. J. Xiao 1993.10. \n");
    return;
}

```

# 直接读写 EGA/VGA 卡 VRAM 区显示汉字

广东省农业气象中心 张金标 刘锦奎

近来,不少文章介绍了在西文环境下显示汉字的方法,大多是从汉字库中或从显示器上获取汉字点阵字模,然后写点来显示汉字。这些方法一般采用逐点显示,每一点都要调用 10H 中断或者访问 VRAM,因而速度比较慢。笔者经过试验发现一种直接向 EGA/VGA 卡的 VRAM 区送入 16×16 点阵字模来显示汉字的方法,介绍如下供同行参考。

我们知道,EGA/VGA 卡的 VRAM 被分成 4 个以 A000 为段地址的独立位平面,用彩色平面方法(Color Plain Method)来表示像素时(模式 D、E、F 和 10H),每个像素用 4 个二进制位来表示,分别在每个位平面占一位,因此,可同时显示 16 种颜色。屏幕上像素的位置(x,y)与 VRAM 地址的关系可用下面的公式计算:

字节地址 = 显存起始地址 + y × 80 + x / 8

二进制位位置 = 7 - (x mod 8)

其中显存起始地址为 A000:0000。

写位平面操作时,由位平面屏蔽寄存器控制哪些平面可以写入,哪些平面不可以写入。控制时,程序先向操作定序地址寄存器(输出地址是 3C4H)送出索引号 2,然后再通过操作定序数据寄存器(3C5H)送出一个字节,该字节的低四位从低到高分别对应 0、1、2 和 3 号位平面,当该位为 1 时表示该平面可写入,为 0 时表示不可写入,而该字节的值对应当前颜色。

显示汉字首先要获取汉字字模,可从汉字库中获得,汉字库的结构不少文章已有介绍,此处不再叙述。16×16 点阵汉字字模有 32B,显示时其排列顺序如下:

第 1 字节 第 2 字节

第 3 字节 第 4 字节

⋮

⋮

第 31 字节 第 32 字节

根据上面分析,由(x,y)计算出汉字点阵信息各字节在 VRAM 中的地址,由于每行两字节按位是连续存放的,因此只要向该地址连续写入此两字节值即能显示出此两字节对应的字形。对其各行也如此,直到全部点阵信息显示完。(见所附程序)

采用本文所述方法显示汉字,免去每一点都要调用中断 10H,显示速度将有所提高,同时也可控制汉字的前景和背景颜色。如本文所附程序显示的汉字前景是黄色,背景是蓝色。

下面程序用 Microsoft C V6.00 编写,在 AST Premium 386/25 VGA 上通过。

附程序清单:

```
#include<stdio.h>
#include<stdlib.h>
```

```
#include<graph.h>
void hzputs(unsigned char *hzs,int x,int y,char co,char bk);
void main()
{
    unsigned char far *pvideo=(char far *)0xa0000000;
    struct videoconfig vc;
    unsigned char *hzs="直接写显示缓冲区显示汉字示例";
    —getvideoconfig(&vc);
    if(!—setvideomode(—VRES16COLOR))
        exit(0);
    hzputs(hzs,50,100,14,1);
    —setvideomode(vc.mode);
}
/* 函数 hzputs 在屏幕指定位置上显示汉字串
   hzs 指向汉字串的指针 x,y 汉字串的左上角坐标
   co 前景色色码 bk 背景色色码 */
void hzputs(unsigned char *hzs,int x,int y,char co,char bk)
{
    int i,j,k;
    unsigned long qm,wm;
    unsigned char cc[32];
    —segment pvseg=0xa000; /* 定义段常量 pvseg 为显存段地址 */
    unsigned char —based(pvseg) *pvoff=0;
    /* pvoff 为基于 pvseg 的基指针 */
    FILE *fp;
    if((fp=fopen("hzk16","rb"))==NULL)
    {
        puts("ERROR: Cannot open HZK16");
        exit(0);
    }
    for(k=0;k<strlen(hzs);k+=2)
    {
        qm=(long)(hzs[k]-161);
        wm=(long)(hzs[k+1]-161);
        fseek(fp,(qm*94+wm)*32,SEEK—SET);
        /* 获取汉字点阵信息 */
        fread(cc,32,1,fp);
        pvoff=y*80+x/8;
        for(j=0;j<16;j++)
        {
            outp(0x3c4,2); /* 控制读写的位平面 */
            outp(0x3c5,co);
            *pvoff=cc[j*2]; /* 往 VRAM 送字模信息 */
            pvoff++;
            *pvoff=cc[j*2+1];
            /* revsert */ /* 将字模求反以显示背景 */
            outp(0x3c4,2);
            outp(0x3c5,bk);
            pvoff--;
            *pvoff=~cc[j*2];
            pvoff++;
            *pvoff=~cc[j*2+1];
            pvoff+=79;
        }
        x+=16; /* 下一个汉字 */
    }
    fclose(fp);
    getch();
}
```

# 实用文件分割存贮程序

辽宁省铁岭市委办公室 金永涛

在目前的微机系统中,其软磁盘容量最大的是 3.5 寸 1.44M 高密盘,而目前的很多应用软件(如汉字系统中的高点阵字库)其长度远远超出软磁盘的容量,这就给这些大文件的备份和拷贝等工作带来了诸多不便。DOS 系统中能够完成这一工作的只有 BACKUP 备份命令和相应的 RESTORE 恢复命令,用此命令虽然可行,但存在一定的局限性,比如对 DOS 版本要求严格,不同的版本之间不能正确备份和恢复;再有就是备份出的文件不能直接使用等。所以编制一个理想的文件分割程序,对于完善 DOS 的功能是非常必要的。文后所附程序,是一个功能齐全方便实用的文件分割程序,可做为 DOS 外部命令来使用。

此程序涉及的有关知识

## 1. 检测磁盘自由空间方法:

此项工作可由 DOS 功能调用 36H 号子功能完成,此调用具体参数如下:

入口: AH=36H, DL=驱动器号(0: 缺省, 1:A, 2:B...)

出口: AX=每簇扇区数 BX=有效簇数

CX=每扇区字节数 DX=磁盘总簇数

当 AX=FFFFH 时,说明驱动器无效。

其中磁盘自由空间可由下列表达式计算出:

自由空间=AX\*BX\*CX

## 2. 十进制 ASCII 码转换为数值方法

代码转换是汇编程序中经常用到,因为从键盘上输入的数据均以 ASCII 码的形式存在,在程序运算过程中必须把相应的 ASCII 码转换为 16 进制数值形式。十进制的 ASCII 码转换为数值其方法非常简单,只需把每位的高四位屏蔽掉,并乘以相应的权数,相加在一起即可。具体方法请见文后程序。

## 程序的使用方法

本程序大部分采用了用户按键选择方法,人机介面很好。运行本程序首先提示用户输入原文件名,此处采用了先进的文件句柄管理方法,故支持路径操作。然后提示用户输入分割后的文件长度,此处共有四种选择:

1. 360K 2. 1.2M 3. 1.44M 4. 任意长度

前三种选择是为了方便软盘存贮,直接按标准的磁盘容量分割文件;第四种选择是为了满足一些特殊的需要,选择此项后,用户可输入用十进制表示的三位数字,不足三位前面有 0 填充。然后提示选择驱动器号:A B C 等,上述作法增加了程序的灵活性,比如可以在 1.2M 软驱上对 360K 磁盘操作等,并且可以在硬盘上完成分割工作,然后再拷入相应的软盘中,这样做对于大批量的备份工作来说,可以大幅度提高速度。上述参数输入完,经确认后,计算机开始进行分割工作。在对每张软盘存贮前,

首先检测其磁盘自由空间是否满足要求,并做出相应处理。在更换磁盘时,将自动显示当前的磁盘次序号。分割后的文件名依次为: DISK. A、DISK. B……DISK. Z, 共可以分为 26 个文件,可满足一般的使用要求,通过修改程序中的有关语句,可以增加分割后的文件数量。

## 文件恢复方法

恢复的方法非常简单,只需用 DOS 的 COPY 命令按顺序把它们合并在一起即可,但恢复时不要忘了携带/B 参数。恢复以后的文件与原文件的内容、长度将会完全一致。对于有经验的使用者,对恢复过程可以编制一段批处理程序,使操作更加方便,具体方法此处不再重复。

源程序按 COM 文件格式编制,经汇编、链接并转换为 COM 文件即可使用。愿它为你的工作带来方便。

## 源程序附后

```

;FGFILE.ASM(FGFILE.COM)
;1994.01.28
code segment
    assume cs:code,ds:code
    org 100h
start: jmp be
size1 dw 0 ;文件长度
    jsz dw 0
msg1 db 0dh,0ah,' 文件分割存贮程序'
    db 0dh,0ah,' 1994.01.18 $ ',0dh,0ah
msg2 db 0dh,0ah,'请输入原文件名: $ '
msg3 db 0dh,0ah,'请选择磁盘容量:'
    db '1(360K), 2(1.2M), 3(1.44M), 4(任意) $ '
msg4 db 0dh,0ah,'请插入第['
    nu db 'A'] 张盘,然后按任一键! $ '
msg5 db 0dh,0ah,'请选择驱动器 a(A:), b(B:), c(C:): $ '
msg6 db 0dh,0ah,'上述参数是否正确(Y/N) $ '
msg7 db 0dh,0ah,'原文件不存在或打不开! $ '
msg9 db 0dh,0ah,'正在工作请稍候..... $ '
msg1 db 0dh,0ah,'目标盘空间不够! $ '
msg1 db 0dh,0ah,'请输入文件长度(4位十进制数,以K为单位): $ '
buff1 db 30
    db 0
oldname db 30 dup(0)
buff db 1024 DUP(0)
buff2 db 5,0,6 dup(0)
newname db '0';存盘符
    db ',disk.'
oname db '0',0
oldhad dw 0
newhad dw 0
dnum db 0 ;存驱动器号
dos macro x
    mov ah,x
    int 21h
endm
be: mov dx,offset msg1
    dos 9

```

```

mov dx,offset msg2
dos 9 ;输入文件名
mov dx,offset buff1
dos 0ah
mov si,offset oldname
lop1: lods b
      cmp al,0dh
      jnz lop1
      dec si
      mov al,0
      mov [si],al
lop2: mov dx,offset msg3
      dos 9 ;输入磁盘类型
      dos 1
      cmp al,'1'
      jz jx1
      cmp al,'2'
      jz jx2
      cmp al,'3'
      jz jx3
      cmp al,'4'
      jz jx7
      jmp lop2
jx1:  mov size1,354
      jmp jx4
jx2:  mov size1,1185
      jmp jx4
jx3:  mov size1,1423
      jmp jx4
jx7:  ;十进制字符转换为数值
      mov dx,offset msg11
      dos 9
      mov dx,offset buff2
      dos 0ah
      mov al,buff2+2
      and ax,0fh
      mov bx,1000
      mul bx
      mov size1,ax
      mov al,buff2+3
      and ax,0fh
      mov bx,100
      mul bx
      add size1,ax
      mov al,buff2+4
      and ax,0fh
      mov bx,10
      mul bx
      add size1,ax
      mov al,buff2+5
      and ax,0fh
      add size1,ax
jx4:  mov dx,offset msg5
      dos 9 ;输入驱动器号
      dos 1
      mov newname,al
      and al,0fh
      mov dnum,al
wai:  mov dx,offset msg6
      dos 9 ;确认参数
      dos 1
      or al,20h
      cmp al,'y'
      jz jx
      cmp al,'n'
      jnz wai

      jmp be
jx:   mov al,60h
      mov oname,al
      mov al,0 ;打开原文件
      mov dx,offset oldname
      dos 3dh
      jnc jx5
      mov dx,offset msg7
      dos 9
      jmp exit
jx5:  mov oldhad,ax
re:   mov dx,offset msg4
      dos 9 ;提示插盘
      mov ah,0
      int 16h
      mov dl,dnum ;检测自由空间
      dos 36h
      cmp ax,0ffffh
      jz re
      xor dx,dx
      mul cx
      ul bx
      mov bx,1024
      div bx
      cmp ax,size1 ;比较空间
      jnb jx6
      mov dl,07
      dos 2
      mov dx,offset msg10
      dos 9
      jmp re ;空间小换盘
jx6:  inc oname ;文件后缀变换
      mov dx,offset newname
      mov cx,0 ;建立新文件
      dos 3ch
      mov newhad,ax
      mov dx,offset msg9
      dos 9
wr:   mov bx,oldhad
      mov cx,1024
      mov dx,offset buff
      dos 3fh ;读 1024 字节
      cmp ax,0 ;读完否
      jz endd
      mov cx,ax
      mov bx,newhad
      dos 40h ;写读出的字节
      inc jsz
      mov ax,jsz
      cmp ax,size1
      jb wr
      mov bx,newhad
      dos 3eh ;关闭一个文件
      mov jsz,0
      inc nu
      jmp re ;处理下一个磁盘
endd: mov bx,oldhad
      dos 3eh
      mov bx,newhad
      dos 3eh
exit: mov ax,4c00h
      int 21h
code  ends
      end start

```

# TVGA1024×768 模式下屏幕换段的实现

西安第四军医大学 张晓东 张宪军

**【摘要】**本文通过对视频缓冲区与屏幕坐标映射关系的分析,基于特殊寄存器的直接编程,实现了 TVGA1024×768 模式下的屏幕换段。给出了用 C 编写的换段函数。

在使用带有 TVGA 卡的微机时,用户常用到它的 5F 模式。当选择该模式的 64K 页模式进行图形显示时,屏幕下三分之一区域常出现混乱,不显示,上三分之一区域则有重叠等显示出错现象。出现这些现象的根本原因就是视频缓冲区换段的处理不当,也即屏幕换段出错。本文着重讨论了如何进行正确的屏幕换段操作。

## 一、内存映射

TVGA 5F 模式即 1024×768 模式对应的视频缓冲区地址为 A0000H~AFFFFH,当选择 64K 页模式时,这一段内存区仅有 64K 有效,因此只对应屏幕上 512 行,即屏幕的上三分之二部分。由于 768 行共需内存映射区 96K,这多余的 32K 怎么分配?这就是屏幕换段问题。

我们知道,屏幕坐标与 CPU 的内存映射关系是通过一个 20 位地址来确定的。地址字格式如下:

19	18	17	16	15	...	0
B	P	S	OFFSET			

B:位 19—18,选择 0—3 号存储器块。(256K/块)。

P:位 17,选择存储器块的 0 页或 1 页。(128K/页)。

S:位 16,选择页中某段(64K/段)。

OFFSET:位 15—0,给分段中的偏移量。

当一屏的像素个数大于 64K 时,即段内地址偏移量不足时,就需做内存换段处理。TVGA 视频缓冲区的换段主要通过特殊寄存器 3C4,0EH 的状态设置来实现。

## 二、3C4,0EH 寄存器及状态设置

3C4,0EH 寄存器的第 3、2、1、0 位相应于地址字中的 19、18、17、16 位,意义如下:

7	6	5	4	3	2	1	0
×	×	×	×				
位 3,	位 2		存储器块				
0	0		0				
0	1		1				
1	0		2				
1	1		3				

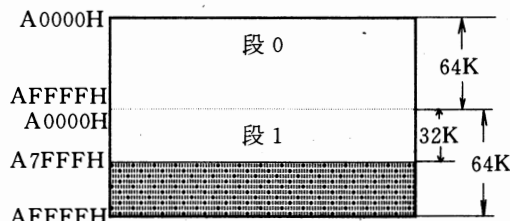
位 1	存储器块中的页
0	读 0 页或写 1 页
1	读 1 页或写 0 页

位 0	存储器页中的段
0	0 段
1	1 段

换段时,只需改变 3C4,0EH 的状态,即向它的读写端口输出 03H 和 02H 两种状态,就可实现换段。

## 三、编写的换段函数

基于上述的原理和方法编写了换段函数。函数名为 Module—Change—Segment(),Y 表示屏幕的纵向坐标,换段后,TVGA 1024×768 模式 64K 页模式下的页面一般映射图如下:



图中,阴影标注的 32K 为保留内存,其它 96K 为显存区。

以上讨论的原理和方法及换段函数,经作者多次使用,正确无误。

程序清单附后:

```
#include <dos.h>
void module—change—segment(unsigned int y)
{
    if((y>512)&&(y<768))
    {
        outportb(0x3c4,0x0e);
        outportb(0x3c5,0x03);
    }
    else if((y<512)|| (y>768))
    {
        outportb(0x3c4,0x0e);
        outportb(0x3c5,0x02);
    }
}
```

## 参考文献

1. 陶伟等“EGA、VGA、TVGA 高级微机图形编程指南与实例”,北京希望电脑公司,1991。
2. 张福炎等“微型计算机 IBM PC 的原理与应用(续二)”,南京大学出版社,1990。

# 背景音乐

中山大学计算机系 郑则仲

一边工作,一边欣赏音乐,这个梦想只要编个不到 100 行的 C 程序即可实现!用 Borland 公司的 C 语言编译器所提供的 keep() 函数,我们很容易就可以写出一个常驻内存的背景音乐程序。它的原理十分简单,无非是通过修改 INT 1CH 用户时钟中断,而在中断服务程序中进行“奏乐”而已。

在这个程序中,采用如下约定来定义“机器简谱”:

1~7 低音 1 到 7

11~17 中音 1 到 7

21~27 高音 1 到 7

1~32 1 到 32 分音符

“机器简谱”写在一文本文件中,在简谱文件中按一个音符一个节拍的顺序排列,音符和节拍间用空白符号(空格,回车,Tab 等)隔开。下面是 C 源程序 bgmusic.c 和一首名为“Yankee Doodle”的歌曲:

```
#include <stdio.h>
#include <dos.h>
#include <stdlib.h>
/* constants defined below */
#define SONGMAX 100U
/* stack and heap length set smaller to shrink the memory */
extern unsigned —stklen = 512;
extern unsigned —heaplen = 2048;
/* you can test to set more smaller */
/* The clock tick interrupt */
#define INTR 0x1C
/* Global variables */
static int tonebase[8]={0,131,146,165,173,196,220,245};
static int i,t,d,len;
static int music[SONGMAX][2];
void interrupt (*oldhandler)();
void interrupt handler()
{
    if (i>=len) i=0;

    if (d==0||i==0) {
        d=music[i][1];
        t=music[i][0];
        if (t) sound(t);
        else nosound();
        i++;
    } else d--;
    oldhandler();
}
```

```
}
int main(int argc,char **argv)
{
    FILE *fp;
    if (argc!=2) { /* check arguments */
        fprintf(stderr,"Usage: bgmusic musicfile\n");
        exit(1);
    }
    if ((fp=fopen(argv[1],"rt"))==NULL) {
        fprintf(stderr,"Can't open file %s\n",argv[1]);
        exit(2);
    }
    i=0;
    while (!feof(fp)) {
        fscanf(fp,"%d %d",&t,&d); /* read tone, delay */
        if (t>20 && t<28) /* tone c2 to b2 */
            music[i][0]=tonebase[t-20]*4;
        else if (t>10 && t<18) /* tone c1 to b1 */
            music[i][0]=tonebase[t-10]*2;
        else if (t<8) /* tone c to b */
            music[i][0]=tonebase[t];
        else music[i][0]=0; /* invalid or zero */
        if (d<=32 && d>0)
            music[i][1]=32/d; /* delay */
        else music[i][1]=0; /* invalid set to 0 */
        if (i++ > SONGMAX) { /* safely checking */
            fprintf(stderr,"Sorry, only %d number of tone supported.\n");
            exit(3);
        }
    }
    fclose(fp);
    len=i; i=0;
    /* get vector of old INT 1Ch */
    oldhandler = getvect(INTR);
    /* install the new interrupt handler */
    setvect(INTR, handler);
    /* keep in memory */
    keep(0, (-SS + (-SP/16) - —psp));
}

一首名为“Yankee Doodle”的歌曲的机器简谱文件“doodle.txt”:
05 08 11 08 11 08 12 08 13 08 11 08 13 08 12 08 05 08 11 08 11 08 12 08
13 08
11 04 07 08 05 08 11 08 11 08 12 08 13 08 14 08 13 08 12 08 11 08
07 08 05 08 06 08 07 08 11 04 11 08 00 08 06 08 07 08 06 08 05 08
06 08 07 08 11 04 05 06 06 16 05 08 04 08 03 08 04 08 05 08 00 08
06 08 07 08 06 08 05 08 06 08 07 08 11 08 06 08 05 08 11 08 07 08 12 08
11 04 11 04
00 01
```

# 空心汉字的显示

湖北宜昌市财贸学校 周苏峡

本文通过读取汉字字模的方法,显示空心汉字,文末给出实例,显示出  $45 \times 48$  点阵空心汉字。该程序使用 Turbo C 2.0 编译,在 PC/XT、286、386 及兼容机上运行通过。

## 一、 $45 \times 48$ 点阵汉字的结构

CLIB48 为  $45 \times 48$  点阵宋体字库,每个汉字字模结构如图 1 所示:

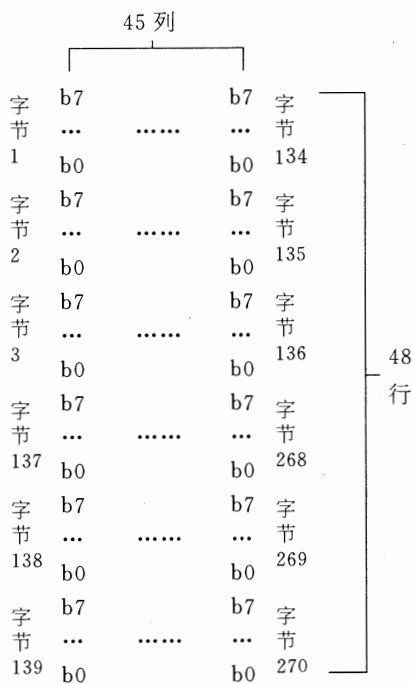


图 1

## 二、空心汉字显示程序

本程序首先将一个汉字字模信息读到二维数组  $a[48]$  中去,然后处理  $a$  数组,如果某位置上的元素(代表一个点)值与它左边一个的值不相等,或者与它上面一个元素的值不相等时,就写点,否则就不写点,从而完成汉字字形边界上的点显示,而非边界上的不显示,显示出空心汉字的效果。

## 三、源程序清单

该程序用 ROM BIOS 方法写点,在屏幕的第 100 行、第一列用黄色显示“宜昌市财贸学校”。

```
#include <stdio.h>
#include <dos.h>
```

```
#include <graphics.h>

main()
{
    int driver=DETECT, mode;
    initgraph(&driver, &mode, "");
    outtextxy(1, 1, "");
    hz45X48("宜昌市财贸学校", "d:\\clib48", 1, 100, 14);
    getch(); restorecrtmode();
}

/* 空心汉字显示函数 */
hz45X48(char *hz, char *hzk48, int x, int y, int color)
{
    int i, j, l, k3, old=y, a[48][45];
    char sec1, sec2, buf[271], *buffer, k1;
    register int k2, k4, k;
    unsigned long index;
    union REGS r;
    FILE *fp;
    fp=fopen(hzk48, "rb");
    while (*hz) {
        old=y; buffer=buf;
        sec1=*hz-160; sec2=*hz+1-160;
        index=(sec1-16)*94+sec2-1; index*=270;
        fseek(fp, index, 0);
        fread(buf, 1, 270, fp);
        k3=x; k4=y;
        for (l=0; l<2; l++){
            for (i=0; i<45; i++){
                for (j=0; j<3; j++){
                    k1=*buffer++; k=128;
                    for (k2=0; k2<8; k2++){
                        if (k1&k) a[k2+8*j+24*l][i]=1;
                        else a[k2+8*j+24*l][i]=0;
                        k/=2;
                    }
                }
            }
        }
        for (i=1; i<48; i++){
            for (j=1; j<45; j++){
                if (a[i][j]!=a[i][j-1]||a[i][j]!=a[i-1][j]) {
                    r.h.ah=12;
                    r.h.al=color;
                    r.x.cx=j+x;
                    r.x.dx=i+y;
                    int86(0x10, &r, &r);
                }
            }
        }
        y=old-y;
        hz|=2; x|=50;
        if (x+45>639){
            x=1; y+=50;
        }
    }
    fclose(fp);
}
```



**APPC (Advanced Program to Program Communication)** 先进的程序到程序通信

**append** 串接, 拼接; 悬挂, 附加, 添加, 追加;

**append mode** 附加方式

**APPI (Advanced Program to Program Interface)** 先进程序至程序界面

**Apple** 苹果; 苹果病毒

**Apple media control architecture (AMCA)** 苹果媒体控制体系结构

**Apple talk filling protocol (AFP)** 苹果令牌填充协议

**Appletalk link access protocol (ATLAP)** Appletalk 链路存取协议

**Appletalk transaction protocol (ATP)** Appletalk 传输协议

**application binary interface (ABI)** 应用二进制界面

**application-centric** 以应用为中心的

**application control and management service (ACMS)** 应用程序控制和管理服务

**application control and management system (ACMS)** 应用程序控制和管理系统

**application control architecture (ACA)** 应用控制体系结构

**application control architecture service (ACA service)** 应用控制体系结构服务

**application control block (ACB)** 应用控制块

**application co-operative service** 应用协作服务

**application enabler** 应用使能程序

**application environment profile (AEP)** 应用环境功能轮廓

**application environment service** 应用环境服务

**application environment specification (AES)** 应用环境规范

**application expert (AE)** 应用专家

**application generator** 应用生成器

**application icons** 应用程序图符

**application information engineering** 应用信息工程

**application integrated architecture (AIA)** 应用集成 [体系] 结构

**application integrator** 应用集成器

**application interface (AI)** 应用界面

**application kit** 应用工具箱

**application management application entity (AMAE)** 应用管理应用实体

**application management application process (AMAP)** 应

# 新 辞 典

用管理应用进程

**application management service** 应用管理服务

**application migration center** 应用移植中心

**application model (AM)** 应用模型

**application operating environment (AOE)** 应用操作环境

**application portability profile (APP)** 应用可移植性功能轮廓

**application process (AP)** 应用进程

**application processor (AP)** 应用处理机

**application processor executive (APX)**

应用处理器执行

**application productivity tool (APT)** 应用软件生产率工具

**application programmer's interface (API)** 应用软件程序员的界面

**application programming interface** 应用程序设计界面

**application prototyping** 应用原型化法

**application protocol control information (APCI)** 应用协议控制信息

**application service element (ASE)** 应用服务元素

**applications integration architecture (AIA)** 应用集成体系结构

**application-specific integrated circuit (ASIC)** 专用集成电路

**application specific memory integrated circuit (ASMIC)** 专用存储器集成电路

**application-specific micro computer (ASMIC)** 专用微计算机

**application specific standard product (ASSP)** 专用标准产品

**application specific standard production (ASSP)** 专用标准产品

**application system (AS)** 应用系统

**application visual system (AVS)** 应用视算系统

**application windows** 应用程序窗口

**applicative language** 功能语言 **appointment scheduler** 计划预定表, 约会安排

**approach angle** 工作角; 前进角

**approach length** 工作长度; 近似长度

**approximately-equal** 近似相等

**April First** 愚人节病毒

**APSE (Ada Programming Support Environment)** Ada 程序设计支持环境

**APSE (Ada Program Support Environment)** Ada 程序支

# MACS 通用会计管理



## 控制系统

●可在微机单机、微机网络上可靠运行

●完全符合会计准则及新会计制度的要求独特的设计思想

●设计论文获得广东省计算机用户协会 94 年会优秀论文

●MACS 企业经营管理与控制系统的全面支持接口完善

MACS 是根据当今世界最为流行的 MRP II 先进的计算机管理方式设计的企业经营管理计算机系统软件包。它将计划、制造、采购、分销、库存、财务高度集成于一体,为企业提供一个完整的解决方案。其中,按新会计制度最新开发的“MACS 通用会计系统”既可独立运行,又可在 MACS 系统支持下跨系统的多模块的集成化运行,制造、库存、采购、工资、固定资产管理模块的财务数据可自动的传送给会计系统自动生成凭证,方便、准确、一体化处理,是现代管理系统的标志。

MACS 通用会计系统充分体现新会计制度的精华,智能化、自动化程度高,功能齐全、结构严谨、工作可靠、安全保密性能好,不但具有完善的账务处理功能更具有会计作业控制功能,使您的会计业务走向自动化、标准化、规范化、无错化的境界,还具有独特的出纳事务管理与实习系统。

### 成果推广与交流 针对性开发 企业管理计算机应用咨询

广东省计算机用户协会 AS/400 专业委员会 联系人:钟恩光 地址:广州江燕路 288 号 邮码:510280 电话:4419977

持环境

APT (Application Productivity Tool) 应用软件生产率工具

APX (Application Processor eXecutive) 应用处理器执行 aqueous cleaning 水洗技术

AR (Artificial Reality) 人工现实

ARC (Advanced RISC Computing) 先进 RISC 计算环境

ARC (Advanced RISC Computing specification) 先进 RISC 计算规范

architectural cost program (ACP) 建筑成本估价软件包

architectural design 建筑设计

architectural design program (ADP) 建筑设计软件包

architectural, engineering and construction (AEC) 体系结构、工程和结构

architecture body 体系结构体

architecture design and evaluation language (ADELA) 体系结构设计和评价语言

architecture level 体系结构级

architecture neutral distribution format (ANDF) 体系结构中中性分布格式

### 广告索引

- 1、 睿普系统工程公司
- 2、 广州金泽科技企业有限公司
- 3、 广州袖珍计算机应用技术服务中心
- 4、 广州宏辉机电工程公司
- 5、 华粤电子系统公司
- 6、 北大方正集团广州方正公司
- 7、 广州市海谊电子仪器实业公司
- 8、 广西北海市天梭信息系统工程公司
- 9、 广州蓝深计算机网络系统公司
- 10、 香港现代电子出版社
- 11、 电脑软件法律保护咨询部
- 12、 广州市泰来新技术发展有限公司
- 13、 广东佛山高新光电子电脑公司
- 14、 电子工业出版社广州科技公司
- 15、 广州国际电脑电子博览中心中外软件廊
- 16、 中美合资广州安斯电子设备厂
- 17、 清华大学科学馆
- 18、 珠海科达电源工业公司

# 怎样使你的软件得到法律保护?

## 电脑软件法律保护咨询部

### 帮您忙!

- 软件法律保护知识普及

(版权法、专利法、商标法、反不正当竞争法等)

- 软件原创性、相似性的分析
- 兼容软件取得版权的途径
- 软件开发过程的有关法律问题
- 软件侵权诉讼准备研究
- 软件销售与使用的法律责任
- 案例研讨与分析

热线电话服务逢星期五

下午 2:30~6:00 时

广东省计算机用户协会

电脑杂志社

广东省电脑商会

广州国际电脑电子博览中心中外软件廊

联合主办

# 请记住热线电话:(020)7504151