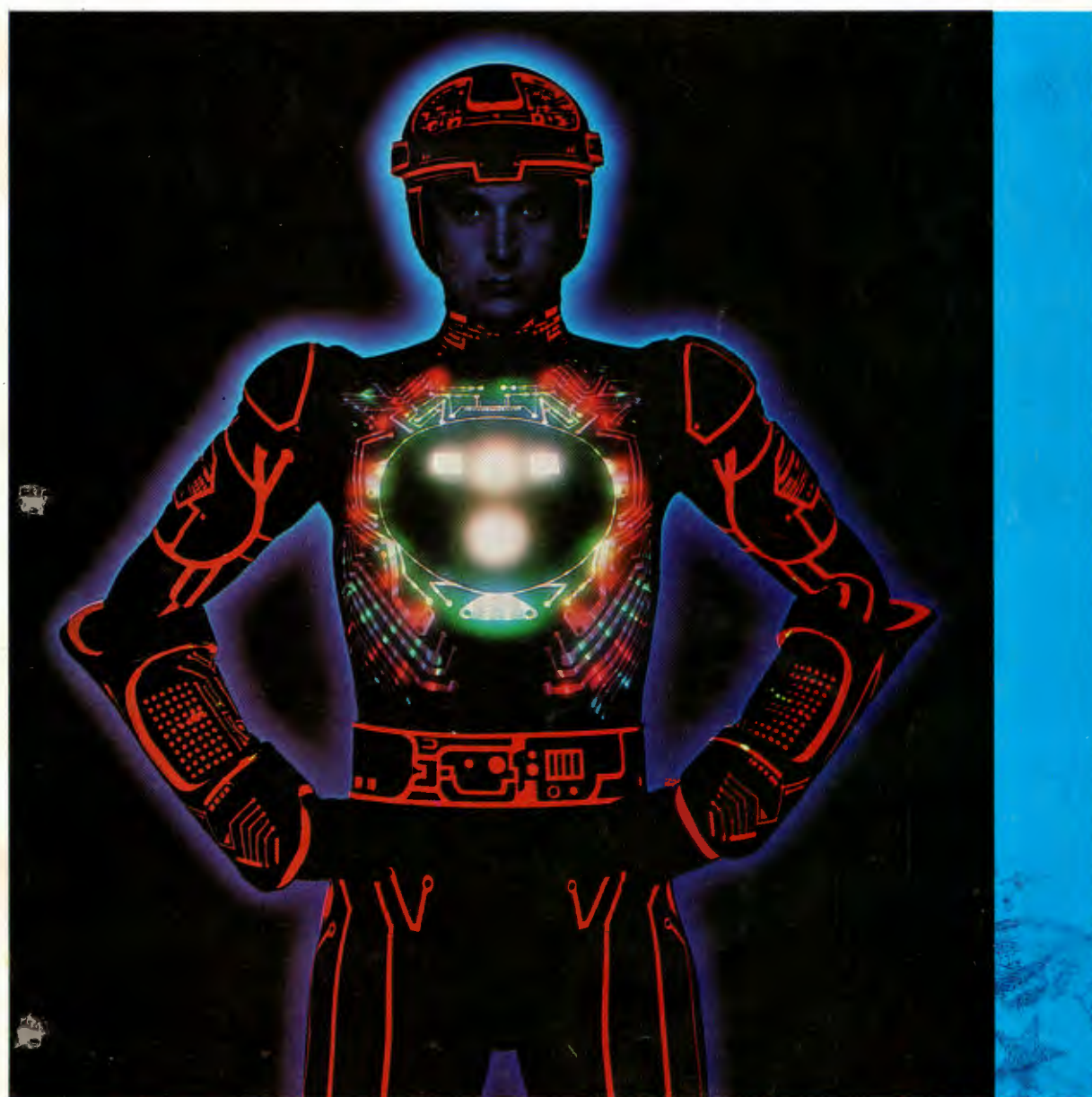


E&C
1992
 ●一九九二年 ●总期第89期
8

電子

ISSN 1000-1077
479
 479

與電腦



● **ELECTRONICS AND COMPUTERS** ●



武汉创意电子研究所隆重推出

创意

CYSC13-2MCS-51, 8098单片单板机

单片机应用文集

何立民 主编

SCB-2型MCS-51, 8098单片单板机

通信工程系 吴 微 马国敏 孔明忠 等编写

摘 要

本文介绍了一种新型单片机，结构合理，成本低廉，应用广泛。该单片机采用MCS-51内核，具有8K字节程序存储器，2K字节数据存储器，16个I/O口，16个中断源，16个定时器/计数器，16个并行I/O口，16个串行I/O口，16个外部中断源，16个外部设备接口。该单片机具有MCS-51和8098两种模式，可在两种模式下工作。该单片机具有MCS-51和8098两种模式，可在两种模式下工作。

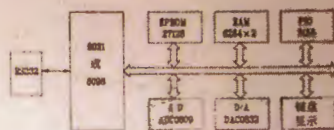


图1 SCB-2型单片单板机结构图

A/D；外部配的A/D是ADC0809，它是8位A/D。若是8098单片机，则片内还有4路10位A/D。

D/A。外部配的D/A是DAC0832，它是8位D/A。若是8098单片机，则片内还有1路8位D/A。

键输入：本机配有25个压电式小按键，其中一个做复位键使用，10个做数据键，8个做命令键。

显示输出：本机配有4个7段数码管组成，4个用单显示地址状态数据；2个显示数据。

单片机提供两种8098或8095片内的SIO，通过一片1488和一片1489经快速I/O与上位机进行通讯，也可在提供的TTY模拟软件

SCB-2型单片单板机硬件分析

该单片机系统结构图如图1所示。该单片机系统结构图如图1所示。

吴 微 文 军 编著



单片机原理及制作

武汉创意电子研究所

国内代号：2-888

定价：0.95元

NS 公司单片微机一览表(资料)

梁合庆 供稿

一、COP400 系列 4 位 NMOS/CMOS 单片微机。0.5~2K ROM, 32×4~160×4 RAM。2.4~6.3V 工作电压。PMP 快速掩膜, MICROWIRE 串行口。多种 I/O 口配置。20~28 脚 DIP、OS、PLCC、SMD 封装。4~16μs 指令。宽温度范围。

商业级 0℃~+70℃	工业级 -40℃~+85℃	工 艺	ROM (字节)	RAM (数据)	I/O 脚	串行 I/O	中断源	堆栈	时基 计数器	管脚
COP413L	COP313L	NMOS 低功耗	0.5K	32	15	Yes	No	2 级	No	20
COP414L	COP314L	NMOS 低功耗	0.5K	32	15	Yes	No	2 级	No	20
COP410L	COP310L	NMOS 低功耗	0.5K	32	19	Yes	No	2 级	No	24
COP411L	COP311L	NMOS 低功耗	0.5K	32	16	Yes	No	2 级	No	20
COP413C	COP313C	CMOS 低功耗	0.5K	32	15	Yes	No	2 级	No	20
COP414CH	COP314CH	CMOS 高速	0.5K	32	15	Yes	No	2 级	No	20
COP410C	COP310C	CMOS 高速	0.5K	32	19	Yes	No	2 级	No	24
COP411C	COP311C	CMOS 高速	0.5K	32	16	Yes	No	2 级	No	20
COP420	COP320	NMOS 高速	1.0K	64	23	Yes	1	3 级	Yes	28
COP421	COP321	NMOS 高速	1.0K	64	19	Yes	No	3 级	Yes	24
COP422	COP322	NMOS 高速	1.0K	64	15	Yes	No	3 级	Yes	20
COP424C	COP324C	CMOS 高速	1.0K	64	23	Yes	1	3 级	Yes	28
COP425C	COP325C	CMOS 高速	1.0K	64	19	Yes	No	3 级	Yes	24
COP426C	COP326C	CMOS 高速	1.0K	64	15	Yes	No	3 级	Yes	20
COP420L	COP320L	NMOS 低功耗	1.0K	64	23	Yes	1	3 级	Yes	28
COP421L	COP321L	NMOS 低功耗	1.0K	64	19	Yes	No	3 级	Yes	24
COP422L	COP322L	NMOS 低功耗	1.0K	64	15	Yes	No	3 级	Yes	20
COP440	COP340	NMOS 高速	2.0K	160	36	Yes	4	4 级	Yes	40
COP441	COP341	NMOS 高速	2.0K	160	23	Yes	4	4 级	Yes	28
COP442	COP342	NMOS 高速	2.0K	160	19	Yes	2	4 级	Yes	24
COP444C	COP344C	CMOS 高速	2.0K	128	23	Yes	1	3 级	Yes	28
COP445C	COP345C	CMOS 高速	2.0K	128	19	Yes	No	3 级	Yes	24
COP444L	COP344L	NMOS 低功耗	2.0K	128	23	Yes	1	3 级	Yes	28
COP445L	COP345L	NMOS 低功耗	2.0K	128	19	Yes	No	3 级	Yes	24

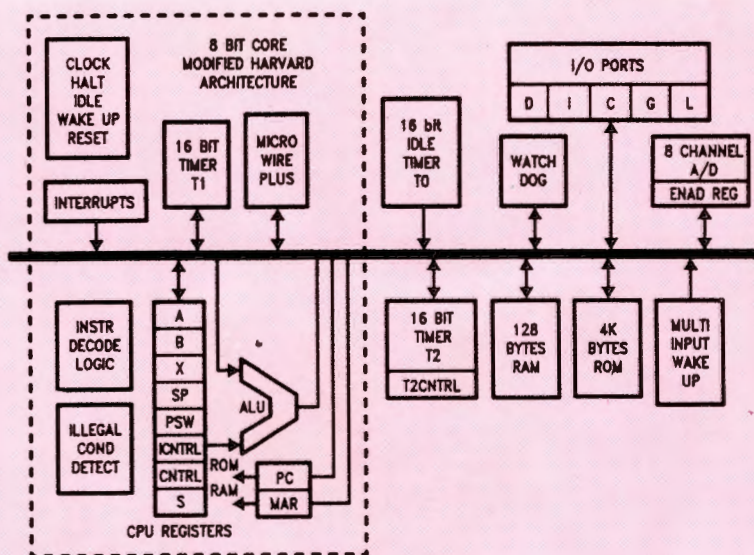
二、COP800 系列 8 位 M²CMOS 单片微机。1~4K ROM, 64~128 RAM。2.5~6.0V 工作电压。A/D、PWM 捕捉定时器、监督定时器、E²PROM。MICROWIRE/PLUS 串行口。多种 I/O 口配置。20~44 脚 DIP、SMD 封装。1μs 指令。宽温度范围。

工业级 0℃~+70℃	军用级 -40℃~+85℃	工 艺	ROM (字节)	RAM (字节)	I/O 脚	串行 I/O	中断源	堆栈	时基 计数器	管脚
COP820C	COP620C	M ² CMOS	1.0K	64	24	Yes	3	In RAM	1	28
COP822C	COP622C	M ² CMOS	1.0K	64	16	Yes	3	In RAM	1	20
COP8620C	COP6620C	M ² CMOS	1.0K	64	24	Yes	3	In RAM	1	28
COP8622C	COP6622C	M ² CMOS	1.0K	64	16	Yes	3	In RAM	1	20
COP8720C		M ² CMOS	1.0KEE	64	24	Yes	3	In RAM	1	28
COP8722C		M ² CMOS	1.0KEE	64	16	Yes	3	In RAM	1	20
COP840C	COP680C	M ² CMOS	2.0K	128	24	Yes	3	In RAM	1	28
COP842C	COP682C	M ² CMOS	2.0K	128	16	Yes	3	In RAM	1	20
COP8640C		M ² CMOS	2.0K	64	24	Yes	3	In RAM	1	28
COP8642C		M ² CMOS	2.0K	64	16	Yes	3	In RAM	1	20
COP880C	COP640C	M ² CMOS	4.0K	128	36	Yes	3	In RAM	1	40/44
COP881C	COP641C	M ² CMOS	4.0K	128	24	Yes	3	In RAM	1	28
COP884CF	COP684CF	M ² CMOS	4.0K	128	21	Yes	10	In RAM	2	28
COP884CG	COP684CG	M ² CMOS	4.0K	192	23	Yes	14	In RAM	3	28
COP884CL	COP684CL	M ² CMOS	4.0K	128	23	Yes	10	In RAM	2	28
COP888CF	COP688CF	M ² CMOS	4.0K	128	33/37	Yes	10	In RAM	2	40/44
COP888CG	COP688CG	M ² CMOS	4.0K	192	35/39	Yes	14	In RAM	3	40/44
COP888CL	COP688CL	M ² CMOS	4.0K	128	33/39	Yes	10	In RAM	2	40/44

(下接封三)

(上接封二)

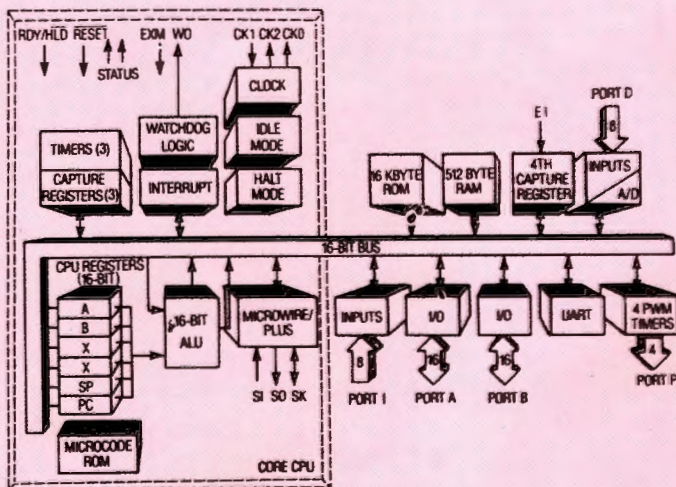
COP888C 系列框图



三、HPC 系列 16 位 M²CMOS 单片微机。8~16K ROM, 256~512 RAM。6 寄存器/8 中断/4 定时器/监督定时器。MICROWIRE/PLUS 串行口。DMA/HDLC/A-D/UART。多至 52 个高速通用 I/O 线、PGA、PLCC、LCL、LDCC、TapepaK 封装。67μs 指令周期(30MHz), 宽温度范围。

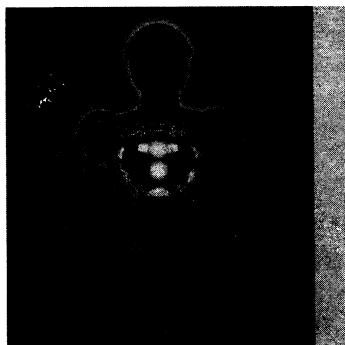
商业级 0℃~+70℃	工业级 -40℃~+85℃	工 艺	ROM (字节)	RAM (字节)	I/O 脚	串行 I/O	中断源	堆栈	时基 计数器	管脚
HPC46003	HPC36003	M ² CMOS		256	52	Yes	8	In RAM	8	68
HPC46083	HPC36083	M ² CMOS	8K	256	52	Yes	8	In RAM	8	68
HPC46004	HPC36004	M ² CMOS		512	52	Yes	8	In RAM	8	68
HPC46064	HPC36064	M ² CMOS	16K	512	52	Yes	8	In RAM	8	68
HPC46104	HPC36104	M ² CMOS		512	52	Yes	8	In RAM	8	68
HPC46164	HPC36164	M ² CMOS	16K	512	52	Yes	8	In RAM	8	68
HPC46400E	HPC36400E	M ² CMOS	0	256	56	Yes	8	In RAM	4	68

HPC46 系列框图





電子
1992
8
與 電腦



• ELECTRONICS AND COMPUTERS •

8

一九九二年

总期第89期

電子與電腦

目 录

• 综述 •

- 计算机信息的安全保护 李 群(2)
量子芯片与人工智能计算机 姚立新(4)

• PC 用户 •

- dBASE II 管理系统中程序多级调用的跟踪 周 为(5)
一种在 M1724 机上打印图形的实用方法 李凯里(7)
物理扇区与逻辑扇区 崔来堂(8)
DOS 命令在 IBM-PC 机通信中的妙用 林立 林 娜(9)
硬盘加锁小程序 刘志存(10)
实用程序三则 任绥海(11)
快而短的排序程序 许野平(12)
1724 打印机驱动程序行距的修改 谭人杰(12)
加密 WPS 文件的解密 谷高平 孟建伟(13)
电脑病毒入侵报警程序 陶秋刚(14)

• 学习机之友 •

- 对几种高精度数值计算方法的改进 陈庆祥(15)
CEC-I 自造字键盘辅助点阵生成程序 汤永进(16)
二维函数曲线拟合 李 涛(17)
在 Applesoft 系统下实现 COMMON 功能 安赵根(18)
通用打印课表程序 程建伟(19)
CP/M 系统文件的恢复方法 李 齐(20)

• 6502 机器语言讲座 •

- 第八讲 子程序设计 朱国江(21)

• 初级程序员级软件水平考试辅导 •

一九九二年计算机初级软件人员竞赛试题 (26)

• 学用单片机 •

- BJS-51 的监控程序 盛焕鸣(31)

• 学装微电脑 •

- 水平多关节型机器人 易齐干(34)

• 电脑巧开发 •

- 点阵字符液晶显示器与单片机接口
..... 张培仁 杨建景(38)

• 电脑游戏机 •

- F BASIC 语言的游戏程序编写技巧
第一讲 游戏程序的概念 于 春(41)

• 维修经验谈 •

- 单色显示器故障维修一例 梁建华(43)
恢复 CEC-I 学习机 DOS 的 I/O 控制 屈晓柳(43)
再谈清洗盘的正确使用 刘德一(44)
对磁盘局部缺损的处理 何管略(44)

• 新书与软件 •

- 电子工业出版社软件部新出版软件介绍 (45)

• 读者联谊 •

- 普及型 PC 个人用户软件交流联谊活动问题解答(八)
..... 王路敬(46)

封一:机器人

封二:NS 公司单片机一览表(资料)

封三:NS 公司单片机一览表(资料)

封四:广告

机械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 副主编:王昌铭

责任编辑:张 丽

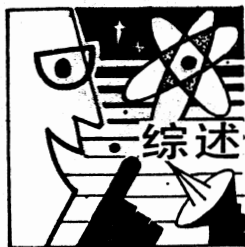
订购处:全国各地邮电局

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 147 号

定价:0.95 元



计算机信息的安全保护

辽宁省财政厅信息中心(110002) 李 群

随着工业化社会向信息化社会过渡的潮流,以计算机技术为主导的高科技成果日益进入社会的各个职能部分,计算机网络迅速发展。据统计,我国现有各种微机四十多万台,中、小型机一万台,计算机已经广泛应用于一系列国民经济的重要部门,在提供决策、制定政策等方面起着越来越重要的作用。然而,由于计算机信息系统和网络本身的脆弱性,以及各种人为因素的影响,计算机信息常常受到干扰、滥用、丢失、甚至被泄漏、窃取、篡改、冒充和破坏;计算机可能受到“病毒”感染而工作瘫痪。利用计算机犯罪也成为高技术犯罪的衍化。加之各种技术性事故、无意疏忽而带来的一系列信息的不安全性,都要求从法律上,组织上及技术上加强计算机信息安全的保护。

一、计算机信息安全保护的范围

信息是一种资源,它是依据某种目的组织起来、经过加工处理和具有一定结构的数据总括。计算机信息可分为应用信息和系统信息两种。对这些信息的威胁大致分为三类:

1. 对计算机应用信息机密性的威胁;
2. 对计算机应用信息真实性和完整性的威胁;
3. 对计算机系统信息的破坏性威胁。

信息保护的主要目的,一是保护信息的机密性,防止未经授权的泄漏;二是保护信息的真实性和完整性,防止未经认可的修改。信息保护有二个重要环节:一是保护通信线路上传输的信息,二是保护存储于计算机系统内的信息。

经计算机通信线路传输的信息,常常受到两种威胁。一是被动窃取,即信息机密性遭到破坏,通常指截取报文而不被察觉出来。保护报文内容不被识破,是依靠对传输信息进行加密变换。二是主动窃取,即信息真实性或完整性遭到破坏,这是指蓄意修改报文流。对这种威胁,加密保护只能提供发现作案,而不能阻止作案。由于我国现在通信网络还没有完全发展起来,针对此类信息的安全的威胁还比较少见。

存储于计算机系统的信息,易于受到和通信线路同样的威胁。威胁到信息机密性的手段有:从主存储器或外存储器查找信息;用合法存取信息的进程为掩护,把信息传给未经认可的用户;或根据公布的一批综合性统计数据,推断出有价值的机密数据。

存储于计算机系统的信息所受到的真实性和完整性威胁,是大量的计算机犯罪的一种结果,这主要指蓄意篡改和破坏计算机的信息(如把某人存款非法提高等);冒充系统获得对其他用户的访问权,而非法

修改合法用户的信息。

另一种对计算机系统信息造成较大破坏性威胁的,就是大量的计算机病毒。这种威胁也是广大计算机用户最常见的。由于我国在计算机立法方面起步较晚,软件产品非法复制、拷贝,加之前几年正处我国掀起普及微机的热潮,为计算机病毒的侵入提供了必要环境,而计算机软、硬件产品的脆弱性又为病毒的侵入提供了技术原因。计算机病毒的传染一般通过三种途径:通过软盘,指不经过严格的检查而轻易使用外界有可能被感染的软盘;通过带有病毒的硬盘,将干净的软盘传染并再扩散;通过网络,这种传染扩散极快,能在很短的时间内传遍网络上的机器。不管是良性病毒还是恶性病毒,轻则占用 CPU 时间和存储空间,影响机器使用效率,重则破坏用户或系统数据,甚至完全毁灭系统。病毒对我国的广大微机用户造成了巨大的威胁,尤应引起高度重视。

二、计算机信息安全保护的措施

为了有效地对各种计算机信息进行安全保护,必须从法律上,行政上及技术上采取相应措施。根据计算机安全专家的思想,保护计算机系统的软件(数据文件和程序)和硬件(设备)有四层安全措施:第一层是法律和社会管理,它通过国家法律和地方法律及社会上允许的行为方式提供保护;第二层是用户本单位的行政规章,规定有关计算机系统使用方法和手段,以及人事结构和人事政策提供保护;第三层是物理保护,指使用防盗和防火技术、软盘保护、限制实际接触计算机系统;第四层是电子的和程序的,是指由硬件器件和在操作系统、数据文件和应用程序中,用软件方法提供的保护。

在国际上,许多国家已经正式颁布了信息保护法。美国于 1974 年正式颁布信息保护条例,1980 年修订了著作权法,正式认定计算机软件是独特的智力产品,是著作权保护的对象。随后,日、英、法等西方国家也为保护软件修改和实施了著作权法。同时,以美国为首的西方国家还竭力以法律手段为软件提供保护,并使之具有国际效力。为了适应改革和对外开放的需要,加速引进国外智力投资,衡量和保证我国引进技术的质量,维护软件开发者合法利益,促进软件业的发展和竞争,防止重复投资,避免低水平重复劳动,我国已制定并颁布了著作权法和软件保护条例,明确规定了软件产品保护的、范围、保护期限及侵权的法律责任,从而为智力产权提供了法律上的保护,为各级司法机关审理侵权行为提供了法律准则。为了维

护软件开发者的利益,迫切需要在全民中普及法制观念,当发生侵权行为时,运用法律手段解决问题。

操作保护措施是指管理上采用的政策和过程,它们保证计算机系统及其信息的安全,虽然管理政策和过程的若干方面已经由外部规定,但是,多数管理政策和过程的实施细则,仍需由内部规定。与信息安全有关的主要管理问题是认可权,即认可谁,根据什么,如何确定。这种保护可以从操作上,组织上提供。主要采取以下方法:

1. 对使用计算机人员的数量加以限制。操作人员应分为一般用户,系统维护人员及特殊用户(如对一个公司的计算机系统,经理则为此类人员,授权级别最高,可以浏览全部信息),对各种人员根据其实际需要,授与相应权限,严格禁止越权操作。

2. 系统地提出并得到上级支持和群众积极参与的安全管理条例及其实施细则,使单位人员认识到信息安全的紧迫性和重要性,自觉地遵守规则,并运用行政手段对违反人员进行惩罚。

物理保护也称实体保护,是对其环境,设施,设备和人员等采取安全措施,使它不致受自然或人为因素的损害。它主要可分为:计算机场地环境和设施保护、实体和信息载体戒备措施。

选择计算机房场地时,应考虑以下问题:

1. 机房不应选择在下列地区:山脚、地基不牢处,靠水或低洼潮湿处,地下有输油输气管道、地震活动频繁及邻近有高层建筑结构处。

2. 应尽量远离有害化学气体、腐蚀性物品和易燃易爆物品存放处。

3. 应远离高压线,避开雷达站、无线电发射台和微波中继站。

4. 应尽量远离强振动源和噪声源。

5. 机房不应设在建筑物高层,因为高层易受外来电磁波干扰。

6. 应有紧急照明供电设施和各种报警装置。

7. 机房应有防火及防水设施。

8. 提供有不同持续工作时间的不间断电源。

9. 保证必要的温度、湿度和防尘指标。

实体和信息载体戒备,可以将处理中心分为二个区:主机房和数据处理区。根据人员实际需要规定进入的区域;在建筑上应防止电磁波破坏,磁带和磁盘应在离开现场的安全处,对载有重要信息的磁带和磁盘,应建库由专人保管,严禁非法的访问与外借。

对信息实体保护除以上措施外,还应当制定应急计划,防备一旦计算机实体发生重大事故并难以迅速恢复时,能够尽可能减少损失。这类应急计划应尽可能详尽地考虑到各种情况下有可能造成的信息丢失或泄漏时的补救方案,以备灾后恢复,如同一现场有两套计算机系统同时工作,互相备用,此时安全度随两机距离增加而加大;对重要信息进行备份等。

对计算机硬件的保护主要保括:中央处理机、存储器、输入输出通道和外围设备的保护,主要由硬件

人员负责定期检查和维修,使计算机系统处于良好的状态,避免由于硬件故障而造成信息破坏。

对计算机信息安全造成威胁最大,破坏程度最深的是大量的针对计算机软件方面的威胁,安全保护的主要工作也在这个方面。由于无论数据还是程序,应用软件还是系统软件,在计算机中都是以文件方式存放的,因而,现在在技术上采用的方法主要有:操作系统保护和文件保护。

对操作系统进行保护,在每次交互式或批量处理会晤期的开始和结束,登录过程必不可少。通过登录,操作系统激活标识和鉴定机构,准确无误地辨识出合法用户。同时,由操作系统负责控制一些安全软件(有些已经固化),一旦发生用户企图修改或无意识地修改系统数据或未被授权的数据,则加以制止。这种方法可以有效地防止一些计算机病毒的侵害。

对文件的保护通常利用口令字、存取权控制、加密和其他一些保护措施,保证文件安全。在用户登录、文件存取或数据项存取期间,设置口令字可对用户身份作出鉴定。一个具体系统选用的鉴定技术,有赖于风险和代价。对可能的威胁、威胁出现的概率及构成损失的估价,与采用保护对策付出的代价存在一种关系。与其他鉴定技术比较,口令字技术付出代价小,口令字手段对制止浏览(即用户利用对系统中一部分合法存取权存取其他未经认可的文件)特别有效。在各级目录和文件上设置口令字,可以对文件存取控制起到保护作用。一个用户申请存取文件,就必须提出正确口令字。这种口令字保护的优点是简便,所需存储空间少。缺点是系统管理人员能取得全部口令字。因为该保护信息存储于系统内。另一种保护措施是把全部文件作加密编码。所有用户都可以存取加密码的文件,但只有合法用户才知道解码的途径。任何时候,用户可以根据需要变更密钥。对文件解码和编码工作,均由操作系统实现。这种方法可以杜绝利用系统内存信息的非法存取,但以编码和解码时的开销为代价。还有一种保护措施即存取权的设置和利用。存取权(许可权)可分为共用许可权和专用许可权。共用许可权的文件或目录,对所有系统确认的用户都是可用的;而专用许可权的文件或目录,则只对那些事先授权的用户是可用的。一个文件或目录只有一种许可权,当同时拥有共用许可权和专用许可权时,则共用许可权取代专用许可权。许可权大致包括下列几种:可读、可写、可执行、恢复、消除、建立、修改和上锁。除以上措施外,还可以在文件正常管理中,统计和追踪异常情况下文件的封锁和非法存取文件的次数,记录文件的跟踪记录(系统日记)。为减少系统开销,可由操作员用命令选择全跟踪、跟踪几项、不跟踪。根据情况,可以把上述几种方法综合利用。

三、计算机安全保护的前景及面临的问题

计算机信息安全保护的范围很广、涉及面大,必须尽可能防止各种不安全因素,特别是随着科技的发展,各国之间的竞争将更趋激烈,以电子技术为先导

的竞争对计算机信息安全提出了更高的要求,即保证计算机信息永远处于安全状态。信息尤其是重要信息更不能丢失或毁坏。这就要求安全专家们从方法学角度,研究计算机系统和通信网络内信息保护方法,使计算机信息保护理论更为系统和完整。可以说,一个好的计算机系统应该在为用户提供灵活方便的功能的同时,也为用户提供良好的安全保护措施。

任何事物都不是绝对的,计算机信息也不可能有百分之百的安全。信息共享与保护、破坏与安全、扩散与保密是互相对立、互为制约的。安全性的提高是对其他方面的限制为代价的。同时,计算机信息保护环境的复杂多变性,使实现一个有成效的、经济可行和易于使用的信息保护系统十分困难,这是因为:

1. 信息保护系统是否有成效,不仅取决于保护措施是否完善,也取决于袭击者的手段是否高超,取决

于系统脆弱性是否易于识破,取决于系统一旦遭到破坏所造成的直接和间接损失。事实上,要估量袭击者采用的手段和系统潜在威胁是极其困难的。

2. 内部和知情人员的破坏,更危险和难以防止。

3. 技术事故和误操作造成的损失和破坏次数较之蓄谋作案大得多。

4. 各类信息保护系统差异很大,研制通用的信息保护系统很困难。

5. 降低信息保护软件、硬件的成本和管理费用,减少多余开销,始终是一个重要因素,尤其在商用系统中。同时,还应当把保护与易于使用结合起来。

总之,计算机使用得越广泛深入,对信息共享与保护的需求就越突出。因此,满足用户的要求,具有合理的安全价格比的系统就是一个良好的计算机系统。

量子芯片与人工智能计算机

山西临汾地区计委经济信息
中心(041000) 姚立新

电子计算机从 1946 年间问世以来的 40 年多里,经历了电子管、晶体管、集成电路、大规模集成电路四代更新,目前正向着采用第五代元器件构成的智能计算机(也叫第五代计算机)方向发展。那么,人工智能计算机与传统的计算机有什么区别呢?与计算机发展密切相关的元器件——芯片发展的现状又是什么样呢?

人工智能计算机与传统计算机的区别

首先,这两种计算机的区别在于传统的计算机采用的是冯·诺依曼结构(也叫冯·诺依曼计算机),而人工智能计算机采用的是非冯·诺依曼结构。所谓冯·诺依曼结构,它的特点是把要处理的数字、文字、图象、声音全部变成 1 或 0 这样的数字进行运算和记忆,而且按指令的内容,一条一条的完成所规定的任务。人工智能计算机中引入了数据流的概念,这种计算机是由数据而不是由指令来驱动程序的执行,它是靠运算器中操作数的全部到达为标志来驱动操作执行的机器,因此,当多个操作数同时满足这一条件时,它们就可以并行地执行而不受程序的顺序限制,据科学家初步估计,数据流计算机的处理速率要比冯·诺依曼计算机至少提高 2 个数量级。

其次,这两种计算机的区别还在于人工智能计算机采用了并行结构,传统计算机的基本运行顺序是从中央存储器中取出数据,对它进行加工,再存回到中央存储器,而在人工智能计算机中采用的却是并行结构。比如:有一种平行计算机,它的一个处理机可以管十来

对它们同时施以一系列的运算,然后再送回到分布式存储器中,这样一来,就极大的提高了计算机的效率。

另外,在人工智能计算机中还采用了比传统计算机更容易设计的机器语言——PROLOG 语言,这种语言是一种逻辑设计语言,它使得程序员能够用逻辑来规定机器的任务而不是象传统计算机那样用程序设计来规定机器在什么时候作什么工作。

芯片发展趋势——量子芯片

计算机离不开芯片的支持。对人工智能计算机来说,芯片的精度要求更高。在日本,科学家已研制出体积只有指甲大小但却具有 64 兆位的记忆芯片,在这种芯片上密密麻麻地分布满了 1.4 亿个电子元件,而连接它们的电路只有 0.4 微米宽,相当于人的头发丝直径的千分之四。但这还是不够的,为了提高芯片的运算速度,还必须缩小电子元件的尺寸。然而科学家认为,当芯片的电路宽度小于 0.1 微米时,电子就不会那么老老实实地在电路里运动了,它们很可能从电路里跳出来,将整个芯片搅得一片混乱,因为此时电路的宽度太窄已使得电子的波动性不可忽略了,此时,电子既具有粒子特性,又具有较强的波动特性,这样就引进了量子芯片。

所谓量子芯片就是利用电子的波动性来设计出的一种高密度芯片,该芯片的问世将使巨大的超级计算机体积变得与现在膝上电脑那么大,并且使得计算机的运算速度到万亿次成为可能。虽然量子芯片的研究还处在初级阶段(比如说,目前日本富士通研制出的实验型量子芯片必须在零下 200℃ 才能工作。),但是,通过科学家的努力,不久的将来这种芯片将进入实用阶段。

总之,人工智能计算机是一种集成度更强、运算速度更快、智能程度更高、更接近于人的思维行为的一种计算机系统,量子芯片的研制为人工智能计算机的设计提供了一种理想的硬件,使将来的计算机体积更小,运算速度更快。



PC 用户

dBASE Ⅲ 管理系统中 程序多级调用的跟踪

甘肃平凉 103 信箱 302 分箱(744000) 周 为

编写或使用 dBASE Ⅲ 管理系统程序的人员常常由于工作上的需要,希望能够迅速地了解管理系统调用各子程序的调用关系和能打印一份该系统完整的程序清单,以便于存档,用于今后该系统的维护、学习,交流和改进编程技巧。但由于每个管理系统中的每一功能还常常包含着多级的子程序调用关系;而且每级中又有着多个数量不确定的子程序,一般情况下想要全面地了解系统的程序调用关系和打印出一份完整的程序清单是不容易的。因为我们通常只能自上而下地对程序进行分析,逐步了解各程序之间的调用关系而打印出全部程序清单。显然这是一件相当麻烦的工作。

笔者针对以上存在的问题编写了一个包括“自动建立管理系统多级调用关系数据库”、“打印系统程序调用关系的结构示意图”和“自动打印系统程序的全部清单”三个功能的子程序。操作者只需通过选项 1 键入当前管理系统的主菜单名,就能自动建立起该系统中的调用关系数据库。随后再键入 2 或键入 3,就可以达到我们期望的目的。

库文件结构—— 文件名: C:\DB. dbf

字段	字段名	类型	宽度	小数
1	PRG 名	字符型	12	
2	JB	数值型	1	

库文件结构—— 文件名: C:\DB1. dbf

字段	字段名	类型	宽度	小数
1	PRG 名	字符型	100	
2	JB	数值型	1	

```
* FILE. PRC
SET TALK OFF
SET SAFE OFF
DO WHILE . T.
CLEAR ALL
CLEAR
STORE 0 TO LL, YY
NAME1 = ""
@5, 24 SAY "*****"
@7, 24 SAY "1. 自动建立子程序多级调用关系数据库"
@8, 24 SAY "2. 打印子程序调用结构示意图"
@9, 24 SAY "3. 自动打印系统程序全部清单"
@10, 24 SAY "0. 退出"
@12, 24 SAY "*****"
@14, 34 SAY "请输入(0-3)" GET YY PICT "9" RANGE 0, 3
READ
DO CASE
CASE YY = 1
```

```
ACCE "主菜单文件名:" TO NAME1
NAME = NAME1 + ". PRG"
IF FILE( "&NAME' ")
SELECT 1
USE DB
ZAP
APPE BLANK
REPL PRG 名 WITH NAME, JB WITH 1
KK = JB + 1
POINT = 1
DO FILE1
POINT = POINT + 1
USE DB
APPEND FROM DB1
APPEND BLANK
REPL JB WITH 0
DO WHILE . T.
USE DB
GO POINT
IF JB = 0
EXIT
ELSE
NAME = PRG 名
ENDI
KK = JB + 1
POINT = POINT + 1
GO POINT
COPY TO DB2 NEXT 100
DO FILE1
IF LL = 0
LOOP
ENDI
USE DB1
APPEND FROM DB2
USE DB
GO POINT
DELE NEXT 100
PACK
APPEND FROM DB1
ENDD
USE
ELSE
? CHR(7)
? '没有此文件'
ENDI
CASE YY = 2
USE DB
SET PRIN ON
```

? “子程序调用结构示意图”

```
DO WHILE .NOT. EOF()
DO CASE
CASE JB=1
? PRG 名
CASE JB=2
? SPACE(4)+"└——"+PRG 名
CASE JB=3
? SPACE(14)+"└——"+PRG 名
CASE JB=4
? SPACE(24)+"└——"+PRG 名
CASE JB=5
? SPACE(34)+"└——"+PRG 名
CASE JB=6
? SPACE(44)+"└——"+PRG 名
CASE JB=7
? SPACE(54)+"└——"+PRG 名
CASE JB=8
? SPACE(64)+"└——"+PRG 名
CASE JB=0
EXIT
ENDCASE
SKIP
ENDD
?
SET PRIN OFF
CASE YY=3
USE DB
DO WHILE .NOT. EOF()
```

```
MM=PRG 名
TYPE &MM TO PRIN
SKIP
ENDD
CASE YY=0
RETU
ENDCASE
ENDD
* FILE1. PRG
USE DB1
ZAP
APPE FROM &NAME SDF
DELE ALL FOR AT("DO",PRG 名)=0
PACK
GO TOP
DO WHILE .NOT. EOF()
NAME=TRIM(SUBS(PRG 名,AT('DO',PRG 名)+3,8))+
". PRG"
IF FILE('&NAME')
REPL PRG 名 WITH NAME,JB WITH KK
ELSE
DELE
ENDI
SKIP+1
ENDD
PACK
COUN TO LL
USE
RETU
```

(接第 42 页)

```
25 PL. "G4A1,F4F1,O3D4O2B1"
30 PL. "B5O4C7;F5E7;G5C7"
35 CLS;ERA 0,1;SP. O.;CG. 0,0;U=21;PAL. B 0,15,
48,48,U
40 A$="AIIIIIIIIIGIGGGGH";B$="HHBHBBBE-
BEEE";K$="ABCDEFGHI";I$="ABCGHIDEF"
45 LOC. 9,10;P. "START!!!";PAU. 80;CLS;X=13;
Y=22;I=0
50 GOS. 100;SW. A$,B$;GOS. 100
55 IF I=0 T. SW. K$,I$;I=1;G. 50
60 G. 150
100 L=LEN(A$);F. P=1 TO L;K=P;IF I=0 T. 110
105 K=L+1-P
110 C=INS. (K$,MI. (A$,K,1))-1;V=C/3;H=C
MOD 3
115 X=X+H+3*(H=2);Y=Y+V+3*(V=2);LOC.
X,Y;P. CH. (207);N.;RE.
150 W=52;F. N=0 TO 1;DE. M. (N)=SP. (N,3+4*N,
1,1,0,0);POS. 236*N,200;N.
155 M. 0,1;IF CRA. (0)=1 T. 165
160 G. 155
165 LOC. 10,10;P. "PERFECT";F. N=0 TO 100;PAL. B 0,
6,48,48,U;SW. U,W;N.
170 PL. "M1V2Y1T2;M1V10Y1T2;T2"
```

```
175 PL. "O3E3G#DGDG#DG;O1#B5B#AB;R"
180 PL. "EGO4CO3E;Y2O2#B6E3;O1C5C"
185 PL. "#GAO4DC;F5#F;#C#C"
190 PL. "O3B#ABO4A;F3O3DFA;D6F3"
195 PL. "GO3AO4FO3B;E5DV15;GA#AB"
200 PL. "#B3GT3FGE1DV12E7;C9;#B5T3GC7"
210 LOC. 0,22;E.
```

程序说明:

1. 10~30 行为开始音乐。
2. 40~115 行打印心形图案,其原理已在 FB ASIC 语言及程序设计中作过介绍。
3. 150~160 行定义玛丽和丽莎运动,注意两卡通运动方向和运动横座标的技术处理。
4. 165 行的作用是使图案闪光,实际上是红、白两种颜色背景的交替显示。
5. 170~200 行是祝贺音乐。

分析上两个简单游戏程序,得出结论如下:游戏程序是具有某种游戏功能的特殊程序。它同其它任何程序一样,具有输入、输出、循环、判断、结束等处理语句,所不同的是,它比一般程序结构更复杂,内容更丰富。所以,我们说游戏程序是电脑实力的充分发挥,是电脑计算、判断、声音、图象、动画等各种功能的集合表现。

注:例 2 的程序清单的指令使用了简写形式,为压缩篇幅,以后各讲均使用简写形式,不再说明。

一种在 M1724 机上打印图形的实用方法

山东烟台大学(264005) 李凯里

许多只具备基本配置的 PC 机用户,为了便于技术信息交流,常常希望使用普通打印机得到较满意的图形。

这里向配有 M1724 打印机的 PC 用户介绍一种简便实用的图形打印方法。其主要特点一是能将屏幕图形完全按比例地拷贝下来,不会出现几何畸变(如当屏幕显示正方形时,打印输出图形仍为正方形)。其二是分有正常尺寸与扩大尺寸两种选择。后者恰为前者之两倍,约占一页打印纸的长度。图为按正常尺寸打印的图形例子(一平面振荡波形的局部立体图)。

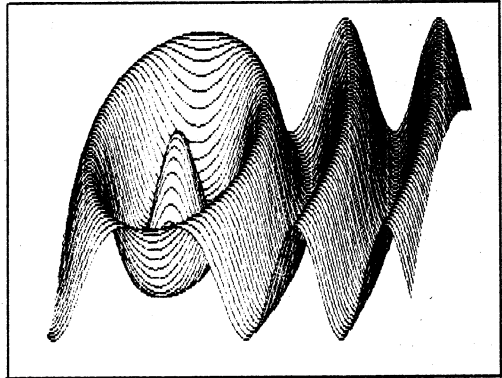
程序用 BASICA 语言写成,用户易于读懂。其中 BX(20)数组用于扩大方式下的打印字节转换。变量 WOB 产生反视效果(黑色底图)。

欲打印的图形事先已经生成,并以 *.GDT 文件名存入硬盘。每次打印可按文件名进行选择。

· 本程序中各循环参数视图形规模及内存格式而定。这里为 16K/帧,首地址在 B8000H 处。对其它规模和存放格式的图形,需修改相应的参数。另外特别指

出,通过适当调整打印控制字,也可将该程序的设计原理应用于其他型号打印机的图形打印。

(注:本例 M1724 机的 DIP 开关第 7 位置于 ON 位置)



图

图形打印程序清单

```
100 '*** M1724 ***
110 CLEAR:CLS:SCREEN 1:KEY OFF:DIM BX(20)
120 BX(0)=&H0;BX(4)=&H30;BX(8)=&HC0;BX(12)
    =&HF0
130 BX(1)=&H3;BX(5)=&H33;BX(9)=&HC3;BX(13)
    =&HF3
140 BX(2)=&HC;BX(6)=&H3C;BX(10)=&HCC;BX(14)
    =&HFC
150 BX(3)=&HF;BX(7)=&H3F;BX(11)=&HCF;BX(15)
    =&HFF
160 FILES "*.GDT
170 PRINT:INPUT "Filename?",FILENAME$
180 CLS:PRINT "0---Normal      2---Enlarge
190 K$=INKEY$:IF K$="0" THEN SX=0 ELSE IF K$
    ="2" THEN SX=2 ELSE 190
200 PRINT:PRINT "1---Black on white    3---White on black
210 K$=INKEY$:IF K$="1" THEN WOB=0 ELSE IF K$
    ="3" THEN WOB=255 ELSE 210
220 DEF SEG=&HB800:BLOAD FILENAME$
230 E$=CHR$(27):LPRINT E$+"@":WIDTH
    "LPT1:",255
240 LPRINT E$+"J"+CHR$(19)
250 IF SX=0 THEN 680 ELSE 260
260 '*** Enlarge ***
270 FOR ROW=0 TO 80 STEP 3
280   LPRINT E$;"=":CHR$(1):CHR$(244);
290   FOR COL=99 TO 0 STEP -1
```

```
300   LOCA=COL*80+ROW
310   BYTE1=PEEK(LOCA+&H2000) XOR WOB
320   BYTE2=PEEK(LOCA+&H2001) XOR WOB
330   FOR K=1 TO 3
340     LPRINT CHR$(BX((BYTE1 AND &HF0)/16));
350     LPRINT CHR$(BX((BYTE1 AND &HF)));
360     LPRINT CHR$(BX((BYTE2 AND &HF0)/16));
370   NEXT K
380   BYTE4=PEEK(LOCA+0) XOR WOB
390   BYTE5=PEEK(LOCA+1) XOR WOB
400   FOR K=1 TO 2
410     LPRINT CHR$(BX((BYTE4 AND &HF0)/16));
420     LPRINT CHR$(BX((BYTE4 AND &HF)));
430     LPRINT CHR$(BX((BYTE5 AND &HF0)/16));
440   NEXT K
450 NEXT COL
460 LPRINT
470 LPRINT E$;"=":CHR$(1):CHR$(244);
480 FOR COL=99 TO 0 STEP -1
490   LOCA=COL*80+ROW
500   BYTE2=PEEK(LOCA+&H2001) XOR WOB
510   BYTE3=PEEK(LOCA+&H2002) XOR WOB
520   FOR K=1 TO 3
530     LPRINT CHR$(BX((BYTE2 AND &HF)));
540     LPRINT CHR$(BX((BYTE3 AND &HF0)/16));
550     LPRINT CHR$(BX((BYTE3 AND &HF)));
560   NEXT K
```

```

570  BYTE5=PEEK(LOCA+1) XOR WOB
580  BYTE6=PEEK(LOCA+2) XOR WOB
590  FOR K=1 TO 2
600      LPRINT CHR$(BX(BYTE5 AND &HF));
610      LPRINT CHR$(BX((BYTE6 AND &HF0)/16));
620      LPRINT CHR$(BX(BYTE6 AND &HF));
630  NEXT K
640  NEXT COL
650  LPRINT
660 NEXT ROW
670 END
680 ' * * * Normal * * * * *
690 FOR ROW=0 TO 80 STEP 3
700  LPRINT E$;"G";CHR$(1);CHR$(244);
710  FOR COL=99 TO 0 STEP -1
720      LOCA=COL*80+ROW

```

```

730  BYTE1=PEEK(LOCA+&H2000) XOR WOB
740  BYTE2=PEEK(LOCA+&H2001) XOR WOB
750  BYTE3=PEEK(LOCA+&H2002) XOR WOB
760  BYTE4=PEEK(LOCA+0) XOR WOB
770  BYTE5=PEEK(LOCA+1) XOR WOB
780  BYTE6=PEEK(LOCA+2) XOR WOB
790  A$=CHR$(BYTE1)+CHR$(BYTE2)+CHR
    $(BYTE3)
800  B$=CHR$(BYTE4)+CHR$(BYTE5)+CHR
    $(BYTE6)
810  LPRINT A$;A$;A$;B$;B$;
820  NEXT COL
830  LPRINT
840 NEXT ROW
850 END

```

物理扇区与逻辑扇区

石家庄铁道学院(050043) 崔来堂

一、两种编址方式

扇区是磁盘操作的最小可寻址单位。PC-DOS对磁盘扇区的管理,分为物理扇区和逻辑扇区两种方式。物理扇区是按照磁道(硬盘为柱面,下同)、磁头、扇区的三维物理位置而进行编址的,逻辑扇区则是将磁盘扇区自始至终进行连续一维编址的结果。

逻辑扇区也称相对扇区,其编址规则是:以盘的0磁道0磁头1扇区为逻辑扇区0,由此开始,对0磁头各扇区依次编号,接着对1磁头各扇区编号,……直到各磁头对应的扇区编完,再编1磁道各磁头的扇区,……这样一直编到末磁道末磁头的末扇区。

二、有关的中断调用

有两种中断调用专用于磁盘的扇区操作。一是ROM BIOS的INT 13H,其入口参数用物理扇区:

AH 子功能号(0:磁盘控制器复位,2:读盘,3:写盘,5:格式化)
AL 扇区数
CH 磁道号
CL 起始扇区号
DH 磁头号
DL 驱动器代号(0:A,1:B,80:C,……)
ES:BX 内存缓冲区指针

另一是DOS的INT 25H(读盘)和INT 26H(写盘),其入口参数用逻辑扇区:

AL 驱动器代号(0:A,1:B,2:C,……)
CX 扇区数
DX 起始逻辑扇区号
DS:BX 内存缓冲区指针

物理扇区直观,但参数较多,如果待操作的若干扇

区不在同一磁道同一磁头内,就必须分几次调用INT 13H。逻辑扇区参数单一,待读写的多个扇区只要编号连续,一次调用INT 25H(26H)即可,使用起来很方便。对于硬盘,物理扇区的概念适用于整个盘空间,逻辑扇区只适用于DOS分区,因此,对主引导扇区的读写不能使用INT 25H(26H),而必须使用INT 13H,所用程序小段如下(设由DEBUG编写,读主引导扇区):

```

-A100
MOV AX,0201
MOV CX,0001
MOV DX,0080
MOV BX,0200
INT 13
INT 20

```

三、换算方法

应用中,两种扇区之间有时需进行换算。

1. 由物理扇区换算为逻辑扇区

5英寸360K软盘的换算公式是:

逻辑扇区号=磁头号*9+磁道号*18+扇区号-1

5英寸1.2M高密软盘的换算公式是:

逻辑扇区号=磁头号*15+磁道号*30+扇区号-1

2. 由逻辑扇区换算为物理扇区

通用公式是:

扇区号=逻辑扇区号MOD每道扇数+1

磁头号=(逻辑扇区号/每道扇数)MOD磁头数

磁道号=逻辑扇区号/每道扇数/磁头数

其中,MOD为除后取余数,"/"为除后取整数商。

DOS模块能够完成上述换算。实现逻辑扇区换算为物理扇区的子程序如下(其中,CODE为段名,BPB

为盘的 I/O 参数表, LOG-S 单元含有待换算的逻辑扇区号):

```

LS-PYA  PROC NEAR
        PUSH CS
        POP DS
        ASSUME DS, CODE
        PUSH AX
        PUSH DX
        MOV AX, WORD PTR LOG-S
        XOR DX, DX
        DIV WORD PTR BPB+13; 除以每道扇数
        INC DL
        MOV PYS-S, DL; 送物理扇区号
        XOR DX, DX
        DIV WORD PTR BPB+15; 除以磁头数
        MOV PYS-H, DL; 送磁头号
        MOV PYS-C, AX; 送磁道号(PYS-C 按字定义)
        POP DX
        POP AX
        RET
LS-PYA  ENDP

```

四、基本应用

磁盘文件的读写过程清晰地说明了两种扇区的应用及它们之间的关系。今以读文件为例简述如下:

DOS 首先从盘的文件目录表 FDT 中查找待读文件名和扩展名, 若存在, 由对应目录项中查得该文件的起始簇号; 再由此访问盘的文件分配表 FAT, 依次查出文件分布在盘文件区的各个簇, 每查得一个簇号, 先按下述公式换算为逻辑扇区:

盘类型	DOS 版本号	换算公式
360K 软盘	2. X	簇号 * 2 + 8
1.2M 软盘	3. X	簇号 + 27

20M 硬盘	2. X	簇号 * 16 + 49
20M 硬盘	3. X	簇号 * 4 + 107

然后利用前述子程序将逻辑扇区换算为物理扇区, 再调用 INT 13H 完成读盘。

DOS 模块中调用 INT 13H 的子程序如下(其中, DRIVE 单元含有驱动器代号, 其它标号的意义同上述子程序):

```

DISKIO  PROC NEAR
        PUSH CS
        POP DS
        ASSUME DS, CODE
        PUSH DX
        PUSH CX
        MOV DX, WORD PTR PYS-C
        MOV CL, 6
        SHL DH, CL; 获取磁道号的高 2 位
        OR DH, PYS-S; 与扇区号组合
        MOV CX, DX
        XCHG CH, CL; 使 CH 为磁道号, CL 为扇区号
        MOV DH, PYS-H
        MOV DL, DRIVE
        INT 13H
        POP CX
        POP DX
        RET
DISKIO  ENDP

```

需要说明的是: 硬盘的柱面数通常大于 256, 8 位的 CH 不够用, 故借用 CH 的高 2 位。这样, 程序中磁道(柱面)号共 10 位: 高 2 位送 CL 的高 2 位, 低 8 位送 CH; 扇区号为 6 位, 送 CL 低 6 位。

DOS 命令在 IBM-PC 机通信中的妙用

中科院广州电子所(510070) 林立
水利部珠江水保局 林娜

介绍 PC 机通信的书刊很多, 一般是利用 PC 机的异步通信适配器与 PC 机、VAX 机、单片机等进行通信, 编写的通信程序大多数采用 BASIC 语言、汇编语言和 C 语言。下面介绍一种用 DOS 命令直接实现 PC 机通信的方法。

首先, 通信双方设置异步适配器参数应一致。例如, 设置异步通信口 COM1: 的传输速率为 1200BPS, 无校验, 8 位数据位, 1 位停止位。在 PC 机中, 用 MODE 命令设置异步适配器参数。

MODE COM1: 1200, N, 8, 1

然后, 利用 DOS 中的 COPY 命令把异步通信适配器 COM1: 作为 I/O 设备进行通信。例如,

COPY COM1: FILENAME 接收文件
COPY COM1: CON: 控制台接收字符
COPY /B FILENAME COM1: 发送二进制文件
COPY CON: COM1: 控制台发送字符

注意: 在异种机进行通信时, 只可以从 PC 机异步通信口发送二进制文件, 不可以接收二进制文件; 而在 PC 机间进行通信时则无此限制。

可见, 利用 DOS 命令实现 PC 机近程通信, 简单、方便、可靠, 可推广应用。由于 DOS 命令没有验错、数据流同步、通信接口状态判别等措施, 对于远程通信不太可靠。

硬盘加锁小程序

陕西师范大学物理系(710062) 刘志存

随着个人计算机的广泛使用,特别是配有硬盘的微机越来越多。很多应用软件和数据都存放在硬盘中,硬盘容量大,使用人员杂,任何用户都可以不加限制地读写硬盘,难以确保文件的安全和保密。因此,对于硬盘宜加以保护,以禁止非法用户的使用。

这样做有如下好处:1. 避免他人使用硬盘时,不慎破坏其中存储的有用信息;2. 防止外来软盘上的病毒感染给硬盘;3. 防止他人查看盘上的保密信息。

本人编制了一小程序,可对硬盘进行加锁,使用极方便。硬盘加锁后,未经解锁,他人只能使用软盘启动系统,而不能使用硬盘。

用 EDLIN、MASM、LINK 对该程序进行编辑、汇编、链接后,再用 EXE2BIN.COM 将程序转化成扩展名为 .COM 的可执行程序(起名为 LOCK.COM),存入软盘中备用。需对硬盘加锁时,运行软盘上的加锁程序 LOCK.COM,则屏幕上显示:

The hard disk is unlocked! Lock it? (y/n)

键入“y”则程序执行完毕,硬盘被加锁,重新开机后硬盘不能引导系统。

下次使用硬盘时,先用系统软盘引导系统,然后运行 LOCK.COM,屏幕上显示:

The hard disk is locked! Unlock it? (y/n)

键入“y”程序执行完毕,硬盘被解锁,此时需重新启动系统才可使用硬盘。

程序在 PC/XT、0520、286 等系列机上运行通过。

```
code segment
    assume cs:code,ds:code,es:code
    mov ax,0201h
    mov bx,0200h
    mov cx,0001h
    mov dx,0080h
    int 13h
    jb q2
    mov si,03feh
    mov bp,0aa55h
    cmp [si],bp
    jz q1
    mov [si],bp
    mov dx,offset d1+100h
    jmp q3
q1: mov [si],ax
    mov dx,offset d2+100h
q3: mov ah,09h
    int 21h
```

```
mov ah,00h
int 16h
cmp al,79h
jz q4
cmp al,6eh
jz q5
jmp q3
q4: mov ax,0301h
    mov bx,0200h
    mov cx,0001h
    mov dx,0080h
    int 13h
    jb q2
    jmp q5
q2: mov dx,offset d3+100h
    mov ah,09h
    int 21h
q5: int 20h
d1 db "The hard disk is locked!"
    db "Unlock it? (y/n)",13,10,"$"
d2 db "The hard disk is unlocked!"
    db "Lock it? (y/n)",13,10,"$"
d3 db "Error! I cannot read or"
    db "write the hard disk!",13,10,"$"
code ends
end
```

大量供应配电脑用显示器及电源

我部现货供应进口彩色/单色(黑白)高分辨率图形显示器,规格有 19"、14"、12"、9"等。并有美国产配电脑用大功率不间断电源,以及各种规格“日立”电缆和装配线供应。以上产品批发价将相当优惠,欢迎来函、来电或来人洽谈。

广东四会县南方电子厂经营部

地址:广东省四会县城高观东路 71 号

电话:(07663)322686

电挂:1311

邮编:526200

实用程序三则

甘肃庆阳长庆一中(745100) 任绥海

一、方便实用的服务程序

本程序虽然短小,功能却多。它可以建立文件,也可以显示已有的文本文件,还能修改任意一行的内容,并把它打印出来。

```
1 SCREEN 2;KEY OFF
2 SOUND 440,8;PRINT "Create or Display? (c/d)
3 CD$=INPUT$(1);PRINT
4 SOUND 440,8;INPUT "File name?",FL$
5 PRINT;SOUND 440,8;PRINT "Maximum number of
  line?";INPUT",LAST
6 DIM A$(LAST)
7 IF CD$="d" OR CD$="D" THEN 16
8 CLS
9 FOR X=1 TO LAST
10 PRINT USING "###";X;PRINT " ";
11 SOUND 440,8
12 LINE INPUT A$(X)
13 IF A$(X)="." THEN 15
14 NEXT
15 PRINT;PRINT;PRINT;GOTO 25
16 CLS;OPEN FL$ FOR INPUT AS #1;PRINT
17 FOR X=1 TO LAST;IF EOF(1) THEN LAST=X;GOTO
  22
18 LINE INPUT #1,A$(X)
19 PRINT USING "###";X;PRINT " ";A$(X);A=
  CSRLIN
20 IF A MOD 24=0 THEN A2$=INPUT$(1);CLS
21 NEXT
22 CLOSE
23 SOUND 440,4;PRINT "Edit the text?";Y$=INPUT
  $(1)
24 IF Y$<>"y" AND Y$<>"Y" THEN 44 ELSE GOTO
  27
25 SOUND 440,4;PRINT "Edit the text?";Y$=INPUT
  $(1)
26 IF Y$<>"Y" AND Y$<>"y" THEN 32
27 SOUND 440,4;INPUT "Which line to edit?",Y1
28 IF Y1=0 THEN 32
29 PRINT SPC(4);A$(Y1);A=CSRLIN-1
30 LOCATE A,5;SOUND 440,4;LINE INPUT A$(Y1)
31 GOTO 27
32 CLS
33 FOR X=1 TO LAST
34 PRINT USING "###";X;PRINT " ";A$(X);A=
  CSRLIN
35 IF A MOD 24=0 THEN A2$=INPUT$(1);CLS
36 NEXT
```

```
37 SOUND 440,4;PRINT "Go on or End? (g/e)";A$=
  INPUT$(1)
38 IF A$="g" OR A$="G" THEN 27
39 OPEN FL$ FOR OUTPUT AS #1
40 FOR X=1 TO LAST
41 PRINT #1,A$(X)
42 NEXT
43 CLOSE
44 SOUND 440,4;PRINT "Lprint the text?";Y$=INPUT
  $(1)
45 IF Y$<>"y" AND Y$<>"Y" THEN GOTO 48
46 FOR X=1 TO LAST
47 LPRINT A$(X);NEXT
48 SOUND 440,4;PRINT "Go on or End? (g/e)";A$=
  INPUT$(1)
49 IF A$="g" OR A$="G" THEN RUN
```

二、字符居中打印程序

我们在设计文件封面或标题时,总要求将字符打印在纸张中部,每打印一次,都要换算字符串长度、纸张宽度、不同字型时的字符宽度等等,往往需要调试多次才能成功。本人编了一个短短的 BASIC 程序,大大简化了这一过程。实际操作时,只需将纸张的中部对准打印机的中部即可。

```
10 KEY OFF;CLS;COLOR 7,0
20 LOCATE 2,27;PRINT "字符居中打印程序"
30 LOCATE 3,27;PRINT "#####"
40 LOCATE 10,22;PRINT "
"
50 LOCATE 6,25;PRINT "请输入要打印的内容";SOUND
  440,4
60 LOCATE 8;COLOR 7,2;PRINT STRING$(80,32);LO-
  CATE 8;LINE INPUT",B$
70 IF B$="." OR B$="*" THEN SOUND 800,8;COL-
  OR 7,0 END
80 IF B$=" " THEN LPRINT CHR$(10);COLOR 7,0;
  GOTO 40
90 SOUND 400,4;COLOR 7,0
100 LOCATE 6,25;PRINT "
"
110 LOCATE 10,22;PRINT "请选择字型(A,B,C,D)";C$
  =INPUT$(1)
120 IF C$="A" OR C$="a" THEN PR$=CHR$(27)+
  "IA";A1=1;GOTO 170
130 IF C$="b" OR C$="B" THEN PR$=CHR$(27)+
  "IB";A1=2;GOTO 170
140 IF C$="c" OR C$="C" THEN PR$=CHR$(27)+
  "IC";A1=1;GOTO 170
150 IF C$="D" OR C$="d" THEN PR$=CHR$(27)
```

```

+“ID”;A1=2;GOTO 170
160 SOUND 800,8;GOTO 110
170 A1=50-LEN(B$)*A1/2
180 F$=STRING$(A1,“ ”)
190 LPRINT CHR$(27)+“IA”+F$+PR$+B$
200 GOTO 40

```

三、屏幕拷贝妙法

我们经常需要将屏幕上的显示信息输出到打印机上。虽然操作系统中有屏幕硬拷贝功能,可以在键盘上方便地实现,但它并不很实用。因为它会把屏幕上所有有用的和无用的信息全部打印出来。而下面的这个小程序,可以随意拷贝屏幕上任何指定区域的信息(区域由 H1-H2,L1-L2 指定)。

```

500 FOR I=H1 TO H2:’行数
600 FOR J=L1 TO L2:’列数
700 C=SCREEN(I,J)
800 A$=A$+CHR$(C)
900 NEXT
1000 LPRINT A$
1100 A$=“”
1200 NEXT

```

快而短的排序程序

山东菏泽县一中(274035) 许野平

本文给出了三个简化的基数排序程序。由于采用了基数排序的基本原理,排序效率当然在众多的算法中要算最高了。它所花费的时间与排序数据成正比例,不会象其它算法那样随数据量增大所需时间急剧增加。在程序设计方面,本文提供的三个程序经过了巧妙的构思,因此整个排序过程只用短短的一行。

程序首先产生 1000 个随机数,然后按三种不同方式排序。程序假设这 1000 个数为学生成绩,sort01,sort02 两程序按学员号顺序输出,但前者允许并列名次,后者取消了并列名次。stor03 程序可按名次顺序依次输出学生成绩单。

在 IBM PC/XT 上实测,sort01 用 5 秒,sort02 用 7 秒,sort03 用 8 秒。如果你仔细地分析一下基数排序的原理并研究一下这三个小程序在设计时要的小花招,一定会相信这样设计的算法是最短也是最快的。

```

10 REM sort01
20 S=1000
30 DIM M(S),H(100)
40 FOR I=1 TO S:M(I)=INT(RND(1)*100);NEXT I
50 PRINT “sorting……”;TIME$=“00:00:00”
60 FOR I=1 TO S:H(M(I))=1;NEXT I
70 FOR I=100 TO 0 STEP -1:T=T+H(I);H(I)=T;
NEXT I
80 PRINT “OK! Time:”;TIME$

```

```

90 PRINT “No”, “Scores”, “Order No”
100 FOR I=1 TO S:PRINT I,M(I),H(M(I));NEXT I

10 REM sort02
20 S=1000
30 DIM M(S),H(100)
40 FOR I=1 TO S:M(I)=INT(RND(1)*100);NEXT I
50 PRINT “sorting……”;TIME$=“00:00:00”
60 FOR I=1 TO S:H(M(I))=1+H(M(I));NEXT I
70 FOR I=100 TO 0 STEP -1:T=T+H(I);H(I)=T;
NEXT I
80 PRINT “OK Time:”;TIME$
90 PRINT “No”, “Scores”, “Order No”
100 FOR I=1 TO S:PRINT I,M(I),H(M(I));NEXT I

10 REM sort03
20 S=1000
30 DIM M(S),H(100)
40 FOR I=1 TO S:M(I)=INT(RND(1)*100);NEXT I
50 PRINT “sorting……”;TIME$=“00:00:00”
60 FOR I=1 TO S;K=M(I);M(I)=H(K);H(K)=I;NEXT
I
70 PRINT “OK Time:”;TIME$
80 PRINT “Order No”, “No”, “Scores”
90 FOR I=100 TO 0 STEP -1
100 P=H(I)
110 IF P=0 THEN 140
120 Q=Q+1;PRINT Q,P,I
130 P=M(P);GOTO 110
140 NEXT I
150 END

```

1724 打印机驱动程序行距的修改

武汉铁路分局电子中心(430071) 谭人杰

我国许多 PC 机用户配置了 M-1724 打印机,并利用 CCBios.1 下的 1724P.EXE 汉字驱动程序打印中文报表。为了使报表表格完全封闭,必须对 1724P.EXE 进行修改。

驱动程序 1724P.EXE 利用 M-1724 打印机行宽控制命令“ESC C n”控制走纸,n=6 表示标准行宽(每英寸 6 行),n=8 表示压缩行宽(每英寸 8 行),只要将 1724P.EXE 中的 ESC C 6 改为 ESC C 8,就可以压缩行距。具体修改方法见程序清单。

对于其他打印机的汉字驱动程序,也可以利用这种方法修改行距,首先查询打印机使用手册中的行宽控制命令,然后参照程序清单中的方法,用 DEBUG 对驱动程序进行修改。

C>COPY 1724P. exe 1724P

C>debug 1724P

-S0,fffe B0 06

1F39:2788

1F39:2AAF

ulf39:2aa0

1F39:2AA0 06 PUSH ES

1F39:2AA1 0900 OR [BX+SI],AX

1F39:2AA3 0000 ADD [BX+SI],AL

1F39:2AA5 B01B MOV AL,1B

1F39:2AA7 E8ADFB CALL 2657

1F39:2AAA B043 MOV AL,43

1F39:2AAC E8A8FB CALL 2657

1F39:2AAF B006 MOV AL,06

1F39:2AB1 E8A3FB CALL 2657

1F39:2AB4 B00A MOV AL,0A

1F39:2AB6 E89EFB CALL 2657

1F39:2AB9 B80000 MOV AX,0000

1F39:2ABC B90011 MOV CX,1100

1F39:2ABF BFE200 MOV DI,00E2

-e2ab0

1F39:2AB0 06.08

-w

Writing 2C00 bytes

-q

C>copy 1724p 1724p. exe

加密 WPS 文件的解密

河南郑州中原制药厂设备处(450066) 谷高平 孟建伟

目前,许多微机都配有 Super—WPS 文字处理系统,用该系统的 WPS. COM 软件进行文字处理相当方便,而且可对其编辑的文件进行加密处理,使文件的安全性得到一定的保证。但很多操作人员在给自己的文件加密后却忘记了密码,在备份文件已被破坏的情况下,对再次打开文件束手无策。经过探索,笔者找到了破解加密 WPS 文件的方法,简述如下:

WPS 文件的文件头由 1024 个字节组成,占两个扇区。被加密的 WPS 文件,密码经 WPS. COM 软件转换后被置于该文件文件头第二相关扇区、偏移量为 221~228 这 8 个字节中,未被加密的 WPS 文件,这 8 个字节的内容为 00。如果读出这 8 个字节的内容,再按本文附表即可查出密码的原形。例如:一个名为 GGP. WPS 的加密文件的解密过程如下:

用 PCTools 软件在 File Functions 菜单下对 GGP. WPS 文件进行 Edit 操作,使当前屏幕置于下图状态。查看偏移量 221~228 八个字节可发现,其内容为:

8B 0B 8A 00 00 00 00 00

可见,密码为三位。查本文附表可知密码为: GOW。

File=GGP. WPS

Relative sector 0000001, Clust 02244, Disk Abs Sec 0009130

Disp.....Hex codes.....

000(00) 00 00 00 00 00 00 00—00 00 00 00 00 00 00 00

016(10) 00 00 00 00 00 00 00—00 00 00 00 00 00 00 00

.....

.....

208(D0) 00 00 00 00 00 00 00—00 1A 0C 00 00 8B 0B 8A

224(E0) 00 00 00 00 00 00 E0 01-D0 01 C0 01 B0 01 A0 01

240(F0) 90 01 80 01 70 01 00 CE-F7 C9 BD A3 D7 A3 D0 A3
附表:

1 2 3 4 5 6 7 8	1 2 3 4 5 6 7 8
F1~F12	FF EF DF CF BF AF 9F 8F 7F 6F 5F 4F 3F 2F 1F 0F FE EE DE CE BE AE 9E 8E 7E 6E 5E 4E 3E 2E 1E 0E
ESC	FD ED DD CD BD AD 9D 8D
空! " # \$ % & ' () * + , - . /	7D 6D 5D 4D 3D 2D 1D 0D
0 1 2 3 4 5 6 7	FC EC DC CC BC AC 9C 8C
8 9 : ; < = > ?	7C 6C 5C 4C 3C 2C 1C 0C
@ A B C D E F G	FB EB DB CB BB AB 9B 8B
H I J K L M N O	7B 6B 5B 4B 3B 2B 1B 0B
P Q R S T U V W	FA EA DA CA BA AA 9A 8A
X Y Z [\] ^ _	7A 6A 5A 4A 3A 2A 1A 0A
{ } ~	F9 E9 D9 C9 B9 A9 99 89 79 69 59 49 39 29 19 09 F8 E8 D8 C8 B8 A8 98 88 78 68 58 48 38 28 18 08

电脑病毒入侵报警程序

苏州光明丝织厂微机室(215001) 陶秋刚

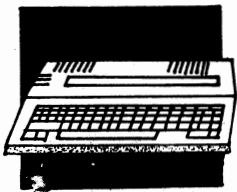
电脑病毒的防不胜防已使人深恶痛绝,在这里笔者向广大用户介绍一种实用的电脑病毒入侵报警程序,使用户能马上发现病毒的入侵并采取相应的措施。

纵观所有文件类电脑病毒,不难发现,这些病毒程序有一个共同特点即它们均是采用修改 DOS 的中断来实现的。于是我们可以这样设想,只要将原先标准的(未被病毒程序修改过的)中断向量表保存起来,并不时同当前正使用的中断向量表作比较,如发现不同便可认为已有病毒入侵了。PC 机的 08H 号中断是一个计时器脉冲中断,每秒产生 18.2 次中断,利用该中断每秒发生 18.2 次的特点不断对新旧向量表进行比较,就起到病毒入侵的发现作用。

笔者基于以上思路设计了一个汇编程序,并在 PC 机上调试通过。该程序能起到对所有文件类病毒入侵的报警功能,如发现病毒的入侵则马上会在屏幕左上角永久显示报警信息。报警信息的显示采用直接写显示缓冲区来实现。在该程序中报警信息为一空白方框。

```
CODE    SEGMENT
        ASSUME CS, CODE, DS, CODE, SS, CODE
        ORG 100H
START:  PUSH CS
        MOV AH, 09
        MOV DX, OFFSET X1
        INT 21H
        MOV AX, 3508H
        INT 21H          ;取原 08H 中断向量
        PUSH ES
        POP DS
        MOV DX, BX
        MOV AX, 256DH
        INT 21H          ;将原 08H 中断置为 6DH 中断
        PUSH CS
        POP DS
        MOV DX, OFFSET DATA2
        MOV AX, 2508H
        INT 21H          ;置现 08H 中断向量
        MOV CX, 68H      ;取标准中断向量表
        XOR AX, AX
        MOV SI, AX
        POP ES
        MOV DS, AX
        MOV DI, OFFSET DATA1
        REP MOVSB
        PUSH CS
```

```
POP DS
MOV DX, OFFSET DATAEND
ADD DX, 300H
INT 27H          ;驻留退出
X1:     DB '病毒报警程序!', 0AH, 0DH
        DB '苏州光明丝织厂微机室', 0AH, 0DH
        DB '          1991. 10. 1', '$'
DATA1:  DB 110 DUP(?),    标准中断向量表
DATA2:  CLD              ;新的 08H 中断入口
        PUSH DS
        PUSH ES
        PUSH AX
        PUSH BX
        PUSH CX
        PUSH DI
        PUSH SI
        PUSH CS
        POP DS
        XOR AX, AX
        MOV ES, AX
        MOV DI, AX
        MOV SI, OFFSET DATA1
        MOV CX, 64H
        REP CMPSB      ;现中断向量表同标准中断向量表比较
        JZ NEXT1
        MOV AX, 0B800H  ;与标准向量表不同,报警处理
        MOV ES, AX
        MOV DI, 0
        MOV SI, OFFSET DATA
        MOV CX, 20H
LOOP1:  PUSH CX
        PUSH DI
        TEST CX, 01
        JZ NEXT2
        ADD DI, 2000H
NEXT2:  MOV CX, 8H
        REP MOVSB
        POP DI
        POP CX
        TEST CX, 01
        JZ X2
        ADD DI, 80
X2:     LOOP LOOP1
NEXT1:  POP SI
        POP DI
        POP CX
        POP BX
        POP AX
        POP ES
        POP DS
        STI
        INT 6DH          ;执行原 08H 中断
DATAEND: IRET
DATA:   DB 0FFH, 0FFH, 0FFH, 0FFH, 0FFH, 0FFH, 0FFH, 0FFH
        DB 30 DUP(0C0H, 00H, 00H, 00H, 00H, 00H, 00H, 03H)
        DB 0FFH, 0FFH, 0FFH, 0FFH, 0FFH, 0FFH, 0FFH, 0FFH
CODE    ENDS
        END START
```



学习机之友

对几种高精度数值计算方法的改进

四川省南充一中(637040) 陈庆祥

我们知道,由于受字长和数据存储方式的限制,计算机输出数值的有效位数只能是固定的几位或十多位(因变量类型而异)。为了获得较长数字运算的准确结果,必须研究高精度的数值运算方法。目前,有关刊物上曾刊登过一些这方面的程序,但这些程序在算法上几乎全部模拟人工笔算,即利用数组来逐位存放多位数的每一位数字和逐位计算的方法。这样尽管也能达到目的。其缺点是处理效率偏低、占用内存空间大,能够处理的数值范围也比较小,可以说这种算法难以发挥计算机的潜力。例如电子与电脑 1989 年第 6 期《一个可以精确求解幂运算结果的程序》,1990 年 10 期《任意两整数的高精度乘法》及 1991 年 4 期《从重赏谈起》,笔者曾对各程序作过试验。第一篇文章中的程序计算 14^{23} 用 55 秒(在 Apple I 上运行,下同),计算 50^{23} 时便发生溢出错误,事实上,它仅能计算结果不超过 40 位的幂。第二篇文章的程序在计算 13 位数与 11 位数的乘积时用 12 秒,计算 30 位与 21 位数的乘积时用 57 秒,计算两个 100 位数的乘积要用近两小时。第三篇文章的程序在计算麦粒总数 $\sum_{i=0}^{63} 2^i$ 时用了 1 分半钟,比笔算快不了多少。

其实,在处理多位数的乘法、幂与阶乘等运算时,我们完全可以采取将数字分段存储和运算的新方法,充分利用整型和实型变量的最多有效位数,这样不仅可以显著地提高处理效率,使计算的耗时量减少为逐位计算的几十分之一,还可以大大减少内存空间的占用,扩大数据处理的范围。

本文后面是笔者用上述思想设计的两个高精度数值运算程序,分别用来计算乘积与幂。容易看出,这两个程序比前面提到的三篇文章中的程序简练得多,但是其处理效率却高出几十倍。而占用内存却大为减少,数值处理范围也相应扩大。如程序 1 计算两个 20 以下的整数乘积仅用零点几秒,两个 30 位数的乘积用 2—3 秒,计算两个 100 位整数的积也不过用 30 秒。程序 2 求 14^{23} 时用 2 秒,计算 9876^{23} 时用 8 秒,求 1991^{100} 的 330 位数也仅用 90 秒。该程序在西文状态下可以计算近 7 万位的数字结果。

这两个程序的算法主要技巧是在数字分段处理上。程序 1 采用字符串补充分段的方法(即凑成 4 的倍数);程序 2 对各次中间计算结果的数字分段比较困难一些,因为其长度是变化的,这里采用了利用对数测出各次积的动态长度,然后再分段存储、计算的方法。用一个整型数组元素存放四位数字,每次具体计算的数

字便不会超过九位。保证了数据计算的准确性。

最后要提到的是算法上的另一种变通处理。上述第三篇文章中求 $\sum_{i=0}^{63} 2^i$ 的问题,因涉及到高精度的幂运算与加法运算,故算法复杂,程序很长且运行费时,其实,用等比数列的求和公式容易推知: $\sum_{i=0}^n 2^i = 2^{n+1} - 1$, 即只需求 $2^{64} - 1$ 即可。这样一来。只要将程序 2 的 70 语句略加改动,便可迅速求出 $\sum_{i=0}^{63} 2^i$ 或 $\sum_{i=0}^{73} 2^i$, 耗时不过 2 秒左右。如果再变通一下,求 $256^8 - 1$, 因为 $2^{64} - 1 = (2^8)^8 - 1 = 256^8 - 1$, 输入 256 与 8 后立即可打印出准确的结果。由此不难看出,灵活地处理算法对于提高程序质量有着非常重要的意义。

程序 1

```
10 REM 正整数的快速高精度乘法程序
20 INPUT "被乘数,乘数=";A$(1),A$(2);L(1)=INT
  ((LEN(A$(1))+3)/4);L(2)=INT((LEN(A$(2))
  +3)/4)
25 IF A$(1)="0" THEN END
27 CALL-198:PRINT"乘积=";
30 S=L(1)+L(2);W=(S+ABS(L(1)-L(2)))/2;DIM
  A%(2,W),B%(S);E=10000
40 FOR I=1 TO 2:A$(I)=RIGHT$("000"+A$(I),L
  (I)*4)
50 FOR J=1 TO L(1)
60 A%(I,J)=VAL(MID$(A$(I),4*(L(I)-J)+1,4))
70 NEXT J:NEXT
80 FOR I=1 TO L(1);H=0
90 FOR J=1 TO L(2);K=I+J-1
100 B=B%(K)+A%(I,J)*A%(2,J)+H
110 H=INT(B/E);B%(K)=B-H*E
120 NEXT J:B%(K+1)=H;NEXT
130 K=K-(B%(K+1)=0);PRINT B%(K+1);
140 FOR I=K TO 1 STEP -1;PRINT B%(I);NEXT
150 CALL-198:PRINT:RUN
被乘数,乘数=8521476391472,58735419635
乘积=500512491762853455352720
被乘数,乘数=369258147369258147147258369147,
147258369963852741123
乘积=543763528774697301272742665307220307963856
1332081
被乘数,乘数=0,0
```

程序 2

```
5 REM 快速求解幂运算精确结果的 BASIC 程序
10 INPUT "A,N=";A,N;IF A<>INT(A)OR A<=0
```



```

OR N<=0 THEN END
20 W= LOG(A)/LOG(10);DS%=N*W/4+1;DIM X%(DS%)
30 X%(1)=A;B=1E4;F=4;C=W+F;L$="000"
40 FOR Y=2 TO N;C=C+W
50 FOR I=1 TO C/F;D=A*X%(I)+E%
60 E%=D/B;X%(I)=D-B*E%;NEXT;NEXT
70 PRINT A;"^";N;"=";X%(I-1);
80 IF I>2 THEN FOR J=I-2 TO 1 STEP -1;PRINT RIGHT$(L$+STR$(X%(J)),4);;NEXT
90 CALL-198;PRINT;RUN
]RUN
A,N=14,32
14^23=229585692886981495482220544
A,N=9876,32

```

```

9876^32=6708023885067056779126280484374086768
00621039896345384011301752850609595990813326500520305
68130359850101425909428305449373925376
A,N=0,0
修改 70 语句后计算《从“重赏”谈起》中的麦粒数,用 5
秒左右或 0.5 秒即求出。
]70 X%(1)=X%(1)-1;PRINT "S=";X%(I-1);
]RUN
A,N=2,64
S=18446744073709551615
A,N=2,73
S=9444732965739290427391
A,N=256,8
S=18446744073709551615
A,N=0,0

```

CEC—I 自造字键盘辅助点阵生成程序

襄樊铁路运输技校(441001) 汤永进

笔者编写了一个工具程序,可象键盘绘图一样方便地生成字模点阵。该程序的主要变量和处理过程简介如下:

A0 为字库存放首址,可按所用调字程序的参数要求置为用户内存区中任一地址(这里配合 91 年 6 期博文置为 \$7000);A 为当前所造字的点阵数据存放首址,每字数据占 32 个字节。L0 为字库总长度,可按需要在内存开销许可的范围内随意增减;INT(L0/32)为该字库最多可容字数。

A 数组存放 16×16 点阵的映象(0 或 1),其中左下标 1~16 为点阵行序,中下标 0~1 为每行左右组别(一行 16 个点要分左右两组各存 1 字节),右下标 0~7 为每组列序。N 数组存放 32 字节的点阵数据码(0~255),其中奇数下标元素顺次存左半点阵数据,偶数下标元素顺次存右半点阵数据。

程序 10 行读入已建立的字库(如无则下滑运行),20~45 行印出标题并在屏幕中间提供一有 16×16 格位的方框,50~70 行建立系统变量;80 行按提示输入所造字的代码后,90 行算出该字首址;100 行将对应内存中可能存在的原造旧字的点阵数据读入 N 数组,110~180 行将这些数据解拆分送 A 数组,并在方框中显映出字模。

190 行设定各绘图中间变量的初始值,200~290 行进入键盘辅助点阵描绘,将各点描绘结果随时存录于 A 数组对应元素。描绘时按四个箭头键在方框中移动光点,按空格键决定下笔还是擦除(同软汉字的造字操作相似)。造字过程中若欲放弃该字重新择码另造,可按 ESC 键;若只擦光底板重造该字则可按 TAB 键。字造好后按回车键,300~330 行便将 A 数组中数据归并到 N 数组中去,340 行将其写入内存,接着 350~

370 行决定是否继续造字,若结束则 380 行将内存字库全部存盘。

子程序 500~520 行行使“黑板擦”功能,负责清画框和清数组;600~620 行作为“笔尖”,负责绘点。

```

5 ONERR GOTO 20
10 PRINT CHR$(4)"BLOAD CEC-ZIKU"
20 PRINT CHR$(4)"PR #3";PRINT
30 HGR2;VTAB 2;HTAB 12;PRINT"CEC-I 造字";HTAB 9;PRINT"字模点阵生成系统"
40 HCOLOR=3;HPOINT 94,60 TO 160,60 TO 160,126 TO 94,126 TO 94,60
45 HPOINT 127,61,HPOINT 95,93;HPOINT 159,93,HPOINT 127,125
50 A0=28672;L0=4096;REM.$7000&.$1000
60 G$=CHR$(7)+CHR$(7);C$=CHR$(11)
70 GOSUB 500;PRINT G$;
80 VTAB 9;HTAB 8;PRINT C$"该字代码(1-";INT(L0/32);;INPUT"(";N
90 A=A0+(N-1)*32
100 FOR I=1 TO 32;N(I)=PEEK(A+I-1);NEXT
110 HCOLOR=3;Y=59
120 FOR I=1 TO 16;L=I*2-1
125 X=93;Y=Y+4
130 FOR J=0 TO 1;N%=N(L+J)
140 FOR K=0 TO 7;A%=N%/2
145 A(I,J,K)=N%-A%*2
150 IF A% THEN N%=A%;NEXT
160 FOR K=7 TO 0 STEP -1;X=X+4
170 ON A(I,J,K)=1 GOSUB 600;NEXT
180 NEXT J,I;PRINT G$;
190 X=97;Y=63;X0=97;Y0=63;H=1;V=1;C=0

```

(转第 20 页)

二维函数曲线拟合

上海交通大学 90111 班(200240) 李 涛

我们在有关数据采集和处理的过程中,经常遇到二维函数之间求解 $y=f(x)$ 的问题,本文是根据样本数据求解一元高次方程:

$$y=a_0+a_1x+a_2x^2+a_3x^3+\cdots+a_nx^n;$$

从而得出二维平面中任意曲线的函数关系式。本程序适合于大批量数据处理,这里只介绍程序使用方法。

程序 1 是统计计算程序。首先输入所需回归次数,然后键入样本数据文件的文件名。该数据文件由程序 2 生成。之后计算机计算出各项系数。按任一键后进入主菜单,对计算出来的函数式作进一步处理。共有 4 项选择:(1)数值计算。依照所得函数式对某一 x 值求解 y 值,这可用于对未知参数预测。(2)显示函数曲线。首先在提示下输入 x 、 y 坐标的作图范围,程序会依照该数据计算出屏幕上每一点所代表的 x 、 y 坐标宽度,并显示出来。此举目的在于进行定量观测。计算机在规定的范围内绘出曲线后,还会将该范围内的原始离散数据点显示在屏幕上,以供比较拟合程度。(3)调入另一批数据。(4)显示拟合出的各项系数。另外,在主菜单下按 H 键会进入高分辨率显示状态,以便随时观察已经绘好的曲线;按 T 键则回到文本状态。

程序 2 是数据文件生成程序。它可将输入的样本资料生成一个以 .DAT 为后缀的 T 型文件,供程序 1 调用。它还可对数据文件进行增加或删除等有关处理。该程序运行每一步均有提示。

程序 1

```
5 DIM A(20,20),B(40),C(20),X(20),Y(20)
6 D$=CHR$(4)
10 TEXT;HOME;VTAB 10;INPUT"WQZ:";W;N=W+1;S=1;P=0
15 PRINT;INPUT"Input file name:";A$
17 PRINT D$;"OPEN";A$;" .DAT"
18 PRINT D$;"READ";A$;" .DAT"
19 INPUT X;INPUT Y;IF X=0 AND Y=0 THEN 42
20 FOR I=1 TO W*2:B(I)=B(I)+X^I;NEXT I
30 FOR J=0 TO W;C(J)=C(J)+Y*X^J;NEXT J
32 X(S)=X;Y(S)=Y
35 PRINT "x";S;"=";X;" y";S;"=";Y
36 S=S+1;GOTO 19
42 PRINT D$;"CLOSE";A$;" .DAT"
43 GET A$;X(S)=0;Y(S)=0
45 HOME;VTAB 6;HTAB 4;PRINT"Calculating..."
50 FOR I=0 TO W
60 FOR J=0 TO W;A(I,J)=B(J+P);NEXT J
70 P=P+1;NEXT I
80 A(0,0)=S
90 FOR I=0 TO W;A(I,N)=C(I);NEXT I
200 FOR I=0 TO W;Q=I;P=I;E=A(I,1)
```

```
210 FOR J=0 TO W
220 IF J=I OR A(J,Q)=0 THEN 280
240 R=A(J,Q)/A(I,Q)
250 FOR K=0 TO N;A(J,K)=A(J,K)-A(I,K)*R;
NEXT K
280 NEXT J;A(N,I)=Q
300 NEXT I
310 FOR I=0 TO W;Q=A(N,I);A(N,I)=A(I,N)/A(I,Q);PRINT "a";I;"=";A(N,I);NEXT
311 PRINT;PRINT"Hit any key to continue...";GET A$
312 HOME;VTAB 6
313 PRINT "(1) CALCULATE"
314 PRINT "(2) SHOW CURVE"
315 PRINT "(3) INPUT NEW DATA"
317 PRINT "(4) SHOW COEF FICIENTS"
320 GET A$;IF A$="2" THEN 400
321 IF A$="4" THEN 600
322 IF A$="1" THEN 345
324 IF A$="3" THEN RUN
325 IF A$="H" THEN POKE-16304,0;POKE-16299,0;POKE-16297,0
326 IF A$="T" THEN TEXT;GOTO 312
345 INPUT"x:";X;Y=0
350 GOSUB 500
360 PRINT "y=";Y
370 PRINT"Continue? (Y/N)";GET A$
375 PRINT;PRINT
380 IF A$="N" THEN 312
390 GOTO 345
400 HOME;VTAB 4;PRINT "Plotting:"
410 PRINT;PRINT;PRINT "Range"
420 INPUT"Xmin:";XN;INPUT "max:";XM;XS=(XM-XN)/279
422 PRINT "X scale:";XS
424 INPUT"Ymin:";YN;INPUT "max:";YM;YS=(YM-YN)/191
426 PRINT"Y scale:";YS
428 GET A$;HGR2,HCOLOR=3
430 FOR I=XN TO XM STEP XS;Y=0;X=I;GOSUB 500
432 IF Y<YN OR Y>YM THEN 436
434 HPLLOT (X-XN)/XS,191-(Y-YN)/YS
436 NEXT
438 S=1
439 IF X(S)=0 AND Y(S)=0 THEN 446
440 IF X(S)<XN OR X(S)>XM THEN 444
441 IF Y(S)<YN OR Y(S)>YM THEN 444
442 HPLLOT (X(S)-XN)/XS,191-(Y(S)-YN)/YS
443 MUSIC 88,45
444 S=S+1;GOTO 439
```

```

446 MUSIC 128,45
450 GET A$;TEXT;HOME;GOTO 312
500 FOR K=0 TO W:Y=Y+A(N,K)*X^K;NEXT
510 RETURN
600 TEXT;VTAB 6;FOR K=0 TO W:PRINT "a";K="";A
(N,K);NEXT
610 GET A$;GOTO 312

```

程序 2

```

10 D$=CHR$(4);DIM X(100),Y(100)
15 HOME
20 FOR I=1 TO 5:READ V,H,A$;VTAB V;HTAB H;
PRINT A$;NEXT
30 DATA 4,6,"DATA EDITOR VER 1.0"
40 DATA 6,4,"(1) INPUT NEW DATA"
50 DATA 8,4,"(2) APPEND DATA"
60 DATA 10,4,"(3) DELETE DATA"
70 DATA 12,4,"(4) QUIT"
100 GET A$;IF A$="1" THEN K=0;GOTO 200
110 IF A$="2" THEN K=1;GOTO 200
115 IF A$="3" THEN 400
120 IF A$="4" THEN END
130 GOTO 100
200 HOME;VTAB 6;INPUT "Input data file name:";A$

```

```

210 HOME;VTAB 3;HTAB 4;PRINT "Inputing data ...";
PRINT
220 HTAB 4;PRINT "When over, hit<A>key..."
230 S=1;PRINT
232 PRINT"x";S;" ";:INPUT X(S)
234 PRINT"y";S;" ";:INPUT Y(S)
236 PRINT"x=";X(S);"y=";Y(S)
238 PRINT"Are you sure? (Y.N.A)"
239 GET B$
240 IF B$="N" THEN 232
242 IF B$="Y" THEN S=S+1;GOTO 232
244 IF B$="A" THEN 250
245 GOTO 239
250 HOME;VTAB 6;HTAB 6;PRINT"Saving to disk..."
260 PRINT D$;"OPEN";A$;". DAT"
262 IF K=1 THEN PRINT D$;"APPEND";A$;". DAT"
265 PRINT D$;"WRITE";A$;". DAT"
270 FOR I=1 TO S:PRINT X(I);PRINT Y(I);NEXT
273 PRINT 0;PRINT 0
275 PRINT D$;"CLOSE";A$;". DAT"
280 RUN
400 HOME;VTAB 6;INPUT "Input file name :";A$
410 PRINT D$;"DELETE";A$;". DAT"
420 RUN

```

在 Applesoft 系统下实现 COMMON 功能

临汾山西师大实验中学(041000) 安赵根

在 CP/M 操作系统支持下的 MBASIC 语言有一个 COMMON 语句,在连接程序时与 CHAIN 程序连用,把指定数组的值传送给被连接的程序使用。而 Applesoft 系统却不具备这种功能(它只能把全部数组都传送下去),这对于被连接的程序来说是对内存空间的极大浪费。

为此笔者根据 SOFTBASIC 变量表的特点编写了一段汇编语言子程序,用于对不需传送的数组从变量表中清除,在 APPLESOFT 系统下实现了与 COMMON 类似的功能。程序采用了相对寻址方式,可调入内存任何位置以便于使用。对一个数组进行清除后,变量表中便没有了这个数组,它所占用的存储单元被后面的数组使用,变量表最后一部分单元被释放。

当某数组不需传送而进行清除时,可先用 POKE 语句将数组类型代码(实型为 0,整型为 % 的 ASC 码、字符型为 \$ 的 ASC 码)送入 253 单元;将数组名首字符的 ASC 码送入 254 单元,将数组名次字符的 ASC 码送入 255 单元(若名只有一个字母,则将 0 送入 255 单元),然后用 CALL 语句调用该汇编子程序即可。如,欲清除数 A\$(20),则可用以下语句实现(子程序首址为 \$9000)。

```

100 POKE 253,ASC("$");POKE 254,ASC("A");POKE
255,0;CALL 9*16^3
9000- A5 FD F0 13 38 E9 24 F0
9008- 07 18 A5 FE 69 80 85 FE
9010- 18 A5 FF 69 80 85 FF A0
9018- 02 A9 00 91 6D C8 91 6D
9020- A5 6B 85 F9 A5 6C 85 FA
9028- A0 00 A5 FE 38 F1 F9 D0
9030- 0A C8 A5 FF 38 F1 F9 D0
9038- 02 F0 18 A0 02 18 B1 F9
9040- 65 F9 85 FD C8 B1 F9 65
9048- FA 85 FA A5 FD 85 F9 A9
9050- 00 F0 D5 A0 02 18 B1 F9
9058- 65 F9 85 FB C8 B1 F9 65
9060- FA 85 FC A5 6D 85 FD A5
9068- 6E 85 FE A0 00 A9 00 91
9070- F9 C8 91 F9 A0 00 B1 FB
9078- 91 F9 E6 F9 D0 02 E6 FA
9080- A5 FB C5 FD A5 FC E5 FE
9088- E6 FB D0 02 E6 FC 90 E4
9090- A5 F9 85 6D A5 FA 85 6E
9098- A5 6D F0 03 C6 6D 60 A9
90A0- FF 85 6D C6 6E 60

```


通用打印课表程序

山东青岛九中微机室(266011) 程建伟

本程序是利用中华学习机的汉字与表格处理功能,打印出样式美观,形式多样的课表,且可大可小,使用方便。程序中的操作也非常简单,每一步都有善意的提示,等待您键入各课目的代码,如下所示:

A-政治 B-语文 C-代数
D-几何 E-英语 F-物理
G-化学 H-历史 I-地理
J-体育 K-生理 L-空白
M-数学 N-语音 O-生物

(用户欲增改课目,只需修改第 20 和 340 句)

输入课目代码时,每天占一行,六行输入完后,程序将自动存档,以备下次使用。程序输出可任选,如改变第 130 句 1659 单元的值,可改变其大小,将第 520 句中的变量 I 与 J 交换,可输出纵向排列的课表等等。

```
10 INPUT "请问用原记录还是输入新纪录(A/B)"; A$
20 DIM B$(6), C$(6,6), N$(15)
30 D$ = CHR$(13) + CHR$(4)
40 FOR I=1 TO 6: READ B$(I); NEXT; FOR I=1 TO 15: READ N$(I); NEXT
50 IF A$ = "A" THEN GOSUB 600
60 IF A$ = "B" THEN GOSUB 700
70 IF A$ <> "A" AND A$ <> "B" THEN RUN
80 INPUT "打印吗(Y/N)"; R$
90 IF R$ = "Y" THEN 110
100 GOTO 170
110 PRINT "准备好了吗,按任意键开始"
120 GET K$; GET K$; PRINT "已经开始了,请等候"
130 POKE 1659,2; POKE 1687,0
150 POKE 1915,0; POKE 2043,40
170 PRINT SPC(23); "课程表"; SPC(12); "92.9"
180 PRINT SPC(7); "
190 PRINT SPC(7); "
200 FOR I=1 TO 6
210 GOSUB 400
220 GOSUB 500
230 NEXT
240 PRINT SPC(7); "
250 PRINT SPC(18); "书山有路勤为径,学海无边苦作舟"
260 PRINT; PRINT; PRINT; PRINT
270 POKE 1659,0
280 INPUT "还打印吗(Y/N)"; R$
290 IF R$ = "Y" THEN 110
300 PRINT "谢谢合作,再见!"
310 END
320 DATA "一","二","三","四","五","六"
```

```
330 DATA "政治","语文","代数","几何","英语","物理","化学","历史","地理","体育","生理",""
340 DATA "数学","语音","生物"
400 PRINT SPC(7); "
410 RETURN
500 PRINT SPC(7); "
510 FOR J=1 TO 6
520 PRINT C$(I,J); "
530 NEXT; PRINT
540 RETURN
600 INPUT "请输入班级"; AA
610 PRINT D$; "OPEN L. TAB-"; AA
620 PRINT D$; "READ L. TAB-"; AA
630 FOR I=1 TO 6; FOR J=1 TO 6
640 INPUT X$
650 C$(I,J) = X$
660 IF X$ = "" THEN C$(I,J) = "
670 NEXT; NEXT
680 PRINT D$; "CLOSE"
690 RETURN
700 DIM L$(6)
710 PRINT "请输入课程代号"
720 FOR I=1 TO 6
730 PRINT "星期"; B$(I); INPUT L$(I)
740 NEXT
750 FOR I=1 TO 6; FOR J=1 TO 6
760 C$(I,J) = N$(ASC(MID$(L$(I),J,1))-64)
770 NEXT; NEXT
780 INPUT "请输入班级"; AA
790 PRINT "请放入数据盘,按任意键"
800 GET K$; GET K$
810 PRINT "请稍候..."
820 PRINT D$; "OPEN L. TAB-"; AA
830 PRINT D$; "WRITE L. TAB-"; AA
840 FOR I=1 TO 6; FOR J=1 TO 6
850 PRINT C$(I,J)
860 NEXT; NEXT
870 PRINT D$; "CLOSE"
880 RETURN
```

]RUN

请问用原记录还是输入新纪录(A/B)B

请输入课程代号

星期一? BJHMEI

星期二? MEFAGI

星期三? BFMEBB

星期四? GJMNLL

星期五? EBAIFM

星期六? FMGBHE

请输入班级 44

请放入数据盘,按任意键

请稍候...

打印吗(Y/N)Y

准备好了吗,按任意键开始

已经开始了,请等候

	1	2	3	4	5	6
一	语文	体育	历史	数学	英语	地理
二	数学	英语	物理	政治	化学	地理
三	语文	物理	数学	英语	语文	语文
四	化学	体育	数学	语文		
五	英语	语文	政治	地理	物理	数学
六	物理	数学	化学	语文	历史	英语

书山有路勤为径,学海无边苦作舟

CP/M 系统文件的恢复方法

四川绵阳南山中学(621000) 李 齐

当你辛辛苦苦在 CP/M 操作系统下编制的实用程序或是生成的重要数据文件遭到了意外删除,一定十分痛心。不过只要你的文件删除后该盘没有存入新的文件,在读完本文后,你就等于具有了使文件起死回生的能力。

用 NIBBLES AWAY 等软件的扇区编辑功能将被删文件所在盘的 03 道读出,可以清楚的看到 CP/M 的目录结构。

CP/M 的目录区依次安排在 03 道的 00、03、06、09、0C、0F 扇区,每个目录占连续的 32 个字节,00 字节为文件所在用户区号,当一个文件被删除后,其值被置为 E5H。

01—08 字节是文件主名,如不足 8 字节,未使用

部分则填以 20H(空格)

09—11 字节为文件扩充名。

13—14 字节未用,值为 00H。

15 字节是文件长度,以记录为单位。

16—32 字节表明该文件在磁盘中的位置,未使用的部分值为 00H。

由上可知,要恢复一个被删除的文件,只需将 03 道读出,找到该文件的索引,将其 00 字节的 E5H 改为该文件所在的用户区号,(一般可改为 00),再写回 03 道,即可大功告成。

这时回到 CP/M 操作系统。哈!,失掉的文件又回来了!

(接第 16 页)

```

200 HCOLOR=3;HPLLOT X,Y
210 P=PEEK(-16384);IF P<128 THEN 210
215 POKE-16368,0;P=P-128
220 HCOLOR=0;HPLLOT X0,Y0
230 ON P=9 GOSUB 500
240 ON P=13 GOTO 300;ON P=27 GOTO 70;ON P=32
    GOTO 280
250 XX=(P=21)*(H<16)-(P=8)*(H>1)
255 X=X0+XX*4;H=H+XX
260 YY=(P=10)*(V<16)-(P=11)*(V>1)
265 Y=Y0+YY*4;V=V+YY
270 X0=X;Y0=Y;GOTO 200
280 T=H>8;U=8+8*T-H
285 C=ABS(A(V,T,U)-1);A(V,T,U)=C
290 HCOLOR=C*3;GOSUB 600;GOTO 200
300 FOR I=1 TO 16;L=I*2-1
310 FOR J=0 TO 1;L=L+J;N(L)=0;E=0.5

```

```

320 FOR K=0 TO 7;E=E*2;N(L)=N(L)+A(I,J,K)*E
330 NEXT;NEXT X=PEEK(49200);NEXT
340 FOR I=1 TO 32;POKE A+I-1,N(I);NEXT
350 VTAB 9;PRINT TAB(9)"继续造字? (Y/N)"C$G$;
360 GET X$;IF X$="Y" THEN 70
370 IF X$<>"N"THEN 360
380 PRINT X$;PRINT CHR$(4)"BSAVE CEC-ZIKU,A"
    A0;,"L"L0
400 END
500 HCOLOR=0;FOR I=96 TO 158;HPLLOT I,62 TO I,
    124;NEXT
510 POKE 109,PEEK(107);POKE 110,PEEK(108);DIM A
    (16,1,7);N(32)
520 RETURN
600 FOR II=X-1 TO X+1;FOR JJ=Y-1 TO Y+1
610 HPLLOT II,JJ;NEXT;NEXT
620 RETURN

```

第八讲 子程序设计

南京大学大气科学系(210008) 朱国江

§ 4. 子程序设计

在程序的编制过程中,常常遇到相同的计算或操作,或者,在一个程序中要多次处理重复计算的问题,如代码转换、数据传送、图形处理、函数运算等等。

可以把这些相同的部份编制成一个独立的程序段,当遇到相同的计算或操作时,就转入这个程序段,从而省去对这些相同部份重新编制程序的工作。

我们把在功能上具有一定独立性,能够完成一定计算或操作,并在程序的不同部位要被经常调用的程序段称作子程序。而把调用子程序的程序称为主程序。

大多数程序都是由一个主程序和几个子程序组成。在程序的适当位置安排调用点,就可以转向子程序,简称为转子。而当子程序执行之后再返回到主程序,简称为返主。转子指令 JSR 和返回指令 RTS,可以实现主程序对子程序的正确调用和子程序到主程序的自动返回。

无条件转移指令控制程序转出以后就不再返回了;而转子指令控制程序转向子程序以后,当子程序执行完毕时还要返回主程序被打断处(叫做断点),即返回到调用它的指令的下一条指令去执行程序。保证子程序正确返回的是返回指令 RTS。

为了准确地完成返回“断点”的任务,调用子程序的指令必须能保存断点;而子程序最后执行的必须是将断点送入 PC 寄存器(程序计数器)的指令。JSR 指令具有保护断点的能力,它将断点地址-1 后送入堆栈 S 保存再转向子程序,而 RTS 指令正好完成自堆栈 S 中取一个地址,加 1 后赋给 PC 寄存器的操作,从而使子程序返回到主程序原来的断点处继续执行。由此可见,JSR 和 RTS 的作用是转移程序的控制权;先转到子程序,再返回主程序。

上述概念较多,但很重要。否则就不能保证主程序对子程序的正确调用;也不能保证子程序到主程序的自动返回,特别是处理多个子程序的嵌套时,对于主程序和子程序的合理衔接问题,都将碰到麻烦。

让我们通过一些实例,来介绍子程序设计吧!读者也许会从这些实例中掌握子程序设计的方法和技巧。

[例 1] 数据块转移

试用子程序改写简单程序设计中[例 1]的数据块转移程序,即要求将从 \$2000 开始的 5 个连续字节的内容转移到 \$4000 开始的 5 个连续字节内。

分析:

• 整个程序设计可以分成一个主程序和一个子程序两个部份。

• 主程序结构十分简单,首先将转移的个数(可以根据题意要求任意定,本例是 5 个)安排在累加器 A 中,然后调用子程序,结束。一共只要安排三条指令。

• 子程序的任务是完成 N 个数据块的传送。取数、送数采用 Y 寄存器的绝对变址寻址方式指令,通过修改 Y 寄存器中的值(放 3 数据长度),达到修改地址指针的目的,并用 BPL 指令控制数据是否取、送完。为了完成 N 个数据的传送,采用循环的方法。特别注意子程序的结束处不应忘记放一条返回指令 RTS。

由上分析,容易写出程序 8.1:

程序 8.1

```
1000- A9 05    LDA # $05 } 主程序
1002- 20 06 10 JSR $1006 }
1005- 60       RTS
1006- A8       TAY
1007- 88       DEY
1008- B9 00 20 LDA $2000,Y } 子程序
100B- 99 00 40 STA $4000,Y }
100E- 88       DEY
100F- 10 F7    BPL $1008 }
1011- 60       RTS
```

运行前: * 2000:01 02 03 04 05

运行后: * 1000G✓

* 4000.4004✓

* 4000-01 02 03 04 05

比较程序 8.1 和简单程序设计中的[例 1]程序 6.1,可以看出:

• 程序 8.1 有更大的灵活性。因为只要改变数据块长度(即 LDA # \$nn 中的立即数),即可完成 N 个数(不超过 FF)的数据块传送而无需改动源程序。

• 程序 8.1 简捷和通用。程序 6.1 若要处理 250 个字节的数据块传送,至少要写上 500 条指令,而程序 8.1 除改变数据块长度外,源程序不加一条指令。

由此可知,程序编制中应注意简捷以节省内存,灵活以方便修改,通用以完成各种类似问题的处理。所有这些程序都是程序设计的质量要求。

[例 2] 确定正数的个数

数据长度放在 \$6000 单元,数据从 \$6001 开始存放,要求统计出数据块中正数的个数。

这个问题在循环程序设计[例 3]中作过类似的介绍,那里是确定负数的个数,因而有关程序设计思想的分析,大同小异,不再重复。但我们这里采用子程序设计方法,请注意结构和指令选用上的差别。

程序 8.2


```

1000- AD 00 60 LDA $ 6000
1003- A0 00 LDY # $ 00
1005- 20 09 10 JSR $ 1009
1008- 60 RTS

```

主程序

```

1009- AA TAX
100A- BD 00 60 LDA $ 6000,X
100D- 30 01 BMI $ 1010
100F- C8 INY
1010- CA DEX
1011- D0 F7 BNE $ 100A
1013- 84 06 STY $ 06
1015- 98 TYA
1016- 20 DA FD JSR $ FDDA
1019- 60 RTS

```

子程序

运行前: * 6000:06 68 F2 87 30 59 2A

运行后: * 1000G

* 04

或: * 1000G

* 6

* 0006-04

[例 3] 找最大值

找一个无符号数据块中的最大值,数据块的长度放在 \$ 6000 单元,数据块起始地址为 \$ 6001,最大值存于 \$ 08 单元。

求最大值问题,我们在循环程序的设计中(程序 7.3)作过介绍,程序 7.3 虽有简短易懂的特点,但通用性不好,数据量一大即不能正确运行。故这里再介绍一个找最大值的程序 8.3。

程序 8.3

```

1000- A9 01 LDA # $ 01
1002- 85 06 STA $ 06
1004- A9 60 LDA # $ 60
1006- 85 07 STA $ 07
1008- AC 00 60 LDY $ 6000 ;数据块长度存 Y 寄存器
100B- 20 11 10 JSR $ 1011 ;转子程序
100E- 85 08 STA $ 08 ;存结果
1010- 60 RTS
1011- A9 00 LDA # $ 00 ;设最大值=0
1013- 88 DEY ;Y-1
1014- 08 PHP ;标志寄存器进栈,保护 Z 标志
1015- D1 06 CMP ( $ 06 ),Y
1017- B0 02 BCS $ 101B ;下一个数>最大值,转
1019- B1 06 LDA ( $ 06 ),Y ;是换最大值→A
101B- 28 PLP ;标志寄存器出栈,
101C- D0 F5 BNE $ 1013 ;判 Y-1 是否为 0
101E- 60 RTS ;否,循环;是,返回主程序

```

运行前: * 6000:09 ;数据块长度

* 6001:75 44 A0 BF 54 C4 8F DD 20 ;源数据

运行后: * 1000G

* 8

* 0008-DD 找出最大值

现在,我们介绍程序 8.3 的设计思想:

• 主程序放在 \$ 1000—\$ 1010 单元,子程序放在 \$ 1011—\$ 101E 单元。在主程序中,数据块首地址放

在一个特定的地方,例如,放在 \$ 06、\$ 07 单元中,前者存数据块首地址低 8 位,后者放数据块首地址高 8 位,而将数据块长度放在 Y 寄存器中。然后调用子程序,存结果结束。子程序实际上是一个完成“找最大值”这个操作的独立程序段,任务就是找最大值。

• 在子程序中,把所得最大值结束放在 A 中,这是为了主程序、子程序的衔接所做的规定,有了这种规定,对于另一个起始地址,另一个长度的数据块,找最大值时只需仿照此例,把起始地址送入 \$ 07 和 \$ 06 单元,把长度送入 Y 寄存器,然后转入“找最大值”子程序,即可在累加器 A 中得到最大值,而无需对子程序作任何改动或修正。使这个子程序具有了通用性。

[例 4] 搬家子程序

这里介绍一个实用的搬家子程序,它可以方便地实现数据块的搬移任务。无论对单个数据块或是多个数据块;无论对少量数据还是大量数据;无论对数据记录还是图形信息都可以方便地实现搬移。

程序 8.4

```

1000- A0 00 LDY # $ 00
1002- A9 00 LDA # $ 00
1004- 85 3C STA $ 3C
1006- A9 60 LDA # $ 60
1008- 85 3D STA $ 3D
100A- A9 10 LDA # $ 10
100C- 85 3E STA # $ 3E
100E- A9 60 LDA # $ 60
1010- 85 3F STA $ 3F
1012- A9 00 LDA # $ 00
1014- 85 42 STA $ 42
1016- A9 70 LDA # $ 70
1018- 85 43 STA $ 43
101A- 20 2C FE JSR $ FE2C
101D- 60 RTS

```

将源数据块起始地址放在 \$ 3C 和 \$ 3D 单元,源数据块结束地址放在 \$ 3E 和 \$ 3F 单元,数据块的目的地址放在 \$ 42 和 \$ 43 单元,都是低位地址在前,高位地址在后,然后调用监控中的搬家子程序(入口地址为 \$ FE2C),即可完成单个数据块的搬移任务。但一定不要忘记应用本程序 8.4 时,注意将 # \$ 00 值放入 Y 寄存器中。

本例是将 \$ 6000—\$ 6010 单元的数据搬移至 \$ 7000—\$ 7010 单元中去。对于任意单个数据块的搬移,只需改变 \$ 3C—\$ 3F 及 \$ 42—\$ 43 单元中的内容。例如,将高分辨率第 1 页的图形搬至高分辨第 2 页图形区,即将 \$ 2000—\$ 3FFF 单元内容搬至 \$ 4000—\$ 5FFF 单元中去,只要改动以下单元值:

```

1007-20
100B-FF
100F-3F
1017-40

```

请注意搬移的信息量,它搬移了 8192 个存储单元的信息。而且其搬迁速度也很快,几乎瞬间完成。这是由于机器语言执行速度特别快的特点决定的。

对于多个数据块的搬移,仅仅用上述子程序 8.4 还不够,因为多个数据块的源地址有几处,搬移到的目的地址也有几个,从哪几个地方搬,结果又放在何处,必须随时调整地址指针。我们可以编一个主程序,设置好调整的地址指针,然后不断调用上述子程序 8.4,使之完成多个数据块的搬移任务。请看下面实例。

[例 5] 三个数据块的搬移

将 \$ 6000—\$ 600F, \$ 6100—\$ 610F, \$ 6200—\$ 620F 单元的内容分别顺序搬移到 \$ 7000—\$ 700F, \$ 7100—\$ 710F, \$ 7200—\$ 720F 的连续单元中去。

这个问题,我们在循环程序设计[例 7]中作过介绍(程序 7.9),现在用调用子程序 8.4 的方法处理。

先给出源程序 8.5,并看一下运行结果,然后对程序作一说明。

程序 8.5

```
1000- A9 60      LDA # $ 60
1002- 8D 2C 10   STA $ 102C
1005- 8D 34 10   STA $ 1034
1008- A9 70      LDA # $ 70
100A- 8D 3C 10   STA $ 103C
100D- 20 25 10   JSR $ 1025
1010- EE 2C 10   INC $ 102C
1013- EE 34 10   INC $ 1034
1016- EE 3C 10   INC $ 103C
1019- 20 25 10   JSR $ 1025
101C- EE 2C 10   INC $ 102C
101F- EE 34 10   INC $ 1034
1022- EE 3C 10   INC $ 103C
1025- A0 00      LDY # $ 00
1027- A9 00      LDA # $ 00
1029- 85 3C      STA $ 3C
102B- A9 62      LDA # $ 62
102D- 85 3D      STA $ 3D
102F- A9 10      LDA # $ 10
1031- 85 3E      STA $ 3E
1033- A9 62      LDA # $ 62
1035- 85 3F      STA $ 3F
1037- A9 00      LDA # $ 00
1039- 85 42      STA $ 42
103B- A9 72      LDA # $ 72
103D- 85 43      STA $ 43
103F- 20 2C FE   JSR $ FE2C
1042- 60         RTS
```

运行前:

```
* 6000.600F
6000- 01 02 03 04 05 06 07 08
6008- 09 0A 0B 0C 0D 0E 0F 10
* 6100.610F
6100- 09 0A 0B 0C 0D 0E 0F 10
6108- 01 02 03 04 05 06 07 08
* 6200.620F
6200- 2F 2E 2D 2C 2B 2A 29 28
6208- 27 26 25 24 23 22 21 03
```

运行后:

```
* 1000G
* 7000.700F
7000- 01 02 03 04 05 06 07 08
7008- 09 0A 0B 0C 0D 0E 0F 10
* 7100.710F
7100- 09 0A 0B 0C 0D 0E 0F 10
7108- 01 02 03 04 05 06 07 08
* 7200.720F
7200- 2F 2E 2D 2C 2B 2A 29 28
7208- 27 26 25 24 23 22 21 03
```

程序 8.5 说明:

• \$ 1025—\$ 1042 就是本节[例 4]中介绍的搬家子程序(\$ 1000—\$ 101D),它们的指令完全一样,只是个别地址的内容不同,存储空间相异而已。其功能是一个独立的完成单个数据块搬家的子程序。

• \$ 1000—\$ 1024 是主程序,它完成第一个数据块首地址、末地址、目标地址的确定和搬迁任务(\$ 1000—\$ 100F),又完成第二、三两个数据块地址指针的调整和搬移任务(分别为 \$ 1010—\$ 101B, \$ 101C—\$ 1042)。

• 本程序执行过程共分三段:① \$ 1000—100F—\$ 1025—\$ 1042;② \$ 1010—\$ 101B—\$ 1025—\$ 1042;③ \$ 101C—\$ 1042。可以看出是一个主程序几次重复调用子程序的结构,最后一次省去了转子指令 JSR,而直接和子程序连成一片,从而完成了主程序和子程序的正确衔接,这是编制程序时应予注意的。

[例 6] 多个子程序的调用

在子程序的设计中,大多数由一个主程序和几个子程序组成,每一个子程序完成的功能不一样,它们之间又具有相互的独立性。以本节[例 5]为例,程序 8.5 中修改地址指针使内存单元加 1 的功能出现二处,可以改成一个独立的子程序,从而使程序 8.5 变成一个主程序调用二个子程序的结构,见程序 8.6。

程序 8.6:

```
1000- A9 60      LDA # $ 60
1002- 8D 20 10   STA $ 1020
1005- 8D 28 10   STA $ 1028
1008- A9 70      LDA # $ 70
100A- 8D 30 10   STA $ 1030
100D- 20 19 10   JSR $ 1019
1010- 20 37 10   JSR $ 1037
1013- 20 19 10   JSR $ 1019
1016- 20 37 10   JSR $ 1037
1019- A0 00      LDY # $ 00
101B- A9 00      LDA # $ 00
101D- 85 3C      STA $ 3C
101F- A9 62      LDA # $ 62
1021- 85 3D      STA $ 3D
1023- A9 10      LDA # $ 10
1025- 85 3E      STA $ 3E
1027- A9 62      LDA # $ 62
1029- 85 3F      STA $ 3F
```

```

102B- A9 00    LDA # $ 00
102D- 85 42    STA $ 42
102F- A9 72    LDA # $ 72
1031- 85 43    STA $ 43
1033- 20 2C FE JSR $FE2C
1036- 60       RTS
1037- EE 20 10 INC $ 1020
103A- EE 28 10 INC $ 1028
103D- EE 30 10 INC $ 1030
1040- 60       RTS

```

程序 8.6 说明:

• 从程序看出,共有四次调用子程序(见 \$ 100D — \$ 1018),调用的是两个子程序(\$ 1019 — \$ 1036 和 \$ 1037 — \$ 1040)。

• 程序 8.6 中,有子程序嵌套的情况。例如,当主程序工作时,首先转入子程序 1(\$ 1019 — \$ 1036),而子程序 1 工作时又转入子程序 2(\$ 1033 — \$ 1036),当子程序 2(监控搬家子程序)工作完毕后返回子程序 1,子程序 1 执行完毕后再返回主程序。在这个过程中,子程序 1 相对于子程 2 而言就处在主程序的地位,这就叫做子程序嵌套。

[例 7] 输入码转换

在键盘上按下三位十进制数,转换成 16 进制数。

如:按键 254 (10 进制数) 显示 FE(16 进制数)

按键 128 (10 进制数) 显示 80(16 进制数)

关于从键盘上按下一位 10 进制数,将其转换成 16 进制数的问题,我们曾在分支程序设计一节中作过介绍,见程序 6.13。现在的问题是,要从键盘上按下三位 10 进制数,将其转换成 16 进制数,除了可以调用前面的程序 6.13(它的任务是把 10 进制数转换成 2 进制数)外,还应多次调用,以完成百位数、十位数、个位数均转换成 2 进制数的问题,同时,还要有另外一个子程序,它完成乘 10 加数的功能,并且不止一次调用。

程序 8.7

```

1000- 20 15 10 JSR $ 1015;取百位数转成 2 进制在 A
1003- 85 06    STA $ 06 ;存 06 单元作为中间结果
1005- 20 15 10 JSR $ 1015取拾位数转成 2 进制在 A
1008- 20 23 10 JSR $ 1023; 转乘 10 加数子程序
100B- 20 15 10 JSR $ 1015; 取个位数转 2 进制在 A
100E- 20 23 10 JSR $ 1023; 转乘 10 加数子程序
1011- 20 DA FD JSR $FDFA;显示结束
1014- 60       RTS
1015- 20 1B FD JSR $FD1B
1018- C9 B0    CMP # $ B0
101A- 90 F9    BCC # 1015
101C- C9 BA    CMP # $ BA
101E- B0 F5    BCS $ 1015
1020- 29 0F    AND # $ 0F
1022- 60       RTS
1023- 85 07    STA $ 07 ;整存 10 位,个位数
1025- A5 06    LDA $ 06 ;调中间结果
1027- 0A      ASL;得 2A
1028- 0A      ASL;得 4A

```

```

1029- 65 06    ADC $ 06 ;得 5A
102B- 0A      ASL ;得 10A
102C- 65 07    ADC $ 07 ;出 10 加数
102E- 85 06    STA $ 06 ;存结果到工作单元
1030- 60       RTS

```

运行: * 1000G

* 254

* FE

* 1000G

* 128

* 80

• \$ 1015 — \$ 1022 是一个子程序,它完成 10 进制数转换成 2 进制数的工作,由于本程序要处理百位数、十位数、个位数转换成 2 进制的工作,所以三次调用了这个子程序。

• \$ 1023 — \$ 1030 是另一个子程序,它完成乘 10 加数的工作,共要调用两次。

• 本程序的两个子程序构成两重子程序结构。

[例 8] 比较两个字符串是否相同

比较两个 ASCII 字符串,看其是否相同。字符串长度在存储单元 6000 中,两个字符串的起始地址各为 6100 和 6200,如果这两个字符串相同,将 00 放入 6001 单元,反之不同放 FF 标志。示范题:

a) (6000)=03 字符串长度

(6100)=43 "C"

(6101)=41 "A"

(6102)=54 "T"

(6200)=44 "D"

(6201)=41 "A"

(6202)=54 "T"

结果: (6001)=FF

b) (6000)=03

(6100)=43 "C"

(6101)=41 "A"

(6102)=54 "T"

(6200)=43 "C"

(6201)=41 "A"

(6202)=54 "T"

结果: (6001)=00

程序 8.8

```

1000- AC 00 60 LDY $ 6000
1003- 20 0A 10 JSR $ 100A
1006- 8D 01 60 STA $ 6001
1009- 60       RTS
100A- A2 FF    LDX # $ FF
100C- 88       DEY
100D- B1 06    LDA ($ 06),Y
100F- D1 08    CMP ($ 08),Y
1011- D0 05    BNE $ 1018
1013- 98       TYA
1014- D0 F6    BNE $ 100C
1016- A2 00    LDX # $ 00
1018- 8A      TXA

```

• 主程序见 \$1000—\$1009, 字符串长度放 Y 寄存器, 转字符串比较子程序, 送结果, 结束。

• 子程序见 \$100A—\$1019, 首先置字符串不同标志 FF→X, Y-1, 逐字比较两个字符串, 若不相同返回; 若相同, 看每个字符是否都比较大完, 是则取 00 标志→X, 返回, 不是再循环上去。

• 在运行本程序前, 应将两字符串起始地存入

(接第 48 页)

统的提示符下, 只要打入文件名 LOWFORM, 然后根据屏幕提示信息打入“回车”键即可。

(3) 运行随机提供的诊断程序

不管是 IBM-PC 系列机或是长城系列机及其兼容机, 都随机提供给用户一个高级诊断程序。利用这个诊断程序也可进行硬盘的初始化操作, 只是不同机型诊断程序的名称不同而已。例如新一代长城系列机提供给用户一个诊断程序盘, 进行诊断程序时, 显示如下: The GREAT Wersonal computer DIAGNOSTICS

Version 6.04(c) copyright ACI1958

SELECT AN OPTION

0-RUN DIAGNOSTIC ROUTINES

1-FORMAT DISKETTE

2-COPY DISKETTE

3-PREPARE FIXED DISK FOR RELOCATION

4-SETUP YOUR SYSTEM

9-EXIT TO SYSTEM DISKETTE

ENTER THE ACTION DESIRED

?

用户选择 0 项后, 诊断程序就会清点系统的配置情况, 并把这一配置情况显示出来。一种可能的显示情况如下:

THE INSTALLED DEVICES ARE

S SYSTEM UNIT 1

S 512KB MEMORY 2

S KEYBOARD 3

S 2 DISKETTE DRIVE(S) AND ADAPTER 6

S SERIAL/PARALLEL ADAPTER

-PARALLEL PORT 9

S SERIAL/PARALLEL ADAPTER

-SERIAL PORT 11

S ALTERNATE SERIAL/PARALLEL ADAPTER

-SERIAL PORT 12

S 1 FIXED DISK DRIVE(S)

AND ADAPTER 17

S TEST THE MATRIX PRINTER

INSTALLED 36

IS THE LIST CORRECT (Y/N)

?

根据屏幕上的提示信息, 回答 Y 并回车, 屏幕又显示如下信息:

SYSTEM CHECKOUT

0- RUN TESTS ONE TIME

06, 07 单元和 08, 09 单元, 而在 6100 和 6200 开始存放字符串的 ASCII 码, 然后 1000G, 即可看到结果。

• 本程序比较两个字符串时, 采用由后向前比。

从以上所举实例, 可以看出在程序设计中, 有一些程序需要反复使用, 为了方便操作, 简化设计, 我们常采用子程序的方法, 使那些在功能上一样, 处理问题相同, 并具有一定独立性的重复操作部份编在一起, 只要注意正确调用和返回, 主程序与子程序衔接完整即可。

1- RUN TESTS MULTIPLE TIMES

2- LOG UTILITIES

9- END SYSTEM CHECKOUT

10-CHECK PASS COUNTER

SELECT THE ACTION DESIRED

?

选择 0 项后, 诊断程序又显示系统配置情况, 因为是对硬盘进行测试, 所以应该选择 17 并按回车键, 这时进入硬盘检测菜单:

REFER TO-TEST 1700

PRESS“ENTER”TO CONTINUE

?

按回车键后屏幕继续显示:

FIXED DISK DIAGNOSTIC MENU

1- WRITE, READ, COMPARE (ON TSET CYLINDER)

2- SEEK TEST

3- HEAD SELECT

4- ERROR DETECTION AND CORRECTION

5- RUN ALL TESTS

6- READ VERIFY

7- FORMAT MENU

9- RETURN TO CONTROL PROGRAM

FOR OPTION 9

TYPE“9”AND PRESS“ENTER”

FOR OTHER OPTION TYPE THE OPTION

DRIVEID(1,C), AND PRESS“ENTER”

?

此时回答 7, C 并回车就进入硬盘格式化菜单:

FORMAT SELECTION MENU

1- CONDITIONAL FORMAT

2- UNCONDITIONAL FORMAT

3- SURFACE ANALYSIS

4- CHANGE INTERLEAVE

9- RETURN TO FIXED DISK MENU

FOR OPTION 9

TYPE“9”AND PRESS“ENTER”

FOR OTHER OPTIONS

TYPE THE OPTION NUMBER

DRIVEID(1,C), AND PRESS“ENTER”

?

如果选择 1, C 则进行条件格式化硬盘; 如果选择 2, C 则进行无条件格式化硬盘。不论选择上述哪种情况, 屏幕均出现警告信息如下:

ALL DEFECTS WILL BE SHOWN ON THE DISPLAY
THEY CAN ALSO BE PRINTED ON LPT1 (转第 33 页)

一九九二年计算机初级软件人员竞赛试题

主办单位: 中国软件行业协会考试指导中心

北京市科学技术协会

《电子与电脑》编辑部

《计算机世界》编辑部

说明: 本试题以 IBM PC 及其兼容机为基本机型。

一、从供选择的答案中选出适合下列表格中的 a~h 的正确答案, 把相应的编号写在答卷纸的对应栏内。(每空 1 分, 共 8 分)

二进制数	八进制数	十进制数	十六进制数
01000010	c	66	g
a	d	127	7F
01010101	125	e	h
b	143	f	63

a, b 供选择的答案:

- (1) 01011110 (2) 01111111
(3) 01111000 (4) 01100011

c, d 供选择的答案:

- (5) 135 (6) 120
(7) 177 (8) 102

e, f, g, h 供选择的答案:

- (9) 92 (10) 85 (11) 52
(12) 99 (13) 42 (14) 68
(15) 73 (16) 55

二、从供选择的答案中选出正确答案, 把相应的编号写在答卷纸的对应栏内。(每空 1 分, 共 22 分)

1. 已知一逻辑表达式为 $F = \overline{A}B + \overline{B}C + \overline{B}C + \overline{A}B$, 化简后可得到逻辑表达式_____。

供选择的答案:

- (1) $\overline{A}B + \overline{B}C + \overline{B}C$ (2) $\overline{A}B + \overline{B}C + \overline{B}C$
(3) $\overline{A}B + \overline{A}B + \overline{B}C$ (4) $\overline{A}B + \overline{B}C + \overline{A}C$

2. DMA 控制器是控制外部输入输出设备与直接传送信息(数据)的逻辑电路。

供选择的答案:

- (1) 运算器 (2) 主存储器
(3) 程序计数器 (4) 指令译码器

3. 计算机中的存储器分为主存储器和辅助存储器两大类。

主存储器用来存放当前运行所需要的程序和数据。它的存取速度_____, 存储容量_____, 价格_____, 它所存储的信息永久性地保留。

辅助存储器用来存放当前不需要立即使用的程序和数据。一旦需要, 再和主存储器成批地交换数据。它作为主存储器的后备和补充, 是主机的外部设备, 与主存储器相比较, 它的存取速度_____, 存储容量_____, 价格_____, 它所存储的信息永久性地脱机保留。

供选择的答案:

- (1) 慢 (2) 大 (3) 快 (4) 能够
(5) 不能 (6) 小 (7) 低 (8) 高

4. 一个字节由 8 位二进制组成, 它除了表示 26 个英语字母、0~9 个数字以外, 还能表示_____个字符。

两个字节所能表示的字符数量是一个字节能表示的字符数量的_____倍。

供选择的答案:

- (1) 2 (2) 8 (3) 128
(4) 220 (5) 64 (6) 256

5. 显示设备的种类繁多。按它的功能分类, 可分为普通显示器和显示终端两大类。显示器和显示终端是两个不同的概念。

显示器的功能简单, 只能用于接收_____信号, 它的控制逻辑和存储逻辑都在_____中, 目前使用的个人计算机系统就是这种结构。

显示终端是由显示器和键盘组成的一套独立完整的输入/输出设备。它具有显示控制与存储、键盘管理、通信控制等功能, 还可以完成简单的编辑操作。它可以通过_____接到远离主机的地方使用。

供选择的答案:

- (1) 音频 (2) 主机接口板
(3) 标准通信接口 (4) 视频
(5) 电缆线 (6) 控制台

6. 在显示技术中, 图形和图象都是由称为像素的光点组成的。显示设备所能表示的像素个数称为_____。像素越密, 图象越_____, 显示设备所显示像素点的亮暗差别、深浅变化称为_____, 它在彩色显示器中则表现为_____不同。

供选择的答案:

- (1) 清晰 (2) 颜色 (3) 灰度级
(4) 分辨率 (5) 模糊 (6) 精确度

7. 激光印字机是激光技术和_____相结合的产物。它的输出速度快, 印字质量高, 是一种_____硬拷贝输出设备。

供选择的答案:

- (1) 击打式 (2) 电子照相技术
(3) 机械印字技术 (4) 非击打式

8. 微机安装时, 应注意电源的供电电压是否是和微机所要求的一致, 同时必须按照_____的规则连接电源线。

供选择的答案:

- (1) 左零(零线)右火(火线) (2) 右零左火

三、从供选择的答案中选出正确答案, 把相应的编号写在答卷纸的对应栏内。(每空 1 分, 共 10 分)

1. 用_____编制的源程序输入计算机后, 可直接运行得到结果。

供选择的答案:

- A. 汇编语言 B. 机器语言
C. BASIC语言 D. C语言

2. _____记录的逻辑顺序与物理位置顺序不一致。

供选择的答案:

- A. 磁带文件 B. 打印机文件
C. 索引文件 C. 顺序文件

- 3 在操作系统处理机管理中,一般把_____当做资源分配和调度的基本单位。

供选择的答案:

- A. 作业 B. 进程 C. 程序 D. 用户

- 4 在操作系统中,文件按其性质和用途可分为几类,其中库文件是指_____。

供选择的答案:

- A. 由用户建立起来的数据库文件
B. 由标准子程序及常用的子程序组成的文件
C. 允许文件主和授权的用户读写和使用的文件
D. 有关操作系统及其系统程序信息组成的文件

- 5 定义一个数组A[a, b],设A[0, 0]的地址是L,这个数组按列存储,其元素的地址从L开始逐渐递增,那么A[i, j]的地址是_____。
($0 \leq i \leq a, 0 \leq j \leq b$)。

供选择的答案:

- A. $L+B \cdot i+(j+1)$
B. $L+(b+1) \cdot i+(j+1)$
C. $L+(a+1) \cdot j+(i+1)$
D. $L+(a+1) \cdot j+1$

- 6 _____不是计算机汉字处理过程中采用的数据结构。

供选择的答案:

- A. 字形表示 B. 内码
C. 外部输入码 D. 打印输出码

- 7 文件的组织包括文件的逻辑结构和物理结构两方面,文件的逻辑结构是指_____;物理结构是指_____。

供选择的答案:

- A. 一个文件内部各部分之间的关系
B. 文件由记录组成或为有序字符的集合
C. 文件在外存上的存放方式
D. 采用连续结构或链接结构的文件

- 8 使用结构化程序设计的方法,使得软件的_____、_____加强。

供选择的答案:

- A. 可读性 B. 可靠性
C. 可移植性 D. 可操作性

- 四. 判断下面的叙述是否正确,正确叙述画√,错误的叙述画×,填在答卷纸的对应栏内。(每空1分,共10分)

- DOS的外部命令以文件形式存在于磁盘上,在使用该命令时才调入内存。()
- 堆栈是一种特殊的线性表,它采用先进先出的原则组织数据。()
- 设计数据库的目的是使用户方便有效地存储和检索数据,并获得信息。()
- 文件系统是在数据库基础上发展起来的。()
- UNIX操作系统是一种通用、交互式的实时操作系统。()
- 计算机病毒是由硬件故障造成的。()
- dBASE是一种关系数据库管理系统。()用表格数据来表示实体之间联系的模型称为关系模型。()
- 反汇编是将机器语言翻译成高级语言。()汇编程序是把用汇编语言编写的源程序翻译成机器语言程序。()

言程序。()

- 五. 从供选择的答案中选出正确答案,把相应的字母写在答卷纸的对应栏内。(每空3分,共15分)

1. BASIC表达式 $\text{INT}(-50+50 \cdot \text{RND}(0))+\text{FIX}(-50+50 \cdot \text{RND}(1))$ 产生随机整数的区间是_____。

供选择的答案:

- A. $(-99, -1)$ B. $[-99, -1]$
C. $[-100, -2]$ D. $(-100, -2)$

2. 已知BASIC语句 $A\$ = "A123\#-3456"$, $L = \text{LEN}(\text{STR}\$(-321)) + \text{VAL}(\text{MTD}\$(A\$, 6, 2))$, 则 $L =$ _____。

供选择的答案:

- A. -1 B. 0 C. 1 D. 2

3. BASIC表达式 $\text{INT}(\text{SIN}(290 \cdot 3.14159/180)) + \text{SGN}(\text{COS}(190 \cdot 3.14159/180))$ 的值是_____。

供选择的答案:

- A. -1 B. 0 C. 1 D. -2

4. 这个流程图对应的程序功能是_____。

供选择的答案:

- A. 求X对Y的模
B. 求X与Y的差
C. 求X与2Y的差
D. 死循环

5. 下面关于循环的说法哪种正确_____。

供选择的答案:

- A. 计数循环, 当型循环, 直到型循环是不可转化的。
B. 计数循环和当型循环可以相互转化, 而计数和直到型循环是不可以相互转化的。
C. 当型循环和直到型循环可以互相转化而计数和后两种循环都不能相互转化。
D. 三种循环之间是可以相互转换的。

- 六. 阅读下列BASIC程序, 将供选择答案中正确答案相应的字母填在答卷纸的对应栏内。(每空3分, 共15分)

1 程序

```
10 Y=1
20 DEF FNA(X)=X^2+2*X+1
30 DEF FNB(X)=FNA(X)-1
40 Y=Y+FNB(FNA(1))
50 PRINT "Y="; Y
60 END
```

运行结果()

供选择的答案:

- A. Y=4 B. Y=26 C. Y=25 D. Y=24

2 程序

```
10 A$="CBACBBAACBAC"
20 B$="BA"
30 V=INSTR(4,A$,B$)
40 Y$=MID$(A$,V,1)
50 FOR I=2 TO 1 STEP -1
60 Y$=LEFT$(A$,I)+Y$+RIGHT$(A$,I)
70 NEXT I
```

80 PRINT Y\$

90 END

运行结果 ()

供选择的答案:

A. CCBACC B. CBBAACC

C. CBCBCAC D. CBABC

3 程序

10 N=0

20 FOR I=1 TO 10

30 I=I+4

40 N=N+I²

50 NEXT I

60 I=I+4

70 N=N+I²

80 PRINT N

90 END

运行结果 ()

供选择的答案:

A. 321 B. 350 C. 564 D. 599

4 程序

10 INPUT X

20 P=0

30 GOSUB 60

40 PRINT P

50 END

60 Y=X²

70 IF Y>1 THEN 100

80 P=P+X²

90 RETURN

100 P=P+X

110 RETURN

这个程序完成的功能是计算下列哪个函数的值 ()

$$A. P = \begin{cases} X^2 & |X| > 1 \\ X & |X| < 1 \end{cases}$$

$$B. P = \begin{cases} X^2 & |X| < 1 \\ X & |X| > 1 \end{cases}$$

$$C. P = \begin{cases} X^2 + X & |X| < 1 \\ X & |X| > 1 \end{cases}$$

$$D. P = \begin{cases} X^2 - X & |X| < 1 \\ X & |X| > 1 \end{cases}$$

5 程序

10 FOR M=3 TO 20

20 FOR I=2 TO M-1

30 IF INT (M/I) = M/I THEN 60

40 NEXT I

50 PRINT M

60 NEXT M

70 END

运行结果 ()

供选择的答案:

A. 2 3 5 7 11 13 17 19

B. 2 3 5 7 9 11 13 15 17 19

C. 3 5 7 9 11 13 15 17 19

D. 3 5 7 11 13 17 19

七. 补充完成下列程序, 使其能完成求任意两个数的最

大公约数, 将答案填在答卷纸对应栏内。(每空 3 分, 共 15 分)

10 INPUT X, Y

20 IF Y>=X THEN 60

30 A=X

40 B=Y

50 GOTO 80

60 A=Y

70 B=X

80 R=A-B

90 IF (1) THEN 120

100 A= (2)

110 GOTO 80

120 IF (3) THEN 160

130 A= (4)

140 B= (5)

150 GOTO 80

160 PRINT B

170 END

这个程序采用的算法是欧几里德辗转相除法。其步骤如下:

(1) 令二个数值放在 A、B 中, A 中为大数

(2) 求用 B 除 A 的余数 R1

(3) 然后求用 R1 除 B 的余数 R2

(4) 再求用 R2 除 R1 的余数 R3

(5) 依次类推, 用新求得的余数除前一个余数, 得到下一个余数

(6) 经过若干次除法以后, 余数 Rn 便会为 0, 这时 Rn-1 就是 A、B 的最大公约数。

八. 从供选择的答案中选出正确答案, 把相应的编号写在答卷纸的对应横线上。(每空 2 分, 共 14 分)

1. 在 dBASE III 中用户文件有 _____ 类, 其中索引文件的默认扩展名是 _____。

供选择的答案:

A. 5 B. 7 C. 8 D. 9

E. prg F. dbf G. ndx H. txt

2. CONFIG.SYS 文件的作用是 _____。

供选择的答案:

A. 通过设置 CONFIG.SYS 改变 DOS 缓冲区数和允许

同时打开的文件数的默认值

B. 设置 DOS 缓冲区的大小

C. 分配内存空间

D. 设置路径

3. 汉字 dBASE III 有 _____ 个工作区, 在每个工作区中, 用户都可单独打 _____ 个库文件。刚进入 dBASE III 时 _____ 号工作区被默认为当前工作区。

供选择的答案:

A. 1 B. 2 C. 5 D. 10

4. 下面 _____ 不是 dBASE III 的字段类型:

供选择的答案:

A. 数字型 B. 日期型 C. 逻辑型

D. 字符型 E. 记忆型 F. 函数型

九. 在汉字 dBASE III 系统下, 用户已建立名为 GZ.DBF 的文件, 用 LIST 命令看显示的结果为: (每空 2 分, 共 26 分)

RECORD#	编号	姓名	工龄	职称	工资
1	01	张一	10	技术员	100
2	04	王二	15	工程师	120
3	03	李三	20	高工	200
4	06	赵四	5	助工	110
5	02	马五	7	工人	80
6	05	严六	25	高工	250

下面命令将对哪些记录进行编辑或替换 (写出记录号即可) ?

1 .USE GZ

.EDIT

记录号()

供选择的答案:

A. 1 B. 2 C. 3 D. 5

2 .USE GZ

.GO 3

.APPEND BLANK

记录号()

供选择的答案:

A. 1 B. 3 C. 5 D. 7

3 .USE GZ

.REPLACE 工资 WITH 工资+20 FOR 工龄 > 15

记录号(),()

供选择的答案:

A. 1 B. 2 C. 3 D. 4 E. 5 F. 6

4 .USE GZ

.LOCATE FOR 职称="高工".AND.工龄>15

.CONTINUE

记录号(),()

供选择的答案:

A. 1 B. 2 C. 3 D. 4 D. 5 F. 6

5 .USE GZ

.GOTO 3

.COUNT NEXT 4 FOR 工龄 < 20 .OR. 姓名="严六"

执行结果 ()

供选择的答案:

A. 1 B. 2 C. 3 D. 4 E. 5 F. 6

6 .USE GZ

.COPY TO GZ1 NEXT 5 FOR 职称="工程师"

.OR. 工龄 > 10

.USE GZ1

.SUM 工资

执行结果()

供选择的答案:

A. 320 B. 570 C. 370 D. 420

7 .USE GZ

.INDEX ON 编号 TO A:GZ.NDX

执行后此文件的逻辑顺序为()

(注:按记录号排列)

供选择的答案:

A. 2,3,4,5,1,6 B. 2,1,6,4,5,3

C. 1,5,3,2,6,4 D. 1,2,3,4,5,6

8 SET TALK OFF

USE GZ

STORE 0 TO A,B,C,D

DO WHILE .NOT. EOF()

IF 工资 > 150

C = C + 工资

A = A + 1

ELSE

D = D + 工资

B = B + 1

ENDIF

SKIP

ENDDO

程序执行后变量

A=(), B=(), C=(), D=()

供选择的答案:

A. 1 B. 2 C. 3 D. 4

E. 400 F. 450 G. 410 H. 860

十. 请将下面的计算机英语术语译成中文.

(1) screen (2) diagnostic routine

(3) interrupt (4) debug

(5) on-line (6) workstation

(7) compile (8) temporary area

(9) virtual address (10) default

十一. 将下面的计算机术语或屏幕信息译成中文:

(1) Program too big to fit in memory

(2) file allocation table bad

(3) Internal stack overflow, system halted

(4) Content of destination lost before copy

说 明

本竞赛试题是初级程序员级软件水平考试辅导综合自测题, 也是我考试指导中心函授班期末复习考试题, 希望广大读者及学员踊跃参赛, 以检验自己的学习成绩及计算机知识水平.

中国软件行业协会考试指导中心

91.7.

培 训 消 息

我中心常年举办文字处理办公自动化初级班, 进行面授教学. 培训内容: DOS, CCDOS, WS, 五笔字型录入, 计算机排版技术, dBASE III 数据库. 每月一期, 月初开学, 每期三周, 讲授 80 学时, 上机 40 学时. 培训费: 走读生 300 元 (包括培训费, 上机费), 资料费实收. 住宿生: 另加住宿费 180 元. 结业经考试合格发给钢印证书.

联系地址: 北京学院路 29 号 (中国地质大学东门内), 中国软件行业协会考试指导中心, 邮政编码: 100083

电话: 2012233-322

一九九二年计算机初级程序员有奖竞赛试题答卷

总分: _____ 学号: _____ (如是函授学员请注明学号)

姓名: _____ 性别: _____ 文化程度: _____ 邮政编码: _____

单位: _____ 通信地址: _____

注意事项:

1. 答案必须在九月二十三日寄出, 以当地邮戳为准(通信地址: 北京学院路29号, 中国软件行业协会考试指导中心. 邮政编码: 100083).
2. 填写答案时, 必须将试题号与对应的答案栏对准, 绝勿弄错, 若某试题有多个答案时, 请顺序给出(共150分).
3. 解答时字迹务必清楚, 字迹不清楚时不能得分.
4. 是否愿意参加复赛: 是 _____, 否 _____.

题 号	小 题 号 与 答 案																得 分
一	a			b			c			d							8
	e			f			g			h							
二	1			2			3			4							22
	5			6			7			8							
三	1		2		3		4		5		6		7		8		10
四	1		2		3		4		5		6		7		8		10
五	1			2			3			4			5				15
六	1			2			3			4			5				15
七	(1)			(2)			(3)			(4)			(5)				15
八	1			2			3			4							14
九	1			2			3			4							26
	5			6			7			8							
十	(1)			(2)			(3)			(4)			(5)				10
	(6)			(7)			(8)			(9)			(10)				
十一	(1)																10
	(2)																
	(3)																
	(4)																
总 分																	150



BJS-51 的 监 控 程 序

北京航空航天大学(100083) 盛焕鸣

教育实验系统 BJS-51 的监控程序约 16K, 固化在一片 27128 中, 实际上它由两个独立的程序块组成。前 8K 为以键盘及 LED 为输入输出手段的管理程序 (BJS-51-LED), 后 8K 为以 PC 机(其他类型计算机或终端)为输入输出手段进行用户程序编写、运行、调试的管理程序 (BJS-51-PC)。在运行时通过接在 27128 地址线 A13 的开关 K1(见图 1)来选择前 8K 或后 8K 投入运行, 因此用户可以根据自己的使用环境通过 K1 来选择。

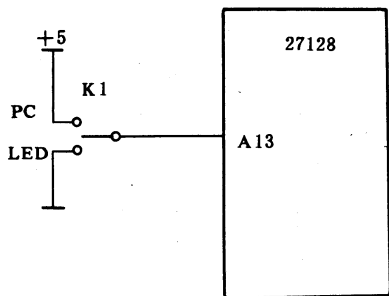


图 1

在单片机系统设计、调试过程中, 大多数人都经历过一个个敲机器码的阶段(而且很多同志由于习惯现在还在敲机器码)。但随着 PC 机的价格下降, PC 机已成为单片机应用系统开发过程中一种必不可少的工具, 从硬件设计阶段的电原理图设计、印制板设计到软件设计过程中的源文件编写、汇编、系统调试以至说明书的编写, PC 机都是最好的助手。今后从事单片机系统设计的工程技术人员为了适应市场竞争机制提高效率, 必须掌握 PC 机的使用, 并把它使用到单片机系统开发的各个环节中。因此, 设计教育实验系统时, 是以有 PC 机支持为最基本运行环境的, 这不仅是为了降低成本满足学校的使用环境, 更重要的是为了使学生会较先进的单片机开发调试技术。

作为补充, BJS-51 也提供在键盘、LED 显示器支持下进行用户程序调试的功能。由于使用键盘、LED 显示器进行程序调试具有携带方便设备简单等特点, 因此特别适合于在现场进行单片机系统故障检查、程序修改、EPROM 写入等工作。考虑到目前还有很多单位不具备 PC 机, 所以 BJS-51 上所配置的键盘-LED 监控程序具有一般仿真器的功能。下面分别对这两个监控程序进行介绍。

一、BJS-51-PC

BJS-51-PC 具有很强的调试功能, 可以代替仿真器进行 51 系列单片机用户程序的调试。这些功能包

括:

1. 状态显示功能

可以显示单片机内部 128(或 256)字节 RAM 的内容, 以及所有特殊功能寄存器及二个 64K 的外部存储器空间的内容。并可以 ASCII 字符方式显示外部 RAM 的内容。

2. 状态修改功能:

可以对内部 RAM、特殊功能寄存器、外部 RAM 空间、寄存器组按字节进行修改。此外还可以对单片机系统的位控区进行逐位修改。

3. 运行控制功能:

可以控制单片机进行单步运行、多步跟踪运行、断点运行、连续全速运行。除全速运行外, 其他各种运行方式都能在运行结束后, 向用户提供运行后单片机系统的状态信息。

4. 数据传送功能:

可以在外部 RAM 空间中 SRAM(静态随机存储器)之间以及 SRAM 与 EEPROM(电可改写的程序存储器)之间进行数据传送。可以在外部 RAM 空间与 EPROM 空间(由 EPROM 固化板开辟的存储空间)之间进行数据传送。此外还能在单片机外部 RAM 空间与磁盘文件之间进行数据传送。

5. 其他功能:

1)反汇编功能。可以对单片机外部 RAM 空间中的机器码进行反汇编, 以列表方式进行显示、存盘。

2)字符串搜索功能。可以在指定外部 RAM 空间中对指定字符串(字符串最多可由十个字符组成)进行搜索, 并在屏幕上显示出现指定字符串的存储地址(出现字符串中首字符的地址)。

3)数据块比较功能。可以对指定的二个数据块进行比较, 若二个数据块中内容不同, 则显示所有不相同单元的地址及内容。

4)EPROM 编程功能。包括编程、读出、检查、检验等功能(必须与 EPROM 编程板配合)。

BJS-51-PC 在系统复位后首先要与终端机进行异步通信的波特率同步。BJS-51-PC 采用测试终端机异步通信信号的位宽度, 通过计算确定定时器 T1 的时间常数并装入到 T1H, 来确定 BJS-51 的通信波特率。因此, 系统复位后 PC 机(或终端机)必须键入一个回车(送一个 ASCII 码 0DH)来提供终端机的波特率时间常数。串口初始化后在 PC 机的屏幕上将显示软件的版本号及提示符“*”。这时用户可以根据需要键入各种监控命令。

BJS-51-PC 的监控命令格式为:

ZZ n1,n2,n3.....〈回车〉

其中 ZZ 为命令字,用以确定哪一种操作。后面的空格是分隔符,表示命令字结束。

n1,n2,n3.....为命令所需要的参数,例如数据块移动的源地址等,不同的命令需要的参数不同。

〈回车〉为整个命令的结束符。系统在接到回车后就开始执行键入的命令。

BJS-51-PC 的各类命令见表 1。

表、BJS-51-PC 监控命令表

命令类型	命令格式	执行功能
状态显示	DX n1,n2〈回车〉	外部数据存储器内容显示
	DC n1,n2〈回车〉	外部程序存储器内容显示
	D〈回车〉	片内数据存储器内容显示
	R〈回车〉	显示局部参数 PC,ACC, SP,B,DPTR,RO,R1,PSW
状态修改	SX n1〈回车〉	外部数据存储器内容修改
	SD n1〈回车〉	片内数据存储器内容修改
	BIT n1〈回车〉	位空间内容修改
	XX〈回车〉	特殊功能寄存器内容修改 XX 为特殊功能寄存器名
	REG n1〈回车〉	当前寄存器组内容修改
数据传送	M n1,n2,n3〈回车〉	在外部 RAM 空间进行数据块传送
	ME n1,n2,n3〈回车〉	把外部 RAM 的内容送到 EEPROM 中去
EPROM 操作	MPJ n1,n2〈回车〉	检查 EPROM 是否全部为 FF
	MPR n1,n2,n3〈回车〉	把 EPROM 的内容送到 RAM 空间
	MPX n1,n2,n3〈回车〉	把 EPROM 的内容与 RAM 的内容比较
	MPK n1,n2,n3〈回车〉	把 RAM 空间的内容固化到 EPROM 中去
运行控制	T n1〈回车〉	从 n1 开始执行一条指令,并显示当前状态
	T n1,n2〈回车〉	从 n1 开始多步执行至 n2
	G n1,n2〈回车〉	从 n1 开始运行至断点 n2

	En1〈回车〉	从 n1 开始连续运行用户程序
其他命令	DI n1,n2,n3〈回车〉	反汇编 n1 到 n2,定位地址为 n3
	DA n1,n2〈回车〉	ASCII 字符显示
	C-C n1,n2,n3〈回车〉	数据块比较,显示不相等单元
	SX n1,n2〈回车〉m1,m2.....〈回车〉	字符串搜索

使用 BJS-51-PC 虽然占用一些单片机资源,但绝大多数情况并不影响用户程序的运行,因此 BJS-51-PC 也常被应用在用户自己的产品中,即在用户设计自己的产品时,把 BJS-51-PC 运行所需要的资源也考虑进去,然后在自己的产品上插上 BJS-51-PC,使之成为具有自开发功能的用户系统。

BJS-51-PC 使用时,需要使用单片机部份资源。因此,在编写用户程序时要把这些资源留给监控程序,以免死机。

BJS-51-PC 需要使用的资源如下:

1. 单片机内第三寄存器区(内部 RAM 18H-1FH)为监控工作区。
2. 位控单元 00H-0AH 为监控标志区。
3. 30H-41H 为监控程序用堆栈。
4. 串行口(RXD, TXD)用于与微机通信,传送命令及数据(在连续运行时用户也可用串行口)。
5. 定时器 T1,为串行通信的波特率发生器,因此有关的状态(TCON, TMOD, SCON, SMOD 中相应位)都不能修改。
6. 外部 ROM 空间中 0000H-1FFFFH。用于放置监控程序。
7. 外部中断 0,用于单步运行控制。
8. 6000H-605FH 为监控程序的外部数据缓冲区,用以存放微机送来的命令、暂存数据以及用户的中断入口地址。

51 系列单片机的中断入口地址(0003H, 000BH, 0013H, 001BH, 0023H, 002BH)均在监控程序地址空间中,用户要使用单片机的中断功能时就无法通过这些单元写入转移指令,转移到用户中断服务中去。为此,在监控程序中上述地址单元已分别写上长转移指令,把中断入口地址转移到用户能写入的 RAM 空间中。经监控程序处理后,在使用监控程序时,单片机的中断入口地址变为:

- 603BH 定时器中断入口地址
- 6043H 外部中断 1 中断入口地址
- 604BH 定时器 1 中断入口地址。
- 6053H 串行口中断入口地址。
- 605BH 定时器 2(8052 中)中断入口地址。

为此,用户在设计应用系统的硬件时,除考虑用户

本身的要求外,还需增加一些资源,它们是:

1. 在 0000H—1FFFH ROM 空间处配置一个 28 脚 IC 座,用于插入监控程序。

2. 在 6000H—7FFFH 空间处配置一片随机存储器 (6264),其读出应由 PSEN 及 RD 同时控制(即 PSEN 及 RD 通过与门后去控制)。

3. 把 8031 的串行口扩展为 RS232C 标准通信口。以便与微机的 RS232 口连通。

把 8031 的串行口扩展为 RS232C 标准口,可使用现成的集成芯片(MC1488,MC1489),但必需有+12V 和-12V 电源。因此更常用的方法是用二个晶体管组成的简易 RS232C 扩展电路(见图 2)。该电路借用微机上 RS232C 中的一-12V 信号来建立负电源,因此该电路只需与 8031 共用一个+5V 单电源即可工作。

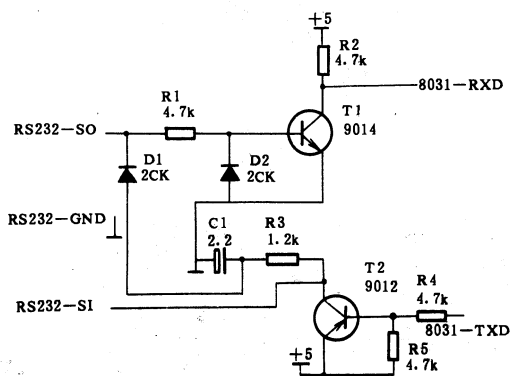


图 2 简易 RS232C 扩展电路

图中:S0 为微机串口发送端

SI 为微机串口接收端

GND 为微机串口地

在微机的 RS232 插座上分别是第 2、3、7 脚。

4. 若要使用监控程序中的打印驱动程序,打印反

汇编清单或其他调试信息,则还要配置一个打印机接口。打印机信号与 8031 的引脚连接如下:

打印机	8031
STB	P3.4
D0—D7	P1.0—P1.7
BUSY	P3.3
地	GND

5. 制作一块监控程序。读者可借助于仿真器把 BJS—51—PC 固化到一片 2764 中。只要固化无误,插入到用户系统后即可使用。需要 BJS—51—PC 的机器码清单可与作者联系。

二、BJS—51—LED

BJS—51—LED 是以 BJ—51 仿真器的监控程序为基础,根据 BJS—51 的特点修改而成。由于它以仿真器的监控程序为基础,因此它具有仿真器类似的功能。BJS—51—LED 的键功能见表 2。

表 2 BJS—51—LED 键功能表

键 名	功 能
WR1	写加 1
ME	向 EEPROM 传送数据
MOV	RAM 间传送数据
F1	设置数据块首地址
F2	设置数据块末地址
RDI	读加 1
RDS	读减 1
EPR	EPROM 操作
FI	数据块填充
INS	插入一条指令
BP	断点运行
FU	多功能键(反汇编,数据块搜索等)
C—C	数据块比较
EXE	连续运行用户程序
MON	返回监控

(接第 45 页)

10. 发光二极管电子广告牌 12,000.00 元/m²
地 址: 武汉武昌珞瑜路 100—1 号
开户银行: 武汉市洪山区科技信用社
帐 号: 967—517

邮 编: 430070 电 话: (027) 712761

传 真: (027) 701803 联系人: 姜艳艳

注:《CYSCB—2 MCS—51, 8098 单片单板机硬件、软件设计原理》将在本刊第 10 期登出。

(接第 25 页)

IS A HARD COPY NEEDED?

ENTER (Y/N)

? Y<回车>

***** WARNING *****

ALL DATA ON THE FIXED DISK WILL BE DESTROYED

DO YOU WANT TO CONTINUE?

ENTER (Y/N)

? Y<回车>

ALL DATA ON DRIVE C WILL BE DESTROYED !

THIS YOUR LAST CHANGE TO CANCEL !

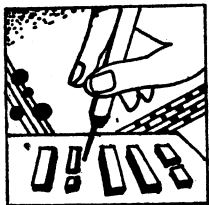
DO YOU WANT TO CONTINUE?

ENTER (Y/N)

?

在这时键入 Y,即进行硬盘的初始化。

该诊断程序既可用在长城系列机 CH、DH、286B、286BH、286EX、386 等机型,也可以用于兼容机。



学装微电脑

水平多关节型机器人

易齐千

本文介绍水平多关节型机器人， $\mu\text{p}-80$ 微电脑套件控制它的两个步进电机和一个电磁铁，实现类似手臂关节动作，在手指部位装上钢笔，能控制其写字。

水平多关节型机器人的结构大体由肩部、肘部、上臂、前臂和手指组成。简图如图 1 所示。

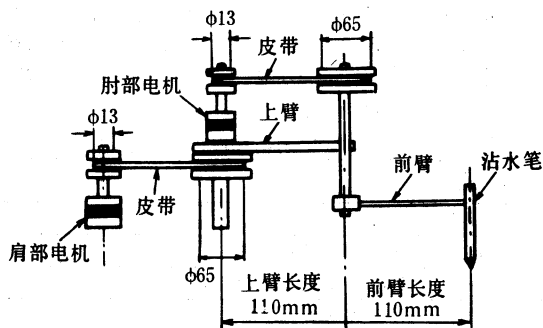


图 1

肩部和肘部驱动源均为 PXB43-01A 步进电机。上臂长度和前臂长度均为 110mm。分别用齿形皮带付传递步进电机动力，带轮减速比为 1/5。手指由吸拉型直流电磁铁与弹簧控制。如写字时，手指夹持元珠笔，电磁铁断电，在弹簧作用下，元珠笔与纸接触，上臂和前臂摆动进行写字。如果直流电磁铁通电，将元珠笔拉起，则脱离与纸的接触。手指夹持元珠笔的移动范围如图 2 所示。

上臂和前臂位置夹角为 60° 时，步进电机 1 个脉冲的移动量为最小移动量。如图 3 所示，计算方法：

$$110\text{mm} \times 1.8/360 \times 2\pi \div 5 = 0.69\text{mm}$$

式中：

$1.8/360 \times 2\pi$ ：1 个脉冲的弧度

5：为减速比

最后，取最小移动量为 0.7mm

$\mu\text{p}-80$ 微电脑套件控制步进电机与电磁铁的接口电路如图 4 所示。

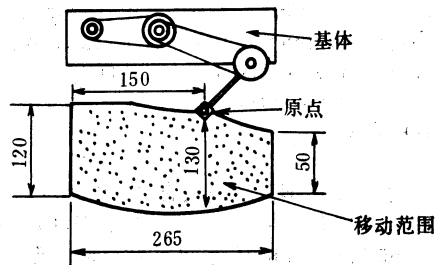


图 2

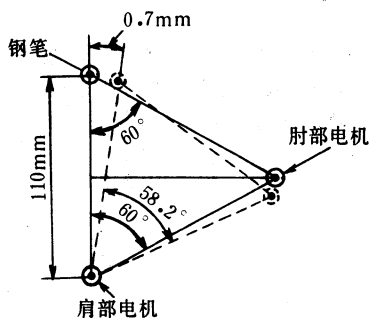


图 3

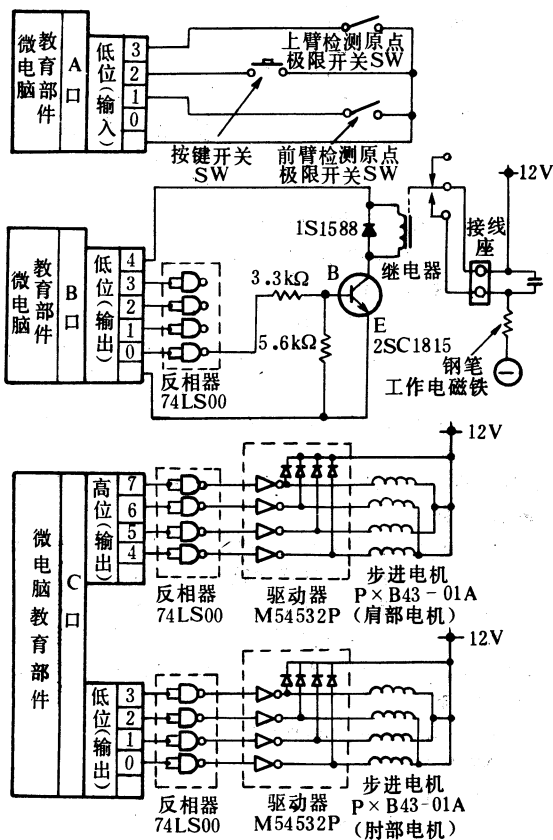


图 4

端口 C 高位控制肩部电机正、反转；端口 C 低位控制肘部电机正、反转。

端口 B 第 0 位输出 ON、OFF 信息、控制继电器部件 ON、OFF，驱动电磁铁，实现手指夹持元珠笔的上升和下降。

端口 A 的第 3 位、第 1 位分别接收上臂与前臂原点位置极限开关发出的信息。检测原点位置之后，按动启动开关，即开始写字。启动开关信息由端口 A 第 2 位接收。

操作过程是微电脑通过移位指令驱动肩部、肘部步进电机回转。端口 B 第 0 位输出 L 电平，经反相器反转为 H 电平驱动电磁铁工作。元珠笔移动到所要求的位置时，肩部电机与肘部电机暂停 0.5 秒。此时，直流电磁铁完成下降或上升动作。例如，画直线的时序是向肩部电机输入 1 个脉冲，向肘部电机输入两个脉冲。如图 5 所示。

介绍书写英文字母 P 的程序。在设置控制码、设置初始数据之后，有 4 个子程序，恢复原点、提笔、落笔、书写动作子程序。主程序流程图如图 6 所示。

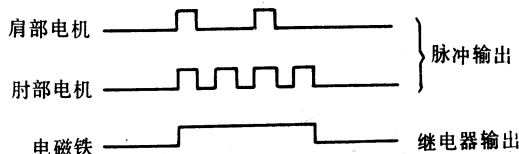


图 5

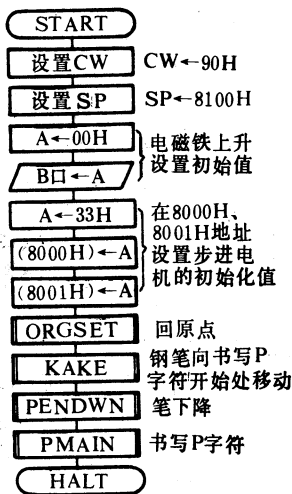
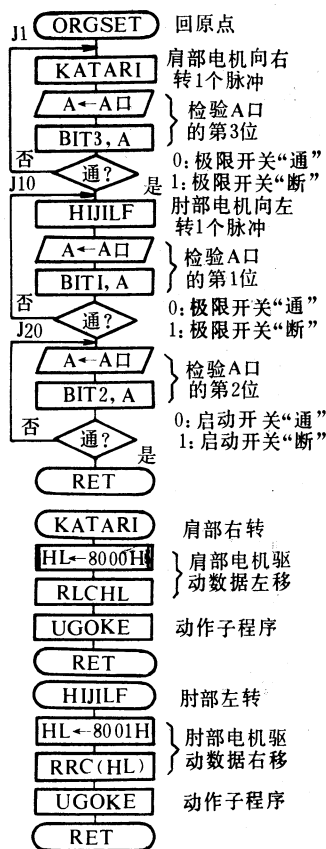


图 6

水平多关节型机器人动作起来，首先，必须决定原点位置。电源为 ON，肩部电机向右转，上臂移动到上臂原点极限开关为 ON 的位置。然后，肘部电机向左移、上臂移动到前臂原点极限开关为 ON 的位置。肘部电机继续左转，前臂移动到前臂原点极限开关为 ON 的位置。由极限开关决定的上臂和前臂的停止位置定为原点。

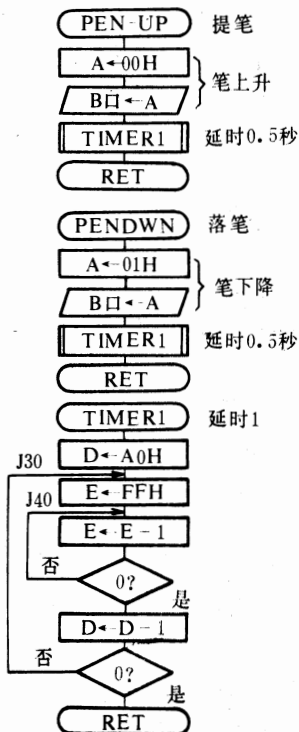


标记	助记符	地址	机器码	注释
ORGSET J1	CALL KATARI	20	CD3C00	回原点
	INA. (00H)	23	DB60	检验原点
	BIT3. A	25	CB5F	极限开关
	FPXEJ1	27	C22000	检验原点
J10	CALLHIFILF	2A	CD4500	
	IXA. (00H)	2D	DB00	极限开关
	BIT1. A	2F	CB4F	
	JPXEJ10	31	C22A00	
J20	IXA. (00H)	34	DB00	
	BIT2A	36	CB57	检验启动
	JPNEJ20	38	C23400	开关
	RET	3B	C9	返回
KATARI	LDHL8000H	3C	210080	肩部电机
	RLC(HL)	3F	CB06	驱动数据
	CALLUGOKE	41	CD6B00	
	RET	44	C9	返回
HJILF	LDHLS001H	45	210180	肘部电机
	RRC HL	48	CBOE	驱动数据右移
	CALLUGOKE	4A	CD6B00	
	RET	4D	C9	返回

图 7

检测到原点之后,按压启动开关、书写开始。恢复原点程序流程图与程序如图 7 所示。

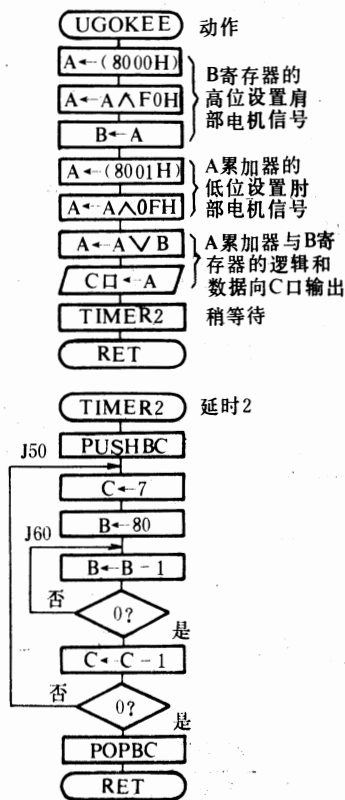
端口 B 向继电器输出 00H,继电器为 ON,输出 01H,继电器为 OFF。电磁铁与继电器相对应,产生 ON、OFF 变化,完成提笔、落笔动作。笔划、文字要隔开,在提笔、落笔之后有延时程序。提笔、落笔流程图如图 8 所示



标记	助记符	地址	机器码	注释
PENUP	LDA. 00H	4E	3E00	提笔
	OUT(01H). A	50	D3 01	
	CALL TIMER1	52	CD5E00	暂时不变
	RET	55	C9	返回
PEXDWN	LDA. 01H	56	3E01	落笔
	OUT(01H). A	58	D3 01	
	CALL TIMER1	5A	CD5E00	暂时不变
	RET	5D	C9	返回
TIMER1	LDD. A0H	5E	16A0	延时 1
J30	LDE. FFH	60	1EFF	
J40	DECE	62	ID	
	JPNEJ40	63	C2 6200	
	DECD	66	15	
	JPNEJ30	67	C2 6000	
	RET	6A	C9	返回

肩部电机、肘部电机动作程序上期讲座曾作介绍,程序流程图如图 9 所示。

书写子程序中注意两个问题,输入给肩部电机、肘

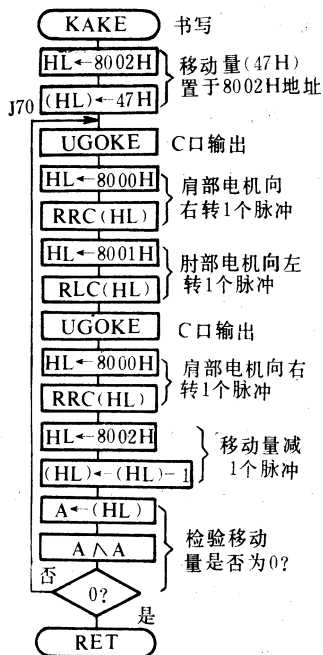


标记	助记符	地址	机器码	注释
UGOKE	LDA, (8000H)	6B	3A00 80	B 寄存器
	AND F0H	6E	E6F0	高位设置
	LDB, A	70	47	肩部电机信号
	LDA, (8001H)	71	3A01 80	B 寄存器低位
	AND 0FH	74	E60F	设置肘部电机信号
	ORB	76	B0	
	OUT(02H). A	77	D302	向 CD 输出
	CALL TIMER2	79	CD7D00	稍等待
	RET	7C	C9	返回
TIMERZ	PUSH BC	7D	C5	延时 2
	LDC, 7	7E	0E 07	
J50	LDB, 80H	80	06 80	
J60	DEC B	82	05	
	JPNE, J60	83	C2 82 00	
	DEC C	86	0D	
	JPNE, J50	87	C2 80 00	
	POP BC	8A	C1	
	RET	8B	C9	返回

部电机的脉冲数,决定上臂和前臂移动距离,即笔的书写长度。改变输入给两个步进电机的脉冲数比例,可改变笔划的倾斜角度。

笔划方向由两个步进电机旋转方向的组合来决定。程序流程图如图 10 所示。书写英文字母 P,首先将 P 字符分解。看出可以一笔写出,分解过程与流程图如图 11 所示。

编出程序后,使用反相器部件、H/L 电平显示部件



标记	助记符	地址	机器码	注释
KAKE	LDHL, 8002H	8C	21 02 80	移动量为
	LD(HL), 47H	8F	36 47	47 脉冲
J70	CALL VGOKE	91	CD 6B 00	
	LDHL, 8000H	94	21 00 80	肩部电机向右移动 1 脉冲
	RRC(HL)	97	CBOE	
	LDHL, 8001H	99	21 01 80	肘部电机向左移动 1 脉冲
	RLC(HL)	9C	CB 06	
	CALL VGOKE	9E	CD 6B 00	
	LDHL, 8000H	A1	21 00 80	肩部电机向右移动 1 脉冲
	RRC(HL)	A4	CB 0E	
	LD HL, 8002H	A6	21 02 80	移动量减 1 脉冲
	DEC(HL)	A9	35	
	LDA, (HL)	AA	7E	检验移动量是否为 0?
	ANDA	AB	A7	
	JPNE, J70	AC	C2 91 00	
	RET	AF	C9	返回

图 10

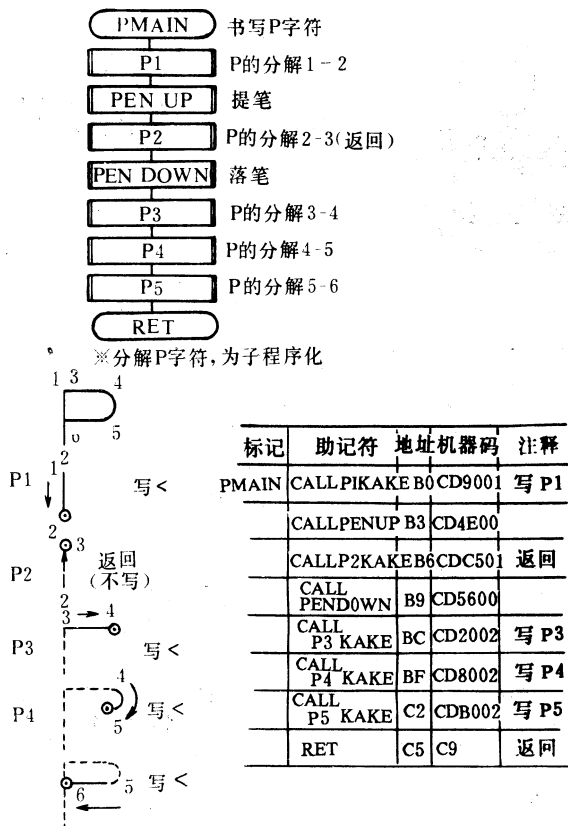


图 11

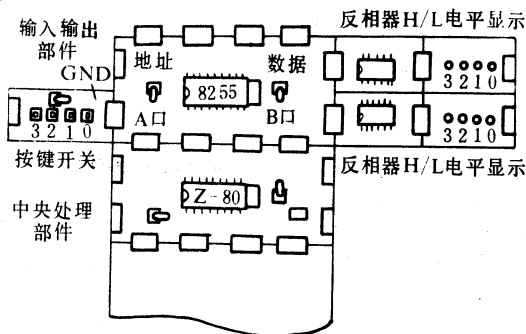


图 12

进行仿真。如图 12 所示。仿真实验时,请将 TIMER2 延时子程序的常数 07H 改为 FFH,仿真之后再改回来。

仿真结果可验证端口 C 高位相连的 H/L 电平显示部件的 LED 灯亮依次左移,按键开关 S₃ 为 ON,端口 C 低位相连的 H/L 电平显示部件的 LED 灯亮依次右移。

如果程序无误,将水平多关节机器人接通 DC12V 电源与 $\mu p-80$ 微电脑套件相连,执行程序,则在低上写出英文字母 P。

水平多关节型机器人也可用于电子元件组装、涂刷粘剂、粉刷油漆等工作。有兴趣的读者不妨可进行实验。订购该机器人请与天津纺织工学院机械系高殿斌同志联系,邮码 300160。



点阵字符液晶显示器与单片机接口

合肥中国科技大学计算中心(230026) 张培仁 杨建景

目前,液晶显示器(LCD)以其低功耗、平板化、无X射线和电磁辐射等一系列显著优点,已取代了其它显示器件成为继CRT之后的第二大显示器件。

液晶显示器件广泛应用于各种仪器、仪表、电子显示装置、计算机显示终端、电子打字机等诸方面。LCD的图形、汉字显示功能亦有极大的推广应用价值。

本文将介绍点阵字符液晶显示器和单片机的接口技术。

一、基本结构

在液晶板上排列着两行5×7点阵的的字符显示位,每行可显示12位。内存180多种字符包括英文大小写字母、数字和书写符号等,用户还可自定义8个5×7点阵的字符。PCB板上有14个引线端,其中八位数据线、三条控制线(见表1),可与微处理器相联(见图1),通过送入指令和数据可对显示内容、显示方式作出选择。

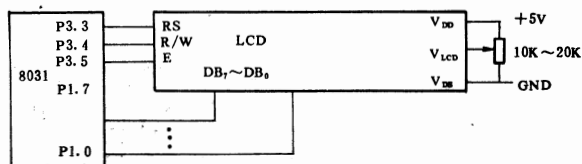


图1 与8位单片机联接示意图

表1 引线功能

引线号	符号	名称	功能
1	V _{SS}	地	0V
2	V _{DD}	电路电源	5V±5%
3	V _{LCD}	液晶驱动电压	见图1
4	RS	寄存器选择	H 数据寄存器 L 指令寄存器
5	R/W	读/写	H 读 L 写
6	E	使能	下降沿触发
7	DB0	8位数据线	数据传输
	.		
	.		
14	DB7		

二、内部结构

DMC202(日本的HD4478D)是字符点阵式的LCD。它采用CMOS工艺的超大规模集成电路,它的内部结构框图示于图2。

三、指令描述

DMC202点阵式LCD,有11条指令,可以编程控制LCD的字符显示。

1. 清屏

指令码:0 0 0 0 0 0 0 0 0 0 1

对应位:RS^{*} R/W₇ D₆ D₅ D₄ D₃ D₂ D₁ D₀(下同)

该指令的执行实际上是写空格码“20H”到所有的DDRAM中。DRAM地址计数器置0,如果光标移位,则回复到屏幕左上角。

2. 返回

指令码:000000001*

*表示无效位(下同),DDRAM地址计数器置0,DDRAM内容不变,光标回到第1行第1列。

3. 设置输入方式

指令码:00000001 I/D S

I/D: I/D=1,当将一字符写入DDRAM,地址增1,光标右移

I/D=0,当从DDRAM读出一字符码时,地址减1,光标左移

CGRAM的读写与DDRAM相同。

S: S=1,整个显示左移(I/D=1),右移(I/D=0)

4. 显示开/关控制

指令码:0000001D C B

D: D=1,显示开

D=0,显示关(显示数据保留在DDRAM中,当D=1时数据又立即显示出来。

C: C=1,显示光标

C=0,不显示光标

B: B=1,光标闪烁

B=0,光标不闪烁

5. 光标或显示移位

指令码:000001 S/C R/L **

当不写数据时光标显示仍可移位,该功能用于改错或寻找显示的数据。

S/C R/L

0 0 光标左移(AC减1)

0 1 光标右移(AC增1)

1 0 所有显示左移,光标随显示移位

1 1 所有显示右移,光标随显示移位

6. 功能设置

指令码:00001 IF N 1 **

IF: 设置接口数据长度

IF=1, 8位数据接口

IF=0, 4位数据接口,数据发送接收必须进行2

二五 . 甲、乙、丙、丁、戊、己、庚、辛、壬、癸

```

CPL P3.5          ;显示“O”
CPL P3.5
MOV P1, #01001101B
CPL P3.5          ;显示“M”
CPL P3.5
MOV P1, #01000101B
CPL P3.5          ;显示“E”
CPL P3.5
MOV P1, #00100000B
CPL P3.5          ;显示“空格”
CPL P3.5
MOV P1, #01010100B
CPL P3.5          ;显示“T”
CPL P3.5
MOV P1, #01001111B
CPL P3.5          ;显示“O”
CPL P3.5
CLR P3.3
MOV P1, #11000000B
CPL P3.5          ;DDRAM 地址设置,换行
CPL P3.5
SETB P3.3
MOV P1, #01010101B
CPL P3.5          ;显示“U”
CPL P3.5
MOV P1, #01010011B
CPL P3.5          ;显示“S”
CPL P3.5
MOV P1, #01010100B
CPL P3.5          ;显示“T”
CPL P3.5
MOV P1, #01000011B
CPL P3.5          ;显示“C”
CPL P3.5
MOV P1, #00100000B
CPL P3.5          ;显示“空格”
CPL P3.5
MOV P1, #01000011B
CPL P3.5          ;显示“C”
CPL P3.5
MOV P1, #01000011B
CPL P3.5          ;显示“C”
CPL P3.5

```

```

CLR P3.3
MOV P1, #00001011B
CPL P3.5          ;关闭整体显示开关,显示字符消失
CPL P3.5
MOV P1, #00001111B
CPL P3.5          ;打开整体显示开关,显示字符出现
CPL P3.5
MOV P1, #00000111B
CPL P3.5          ;设置成整体移动
CPL P3.5
SETB P3.3
MOV P1, #00000000B
CPL P3.5          ;显示自编字符“!”
CPL P3.5
CLR P3.3
MOV P1, #00011100B
CPL P3.5          ;整体向右移
CPL P3.5
MOV P1, #00011100B
CPL P3.5          ;整体向右移
CPL P3.5
MOV P1, #00000010B
CPL P3.5          ;返回
RET

```

程序说明:

1. 因为清屏、返回指令的执行需 1.64ms, 其它 9 条指令的执行需 40μs, 所以本程序最好采用单步执行, 否则必须传递数据前后加延时程序。

2. 自编字符的 CGRAM Data, 本程序自编字符 “!”, 其 CGRAM Data 为:

D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀
x	x	x	0	0	1	0	0
			0	0	1	0	0
			0	0	1	0	0
			0	0	1	0	0
			0	0	1	0	0
			0	0	0	0	0
			0	0	1	0	0
x	x	x	0	0	0	0	0

消 息

电子工业出版社《电子与电脑》杂志编辑部将于十月初在北京举办为期三周的“现代管理技术——计算机软件基础培训班”。

培训内容:微机 DOS 操作系统、字处理软件、FoxBASE 数据库。

培训费:330 元(含上机费。每个学员每天上机 2 小时)

资料费:50 元(实报实销)

联系人:中国人民大学信息中心(100872) 张红卫

电话:255.5431-2516(或 2514)

敬告读者

本刊原有的程序清单已全部售完,请读者不要再汇款至我部。

编辑部



F BASIC 语言的游戏程序编写技巧

电脑游戏机

山东苍山机械电子化学工业局(277700) 于 春

许多玩过电脑游戏的朋友提出这样的疑问:丰富多彩的游戏画面是怎样形成的?游戏里的主人翁为什么会受操纵器的控制?一盒录像带最多放像2~3个小时,而很小的游戏卡为什么放起来没有完?读了“键盘电视游戏机的F BASIC 语言及程序设计”一文后,基本清楚了游戏是通过编制程序实现的。因此,不仅跃跃欲试,想自己动手编一个游戏程序。但想时容易做时难,多次尝试,都不成功,实在令人沮丧。实际上,游戏程序有它独特的程序结构和编写技巧。不得要领,冒然编程,即使有经验的编程老手,也往往走弯路,何况我们初学者呢。

本文拟分游戏程序的概念、故事情节和游戏结构、程序结构和程序框图、游戏程序的设计过程、游戏程序的编制方法、游戏程序实例分析六讲,深入浅出,由简单到复杂,通过“成龙救金凤”游戏程序的编写,介绍用F BASIC 语言编写游戏程序的方法和技巧。同时,每一讲还将穿插介绍部分趣味游戏程序实例,供大家练习。最后一讲将和读者一起分析几个完整游戏程序的结构和编程技巧。从而使读者逐步弄通游戏程序的结构,逐步掌握游戏程序的设计过程,功能模块的划分及程序编写、子程序的调用链接技术,最终自己编出满意的游戏程序。

第一讲 游戏程序的概念

我们先通过两个简单的游戏实例,认识游戏程序。

例一、猜数游戏

这是一个智能类游戏。本游戏由电脑产生一个随机数字,告诉你猜数的数值范围和最多允许试猜的次数,由游戏者使用键盘输入方式猜数。若在规定的次数猜中,则奏一段明亮的音乐向你祝贺,然后进入下一关;若猜不中则放一段低沉的音乐,并提示你所猜的数值与谜底数值的比较情况,若猜的数小则提示你猜大一点的数,否则猜小点的数。若在规定的次数没有猜中,则询问是否继续游戏。你共有三次机会,若三次都没有在规定的次数内猜中则结束游戏。本游戏共11关,猜数的范围一关比一关大,难度越来越高。你若能连破11关,则高奏欢快的乐曲祝贺你的成功。

运行程序,在一段欢快的音乐声中显示出标题画面:

```
*****
*           GUESS NUMBER           *
*****
```

Please check:

STAGE 1--11 ?

画面为蓝色,其中标题 GUESS NUMBER 为黄色。提示请你选择1~11关中的任一关。假如选3,回车后,则清屏,显示:

```
*****
*           STAGE           3           *
*****
```

IN DATA<128 MAX TIME=7

TIME=1 IN DATA=?

电脑提示,现在是第三关,猜数范围小于128,最多允许猜7次,现在是第一次,请输入试猜数。假如电脑选的数是70,而我们猜的数是80,则演奏一段较低沉的音乐,在屏幕底行显示

80 is big! Give a smaller data

提醒“80大了,给一个小点的数”。这时输入提示行变为

TIME=2 IN DATA=?

如果我们猜中了70,则演奏一段明亮的音乐,清屏,显示红色的:

YOU ARE RIGHT!!!

PLEASE GO INTO STAGE 4

“你猜对了!请进入第4关”。然后清屏,显示第4关画面。

程序清单如下:

```
5 REM "NO. 1 GUESS NUMBER"
10 CLS;CLEAR;CGEN(2);A=32;B=5;G=0;A$="*
   *****"
15 FOR I=8 TO 20;COLOR I,6,2;NEXT
20 LOCATE 5,4;PRINT A$;LOCATE 5,6;PRINT "└┐
   GUESS└NUMBER└┐*";LOCATE 5,8;PRINT A$;
   LOCATE 7,12;PRINT "Please└check┐;"
25 LOCATE 7,16;PRINT "STAGE└┐1--11└┐";GOSUB
   250
30 INPUT E;IF E=11 THEN A=32767;B=15;GOTO 70
40 IF E=1 THEN 70
50 FOR I=2 TO 10;A=A+A;B=B+1;IF I=E THEN 70
60 NEXT
70 X=RND(A);F=0
80 CLS;LOCATE 5,4;PRINT A$;LOCATE 9,6;PRINT
   "STAGE└└└└┐└┐B-4;LOCATE 5,8;PRINT A$;LO-
   LOCATE 0,12;PRINT "IN└DATA└<;"A;"└MAX└
   TIME="B
90 F=F+1;IF F>B THEN 170
95 LOCATE 0,16;PRINT "└┐....."(28个空格)
100 LOCATE 0,16;PRINT "TIME=";F;"└┐IN└DATA
   =";;INPUT D
110 IF D<>X THEN 200
```

```

120 IF A=32767 THEN CLS;FOR I=5 TO 20,COLOR I,7,
1,;NEXT ;LOCATE 5,7;PRINT "CONGRATULATION
!!!";GOSUB 250;FOR I=0 TO 10000;NEXT;GO-
TO 180
130 CLS;FOR I=5 TO 20,COLOR I,7,1;NEXT;LOCATE 5,
7;PRINT "YOU ARE RIGHT!!!";GOSUB 400
140 LOCATE 5,15;PRINT "PLEASE GO INTO STAGE
" B-3;FOR I=0 TO 10000;NEXT
150 IF A=16384 THEN A=32767;B=B+1;GOTO 70
160 A=A*2;B=B+1;GOTO 70
170 CLS;G=G+1;LOCATE 5,5;PRINT "TIME IS OVER
";B;!!!!";LOCATE 4,10;PRINT "ON OR
END?(Y/N)";Y$=INKEY$(0);IF Y$<>"N"
AND G<4 THEN 70
180 CLS;LOCATE 8,8;PRINT "GOOD--BY!!!!";
LOCATE 24,20;PRINT " ";END
200 LOCATE 0,20;PRINT "....."(38 个空格);GOSUB
350;LOCATE 0,20
210 IF D>X THEN PRINT D;"is big! Give a smaller
data";GOTO 90
220 PRINT D;"is small! Give a bigger data";GOTO 90
250 PLAY "MIV12 Y3T2;M0V9M1V9Y2T2;M1T2"
260 PLAY "O3E5GA;O2G5O3CC;O2C5RE"
270 PLAY "O4C6D3O3AO4C;O2GO3EC;GRE"
280 PLAY "O3G5E6G3;O2GGO3C;CRE"
290 PLAY "DCDEG9;O2GO3CCO2GGO3C;GRECGC"
300 PLAY "E5GAO4C6D3;O2GO3CCO2GO3E;CREGR"
310 PLAY "O3AO4CO3G5E6G3;CO2GGO3C;ECRE"
320 PLAY "DCDEC9;O2GO3CC9;GRECGE"
330 RETURN
350 PLAY "M1V12Y0T2;M0V9Y2T2;MIT2"
360 PLAY "O2B3ABAGA;O2C5EG;O1G5O2C3CC5"
370 PLAY "GAO3CO2AO3CR;EC7;O1G3GG7"
380 RETURN
400 PLAY "M1V7Y1T3;M0V7M1V6Y2T2;M1T2"
410 PLAY "O3G3O4CO3GO4C;O3C5E;O2G5C3C"
420 PLAY "O3AO4CO3AGEG;GEC;C5G3GG5"
430 PLAY "EDCEC;O2GAG;C3CC5G3G"
440 RETURN

```

程序说明:

1. 15、20 行打印标题画面。其中 15 行使标题“GUESS NUMBER”(猜数)显示黄色,以使标题醒目。

2. 25~60 行为游戏者选关处理程序。游戏者可在 1~11 关中任选。从 1 关至 11 关猜数范围依次是 32、64、128、256、512、1024、2048、4096、8192、16384、32767。

3. 70~100 行,游戏者在选关后,电脑根据所选关次自动生成随机数,要求游戏者输入试猜数。

4. 110 行是没有猜中的转移处理程序。

5. 120~160 行是猜中处理程序。其中 120 行是冲破 11 关的处理程序。这时,显示红色的贺词“CONGRATULATION!!!”意思是祝贺你的成功! 130~160 行是升级处理语句,即进入下一关。每关的猜数范围均是上一关的两倍(第 1、11 关除外)。在第 10 关升第 11

关时,由于 $16384 \times 2 = 32768$ 已超出 F BASIC 取值上限,将因产生溢出而停机,因此设置了 150 句进行拦截处理。使第 10 关升 11 关时,直接令 $A=32767$,而不过 160 句。

6. 170~180 行是结束处理程序。其中 170 行记录了游戏的次数。程序中共给游戏者三次接关的机会,若接关 3 次没有猜中,则结束。

7. 200~220 提示猜数的方向。

8. 250~330 行为标题、祝贺音乐子程序。

9. 350~380 行为猜错音乐子程序。

10. 400~440 行为猜对音乐子程序。

注——程序中所有提示均使用了英文。读者如感不便,可换成汉语拼音。“ ”表示空格。

你能连破 11 关吗? 请上机试一试。

结果怎么样? 你可能侥幸猜中一、两次,但大多数猜不中,很难升级到下一关。那么能否顺利地连破 11 关呢? 回答是肯定的。

我们以第一关为例说明实现的方法。假设电脑产生的随机数是 19。由于第一关的猜数范围是 32,所以第一次我们选 0~32 中间的数,猜 16,这时电脑提示“16 小了,请猜大一点的数。”这说明所猜的数在 16~32 之间,范围缩小了一倍。第二次我们猜 16~32 的中间数 24,则电脑提示“24 大了,请猜小一点的数。”这说明所猜的数在 16~24 之间,猜数范围又缩小了一倍。第三次我们仍猜中间数 20,电脑提示“20 大了,请猜小一点的数”。第四次我们猜 16~20 的中间数 18,电脑提示“18 小了,请给大一点的数”。第五次我们猜 18~20 之间的数 19,则必然猜中。

从上述猜数过程看,每猜一次,猜数范围缩小一次,第一关允许猜五次,正好把猜数范围缩小到 1 个数。所以不论电脑选中是 32 之内的哪个数,最多五次必然猜中。如果是第二关,猜数范围是 0~64,那么我们第一次猜 32 把猜数范围缩小一倍,又与第一关相同了。由于每升一关,允许猜数次数增加一次,所以,关关能破,必能顺利通过 11 关。

以上的猜数方法,在计算机术语中叫折半查找法,是计算机数据查找技术中常用的方法。其原理已超过本文范畴;有兴趣的读者可参阅有关书籍,不再赘述。

现在你已经掌握了猜数规律,验证一下,是否实用有效。

例二 爱的结合

这是一个由电脑自动完成的动画游戏,游戏开始在简短的音乐声中,屏幕上画出了鲜红的心形图案,随后从屏幕的左下角、右下角分别走出了玛丽哥哥和丽莎小姐,两人相向而行,相遇后,显示大红的背景,奏出欢快的音乐,打印出爱的结合的祝词,结合的爱心放射出耀眼的光芒。程序清单如下:

```

5 REM "NO. 2 PERFECT"
10 PL. "M1V9Y2T3;M1V7Y1T3;M1T3"
15 PL. "O3E5G;O3C5C;O2G5G"
20 PL. "O4CO3B4A1;CC;#F#F"

```

(转第 6 页)



单色显示器故障维修一例

广东珠海拱北海关技术处(519020) 梁建华

故障现象:

一台 OPCON 牌(台湾产)单色显示器,按下开关后,无光栅,且指示灯不亮。

维修:

该显示器是采用开关电源来为行、场电路提供电压的,由于按下电源开关后,指示灯不亮,而该指示灯的电压是经开关电源整流后提供的。经检查,电源开关及整流部分均正常,初步确认为开关电源部分的故障。

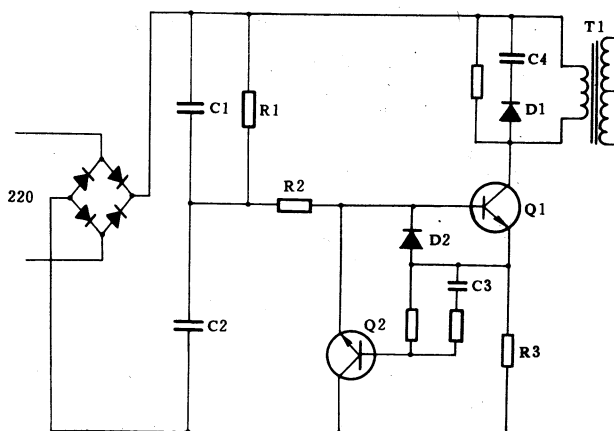
接通电源,用示波器检测开关管 Q1 的 C 极,没有锯齿波,开关管没有起振,说明是开关管及其控制部分的故障。拔掉电源,用万用表测量该部分电路,发现如图中的 Q2 管 CE 极短路,电阻 R3 开路,更换这两个元件后,再加电测试,发现有时开机正常,有时开机则无光栅,指示灯亦不亮,在出现故障的时候,关掉电源,用万用表测量时,发现当表笔碰到 Q1 的 C 极时,指示灯瞬间亮一下,说明电容所存的电荷还没有泄放完,开关管处于临界导通状态,用万用表测量开关管附近的元件,发现电阻 R2 的阻值变大,用原阻值的电阻替换后,每次开机均显示正常。

分析:

由于电压变化,电阻 R3 被烧开路,导致 Q2 管的 CE 极短路,而 R2 阻值变大,开关管 B 极电压降低,使

开关管有时不能导通,造成开机显示时有时无。值得注意的是,R2 阻值变大,往往很易忽略,由于显示时有时无,很容易错误地认为是电源开关的问题,这时,应仔细测量整流管后的电路是否有电压,开关管是否起振,如果确认故障是由开关管不起振造成的,应重点查一下与开关管导通、截止有关的元器件。

注:部分无关的线路没有画出。



图

恢复 CEC—I 学习机 DOS 的 I/O 控制

柳州钢铁厂技工学校(545002) 屈晓柳

把 DOS3.3 磁盘操作系统从磁盘装入 CEC—I 内存后,在 BASIC 程序中选择 I/O 输入输出设备时,须将“IN#S”和“PR#S”命令作为一条特殊的 BASIC 语句被调用,即命令的前面要加入 CTRL-D 或 CHR\$(4)。

语句格式:

[行号]PRINT “CTRL-D IN#S/PR#S”

或 [行号]PRINT CHR\$(4); “IN#S/PR#S”;

PRINT

说明:参数 S 表示外设所占用的槽口号,S=3 表示进入汉字系统;S=1 表示接通打印机。

如果没有把“IN#S/PR#S”命令当作特殊 BASIC 语

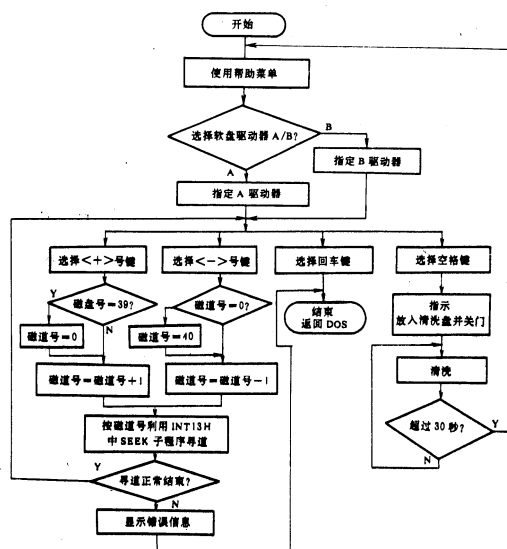
句处理,而运行含有“IN#S/PR#S”语句的程序,DOS 命令将失效,产生的原因,是监控程序的 I/O 控制寄存器 KSW(地址为 \$38,\$39 单元)和 CSW(地址为 \$36,\$37 单元)的内容指出的不是 DOS 操作系统下的 I/O 控制寄存器 KSW 和 CSW 的地址(\$9E81,\$9EBD)。

恢复 DOS 命令的功能,须改写 \$36~\$39 单元的内容,使监控程序的 I/O 控制寄存器 KSW 和 CSW 指向 DOS 操作系统下的 I/O 控制寄存器 KSW 和 CSW。即:在 \$36,\$37,\$38,\$39 单元中顺次送入如下数据 \$BD,\$9E,\$81,\$9E,更简单的办法是在监控状态下执行 3D0G/,这样系统将恢复 DOS 命令的功能。

再谈清洗盘的正确使用

安徽蚌埠 86181 部队基础教研室(233000) 刘德一

本刊 1991 年 10 期中《怎样使用清洗盘》一文,简单地介绍了清洗盘的使用方法,但忽略了一个很重要的问题,即用什么命令来调盘进行清洗。目前大多数用户均采用 DIR 命令,但它只能使磁头在清洗盘的 00 道上清洗,多次使用就会造成“越擦越脏”的现象。



清洗盘的正确使用应遵循以下原则:

1. 仔细阅读使用说明,因为目前清洗盘有干式和湿式两种,其使用方法也不尽相同。

2. 清洗时间一般在 30~60 秒之间,时间太长会损坏磁头。

3. 不能反复使用清洗盘的某一磁道,最好第一次用 00 道,第二次用 01 道,依次类推,并记录在清洗盘标签上,以便下次使用时参考。

4. 清洗盘的使用寿命一般为 15 次。

为解决磁头定位到某一磁道问题,笔者用汇编语言编写了磁头清洗软件,由于篇幅有限,这里只能介绍该软件的粗框图及编程时应注意的问题。完整的源程序已收入本人编写的《PC 机用户实用维修技法》一书,另备有软件,有兴趣者可与本人联系。

在编程时应注意以下几点:

1. 应在驱动器门打开的情况下选择磁道,以免磁头在寻道时受到损伤。

2. 正向寻道时不能超过 39 道,反向寻道时不能小于 00 道,以免磁头“内撞”或“外撞”。

3. 磁道确定后,再提示放入清洗盘并关好驱动器门,然后开始清洗。

4. 清洗时间最好让操作者随意选择。

对磁盘局部缺损的处理

广东汕头金山中学(515036) 何管略

DOS 在格式化磁盘时,会将缺损的簇号用 FF7 作标志,自动登录在文件分配表 FAT 中,此后的读、写操作,都不理会这些坏簇。

但是,磁盘使用一段时间之后,难免出现新的缺损簇号。轻度缺损,只引起读盘出错;严重缺损,将导致每次的读、写操作,都报出错。DOS 对格式化以后新出现的缺损簇号,没有提供检查、登录的功能。Copy 命令中的 /V 开关,只能帮助写盘,不会帮助读盘。Chkdsk 命令也无能为力,因为它只检查 FAT 表中的各项标志,而不检查数据区中的缺损情况。

对于这类有局部缺损磁盘,丢弃不用,似乎可惜;继续使用,又会出错,干扰正常操作,实在令人左右两难。我们从实践中摸索出几点处理体会,简介于下:

(1) 局部封存文件。如果只在个别的小型文件中,

读盘出错的话,可将该文件改名(最好将它置为不可见属性),此后对它不删、不读、不写,作坏簇处理。

(2) 重新格式化。如果有多处或对大型文件读盘出错,宜先将其其它完好的文件,转录到另外的软盘上,再将该软盘重新格式化,让 DOS 去自动登录坏簇。

(3) 补录缺损簇号。对于硬盘来说,重新格式化前后的工作量,可能很大,令人生畏。最好调用工具软件 PCTools 中的 DiskVerify 功能,它会从已用的和未用的全部簇号中,将格式化以后新出现的所有坏簇,逐个检查出来,分别补记在 FAT 表中。工具软件的不同版本,功能基本相同;区别只在于版本较高(如 6.0),速度较快,菜单位置,稍有不同而已。此法功能强大,操作简便,软、硬盘都可适用,效果很好。

电子工业出版社软件部新出版软件介绍

一、PC 及其兼容机软件

1. C 语言科学与工程程序库 软盘:1 片

定价:85 元

本软件以 C 语言函数的形式,针对科学研究与工程应用领域中的数据分析与数据处理及工程图形等实际问题,编写了数理统计、线性规划、特殊函数、非线性方程求解、工程图形、三维图形及异步通信等十几类程序。程序经过严格测试,准确无误,具有很强的实用性。

本软件可运行于各类微型计算机。

* 与本软件配套的“C 语言科学与工程程序库”一书由电子工业出版社同时出版。

2. 小学一、二年级数学语文练习 软盘:1 片 定价:70 元

本软件是按全国统一教材和教学大纲内容的要求编制的。数学部分可在 100 以内随机出题(加减法),每道练习题有解答和对与错的判断,每次可演算 10 道题。语文部分包括一年级的 630 个汉字和二年级的 817 个汉字拼读等练习。

二、中华学习机及其兼容机

1. 拼注练习 软盘 1 片 定价:35 元

本软件将学习寓于娱乐之中,集动画、音响于一体。它包括单音和多音汉字的拼音与注音、文件编辑(建立练习答案,查看和修改答案等)和打印答案。

该软件需配 STC2.0 汉字库和两台软驱动器。

2. 柴油机 软盘:1 片 定价:30 元

本软件是初中物理教学课件。它以直观图象模拟柴油机的工作过程,分柴油机构造、4 个冲程、连续工

作等部分,模拟速度可调。

3. 高能实验室 软盘:1 片 定价:30 元

本软件是高中物理教学课件。它演示“回旋加速器”、“质谱仪”、“威尔逊云室”的工作原理及其过程,形象生动。它操作简单,有汉字提示。

4. 大学英语快速阅读 软盘:2 片 定价:30 元

本软件可用于测试学生的英语阅读能力,它包括《大学英语快速阅读》4 册书的全部文章。学习者可选择“限时”和“自控时”两种方式阅读课文。读完每课后要回答 10 道与课文有关的选择題,然后计算机给出对本课的阅读理解率和阅读速度,据此即可了解学习者的阅读能力和程度,以便改进学习方式。

5. 长城加密系统 软盘:1 片 定价:75 元

它是用于整片软盘加密的工具软件,能有效地对所有在 DOS3.3 操作系统下开发的软件进行加密;加密性能可靠,加密后的软件用现在流行的解密拷贝均无法复制;加密过程简单,中途无需换盘,加密一次完成。

6. 应用软件集 ASOCS 软盘:1 片 定价:60 元

本软件集包括文本编辑、魔窗、图形处理、电子报表、记录文件、数学教师和电子琴七个部分。它内容丰富、实用性强。

文本编辑用于中英文文字处理;魔窗是英文字处理软件,它具有单词输入正确性的自检功能;图形处理可方便地使用图形移动、放大、倒置、对称和改色等;电子报表是一种通用财务软件;记录文件可作为一般性文件记录应用;数学教师是辅助数学教学(中学数学)软件;电子琴是一个多音色可变调的键盘演奏软件。

封底说明

武汉创意电子研究所系武汉东湖高技术小区的一家高技术企业。全所科技人员 21 名,其中具有硕士学位以上的有 9 名、其余皆具有大专以上学历,本所技术员吴微、马国敏、孔曙光、罗维国继 1990 年在《电子与电脑》杂志(笔名吴中国)里公布了 SCB-1 MCS-51 单片单板机以后,又于 1991 年在北航出版社出版的《单片单板机应用文休》中以 15 万字的篇幅公布了 SCB-2、MCS-51、8098 单片单板机,该机是一种性能优越、结构合理、成本低廉的两用性单板机。该机操作方便,功能完善,既便于学习,又适用应用开发,创意 CYSCB-2 MCS-51、8098 则是上面机型的改进型。下面是本所产品的报价单:

1. CYSCB-2 MCS-51、8098 单片单板机

(含主机板、键盘板、CYSCB-51 系统盘、CY8098 系统盘、通讯电缆) 980.00 元。

2. CYSCB-1 MCS-51 单片单板机

(含主板机、键盘板 CYSCB-51 系统盘、通信电

缆) 500.00 元

3. CYD-1 单片机 TV/CRT 显示接口板 400.00 元

4. CYD-1 全系列智能 EPROM 编程器 300.00 元

5. CYPRN-1 256×256×8 BITS 图像板 3500.00 元

6. CYEM-10 电镜图像分析系统(IBM386 系统一套,其中内存 2M、40M 硬盘,1.2M+360K 软驱、鼠标、TVGA 卡及显示器、扫描电路图像实时采集与输出接口、图像分析与处理软件) 25000.00 元

7. CYP-1 电脑打像机(两种配置)

①(IBMPC 主机、CCD 摄影头、14 英寸黑白监视器、24 针打印机) 14800.00 元

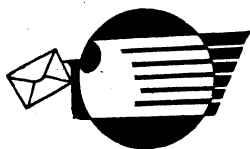
②CCD 摄像头、14 英寸黑白监视器、图像处理板、打像软件与

控制机盒、打印机配套 8800.00 元

8. CYT-1 1 加电电脑电话分路器 2000.00 元

9. 固体(全数字化)录音机 500.00 元

(转第 33 页)



普及型 PC 个人用户软件交流联谊活动 问题解答(八)

读者联谊

北京中国农科院计算中心(100081) 王路敬

24. 磁盘的存储一般格式是怎样的?

一个磁盘,不是任何位置都可以储存数据信息的。磁盘在使用之前必须经过“格式化”处理,使之符合 DOS 的规定。磁盘格式化时把盘片划分成一个个同心圆的磁道,数据是存储在磁道里的。每一个磁道又划分成一个个扇区,扇区是磁盘的基本存储单位,每当读写磁盘数据时总是读写一个个完整的扇区。例如软盘,在 DOS 2.0 或 2.10 版本下,把软盘(双密度)划分为 40 个磁道,每磁道 9 个扇区,每个扇区可存储 512 字节的内容。磁盘的两个面都可以使用。磁盘的面、磁道、扇区都是固定编号的。面的编号是 0 或 1,磁道的编号是从 0 至 39。0 磁道在磁盘的最外圈,39 磁道在磁盘的最内圈。扇区的编号是 1 至 9,整个盘片有 720 个扇区。

DOS 管理磁盘时,把磁盘上的所有扇区有规则地按一定的次序连续排列,并且尽可能地按照这样的次序使用。排列后的每一扇区也有相应的编号,称为逻辑扇区号或称相对扇区号。格式化命令 FORMAT 对磁盘进行磁道和扇区划分的同时,还把扇区的每个字节的内容置为一个特定的值——F6H,用来表示扇区是空的。

DOS 把磁盘的扇区根据用途的不同划分为四个区域,它们是:

(1) 引导记录程序区

该区只占一个扇区,即 0 面 0 道 1 扇区,相对扇区号为 0。每一个 DOS 磁盘都有一个 DOS 引导记录扇区,它是磁盘格式化操作时所建立的。这一 DOS 引导记录是磁盘启动操作系统所必须的,它在 DOS 启动时即自动装入到内存特定的区域开始运行,它的主要作用是负责将 DOS 盘上的两个系统隐含文件 IBMBIO.COM、IBMDOS.COM 装入内存,如果引导失败或者非 DOS 系统盘,则给出错误信息“Disk boot failure”或“Nonsystem disk or disk error”。

DOS 引导记录扇区中除了包括引导程序外,还包括了两张磁盘参数表。第一张表称为磁盘输入/输出参数表,它位于引导记录扇区相对字节 0BH 至 1CH。另一张表称为磁盘基数表,位于相对字节 21H 至 2BH。表中的各项参数与引导扇区中的其他数据一道描绘了整个磁盘的各种状态的使用情况,磁盘的输入/输出参数表是供 DOS 管理文件使用的,磁盘基数表是供驱动程序直接控制硬件使用的。

(2) 磁盘文件分配表区

这是一个相当重要的区域,它用来记录每一个文件在磁盘空间的分配情况,磁盘空间分配以簇为单位,也就是说分配给文件的存储空间都是一个簇,而

不管该文件是否占满整个簇,一连串的“簇链”,便指出了文件在磁盘数据存储区域中所占用空间对应的各个簇号。簇号与数据存储区域的扇区是依次对应的。在实际使用中,文件占用的磁盘扇区位置不一定是连续的,但通过“簇链”就能够搜索到文件的每一个连续的或不连续的扇区位置。一个文件的“簇链”从该文件目录登录项的“起始簇号”开始,它记载着分配给文件的第一个簇号,第一个簇号的 FAT 位置上的内容包含着第二个簇号,第二个簇号位置上的内容又包含第三个簇号,依次类推,直至分配到最后一个簇。

磁盘文件分配表简称 FAT 表,它包含以下四个方面的信息内容:

- (1) 磁盘的类型
- (2) 扇区使用情况
- (3) 指出一个文件的后继簇
- (4) 扇区封锁信息

在 FAT 表中,用 1.5 个字节即三位十六进制数来记录一个簇号,这 1.5 个字节称为一个登录项。也有一些磁盘是用二个字节来记录一个簇号的。FAT 表从第二个登录项即第 3 个字节开始记录簇号,所以有效簇号是从 002 开始的。

FAT 表的前三个字节有特殊用途。第 1—2 字节固定为 FFFFH,第 0 字节用来标识磁盘类型,规定如下:

- FFH——表示双面每道 8 扇区软盘;
- FEH——表示单面每道 8 扇区软盘;
- FDH——表示双面每道 9 扇区软盘;
- FCH——表示单面每道 9 扇区软盘;
- F8H——表示硬盘;

FAT 表每个登录项的内容包含三位十六进制数值,含义如下:

- 000H——表示一个未被使用但可以使用的簇;
- FFDH-FF6——表示保留的簇;
- FF7H——表示相应的簇为坏簇,不能使用;
- FF8-FFFH——表示该簇为文件中最后一个簇,

即表示一个文件的结束。

其他在登录项中出现的任何十六进制数值都是指针,指向文件位置分配链中下一簇的簇号。而文件的第一个簇号是由文件目录的第 26—27 字节来指出。坏簇一般是在磁盘格式化时发现缺陷被检测出来而将其打上坏簇标记不再使用。但有一点例外,计算机病毒程序常常自动地设置一些坏簇来保护自己,这种坏簇是假坏,而不是物理上的损坏。

为了确保磁盘文件建立、修改、读写或删除操作的可靠性,在实际使用中,DOS 设置了两份内容完全

一致的 FAT 表,这样即使由于盘片磨损或软件故障引起一个 FAT 表读出错误,还可以通过访问另一个 FAT 表来完成相应的操作。如果两个 FAT 表中都出现一个非法的簇域,则在使用 CHKDSK 命令时系统将显示如下信息:Allocation error for file size adjusted

该信息指出在指定的文件簇链中发现一个非法簇域,将在文件有效末尾截断该文件。如果两个 FAT 表出现更为严重的故障,则 DOS 会提出警告:

File allocation table bad drive x

表示该磁盘根本不能使用,需要重新建立。

不管是建立一个新文件,还是对一个文件进行扩展,DOS 对一个文件分配簇号时,总是扫描 FAT 表,以便找到可用簇,而不论簇号是否连续。当一张盘多次使用文件修改、删除等操作时更是如此。

(3) 磁盘文件目录表区

磁盘文件目录表具体描述文件名、子目录名、卷标及相应的信息。DOS 就是利用文件目录表掌握磁盘上每个文件所在的路径、文件属性、分配的位置、文件的长度以及建立或修改的日期和时间。当应用程序对某一磁盘文件提出读写请求时,DOS 首先在文件目录表上查找是否存在该文件。如果存在,则根据文件分配的位置并利用文件分配表及磁盘参数表,将对一个文件的请求转换成对某一相应逻辑扇区的请求。后一请求则由磁盘设备驱动程序变成实际的物理地址,最后,依赖于硬件 I/O 驱动程序来完成对磁盘的信息存取。

(4) 数据区

磁盘的数据区又称用户区。值得注意的是不同 DOS 版本及不同磁盘类型,其磁盘空间划分亦不相同,如下面表中有关数据即说明这个问题。

磁盘类型	FAT 起 始扇区	每个 FAT 长度	目录扇 区数	目录 项数	扇区 数/簇	数据区起 始扇区
单面 8 扇区	1,2	1	4	64	1	7
双面 8 扇区	1,2	1	7	112	1	0A
单面 9 扇区	1,3	2	4	64	1	9
双面 9 扇区	1,3	2	7	112	2	0C
10M 硬盘	1,9	8	20	512	8	31

25. 硬盘空间的划分有何特点?

硬盘的组织方式与软盘有其相似之处,软盘有 40 个磁道,编号为 0 到 39;10MB 硬盘相应应有 306 个柱面(与磁道等效),编号为 0—305。柱面 305 通常保留为诊断用,所以只有柱面 304 是实际可用的。10MB 硬盘有 2 个盘片,因而有 4 个面,从 0 到 3,每个盘面在每个圆柱面中为一个磁道,每个磁道有 17 个扇区,对应于软盘上的每个磁道有 8 个或 9 个扇区。扇区大小保持一致,都是 512 个字节。

一个硬盘系统可分为 4 个“分区”。划分这些分区的目的是支持 DOS 以外的操作系统。最多可以有四种磁盘操作系统,每一种操作系统在硬盘上建立自己专

用的分区操作时都由一个相应的实用程序来完成。例如 DOS 的分区实用程序名为 FDISK.COM,CP—M/86 的分区实用程序名为 HDMAINT。分区的具体数目(1—4)和 DOS 分区的大小(多少个柱面)可由 DOS 命令 FDISK 设定。需要注意的是,分区的修改将会擦掉磁盘上的数据,所以如果要改变分区而又要保留数据,唯一的办法是把数据先作备份,以后再重新装入。

硬盘空间的划分与软盘所不同的是每一硬盘上都有主引导记录和分区引导程序。硬盘的第 1 个扇区是主引导记录,它含有本硬盘的一个分区表。操作系统的启动工作由分区引导记录来做,而主引导记录程序的任务之一就是找到哪一个磁盘分区被标记为“活动分区”,然后把控制权交给该分区所对应的引导记录。至于哪个分区是实际可用的,即活动分区,可以通过调用 FDISK 命令的“改变活动分区”功能加以改变。

对于一个 DOS 分区,其文件分配表的大小、根目录大小以及每个簇的扇区数都与 DOS 的版本和分区大小有关,当 DOS 分区占据整个磁盘时,其文件分配表的大小、根目录大小与硬盘容量有密切关系。具体数据可从各类磁盘的基本输入/输出参数表查出。DOS 分区内的空间划分与软盘完全相同,也被划分为 DOS 引导区、文件分配表区、文件目录区和数据区。

26. 用户从硬盘分区信息表中可得到哪些信息?

为了实现多个微机操作系统共享硬盘资源,整个硬盘在逻辑上划分为 1 至 4 个分区。IBM—PC 及其兼容机用户可以通过 PC—DOS 随机提供的实用程序 FDISK 来指定各分区号的大小和起止的柱面号,磁道号和扇区号。分区信息表就是用来保持这些分区信息的。

硬盘分区信息表由 4 项组成,每区占一项。每一项由 16 个字节组成。整个分区信息表占 64 个字节。第一分区信息具体内容如下表所示:

字节	内容
0	引导标志
1—3	分区起始地址
4	系统标志
5—7	分区终止地址
8—11	起始扇区号
12—15	实用扇区数

第 0 字节:为引导标志。如果该字节值为 00H 表示系统加电或复位自举时不从这个分区引导操作系统。如果该字节值为 80H,则表示系统加电冷启动或热启动复位时从这个分区中引导操作系统。在系统冷启动或热启动引导过程中会检测分区信息表所有 4 项中的这个引导标志。PC—DOS 的 FDISK 分区命令可以置 PC—DOS 引导分区中这个字节值为 80H,而其余分区的这个字节为 00H。若指定从某一分区上引导操作系统,这个分区的引导程序必须在该分区的第一扇

区。

第1字节:为本分区起始磁头号

第2字节:为本分区起始扇区号。在这个字节中只有低6位是表示扇区号,而高2位则表示柱面号的最高两位。

第3字节:为本分区起始柱面号。

第4字节:为本分区操作系统代码。PC-DOS的名称代码为01H。

第5字节:本分区结束磁头号。

第6字节:本分区结束扇区号。

第7字节:本分区结束柱面号。

第8-11字节:在本分区之前已经使用掉的扇区数。

第12-15字节:本分区所使用的扇区数。

27. 如何正确的读出硬盘主引导程序和 PC-DOS 分区的引导程序?

了解这个问题首先要明确硬盘主引导程序和 PC-DOS 分区引导程序的区别和联系。硬盘的主引导程序在硬盘的第一扇区中,不属于任何一个分区。也就是说,它不属于任何一个操作系统,它是各操作系统的共同部分。它的作用就是查看分区信息表中的4个分区引导标志。当某一分区的引导标志为80H时,主引导程序就把这一分区的第1扇区读入内存0000:7C00H处,并从那儿开始执行,主引导程序是隐含扇区,不能用DEBUG程序读写它,只能用中断13H来读它,可以用下面的汇编程序:

```
MOV DX,0080
MOV CX,0001
MOV AX,0201
MOV ES,SEG BUF
MOV BX,OFFSET BUF
```

INT 13

而PC-DOS分区引导程序则占据着分区的第1扇区,当系统是从硬盘启动引导时,硬盘主引导程序在执行过程中把它读入内存0000:7C00H处,并从该地址开始执行这个分区引导程序,此时该分区引导程序做如下三方面的工作:

(1)为PC-DOS分区引导程序的执行以及以后操作系统的运行提供一张表格,详细驱动器的参数。这些参数包括:每扇区的字节数,每簇的扇区数,保留扇区数,FAT表的数目,总扇区数,介质说明字节,每个FAT表的扇区数,每道扇区数,磁头号,隐藏扇区数。

(2)根据上面这张表格提供的信息,把文件根目录区的第1扇区读入内存,查系统的两个主要隐含文件IBMBIO.COM和IBMDOS.COM是否存在。如果存在,则把IBMBIO.COM读进内存0050:0000H处并从此开始执行IBMBIO.COM。

(3)提供一张有关软盘驱动器的电参数表。PC-DOS分区引导程序不是隐含扇区,用DEBUG可以读写。

28. 软盘和硬盘 PC-DOS 分区引导程序有否差别?

软盘PC-DOS引导程序从组成主要包括磁盘驱动器的结构参数表、软盘驱动器的电参数表和引导程序三部分。对于各种各样的硬盘,其PC-DOS分区引导程序完全是一样的,差别仅在于磁盘结构参数有些项的内容不同。例如IBM-PC/XT10MB硬盘每簇扇区数为8个,而IBM-PC/AT则为4个;总扇区数IBM-PC/XT 10MB硬盘为5103H,而IBM-PC/AT为A307H等等。

不管软盘还是硬盘PC-DOS分区引导程序,这一扇区的逻辑扇区号总是0,该扇区的内容可以通过DEBUG程序装入命令L读入内存,然后通过显示内存单元内容的命令D在屏幕上显示出来,读入的时候指定起始扇区号为0,装入的扇区数为1,驱动号规定A为0,B为1,C为2。例如读C盘上PC-DOS引导扇区内容,其操作:

```
A>DEBUG
-L 100 201
-D 100 2FF
```

须指出,PC-DOS分区引导程序不一定就在硬盘的第2扇区上,因为PC-DOS分区不一定就在硬盘的第1个分区,另外即使PC-DOS分区是硬盘上的第1个分区,在硬盘的主引导程序和PC-DOS分区引导程序之间还可能有其分隐含扇区。例如IBM-PC/XT和PC/AT的隐含扇区数包括主引导程序分别为1和17。

29. 在什么情况下硬盘需要进行初始化? 有哪些方法?

硬盘初始化就是对硬盘进行物理格式化。一般在下列情况之一出现时,均需对硬盘进行初始化操作。

- (1)用户更换新的硬盘驱动器;
- (2)移动机器前没有进行磁头复位的操作,而造成磁盘面损坏;
- (3)硬盘磁头的读写时断电,使磁头划伤磁盘面;
- (4)在硬盘上试图装入几种不同的操作系统。例如,若原来盘上装有多用户操作系统XENIX和单用户操作系统DOS,这时,如果想装入另外一种多用户操作系统就需要对硬盘进行初始化。

对硬盘初始化可以通过以下三种方法进行:

(1)借助ROM BIOS程序

在控制板上的硬盘ROM BIOS程序中提供了硬盘初始化程序。如长城0520系列微机就可以在DEBUG下通过G=C800:0005命令来执行这个初始化程序。

请注意,初始化程序入口地址随控制器板的不同可能有所不同。

(2)运行LOWFORM.EXE程序

长城系列机或IBM-PC系列机及其兼容机有不少机型提供了初始化程序LOWFORM.EXE。在操作系

(转第25页)