

E&C

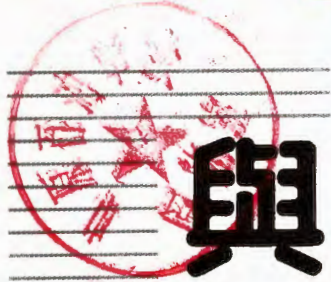
1994

●一九九四年 ●总期第 111 期

6

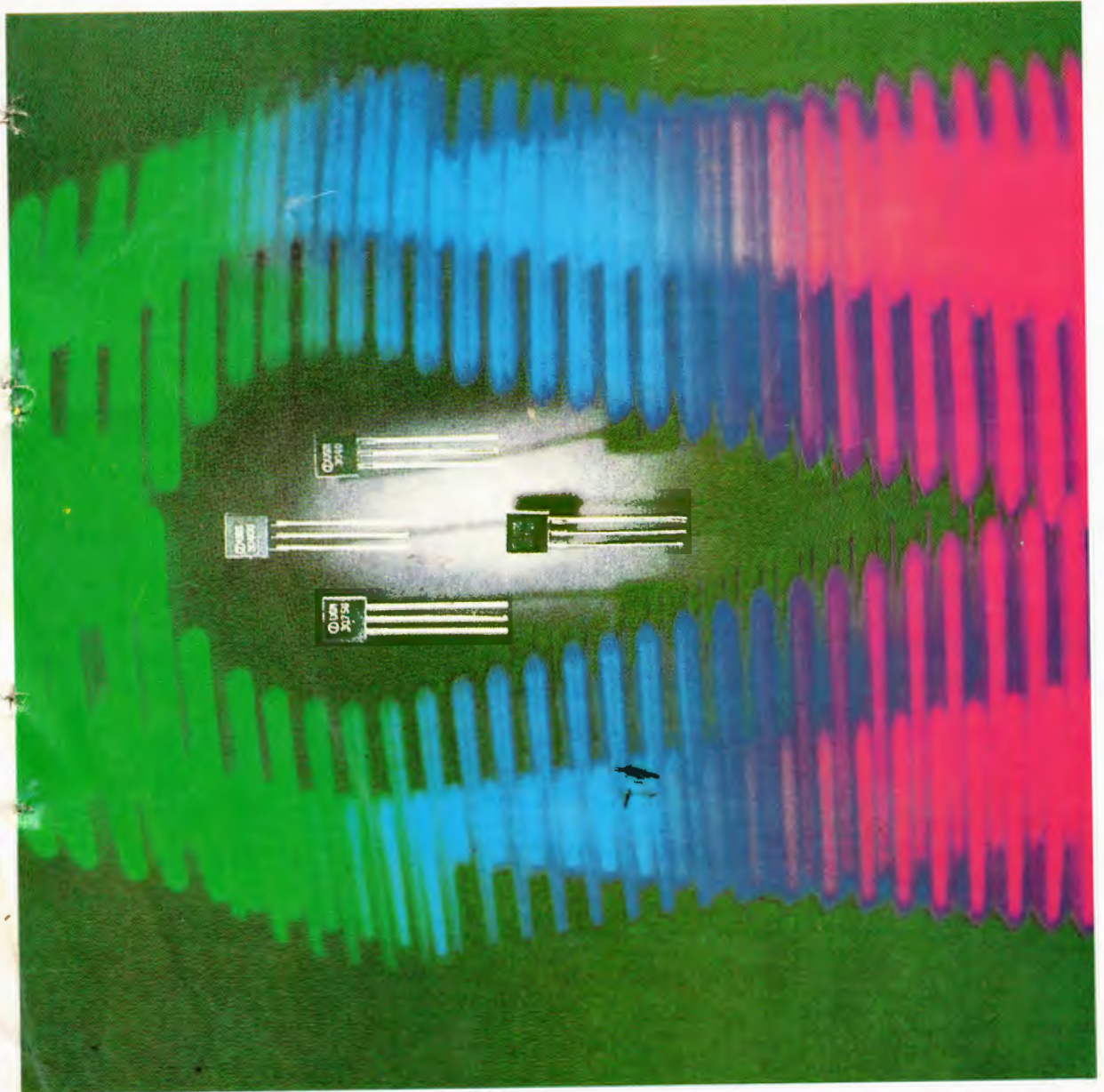
電子

ISSN 1000 1077



與

電腦



• ELECTRONICS AND COMPUTERS •

DBW • SBW系列单相 • 三相 大功率全自动补偿式电力稳压器

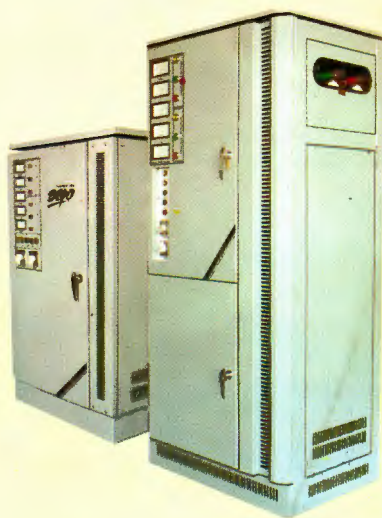
新康电源——与众不同的新设计！

我厂引进西欧技术，根据我国用电国情专业生产DBW • SBW系列单相 • 三相大功率全自动补偿式电力稳压器，是为电压不稳而设计的国内目前最为先进的稳压电源。

该电源性能不同一般，具有报警装置、自复装置、避雷保护、缺相保护、三相分调、双速调节、相序指示等设施。还可根据引进任何国家的设备所需电压，配备隔离变压器。即使是在千载难逢的特殊情况下，“新康”电源同样能为你排忧解难，百改不厌，保您满意。



指标：稳压精度：额定电压 $\pm 1 \sim 5\%$ (可调)
稳压范围：额定电压 II 型 $\pm 20\%$ (304~456V)
 III 型 $\pm 25\%$ (285~475V)
波形畸变：0.1%
损耗：1.5% (平均值)
应变时间： $\leq 0.1s$ 有过压、故障自动保护
规格：三相：10~1200kVA
 单相：3~300kVA



来函必复，来厂洽谈购机专车免费接送



上海新康电器仪表厂

销售科地址：上海市西康路1029号 电话：(021)2587983、2552309 电挂：6351 传真：(021)2587983 邮编：200060

厂址：上海市交通路3250弄 电话：6954135

开户行：工行普办长分处 帐号：210-04764496

● 国内代号：2-888 定价：2.40 元

MCS-51指令系统速查卡(一)

	0	1	2	3	4	5	6. 7	8—F						
0	NOP	AJMP0	LJMP addr16	RR A	INC A	INC dir	INC @Ri	INC Rn						
1	JBC bit,rel	ACALL0	LCALL addr16	RRC A	DEC A	DEC dir	DEC @Ri	DEC Rn						
2	JB bit,rel	AJMP1	XRT	RL A	ADD A, #da	ADD A,dir	ADD A,@Ri	ADD A,Rn						
3	JNB bit,rel	ACALL1	RETI	RLC A	ADDC A, #da	ADDC A,dir	ADDC A,@Ri	ADDC A,Rn						
4	JC rel	AJMP2	ORL dir,A	ORL dir, #da	ORL dir, #da	ORL A,dir	ORL A,@Ri	ORL A,Rn						
5	JNC rel	ACALL2	ANL dir,A	ANL dir, #da	ANL dir, #da	ANL A,dir	ANL A,@Ri	ANL A,Rn						
6	JZ rel	AJMP3	XRL dir,A	XRL dir, #da	XRL dir, #da	XRL A,dir	ANL A,@Ri	XRL A,Rn						
7	JNZ rel	ACALL3	ORL C,bit	JMP @A+DPTR	MOV A, #da	MOV dir, #da	MOV @Ri, #da	MOV dir, #da						
8	SJMP rel	AJMP4	ANL C,bit	MOVC A,@A+PC	DIV AB	MOV dir,dir	MOV dir,@Ri	MOV dir,Rn						
9	MOV DPTR, #da	ACALL4	MOV bit,C	MOVC A,@A+DPTR	SUBB A, #da	SUBB A,dir	SUBB A,@Ri	SUBB A,Rn						
A	ORL C,/bit	AJMP5	MOV C,bit	INC DPTR	MUL AB		MOV @Ri,dir	MOV Rn,dir						
B	ANL C,/bit	ACALL5	CPL bit	CPL C	CJNE A, #da,rel	CJNE A,dir,rel	CJNE @Ri,da,rel	CJNE Rn, #da,rel						
C	PUSH dir	AJMP6	CLR bit	CLR C	SWAP A	XCH A,dir	XCH A,@Ri	XCH A,Rn						
D	POP dir	ACALL6	SETB bit	SETB C	DA A	DJNZ dir,rel	XCHD A,@Ri	DJNZ Rn,rel						
E	MOVX A,@DPTR	AJMP7	MOVX A,@R0	MOVX A,@R1	CLR A	MOV A,dir	MOV A,@Ri	MOV A,Rn						
F	MOVX @DPTR,A	ACALL7	MOVX @R0,A	MOVX @R1,A	CPL A	MOV dir,A	MOV @Ri,A	MOV Rn,A						
可位寻址 SFR														
RAM 中可寻址位														
P0.7	P0.6	P0.5	P0.4	P0.3	P0.2	P0.1	P0.0	80	P0	A E0	位地址	字节	位地址	字节
TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0	88	TCON	B F0				
P1.7	P1.6	P1.5	P1.4	P1.3	P1.2	P1.1	P1.0	90	P1	SP 81				
SM0	SM1	SM2	REN	TB8	RB8	T1	R1	98	SCON	DPL 82	07—00	20	47—40	28
P2.7	P2.6	P2.5	P2.4	P2.3	P2.2	P2.1	P2.0	A0	P2	DPH 83	0F—08	21	4F—48	29
EA			ES	ET1	EX1	ET0	EX0	A8	IEC	TL0 8A	17—10	22	57—50	2A
P3.7	P3.6	P3.5	P3.4	P3.3	P3.2	P3.1	P3.0	B0	P3	TL1 8B	1F—18	23	5F—58	2B
			PS	PT1	PX1	PT0	PX0	B8	IPC	TH0 8C	27—20	24	67—60	2C
CY	AC	F0	RS1	RS0	OV		P	D0	PSW	TH1 8D	2F—28	25	6F—68	2D
SMOD				GF1	GF0	PD	IDL	87	PCON	SBUF 99	37—30	26	77—70	2E
GATE	C/ \bar{T}	M1	M0	GATB	C/ \bar{T}	M1	M0	89	TMOD		3F—38	27	7F—78	2F
不可位寻址 SFR														

MCS-51 指令系统速查卡(二)

传送,交换,栈出入指令	算术,逻辑运算指令	转移指令
MOV A,Rn Em dir E5() @Ri EK #da 74()	ADD/ A,Rn 2m/3m /ADDC dir 25/35() @Ri 2k/3k #da 24/34()	ACALL add11 b1() LCALL add16 12() AJMP add11 a1() LJMP add16 02() JMP @A+DPTR 73 JZ/JNZ rel 60/70() CJNE a,#da,rel B4() CJNE A,dir,rel B5() CJNE @Ri,#da,rel Bk() CJNE Rn,#da,rel Bm() DJNZ Rn,rel Dm() DJNZ dir,rel D5() NOP/RETR/RET 00/32/22 SJMP rel 80()
MOV Rn,A Fm dir Am() #da 7m()	SUBB A,Rn 9m dir 95() @Ri 9k #da 94()	布尔指令集
MOV dir,#da 75() A F5() Rn 8m() dir 85() @Ri 8k()	MUL/DTV AB A4/84	
MOV @Ri,A Fk dir Ak() #da 7k()	INC/ A 04/14 /DEC Rr 0m/1m dir 05/15() @Ri 0k/1k	
MOV DPTR,#DA 90() E0	INC DPTR A3	
MOVX A,@RI Ej @DPTR E0	CLR/CPL A E4/F4	MOV C,bit A2() MOV bit,C 92() CLR C/bit C3/C2() SETB C/bit D3/D2() CPL C/bit B3/B2() ANL C,bit 82() ANL C,bit 80() ORL C,bit 72() ORL C,bit A0() JC/JNC rel 40/50() JB bit,rel 20() JNB bit,rel 30() JBC bit,rel 10()
MOVX @Ri,A Fj	RR/C A 03/13	注释
MOVX ODPTR,A F0	RL/C A 23/33	
MOVX A,@A+DPTR 93 @A+PC 83	ANL/ A: Rn 5m/4m /ORL dir 55/45() @Ri 5k/4k #da 54/44()	
XCH A,Rn Cm @Ri Ck dir C5()	ANL/ dir,A 52/42() /ORL #da 52/43()	
XCHD A,@Ri Dk	XRL A,Rn 6m dir 65() @Ri 6k #da 64()	页 a 0 1 2 3 4 5 6 7 b 0 2 4 6 8 A C E 1 3 5 7 9 B D F
PUSH dir C0()	XRL dir,A 62() #da 63()	Rn R0 R1 R2 R3 R4 R5 R6 R7 8 9 A B C D E F
POP dir D0()	DA A D4	Ri R0 K 6 j 2 R1 7 3
SWAP A C4		

注释:dir:8位内部 RAM 单元及 SFR 地址

#da:8位数据

add:目标地址;add11:11位目标地址,高三位为页面地址,隐含在操作码 a,b 中

rel:地址偏移量(补码),即下条指令地址为基址偏移-128,+127字节

bit:可位寻址的位地址,内部 RAM 的20H-2FH单元及可位寻址的 SFR

电子与电脑

一九九四年
总期第 111 期

目 录

· 综述 ·

486 微处理器产品动向 吴腾奇(2)

· PC 用户 ·

大型数据文件的快速分类 黄正瑞(4)

动态图形库的建立及其应用 李明章(7)

HGC 卡高分辨率图形的屏幕拷贝 傅叔平(9)

中文界面设计与 SPT 图像 张少猷(10)

方便地读写硬盘主引导扇区 赫 建(12)

使用金山 WPS 3.0F 系统配置文件的设置

..... 何奇继(13)

软盘容量扩充技术 邱向群(14)

PC Tools 9.0 功能简介 姜 宏(15)

一种新病毒的分析与防治 高 峰(16)

· 学习机之友 ·

在中华学习机上真正实现子目录功能 秦燕军(18)

任意数的高精度快速乘法程序 徐 永(20)

· DOS 操作系统分析讲座 ·

第三讲 屏幕显示系统(续二) 崔来堂(21)

· 学用单片机 ·

DP-851 单片机系统 胥筱汀(25)

用 8031 单片机组装细胞分类计数器 包起龙(28)

一则教材中常见错误的更正 朱立奇(29)

· 电脑巧开发 ·

语音合成技术在电脑游戏机中的应用

..... 易继错 张京伟 仇之明(32)

一种打印机共享电路 袁汝钧(35)

· 维修经验谈 ·

第二讲 针式打印机的结构及各部分的功能

..... 赵继文(36)

GW500 型彩显特殊故障检修一例 陆昌平(38)

· 电脑游戏机 ·

第六章 背景画面绘制的编程技巧(续)

..... 于 春(39)

· Windows 讲座 ·

Microsoft Windows 3.1 应用基础

——桌面机办公用具 徐鹏力(42)

· 初学者园地 ·

· 工具软件 PC Tools 的应用 王路敬(46)

MS-DOS 5.0 中 DIR 的几项新增功能

..... 贾纳豫(47)

Super-CCDOS 修改一则 邱建斌(48)

· 多媒体世界 ·

支持多媒体应用的网络能力 周力丹(49)

· 电脑通信 ·

用内部电话网组建微机局域网系统(LAN)

..... 汤忠庆(52)

实时系统时钟减慢的分析及解决 李开新(54)

· IC 电路应用 ·

三角函数波形生成 IC-AD639 及用法 李兰友(55)

· 读者联谊 ·

也谈怎样使用 SMARTDRV.SYS 加速 WPS

..... 韦 韬(53)

电子工业部电子工业出版社主办

编辑、出版:《电子与电脑》编辑部

(北京 173 信箱 邮政编码:100036)

印刷:北京三二〇九厂

国内总发行:北京报刊发行局

国内统一刊号:CN11-2199

邮发代号:2-888

国外代号:M924

出版日期:每月 23 日

主编:王惠民 特约编审:苏子栋

责任编辑:施玉新

订购处:全国各地邮电局

国外总发行:中国国际图书贸易总公司

(北京 399 信箱 邮政编码 100044)

广告经营许可证:京海工商广字 0030 号

定价:2.40 元

486 微处理器产品动向

吴腾奇 广州华南师大物理系(510631)

读者们如果有兴趣逛电脑商场,应该发觉市场上好像多了很多不同名称的 486 CPU,有 SX、DX、DX2、SLC……名称一大堆,弄也弄不清。记得在早期,常见的 8086 只有一种,286 也是只有一类,到 386 时便来了 SX 及 DX,而 486 则变、变、变,孙悟空变戏法,吹一根毛变出很多孙悟空来。

一、何来那么多孙悟空?

事实上,市场上所售的 486 已不一定是由 Intel 公司的生产线来的。情况和 386 时代一样,很多厂家眼见 Intel 的 CPU 卖到断行断市,于是马上加入市场,要和 Intel 分饼吃。486 时代比 386 时代更混乱,不单只有 AMD 和 Intel 对决,CYRIX 和 IBM 也誓师出发,加入混战。

市场上的 486 除了可用生产商来区分,也可用其工作频率来分类,计有 25MHz、33MHz、40MHz、50MHz、66MHz 及快来临的 100MHz 等等。此外,也可按操作电压分为 5V 的 CPU 和 3.3V 的 CPU,前者用于日常的桌上电脑,电源是由市电供应,后者主要用于膝上计算机(LAPTOP)等用电池作电源的电脑内,不单要考虑低电压,也要考虑耗电量的问题。现试以生产商为分类条件,向大家简介变变的 486CPU 产品动向。

二、Intel 公司

1989 年,Intel 推出了第一颗 486,名叫 80486 或 i-APX468,现在则称为 486DX。Intel 设计 486 时的构思是把 386、387 算术协处理器及 82385 快存控制器的功能集于一身,再加上内置 8K 快存记忆,而运算则比 386 快一倍。因此,PC 机生产厂只要对其 25MHz 及 33MHz 的 386 母板作小小改动后便可安上 486,不但减低母板设计成本,也把 PC 机的能力提升。可是 486 的制造成本不轻,售价也不便宜,最大的缺点是 486 耗电量甚高,不适合 LAPTOP 等机使用。

Intel 对此问题的解决方法是推出 486SX,工作频率是 16MHz 及 20MHz,较 486DX 为慢,因而耗电量也减低。此外 486SX 是使 486DX 的内置浮点数学运算单元(FPU)失能(disable),这样不单可减低生产线上不合格废品晶片的数量,也间接降低了成本,而售价也可降低。后期的 486SX 更直接把晶片上 FPU 的线路去掉,但不论是早期的 486SX 或后期的 486SX,同样都有一个好处,工作频率低而不用处置快存记忆,更进一步降低 PC 机生产商的制作成本。

除了 486SX、486DX,Intel 的工程师为了进一步提升 CPU 的功能,但又不想加重母板的成本,于是来了一招“时钟倍频”(clock-doubler)技术,其构思是母板

以一个较低的工作频率操作,如 25MHz 或 33MHz,但这个频率的钟信号一进入 CPU 便被倍频,成 50MHz 或 66MHz,而 CPU 内部便以此频率工作,因而达成前述的要求。

这个构思后来便叫过驱动(overdrive),只要把 25MHz 的 486DX 拿起,放入有 Clock-doubler 的 overdrive 晶片,便成为了 50MHz 的 486 了。这个 overdrive 晶片也就是 DX2,计有 50MHz 的 DX2 及 66MHz 的 DX2,分别代替 25MHz 的 486DX 及 33MHz 的 486DX。可是 486SX 又如何呢?当用家把 487SX 安于 486SX 母板上的 overdrive 插座后,这颗 487SX 便会把 486SX disable,而自己来当皇帝,成为母板上可操作的唯一 CPU。

说穿了,487SX 根本就是 486DX,不过芯片引出脚排位不同吧!但是,不久后,Intel 会推出 DX2 版本(即有 clock-doubler)的 486SX overdrive。

Intel 推出了 DX2 后,虽然加强了 CPU 的功能,但是对 LAPTOP 市场来说,却没有掀起大风浪,因为其 CPU 的耗电量仍然不合乎 LAPTOP 的低耗电量及低电压的要求。Intel 的 486SL 于是推出了,跟后还有 486SX、DX 及 DX2 也推出 SL 系列。486SL 是一颗 25MHz 的 3.3V 486 晶片,主力攻占 LAPTOP 市场,其后将到的 SL 系列,或称 SL enhanced(增强)系列,将会有 3.3V 及 5V 的种类,但一样加上电源管理单元(Power Management Unit)。

Intel 最后的动向是生产 25MHz 及 33MHz 的 486SX; 33MHz 及 50MHz 的 486DX; DX2 则有 40MHz、50MHz、66MHz,而 40MHz 的 DX2 只有 3.3V 版,50MHz 的 DX,50MHz 的 DX2 及 66MHz 的 DX2 则只有 5V 版。

三、AMD 公司

AMD 成功打入 386 市场后,当然不会放过 486 市场,其 CPU 的特色在于和 Intel 的 CPU 用相同工作频率却可得到更快的速度,还有其 3.3V 的低电压特性,主要占取 LAPTOP 市场。

AMD 当然也有生产 5V 的 486 晶片,名为 AM486DX,分别有 33MHz 及 40MHz 两种版本,同样有内置 8K 快存记忆,和 Intel 的 486DX 大致一样。而 3.3V 的 486 晶片则叫 AM486DXLV,名称后面的 LV 是 Low Voltage 的缩写,即低电压之意,其实是一颗以 3.3V 电压工作的 33MHz AM486DX,并内置电源管理单元吧!那么 DX2 呢? AMD 当然不会大意,于是又有和 Intel 50MHz 的 486DX2 兼容的 AM486DX2-

50,看来 Intel 要招架的颇辛苦了。

四、CYRIX 公司

CYRIX 除了在数学协处理器市场上和 Intel、AMD 等大打出手外,对 486 的市场也不放过。CYRIX 所用的方法却和 AMD 不同,不是 Intel 486 晶片的直接代用品,而是 386 的升级代用品。

CYRIX 的 CX486SLC/E 便是一颗 386SX 升至 486SX 的晶片,但是内置的快存记忆只有 1K,并且可以和 CYRIX 自己的数学协处理器协调运作。虽然 CX486SLC/E 的脚位和脚数与 386SX 一样,可以直接插入 386SX 的插座,但由于要处理内置的快存记忆,所以主板及 BIOS 均要作些少改动。

CX486SLC/E 有 25MHz 及 33MHz 两款,均可以用 3.3V 电压工作,虽然只有 16 位的外部数据总线,因而比 Intel 486SX 较慢,但是由于晶片较细和耗电低,所以对 LAPTOP、NOTEBOOK(笔记本型)电脑等十分适用。此外 CYRIX 并在晶片上加上了一组硬件的乘法运算器,所以一些浮点运算会比 Intel 的 486SX 更快。

那么有没有 386DX 的升级版呢? CYRIX 的 CX486DL 便是答案。CX486DL 和 CX486SLC/E 大致差不多,只是采用了 32 位数据总线。

除了升级版外,CYRIX 也推出一些代用晶片,如 CX486S 便是可以直接插入 486SX 的插座,并提供了 3.3V 工作电压选择,价钱也比 Intel 的 486SX 更便宜,速度更高,有 33MHz、40MHz 及 50MHz。此外 CYRIX 也推出和此晶片一起用的数学协处理器,希望打垮 Intel 的 486SL 及 AMD 的 AM486DXLV。CX486S 唯一的不足之处在于只有 2K 快取记忆,但 CYRIX 却以 Write-Back(回写)的方法来提高快取记

忆的效率。

好了,有没有 DX2 版? 有,CX486S2 是也。此晶片和 CX486S 大致一样,不过采用倍频方法,有 40MHz 及 50MHz 两款,均是可用 3.3V 工作电压,直接替代 20MHz 及 25MHz 的 486SX。

那么有没有在片数学协处理器的 486 晶片? 据说, CYRIX 已密锣紧鼓,要推出代替 486DX 及 486DX2 的 CRIX 486 晶片,名称叫 M7,内置的快取记忆加至 8K,工作频率有 DX 版的 33MHz、40MHz、50MHz 及 DX 版的 40MHz、50MHz 及 60MHz。

五、IBM 公司

IBM 因为和 Intel 有合约,可以生产并修改 Intel 的 386 及 486,于是 IBM 自己造了 386SLC,一颗有 1K 内置快取记忆的 25MHz 386SX,跟着又重施故技,造了 486 SLC。IBM 的 486SLC 除了可用 3.3V 工作电压外,并内置了比 Intel 多一倍的 16K 快取记忆,但和 CYRIX 的 CX486SLC/E 一样只有 16 位外部数据总线,只有一颗没有了电源管理单元的 486SX。也因为合约问题,IBM 不可销售单独的 486SLC,只可内部使用或安于母板上。

对于 Intel 的 DX2 系列,IBM 当然不会走眼,于是 IBM 推出了 486SLC2,有 50MHz 及 66MHz,并且内置 16K 快取记忆和 3.3V 工作电压,但唯一缺点便是 16 位数据总线。

IBM 除了推出 DX2 系列的 486SLC2,还把倍频技术更推上一层楼,采用三倍频率器 clock-tripler。晶片功能主要和 486SLC2 一样,但都有 75MHz 及 100MHz 两款代替 25MHz 及 33MHz 的 486,而晶片便暂叫蓝闪电 Blue Lighting。

现货可供美国 BOURNS 公司网络电阻

宁波国际工贸实业有限公司,现货可供各种规格美国 BOURNS 公司生产的各种型号,各种阻值的单排直插(SIP),双排直插(DIP),表面贴装(SMD)网络电阻,本公司可竭诚为用户提供周到的服务,批发、零售、邮购,任君选择。

规 格	零 售	批 发
单排直插(SIP)	8 分/每脚	4.5 分/每脚
双排直插(DIP)	16 分/每脚	12 分/每脚
表面贴装(SMD):001.002 系列	6 元/每只	4 元/每只
003 系列	9 元/每只	6 元/每只

联系人:蒋佩勇先生 赵端先生

联系地址:宁波市大沙泥街 61 号 207 房间

邮编:315000

电话:7302490 传真:(0574)7302759

大型数据文件的快速分类

黄正瑞 暨南大学苏州苑 31—303

分类通常被理解为按规定次序重排给定的一组数据,因此分类也称为排序,目的往往是加速对分类集合元素的检索。本文介绍一个实用的分类算法,它适用于大型数据文件的分类。

一、前言

分类是计算机软件工程的一个重要课题,特别在数据处理系统中,往往成为计算机处理的主要内容之一,占用大量的 CPU 时间。因此分类算法的研究历来就受到普遍的关注,以致成果累累,精巧算法美不胜收。

但不无遗憾的是除了归并之外绝大多数算法都只适用于数组分类。美国 Herbert Schildt 在著名的《C 语言大全》一书中,曾将磁盘文件作为一个外部大数组,然后采用快速分类法直接对文件进行分类。但由于单个记录地对文件进行读写和移动,造成频繁地读写磁盘,以致程序虽然简短而时间开销却成倍地增加,效率低到无法令人接受。可见人们对文件尤其对大型数据文件的分类仍感到棘手。

大型数据文件的特点,一是数据量大,记录往往数以千万计;二是分类关键字一般是姓名、名称、主题词之类的字符串;三是由于文件存储在磁盘上,存取速度慢。因此文件分类要比数组分类复杂得多。例如在文献检索系统中,为了建立倒排索引文件,一般需要先从前文中抽出检索词连同文献号等信息,生成检索词文件,然后按检索词对文件分类,在此基础上才能有从有序文件中输出以生成倒排索引和词典文件。显然检索词的数量是相当庞大的。一个中文科技期刊数据库每年新增文献数十万件,光从中抽出的主题词条款就有一两百万之多,对如此庞大的主题词文件的分类无疑是十分耗时的。

有鉴于此,笔者在文献检索系统 SOCI 的设计中采用了一种内外结合的算法,再利用 C 语言大批量读写磁盘文件的特点,较好地解决了大量信息的分类问题。为了叙述的方便,我们称之为快速归并分类法。

二、算法

现在我们结合文献检索系统主题词文件的分类来介绍快速归并分类法。主题词文件的记录有三个字段:

一为 13 个字节的主题词,二为 1 个字节的文献发表年份,三为 4 个字节的文献地址。其记录在 Turbo C 语言中用一个结构来表示。

快速归并分类法由以下步骤实现

1. 分组,即将待分类文件分为 2^n 组,我们将在第三步看到其中 n 就是归并趟数。除了最后一组的记录数可能少若干个之外,其它各组长度相等。分组还应考虑内存数据可用空间的大小,使得每组的数据能为内存结构数组所容纳。例如在 64K 内存数据空间的情况下,如果记录长度不超过 20 个字节,则每组记录可多达 3000 个左右。显然,由于受组数和内存空间的限制,这个值需要动态决定。例如可用如下 C 语言程序段确定 n 和每组的记录数 sum :

```
n=0;
sum=ftell(fp)/sizeof(dfrec);
while(sum>3000){sum=ceil(sum/2);n++;}
while(sum%3!=0)sum++;
```

其中第一个循环确保每组不多于 3000 个记录,并同时保证组数为 2 的 n 次幂。第二个循环保证 sum 能为 3 所整除,这是考虑到归并时是将两个数组的数据归并入第三个数组,数组空间只能是内排序时的三分之一,也即每组的数据要分三次读入数组进行归并。

2. 内排序,即逐一将每组记录读入内存结构数组,并用快速分类法或希尔分类法进行数组排序,将结果以组为单位交替地写入两个文件。这样每组记录就是一个有序的初始归并段。下面是用 Turbo C 语言编写的分组排序函数 `sort`,该函数通过 `fread` 和 `fwrite` 函数成批读写磁盘文件,每读入一组记录,即调用 `quick` 函数进行内排序。`quick` 采用快速分类法,所以这个过程相当快。

```
void quick(struct df dfr[],int left,int right)
{
    int x,y;char p1[13],p2[13];
    char ch;long add;
    x=left;y=right;
    strcpy(p1,dfr[(left+right)/2].key);
    do{
        while(strcmp(dfr[x].key,p1)<0 && x<right)x++;
```



```

while(strcmp(dfr[y].key,p1)>0 && y>left)y--;
if(x<=y){
strcpy(p2,dfr[x].key);
strcpy(dfr[x].key,dfr[y].key);
strcpy(dfr[y].key,p2);
ch=dfr[x].year;
dfr[x].year=dfr[y].year;
dfr[y].year=ch;
add=dfr[x].addr;
dfr[x].addr=dfr[y].addr;
dfr[y].addr=add;
x++,y--;
}
}while(x<=y);
if(left<y) quick(dfr,left,y);
if(x<right) quick(dfr,x,right);
}
void sort()
{
FILE *fp, *fp1, *fp2;
struct df dfrec,dfr[3100];
register i,j;int m,flag;long sum;
if((fp=fopen("subf.dat","rb"))==NULL)return
if((fp1=fopen("subf1","wb"))==NULL)return
if((fp2=fopen("subf2","wb"))==NULL)return
n=0;flag=0;fseek(fp,0L,2);
sum=ftell(fp)/sizeof(dfrec);
fseek(fp,0L,0);
while(sum>3000){sum=sum/2+1;n++;}
t=sum;
while(t%3!=0) t++;
while(! feof(fp)){
if((m=fread(dfr,sizeof(struct df),t,fp))==0) break;
quick(dfr,0,m-1);
if(flag==0){
fwrite(dfr,sizeof(struct df),m,fp1);
flag=1;
}
else{
fwrite(dfr,sizeof(struct df),m,fp2);
flag=0;
}
}
fclose(fp);fclose(fp1);fclose(fp2);
remove("subf.dat");
}

```

3. 归并,即进行 n 趟 2 路归并以完成整个分类过程。归并趟数 n 在分组时已经确定,一般而言,由于初始归并段的记录较多, n 值不会太大。例如,对一百万个主题词记录的文件,如果初始归并段为 2000 个记录,则 $n=9$,也就是说只需进行 9 趟归并就能完成分类。下面给出的 merge 函数是主题词文件的归并程序。由于 Turbo C 的成批读写功能,我们实质上已将文件归并转化为数组归并问题。函数中分别从文件 subf1 和 subf2 读数据进 a 、 b 两个结构数组,然后归并进数组 c 。一旦当 c 满,即将其写入文件 subf3 或 subf4。由于 a 和 b 的体积均定义为初始段的三分之一大小,所

以首趟归并时每次读盘均从文件读入初始段的三分之一记录。这样每个初始段要三次才能读完,第 2 趟归并时每段要读 6 次,第 3 趟归并每段要读 12 次,以此类推。在函数中设变量 $a2$ 和 $b2$ 控制每个归并段需要的读盘次数,而由 $a1$ 和 $b1$ 分别记录两个归并段已经读盘的次数。

整个分类过程由以上三个函数完成,只需加上一个简单的主函数,就是一个完整的分类程序。

```

#include<stdio.h>
#include<stdlib.h>
#include<conio.h>
struct dr{
char key [13];
char year;long addr;
};
static int t,n;
main()
{
sort();
merge();
}
void merge()
{
FILE *fp1,*fp2,*fp3,*fp4;
struct df dfrec,a[1005],b[1005],c[1010];
register i,j;
int t3,t4,k,m,r,s,flag,a1,a2,b1,b2;
a2=b2=3;
for(i=n;i>0;i--){
if((fp1=fopen("subf1","rb"))==NULL) return
if((fp2=fopen("subf2","rb"))==NULL) return
if((fp3=fopen("subf3","wb"))==NULL) return
if((fp4=fopen("subf4","wb"))==NULL) return
k=1<<(i-1);flag=0;
for(j=0;j<k;j++){
a1=b1=1;m=r=s=0;
t3=fread(a,sizeof(dfrec),t/3,fp1);
t4=fread(b,sizeof(dfrec),t/3,fp2);
while(1){
if(strcmp(a[r].key,b[s].key)==0){
strcpy(c[m].key,a[r].key);
c[m].year=a[r].year;
c[m++].addr=a[r++].addr;
strcpy(c[m].key,b[s].key);
c[m].year=b[s].year;
c[m++].addr=b[s++].addr;
}
else if(strcmp(a[r].key,b[s].key)<0){
strcpy(c[m].key,a[r].key);
c[m].year=b[r].year;
c[m++].addr=a[r++].addr;
}
else{
strcpy(c[m].key,b[s].key);
c[m].year=b[s].year;

```

```

c[m++].addr=b[s++].addr;
}
if(m>1000){
if(flag==0) fwrite(c,sizeof(dfrec),m,fp3);
else fwrite(c,sizeof(dfrec),m,fp4);
m=0;}
if(r>=t3){
if(a1<a2){
t3=fread(a,sizeof(dfrec),t/3,fp1);
r=0;a1++;}
else{
if(flag==0){
fwrite(c,sizeof(dfrec),m,fp3);
fwrite(&b[s],sizeof(dfrec),t4-s,fp3);
while(b1<b2){b1++;
t4=fread(c,sizeof(dfrec),t/3,fp2);
fwrite(c,sizeof(dfrec),t4,fp3);}
}
else{
fwrite(c,sizeof(dfrec),m,fp4);
fwrite(&b[s],sizeof(dfrec),t4-s,fp4);
while(b1<b2){b1++;
t4=fread(c,sizeof(dfrec),t/3,fp2);
fwrite(c,sizeof(dfrec),t4,fp4);}
}
break;
}
}
if(s>=t4){
if(b1<b2){
t4=fread(b,sizeof(dfrec),t/3,fp2);
s=0;b1++;}
else{
if(flag==0){
fwrite(c,sizeof(dfrec),m,fp3);
fwrite(&a[r],sizeof(dfrec),t3-r,fp3);
while(a1<a2){a1++;
t3=fread(c,sizeof(dfrec),t/3,fp1);
fwrite(c,sizeof(dfrec),t3,fp3);}
}
else
fwrite(c,sizeof(dfrec),m,fp4);
fwrite(&a[r],sizeof(dfrec),t3-r,fp4);
while(a1<a2){a1++;
t3=fread(c,sizeof(dfrec),t/3,fp1);
fwrite(c,sizeof(dfrec),t3,fp4);}
}
break;
}
}
}
if(flag==0) flag=1;else flag=0;
}
fclose(fp1);fclose(fp2);fclose(fp3);fclose(fp4);

```

```

if(remove("subf1")==-1) printf("remove error");
if(remove("subf2")==-1) printf("remove error");
if(rename("subf3","subf1")!=0)printf("rename error");
if(rename("subf4","subf2")!=0)printf("rename error");
a2*=2;b2*=2;
}
if(flag==0){
rename("subf2","subsort");
remove("subf1");remove("subf2");
}
else{
rename("subf1","subsort");
remove("subf2");remove("subf1");
}
}

```

三、评价

现在我们从时空两方面来分析上述分类方法。我们知道数组分类由于采用快速分类法或希尔分类法,无需额外的空间开销。但归并时在内存要定义三个数组,两个存待归并数据,一个存归并结果;磁盘上要设四个文件,两个为待归并文件,两个为归并结果,所以磁盘空间需要量是数据量的两倍。

在时间方面,内排序所花的开销主要在比较、移动和读写磁盘上,归并虽然也有上述三项开销,但相对而言主要还是读写开销。设记录总数为 m ,分为 2^n 组,则每组记录数为 $\frac{m}{2^n} = 2^{-n} \cdot m$,由于对每组采用快速分类法排序,所以第二步所需的总比较次数为:

$$2^n \cdot 2^{-n} \cdot m \log(2^{-n} \cdot m) = m \cdot \log(2^{-n} \cdot m)$$

移动总次数近似为:

$$2^n \cdot \frac{2^{-n} \cdot m}{6} \cdot \log(2^{-n} \cdot m) = \frac{m}{6} \log(2^{-n} \cdot m)$$

在归并过程中每趟至多将 m 个记录比较和复写一次,所以移动和比较次数至多为 mn 。但两段归并时其中必有一段的元素无需进行比较和移动,就能直接写回文件,所以实际的比较次数和移动次数必小于 mn 。而且归并是在数组之间进行,数组元素的移动当然要比移动文件记录的开销为小。另外,数组分类阶段需要对磁盘文件全部记录读写一次,然后每趟归并都需要对全体记录读写一次,所以记录在内外存储器之间的读写总次数为 $n+1$,由于本算法中归并趟数 n 很小,并且采用 Turbo C 一类语言编程,读写磁盘以几十 KB 为单位进行,从而使读写磁盘的开销受到很大的控制。

因此,尽管快速归并分类法并无理论上的突破,但由于巧妙地综合应用了现有的技术,仍取得了非常理想的效果。经测试,在 AST 386/33 微机对 14310 个主题词记录进行排序,总共只用了 17 秒钟,其中记录长度为 18 个字节,排序关键字为 13 个字节的字符型数据即主题词。该算法已在文献检索系统 SOCI 中采用。

动态图形库的建立及其应用

李明章 山东省莱州市三山岛金矿(261418)

目前,DFI扫描仪已广泛应用于数据库管理系统,如在人事档案图文管理系统中,可用来把职工的相片,身份证等图形信息输入计算机,以磁盘文件形式存储,可随时调用显示,并可和数据库记录中的文字信息同屏输出,实现图文并茂的显示效果,但由于每个图片都生成一个图形文件,对较大的人事部门,将产生数以千计的图形文件,大量结构相同的文件头及图形点阵信息中的多余字节,必将产生大量的数据冗余,造成计算机资源的浪费,为此,我们把各图形文件的点阵信息储存在一个总体文件——图形库中,图象只是库中的某个特定位置的记录而已,以下几个程序实现对图形库进行增添,覆盖,显示,并可在213H汉字系统下直接被dBASE,FoxBASE等数据库语言调用,通过PRG文件控制参数变化,对图形库进行操作,实现图文混屏输出:

一、图形库的建立:

程序libcreat创建图形库的头信息,图形库文件长6字节,为图形的宽和高,即图象块占据的列数和行数。

```
/* libcreat.c */
#include<stdio.h>
#include<io.h>
#include<fcntl.h>
#include<sys\stat.h>
main(int argc,char * argv[])
{
    int temp-int=0;
    int lib_file;
    char file_name[40];
    if(argc<3)
    {printf("\n 形如:图文档库名 记录宽 记录高")
    exit(0);}
    strcpy(file_name,argv[1]);
    if((lib_file=open(file_name,O_CREAT|O_BINARY,S-I-
    WRITE|S-IREAD))==-1)
    {exit(1);}
    write(lib_file,&temp-int,2);
    temp-int=atoi(argv[2]);
    write(lib_file,&temp-int,2);
    temp-int=atoi(argv[3]);
    write(lib_file,&temp-int,2);
    close(lib_file);}
```

二、追加记录:

程序libappen实现向图形库中追加图形,它需要的参数为图形库名,追加图形文件名及图形记录号,即

追加位置。当该记录号有图形记录时,将被覆盖。

```
#include<stdio.h>
#include<string.h>
#include<conio.h>
#include<fcntl.h>
#include<sys\stat.h>
#include<stdlib.h>
#include<graphics.h>
#define MAX-1 240
#define MAX-2 36
int IMG- WIDTH, IMG- HIGH, HORI- DOTS, DOT-
COLS;
unsigned char dot-matrix[MAX-1][MAX-2];
main(int argc,char * argv[])
{
    int width,height,infile,i,j,image-no=1,outfile;
    long offset;
    char filename[40];
    char filename1[40];
    unsigned char dot-byte,byte1,byte2;
    if(argc<=3{
        printf("\n 形如:图形库名 记录名(IMG图形文件名)图形
        记录号:");
        exit(0);}
    strcpy(filename1,argv[1]);
    strcpy(filename,argv[2]);
    image-no=atoi(argv[3]);
    image-no--;
    if(access(filename,0)){printf("%c",7);exit(1);}
    if((infile=open(filename,O_RDONLY|O_BINARY))==-
    1) exit(1);
    lseek(infile,(long)0x0c,SEEK-SET);
    read(infile,&byte1,1);
    read(infile,&byte2,1);
    width=byte1*256+byte2,width/=8;
    read(infile,&byte1,1);
    read(infile,&byte2,1);
    height=byte1*256+byte2;
    if((outfile=open(filename1,O-RDWR|O_BINARY))=-
    1) exit(1);
    read(outfile,&IMG- WIDTH,2);
    read(outfile,&IMG- WIDTH,2);
    read(outfile,&IMG- HIGH,2);
    offset=image-no*IMG- WIDTH*IMG- HIGH;
    lseek(outfile,offset,SEEK-CUR);
    setmem(&dot-matrix[0][0],MAX-1*MAX-2,0x00);
    for(i=0;i<min(height,IMG-HIGH);i++){
```



```

lseek(infile, (long)0x02, SEEK_CUR);
for(j=0; j<width; j++){
    read(infile, &dot_byte, 1);
    if(j<IMG_WIDTH) dot_matrix[i][j] = ~dot_byte;
}
for(i=0; i<IMG_HIGH; i++){
    write(outfile, &dot_matrix[i][0], IMG_WIDTH);
    close(infile);
    close(outfile);
}

```

三、定位显示:

程序 libdisp 实现图形的定位显示,它需要的参数有图形库名,要显示的图形记录号,显示位置坐标 x, y 及显示方式(1 正常/0 反白)。

```

#include<stdio.h>
#include<string.h>
#include<conio.h>
#include<fcntl.h>
#include<sys/stat.h>
void display_photo(char filename[], int image_number, int
xx, int yy, int zz)
{
    int width, high, file, i, j;
    long offset;
    unsigned char dot_byte;
    if((file=open(filename, O_RDONLY|O_BINARY)) == -
1) exit(1);
    read(file, &width, 2); read(file, &high, 2);
    read(file, &high, 2);
    offset = image_number * high * width;
    lseek(file, offset, SEEK_CUR);
    for(i=0; i<high; i++){
        for(j=0; j<width; j++){

```

```

        read(file, &dot_byte, 1);
        if(zz==1)
            pokeb(0xa000, (xx+i) * 80 + yy + j, dot_byte);
        else pokeb(0xa000, (xx+i) * 80 + yy + j, ~dot_byte);
        }
    }
    close(file);
}
main(int argc, char * argv[])
{
    char filename[40];
    int xx, yy, zz, image_no;
    if(argc<=5){
        printf("形如:图形库名 图形记录号 x 坐标 y 坐标 1 正常/0
反白显像");
        exit(0);
    }
    strcpy(filename, argv[1]);
    image_no = atoi(argv[2]);
    image_no--;
    xx = atoi(argv[3]);
    yy = atoi(argv[4]);
    zz = atoi(argv[5]);
    display_photo(filename, image_no, xx, yy, zz);
}

```

四、程序缺点:

尽管在 libcreat 图形库初始化程序中允许用户输入图形的高和宽,但图形库一经建立,各记录的长度就确定了,即各图形的宽和高就一定了,程序中没有考虑不等长图形的处理,感兴趣者不妨一试。

运行软硬件环境:

2.13H 汉字系统, VGA, DOS 3.3 DOS 5.0

扫描仪: DFI-HS3000 Plus 手持式黑白扫描仪, 生成 IMG 图形文件。

序号	编号	姓名	性别	工龄	用工制度	人员分类	等级	标准工资	日工资	计工资
0002	010101	倪贤臣	男	35	10	40	4.5	224.00	8.78	224.00
0003	010102	郭晓斌	男	15	1		9.5	140.00	5.49	140.00
0004	020101	栾树泰	男	30	1		3.5	244.00	9.57	244.00
0005	020102	丁增光	男	12	1		0.0	132.00	5.18	132.00
0010	020103	王清章	男	15	1		9.5	140.00	5.49	140.00
0006	020104	张建博	男	12	1		9.5	140.00	5.49	140.00
0007	020105	李萍	女	15	1		7.5	124.00	4.86	124.00
0008	020107	姜晓敏	女	1	1		0.0	0.00	0.00	0.00
0012	020201	康月红	女	1	2		0.0	0.00	0.00	0.00
0013	020202	张丽艳	女	1	16	62	0.0	0.00	0.00	0.00
0014	020203	王振强	男	1	10	62	0.0	0.00	0.00	0.00
0015	020204	刘军	男	1	21	62	0.0	0.00	0.00	0.00
0016	020205	李德彬	男	1	10	62	0.0	0.00	0.00	0.00
0017	020206	胡宝玲	女	1	21	62	0.0	0.00	0.00	0.00
0018	020207	刘金书	男	1	22	62	0.0	0.00	0.00	0.00
0019	020208	施清蕊	女	1	22	62	0.0	0.00	0.00	0.00
0020	020209									



HGC 卡高分辨率图形的屏幕拷贝

傅叔平 成都市成人教育学院 (610017)

一、问题的提出

目前, HGC(Hercules) 高分单显双频卡的使用十分普遍, 它具有的 720×348 高分辨率图形显示功能尤使它具有广泛的适应性, 因而获得很多的优秀软件的支持。但是 DOS 并不完全支持它的图形屏幕的硬拷贝。在 DOS 的支持下, 只能拷贝它在模拟 CGA 模式时的中分辨率 640×200 和低分辨率 320×200 的屏幕图形拷贝, 而对它用得最多的 720×348 高分辨率图形的屏幕拷贝却无能为力。为此, 作者采用下文所述的原理和方法很好地解决了这个问题。

二、基本原理和方法

在图形模式下, 屏幕上每一个点对应着内存显示缓冲区中的一位, 因此只须在图形屏幕对应的显示缓冲区中找到对应屏幕上点的相应数据, 再转换成打印机在图形打印方式下所要求的列方式的位图形数据序列, 直接送打印机端口输出。首先要搞清楚的是 HGC 卡显示缓冲区的结构及打印机的位图形打印方式。

在 720×348 高分辨图形方式下, 每一个扫描行 720 个点需要 90 个字节来表示。HGC 卡的显示缓冲区从地址 B0000H 开始, 共占用 $90 \times 348 = 31320$ 个字节, 其屏幕扫描线与显示缓冲区地址的关系采用了以每 4 线为模的方式, 因此内存显示缓冲区也分成了相应的 4 个区, 各区间隔为 2000H 字节, 每区的起始地址分别是: B0000H、B2000H、B4000H、B6000H, 各区内的地址是连续排列的, 它们的关系如下:

第 0 线: B0000H, B0001H, …… , B0059H

第 1 线: B2000H, B2001H, …… , B2059H

第 2 线: B4000H, B4001H, …… , B4059H

第 3 线: B6000H, B6001H, …… , B6059H

第 4 线: B005AH, B005BH, …… , B00B3H

第 5 线: B205AH, B205BH, …… , B20B3H

第 6 线: B405AH, B405BH, …… , B40B3H

第 7 线: B605AH, B605BH, …… , B60B3H

……

第 344 线: B1E3CH, B1E3DH, …… , B1E95H

第 345 线: B3E3CH, B3E3DH, …… , B3E95H

第 346 线: B5E3CH, B5E3DH, …… , B5E95H

第 347 线: B7E3CH, B7E3DH, …… , B7E95H

在位图形方式下, 打印针与位图形数据的对应关系见图 1 (为简明起见, 此处以九针打印机为例)。

从图 1 看到, 传送给打印机的位图形数据字节应为列方式, 而显示缓冲区的字节对应屏上的点是行方式, 必须进行转换, 而且由于打印头针距的限制, 拷贝出的图形的长宽比例与屏幕显示的正好相反, 所以作者采用了如下非常简捷的巧法:

再仔细观察显示缓冲区的地址排列方式, 假如我们把屏幕图形顺时针旋转 90° 后再送到打印机输出, 此时, 屏幕的左下角对应打印纸的左上角, 屏幕的右上角对应打印纸的右下角。这样一来, 最左一列字节变成了最上一行字节, 最重要的是, 这横排的一行字节的每个字节的各位对应的点恰恰成了打印机所需要的竖列方式, 各点与打印头的针的位置正好对应! 换句话说, 对字节不需再作任何转换, 只要根据显示区的结构, 找出各字节输出的先后顺序即可。而且, 这样拷贝下来的图形的长宽比例与屏幕显示接近。

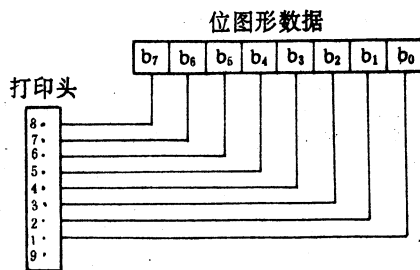


图 1

显然, 以这种方式输出时, 应按列从屏下方往上取字节, 然后作为一个行序列传送给打印机。另外, 适当选取字节的起止位置就可作屏幕的局部拷贝。

三、打印机输出

本文采用了直接向打印机端口传递数据的方法, 这样本文的程序独立于任何系统, 可在 PC 系列机的任何操作系统下使用。

四、程序实现

程序用 Turbo C 2.0 编制。屏幕拷贝函数 Copy-HGCscreen() 的四个参数 left, top, right, bottom 分别是屏幕拷贝范围的左、顶、右、底的座标参数, 为不增加程序的复杂性起见, 其中的 left, right 以 8 个点为单位, top, bottom 以 4 个点为单位。当 left=0, top=0, right=89, bottom=86 时则拷贝整个屏幕。读者还可根据打印机的使用说明书改变相应的打印机指令以不同的打印密度输出图形。

示例的主函数 main() 先在屏幕的左上部画若干个同心圆并以不同的方式填充各圆环, 然后将这个图形的局部在打印机上拷贝下来。图 2 是它的运行结果。

本程序在 Epson 1600K 二十四针打印机和 Epson 系列的九针打印机上均通过。

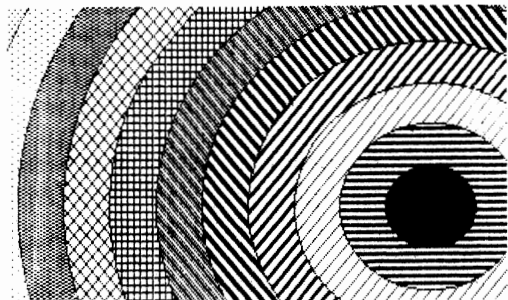


图 2

```
#include <graphics.h>
#include <conio.h>
main (void)
{
    void copyHGCScreen();
    int driver,mode,i,center;
    driver=DETECT;
    initgraph(&driver,&mode,"");
    center=50;
    for(i=1;i<12;i++)
    {circle(center,center,i*30);
    setfillstyle(i,WHITE);
    floodfill(center+(i-1)*30+1,center,WHITE);
    }
    getch(); /* wait until keypress */
    copyHGCScreen(0,0,40,40); /* screen copy */
}

void copyHGCScreen(int left,int top,int right,int bottom)
/* left,right;8 dots/unit; top,bottom;4 dots/unit */
{
```

```
void printer_out(int);
int lines,groups,totalgroup,linegroup,i,j,k;
unsigned CrtSegment,linehead,firsthead,a4,a44;
totalgroup=348/4;CrtSegment=0xb000;
groups=bottom-top;
firsthead=0x7e3c-(totalgroup-bottom)*0x5a;
lines=(bottom-top)*4;
printer_out(27);printer_out('A');printer_out(8);
for(linehead=firsthead+left;linehead<firsthead+right+1;
linehead++)
{a4=linehead;
printer_out(27);printer_out('K');
printer_out(lines%256);printer_out(lines/256);
for(j=0;j<groups;j++)
{a44=a4;
for(k=0;k<4;k++)
{printer_out(peekb(CrtSegment,a44));
a44-=0x2000;
}
a4-=0x5a;
}
printer_out(0x0a);
}
printer_out(27);printer_out('A');printer_out(12);
}

void printer_out(int prndat)
{int prnout_data,prnin_mode,prnout_ctrl;
prnout_data=peek(0x40,0x08);
prnin_mode=prnout_data+1;
prnout_ctrl=prnout_data+2;
outportb(prnout_data,prndat);
while(! (inportb(prnin_mode)&0x80)); /* printer busy */
outportb(prnout_ctrl,0x1f);
outportb(prnout_ctrl,0x1e);
}
```

中文界面设计与 SPT 图像

张少猷 北京理工大学 40914 班(100081)

在计算机软件开发中,界面的设计,尤其是中文界面的实现显得很重要。市面流行的中文游戏中充分体现了这一点。如游戏开始的版权、游戏名称、简介,都以各种精美汉字显示,给人以亲切的感觉。

如果用 C 语言在西文状态下用点、线绘制这些汉字,可是一项非常烦琐的工作。而直接读取点阵字库显示汉字,也有它的局限性。首先是字体呆板,因为一般

点阵字库只有宋体或仿宋体;其次是字的大小受限制,虽然可使用一些放大平滑技术,但毕竟费时、费力,且效果不好。所以,我们应该使用一些现成的造字软件。

WPS 文字处理系统是目前国内使用得比较广泛的文字处理系统,我们可以调用其中的汉字图文编排系统 Super-Star V1.10(SPT)生成大小不同、形式各异的汉字图形,而后利用这些图形数据直接写屏,在屏

幕上实现精美的汉字界面。

本文对 SPT 图像格式进行分析,介绍其调用方法,并附上完整的程序清单,实现 SPT 图像在 EGA 中的彩色显示。

一、原理

a. *.SPT 图像数据的结构:文件偏移量 0~39h 的 64 个字节为文件头,包括文件类型、作者姓名等。其中图像宽度值在偏移量 22h~23h 处,高度值在 24h~25h 处。长度和宽度以像素点为单位。

从偏移量 40h 开始为图像数据。数据按图像从左到右,从上到下描述。因为 SPT 为黑白二值图像文件,所以数据长度为图像长度×图像宽度/8 个字节。

b. EGA 方式 16 色显示原理:在 EGA 16 色图形方式下,用到 4 个颜色平面,分别代表蓝(B)、绿(G)、红(R)、亮度(I)。这四种颜色平面叠加,可合成 2^4 种颜色。因为它们都占用相同内存地址: A000:0000h,所以需要用时序发生器中颜色平面允许写寄存器,设置需要写的颜色平面。详细资料可见《EGAVGA 程序员手册》(北京大学出版社)。

二、界面设计过程

①在 Super-CCDOS 状态下进入汉字图文系统 SPT;

②根据实际需要设定版面宽和高;

③选择“文字编辑”中汉字输入功能,并设置好字体(楷体、宋体、黑体等);字号(0 号、1 号,自定义等);

④把闪烁的方框移到版面中合适位置,然后进行汉字输入。同时还可利用 SPT 中的各种作图功能,进一步美化版面;

⑤存盘时,选择.SPT[S]格式,非压缩方式[E]来存。

至此,精美的汉字界面生成了,可以直接利用本文所附程序在西文状态下重现。如果把本程序稍加改制成函数,并把 *.SPT 图像的文件头、文件体做一些加密处理,便可作为版权页使用。

本程序在 AST Premium/286 上,用 Turbo C2.0 的 Compact 方式编译运行通过。

```
/* VIEWSPT.C */
#include "stdio.h"
#include "graphics.h"
#include "dos.h"

void init();
main()
{int m[2];
 unsigned int wide,high,size;
 char key;
 char far *buf;
 char name[13];
 FILE *fp;
 int i,j,col;
 clrscr();
```

```
gotoxy(18,10);
printf("Which file do you want see [ .SPT ],");
scanf("%s",name);
gotoxy(20,12);
printf("Which color do you want[0--15],");
scanf("%d",&col);
fp=fopen(name,"rb");if(!fp){printf("\n7\n7\n7Read
error\n");exit(1);}
fseek(fp,0x22,SEEK_SET);
fread(m,sizeof(int)*2,1,fp);
wide=m[0]+2;high=m[1]+2;
gotoxy(22,14);
printf("Width: %d; Height: %d;",wide,high);
gotoxy(22,24);
printf("Press any key to continue.");
key=getch();
if(key=='q'){printf("EXIT by user! \n");exit(0);}
size=(int)(wide/8);
init();
outportb(0x3c4,2);
if(col&0x1){buf=(char far *)0xa0000000;
 outportb(0x3c5,1);
 fseek(fp,0x40,SEEK_SET);
 for(j=0;j<high;j++){fread(buf,size,
 1,fp);buf+=80;}}
if(col&0x2){buf=(char far *)0xa0000000;
 outportb(0x3c5,2);
 fseek(fp,0x40,SEEK_SET);
 for(j=0;j<high;j++){fread(buf,size,1,
 fp);buf+=80;}}
if(col&0x4){buf=(char far *)0xa0000000;
 outportb(0x3c5,4);
 fseek(fp,0x40,SEEK_SET);
 for(j=0;j<high;j++){fread(buf,size,1,
 fp);buf+=80;}}
if(col&0x8){buf=(char far *)0xa0000000;
 outportb(0x3c5,8);
 fseek(fp,0x40,SEEK_SET);
 for(j=0;j<high;j++){fread(buf,size,1,fp);
 buf+=80;}}
fclose(fp);
getch();
restorecrtmode();
}

void init()
{int driver,mode;
 driver=DETECT;mode=0;
 initgraph(&driver,&mode,"");
}
```

方便地读写硬盘主引导扇区

赫 建 第一汽车集团公司模具中心(130011)

硬盘主引导扇区是一个非常重要的部位,各种引导型病毒都攻击此处。另外,各种人为或意外软、硬件故障也常破坏该处,致使系统不能自举,甚至硬盘内容无法取出,造成重大损失。因此,妥善保存和及时恢复该区内容是很有必要的。

笔者用汇编语言编制了一个短小程序,专门用于读写主引导扇区内容,已经应用两年多,效果良好,多次处理险情,化险为夷。现介绍如下:

首先使用编辑软件建立汇编语言源程序 CHDOS. ASM;经编译、连接生成相应的. EXE 文件;再经 EXE2BIN 程序转换成相应的. COM 文件;最后将 CHDOS. COM 程序拷贝到经常使用的 DOS 系统盘上,以后便可以当作 DOS 系统的一个工具软件使用。

使用时键入 CHDOS;输入文件名(可含盘符和路径);选择读写方式(R—读,W—写)。执行结果有提示。

建议:在各微机正常时读取其主引导扇区内容,并由文件名加以区别,妥善保管(最好备份两份以上),以防不测。硬盘重新分区须更新备份主引导扇区内容。

```
CODE    SEGMENT PARA PUBLIC 'CODE'
        ASSUME CS:CODE,DS:CODE
START   PROC FAR
        LEA DX,MSG1
        MOV AH,9
        INT 21H
        LEA DX,KBUF
        MOV AH,0AH
        INT 21H
        LEA DI,FILE
        XOR AH,AH
        MOV AL,BYTE PTR KBUF+1
        ADD DI,AX
        MOV AL,0
        MOV [DI],AL
        LEA DX,MSG2
        MOV AH,9
        INT 21H
        MOV AH,1
        INT 21H
        AND AL,5FH
        CMP AL,'R'
        JZ  READ
        CMP AL,'W'
        JZ  WRITE
```

```
        INT 20H
READ:   MOV AX,201H
        MOV BX,OFFSET DOS
        MOV CX,1
        MOV DX,80H
        INT 13H
        LEA DX,FILE
        MOV CX,0
        MOV AH,3CH
        INT 21H
        JC  ERROR
        MOV BX,AX
        LEA DX,DOS
        MOV CX,512
        MOV AH,40H
        INT 21H
        JC  ERROR
CLOSE:  MOV AH,3EH
        INT 21H
        LEA DX,MSG4
        MOV AH,9
        INT 21H
EXIT:   INT 20H
ERROR:  LEA DX,MSG3
        MOV AH,9
        INT 21H
        INT 20H
WRITE:  LEA DX,FILE
        MOV AX,3D00H
        INT 21H
        JC  ERROR
        MOV BX,AX
        LEA DX,DOS
        MOV CX,512
        MOV AH,3FH
        INT 21H
        JC  ERROR
        MOV AX,301H
        MOV BX,OFFSET DOS
        MOV CX,1
        MOV DX,80H
        INT 13H
        JMP CLOSE
START   ENDP
MSG1    DB 0AH,0DH,'Input DOS'
        DB 'bootstrap filename: $'
MSG2    DB 0AH,0DH,'Input read'
```

```

DB      'or write(r/w)=$'
MSG3    DB      0AH,0DH,'File error! $'
MSG4    DB      0AH,0DH,'Operation ok! $'
KBUF    DB      30
        DB      0

```

```

FILE    DB      30 DUP(0)
DOS     DB      0
CODE    ENDS
        END      START

```

使用金山 WPS 3.0F 系统配置文件的设置

何奇继 广州通讯设备厂产研二所(510630)

北大方正集团公司与香港金山公司共同开发的中文系统 SPDOS 6.0F 和文字处理软件包 WPS 3.0F, 由于其操作简便、功能齐全, 界面友好、实用方便, 越来越赢得广大用户的使用和信赖。

但由于金山 WPS 3.0F 占用较多的常规内存, 而一般用户的计算机都装有 MOUSE、DOSKEY 等设备驱动程序, 用户往往由于系统配置文件 CONFIG.SYS 和批处理文件 AUTOEXEC.BAT 的设置不当, 造成在 WPS 文件编辑排版中, 屏幕汉字显示缓慢或无法模拟显示(提示内存不够)。要经常修改 CONFIG.SYS 和 AUTOEXEC.BAT 文件, 非常麻烦!

其实, MS DOS 6.0 提供的内存管理程序, 可以让用户在启动计算机时根据自己将要运行的任务选择系统内存配置。用户最好将设备驱动程序、五笔字型驱动程序装入上位内存(UMB), 以留出更大的可用常规内存供用户程序使用。

下面给出 MS DOS 6.0 下的 CONFIG.SYS 和 AUTOEXEC.BAT 文件的设置供用户参考。在启动计算机时, 配置 1, 主要为运行中文版 Windows 3.1 软件提供内存环境; 配置 2, 主要为运行中文系统 SPDOS 6.0F 和 WPS 3.0F 文字处理软件包提供内存环境。该批处理文件将 FASTOPEN、DOSKEY、QMOUSE 等驱动程序装入上位内存(用命令 LH), 留给用户多达 619KB 的常规内存。用户可根据自己计算机的实际环境修改使用。修改之后, 可用 MEM/C 或 MEM/D 等外部命令察看可用常规内存的大小。

CONFIG.SYS

[menu]

menuitem=windows-config, WINDOWS CONFIG SETUP; 选项 1

menuitem=spdos60-config, WPS 3.0F CONFIG SETUP

; 选项 2

menucolor=15,1; 白色前景, 蓝色背景

menudefault=1,10; 保持计数 10 秒

[windows-config]

DEVICE=C:\DOS\HIMEM.SYS; 安装扩展内存管理程序
 DEVICE=C:\DOS\EMM386.EXE RAM HIGHSCAN
 ; 支持扩充内存和 UMB 区的存取

BUFFERS=15,0

FILES=30

DOS=UMB; 指定 DOS 保持与上位内存(UMB)的链接
 LASTDRIVE=E

FCBS=4,0; 允许 DOS 同时打开 4 个文件控制块

DOS=HIGH; 指定 DOS 保持与扩展内存的链接

DEVICEHIGH=C:\DOS\SETVER.EXE

; 安装版本设置驱动程序到上位内存

SHELL=C:\DOS\COMMAND.COM/P/E:1024

; 指定命令解释程序及环境大小

STACKS=9,256; 支持动态地使用堆栈去处理硬件中断

SWITCHES=/K; 使增强型键盘起常规键盘的作用

[spdos60-config]

DEVICE=C:\DOS\HIMEM.SYS

DEVICE=C:\DOS\EMM386.EXE NOEMS

; 支持上位内存访问, 但阻止扩充内存访问

DOS=HIGH,UMB

BUFFERS=15,0

FILES=30

AUTOEXEC.BAT

@ECHO OFF

PROMPT \$p\$g

SET PATH=C:\WINDOWS;C:\DOS;C:\MASM;C:\WPS60;D:\TG

SET TEMP=C:\WINDOWS\TEMP

LH C:\DOS\SMARTDRV.EXE

; 安装 SMARTDRV 到 UMB 区, 启用默认值

LH C:\DOS\FASTOPEN C: = 50 D: = 50; 安装 FASTOPEN 到 UMB 区

LH C:\DOS\DOSKEY/BUFSIZE=50/INSERT/HISTORY; 安装 DOSKEY 驱动程序到 UMB 区

LH C:\DOS\QMOUSE/2/2

; 安装 QMOUSE 驱动程序到 UMB 区

软盘容量扩充技术

邱向群 西安交通大学电气工程系(710049)

一般的 5.25 英寸低密软盘经标准格式化后,分 40 磁道,共 362496(即 360K)字节,下面就介绍几种将软盘容量稍加扩充的方法。

一、利用 PC 工具进行扩充容量格式化。

将低密盘放入高密驱动器中,进入 PC 工具后,选格式化参数时将其作为高密 1.2M 盘格式化。这是最简单的一种扩充软盘容量的方法,但也是最不可靠的。一盘一张 360K 盘用此法可以格式化为 900K 左右,但一般来说仅有不到 500K 的容量是可用的,而且在 400K 以后的容量中存放数据有丢失的危险。

二、修改磁盘 I/O 参数表

IBM PC 及其兼容机使用的 5.25 英寸双面双密软盘标准格式是每面 40 个磁道,每道 9 个扇区。双面双密软盘为 48TPI,即每英寸 48 个磁道,5.25 英寸软盘的读写窗宽度为 1 英寸,因而每面可以容纳 48 个磁道。0—39 磁道的互换性(指磁盘机和磁盘的互换性)较好,内圈 8 个磁道互换性较差,故而舍去不用,标准格式化程序只用到 40 个磁道。但实际上一般的磁盘机都可以正常读写 44 个磁道,即 0—43 道。有些原装高质量磁盘机甚至还可以多读写几个磁道。为了保证数据的安全,建议以使用到 48 磁道为限。

由于标准的格式化程序只能格式化出 40 个磁道,所以额外磁道的生成要自己编程实现。下面介绍一种将 5.25 英寸双面双密磁盘格式化为 44 磁道,使总容量增加 10% 的方法。(以下修改适用于 DOS 2.0)

1. 取一张未格式化的双面双密软盘,放入 A 驱动器中,执行以下操作(假定当前操作目录中有 FORMAT.COM 和 DEBUG.COM 文件):

- ①C>FORMAT A:/S
- ②C>COPY FORMAT.COM A:
- ③C>COPY DEBUG.COM A:

2. 修改 FORMAT.COM 程序的指定字节:

A>DEBUG FORMAT.COM

- E 1312 C0
- E 1300 B0
- E 1336 18 03
- E 1324 8C
- E 0FFA 2C
- W

Writing 1780 bytes

-Q

3. 修改 IBMBIO.COM 文件属性,使其可被改写:

A>DEBUG IBMBIO.COM

-L 0051

-EB 20

-W 0051

-Q

4. 修改 IBMBIO.COM 文件指定字节:

A>DEBUG IBMBIO.COM

-E 054A 60

-E 055D 2C

-W

Writing 1200 bytes

-Q

5. 恢复 IBMBIO.COM 文件属性,使其为隐含:

A>DEBUG IBMBIO.COM

-L 0051

-EB 27

-W 0051

-Q

(4,5 二步可以通过 PC 工具来完成,这里就不详细介绍了)

6. 用这张修改过 IBMBIO.COM 的 A 盘重新启动机器,将一张未格式化过的双面双密盘放入 B 驱动器中:

A>FORMAT B:

格式化完之后 B 盘即为 44 个磁道的软盘。

三、利用现成的软盘扩充程序:

意大利人 A. Pasquale 编写过一个 800.exe 的程序(笔者使用的是 1.40 版),这是一个内存驻留程序(仅占 864bytes),运行后可将软盘格式化出几种非标准容量:

命令格式	格式化后容量
FORMAT [B:] /T:40/N:9	360KB
FORMAT [B:] /T:40/N:10	400KB
FORMAT [B:] /T:80/N:9	720KB
FORMAT [B:] /T:80/N:10	800KB
FORMAT [B:] /T:80/N:15	1200KB
FORMAT [B:] /T:80/N:17	1360KB

以上四条命令可对 5.25 英寸高密盘进行操作,其中前两条可对 3.5 英寸低密和高密盘进行操作,后两条对 3.5 英寸高密盘进行操作。

FORMAT [B:] /T:80/N:18	1440KB
FORMAT [B:] /T:80/N:20	1600KB

以上两条命令仅可对 3.5 英寸高密盘进行操作。

800.exe 程序运用起来是十分方便的,运行

800.exe 后,原来 DOS 中的 FORMAT 命令即会被自动修改,只需按上述方法输入命令即可。例如希望将一张 5.25 英寸低密盘格式化为容量为 400K 的启动盘,可按以下步骤进行:

①运行 800.exe

②键入:FORMAT A:/S/T:40/N:10

如果希望恢复为原来 DOS 的 FORMAT 命令,重新启动一次即可。800.exe 在将磁盘格式化为 400KB,800KB,1360KB 及 1600KB 这些非标准容量时,速度会慢一些。800.exe 是一个十分有用的软件,不仅可将磁盘格式化为非标准容量,还有许多实用的功能。例如运行 800.exe 后,原来 DOS 的 Diskcopy 和 Diskcomp 这两条仅可对相同磁盘格式操作的命令,就不受此限制了,例如可以将 A 驱动器中 5.25 英寸盘直接用 Diskcopy 拷贝到 B 驱动器中的 3.5 英寸盘中了:

C>diskcopy a: b:

这样就使操作极为方便。其它一些功能限于篇幅就不介绍了。800.exe(1.40 版)支持 DOS 3.30 及其以上版本,它是一个非加密程序,程序中说明此程序可以自由拷贝。

四、结论:

磁盘容量的扩充从软磁盘结构及磁盘机工作原理上来说是完全可行的。只要不超过其可格式化容量范围,数据相对来说也是可靠的。一些将磁盘硬格式化的方法,例如本文介绍的方法一,或用 800.exe 将 5.25 英寸低密盘格式化为 720KB 或更多的容量应该极力避免的,这样做不但存放的数据极不可靠,而且易造成软盘损坏。将软盘容量适当的扩大,不但可以多存储数据,而且还可以起到简单加密的功能。

PC Tools 9.0 功能简介

姜 宏 北京大学 9 公寓 107 室(100871)

1993 年,Central Point 公司推出了 PC Tools 工具软件系列产品的最新版——9.0 版。其功能之强,用途之广前所未有。它具有的强大功能和优良的用户界面,成为广大 PC 机用户的必不可少的工具软件。在 92 年的 8.0 版的基础上增强了磁盘优化、修复、辑毒和拷贝的功能,并且能更好地支持 Windows 3.x,它的 For Windows 版更有 Windows 4.0 的美誉。它支持 MS DOS 6.0(6.2)的 Double Space 功能和 Stacker 3.0(3.1)的硬(软)盘增容功能,这是 8.0 版所不具备的,它还可以将 DOS 6.0 的 Dblspace 盘还原,使得用户更加灵活地保存数据资源。下面就一些新增的功能做一介绍。

一、查毒工具——CPAV 2.0

在 PC Tools 9.0 中,重点增强了对病毒的防范:不论是在安装或是其它程序执行前。在主菜单中选择 Anti Virus 功能,即可进入 CPAV 2.0,这是一个最新的病毒查找软件,共可清除 2026 种病毒,包括引导扇区病毒,文件病毒和多媒体病毒以及网络病毒等。对某些采取了所谓“反跟踪”的病毒实行“反窃取”(Anti-Stealth),另外,为区别以往的 CPAV 1.X,PC Tools 9.0 的检验文件更名为 SMARTCHK.CPS,以示区别。

有一些软件的安装盘上如果染上了病毒,那么它上面的压缩文件将是不安全的,对这一点来说,以往的查毒软件就显得无能为力了。可 PC Tools 9.0 提供了压缩软件查毒功能,一些常用的压缩软件文件格式它都能识别。下面列出的几个是较常见的。

文件后缀

所用的压缩软件及较新版

.ARJ	ARJ 2.41a
.LZH	LHA 2.14
.ZIP	PKZIP 2.39
.ICE	ICE 2.01
.ZOO	ZOO 2.00
.ARC	ARC 2.30

有一些 EXE 文件采用较特殊的方法压缩,使得压缩后文件仍能正常运行。PC Tools 也对这类文件提供了可靠的检验手段,这类软件国内以 LZEXE 和 PKLITE 为常见。

PC Tools 9.0 还为可以查找的 2026 种病毒的查看提供了详细的资料,只需按下 F9 后即可查看。

二、磁盘优化工具

用户的磁盘,尤其是硬盘,经过一段时间的读、写、删操作后,许多文件,哪怕是几万字节长的文件也很可能在分散的簇上存放,时间久后,硬盘速度会明显下降。用户可以用 DOS 命令 MIRROR d:做一次硬盘映像存放,如果用时超过 2 秒,则应考虑使用 PC Tools 9.0 提供的 Optimizr 优化工具。它可以通过选择把硬盘可用空间移到后面,让数据优先存放。

这个功能 PC Tools 8.0 和 MS-DOS 6.0 及 6.2 也有,但用户如果用了类似 Double Space 和 Stacker 的硬盘增容系统,这些软件便不起作用,但 PC Tools 9.0 提供了对 Dblspace 盘和 stac-vol 盘的优化功能,让用户更放心地使用这些软件。顺便说一句,MS-DOS 6.0 的 Double Space 功能是不完善的,用户的数据可能丢失!MS-DOS 6.2 版和 Stacker 3.x 解决了上述问题,因此,用户应该考虑更新你的 Dblspace 版本。

在 PC Tools 9.0 的 Optimizr 硬盘优化工具中,用

户可以有如下选择:

1. 将已分配的目录移到磁盘开头,并不完整化文件。用户可在进行目录操作后使用此项。

2. 将已分配的簇移到磁盘开头,也不完整化文件。在用户删除或拷贝大文件之前,应选择此项以确保文件连续分配。

3. 完整化文件。但并不将之移至磁盘开头,这个选择相对比较省时,但须频繁使用才能显示出优越性。

4. 完整化文件并将其移到磁盘开头,它可以确保用户在相当长的时期内不用再次优化磁盘,但较费时间。

5. 在(4)基础上将未用簇清理,以增加可用簇,此选项最费时间。

由此可见,PC Tools 9.0 为用户提供了一个相当强大的磁盘维护工具。

三、其它功能简介

1. 快速拷贝。用户可以用 FASTCOPY 命令高速拷贝软盘。

2. 高级格式化。用户可将 1.2M 盘格式化成为 1.68M 容量。

3. 系统检验。用户可以全面检查主机,包括 CPU

速度,驱动器速度和整体速度的相对比较,可靠性高。

4. 高速备份。CPBACKUP 程序可以以 2M/秒的速度备份数据。

5. 高级加密。用 PCSECURE 加密的文件未经加密者解密,任何人不能使用。

6. 磁盘修理。DISKFIX 的新版比旧版提供了更为行之有效的修理办法。

7. 字处理。包括对文件中单词拼写的检验。

8. 支持 Windows 3.x。Window 3.x 可以使用任何 PC Tools 9.0 的实用程序。

9. 强大的宏(Marco)语言。

10. 高速缓冲程序 PC-CACHE 及内存管理学习程序 RAMBoost。

11. 支持 TVGA、SVGA 显示方式。

以上对 PC Tools 9.0 版工具软件进行了简单介绍,希望用户能够对它的功能有一个全面大致的了解,为自己的计算机配备高级的工具软件。PC Tools 9.0 中许多功能这里因篇幅所限未能介绍,读者如有问题或是要交流软件使用经验,欢迎与作者联系。PC Tools 9.0 是一个全新的工具软件,对它的许多功能都未能深入了解,本文不妥之处,敬请指正。

一种新病毒的分析与防治

高峰 太原市公安局计算机通信处(030001)

前不久,笔者在上机时发现了一种新病毒。由于该病毒隐蔽性较好,而且表现部分最早也要等到 1994 年 9 月 9 日以后才发作,所以还没有引起用户足够的重视。据笔者所知,已经有为数不少的机关、学校和企业的计算机染上了该病毒,给用户造成了潜在的危害。经过分析,笔者基本上掌握了该病毒的防治方法,特借本刊一角介绍给大家。同时需要说明,金辰公司最新的 KILL68 可以消除该病毒,并称之为 Dabi 病毒。

一、病毒的危害和表现

虽然病毒编制者原来的想法并不想对系统运行造成影响,但病毒中许多地方使用了远跳转指令,这样在病毒的执行、传染过程中,字节有时就自行发生变异。例如病毒驻留时的 RETF 指令(操作码 CBH)变异成 RET 指令(操作码 C3H)或 RETF * * * * 指令(操作码 CAH),都是因为某一位发生变化。这些变种病毒就极易引起系统死机。笔者正是在系统死机的情况下发现该病毒的。在笔者分析病毒的过程中,也发生过远跳转指令造成字节变异的情况。此外,传染上该病毒的计算机系统,在运行时速度有所减慢,这一点在用 DIR 命令列目录时便可察觉。

病毒有一个演奏音乐的表现部分,发作时间是 1994 年(或以后某年)的 9 月 9 日或 12 月 26 日,这两

天是毛泽东主席逝世和诞辰纪念日。发作条件是病毒进入内存 1 小时后,具体表现为 9 月 9 日演奏乐曲《东方红》,12 月 26 日演奏乐曲《浏阳河》。如果不修改机器日期,乐曲将一直演奏下去,但一般不影响程序的其它操作,除非用户程序中也需演奏音乐,这时扬声器将优先演奏用户程序的音乐,在间隙中演奏病毒音乐,使用户程序的音乐受到很大的干扰。

二、病毒的传染

Dabi 病毒寄生在文件的尾部,长度为 1465 字节,它主要靠以下四条途径传染 .COM 和 .EXE 文件。

① 拷贝已带毒的文件。

② 运行带毒的文件。这种途径只能传染 C 盘根目录下的 COMMAND.COM 文件。

③ 在内存中含有病毒的情况下查看文件(主要指用 DIR 命令,也包括其它带有搜寻和匹配功能的命令)。这种途径传染范围最广,是主要的传播途径,所有被查看的未带毒的 .COM 和 .EXE 文件均被传染。

④ 用带毒的软件编译可执行文件。新生成的可执行文件和 C:\COMMAND.COM 将被传染。

三、病毒的感染过程

Dabi 病毒在计算机中以两种形式存在,我们姑且把存在于内存中的称为动态,附着在文件尾部的称为

静态。病毒在动态时具有主动传染能力,凡在这时执行或被查看的可执行文件都将被传染(已带毒的除外);病毒在静态时只能通过文件拷贝或文件调用被动地传染。动态病毒通过系统重新启动就可以消除,而静态病毒只能通过截取或删除文件消除;静态病毒通过执行带毒文件驻留内存成为动态病毒,动态病毒通过传染附着在文件尾部成为静态病毒。

在由静态到动态的转换中,Dabi病毒通过一个特有的标志来避免重复驻留内存,即修改INT 21H中30H功能调用BL的返回值。正常状况下BL应当返回00H,病毒将其修改为5BH,并在病毒体的最前部执行这一功能调用,如果BL返回值为5BH,则转向执行正常的原文件;否则病毒首先将自身驻留内存高端,然后修改INT 21H和INT 8H两个中断向量的入口地址,保存INT 24H的入口地址,最后对C盘根目录下的COMMAND.COM进行传染。

对C:\COMMAND.COM的传染过程也适用于其它的.COM或.EXE文件,只有一点例外,就是病毒编制者没有考虑无C盘或C盘根目录下无COMMAND.COM文件的情况,目前的计算机也几乎不可能出现这两种情况。因此一旦遇上这两种情况,计算机运行带毒文件时就会死机。

对可执行文件的传染同样也有标志来避免重复传染。病毒一旦传染某个文件,就根据它是.COM还是.EXE文件设置标志,使文件只被传染一次。具体来说,.COM文件的标志是:文件的第四、五字节所组成的字是由第二、三字节所组成的字循环右移一次得来的;.EXE文件的标志是:SP寄存器的初值与IP寄存器的初值之差为693H。

如果文件中没有上述两种标志,病毒先把文件的属性和时间取出保存,文件属性改为读/写,然后把已存在内存中的病毒体链接在正常文件的尾部,对文件头做相应的修改使其再次执行时能先跳转到病毒部分执行,后执行正常程序,同时置好避免重复传染的标志,最后将正确的属性和时间置回。如果文件中已经有标志,就转向执行正常的程序。

当病毒完成了驻留内存和传染C:\COMMAND.COM的过程后,就通过修改后的INT 21H来监视系统的动作,INT 21H一共被修改了4个功能调用:

- ① AH=4BH,AL=00H,(装载且执行)。
- ② AH=11H,(搜索第一个匹配文件)。
- ③ AH=12H,(搜索下一个匹配文件)。
- ④ AH=30,(取DOS版本号),该功能调用即前面所提到的第一种标志。

以后当未带毒的.COM或.EXE文件被执行或查看,都将被内存中的病毒察觉并传染,传染过程与传染C:\COMMAND.COM相同,然后转向执行正常的程序或动作。

Dabi病毒的表现部分即演奏乐曲部分很有特点。它修改了INT 8H,利用该中断每秒被调用18.2次达到奏乐的目的。它用一个字节同时包含音符代码和节

拍,前4位代表音符,后4位代表节拍,即前4位所代表的音符的演奏时间,而不是象一般方法那样,把节拍和音符代码分成两组字符串。

四、病毒的隐蔽技术

Dabi病毒的隐蔽技术主要表现在三方面:首先,在内存中驻留有病毒的情况下,用DIR命令查看到的文件长度和时间均与正常情况无异;其次,病毒程序代码中的一些关键部分进行了加密处理,病毒执行时先自行解密,调用结束后再进行一次加密处理,这样即使使用DEBUG等工具发现了病毒,也不容易读懂。再次,病毒在传染时需要写盘,于是它将INT 24H中断程序修改为只执行IRET一条语句,这样即使遇到带有写保护的磁盘,计算机也不会发出严重错误警告,病毒也就得以隐蔽。

由于病毒没有采用诸如修改INT 01H破坏跟踪等手段,因此以上方法不会给破译病毒带来太大的麻烦,这也就是为什么称之为“隐蔽技术”而不是“反跟踪技术”的原因。

五、病毒的查寻与消除

查寻内存中是否含有Dabi病毒可以通过检查其标志进行,利用DEBUG来查看执行INT 21H的30H功能调用后BL的返回值。

```
C>debug
-a100
xxxx:0100 mov ah,30
xxxx:0102 int 21
xxxx:0104 int 3
xxxx:0105
-g=100
-rbx
```

如果BL的值为5BH,则说明内存中已含有病毒。

对一个文件的检查需要检查其长度。用DIR命令查看文件并记录下其长度,然后再用PC Tools查看其长度,如果看到的长度(文件的真实长度)比记录的长度长了1465(5B9H)字节,则说明该文件已传染上了Dabi病毒。如果手头上暂时没有PC Tools,则可用干净的系统盘重新启动,用DIR命令查看也可以看到真实的文件长度。注意查看前不要再运行任何程序以防病毒驻留内存。

消除内存中的Dabi病毒可以采用释放内存块的方法,或用干净的系统盘重新启动。在保证内存无毒的情况下才能针对不同类型的可执行文件用无毒的软件工具如DEBUG进行消毒。

(一)对.COM文件:

1. 用rcx命令求出带毒文件长度。
2. 将此长度减去1465字节,得到正确的文件长度。
3. 从带毒文件尾部向前移动149字节,从此时所得到的地址开始的5个字节即为经过加密的原文件开头的5个字节。
4. 将这5个字节依次与05h、04h、03h、02h、01h相异或。

5. 将异或后的 5 个字节顺次填回文件头部。(从 0100h 开始)。

6. 用 rcx 命令置入正确的文件长度。

7. 用 W 命令将正确文件写回磁盘。

(二)对 .EXE 文件

文件消毒前先将文件的扩展名改为非 exe, 然后根据文件长度区别对待, 如果文件长度小于等于 64k, 则 1、2 步同(一), 以下略有不同:

3. 从带毒文件尾部向前移动 149 字节, 从此时所得到的地址开始的 10 个字节即为经过加密的原文文件头偏移 0eh~17h 处的 10 个字节。

4. 将这 10 个字节依次与 0ah、09h、08h……02h、01h 相异或。

5. 将异或后的 10 个字节顺次填回文件头部。(从 010eh 开始)。

6. 把正确的文件长度除以 512(200h)后所得到的余数(一个字)放入文件头偏移 02h 处, 所得到的商(如果余数不为零, 则商加 1)放入文件头偏移 04h 处。

注意: 存放时低字节在前, 高字节在后。

7. 用 rcx 命令置入正确的文件长度。

8. 用 W 命令将正确文件写回磁盘。

9. 将文件名改回。

对于带毒文件长度大于 64K 的情况, 第 3 步建议先用 PC Tools 查找下列字符串:

L4QODGDIIB+GLO

从该字符串头开始向前移动 132 个字符, 同样也可以得到经过加密的原文文件头偏移 0eh~17h 处的 10 个字节。以下从第 4 步开始同前。

以上手工消毒的缺点是比较繁琐, 另外容易遗漏隐含文件和直接对只读文件进行消毒, 这都会造成消毒不彻底。因此建议先用 PC Tools 查看文件的带毒情况, 并将隐含文件和只读文件的属性改为正常的读/写属性, 待消毒后再将属性改回。

依据以上原理, 笔者编制了解毒程序, 已经在 AST386、联想 486 微机上调试通过。由于篇幅所限, 源程序从略。

在中华学习机上真正实现子目录功能

秦燕军 上海市番禺中学(200052)

在 PC 机操作系统的发展过程中, 从 PC-DOS 2.0 版起做出了重大改革, 提供了一种新的目录结构, 即由根目录和多级子目录一层层展开形成的树形分叉目录结构。引入树形结构技术的好处, 一是可以对相关的文件进行归纳, 使得主目录变得简洁明了; 二是可以便利多用户使用而不会相互干扰, 总之, 就是便于对磁盘上的文件进行管理。但可惜的是, 在 CEC DOS(即 Apple DOS 3.3)中却没有实现这种树形目录结构, 使中华学习机的用户颇感不少遗憾。

Apple 机(CEC 的仿制前身)曾拥有大量的用户, 在我国, 国家教委也在中小学里大力扶植中华学习机。虽然十六位机最终势必会取代八位机, 可从囿于财力及避免浪费的角度考虑, 不可能说换就换, 因此也就造成了学校里学习用机与社会上实际用机不相配套的矛盾, 而矛盾又集中表现在操作系统上的大相径庭。为了尽可能地缩小这种矛盾, 从减少两者在操作系统上的差异为出发点, 笔者在中华学习机上模拟了 PC-DOS 中的树形目录结构, 并且为中华学习机增加了相关的操作命令。

本文所提出的子目录功能, 不同于以往见诸报刊的多目录功能, 而是真正意义上的子目录功能, 它按照下列步骤予以实现。

一、建立新的操作命令名 MD、CD、DIR

本文是通过修改 Apple DOS 命令名表来建立新的操作命令的, 新的操作命令覆盖了原 DOS 中不常用

的 IN#、INT、FP 三条命令, 而为了尊重中华学习机用户的习惯, 仍旧保留了 CATALOG 命令, 修改前请先输入 PR#6 启动标准 DOS 3.3, 然后键入 CALL-151 进入监控状态(下同)。

* A8E4:44 49 D2 (DIR)

* A8EF:43 C4 4D C4 (CD MD)

* A8F3(A8F4. A908M)

然后修改各命令的有效键语表:

* A931:60 70

* A935:60 70 61 70

最后修改各命令的入口地址:

* 9D46:B2 B6

* 9D4C:EE B6

二、实际建立子目录命令 MD

子目录是作为文件的形式存在于磁盘上的, 这个文件的内容就是子目录中存放的各个文件名及其特征。为了节省中华学习机磁盘的有限空间, 本文则通过在磁盘上设置一个空文件来达到建立子目录的目的, 所谓空文件, 是指这个文件只占用一个扇区, 在 Apple DOS 中, 这个磁区是应该用来存放文件的磁道/扇区表(T/S 表)的, 本文取消了这个 T/S 区表, 而直接用来存放子目录的内容, 这个扇区在磁盘上的位置会由子目录文件索引指出。为了避免用户错误地使用这个空文件或重复建立同名子目录, 本文定义这个文件的类型为新型, 且建立后立即设置为只读。

* B6EF:A9 20 20 D5 A3 20 EA A2 20 71 A2 4C EA A2

三、实现更改子目录的命令 CD

在 PC-DOS 中,当前目录及上级父目录的位址是以专用文件项的方式记录的。本文没有采用这种方法,原因是为了节省子目录那有限的容量。

本文采取的方法是在内存中开辟了一个目录堆栈区,共有 30 个字节,可以建立 15 级子目录,这对一般的用户来说已经宽绰有余了。

* A57A:AD 75 AA C9 DD F0 14 A9 20 20 D5 A3
AE 9C B3 BD C6 B4 BC C7 B4 20 98 A5 4C EA
A2 4C D7 B6 EE 2E A2 EE 2E A2 AE 2E A2 9D
DF BC 48 98 9D E0 BC 68 8D 2F A2 8C 30 A2 60

* B6D7:AE 2E A2 F0 12 CE 2E A2 CE 2E A2 AE
2E A2 BD DF BC BC E0 BC 4C AA A5 60

* A22F:11 0F 设置当前目录为根目录

* BCDF:11 0F 目录栈初始化

* A22E:0 栈指针初始化

四、实现列出子目录命令 DIR

本文提出的列子目录功能是建立在原 Apple DOS 中的 CATALOG 命令基础之上的。在中华学习机中,DOS 对文件的管理是通过磁盘上的 VTOC 表来实施的,换言之,在每次管理文件时 DOS 都先要把 VTOC 表读入内存然后再加以分析使用,并且在 VTOC 表中指出了当前所用目录所在磁道/扇区。所以,我们只需在 VTOC 表被读入内存后,着手对其修改,即可达到改变目录路径的目的,而需要改变的目录路径可通过 CD 命令得到。

* B6B3:AD 75 AA C9 A0 D0 08 A9 DB 8D 75 AA
4C 6E A5 AE 2E A2 8E B1 A5 20 7A A5 20 6E
A5 AE B1 A5 8E 2E A2 4C E5 B6

* B00E:4C 69 BA

* BA69:C9 01 F0 0F 48 A9 11 8D BC B3 A9 0F 8D
BD B3 68 4C 52 B0 20 52 B0 AD 5F AA 29 DF F0
0D AD 2F A2 8D BC B3 AD 30 A2 8D BD B3 60
4C 8D B7

* B78D:8D 2E A2 4C E2 B6

修改供列目录使用的显示字符:

* B3B0:BA A0 D9 D2 CF D4 C3 C5 D2

使列目录时显示所列目录的级数:

* ADBC:2E A2 08 4A 28 85 44 20 42 AE

修改被显示时的子目录文件类型:

* B3AD:95

最后用 CTRL-C 退出监控状态,然后 INIT HELLO 格式化一张具有子目录功能的新盘。

五、各命令的使用格式及说明

共同说明:所建立的三条命令都可使用 V(卷号)、S(槽号)、D(驱动器号)三个参数。

1. MD 建立子目录命令

格式:MD 子目录名

作用:在当前目录中建立一个所给子目录名的子目录。

说明:a. 所给子目录名不能与当前目录中的其他

文件名相同,但是只要这些文件被定义在不同的目录里,他们仍可以有相同的名字。

b. 当所给子目录名与当前目录中的其他子目录名相同时,DOS 不会建立新的子目录,但同名子目录被定义在不同的目录里时,仍会建立新的子目录。

c. 子目录名不能省略,而且最多只能用三十个字符表示,即与文件名的定义规则相同,但由于本文采用“J”字符作为父目录的标识符(见下文),因此子目录名的第一个字符不能为“J”。

2. CD 更改子目录命令

格式:1: CD 子目录名

作用:更改当前目录为所给子目录名的下级子目录

格式:2: CD]

作用:返回当前目录的上级父目录

说明:a. 所给子目录名必须是当前目录中已被建立的子目录,即只能逐级更改,而不能越级更改。

b. “]”字符可用 SHIFT-M 键得到,并且“]”字符后所携带的其他字符将被忽略,但 V.S.D 三个参数仍被保留。

c. 若当前目录为根目录,则 CD] 命令使用后仍在当前根目录中。

3. DIR 列出子目录命令

格式 1: DIR

作用:显示当前目录中的所有文件及子目录

格式 2: DIR 子目录名

作用:显示所给子目录名的下级子目录中的所有文件及子目录。

格式 3: DIR]

作用:显示当前目录的上级父目录中的所有文件及子目录。

说明:a. 不管是显示下级子目录还是显示上级父目录,显示后都将返回当前目录,即不会改变当前目录的路径。

b. 对子目录名与“]”字符的说明同 CD 命令。

c. 若当前目录为根目录,则 DIR] 命令显示的仍将是当前根目录中的内容。

d. 不管使用的是何种格式的 DIR 命令(下同),使用后都会在第一行出现 DIRECTORY:×××;其中×××为三个数字,表示当前显示的是第几级的子目录,根目录用 0 级表示。

e. 所显示的子目录文件类型为空,且向前移动一个字符的位置,以示与文件的区别。

六、与 PC-DOS 的差异

本文虽然是模拟 PC-DOS 的思路设计,但为了不至于对中华学习机的用户产生误导作用,有必要对其存在的差异加以指出。

1. 在 PC-DOS 中,建立、更改、列出子目录时都可以进行越级操作,而不象本文所述的只能进行逐级操作。

2. 在 PC-DOS 中,父目录用“..”两个小数点表

示,而不象本文中的用“]”字符表示。

3. 在 PC-DOS 中,不管当前处于哪级目录,都可以直接返回根目录(根目录用斜杆“\”表示),为了弥补这一缺憾,本文利用 CATALOG 命令来完成直接返回根目录的工作。不管你在哪级目录中使用了 CATALOG 命令,使用后系统都将设置当前使用的目录为根目录,而 CATALOG 命令的其他功能照旧。

4. 在 PC-DOS 中,有专门的删除子目录命令 RD,且被删除的子目录内容必须为空。而本文没有建立专门的删除子目录命令,在删除子目录时与删除文件一样都使用 DELETE 命令。值得注意的是,删除了子目录以后,子目录中的所有文件都将被删除,因此,希望读者慎用删除命令。

5. 在本文中 CD 命令必须携带子目录名或父目录符号,在 PC-DOS 中,不携带参量的 CD 命令用以显示当前的目录路径。

七、使用本系统需要注意的事项及技巧

1. 本系统由于采用目录堆栈区来记录当前的目

录路径,因此,当前所使用的目录路径只取决于内存中的记录,而与所用磁盘无关,所以,你也可以换用其他未经本系统格式化过的磁盘,但在换用前,必须利用 CATALOG 命令回到根目录,否则会出现意料不到的后果。

2. 由于目录堆栈区不受保护,所以当你开辟的子目录级数大于 15 级时,也会出现意料不到的后果,但子目录的个数可以远大于 15 个。

3. 对于已经利用本系统建立过子目录的任意磁盘,都不能对其进行 FIX FILES SIZES(调整文件占用的磁盘扇区数目)操作。

4. 未携带本系统,但在本系统子目录下记录过文件的磁盘在使用中将不能进入这些文件,可以起到加密的作用。

5. 本文在 CEC-I 及 Apple 机上通过,除了被撤消的 IN#,INT,FP 三条命令外,其余的 DOS 命令不论是在根目录还是在子目录都能被正确执行,且在子目录下格式化新盘有意料不到的加密效果,不妨一试。

任意数的高精度快速乘法程序

徐 永 福建师范大学物理系(350007)

微型计算机所能处理的实数的有效位数及数值范围都是有限的,当用它进行乘法运算时,若结果的有效位数超过某限度,则按科学记数法输出,多余的位数将被丢失,有时不能满足精度要求;若结果超过某一上、下限范围,则产生溢出错误或输出为 0。例如 Apple 机的有效位数为 9 位,所能表示的数值范围的上限为 10^{38} ,下限为 3×10^{-39} ,当乘积(或被乘数、乘数本身)超过上述范围时,就不可能得到精确的结果或出错。目前有关刊物上曾发表一些高精度的乘法程序。但这些程序或者在算法上模拟人工笔算,对输入的多位数的每一位进行逐位计算,运算速度较慢,或者只能进行正整数乘法,实用范围有限。另外,到目前为止发表的所有高精度乘法程序中,只能对按常规记数法表示的数进行处理和运算,不能处理按科学记数法表示的数。这样,当所处理的数的数量级很大或很小时,不论是对数据的输入还是结果的读出都极为不便。

本文给出的任意数的高精度快速乘法的通用程序,无论所处理的数据是按常规记数法逐位输入,还是按科学记数法输入,都能得到正确的输出。当结果的总位数超过 9 位时,按科学记数法输出,否则按常规记数法输出,不仅符合人们的记数习惯,而且,当输出的位数较多时,能一目了然地看出结果的数量级,使用极为方便,其运算速度也快于至今为止已发表的用 BASIC 语言编写的高精度乘法程序。

本程序用字符串型数组 A\$(2)接收被乘数和乘数,用二维整型数组 A%(2,W)将被乘数和乘数按 4 位一组分段存储和运算,用实型数组 M(L)存储运算的中间结果,充分利用了所能记录的最多有效位数(9 位),同时,用该数组的低四位记录乘积的最终结果。这样既充分利用各种类型变量的最多有效位数,又能提高运算速度,减少占用的内存,扩大处理范围。

该程序运算 100 位 \times 100 位的数只需 24 秒。现将该程序的主要部分简要说明如下:

40~140 行是对输入的数据进行整理,去除被乘数和乘数的负号、科学记数法中的指数部份(包括符号“E”)、小数点以及数据前后多余的“0”,并用 U 和 C 分别记录整形后结果的符号和指数值。如输入“-0.03548E-12”和“4352600E8”两数,经该段程序处理后分别为“3548”和“43526”,且 $U=-1, C=-7$ 。

170~200 行为按四位一组将被乘数和乘数分别赋予整形数组 A%(1,W)和 A%(2,W)。

220~270 行计算乘积。该处使用双重循环,并采用内循环 10 次后才对部分积进行一次规整修正的算法。这样既能最有效地利用实型数组 M(L)的最多有效位数(9 位),使部分积的位数不致丢失,又能减少规整次数,提高运算速度,是本程序的特色之一。

280~300 行为去除运算结果前后多余的“0”,并计算最终结果的有效位数及指数值。

330~430 行为计算结果的输出。当位数小于等于 9 时,按常规输出,否则按科学记数法输出。

```

10 HOME; CLEAR; INPUT "被乘数="; A$(1); A(1) =
  LEN(A$(1)); INPUT "乘数="; A$(2); A(2) = LEN
  (A$(2)); C=0; U=1
20 PRINT "("; A$(1); ")" * "("; A$(2); ")" = ";
30 IF A$(1) = "0" OR A$(2) = "0" THEN PRINT "0";
  END
40 FOR J=1 TO 2: IF LEFT$(A$(J),1) = "-" THEN U
  = -1 * U; A$(J) = MID$(A$(J),2); A(J) = A(J) -
  1
50 FOR I=1 TO A(J)
60 IF MID$(A$(J),I,1) < ">"E" THEN NEXT I; GOTO
  80
70 C=C+VAL(RIGHT$(A$(J),A(J)-I)); A(J)=I-
  1; A$(J)=LEFT$(A$(J),A(J))
80 FOR I=1 TO A(J)
90 IF MID$(A$(J),I,1) < ">". THEN NEXT; GOTO
  110
100 C=C-A(J)+I; A(J)=A(J)-1; A$(J)=LEFT$(A
  $(J),I-1)+MID$(A$(J),I+1)
110 FOR I=1 TO A(J); IF MID$(A$(J),I,1) = "0"
  THEN NEXT I
120 A(J)=A(J)-I+1; A$(J)=RIGHT$(A$(J),A(J))
130 FOR I=A(J) TO 1 STEP -1; IF MID$(A$(J),I,1)
  = "0" THEN NEXT I
140 A$(J)=LEFT$(A$(J),I); C=C+A(J)-I; A(J)=I
150 L(J)=INT((A(J)+3)/4); NEXT J
160 L=L(1)+L(2); W=(L+ABS(L(1)-L(2)))/2; N=
  10; E=1E4; DIM M(L), A%(2,W)
170 FOR I=1 TO 2: A$(I)=RIGHT$("000"+A$(I),L
  (I)*4)
180 FOR J=1 TO L(I)
190 A%(I,J)=VAL(MID$(A$(I),4*(L(I)-J)+1,4))
200 NEXT J; NEXT I
210 FOR K=1 TO L; M(K)=0; NEXT K
220 FOR I=1 TO L(1)
230 FOR J=1 TO L(2); K=I+J-1

```

```

240 M(K)=A%(1,I)*A%(2,J)+M(K); NEXT J
250 IF I<N THEN NEXT I
260 FOR M=N-9 TO K; M0=INT(M(M)/E); M(M)=M
  (M)-M0*E; M(M+1)=M(M+1)+M0; NEXT M
270 IF K<L-1 THEN N=N+10; NEXT I
280 IF M(L)=0 THEN L=L-1
290 FOR K=1 TO L; IF M(K)=0 THEN NEXT K
300 FOR I=1 TO 3
310 IF M(K)/10=INT(M(K)/10) THEN M(K)=M(K)/
  10; NEXT I
320 L0=K; A0=I-1; N0=LEN(STR$(M(L)))+4*(L
  -L0)-A0; C=C+4*(L0-1)+A0
330 IF U=-1 THEN PRINT "-";
340 IF N0>9 OR C<=-9 OR C+N0>9 THEN 380
350 IF L=L0 THEN PRINT M(L)*10^C; END
360 IF L=L0+1 THEN PRINT (M(L)*10^(4-A0)+M
  (L0))*10^C; END
370 PRINT (M(2+L0)*10^(8-A0)+M(1+L0)*10^
  (4-A0)+M(L0))*10^C; END
380 PRINT LEFT$(STR$(M(L)),1); ". "; MID$(STR
  $(M(L)),2);
390 IF L=L0 THEN 430
400 IF L=L0+1 THEN 420
410 FOR I=L-1 TO L0+1 STEP -1; PRINT RIGHT$("
  000"+STR$(M(I)),4);; NEXT I
420 PRINT RIGHT$("000"+STR$(M(L0)),4-A0);
430 PRINT "E"; C+N0-1; END
RUN
被乘数=-1.7253125E-119
乘数=3.648192E124
(-1.7253125E-119)*(3.648192E124)=-629427.126
RUN
被乘数=1.23456789012345678901234567890E29
乘数=987654321098765432109876543210
(1.23456789012345678901234567890E29)*(987654321098
765432109876543210)=1.2193263113702179522618503273
36229233322374638011112635269E59

```

DOS 操作系统分析讲座

第三讲 屏幕显示系统(续二)

崔来堂

今举例说明调用显示驱动程序 INT 10H 的方法:

[例 1] 显示一个缓冲区中的 ASCII 字符序列。

; 显示字符序列的过程 DISP_ASC

; 入口参数: SI=缓冲区字节偏移, CX=缓冲区中字符数

DISP_ASC PROC NEAR

NEXT:

LODSB

MOV AH,OEH

MOV BX,0

; 取待显字符到 AL

; 子功能号

; 当前页面 0


```

        INT 10H          ;调用
        LOOP NEXT        ;循环,直到显示完毕
        RET
DISP_ASC ENDP

```

[例 2]显示一个缓冲区中的 16 进制数码。

;显示 16 进制数码程序 DISP-HEX(设有 10 个数码,占 5 个字节)

```

CODE    SEGMENT
        ORG 100H
        ASSUME CS:CODE,DS:CODE
MAIN     PROC FAR          ;主程序
        MOV SI,OFFSET BUFF ;SI=缓冲区偏移
        MOV CX,5          ;CX=缓冲区字节长度
START:   LODSB             ;取一字节 16 进制数(两位)到 AL
        PUSH CX            ;位)到 AL
        CALL SUB1          ;转换、显示
        POP CX
        LOOP START        ;循环,直至显示完毕
        MOV AH,4CH
        INT 21H           ;返回 DOS
MAIN     ENDP
SUB1     PROC NEAR        ;一字节中两位 16 进制数分解子程序
        PUSH AX
        MOV CL,4
        SHR AL,CL
        CALL SUB2        ;处理高 4 位
        POP AX
        AND AL,0FH
        CALL SUB2        ;处理低 4 位
        RET
SUB1     ENDP
SUB2     PROC NEAR        ;使 16 进制数转换为 ASCII 码子程序
        ADD AL,90H
        DAA
        ADC AL,40H
        DAA
        CALL SUB3
        RET
SUB2     ENDP
SUB3     PROC NEAR        ;显示子程序
        MOV AH,0EH
        MOV BH,0
        INT 10H
        RET
SUB3     ENDP
BUFF     DB 01H,23H,45H,67H,89H ;缓冲区待显示
CODE     ENDS             ;内容举例
        END MAIN

```

运行此程序后,将把程序缓冲区 BUFF 中预先置入的 16 进制数(每字节两个),依次显示出来。

[例 3]字符的内烁、反相显示和屏幕卷动。

```

CODE SEGMENT
        ORG 100H

```

```

        ASSUME CS:CODE,DS:CODE

```

```

MAIN     PROC FAR          ;主程序
        CALL CLRSCN       ;清屏
START:   CALL SETCUR       ;设置光标
        CALL CLRARE       ;清屏幕左上部位
        CALL PROMPT       ;显示提示信息
        CALL KEYIN        ;键入字符串
        CMP COUNT,00      ;是否停止?
        JE RETN           ;是,转
        CALL DISP         ;否,显示字符串
        JMP START         ;进行下一次
RETN:    MOV AH,4CH        ;返回 DOS
        INT 21H
MAIN     ENDP
CLRSCN   PROC NEAR        ;清屏子程序
        MOV CX,0000       ;左上角坐标
        MOV DX,184FH      ;右下角坐标
        MOV BH,07         ;常规属性
        MOV AX,0600H      ;全屏幕上卷
        INT 10H
        RET
CLRSCN   ENDP
SETCUR   PROC NEAR        ;设置光标到 4 行 4 列子程序
        MOV AH,02
        MOV BH,00
        MOV DH,04
        MOV DL,04
        INT 10H
        RET
SETCUR   ENDP
CLRARE   PROC NEAR        ;清屏幕左上部位子程序
        MOV CX,0000       ;0 行 0 列
        MOV DX,0616H      ;6 行 22 列
        MOV AX,0607H      ;上卷 7 行
        MOV BH,07
        INT 10H
        RET
CLRARE   ENDP
SETCUR   PROC NEAR        ;设置光标子程序
        MOV AH,02
        MOV BH,00
        MOV DH,ROWCTR ;行值
        MOV DL,COLCTR ;列值
        INT 10H
        RET
SETCUR   ENDP
PROMPT   PROC NEAR        ;显示提示信息子程序
        MOV AH,09
        LEA DX,NAMES
        INT 21H
        RET
PROMPT   ENDP
KEYIN    PROC NEAR        ;键入字符串子程序
        LEA DX,BUFF
        MOV AH,0AH
        INT 21H

```

```

RET
KEYIN ENDP
DISP PROC NEAR ;显示键入的字符串子程序
LEA SI,STRING ;SI=字符串字节偏移
MOV COLCTR,32
REP: CALL SETCUR ;设置光标
MOV AH,09
MOV AL,[SI] ;AL=待显字符
MOV BH,00 ;0 页
MOV BL,0F0H ;设置反相、闪烁属性
MOV CX,01 ;每次显示的字符数
INT 10H
INC SI
INC COLCTR
DEC COUNT ;该串显完否?
JNZ REP ;未完转
CMP ROWCTR,20 ;是否显到 20 行?
JB REPP ;未到转
MOV AX,0601H ;到,屏幕上卷 1 行
MOV BH,07
MOV CX,0000
MOV DX,184FH
INT 10H
MOV ROWCTR,19
REPP: INC ROWCTR ;准备在下一行显示
RET
DISP ENDP
BUFF DB 20 ;键入缓冲区的容量
COUNT DB ? ;键入的实际字符数
STRING DB 20 DUP(?) ;缓冲区
NAMES DB 'String:',0DH,0AH,'$' ;提示信息
ROWCTR DB 10 ;显示的起始行值
COLCTR DB 32 ;显示的起始列值
CODE ENDS
END MAIN

```

程序运行时,首先在屏幕左上部位显示提示信息'String:',此后即可键入字符串(勿超过 32 个字符),回车后,就会在屏幕中部反相、闪烁显示出来;再键入,再显示...;直至空串时打回车,即返回 DOS。当显示的内容达到屏幕第 20 行(快到底部)时,每次再显示,屏幕将上卷一行。

[例 4]用程序小段,显示汉字字符串“中国”。

要显示汉字,需先知汉字的内码。求得汉字内码的简便方法是:在汉字操作系统下,进入 DEBUG,由 DB 伪指令对汉字字符串进行字节定义,再由 D 命令显示出来即可。操作如下:

```

C>debug
-A100
DB '中国'
(回车)
-D100 L2
-Q

```

显示出相应字节的内容是:D6 D0 B9 FA,即所求的内码串。因此,显示汉字字符串的程序段落是:

```
C>debug
```

```

-A100
MOV AH,02 ;设置光标子功能
MOV BH,0 ;0 页
MOV DX,0101 ;1 行 1 列
INT 10
MOV AH,0E ;显示字符子功能
MOV AL,D6 ;内码
INT 10
MOV AL,D0
INT 10
MOV AL,B9
INT 10
MOV AL,FA
INT 10
INT 3
(回车)
-G=100
-Q

```

五、屏幕功能的扩展

所谓屏幕功能的扩展,是指允许用户使用规定的字符串序列,设置屏幕的显示功能。与键盘的情况相似,只要系统盘的根目录中含有文件 CONFIG. SYS 和 ANSI. SYS,并将语句 DEVICE=ANSI. SYS 写入 CONFIG. SYS 中,系统启动后,就具有了屏幕扩展功能。若文件 ANSI. SYS 在子目录中,则上述语句中应包含相应路径。

这里所用的字符串,也由"ESC["打头,具体请见下列各表。

1. 光标位置的控制

表 3-7 光标位置控制字符串

控制字符串	功 能
光标定位 ESC[*;*H	光标移到参数*;*指定的位置。前*表示行号,后*表示列号,隐含值均为 1。若无参数,光标移至左上角。
光标上移 ESC[*A	光标上移*行,隐含值为 1。光标列位置不变。若光标已在顶行,则该控制串不起作用。
光标下移 ESC[*B	光标下移*行,隐含值为 1。光标列位置不变。若光标已在底行,则该控制串不起作用。
光标左移 ESC[*C	光标右移*列,隐含值为 1。若光标已在最右列,则该控制串不起作用。
光标右移 ESC[*D	光标左移*列,隐含值为 1。若光标已在最左列,则该控制串不起作用。

2. 清除屏幕

表 3-8 清屏控制字符串

控制字符串	功 能
清 屏 ESC[2J	清除屏幕上的全部内容,光标移至左上角。
清 行 ESC[K	清除自光标位置到行末的所有内容(包括光标位置)。

3. 设置显示方式

设置显示方式的控制字符串为 ESC[=*h, 其中,* 的值及其含义如下:

表 3-9 显示方式控制字符串

*=0:40*25 字符方式,黑白	*=4:320*200 图形方式,彩色
*=1:40*25 字符方式,彩色	*=5:320*200 图形方式,黑白
*=2:80*25 字符方式,黑白	*=6:640*200 图形方式,黑白
*=3:80*25 字符方式,彩色	

4. 设置屏幕属性、颜色

设置屏幕属性、颜色的控制字符串是 ESC[*;*m, 其中,前后的 * 值分别表示前景和背景;只有一个 * 时表示特殊属性。* 值及其意义如下:

表 3-10 屏幕属性控制字符串

特殊属性	前 景		背 景	
00 关闭属性, 成黑底白字	30 黑色	34 蓝色	40 黑色	44 蓝色
01 粗体高亮度	31 红色	35 洋红色	41 红色	45 洋红色
05 闪烁	32 绿色	36 天蓝色	42 绿色	46 天蓝色
07 反相显示	33 黄色	37 白色	43 黄色	47 白色
08 消隐不可见				

由上知,只要选择相应的控制字符串和恰当的参数,就能将屏幕设计得赏心悦目,并方便地进行相应的操作。实现屏幕扩展功能的方法有下述两种:

(1) 由 DOS 命令 Prompt 和参数 \$e 实现。Prompt \$e 向 DOS 发送控制代码 ESC(即 ASCII 码 1BH),故可用来定义屏幕。

[例 1]C>Prompt \$e[32;43m

表示将屏幕设置成前景为绿色,背景为黄色。

[例 2]C>Prompt \$e[1m

表示将屏幕的局部前景设置成粗体高亮度,以便突出某些显示内容。

[例 3]C>Prompt \$e[8m

表示将屏幕设置成消隐方式,这样,口令输入时,别人不可见,以便保密。

需要注意的是:使用上述方法时,可考虑将相应的 Prompt 命令写入批文件中(必要时写入自动批文件中),以提高效率;但在批文件的各个相应 Prompt 命令之后,还应有一条 Prompt \$p\$g 命令,用以恢复可以显示路径的正常提示符。

(2) 由程序实现。

[例 4]用程序小段,将光标定位在 12 行 40 列(即屏幕中央)。

C>debug

-A100

MOV AH,09 ;显示字符串子功能号

MOV DX,200 ;控制字符串的字节偏移

INT 21

INT 3

-A200

DB 1B,'[12;40H\$';控制字符串

-G=100

注:该例执行时,由于 INT3 要显示各寄存器的状态,光标将一闪而过,应仔细观察。

[例 5]用程序小段,将屏幕设置成绿底红字,并显示 10 个字符 A。

C>debug

-A100

MOV AH,09

MOV DX,200

INT 21

MOV DX,220

INT 21

MOV DX,240

INT 21

MOV AH,0A ;显示字符串子功能号

MOV AL,41 ;字符'A'的 ASCII 码

MOV BX,0 ;0 页

MOV CX,0A ;显示字符个数

INT 10

INT 3

-A200

DB 1B,'[=3h\$';设置显示方式

-A220

DB 1B,'[2J\$';清屏

-A240

DB 1B,'[31;42m\$';设置前景和背景颜色

-G=100

(下转第 41 页)

DP-851 单片机系统

胥筱汀 北京西城区青少年科技馆(100035)

本期开始介绍《电子老鼠》各组成部分电路原理。

计算机是《电子老鼠》的重要组成部分,为了使计算机能够一机多用,既能自开发程序驾驶《电子老鼠》闯迷宫,又能做其他应用,必须选择一个优良的线路。北京市单片机应用技术协会推出的《DP-851 单片机普及板》就是这种优良机型(目前正在修改提高)。选择这个机型是因为它有几个突出的特点。

配套性

北京市单片机应用技术协会不但推出《DP-851 单片机普及板》机型,还与中国计算机学会联合举办了《单片机普及函授班》,编写了教学实验大纲,出版了教材,实验指导书,组织了一支辅导咨询队伍,为学员提供一系列服务。在普及函授班的基础上又开办了《工业应用设计班》,形成了一条龙的系列培训。

配套性的另一个方面,是与单片机发展技术手段配套,既能完成用键盘机器码输入进行编程、调试等最基础的试验,也能在 PC 机的支持下,完成使用汇编语言编程、调试的实验。它还将开发程序、固化程序和实际应用三方面的功能融为一体,实现一机多用,成龙配套,给学习和应用提供极大方便。

适应性

该机适应多个层次的学习或应用对象使用,既能适应高中生、职业中学的学生和本科生学习和做基础实验使用,也能适应研究生、科技人员做课题实验应用,还能直接将它应用于某个工作现场。

它的适应性还表现在采用模块化结构,总线和 I/O 口线和功能较强的监控系统全部对用户开放。在开发程序和调试时,有 PC 机支持能较快的工作,无 PC 机支持,用薄膜键盘也能完成;在应用上,将实验板挂在主机上,可构成多种基础试验的设备;将《电子老鼠》挂在主机板上,就可以用程序驱动《电子老鼠》闯迷宫,将工业测控设备挂在主机板上,它也能圆满完成任务。

开放性

《DP-851 单片机普及板》的全部资料对社会开放,电路原理图、子程序入口地址和监控程序全部公开,以使用户调用,借鉴、研究。

本机的全部资源对用户开放,在开发状态时,监控程序占用了 8155 的一些口线,一旦转为应用状态,普及板上的全部资源都归用户使用。

从开发状态到应用状态巧妙地转换,并将监控程序中的一些功能模块提供给用户使用,是该机的一大特色。

工作在第一线的广大教师和科技人员,有丰富的实践经验,无疑也能用自己的成果来丰富这套教学、应

用系统,使其功能更强,适应面更广。

DP-851 的工作原理

DP-851 指的是一个单片机系统。如果把它和微型计算机相比,DP-851 就是一个以 8031 单片机芯片作中央处理器的微型计算机。它体积虽小,五脏俱全。它包括:单片机芯片、程序存储器、数据存储器、键盘、显示、串行接口控制电路,其框图如图 1。

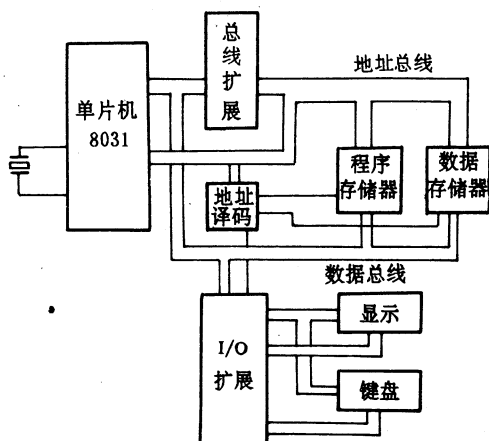


图 1 DP-851 主板框图

8031 是一个单片机芯片,单片机是单片微型计算机的简称。在一个芯片上,集成了中央处理器、随机存储器、只读存储器(8031 没有此项)、I/O 接口、可编程定时/计数器等。这样一个芯片几乎包括了微型计算机的主要部件,故称之为单片机。提到“单片机”时,有时是指单片机芯片,如 8031、8051;有时是指单片机系统,如 DP-851,简称“单片机”是为了简洁方便。

单片机在软件和硬件方面,都有其独到之处,它不追求规模大,但力争小而全,尽量满足各方面的需要。以单片机组成的系统,体积小、功耗低、价格便宜,因此得到日益广泛的应用。单片机系统涉及到了微型计算机的一系列技术,因而它也是学习微型机的典型范例。

8031 是 Intel 公司 MCS-51 单片机系列中的一个,图 2 是 8031 的引脚和逻辑符号图。其引脚功能说明如下:

端口 0(P0.0~P0.7):8 位双向 I/O 口。当访问外部存储器时,它分时输出地址的低八位地址、输出或输入数据。

端口 1(P1.0~P1.7):8 位双向 I/O 口。

端口 2(P2.0~P2.7):8 位双向 I/O 口。在访问外部存储器时,它输出高八位地址。

1	P ₁₆	V _{CC}	40
2	P ₁₁	A _D /P ₀₀	39
3	P ₁₂	AD ₁ /P ₀₁	38
4	P ₁₃	AD ₂ /P ₀₂	37
5	P ₁₄	AD ₃ /P ₀₃	36
6	P ₁₅	AD ₄ /P ₀₄	35
7	P ₁₆	AD ₅ /P ₀₅	34
8	P ₁₇	AD ₆ /P ₀₆	33
9	RST/VPD	AD ₇ /P ₀₇	32
10	P ₃₀ /RXD-EA/V _{PP}		31
11	P ₃₁ /TXDALE/PROG		30
12	P ₃₂ /INT ₀ -PSEN		29
13	P ₃₃ /INT ₁ -A ₁₅ /P ₂₇		28
14	P ₃₄ /T ₀	A ₁₄ /P ₂₈	27
15	P ₃₅ /T ₁	A ₁₃ /P ₂₅	26
16	P ₃₆ /WR	A ₁₂ /P ₂₄	25
17	P ₃₇ /RD	A ₁₁ /P ₂₃	24
18	XTAL ₂	A ₁₀ /P ₂₂	23
19	XTAL ₁	A ₉ /P ₂₁	22
20	V _{SS}	A ₈ /P ₂₀	21

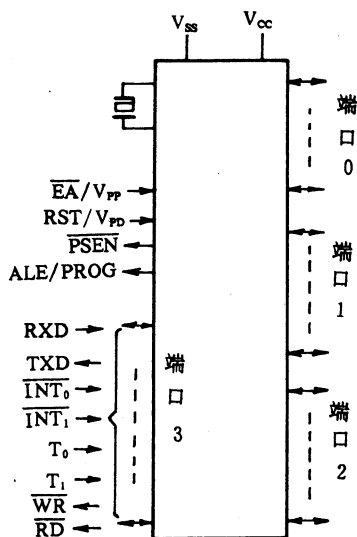


图2 8031的引脚和逻辑符号图

端口3(P_{3.0}~P_{3.7}):8位双向I/O口。这八个引脚还有第二功能:

- P_{3.0}/RXD:串行口输入。
- P_{3.1}/TXD:串行口输出。
- P_{3.2}/INT₀:外部中断0。
- P_{3.3}/INT₁:外部中断1。
- P_{3.4}/T₀:定时器/计数器0。
- P_{3.5}/T₁:定时器/计数器1。
- P_{3.6}/WR:对外部存储器的写信号。
- P_{3.7}/RD:对外部存储器的读信号。

RST/VPD:单片机复位信号输入端,高电平有效。此脚上高电平连续保持两个机器周期以上,将使单片机复位。在VCC掉电期间,可在此脚接上备用电源,以保持内部RAM中的数据。

ALE/PROG:低八位地址锁存信号。当访问外部存储器时,ALE输出高电平,将低八位地址锁存在373

芯片中。即使不访问外部存储器,ALE仍以固定频率输出信号。其频率为时钟振荡频率的六分之一。对于有内部ROM的单片机,此脚还用输入编程脉冲。

PSEN:外部程序存储器选通信号。在从外部程序存储器中取指令(或常数)期间,每个机器周期内PSEN两次有效(低电平)。但访问外部数据存储器时,PSEN端不会有效。

EA/VPP:访问外部程序存储器的控制信号。低电平有效。当EA保持低电平时,8031只访问外部程序存储器,不管是否有内部程序存储器。EA为高电平时,8031访问内部程序存储器,当地址超过范围时,自动转向访问外部程序存储器。对于有片内EPROM的单片机,编程时此脚施加编程电源(VPP)。

V_{CC}:接5V电源正极、V_{SS}:接5V电源负极。

8031的内部结构

8031是一个8位(运算是8位)单片机。片内有256字节RAM,没有ROM。图3是8031的内部结构框图。

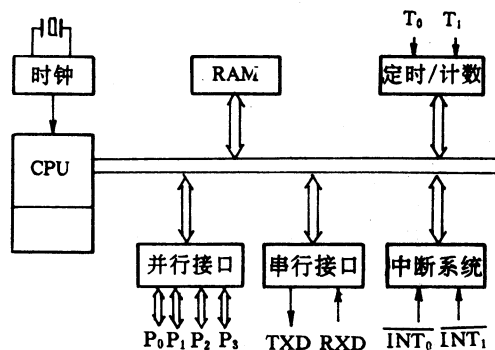


图3 8031内部结构图

CPU:中央处理单元。

中央处理单元由控制器和运算器组成。从功能上讲,运算器包括:算术逻辑部件ALU、累加器A、寄存器B、程序计数器PC、程序状态字寄存器PSW、堆栈指针SP、数据指针寄存器DPTR和逻辑运算部件。控制器包括:指令寄存器、指令译码器、控制逻辑阵列。

ALU在控制逻辑阵列发出的控制信号的控制下,实现一些算术、逻辑操作。如加法、减法、8位无符号数的乘法和除法、逻辑“与”、“或”、“异或”、移位、半字节交换等操作。ALU还有一些直接按位操作功能,如置位、清零、求补、条件判转、逻辑“与”、“或”等。在需要按位运算时,位操作指令提供了把逻辑等式直接变换成软件指令的简单明了的方法。

累加器A是一个最常用的8位专用寄存器。指令系统中采用A作为累加器的助记符。指令中的操作数之一,通常都来自累加器。算术运算指令的运算结果,都存放在累加器A或AB寄存器对中。

寄存器B是一个8位寄存器。用于乘除法运算。乘法运算时,B是操作数,乘积存于AB中。除法运算时,A是被除数、B是除数,其商存于A,余数存于B。

程序状态字寄存器 PSW 是一个 8 位寄存器,用于存放程序运行的状态信息。它由硬件自动设置。条件转移指令将根据它的有关位的状态,实现程序转移。PSW 寄存器的编址为 D0H,按位寻址时,位地址为 D₀~D₇。各位的含义如下:

D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀
CY	AC	FO	RS1	RS0	OV	—	P

CY:进位标志,是累加器 A 的溢出位。当操作结果在最高有进位(加法)或借位(减法)时 CY 被置 1,否则被清零。在做位操作时,CY 起着累加器的作用,操作位之一必须在 CY 中。

AC:辅助进位标志,是低 4 位向高 4 位进位或借位的标志。当 D₃ 向 D₄ 位进位时,AC 被置 1,否则被清零,BCD 码调整时,也用到 AC。

FO:由用户定义的标志位,可以通过软件对其进行置位/复位或测试,以控制程序转移。

RS1、RS0:寄存器组选择位,可由软件设置。

RS1 RS0 工作寄存器组号

0	0	0 组
0	1	1 组
1	0	2 组
1	1	3 组

OV:溢出标志位。用作有符号算术运算的溢出标志,由硬件设置。如加减运算结果超出累加器 A 所能表示的符号数的范围(-128~+127)时,OV 被置 1。

在无符号乘法运算中,积超过 255,在除法运算中除数为 0,均置 OV 为 1。

P:奇偶标志位。用于表明累加器 A 的 8 位中,1 的个数的奇偶性(个数为奇数时 P=1),由硬件自动设置。

DPTR:数据指针,这是一个 16 位的寄存器,可分

为 DPH 和 DPL 高低两个字节。在 CPU 访问外部数据存储时,用 DPTR 作为地址指针。

并行 I/O 口

8031 的 32 根 I/O 线分为四个双向并行口 P0~P3。每根 I/O 线都能独立地用作输入或输出。8031 内部 I/O 线配有锁存器,输出驱动器和缓冲器(三态门)。

P0 口受内部控制信号控制,可以分时切换地址/数据功能,作总线用,在没有外部存储器的情况下,作双向 I/O 口使用。

P1 口并行输入、输出口,可以按位操作。

P2 口有外部存储器时,作地址高 8 位输出口用;在没有外部存储器时,可作双向 I/O 口使用。

P3 口除了用作一般 I/O 口外,每一根线都可执行与口功能无关的第二种功能(前面已有说明)。

串行 I/O 口

串行口是一个全双工通信接口,即能同时进行发送和接收。可以以异步通信方式(UART)和外部设备通信,也可作同步移位寄存器使用。

定时器/计数器

8031 内的可编程定时器/计数器,由控制位 C/T 来选择其功能。作为定时器时,每个机器周期加 1(计数频率固定为时钟频率的 1/12)。作为计数器时,在外部事件脉冲的负沿时加 1(最高计数频率为时钟频率的 1/24)。

时钟

8031 内部有一个用于构成振荡器的高增益反相放大器,与外接晶体形成谐振回路,产生时钟信号。晶体可以是 6MHz 的,也可以是 12MHz 的。片内时钟发生器将振荡信号二分频,为芯片提供 2 相时钟信号。

一个机器周期由 6 个时钟状态组成,每个时钟状态又是由 2 个振荡脉冲组成,因此一个机器周期包括 12 个振荡脉冲。

正普数据库 ——数据库专业制作公司的奉献

1. 正普中国邮政编码数据库(综合) V94.4 500 元
全国 18 万家企事业单位名称,地址,邮政编码。
2. 正普中国高校数据库 V94.4 300 元
全国 1100 家高等院校名称,地址,邮政编码,学科方向,是否博士硕士学位。
3. 正普中国报刊数据库 V94.4 600 元
全国 2000 家报纸,7000 家期刊杂志名称,地址,邮政编码,电话。
4. 正普中国出版社数据库 V94.4 300 元
全国 1000 家图书音像出版社名称,地址,邮政编码,电话。
5. 正普中国图书音像发行机构数据库 V94.4 600 元
全国 6000 家发行单位,4000 家新华书店名称,地址,邮政编码,电话。
6. 正普中国仪器仪表企业数据库 V94.4 500 元。

全国 1000 家仪器仪表企业的名称,地址,邮政编码,电话,主要产品。

7. 正普中国信息服务机构数据库 V94.4 300 元
全国 2000 家信息咨询,公关,广告及中介服务机构的名称,地址,邮政编码,电话,服务范围。

正普数据库有准确、有效、全面、实用的特点,用户可以根据任意条件对正普数据库进行查询,可批量打印信封,符合标准信封规范效果精美,标签打印功能强大,帮助您树立良好形象。

正普公司还有“正普通用发行管理系统”,“正普信封打印专用系统”,“正普期刊发行管理系统”,“正普超级加密工具”,“正普公共关系信息管理系统”,“正普个人事务管理系统”,“正普文档管理系统”等软件可供选购,欢迎垂询。

北京市正普电子技术公司

通信地址:100081 北京海淀区白石桥路 30 号正普公司

电话:8356536-230,8356538-230

寻呼:2025566-19986/19987

开户行:北京工商银行海淀分理处双信

户名:北京市正普电子技术公司 账号:071280-41

用 8031 单片机组装细胞分类计数器

包起龙 北京铁路总医院(100038)

细胞的分类计数数值在医院临床检验、疾病的诊断中起到重要作用。分类就是把血液中的淋巴细胞,早幼粒,中幼粒,晚幼粒,杆状,单核,多核,嗜酸,嗜硷,原始等细胞数量分别统计记录下来,和正常值进行比较作为诊断评价疾病的依据。

目前医院临床化验室在做分类统计时,还采用机械式计数器。这类计数器在记录计数值时,用手指按动按键,该计数器按键按动抬度高,手指容易疲劳,显示数字字形小,读取易错,使用不便,同时传动咬合齿轮容易磨损,故障率较高。

我们用 8031 单片机和有关器件设计组装了一台白细胞分类计数器。在我院临床检验使用较为满意。它具有以下功能:

1. 能配合临床检验要求,记录白细胞中各项分类(淋巴细胞,早,中,晚幼粒,杆状,单核,多核,原始,嗜酸,嗜硷等细胞)的数量。
2. 在记数时,每按一下相应按键会发声一下,以保证不会产生误操作。
3. 根据临床检验要求分类记录总数满一百时,仪器自动报警,停止计数,等候列数。
4. 仪器复位前可反复列出所记录的数据。
5. 在分类输入时,若按错键有修改功能,以保证正确输入。

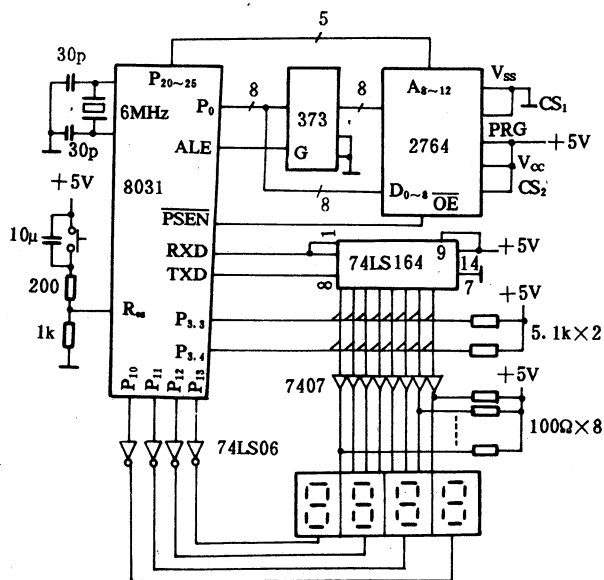


图 1

硬件说明

该仪器核心采用 MCS51 系列 8031 单片机最小系统。它由 8031、74LS373、EPROM2764 组成。对 MCS51 的 8031 来说,片内不具有 ROM 或 EPROM,只有扩展了外接程序存储器后才具有计算机的最基本部分。74LS373 是带三态门的八位 D 锁存器,它利用控制信号 ALE 的下降沿将 P0 口的输出地址锁存。数据总线八位也由 P0 口提供,该口为三态双向口,单片机与外部信息的交换全部由它传送。EPROM2764 用来存放本机的监控程序。由于随机存储的数量不大,8031 片内的 RAM 已足够,不用外扩展 RAM。8031 串行口没有别的用途,为节省硬件配置,人机对话用串行口输入

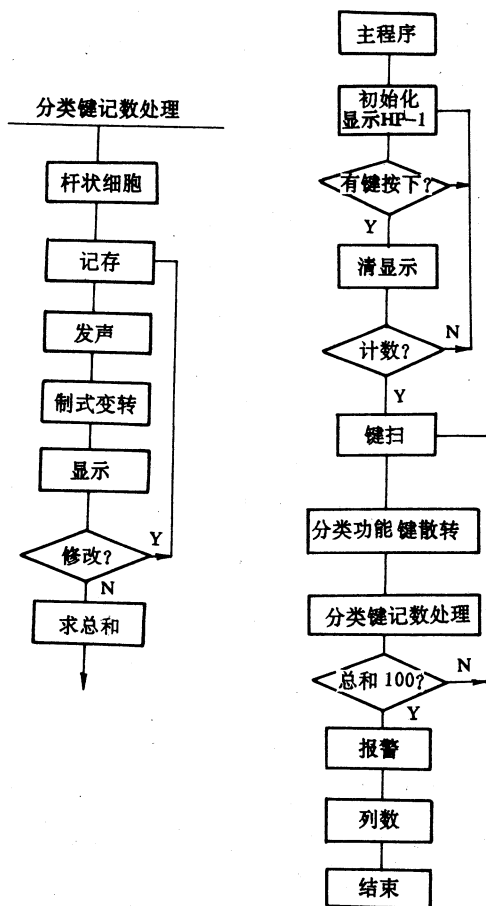


图 2

一则教材中常见错误的更正

朱立奇 安徽徽州师范专科学校(245021)

JMP @A+DPTR 指令是 MCS-51 系列单片计算机的一条非常有用的间接转移指令,用它可以很方便地实现程序的多分支转移结构(简称散转)。但在散转点超过 256 时,需正确地计算偏移量和修改数据指针。在教学中发现有数本教材均存在不同程度的错误(见注),现将此例分析如下:

1. 因散转点超过 256 点,需用两个字节来表示待散转点的数值,设该数值存于工作寄存器(R3R2)中;

2. 因散转表均为用三字节指令(LJMP ADDR16)构成,故散转点的数值需乘以 3 以后才能得到相对于散转表首地址的偏移量;

3. 因 MCS-51 系列单片计算机的最大程序空间为 64K 字节,(R3R2)乘 3 的积仍可用两个字节来表示;

4. 由乘式:

$$\begin{array}{r} \text{R3} \quad \text{R2} \\ \times \quad 3 \\ \hline 3\text{R2H} \quad 3\text{R2L} \\ 0 \quad 3\text{R3L} \end{array}$$

可知,只须将 3R3L 和 3R2H 分别加到数据指针 DPTR 的高字节上,3R2L 放在累加器 A 中,再用 JMP @A+DPTR 指令就可实现散转点大于 256 点的跳转了。

程序清单如下:

```
MOV DPTR, #TJB      ;设散转表基地址
MOV A, R3             ;(R3) * 3
MOV B, #3
MUL AB               ;(A) * (B) ← (BA)
ADD A, DPH            ;(DPH) ← 3R3L + (DPH)
MOV DPH, A
MOV A, R2             ;(R2) * 3
MOV B, #3
MUL AB
XCH A, B
ADD A, DPH            ;(DPH) ← 3R2H + (DPH)
MOV DPH, A
XCH A, B              ;(A) ← 3R2L
JMP @A+DPTR           ;间接转移
TJB: LJMP PRG0        ;散转表首址
      LJMP PRG1
      .
      .
      .
      LJMP PRGn
```

注:1. 徐淑华等编《单片微型机原理及应用》哈工大出版社 91 年 6 月第一版 第 152 页

2. 孙涵芳 徐爱卿编著《MCS-51/95 系列单片机原理及应用》北京航空航天大学出版社 88 年 2 月第一版 第 196 页

3. 丁志刚 李刚民编《单片微型计算机原理与应用》电子工业出版社 90 年 5 月第一版 第 202 页

(上接第 28 页)

输出方式,外接移位寄存器 74LS164 构成键盘和显示部分。显示方式采用动态显示方式,用四位 LED 显示器,因记数不超过三位,设置四位已满足要求。四个 LED 的位选信号由 8031 的 P1 口的低四位,经驱动器 7407 提供。段控制信号由串行口经 74LS164 提供。采用动态方式节省硬件,降低成本,缩小体积(见图 1)。

软件说明

软件的任务是编制一个监控程序,由它在各个时刻发出不同指令,指挥,协调各个部分工作。整个监控程序由主程序模块、键扫子程序模块、显示子程序模块、计数子程序模块、十六进制转换成十进制子程序模块、修改子程序模块、发声报警子程序模块、列数字子程序模块等组成,流程图见图 2。

因程序太长,篇幅有限,如需者可与作者联系。

北京西城电子仪表厂产品介绍

• DP851

“DP851 单片机普及板”是一种模块式结构的 51 系列单片机系统,它以多种方式提供给用户。该系统适用于初学者,中小學生,工矿企业科技人员及其他电脑爱好者。

DP851 整机,438 元/套。

单片型自开发型套件(主板+PC 机电缆),售价 279 元/套,自开发型 B 型套件(主板+薄膜键盘板),售价 324 元/套。

教学实验板配置(自开发器+实验板+辅助件),售价 397.5 元/套。

另备有固化板,三位半 A/D,四位半 A/D,A/D、D/A 板等多种扩展板。

• 固体录音机

固体录音机用 T6668 语音处理器组成录放系统,可由微机等进行 CPU 控制和外部开关人工控制。T6668 是一种专门用于语音信号的记录和再生的超大规模 CMOS 集成电路,该电路采用动态存储器容量可达 1M 位。套件价格 125 元/套,另加邮费 10 元。

• 普及型逻辑笔

普及型逻辑笔用红、黄、绿三种颜色的发光二极管显示高电平、低电平、高阻及脉冲的有无四种状态,它是学习 TTL 集成电路的好帮手。买一个正规的逻辑笔要花费近百元,自己动手,十几元就可以自己组装,既经济,又实惠。原理及制作过程分别参看 93 年第 9 期和 94 年第 2 期《电子与电脑》,价格:套件 15 元/套,邮费 2 元,整机 20 元/套,邮费:价格 \times 6%。

• SW 系列

本厂生产的数字仪表共分三大类十个方面的一百多个品种,该仪表与不同传感器配合使用,可测量温度、压力、电流、电压等多种物理量,并能实现指示调节和报警。

各种仪表均可按用户要求带有 0—5V,1—5V,0—10mA,4—20mA 等一组标准信号输出。

本仪表测量精度高,显示直观,抗震性能好,使用寿命长。

同时本厂生产各种类型的传感器,并能与仪表配套,价格优惠。

• 本厂承接制作各种智能广告牌。

保证信誉优质服务

联系电话:8512947 联系地址:北京西城区月坛南街 83 号

邮 编:100045 联系人:张文奇 庞文姬

开户行:工商银行百万庄分理处 帐号:014—047014—96

北京市西城区电子仪表厂

北京市单片机应用技术协会宗旨

推广单片机普及应用,开发、咨询、服务融为一体
适应应用者不同需求,主板、选件、软件自成系列

一、通用单片机应用板/智能数字表系统“BJ-10 系列”简介

为推动单片机应用,减少应用开发中的重复性劳动,北京市单片机应用技术协会开发出 BJ-10 系列单片机应用板/智能数字表系统。该系统可适应不同用户的要求,提供通用的硬件环境和采样,显示,打印,报警,时钟,数制转换,数值运算等通用软件。对于不同的应用,只需选择适当的接口,增加外围扩展电路,即可组成智能仪表,数据采集器,或程序控制器等。用户不必重新设计调试硬件电路,不必编制通用程序,因而可大为缩短应用性开发的周期。

用户在选用适用系统,经开发定型后,BJ-10 设计开发者可按用户的要求对适用板进行专门设计,以便继续为用户提供硬件(软件)条件。几年来的实践表明,双方各自发挥优势的合作方式,前景广阔。

二、BJ-10 系列及选件报价表(94.5) 批量优惠 价格面议

序	应用板及选件名称	型号	单价
1	BJ-10(A)通用智能数字仪表板(基本型)	BJ-10A	510 元/块
2	BJ-10SD 自开发型通用智能数字仪表板(自开发型)	BJ-10SD	890 元/块
3	BJ-10B 自开发型通用智能数字仪表板(用户型)	BJ-10B	640 元/块
4	BJ-10C 通用智能数字仪表板(简化型)	BJ-10C	490 元/块
5	BJ-10D 通用智能数字仪表板(光电隔离型)	BJ-10D	996 元/块
6	BJ-10X 通用智能数字仪表板(最小型)	BJ-10X	100 元/块
7	BJ-113 三位半 A-D 选件板	BJ-113	120 元/块
8	BJ-1132 二个三位半 A/D 选件板	BJ-1132	240 元/块
9	BJ-114 四位半 A/D 选件板	BJ-114	240 元/块
10	BJ-118 八路八位 A/D 选件板	BJ-118	210 元/块
11	BJ-121 一路八位 D/A 10MA 输出	BJ-121	140 元/块
12	BJ-122 二路八位 D/A 选件板	BJ-122	290 元/块
13	BJ-10574 高速 A/D 选件板	BJ-10574	1100 元/块
14	BJ-10574D 高速 A/D D/A 选件板	BJ-10574D	1700 元/块
15	BJ-10VF 压频 A/D 选件板	BJ-10VF	88 元/块
16	BJ-10GD32 三十二路光电隔离板	BJ-10GD32	880 元/块
17	BJ-10K31 单端三十二路模拟开关板	BJ-10K31	100 元/块
18	BJ-10K32 双端三十二路模拟开关板	BJ-10K32	160 元/块
19	BJ-10K24 二十四位 I/O 扩展板	BJ-10K24	110 元/块
20	BJ-10KRS 串行显示扩展板	BJ-10KRS	120 元/块
21	BJ-10K40 四十位显示扩展板	BJ-10K40	120 元/块
22	BJ-10KCD 液晶显示扩展板	BJ-10KCD	400 元/块
23	BJ-10KDG 看门狗电路扩展板	BJ-10KDG	80 元/块
24	BJ-10YSZ 硬时钟电路扩展板	BJ-10YSZ	90 元/块
25	BJ-10DBH 掉电保护板	BJ-10DBH	90 元/块
26	BJ-10EPR 自升压 EPROM 固化板	BJ-10EPR	240 元/块
27	BJ-101DY 智能表配套电源板	BJ-101DY	160 元/块
28	BJ-10PC IBM-PC 与 BJ-10 通讯软件	BJ-10PC	100 元/片
29	BJ-10DL IBM-PC 与 BJ-10 通讯电缆	BJ-10DL	50 元/条
30	BJ-10RJK 软件库	BJ-10RJK	100 元/片
31	BJ-10YY 语音板	BJ-10YY	200 元/块

开发单位:北京市单片机应用技术协会(BJDP)

销售单位:北京市技术交流站 开户行:工商北京珠市口办事处(账号:089632-30)

地址:北京市虎坊路 13 号(邮编:100052) 联系人:何维才(电话:3035931-2143)

语音合成技术在电脑游戏机中的应用

易继锴 张京伟 仇之明 北京工业大学自动化系(100022)

随着信息工程的不断发展,计算机技术不仅能处理文字信息和进行数学计算,还能与其它传媒方式接轨,处理图像、声音等更丰富的信息。计算机语音技术要涉及到数字信号处理、微机技术、人工智能、模式识别、语言声学、语言学等许多科学领域,这就意味着它在众多领域中的应用价值。

目前,家庭游戏机作为一种高技术的娱乐产品已进入千家万户,并受到人们的普遍喜爱。常接触游戏机的人们也许会注意到,在一些游戏机中伴有说话声,譬如在“麻将”游戏中有“快发牌”的语音;在“篮球”游戏中有“double dribble”的英文发音等等。游戏中的人物会发声无疑给游戏本身增添了趣味性和生动性,本文也正是由此出发,研究开发了电脑游戏机上的语音合成技术。

1. 电脑游戏机的硬件结构

目前市场上销售的游戏机虽然种类繁多,但硬件线路都是源于日本任天堂游戏机,所不同的只是在制式转换方面加了一些改动,其原理框图如图1所示。

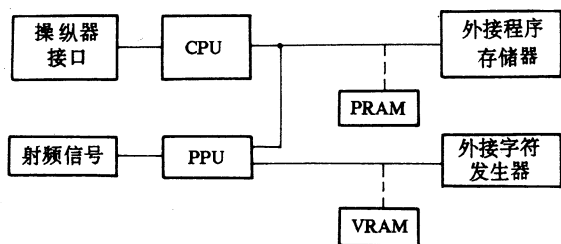


图1

CPU 是专用集成电路 6527,其指令系统与 6502 微处理器兼容,内部功能十分强大。PPU 是显示控制器,游戏中背景画面和卡通就是通过 CPU 对 PPU 的控制完成的。我们所要使用的声音处理控制器是隐藏在 CPU 的内部,通过对其声音控制寄存器写入不同的控制字,游戏机就能够按编程要求发出合成的语音。

2. 语音信号的编码方式

游戏机能够发出语音,主要还在于有存储好的语音数据,这部分语音数据是通过采样得到的原始语音信号进行编码得到的。

目前,利用计算机进行语音合成有多种方法。主要有:波形存储法和参数合成法。还有一种规则合成法。这三种方法其实都是一个复杂的语音信号的数字处理过程,包括编码、传输、存储和译码。

波形存储法是一种相对简单的语音合成技术,基

本思想就是将人讲的一些单词、词组或短语的声波波形通过 A/D 转换存入存储单元,在使用时,将相应的数字量调出,经 D/A 转换成模拟信号输出到扬声器。这种方法实现起来简便,合成后的音质也能接近人的通常发音质量,适用于单词、短语的合成,但所需存储容量很大,不适于句子的合成。参数合成法是一种分析合成法,是基于分析和模拟人的发音器官来寻求建立描述方程以进行语音合成。规则合成法是将发音的音素所表示的语音信号存储起来,拼音随时取用。参数合成法和规则合成法实现起来要比波形存储法复杂得多。

通过对一些游戏软件中的语音数据进行分析,根据其发音的特点和多是一些短语的现象,可知其语音数据是采用波形存储法得到的。因游戏机中的内存有限,仅有 32KB 可供用户使用,所以对语音数据就要求少而精。这样一来,那种对语音波形按照一定周期的采样,存储量大,不经压缩处理的脉冲编码调制方式就不能满足要求。因此就必须对实际采进来的语音信号进行压缩,而且不能丢失原语音信号中所表达的信息。语音合成技术中有差值量化方式可满足上述要求。DM 调制方式(Delta Modulation)是差值量化的最简单应用。对实际采样进来的原始语音数据由软件进行 DM 调制,如采用 8 位 A/D 转换语音信号,那么经 DM 调制后的语音数据量仅为原语音数据长度的 1/8。下面详细讲讲 DM 调制方式。

DM 编码方式用的是一阶固定阶距预测器,采用差值量化方式,以位(bit)为单位进行编码,即对一个语音信号的信息表示方式是基于最低限度的位。在此方式中,二值逻辑 0 和 1 表示着语音信号值的高低。

在构成系统时,先确定好一个固定的增量 δ (delta),然后对由以前的编码得到的语音信号值与现在输入的语音值的残差信号进行编码。也就是说要得到眼前输入的语音信号值 S_n ,以便与过去时刻预测编码得到的语音信号值 S_{n-1} 相减得到差值信号 e_n 。减小 e_n ,从而使 S_{n-1} 接近先前的语音信号 S_n ,确定 + Δ 还是

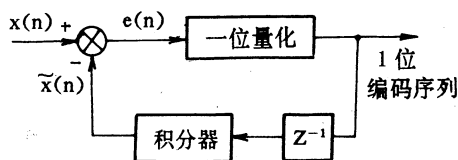


图2

$-\Delta$,以决定编码时是对该位置“1”或“0”。DM 编码框图如图 2 所示。

$e(n) = X(n) - \hat{X}(n)$,若 $e(n) > 0$,则编码输出为“1”, $\hat{X}(n) = \hat{X}(n-1) + \Delta$;若 $e(n) < 0$,则编码输出为“0”, $\hat{X}(n) = \hat{X}(n-1) - \Delta$ 。

DM 编码方式虽然简单,但也有两点不足。第一,当输入信号的变化快,即输入信号的斜率较大时,用固定阶距对输入信号量化, $\hat{X}(n)$ 脉冲波形将向 $X(n)$ 的右侧移动,即恢复波形跟不上原来的波形,造成“斜率过载”,从而产生量化误差,造成声音失真。第二,在残差为零或接近零的区域,量化后的码化器输出后“0”和“1”的交替序列,这种随机交变电平成为随机噪声干扰原来的电平,产生的噪声被称为颗粒噪声。为减小由斜率过载造成的失真,需要大幅度提高采样率来保证音质,要想得到信噪比 40dB 左右的合成语音,采样频率须提高到 100kps 以上。通过选择合适的 Δ 值,也可取得一定的效果,特别是对颗粒噪声的消除,选择的阶距较小得到的效果会好。增量调制的最佳阶距的经验公式是:

$$\Delta_{opt} = \{E[X(n) - X(n-1)^2]\}^{1/2} \text{Ln}(2F_0)$$

F_0 为超采样指数,表示为:

$$F_0 = FS/2F_N$$

F_S 为采样频率, F_N 为输入信号的奈奎斯特频率。

语音的合成方式还有 DPCM, ADM, APC, ADPCM 等多种。经试验,在电脑游戏机上以 DM 方式编码的数据输出音质较好。

3. 原始语音波形数据采集

通过对模拟语音信号进 A/D 转换,采集可得到语音数据。声音包含有音调、音色和响度三种特征。其中音调反映声音的基本频率,在声音波形中,表现为单位时间过零次数的多少;响度反映了声音的强弱变化,在波形中表现在幅度的变化;音色反映声音的高次谐波特征,决定了声音波形的复杂程度。要想得到音质较高的语音信号,那么对于声音的三个特征就都要采集。语音的变化频率通常是在 300Hz~5000Hz 之间,根据香农采样定理,则采样频率不能低于 10KHz。又因是采用了 DM 编码方法,这种编码方式要降低误差就必须大幅度提高采样率。基于上述的三个原因,就对硬件电路,特别是 A/D 转换器提出了较高的要求。在国内采用的 A/D 器件中,AD574 是较为理想的选择。其转换时间典型值为 25 μ s,能保证对语音信号的采样率达到 20kps 以上。用 PC/XT 机对模拟语音信号采样,所得语音波形可以达到要求,经 DM 调制后,在游戏机上可得到较好的还原效果,尤其对中文的还原效果更好。

为了最大限度地降低语音信号的失真,在选用器件上要尽可能选用低噪声、线性度高的器件。例如,对语音信号的放大,本采样系统采用了具有低噪、宽频、高速特点的 NE5532 作为滤波放大元件,达到了较好的效果。采集电路硬件框图如图 3 所示。

语音信号经放大滤波后送到 AD574 进行转换。

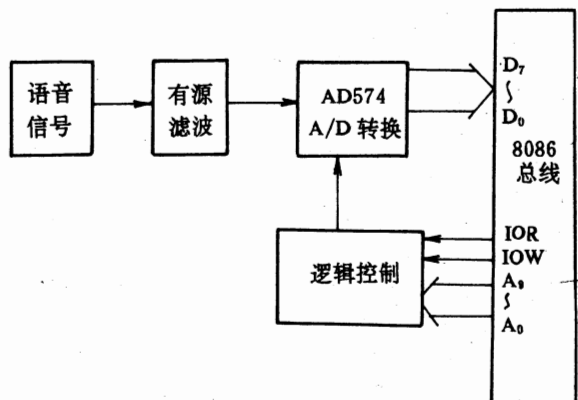


图 3

通过逻辑控制 AD574 的转换与读取,可将语音数据读入数据缓冲区。

为了保证语音数据采集和处理的速度,采样程序必须采用汇编语言编制。其框图如图 4 所示。

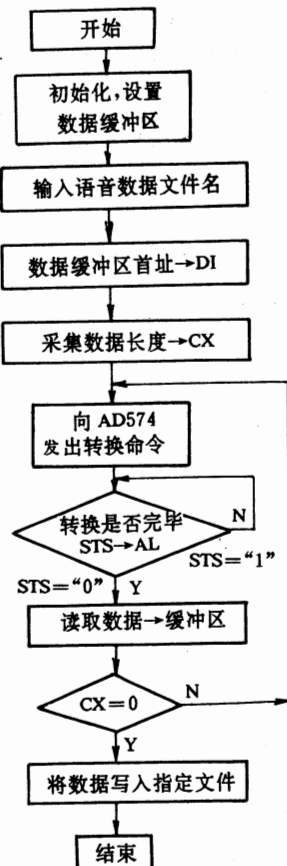


图 4

4. DM 调制方式的软件实现

目前市场上有多种语音合成芯片,如东芝的 T6668,联华的 UM5100,TI 公司的 TSP5220 等。这些芯片都是通过芯片内部的硬件结构实现不同的合成方式。我们在实践中采用了 C 语言编程实现 DM 调制,

效果良好。程序框图如图 5 所示。

图 6 给出了原始语音信号和根据 DM 编码合成技术构造的语音波形。由图可见,原始语音变化越平缓,则预测语音信号的跟踪性能越好,残差越小;反之,当原始语音信号出现较大幅度跳动时,预测信号的跟踪性能下降,残差增大。由此也可看出,采样率越高,用 DM 方式合成的效果就越好。

5. DM 调制语音合成技术在游戏机上的实现

将经 DM 调制后的语音数据调入游戏机,通过向语音控制寄存器写控制字,可将数据经 6527CPU 内部 D/A 转换后输出到电视机的扬声器。语音数据的传输采取的是 DMA 传输方式,一次最大传输语音数据量不能超过 4KB,这就限制了语音一次输出的时间不能

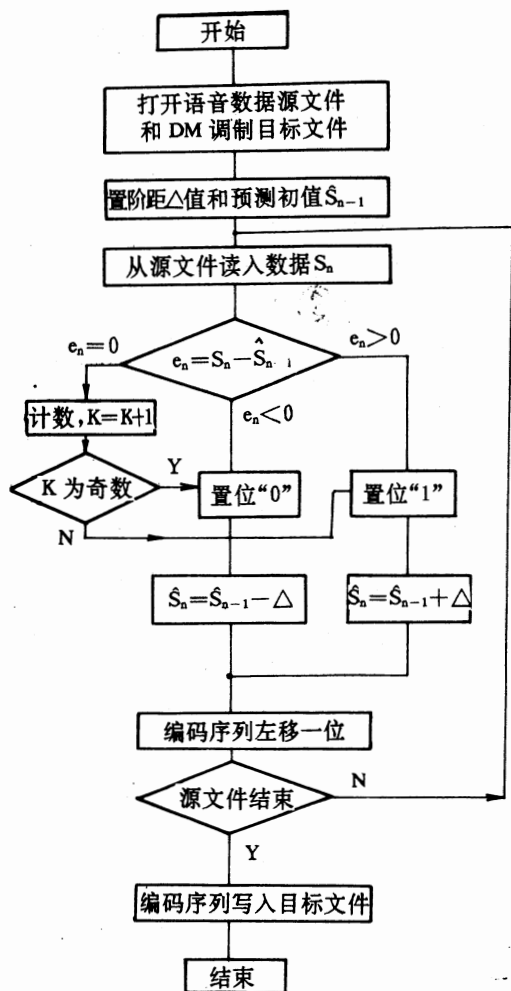


图 5

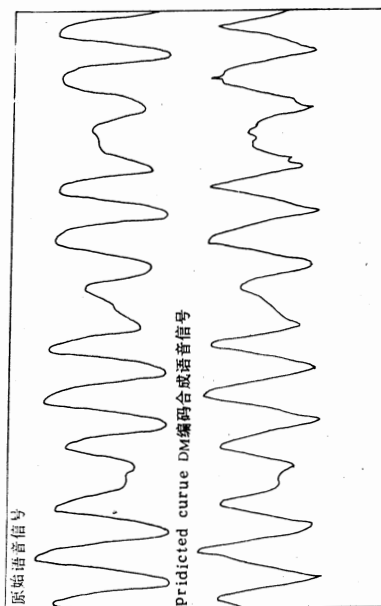


图 6

超过 1.5 秒。在此时间内,只能发一、两个单词的语音,这也正和人们在玩游戏时所注意到的一样,一个游戏中只有一、两个短促的语音,这主要是受到了游戏机数据传输方式和容量的限制。

广东省潮阳市陈店环东电子有限公司

电子元器件

陈店——汕头特区的所辖镇,它以其天时、地利、人和,而成为全国最大的电子市场之一,陈店环东电子有限公司正是拥有这有利条件,而使她的业务范围遍布全国各地,经营的各类进口电子元件品种达到几千种型号,一直以来,都在北京的《家用电器》《电子世界》《家电维修》《北京电子报》上海《现代通信》《实用无线电》成都《电子报》广东《广东电子》等杂志、报刊上刊登邮购广告,和全国各地许多客户、厂家公司建立了长期友好的业务往来,因此,如果你想得到该公司的最详细报价目表和简介,那么,请你来函付资贰元索取。热情欢迎来函、来电、咨询、洽谈合作、邮购。

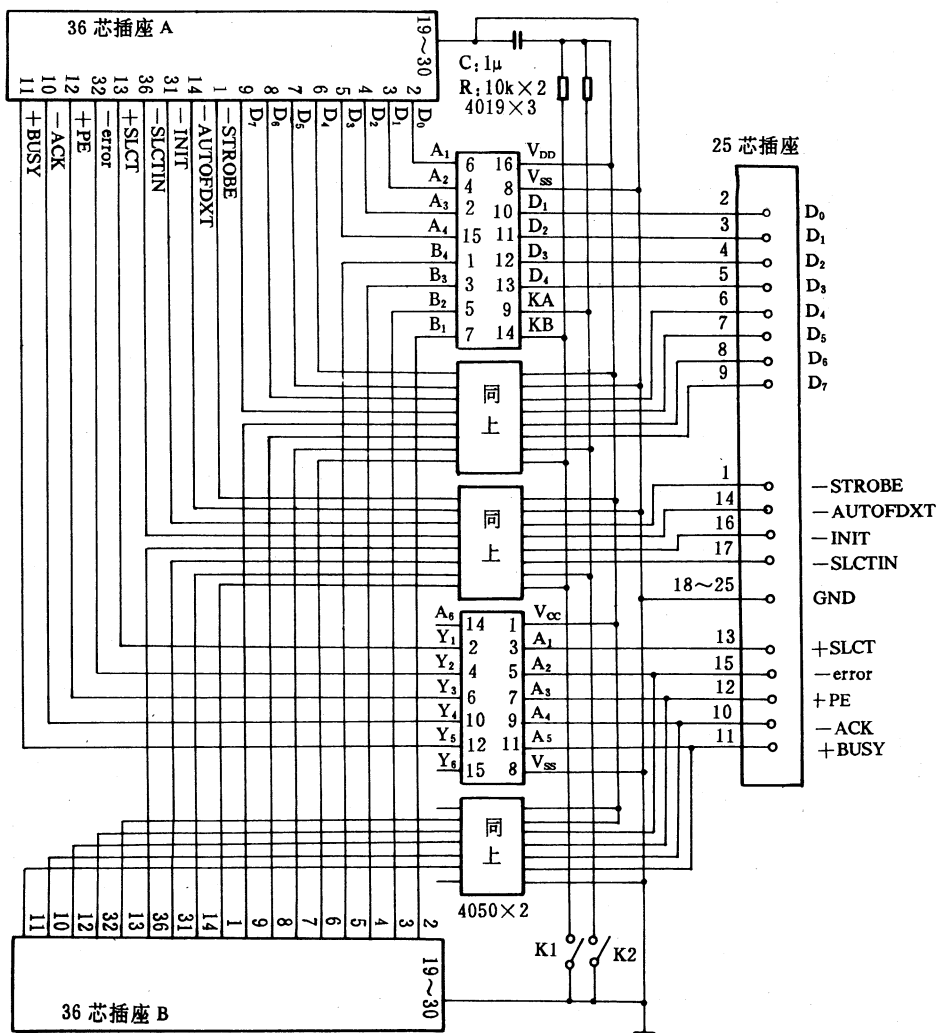
地址:广东省潮阳市陈店镇东方一路 88 号
电话:公司(07644)481478 邮购部:484573
传真:485649 邮编:515152
经理:蔡镇荣 手提:(0754)9019654

一种打印机共享电路

袁汝钧 武汉工业大学凯狮公司(430070)

下图给出了两台计算机共享一台打印机的接口电路。用打印机电缆分别将 36 芯插座 A、B 与计算机 A、B 的打印口相联, 25 芯插座与打印机相联。当连动开关 K1 按下时, 3 片 CD4019 同时选中 A 组数据, 即由计算机 A 享用打印机, 当 K2 按下时, 由计算机 B 享用

打印机。打印机的输出信号由 CD4050 分别送往计算机 A、B, 实现信号全隔离, 并巧用打印机 SLCT 信号线给 CD4019 和 CD4050 供电。按此电路制作的不需电源的打印机共享器具有通用性。使用时应避免 K1、K2 同时按下或不按下。



第二讲 针式打印机的结构及各部分的功能

赵继文 西安陕西省计量测试研究所(710048)

一、针式打印机的结构

针式打印机是由微型计算机、精密机械和电气构成的机电一体化智能设备。其结构框图如图 1 所示,它可以概括性地划分为打印机械装置和电路两大部分。

打印机械装置:主要包括打印头、字车机构、色带机构和走纸机构,另外还有机壳和机架。

电路:主要由控制电路、驱动电路、打印机状态检测电路、DIP 开关、操作面板电路、接口和电源等组成。

二、打印机械装置

1. 打印头

形。打印机基本上都采用单向步进循环的环形色带机构,在打印过程中,字车左右横向移动时,通过换向机构使色带驱动轴做单向同步旋转,并带动色带盒中色带周而复始地循环,使其不断改变色带撞击位置,以保证色带均匀使用,从而延长色带使用寿命。

4. 走纸机构

走纸机构是驱动打印纸换行的机构。它以走纸电机为动力源,拖动走纸机构使打印纸按照规定行距向前或向后移动。走纸机构主要有两类:一类是摩擦式,适用于无走纸孔的打印纸;另一类是链轮式,适用于有走纸孔的打印纸。大部分打印机都同时具备这两种机构。

三、电路

1. 控制电路

打印机的主控电路本身是一个完整的微型计算机,一般由微处理器(通称 CPU)、RAM、ROM、地址译码器和 I/O 电路等组成。此外,还有打印头控制电路、字车电机控制电路和走纸电机控制电路等。微处理器是控制电路的核心,由于当前微电子技术的高速发展、单片机已将微计算机的 CPU、RAM、ROM、I/O 电路、定时/计数器、串行接口和中断系统等集成在一个芯片上,所以许多打印机都采用高性能的单片机替代微处理器及其外围电路。

控制电路的主要功能如下:

(1)联接主计算机(以下简称主机)并和主机通信。即通过并行或串行接口接收来自主机的打印信息和控制命令,把打印机的打印信号和回答信号送给主机。

(2)控制字车横向左、右移动。

(3)控制走纸机构工作。如走纸、换行、换页、调整行距以及反向走纸等。

(4)控制打印头出针操作。按照控制命令打印各种字符、数字、汉字、图形,并变换字体、字形大小和格式等。

(5)读取操作面板上的开关状态,并控制联机/脱机、换行和换页等。

(6)检测 DIP 开关状态。按照 DIP 开关状态设置打印机的参数,如选择国际字符、选择页长、走纸行距、打印方式(高密打印或高速打印)等。

(7)通过检测电路检测打印机状态传感器的状态,异常时发出警报。

(8)控制打印机的自检功能。即在脱机状态下,人工输入自检命令,令其执行 ROM 中的自检程序,来检

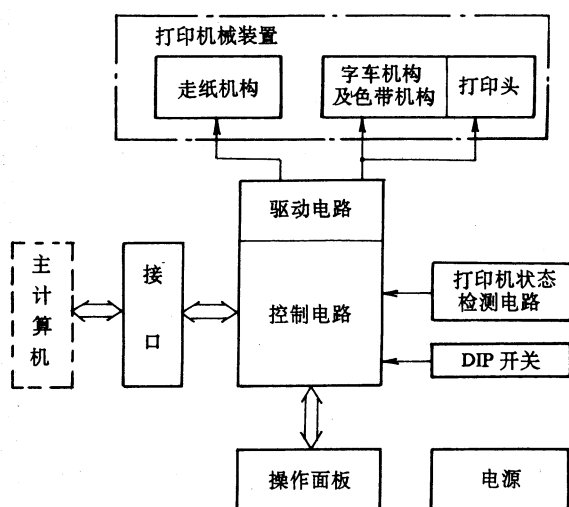


图 1 针式打印机结构

打印头又叫印字机构,它装载在字车上,用于印字,是打印机的关键部件之一。针式打印机按针数分类,可分为 1、5、7、8、9、12、14、16、18、24、32、36 和 48 针,目前常用的是 9 针和 24 针。

2. 字车机构

字车结构是装载字车,并以字车电机为动力源,在传动件的拖动下,沿导轨做左右往复直线间歇运动的机构。

3. 色带机构

色带的作用如同复写纸一样,当打印针撞击色带时通过色带的复印作用,在纸上印出字符(汉字)或图

测打印机的功能是否良好。

2. 驱动电路

驱动电路包括 3 部分:打印头驱动电路、字车电机驱动电路和走纸电机驱动电路。这 3 部分的驱动都是在控制电路的控制下驱动的,为了叙述方便将与此 3 部分密切相关的控制电路纳入驱动电路中,故以下将称为打印头控制与驱动电路、字车电机控制与驱动电路和走纸电机控制与驱动电路。

(1) 打印头控制与驱动电路

打印头控制与驱动电路的功能是接收字符发生器(即字库)输出的针数据信号,并对该信号进行功率放大后,送给打印头以驱动打印针出针操作。

(2) 字车电机控制与驱动电路

字车电机控制与驱动电路的功能是驱动字车电机做正、反向运转,以便拖动字车在导轨上执行左、右横向运动。其运动形式有下列 4 种:

a. 打印机加电开机时,ROM 中监控程序进入初始化阶段,无论字车在开机前停在什么位置,都将驱动字车返回左端初始位置(Home)。

b. 打印机单向打印时,驱动字车由左向右运动并执行打印,当一行字符打印完毕后,字车立刻返回左端初始位置(即回车),然后再开始打印下一行。

c. 打印机双向打印时,字车由左向右运动,执行完一行打印后,驱动字车从下一行右端(即换行)开始从右向左运动,执行反向打印。

d. 驱动字车执行加速、匀速和减速运动。

打印机的字车电机通常都采用四相步进电机,步进电机每转动一个步距角时,字车相应地移动一微步,以使字车在导轨上做间歇运动。微步距越小,打印点之间的中心距越小,横向点密度就越大,打印的字形也就越逼真,说明打印机性能优越。目前,国内常见 24 针打印机的字车电机步距角都设计为 0.9° 或 1.8° ,相应的字车微步距为 0.141mm ($1/180$ 英寸)或 0.282mm ($1/90$ 英寸)。

(3) 走纸电机控制与驱动电路

走纸电机控制与驱动电路用于驱动走纸电机正向或反向运转,从而拖动走纸机构执行下述操作:

a. 在脱机状态下,按动操作面板上换行(LF)键时,走纸一行,通常行距是 4.23mm ($1/6$ 英寸)。按动换页(FF)键时,走纸一页,通常为 279.4mm (11 英寸)或 304.8mm (12 英寸)。

b. 在打印期间,打印完一行后,在换行命令的控制下,走纸一行,其行距依照打印机当前设置而定。

c. 联机打印时,当打印曲线、图形和纵向放大汉字时,按规定能将打印纸移动一微步。如 M-1724 打印机的微步距是 0.212mm ($1/120$ 英寸);LQ-1600K、LQ-2500K 和 AR-2463 打印机的微步距是 0.141mm ($1/180$ 英寸);CR-3240 和 AR-3240 打印机的微步距是 0.07mm ($1/360$ 英寸)。

d. 能在规定的最大走纸量的范围内,执行正向走纸或反向走纸。

走纸电机通常亦采用四相步进电机。

3. 打印机状态检测电路

打印机状态检测电路用于检测打印机的工作状态,其设置因机型而异,这里只介绍几种主要检测电路。

a. 字车初始位置(Home)检测电路。当打印机开机时,或接收到主机的初始化信号时,或打印过程中遇到回车控制命令时,字车都应返回左端初始位置。该电路就是用来检测在上述状态下字车是否每次都能返回初始位置。电路中传感器一般用光敏遮断器或簧片式机械开关。

b. 纸尽检测电路。该电路用于检测打印机是否有纸,所用传感器有反射型光敏传感器、干簧开关、簧片式机械开关和霍尔集成电路等几种。

上述两种检测电路是各种针式打印机必须具备的。

c. 打印头温度检测电路。近期推出的汉字打印机都备有该电路,用于防止打印头因连续打印产生高温而损坏打印头。温度传感器都采用热敏电阻,装在打印头内部。打印期间,当打印头温度上升到一定限度时。就变双向打印为单向打印,继续升到极限值时停止打印,字车只做往复横向移动;当打印头温度降下来后,自动恢复正常打印。

d. 送纸调整杆位置检测电路。一般用簧片式开关的“ON”与“OFF”状态设置走纸方式。

4. DIP 开关读入电路

安装在打印机底板上的 DIP 开关主要用于设置打印机的若干打印功能状态,如选择国际字符集、行距、页长、零字体、高速或高密打印方式等。近期推出的一些汉字打印机如 CR-3240、AR-4400、NEC P6300 等,不再专门设置 DIP 开关,它们利用操作面板上的功能键就可以设定打印机的开机功能状态,即进行 EDS 设定(EDS 是英文电子 DIP 开关的缩写)。

5. 操作面板电路

操作面板上有若干按键和指示灯,其设置因机型而异,但基本功能如联机/脱机、换行、换页、纸尽报警等都是相同的。

6. 接口

打印机通过接口和主机联接。各种打印机都配有 Centronics 标准并行接口;有的打印机还配有串行接口,一般采用 RS-232C 型。

7. 电源

针式打印机的电源主要有两种:一种是 +5V 电源,供给逻辑电路和操作面板上指示灯;另一种是高压电源(电压因机型而异),用于驱动字车电机、走纸电机和打印头。

第六章 背景画面绘制的编程技巧(续)

于 春 北京裕兴机械电子研究所(100035)

二、整幅画面绘制的编程技巧

绘制整幅画面时,一般采取关显示后用查表法送显。查表法绘图有多种形式,本节仅选择介绍几种最常用的方法。

1. 直接送显绘图法(《小学算术一年级》绘图模式)

《小学算术一年级》游戏是日本东京小学推荐的优秀教学软件,它的背景画面绘制采取了最原始的方法——直接查表法。其绘图源程序如 No. 6—9。

No. 6—9 《小学算术一年级》绘图源程序

AD04	8518	STA \$18	;保护 A 值	AD41	C8	INY	
AD06	A900	LDA # \$00		AD42	B111	LDA (\$11),Y	
AD08	8516	STA \$16	;设置取数方式	AD44	8514	STA \$14	;每组送数的数量
AD0A	8619	STX \$19	;保护 X 值	AD46	C8	INY	
AD0C	841A	STY \$1A	;保护 Y 值	AD47	B111	LDA (\$11),Y	
AD0E	68	PLA		AD49	8515	STA \$15	;本次送数组数
AD0F	850D	STA \$0D		AD4B	C8	INY	
AD11	68	PLA		AD4C	A514	LDA \$14	
AD12	850E	STA \$0E		AD4E	C515	CMP \$15	;若(\$14)<(\$15)则转
AD14	A001	LDY # \$01		AD50	9071	BCC \$ADC3	;\$ADC3,改纵向送数
AD16	B10D	LDA (\$0D),Y		AD52	A615	LDX \$15	
AD18	8511	STA \$11	;绘图数据区地址低位	AD54	8A	TXA	
AD1A	C8	INY		AD55	48	PHA	;送数组数入栈保护
AD1B	B10D	LDA (\$0D),Y		AD56	A504	LDA \$04	
AD1D	8512	STA \$12	;绘图数据区地址高位	AD58	D032	BNE \$AD8C	
AD1F	A50D	LDA \$0D		AD5A	A500	LDA \$00	
AD21	18	CLC		AD5C	48	PHA	
AD22	6902	ADC # \$02		AD5D	2035AC	JSR \$AC35	;屏蔽 \$2000 的 D7 位
AD24	850D	STA \$0D		AD60	A500	LDA \$00	
AD26	9002	BCC \$AD2A		AD62	8D0020	STA \$2000	;设置送数方式
AD28	E60E	INC \$0E		AD65	A510	LDA \$10	
AD2A	A50E	LDA \$0E		AD67	8D0620	STA \$2006	;置 PPU 地址高位
AD2C	48	PHA		AD6A	A50F	LDA \$0F	
AD2D	A50D	LDA \$0D		AD6C	8D0620	STA \$2006	;置 PPU 地址低位
AD2F	48	PHA	;存子程序返回地址	AD6F	A614	LDX \$14	;以 X 为取数计数器
AD30	A000	LDY # \$00		AD71	B111	LDA (\$11),Y	;从数据区取数送入
AD32	A516	LDA \$16		AD73	8D0720	STA \$2007	;PPU
AD34	C902	CMP # \$02	;若 \$16=2 则不改变	AD76	A516	LDA \$16	;若(\$16)=01 则一个
AD36	F00A	BEQ \$AD42	;PPU 地址	AD78	C901	CMP # \$01	;数据连送 X 次
AD38	B111	LDA (\$11),Y		AD7A	F006	BEQ \$AD82	;若(\$16)=00 则连送
AD3A	850F	STA \$0F	;PPU 地址低位				其后的 X 个数据
AD3C	C8	INY		AD7C	E611	INC \$11	令取数地址低位加一
AD3D	B111	LDA (\$11),Y		AD7E	D002	BNE \$AD82	
AD3F	8510	STA \$10	;PPU 地址高位	AD80	E612	INC \$12	;若低位为零则令高位加一
				AD82	CA	DEX	
				AD83	D0EC	BNE \$AD71	
				AD85	68	PLA	
				AD86	2039AC	JSR \$AC39	
				AD89	4CACAD	JMP \$ADAC	
				AD8C	205DAE	JSR \$AE5D	
				AD8F	A614	LDX \$14	
				AD91	8A	TXA	
				AD92	2081AE	JSR \$AE81	
				AD95	B111	LDA (\$11),Y	

```

AD97 2081AE JSR $AE81
AD9A A516 LDA $16
AD9C C901 CMP # $01
AD9E F006 BEQ $ADA6
ADA0 E611 INC $11
ADA2 D002 BNE $ADA6
ADA4 E612 INC $12
ADA6 CA DEX
ADA7 D0EC BNE $AD95
ADA9 208DAE JSR $AE8D
ADAC A50F LDA $0F
ADAE 18 CLC ;令 PPU 地址加 20H,
ADAF 6920 ADC # $20 ;使其指向下一行
ADB1 850F STA $0F
ADB3 9002 BCC $ADB7
ADB5 E610 INC $10 ;若有进位则令高位加一
ADB7 68 PLA
ADB8 AA TAX
ADB9 CA DEX ;令送数组计数器减一
ADBA D098 BNE $AD54 ;不为零则转 $AD54
ADBC A518 LDA $18 ;继续送数,若为零则
ADBE A619 LDX $19 ;恢复原 A、X、Y 寄存器
ADC0 A41A LDY $1A ;值结束送数
ADC2 60 RTS

```

复位画面数据区如下:

B45E 0020
PUU 地址

B460 20 20 CE CE CE CE CE CE CE CE CE CE CE CE CE CE CE CE CE
—本次送数的组数存于 \$15
—每组送数数量置于 \$14。以后的数据都是绘图数据

B470~B4DF 各单元内容均为 CE
B4E0 CE CE CE CE CE CE CE CE CE CE CE E0 E1 E2 E3 82 83
B4F0 82 83 CE CE CE CE A2 A3 CE CC CD CE CE CE CE CE CE
B500 CE CE CE CE CE CE CE CE CE CE CE E4 E5 E6 E7 92 93
B510 92 93 CE CE CE CE B2 B3 CE DC DD CE CE CE CE CE CE
B520~B55F 各单元内容均为 CE
B560 CE CE CE CE CE CE CE CE CE CE CE CE CE 80 81 80 81 E0
B570 E1 E2 E3 E8 E9 60 71 CC CD CE CE CE CE CE CE CE
B580 CE CE CE CE CE CE CE CE CE CE CE CE CE 90 91 90 91 E4
B590 E5 E6 E7 EA EB CE CE CE DC DD CE CE CE CE CE CE CE
B5A0 CE CE CE CE CE CE CE CE CE CE CE CE CE CE CE CE CE
B5B0 CE CE CE CE CE CE CE CE CE CE CE CE CE CE CE CE CE
B5C0 CE CE CE CE CE CE C4 C6 C6 C6 C6 C6 C6 C6 C6 C6
B5D0 C6 C6 C6 C6 C6 C6 C6 C6 C6 C6 C6 C6 C5 CE CE
B5E0 CE CE CE CE CE CE C7 CF CF CF CF CF CF CF CF CF CF
B5F0 C8 CF E0 19 E2 E3 CF CF CF CF CF CF CF CF CF CF CF
B600 CE CE CE CE CE CE C7 CF CF CF CF CF CF CF CF CF CF
B610 D8 D9 E4 E5 E6 E7 CF CF CF CF CF CF CF CF CF CF CF
B620 CE CE CE CE CE CE C7 CF CF CF CF CF CF CF CF CF CF
B630 CF CF CF CF CF CF CF CF CF CF CF CF CF CF CF CF CF
B640 CE CE CE CE CE CE C7 CF CF CF CF CF CF CF CF CF CF
B650 CA CB E0 19 E2 E3 CF CF CF CF CF CF CF CF CF CF CF
B660 CE CE CE CE CE CE C7 CF CF CF CF CF CF CF CF CF CF
B670 DA DB E4 E5 E6 E7 CF CF CF CF CF CF CF CF CF CF CF
B680 CE CE CE CE CE CE C7 CF CF CF CF CF CF CF CF CF CF
B690 CF CF CF CF CF CF CF CF CF CF CF CF CF CF CF CF CF

```

B6A0 CE CE CE CE CE CE C7 CF CF CF CF CF CF CF CF CF CF
B6B0 C8 CF E0 19 E2 E3 CF CF CF CF CF CF CF CF CF CF
B6C0 CE CE CE CE CE CE C7 CF CF CF CF CF CF CF CF CF CF
B6D0 D8 D9 E4 E5 E6 E7 CF CF CF CF CF CF CF CF CF CF
B6E0 CE CE CE CE CE CE C7 CF CF CF CF CF CF CF CF CF CF
B6F0 CF CF CF CF CF CF CF CF CF CF CF CF CF CF CF CF CF
B700 CE CE CE CE CE CE C7 CF CF CF CF CF CF CF CF CF CF
B710 CA CB E0 19 E2 E3 CF CF CF CF CF CF CF CF CF CF
B720 CE CE CE CE CE CE C7 CF CF CF CF CF CF CF CF CF CF
B730 DA DB E4 E5 E6 E7 CF CF CF CF CF CF CF CF CF CF
B740 CE CE CE CE CE CE D4 D6 D6 D6 D6 D6 D6 D6 D6 D6
B750 D6 D6 D6 D6 D6 D6 D6 D6 D6 D6 D6 D6 D5 CE CE
B760~B78F 各单元内容均为 CF
B790 CF 08 18 09 08 CF CF 09 CF 18 CF CF CF CF CF CF
B7A0~B81F 各单元内容均为 CF
B820 CF CF 00 00 00 00 00 00 00 00 00 00 00 00 0B 02 08
B830 00 00 50 50 5E 52 50 5F 50 50 55 55 55 55 55 55
B840 55 55 55 55 55 55 55 55 55 55 55 55 55 55 55 55
B850 55 55 00 00 00 00 00 00 00 00 00 50 50 50 50 50
B860 50 50

```

No. 6—9 所调用的子程序

```

AC35 A500 LDA $00
AC37 297F AND # $7F
AC39 8D0020 STA $2000
AC3C 8500 STA $00
AC3E 60 RTS

```

程序 No. 6—9 中已作了详细注释,不多解释。下面给出供我们直接调用的子程序 No. 6—10。

No—10(直接送显)背景绘图子程序(一)

```

7EF1 A9 FA LDA # $FA
7EF3 85 EE STA $EE
7EF5 A9 7E LDA # $7E
7EF7 85 EF STA $EF
7EF9 60 RTS
7EFA 20 00 7F JSR $7F00
7EFD 4C 0F 88 JMP $880F
7F00 48 PHA
7F01 8A TXA
7F02 48 PHA
7F03 98 TYA
7F04 48 PHA
7F05 A9 00 LDA # $00
7F07 85 50 STA $50 ;绘图数据区地址低位存 $50
7F09 A9 78 LDA # $78
7F0B 85 51 STA $51 ;绘图数据区地址高位存 $51
7F0D A0 00 LDY # $00
7F0F B1 50 LDA ($50),Y
7F11 85 52 STA $52 ;PPU 地址低位存 $52
7F13 C8 INY
7F14 B1 50 LDA ($50),Y
7F16 85 53 STA $53 ;PPU 地址高位存 $53
7F18 C8 INY
7F19 B1 50 LDA ($50),Y
7F1B 85 54 STA $54 ;每组送数的数量存 $54
7F1D C8 INY
7F1E B1 50 LDA ($50),Y

```

7F20 85 55	STA \$ 55	;本次送数组数存 \$ 55	7F58 90 02	BCC \$ 7F5C	
7F22 C8	INY		7F5A E6 53	INC \$ 53	;若有进位则令高位加一
7F24 A6 55	LDX \$ 55		7F5C 68	PLA	
7F26 8A	TXA		7F5D AA	TAX	
7F27 48	PHA	;送数组数入栈保护	7F5E CA	DEX	;令送数组数计数器减一
7F28 A5 32	LDA \$ 32		7F5F DO C5	BNE \$ 7F26	;不为零则继续送数
7F2A 48	PHA	; \$ 2000 原值入栈	7F61 68	PLA	;若为零则恢复原 A、
7F2B 20 67 7F	JSR \$ 7F67	;屏蔽 \$ 2000 的 D7 位	7F62 AA	TAX	;X、Y 寄存器值,结束
7F2E A5 32	LDA \$ 32		7F63 68	PLA	;送数
7F30 8D 00 20	STA \$ 2000	;设置送数方式	7F64 A8	TAY	
7F33 A5 53	LDA \$ 53		7F65 68	PLA	
7F35 8D 06 20	STA \$ 2006	;置 PPU 地址高位	7F66 60	RTS	
7F38 A5 52	LDA \$ 52		7F67 A5 32	LDA \$ 32	
7F3A 8D 06 20	STA \$ 2006	;置 PPU 地址低位	7F69 29 7F	AND # \$ 7F	
7F3D A6 54	LDX \$ 14	;以 X 为取数计数器	7F6B 8D 00 20	STA \$ 2000	
7F3F B1 50	LDA (\$ 50),Y		7F6E 85 32	STA \$ 32	
7F41 8D 07 20	STA \$ 2007	;从数据区取数送入 PPU	7F70 60	RTS	
7F44 E6 50	INC \$ 50	;令取数地址低位加一			
7F46 D0 02	BNE \$ 7F4A				
7F48 E6 51	INC \$ 51	;若低位为零则令高位加一			
7F4A CA	DEX				
7F4B DO F2	BNE \$ 7F3F				
7F4D 68	PLA				
7F4E 20 6B 7F	JSR \$ 7F6B				
7F51 A5 52	LDA \$ 52	;令 PPU 地址加 20H,			
7F53 18	CLC	;使其指向下一行			
7F54 69 20	ADC # \$ 20				
7F56 85 52	STA \$ 52				

现在,只要把绘图数据按 No. 6—9 中 \$ B45E 的格式写入 \$ 7800,就可绘出你希望的背景画面。限于篇幅,不再举例,留给读者练习。

从 No. 6—9、No. 6—10 可明显看出,直接送显绘图法的优点是:绘图数据区为平面结构,数据与显示单元的关系一一对应,单纯从数据区就可看出绘图效果,背景画面设计直观、容易,修改方便,擅长于细腻画面的描绘,特别适合初学者使用;缺点是:由于每一个显示单元和配色单元都要占一个内存单元,一幅背景画面要占用 1024 个内存单元,故数据区较庞大。

(上接第 24 页)

上述二例讲述了实现屏幕扩展功能的基本编程方法。但实用中如果采用这种方法,定义一次屏幕,就需要编写一段程序,比较麻烦。我们可以编写一段通用的程序,来实现屏幕扩展功能呢,请见下例:

[例 6]编写实现屏幕扩展功能的通用程序。

C>debug

—A100

MOV DX,117 ;DX=键入缓冲区的字节偏移

MOV AH,0A ;键入字符串子功能号

INT 21

MOV AL,1B ;控制代码 ESC 送 118 单元

MOV[118],AL

MOV DX,118

MOV AH,09

INT 21 ;按照键入的字符串定义屏幕

MOV AH,4C

INT 21 ;返回 DOS

—RCX

CX 0000

:20 ;确定该程序的字节长度

—N ESC.COM ;确定该程序的文件名

—W ;写入磁盘

—Q

用 DEBUG 编写、汇编了该程序,并以文件名 ESC.COM 写入磁盘。执行时,在文件名 ESC 之后打回车,再打"["及其后的相应字符串即可,使用起来十分方便。特别是,本通用程序不仅适用于屏幕,而且也适用于键盘功能的扩展(相应的控制字符串请见第二讲的四),这是一段较好的实用程序。

掌握屏幕扩展功能,可以丰富我们的编程方法和技巧,简化操作,提高效率,因此是很有必要的。

Microsoft Windows 3.1 应用基础

——桌面办公用具

徐鹏力 中国科学院软件研究所

Windows 作为一种集成化图形操作环境,不但提供对计算机软硬件资源的控制功能,还为用户提供了丰富多彩的桌面办公用具——Accessories。利用桌面办公用具,用户可处理日常工作中写作、画图、便笺、时钟、日历及计算器等诸多工作。

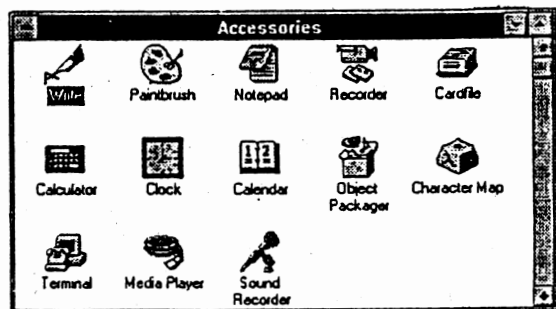


图1 桌面办公用具

一 书写器(Write)

字处理是办公系统中常见工作之一。Write 作为一种 Windows 字处理应用程序,经常被用来打印各种文件、报表及个人往来信件。与传统的 WPS 文字处理系统相比,Write 具有易于学习掌握、排版及打印字体美观等许多优点。尤其是 Windows 3.1 中文版推出以来,Write 在中西文字处理及图文混排方面更为用户提供了方便。尽管 Write 是以文字为主的处理程序,但

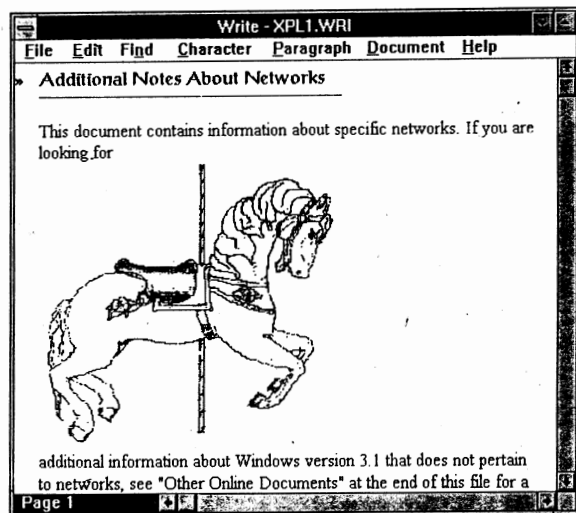


图2 书写器——Write

它同时也能够对图形进行简单处理,图2即为一图文混排实例。

1. 建立文档 初启 Write,首先应对 File 中 Print Setup 进行纸张大小、来源等选项设置。窗口工作区一闪烁光标为各种输入的开始插入点。当 Write 窗口中无正文时,紧随其后是文件尾标记,指示文档的结束处。建立文档时,键入的正文将从插入点开始填入 Write 窗口,到达行尾时,Write 将自动折行。若想开始一个新段落,可按回车键;若想在段落间留出一空白行,可按两次回车键;若键入过程有错,则可按 Backspace 键退回插入点并删除错误。

文档建立后,利用 File 中的 Save As... 菜单命令可对新文件存盘。若是对一已存过盘的文件进行修改后则只需用 Save 命令再次保存。

2. 格式化文档 文档建立后,需要对键入的文字编辑、格式化以满足排版需求。Write 软件要求用户在执行编辑命令前先进行文本块(字、字符、行、段或全文)选择。最常用的方法是按住鼠标键从选择光标起点拖到选择终点。由于窗口工作区文档左边的区域是选择区,故亦可利用下述方法在选择区中进行选择。

- 行 在行首单击鼠标进行单行选择,若想进行多行选择,按住鼠标器按钮并拖动指示针。
- 段落 在段落中任一 行首双击鼠标器按钮。
- 任意两点之间 在选择起点单击鼠标,然后在选择终点处同时按下 Shift 及鼠标键。
- 全文档 同时按下 Ctrl 及鼠标键。

经过选择的文本块都将变为相反颜色。此时可利用 Write 提供的格式化排版命令进行相应处理。

Character(字符)菜单提供字符格式化命令:字体(从 Fonts 中选择 Arial、Courier 等);字号(Reduce、Enlarge Font 或从 Fonts 中选择 size);印刷风格(Bold 黑体、Italic 斜体及 Underline 下划线等)。

Paragraph(段落)菜单提供段落格式化命令:自然段对齐风格(left 左对齐、Right 右对齐、Centered 居中、Justified 撑满);行间距(Single Space 空一行、1 1/2 space 空一行半、Double Space 空二行);缩进格式设定 Indents。段落格式化命令对插入光标当前所在段落或被选择的段起作用。

Document 菜单提供对整个文档外观控制命令:Header(页题头)、Footer(页脚注)及 Page Layout(页内布局)等。

格式化后的文本应及时用 Save 命令再次存盘。

3. 打印文档 对一已编辑排版的文档打印需选取 File 菜单中 Print 命令。若需要,在打印前还可进行设置纸张大小、打印方向等工作。

二 Paintbrush(画笔)

绘图也是常见的日常办公工作之一。Paintbrush 是一个用于产生和编辑图形图像的应用程序。它不但可生成文字标语、线条图或绘制复杂的艺术图案,甚至可以修改由扫描仪生成的图像。图 2 及图 3 中马匹即为 Paintbrush 完成。Paintbrush 一个显著特点就是与其它应用程序(例如 Write)密切配合,构造图文混排环境,给人以耳目一新之感。

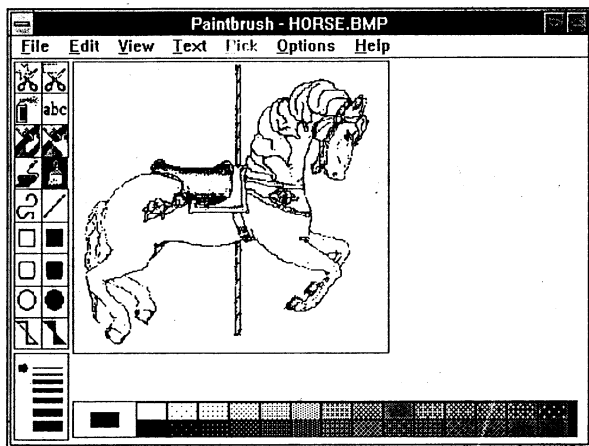


图 3 画笔——Paintbrush

如图所示,Paintbrush 应用程序窗口左上部提供 18 种绘图工具;左下部为一线条尺寸框,用于选择绘图时的线宽;窗口下部为一调色板,含有 28 种缺省颜色或图案。作图的主要工作区是屏幕右上部分。

1. 利用绘图工具作图。Paintbrush 作图实质是每个绘图工具分别作用后的集成。绘制图形时,一般先在调色板上以鼠标左、右键分别选择图形的前景及背景颜色,再从 File 中选择 New 命令。对于想获取的绘图工具,只需在工具肖像上单击鼠标键即可。

- (不规则)剪刀:用鼠标拖动形成一块不规则状区域,以便对区域内的内容作删除、拷贝和其它操作。
- (规则)剪刀:用拖动鼠标方式形成一规则状区域,进行如上的操作。
- 喷枪:产生雾状效果,可描绘云彩或阴影等。
- 文字工具:在图画中选定地方利用键盘键入文字。
- 颜色擦除器:拖动鼠标时可改变经过地方的前景色。
- 擦除器:拖动鼠标时可从图画中擦除某个物体。
- 着色滚筒:利用前景色填充一闭合区域。
- 刷子:手工绘制任意图案。
- 曲线:绘制任意光滑曲线。
- 直线:绘制直线。
- 方框:绘制空心矩形(正方形)框。
- 填充的方框:绘制用前景色填充的矩形(正方形)框。

- 圆角框:绘制空心圆角矩形(正方形)框。
- 填充的圆角框:绘制用前景色填充的圆角矩形(正方形)框。
- 圆/椭圆:绘制空心的圆或椭圆。
- 填充的圆/椭圆:绘制用前景色填充的圆或椭圆。
- 多边形:绘制闭合不规则的多边形。
- 填充的多边形:绘制用前景色填充的不规则多边形。

2. 图形编辑艺术化 Paintbrush 提供的编辑功能主要有:

- Edit 菜单中 cut/copy 命令可将剪刀剪下的画块剪贴/复制到剪贴板,再用 Edit 中 Paste 命令将该画块粘贴到绘图区的其他地方。(图 2 中的马匹即是在 Paintbrush 中完成后通过 copy/cut 命令拷贝或剪切到剪贴板中,再利用 Write 中 Edit 菜单中的 Paste 命令完成。)
- View 菜单中 Zoom In 命令可将一块选中图形放大至象素可编辑状态,再利用绘图工具对每一个象素进行颜色处理。Zoom Out 命令可恢复正常显示状态。
- Pick 菜单中提供一些图形艺术化效果命令。Flip Horizontal 可将用剪刀切下的画块左右翻转;Flip Vertical 使画块上下翻转;Inverse 使画块各种颜色变为对应的相反色;Shrink+Grow 使画块放大或缩小;Tilt 使画块倾斜。

三 便笺(Notepad)

Notepad 是一可以读写简单文本文件的编辑器。它的主要功能是用来记笔记、写备忘录和生成或修改在 MS-DOS 命令下执行的批处理文件(例如 AUTOEXEC.BAT),以及 Windows 的初始化文件如 WIN.INI,SYSTEM.INT。

但是 Notepad 应用程序处理的文件最大字符只能为 50000 个,因而对于要编辑超过该数目字符的文件,一般应选用 Write 或其它字处理程序。

四 记录器(Recorder)

记录器是 Windows 允许记录被称为宏(macro)的击键序列和鼠标操作的工具。宏的内容可以是一条或多条 Windows 或应用程序的命令,也可以是某些信息,如需反复多次输入到应用程序中的文字。而有些宏则同时具备上述两种内容。宏的重现可用记录宏时所定义的单个按键或组合键来完成。例如,在 Write 处理程序中我们要多次反复输入“中国科学院”这个词,不妨用记录器把它定义为宏,并赋之以组合键 Ctrl+C,以后在 Write 中每次欲键入“中国科学院”这词时只需按 Ctrl+C 键即可。

五 卡片盒(Cardfile)

Cardfile 是一个电子索引卡片系统,可用来对一组索引卡片自动排序。电话号码、通讯地址、图书目录等都是 Cardfile 经常处理的对象。图 4 为一通讯地址及电话号码卡片盒。

窗口中 View 菜单提供了卡片(Card)和列表(List)两种显示方式(图 4 为卡片显示方式)。利用 Card 菜单中的命令可实现:添加、删除和复制卡片工

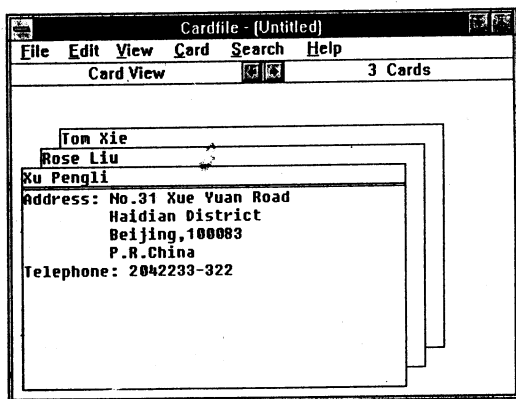


图 4 卡片盒—Cardfile

作。Search 菜单则提供了对整个卡片盒中卡片搜索的命令。

六 时钟(Clock)

时钟是 Windows 提供报告时间的工具(见图 5)。在多任务的 Windows 运行环境中常把 Clock 最小化为肖像,需要时亦能一目了然。Clock 在控制盒菜单中有一 Always On Top(永远在顶部)命令,若选择了此命令,即使在其它应用程序处于最大化运行状态时,Clock 亦是可见的。Windows 提供两种时钟显示方式(见图 5),左边是数字式,右边则为模拟显示方式。两种显示方式的切换由 Settings 菜单中 Digital 与 Analog 命令实现。当时钟显示为数字方式时可通过 Settings 菜单中 Set Font 命令改变时钟显示字体。

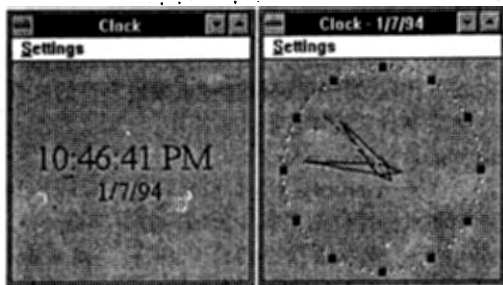


图 5 时钟—Clock

七 计算器(Calculator)

为了便于日常计算与统计,Windows 提供了两种计算器——标准计算器(见图 6)和科学计算器。前者允许进行简单一般运算,而后者则允许进行高级科学计算和统计计算。两者的切换是通过选择 View 菜单中的 Standard 和 Scientific 实现的。计算器的使用方法与日常生活中的计算器大体一致。

八 日历(Calendar)

日历是一个简单的日程安排程序(见图 7)。实际工作中,我们经常通过它完成每天工作备忘录和指定时间的约会安排等工作。Calendar 提供日历和月历两种显示方式,必要时还可以发声以提醒用户注意安排的约会时间很快就到。

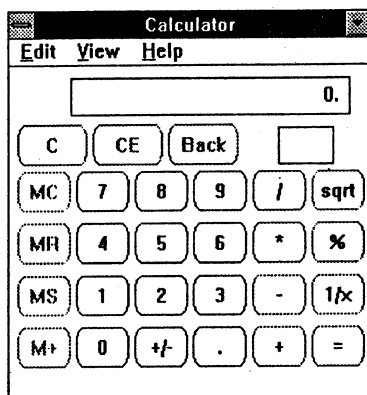


图 6 标准计算器

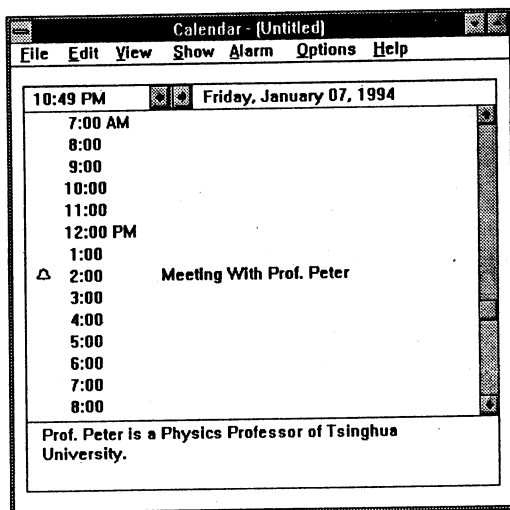


图 7 日历—Calendar

View 菜单提供了日、月历显示方式的转换命令。Show 菜单则提供查看今日、上月/日、下月/日或某一特定日期的命令。Alarm 的菜单命令用于在日历中设定闹钟。而 Options 菜单中 Mark 命令可为某一天做一特殊标记,以标识该天与众不同。Special Time 允许在日历中插入一个特殊的时间。Day Settings 用来规定日历中显示时间的间隔是 15、30 还是 60 分钟,12 还是 24 小时制,从几点开始显示等。

九 字符映射表—Character Map

字符映射表是 Windows 帮助用户在应用程序文本文件中输入特殊字符的一种工具,图 8 所示是 Times New Roman 字体中所能使用的特殊字母或符号。

窗口左上角下拉式列表框用于选择系统所允许使用的字体,列表框下面显示了所有所选字体允许使用的特殊字符及符号。一般有两种方式选定一特殊字符或符号:

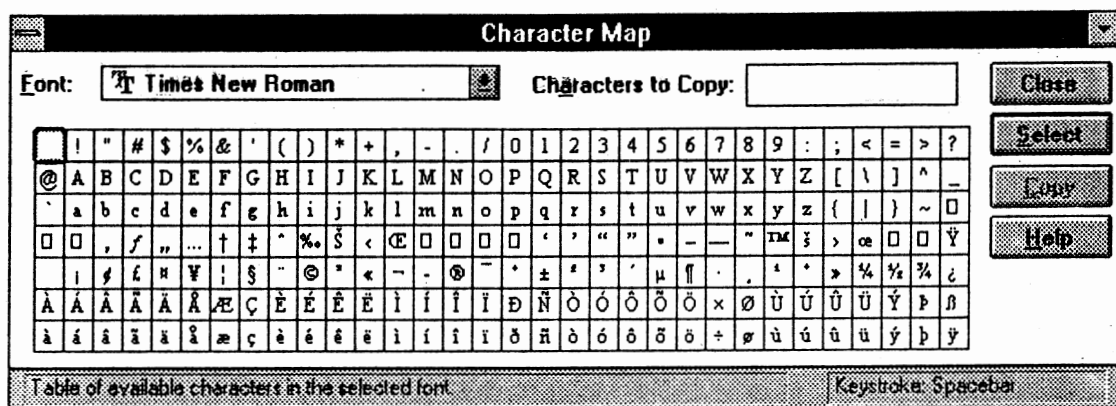


图 8 字符映射表

方法 1: • 利用鼠标单击想要的字符;

- 选择 Select 命令;
- 选择 Copy 命令;(此时窗口下面提示该字符已被拷贝到剪贴板中)
- 激活应用程序(如字处理 Write),选定特殊字符字体及在工作区中欲放置的字符位置后,选择 Edit 菜单中 Paste 命令。

方法 2: • 选中 Character Map 中欲采用的特殊符号并记下窗口右下角 Keystroke(即符号所对应的键盘输入码,如 Alt+0151)。

- 激活应用程序(如字处理软件 Word),选定字符字体及放置位置后,从键盘输入符号所对应的键盘输入码(如 Alt+0151)。

上面两种方法有两点请读者注意:1. 字体必须一致。从什么字体的字符映射表中取出的特殊符号在应用程序中只能在相同的字体中采用;2. 第二种方法在键入数字时(如 0174)应采用键盘上的数字小键盘。

在 Windows 桌面办公用具中,还有 Media Player(媒体播放器)、Terminal(终端仿真程序)等,限于篇幅,就不再介绍,关于 Object Packager 将在下期讨论。

至此,我们已对 Windows 3.1 桌面办公用具——Accessories 做了介绍。合理、高效、灵活地利用这些工具将对我们办公自动化工作起到积极的促进作用,提高到一个新的工作水平。

下期我们将讨论 Windows 3.1 中较深入的技术及使用技巧。

电子工业出版社软件部最新推出 系列电子词典软件

1. 英汉计算机词典——在计算机上直接翻译英文计算机书籍的理想工具

- 近 20000 条最新计算机技术词汇
- 在任何编辑器或其他状态下直接按热键进入词典查询

- 联想快速输入英文单词或词组
- 一词多个中文表达,选中的中文表达自动输入
- 适用于所有的中文 DOS(CCDOS,UCDOS 和 SUPER DOS)

2. 计算机屏幕提示联机速查词典——计算机初学者的必备工具软件

- 几乎 DOS 下可能出现的所有提示(近 4000 条)
- 在任何状态下按热键直接进入词典查询
- 联想快速输入英文提示
- 适用所有中文 DOS

3. DOS 命令联机速查词典——DOS 用户的必备工具软件

- 包括几乎所有 DOS 命令的所有可能用法(近 1000 条)

- 使用 DOS 命令时查阅最方便

- 联想快速输入 DOS 命令

- 查阅 DOS 命令的功能和格式

- 查阅 DOS 命令的各种使用示例

4. 其他词典正在开发中(欢迎广大读者提供建议和词典)

(上接第 56 页)

主要参数:

R_{ON}	30 Ω (最大 50 Ω)
R_{ON} 漂移	0.5%/°C
I_s (OFF)	0.1nA
I_D (OFF)	0.1nA
I_D (ON)	0.1nA
t_{ON}	50ns
I_{DD}	10mA
I_{SS}	6mA
消耗功率	240mW

工具软件 PC Tools 的应用

王路敬

(12) List(打印文件目录功能)

该功能打印列表的文件目录。

由于 List 功能是打印屏幕列表的文件,因而在打印前用 F8 键选择所需的列表,此时 F9 与回车键是无效的。

按下 F8 后,屏幕显示如下信息:

Enter the

:

Ext=[*]

path=A:*.*

Name and Extension correct as entered,(Y/N)? [Y]

F10=Blank the Name and Extension. Esc=Return.

上面对应的中文信息是:

路径=A:*.*

请输入要显示的目录清单的——关键字

请输入作为查找关键字的文件名和扩展名,以此关键字列出它们的目录清单。在某个位置上。如果不是一个确定的字符,则用“?”来代替;如果不是一个确定的字符串,则用“*”来代替。

Name=[*]

Ext=[*]

以上输入的文件名和扩展名是否符合要求,(Y/N)? [y]

F10=删除已输入的文件名和扩展名。Esc=退出。

[例]打印 C 盘上所有以字符“PC”开头的文件名
选择 C 盘根目录列表,按 F8 键并输入列表变量:

Name=[pc*]

Ext=[*]

回车后,屏幕仅显示以“PC”开头的文件及属性,此时键入“L”,则将显示的文件名、文件扩展名、文件长度、属性及文件建立的日期、时间打印输出。若在打印中按 Esc 键随时可中止打印。

(13) Sort(排序功能)

该功能对目录中的文件进行排序,用于排序的文件属性有:文件名、扩展名、文件长度、日期/时间。

注意,IBMBIOS.COM、IBMDOS.COM 不参加排序。

按“S”键后,屏幕提示:

Choose the sorting method desired. If it appears as desired, enter "U" to update the directory on the disk.

F7=by NAME F8=by EXT F9=by SIZE F10=by DATE/TIME ESC=return

中文信息的含义是:

请您选择目录的排序方法。如果选择的目录次序

已适用,键入“U”则按当前显示次序记入磁盘。

F7=按名称 F8=按扩展名 F9=按长度 F10=按日期/时间 ESC=返回

[例]将 B 盘根目录按文件长度排序

①在选择 B 盘后,按“S”键进入排序功能。

②键入 F9,即开始按文件长度排序。此时屏幕下端显示:

A=Ascending or D=Descending [A]

Press ESC to Exit

按“A”键由小到大排序,按“D”键由大到小排序

③按“U”键将排好的目录写入磁盘

(14) Help(帮助功能)

按下“H”键后,屏幕显示有关文件服务功能的清单,借助该清单可进行有关文件操作。

按“H”键后,屏幕显示的文件服务功能的有关信息,英汉对照如下:

Help for "File Functions" (see the manual for detailed explanations)

Press: "C" to COPY a file or files.

"M" to MOVE a file or files (deletes the SOURCE file).

"O" to COMPARE the contents of files.

"F" to FIND a character string in a file or files.

"R" to RENAME a file or files to another name.

"D" to DELETE a file or files.

"V" to VERIFY a file is readable or attempt repair of a file or files.

"E" to VIEW or EDIT (change the contents) of a file or files.

"A" to view or change a file or files ATTRIBUTES and date/time stamp.

"P" to PRINT the contents of a file or files

"W" to use the word processor on a file.

(press "Esc" to return or any other key to see more HELP information)

对应中文信息是:

按: "C"(COPY)复制一个或几个文件。

"M"(MOVE)移动一个或几个文件[并把源(Source)文件删除]。

"O"(COMPARE)比较几个文件的内容。

"F"(FIND)在一个或几个文件内寻找一个字符串。

"R"(RENAME)给一个或几个文件改变文件的名称。

"D"(DELETE)删除一个或几个文件。

"V"(VERIFY)检验一个或几个文件的可读

出性用以准备修复它们。

“E”(VIEW OR EDIT)显示并可直接编辑一个或几个文件的内容。

“A”(ATTRIBUTES)改变一个或几个文件的属性和日期/时间标记。

“P”(PRINT)打印一个或几个文件的内容。

“W”(WORD PROCESSOR)对一个进行字处理。

(键入“Esc”则结束帮助,按其它键则显示更多的帮助信息。

帮助功能第二屏信息是:

Help for "File Functions" (continued)

Press: "L" to print a directory LIST of the currently displayed directory.

"S" to SORT the currently displayed directory and optionally update it.

"F1" to UNselect all files you have selected.

"F2" to alternate between the 13 or 26 file directory list formats.

"F3" to display the "Disk and Special Functions" PC Tools menu.

"F8" to enter an argument to display files matching that argument.

"F9" to enter an argument that will SELECT files matching the argument.

"F10" to change to a different drive and /or path.

"Esc" to end PC Tools.

You may press the shift and "PrtSc" key on any screen to print that screen.

(press "Esc" to return or any other key to see Page 1 of HELP information)

上面的中文含义是:

对“文件服务功能的”说明(继续)

按:“L”(LIST)列表打印当前显示的目录。(请准备好打印机)

“S”(SORT)对当前显示的目录进行排序并将更新的目录存入磁盘。

“F1”取消您已选择(标定)的文件。

“F2”交替显示每屏 13 或 26 个文件的格式。

“F3”显示“磁盘管理与特殊功能”菜单。

“F8”则先输入需要的文件参数,再按此要求显示目录。

“F9”则先输入需要同时处理的文件参数,再按此要求显示目录。

“F10”改变当前的驱动器或改变路径。

“Esc”结束 PC Tools。

您在调入屏幕图形拷贝文件后,可按“Shift”+“PrtSc”键来打印显示的屏幕内容。

(键入“Esc”则结束帮助,按其他键则显示帮助说明的第 1 页).....

MS-DOS 5.0 中 DIR 的几项新增功能

贾纳豫 云南省玉溪师专微机室(653100)

DOS 中的命令 DIR(directory)是任何一个计算机用户必须掌握的一条内部命令,无论是硬盘还是软盘上任意目录下的任意文件,均可方便地使用 DIR 命令来查找,特别加上通配符“?”与“*”后,更加方便。

但 DIR 也有其不足之处,如:无法查到隐藏文件;某个盘上所有目录下的全部文件及文件数、共占字节数等无法查出(只有剩余空间字节数)。DOS 5.0 中 DIR 新增的几个功能弥补了这一不足。

格式:

DIR [drive:][path][filename][P][W][S][B][L][A[:属性]][O[:索引顺序]]

从格式中看出,除了以前 DOS 中用的/P 与/W 之外,又增加了几个可选项,它们的功能分别如下:

/S 显示当前目录和该目录的所有下一级子目录中的全部文件,及文件数、所占字节数等。如指定了文件名(可用通配符“?”、“*”),找到后就显示其目录、文件名等信息。未找到时显示 File not found。

/B 以“名·后缀”的格式每行显示一个文件,不

显示首尾的其它信息,也不显示文件数目(同/W 联用时,/W 的功能失效,但可以同/P 联用,仍满屏暂停)。

/L 以小写字母显示目录、文件名。开头、结尾等其它内容的显示形式不改变。

/A[:属性] 显示指定属性的文件及其所在目录。如果/A 后不选,则显示当前目录下的所有文件(包括系统隐藏文件等)。冒号属可选项,如指定了某个或多个属性(属性间不用空格),就显示有此属性或有这些属性的文件。可选用的参数如下:

H 隐藏文件	-H 非隐藏文件
S 系统文件	-S 非系统文件
D 目录	-D 不含目录
A 已登档文件	-A 未登档文件
R 只读文件	-R 读写文件

/O[:索引顺序] 按某顺序对文件进行索引。/O 后无可选项时,显示当前目录下全体目录和全体文件名。冒号属可选项。可选用的参数如下:

N 文件名、目录以字母顺序显示

-N 文件名、目录以字母逆序显示

E 扩展名以字母顺序显示

-E 扩展名以字母逆序显示

D 按建立文件的时间、日期早晚顺序显示

-D 按建立文件的时间、日期晚早顺序显示

S 按字节数大小的顺序显示(目录在前)

-S 按字节数小大的顺序显示(目录在后)

G 目录在前、文件在后显示

-G 文件在前、目录在后显示

如果用户想要查看 C 盘上的全部目录和文件名及数量时,可用

C>DIR /S[/P]↵

屏幕上就显出所有的目录及其所含文件,最后出现

文件总数 ××××个 所占字节数 ×××××
××bytes

剩余字节数 ××××××××bytes

此时的 DIR 功能完全可替代 TREE 的功能,因为 TREE /F 一是一个外部命令,使用上不如 DIR 方便;二是它只显示目录、文件,没有有关文件所占字节数、总字节数等信息。

如果用户想要知道 C 盘上有哪些隐藏文件时,可用

C>DIR/S /A:H↵

屏幕上就显出目录及其所含的隐藏文件名,最后有隐藏文件个数及所占字节数,如此一来,用户可不用 PC Tools 等工具软件,便能了解 C 盘上的全部隐藏文件。

由此可见,MS-DOS 5.0 中指令 DIR 的功能是很强的,只要你对某些可选项熟悉,就可给你的使用带来极大的方便。

Super-CCDOS 修改一则

邱建斌 浙江电子工业学校(310015)

基于 1990 年 1 月版的 Super-CCDOS V4.04 的 WPS V1.2 虽比不上 WPS V2.1,但在某些方面还有其自身的优点,如编辑时不必每次按键都要访问磁盘,从而使处理速度比 WPS V2.1 快多了,这对于使用速度较低的 PC、PC/XT 机用户来说很合胃口;更主要的是 WPS V1.2 因没有象 WPS V2.1 一样需要 4 兆多的大字库而使之在编辑时可以不用硬盘,这一点对举办 WPS 培训班非常有利:一是可以利用一些单位淘汰下来的 PC 机或没硬盘配置的 PC/XT 机,从而降低了成本,二是可以节省学员的费用,因为只要两张 360K 软盘就可使用。

这两张盘里的文件一般是这样装配的:A 盘除操作系统的三个模块文件以外,还有安装字库的 CHLIB.COM 文件、字库文件 CCLIBJ.DOT 和扩充汉字输入输出功能的 VDKEY.COM,再加一个自动批处理文件 AUTOEXEC.BAT,所剩空间已寥寥无几;B 盘有挂接拼音输入的文件 PY.COM 和挂接五笔字型输入的文件 WBX.COM 及 WPS 的三个文件 WPS.EXE、WPS1.OVL 和 WPS2.DVL,还有较多的可用空间可同时用来存储编辑好的文件,即做数据盘使用。这样 A 盘可以贴上写保护,B 盘就不行。

但由于现在计算机病毒泛滥成灾,而 B 盘里还有 .COM 和 .EXE 文件,因而 B 盘经常出现使用几次以后就不能再使用,又得重新拷贝的情况。这样既浪费了时间,又挫伤了学员的积极性,使学习效果大大降低。

如果要防止计算机病毒的侵染,则可以通过把上述的两张盘都贴上写保护而再加一张盘做数据盘的方法

解决。但这种方法还是不尽人意,笔者有一种方法既可防止病毒侵染又不需增加软盘。这种方法是通过修改 CHLIB.COM 使之装字库时能到 B 盘上寻找 CCLIBJ.DOT 文件以读取汉字点阵。这样就可以把 CHLIB.COM 与 CCLIBJ.DOT 放在不同盘上使用,也就可以把所有的(.EXE)和(.COM)文件放入 A 盘里,而 B 盘上只有一个 CCLIBJ.DOT 文件。这样 A 盘还是贴上写保护,而 B 盘的 CCLIBJ.DOT 是不受病毒感染的,并且也有足够多的可用空间来存储编辑好的文本。

对 CHLIB.COM 的修改如下:

先用 PC Tools 的 F10 选择有 CHLIB.COM 文件的目录进入文件服务功能菜单,再把光带移动到 CHLIB.COM 处,按 E(Edit/view)进入文件磁盘映象的编辑功能,并利用 F2 键把文件翻至相对扇区数为 00035 的地方,再按 F3 键进入可修改状态并把光标移到偏移量为 0336 处,再按 F1,使光标出现在右边区域,把 C:\CCLIB 中的第一个 C 改为 B,再按 F5,把修改过的内容存盘。接着可以直接利用 PC Tools 把各有关文件依上所述分别拷入 A 盘和 B 盘,再在 A 盘上贴上写保护就可使用。

值得一提的是,对 CHLIB.COM 的修改不影响修改前的文件装配方案(即 CHLIB.COM 与 CCLIBJ.DOT 同在 A 盘里的形式)的使用。另外,如果 WPS1.OVL 和 WPS2.OVL 同时在 B 盘上也有拷贝,那么虽使 B 盘空间减少了一些,但有时候使用起来会更方便。

支持多媒体应用的网络能力

周力丹 译

1. 引言

虽然许多因素正激励着 BISDN 的发展,但将多媒体纳入市场的期望相当强烈,因此,需要对尚不清楚的多媒体应用的网络需求进行了解。多媒体的功能在于它提高了通信的能力并且以一种或多种介质将信息传送到一个或多个用户,促进群体协作。于是,提出了以最适当的媒体即混合媒体进行通信的思想和概念。可利用的媒体的类型包括语音、立体声、文本、编码数据、光栅和矢量图形、静态图像和全动态视频。

本文按如下方式组织。第二节对多媒体应用进行分类。第三节介绍服务原语体系结构的概念,它是满足多媒体应用所具有的通信需求的机制。第四节指出支持多媒体应用所需求的网络能力,第五节提供一个管理多媒体网络性能的基本服务手段。

2. 多媒体应用

多媒体应用分类为:延迟通信、交互通信、制作以及信息传递和存储。

2.1 延迟通信

延迟通信是一种信息的单向传递,在传递过程中发送者和接收者在时间上不需要同步。发送者能够使用多媒体制作工具、单一媒体编辑器和超级编辑器来组织文件。然后,将完成的文件存贮起来。能够被编辑的延迟通信文件有两种类型:多媒体邮件和多媒体应答。

2.1.1 多媒体邮件

当发送者知道或估计接收者不在场,或当信息可以被延迟发送或不要求交互通信传递的情况下,多媒体邮件是很有用的。一个发送者建立文件、审阅并编辑这个文件,然后把它存入一个邮箱系统中。目前已实现了几个不同的多媒体邮件系统。

2.1.2 多媒体应答

多媒体应答允许利用编辑资源实时地产生多媒体信息或文件。这种应用不象多媒体邮件那么复杂庞大。在发送者想要公开与一些目标进行交互会话但又发现目标的当事人不确定时,这类应用是很有用的。

2.2 交互通信

交互通信的特征是在所有的与会者之间进行双向的、实时的信息交换。多媒体交互通信可分为会议和协商两类。

2.2.1 会议

会议的目标是以直接的和实时的方式交流信息。多媒体会议有三种不同的方式:电话、浏览和集会。

电话方式:多媒体会议电话的含义是两个或两个

以上的用户相互使用比较方便的媒体进行直接通信。例如,一位医生可以召集一个音频/视频电话会议与几位医学专家进行商讨。

浏览方式:浏览是以人们交互活动的模式为基础,人们随机搜索一些局部区域查找某个要与之交流的目标。“漫游者”是一个 Bellcore 的模型系统,它使用户沿着一条设置在多媒体终端上的虚拟通道漫游。当用户沿着虚拟通道经过一个办公室,他们可以看见办公室里是否有人,决定是否可以开始通信。

集会方式:集会是社会团体中人们开会的方式为基础。假设一个虚拟会议场面,通过多媒体终端,用户能够进入一个虚拟的会议房间并且与其他已经同样进入这种会场的人员进行交谈。

2.2.2 协商合作

协商合作的目的是完成交换辅助信息的某些任务,通常是间接的。协商通信的模式是一种物理上分别独立的个体共享分布应用以完成某项工作。这一处理类型称为 Computer-Supported Cooperative Work (CSCW)。

2.3 多媒体制作

多媒体制作用于编制和修改多媒体文件。其基本组成部分是多媒体编辑器,它控制一系列单个媒体编辑器和超级编辑器中进行的存取访问和相互配合。单个媒体编辑器用于建立和修改不同内容的文件,这些文件的内容是单个媒体目标,例如文本和视频。随后,利用超级编辑器将文件内容组织成页,并且定义每页内容之间的相互关系(同步)。

2.4 信息传递和存储

信息传递可使用户访问远地多媒体信息数据库。信息存储则允许用户将多媒体信息存储在由网络或服务器提供的设备中。有三种传递和存储应用:检索应用、分布应用和存储应用。

2.4.1 检索应用

检索应用可使用户就一段特定的信息做直接的查询。下面讨论两个典型的检索系统:多媒体数据库和电子知识系统。

多媒体数据库:多媒体数据库能够用一种综合的方法存储、索引和检索文本、音频、视频和图象。这些多媒体数据库可以以分级或同级的方式彼此互相交叉参照,把一个数据库作为从许多其它数据库进行信息采集和信息综合的集中点。

电子知识系统:电子知识系统可使用户经过一个标准的用户接口从广泛的各式各样的信号源中访问和

操作多媒体信息。例如一个系统允许用户连接多个数据库并且对它们进行存取访问,好象它们是一个单独的数据库。

2.4.2 分布应用

分布应用依赖于从某些信息源点向用户听众发送的单向的多媒体信息流。信息接收用户对信息源有一个低带宽的上溯连接用于发送信令,例如:启动、停止、暂停、倒转、快速向前、慢速搜索扫描等。

2.4.3 存储应用

可能有这样的情况,当用户要存储多媒体信息而在本地又没有适当的存储资源,在这种情况下,存储设备可以利用网络或服务器对文件进行临时存储。

3. 服务原语体系结构

服务原语体系结构根本的前提是硬件资源、虚拟资源、算法、服务和单元单元的抽象等效。硬件资源是执行有用功能的物理实体,例如一个视频光盘系统;虚拟资源是实现硬件资源的软件,例如一个执行视频编码/译码的程序;算法是软件过程,例如调用进程;服务的恰当定义就是网络对最终用户的特殊通信需求提供支持的全部功能的集合,例如多兆位数据交换服务(SWDS)。服务单元是专用的网络功能,它比资源和算法更复杂,但不如服务复杂。就此而论,与所有这些概念等价的实体就称为服务原语。

做这种等效的用途是使网络管理系统能用统一的方式联络和管理这些实体。其中,详述了一种特殊的结构方法,详述在网络内如何对服务描述进行模拟和通信,如何将服务原语组合成为服务,如何对此进行管理以及以后怎样取消。

通过服务原语体系结构提供的灵活性,可使BISDN去处理潜在的组合中的复杂性服务。BISDN允许用户以建立全新的服务方式合并或连接服务器与/或网络能力。此外,BISDN必须满足未来的服务要求,满足那些到目前为止还不能预见的应用。

4. 支持多媒体应用的网络能力

这一节描述在复合媒体同步、多媒体数据结构操作、信息存储和检索、网络控制、多路终点联系和目录信息诸方面BISDN的专用多媒体能力。

4.1 复合媒体同步

多媒体应用应允许用户改变事件的顺序并且修改多媒体信息的表现。为了实现这些功能,必须使不同的媒体彼此相互独立。以这种方式,可以随机访问不同的媒体,使用最适合的通信协议对每种可用的媒体进行处理。多媒体应用通过提供的综合控制能力掩盖了多媒体与用户之间的距离。

为了定义不同媒体之间的共原性和相互关系,多媒体应用应允许用户规定不同媒体如何实现彼此之间(复合媒体同步)的同步。一旦所要求的同步被定义了,网络就应该保证在终端接口上保持已建立的同步。也就是说,必须克服在分别载有不同媒体的多路相关的通信通道中存在的差分延时。网络通过向终端传递附加的同步信息实现这一功能,这些终端负责恢复复合

媒体同步信号。

4.2 多媒体数据结构操作

在BISDN中有不同类型的媒体,而每一种媒体具有多种可能的编码形式。这种情况,再结合与网络连接的用户的不同终端类型以及各种与BISDN互相配合的网络的情况,要求设计内部多媒体数据格式以便于对信息的有效操作、整理汇集与传输。出于这一目的,将多媒体数据结构操作划分为两类:格式操作和内容操作。

4.2.1 格式操作

格式操作仅仅适用于信息的物理结构而不涉及信息的内容。格式操作包括代码转换、媒体变换和表现形式转换,但不仅限于这些功能。

代码转换:每当两个或多个沿用不同编码标准的终端寻求建立一种联系的时候就需要进行代码转换。那种在A-规则和 μ -规则之间进行代码转换的PCM编码规则是在语音电话网中已使用的性能。

媒体转换:每当具有不同媒体处理能力的终端之间企图建立联系时就需要媒体转换。当一个采用先进技术装备的多媒体终端要与一个不支持某些媒体的终端进行通信时就可能提出媒体转换的要求。媒体转换对于禁止用户使用权也是很有用的。

表现形式转换:每当一个多媒体信息流或文件遵循的表现标准被改变,为了使多媒体终端用户接口更方便地表现,就需要媒体形式转换。例如,在两个不同的窗口操作系统标准之间进行的转换。

4.2.2 内容操作

内容操作着重于处理多媒体数据流或文件的信息内容。它的目标是保存宽带网内的资源而不改变正被传递的信息的含义。内容操作能力可分为两类:信息压缩和信息筛选。

信息压缩:信息压缩涉及对数据流采用特殊的编码技术以降低频带宽度。采用压缩技术对最终用户来说效果是否显著取决于多种因素,如原始编码规则、压缩技术以及被压缩媒体的特性。

信息筛选:信息筛选涉及对信息语义内容的分析。只选择满足用户方面需求的标准信息。例如:Passive Information Grazing System(PIGS)是Bellcore的一个标准模型。在PIGS中,信息筛选可以在靠近信息源处,在网络内完成,由终端用户完成,或者由上述某几种的组合来完成。

4.3 信息存储和检索

多媒体邮件和应答应用两者都要求网络象自动服务发售机那样使“邮箱”保持与每一个用户的联系。应答应用还包括一个输出邮箱,该邮箱包含回答呼叫用户的信息。多媒体传输应用可要求网络保持一个全局索引指针针对多媒体数据库进行网内网外的查询。此外,所有这些多媒体应用必须要求有访问存储的信息或文件的保密限制。

4.4 网络控制

资源的控制和管理是网络所必须具有的基本能力

之一。网络控制的现行方法必须采取重大步骤才能满足未来多媒体应用的要求。

4.4.1 呼叫模块

为了对复杂的资源配置进行控制,必须定义合适的呼叫模块。呼叫模块确定在一个呼叫的前后关系中能够被操纵的目标,在这些目标上可能进行的操作以及这些目标的关系或目标之间可能的交互作用。有关多媒体应用的呼叫模型应包括多通道、多用户管理和控制,以及呼叫和连接控制的识别。网络内现行的多用户通信处理受到限制并且相对固定。此外,目前的电话系统中,在一个呼叫和用于支持呼叫的物理连接之间是一对一的关系。预计在多媒体服务中,一个呼叫内可以有多个连接,把每一个连接看做是一个能够被操纵的目标,也即当呼叫时,可以将一个个连接加上或减去。这些思想引出了将呼叫目标从包含它的连接中独立出来的概念。为了有效地管理多路连接,把一个呼叫内的彼此间的必然关系联合起来也许是很有用的。这种成组的连接可被看作是一个呼叫内的独立可控的实体。

4.4.2 动态的带宽分配

在普通的电话网络中,通信通道以固定的带宽增量被分配和管理。将来,一些多媒体应用将允许插入可变带宽的媒体,网络控制必须能有效地处理带宽利用的动态特性。网络将在两种场合受到影响。第一,为了有效地容纳不同的带宽对网络的呼叫,需要一种入口控制方案。第二,在对所有现存的呼叫提供合理的服务质量的同时,为了管理带宽使用中出现的起伏,需要一个拥塞控制系统。这些方案目前正在研究之中。

4.4.3 信令

为了支持多媒体应用,信令必须提供有关呼叫和有关连接的信息之间的差别。此外,网络内的信号实体怎样在用户的终端设备上表现出来,这些对用户来说是有用的。另一方面,信令是对一个呼叫的服务质量(QOS)的建立和维护。QOS与多媒体有关的一些服务质量参数包括带宽、信令响应延时、传送延时灵敏度、抖动灵敏度以及信息损失灵敏度。

4.4.4 服务原语管理

如前所述的一个服务原语体系结构被看作是与多媒体应用相关的进行复杂寻址的有效方式。服务原语的管理(例如组合、译码、分配)属于网络控制。为了描述服务原语,规定对已给定的服务的服务原语和控制流程使用服务正本。这些服务正本的语法与未来网络的信令信息相关。

4.5 多路终端桥接

在多媒体应用中,为了使在目标点能够综合表现来自不同地点的资源信息,就需要一个桥。就灵活性来讲,桥能够动态地增加/减少与会成员,动态地离开或加入会议,定义一个会议的表现形式(表现控制)以及将控制转移到其它会议。桥还可为与会者建立分会议。分会议是由原会议的部分成员组成,这些成员想要进行个人之间的信息交流。为了支持分会议,要求有一个

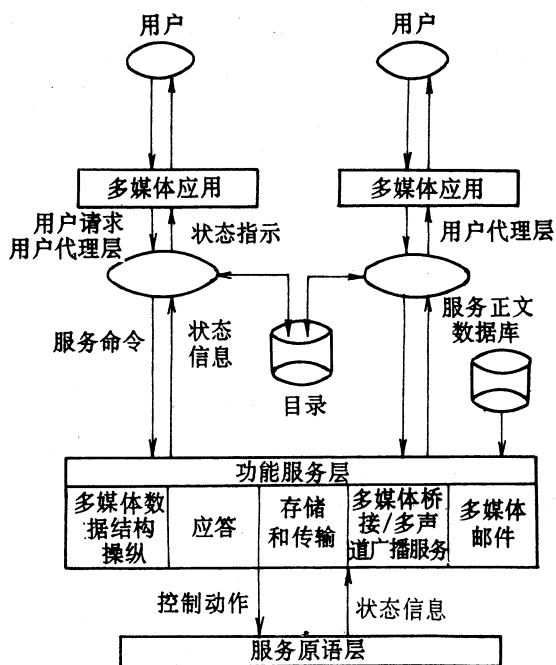
音频窗口。在这种情况下,分会议中的与会成员除了有主会议提供的音频通道外还有他们自己专用的声音通道。

4.6 目录信息

电信网络的目标之一就是提供多方面的连通性。为了实现这一目标要求有目录信息。目录信息向用户提供访问权利、资源以及服务。访问权利不仅包括在哪里查找用户、资源或服务,而且还包括怎样访问它们。根据这个定义,目录信息应包括用户姓名地址映象、用户(潜在)能力信息、资源/服务的地址映象、资源/服务能力信息、鉴别以及用户提问档。

5. 管理多媒体网络性能的基本服务方法

为了管理在第四章中描述的网络性能,建议一种基本服务方法。该方法是在网络内采用一种控制等级,在控制等级的最低层是服务原语层,第二层是处理分配、汇集、控制以及服务原语定位的功能服务层。最高层是处理用户服务要求以及协调功能服务的用户代理层。下图说明了这些概念。



图

5.1 用户代理层

用户代理层表示一个简易的、一致的用户应用接口。他们掩盖了网络对用户的复杂感。在我们的模型中使用的用户代理层与在 CCITT X.400 系列建议对信息处理系统中定义的用户代理相似。

每一个 BISDN 用户都通知他们自己的用户代理请求网络服务。这些代理确定实现用户请求所要求的网络功能,并向相应的功能服务组发出服务命令。用户 (下转第 54 页)

用内部电话网组建微机局域网络系统(LAN)

汤忠庆 湖州市自来水公司(313000)

随着微电子技术、网络技术、信息包传送技术和个人计算机技术的成熟和推进,以及对资源共享需求的增大,局域网络系统(LAN)技术在近几年得到了蓬勃的发展和广泛的应用。

局域网络系统在自己的发展过程中,形成了区别于远程网络的构成技术,包括体系结构、网络访问控制方法及网络拓扑结构等。目前局域网络的品种繁多、各有特色。但任何一种局域网中的计算机或计算机外部设备,都要通过某种网络接口部件来实现相互连接、信息传输和资源共享。组成一个局域网络系统的硬设备是通信介质及其附属设备、网络接口机和网络连接器、网络服务器等三部分。

本文主要涉及局域网的通信介质及网络接口机这两部分。常用的通信介质是架空明线、双绞线、电缆线、微波线路和卫星线路等。对于大多数单位来说,单独架设架空明线,既不经济也不实际,因为架线涉及到

许多方面,用无线通信是一个解决问题的办法,但也存在一个棘手的问题,需向无线电管理委员会申请通信频道,在空间资源日益紧张的今天,频率成了宝贵的财富,办这手续非常麻烦,而且需要不少运行费用。针对我国许多大中型企业具有内部电话网的特点,一个既经济又实用的方法就是利用内部电话网组建微机局域网络系统。显然这是最佳方案,但是具体实施还有许多实际问题,特别是要从技术解决如下问题。

利用内部电话网组建微机局域网络系统,首先必须达成这样一种协议:数据通信的优先级要高于电话通信的优先级。只有这样在电话通话时,如要数据通信,马上切断电话线路,交给数字设备传送数据用,数据传输完毕,电话线路又接通,其间隔约1~2秒,可以照常通话;反过来,在数据通信时,该路通话不能接通,待过一两秒后数据传输完了才能通话。按照此要求,笔者设计了电路(见图1),圆满地解决了这个问题。

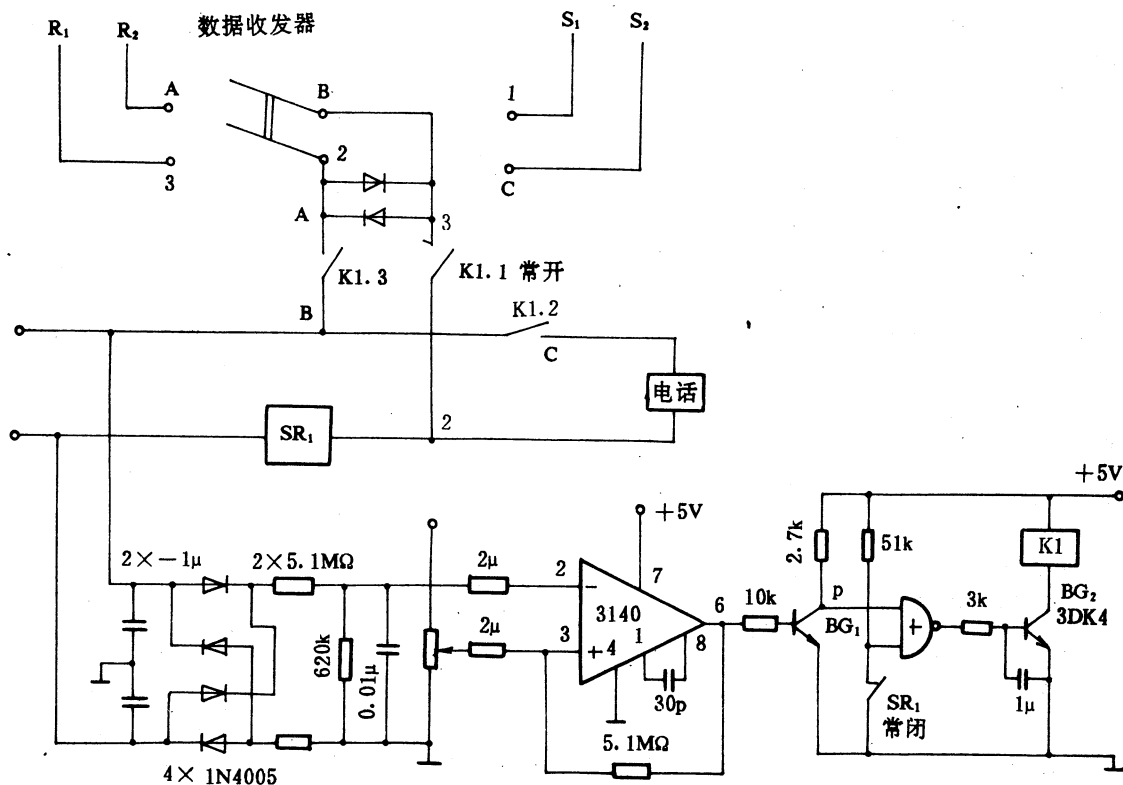


图 1

下面把电路原理简述如下:线路中 P 点电压和电话线两端电压存在如下关系(见图 2)。

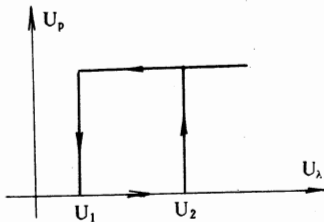


图 2

上图为滞环特性,经过实验整定到 $U_1 = 3V$, $U_2 = 17V$, 这样当电话线的电压超过 17V 时,继电器 K_1 不动作,当电话线间电压小于 3V 时,继电器 K_1 动作,切断与电话机的接线和终端机相接,当电压在 3V—17V 之间时, K_1 触点保持以前的状态。

图 1 原理是这样的,正常情况时,外线有 24V 电压,经过线路以后,仍有近 24V 电压,经过 2 个 5.1M 高阻以后,送入 3140,显然 3140 的输出端是低电平,这样三极管 BG_1 截止,集电极 P 高电平,经或非门以后, BG_2 三极管的基极也为低电平,因此 BG_2 也截止, K_1 继电器的线圈没有电流流过,它的触点不动作;当

外线的电压降到 3V 以下时, 3140 输出高电平,这样 BG_1 导通,因此 P 点为低电平。因为 SR_1 的触点为常闭,所以经或非门以后,输出端为高电平,因此 BG_2 三极管也导通,继电器 K_1 的线圈有电流流过,因此与电话线切断做好数据通信的准备,外线与终端机接通。当通信完成以后,中继机端又将外线接 24V 电压,又使 BG_2 三极管截止, K_1 线圈中无电流流过,因此各触点处于常态,即外线与电话机连接,终端机与外线断开独立工作。

图中的 SR_1 是电话继电器,实际上是电流继电器,它是检测外线是否有电流流过的。因为正常情况时,不打电话也不通信时,外线中无电流流过,只有 24V 电压,因此电流继电器不动作,当拿起电话时,有电流流过, SR_1 常闭触点动作,使或非门输入端有一端为高电平,因此 BG_2 三极管截止, K_1 继电器触点不动作,保持常态。

从以上分析可见, SR_1 电话继电器是为了使通信切换开关可靠工作而设置的,即在打电话时,外线电压降得再低,照样不会使 K_1 动作。

本系统已经在实践中得到了应用,解决了通话和数据通信的难题,是一个经济实用的优选方案。在微机日益普及的今天,微机联网已是大势所趋,把这方案推荐给同行,相信会给同行一点启发和帮助的。

也谈怎样使用 SMARTDRV.SYS 加速 WPS

韦韬 咸阳 4400 厂玻璃分厂 102 车间研磨组

贵刊在 93 年 7 期和 94 年 1 期上都提到了用 SMARTDRV.SYS 来加速 WPS 的方法,我也来说一下我遇到的问题,同电脑玩家们探讨。

原先我用的是 DOS 5,大家都知道, WPS 2.2 和 SPDOS 5.21 与 DOS 5 有中断上的冲突,后来我改用了 Liuph DOS 5.10 (SPDOS 5.10 的改版,可用于 DOS 5、DOS 6) 和 WPS 2.1,不知是改版的原因,还是 SPDOS 版本较早的缘故, SPLIB 无法将字库读入扩展内存 XMS,于是我就用了 SMARTDRV.SYS。当时 CONFIG.SYS 是这样编的:

```
DEVICE=C:\DOS\HIMEM.SYS (建立 XMS)
DEVICE=C:\DOS\EMM386.EXE NOEMS (建立 UMB)
DOS=HIGH,UMB
DEVICEHIGH=C:\DOS\SMARTDRV.SYS 512 256
(建立高速读缓存 512KB)
.....
如不用上位内存(UMB),可:
DEVICE=C:\DOS\HIMEM.SYS
DOS=HIGH
DEVICE=C:\DOS\SMARTDRV.SYS 512 256
.....
```

这样设定后, WPS 运行时将字库读入 XMS 中的

高速缓存中,既节省了基本内存,又不影响运行速度。但是,当我将机器升级至 DOS 6 时却出现了问题,软件中的新版 SMARTDRV.EXE (用于 AUTOEXEC.BAT 中,当有 UMB 时,用 LOADHIGH 或 LH 加载,也可在 DOS 提示符下加载,功能更强)用于 WPS 运行速度大幅下降,频繁读盘,这一现象还请指点迷津。在不得已的情况下,我只好重新使用 DOS 5 的 SMARTDRV.SYS。具体方法是先在提示符下输入:

```
C:\>SETVER SMARTDRV.SYS 5.00
```

上面这一步是先设定 SMARTDRV.SYS 的版本号,以免出现 DOS 版本号错误。然后这样编制 CONFIG.SYS 文件:

```
DEVICE=C:\DOS\HIMEM.SYS
DEVICE=C:\DOS\EMM386.EXE NOEMS
DOS=HIGH,UMB
DEVICEHIGH=C:\DOS\SETVER.EXE
DEVICEHIGH=C:\DOS\SMARTDRV.SYS 512 256
.....
```

这下 WPS 又在与 DOS 5 下时一样好用了。当然,换新版的 WPS 3.0 和 SPDOS 6.0 是更加好的、一劳永逸的办法。我用的是 386SX 机, 2M 内存。

实时系统时钟减慢的分析及解决

李开新 山东济宁煤气公司微机室(272137)

我们在 AT 机上用 MS-C(V5.0)语言成功地开发了煤气通信遥测系统,取得了良好的社会效益,但在运行中发现,该实时软件的时钟每天在以 10 分钟左右的误差减慢,如果不复位机器,误差将积累下去;退出应用程序,用 TIME 命令可见时钟减慢,一经复位机器,时钟即恢复正常。

经分析发现,系统实时时钟正常,系统时间减慢是由于串行通信模块中频繁的中断屏蔽,影响了系统时钟计时造成的,原来程序中用 `_dos_gettime()` 函数即相当于 DOS 系统调用 `0x2c` 读取当前时间,也是不准确的。

当然,解决这一问题的根本途径是重写通信模块,但原模块用宏汇编语言写成,重写不仅工作量大,而且可能影响已实现的功能。

我们最初采用 INT 1A 调用直接读实时时钟的方法,虽然解决了应用的问题,但如果退回 DOS,用 TIME 命令显示的系统时间仍然减慢。

经过比较,我们最后采取了直接从端口读实时时钟,定时进行系统时钟校时的方法,不仅较好地解决了应用问题,而且不影响系统时钟。

现在把我们的实现方法附后,供搞实时系统的同行参考,如有欠妥之处,或有更好的解决方法,敬请指正。

```
#include <dos.h>
#include <stdio.h>
#include <string.h>
#include <math.h>
#include <conio.h>
#define RTC_PORT_W 0x70
#define RTC_PORT_R 0x71

set_clock()
{
    char trans[4];
    struct dosdate_t date;
```

```
struct dostime_t time;
int second;
int minute;
int hour;
static int fg;
outp(RTC_PORT_W, 0x00);
second = inp(RTC_PORT_R);
outp(RTC_PORT_W, 0x02);
minute = inp(RTC_PORT_R);
outp(RTC_PORT_W, 0x04);
hour = inp(RTC_PORT_R);
sprintf(trans, "%x", hour);
hour = atoi(trans);
sprintf(trans, "%x", minute);
minute = atoi(trans);
sprintf(trans, "%x", second);
second = atoi(trans);
if (minute != 59) fg = 1;
if (minute == 59 && fg == 1)
{
    fg = 0;
    time.hour = hour;
    time.minute = minute;
    time.second = second;
    _dos_settime(&time);
}
```

display_clock()

```
{
    char date_time[14];
    - settextcolor(12);
    - strdate(date_time);
    - settextposition(23, 4);
    - outtext(date_time);
    - strtime(date_time);
    - settextposition(23, 14);
}
```

(上接第 51 页)

代理的特点是它们具有处理复杂的用户需求的能力,这些请求是关于不同网络性能相互协调的要求。

5.2 功能服务层

功能服务层管理服务原语并且对已确定的网络功能进行组织。功能服务将用户代理的命令翻译成识别、分配、建立、操纵和重定位服务原语等控制动作。一个完整的进程是一个服务正文的运行,服务正文中包含

操纵特定命令的控制动作的指令和名称,来自用户代理命令的参数作为数据输入到正文处理器。服务正文可被存储在与功能服务相关的数据库中,并且根据他们操纵的命令类型做索引。多媒体特殊的功能服务类型包括多媒体数据结构、操纵服务、多媒体桥接/多声道广播服务、存储和检索服务、应答服务和多媒体邮件服务。

周力丹译自“90 GLOBECOM”Vol. 1 PP322—326

三角函数波形生成 IC—AD639 及用法

李兰友

(接上期)

四、使用 AD639 构成的正弦波发生器

1. 指标

输出频率 DC~500kHz

失真率 0.1%以下

输出电压 最大 20Vp-p, 可调

输出阻抗 600Ω

2. 电路

使用 AD639 构成的正弦波发生器线路如图 11 所示。电路采用三角波转换为正弦波的方式。

本电路主要由三角波生成电路和三角波——正弦波转换电路两部分组成, 另外还有电源电路和基准电压生成电路。

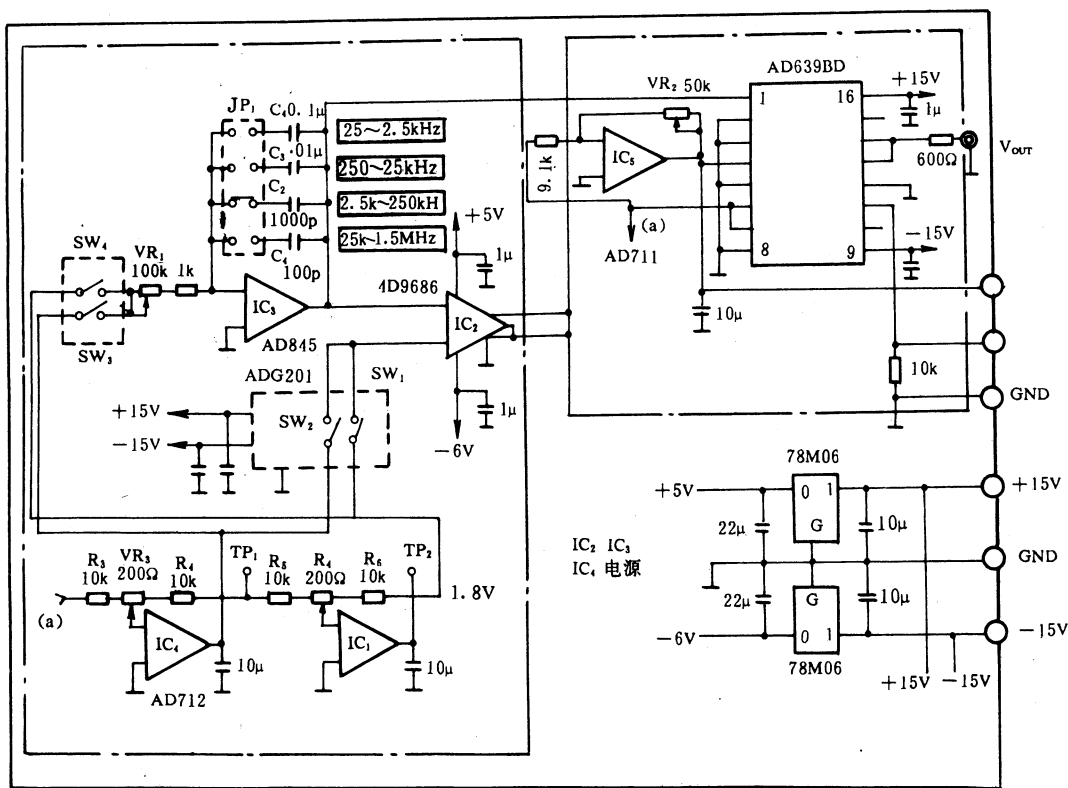


图 11 正弦波发生器电路

(1) 三角波生成电路

AD639 无在片三角波生成电路, 因此, 需在外部做一个三角波发生器。如图 11 所示的那样, 三角波发生器由积分器 IC₃ (AD845) 和高速比较器 IC₂ (AD9686BQ) 组成。模拟开关采用高速开关 ADG201HS。IC₄ (AD712) 为基准电压电路, 生成 +1.8V 和 -1.8V 的基准电压。

工作过程简述如下:

设最初比较器输出为 Q="L", \bar{Q} ="H"。这样, 模

拟开关 ADG201HS 的 SW₁、SW₃ 闭合, 1.8V 的电压加到比较器的反相输入端并做为比较器的基准电压。SW₃ 闭合时, 积分器 AD845 输入 -1.8V 的电压, 积分器输出开始上升, 相当于图 12 所示的 T₁ 期间。

积分器输出上升到 1.8V, 即与基准电压相等时, 比较器的输出 Q 变为 "H", \bar{Q} 变为 "L", 开关 SW₂ 和 SW₄ 闭合。SW₂ 闭合使比较器的基准电压变为 -1.8V, SW₄ 闭合, 比较器输入为 +1.8V, 积分器输出减小, 相当于图 12 所示的 T₂ 期间。

电路中 $C_1 \sim C_4$ 为积分电容, 电位器 VR_1 用于频率微调。电位器 VR_3 和 VR_4 为基本电源回路的调整元件, $R_3 \sim R_6$ 使用精度 1% 的金属膜电阻。

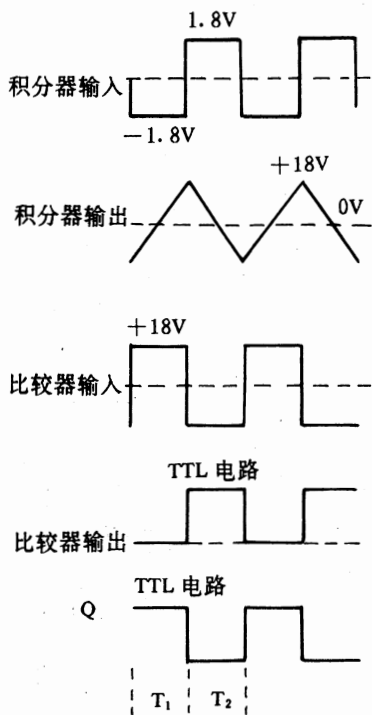


图 12 三角波发生器工作原理

(2) 三角波——正弦波转换电路

本部分电路中其他接法与图 2 相同, 不同点为:

① U_p 空置, 由 U_1 和 U_2 的输入电压控制正弦波输出的振幅。

输出电压调整由运放 IC_5 完成, VR_2 用于微调, 使输出电压 V_{out} 可在 $10mV \sim 10V_{opp}$ 内可变。

② GT 使用外部输入控制。

(3) 电路调整

① 将 JP1 接通 C_3 , 调整 VR_1 , 使输出频率在 1KHz;

② 调整 VR_2 , 使输出 V_{out} 为 $20 V_{P-P}$ 。

③ 调 VR_3 、 VR_4 , 使 TP1 和 TP2 电压为 1.8V。

3. 元件说明:

(1) 比较器 AD9686

AD9686 引脚如图 13 所示。

AD9686 主要参数:

输入特性

输入失调电压	1.0mV(25°C)
输入失调漂移	10 μ V/°C
输入偏移电流	4 μ A
输入失调电流	0.4 μ A(25°C)
输入阻抗	100k Ω
输入电容	3pF(25°C)

输入电压范围 -3.3~4.5V

共模抑制比 85dB

启动特性

逻辑“1”电压 2.0V

逻辑“0”电压 0.8V

逻辑“1”电流 100 μ A

逻辑“0”电流 100 μ A

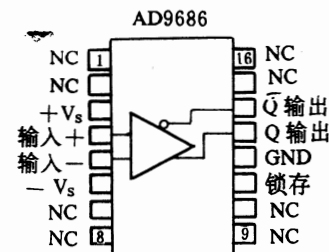


图 13 AD9686 引脚

数字输出

逻辑“1”电压 3.5V

逻辑“0”电压 0.3V

开关特性(传递延迟时间)

输入——高输出 7ns(25°C)

输入——低输出 7ns(25°C)

锁存启动——H 输出 7ns(25°C)

锁存启动——L 输出 7ns(25°C)

最小建立时间 1 μ s

最小保持时间 1ns

电源

正电源电流(5.0V) 30mA

负电源电流(-6.0V) 26mA

电源电压变动抑制比 65dB

(2) 开关 ADG201HS

ADG201HS 引脚如图 14 所示。输入 L 时, 开关置 ON; 输入 H 时, 开关置 OFF。

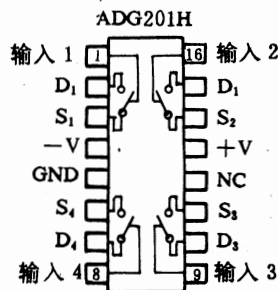


图 14

(下转第 45 页)