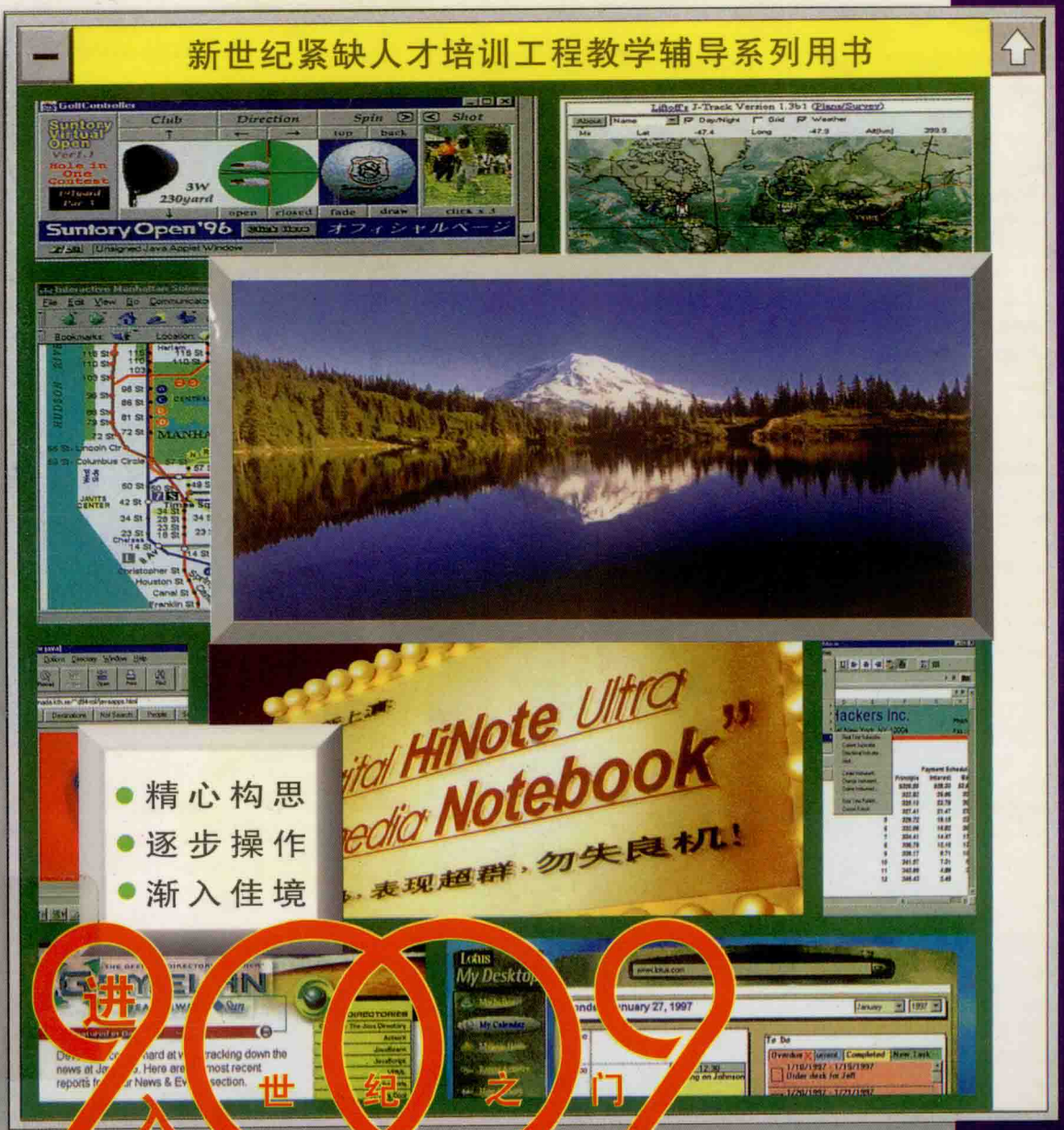
2002年秋季版

東方激光

计算机应用能力中级考试

上机实验指导

新世纪紧缺人才培训工程教学辅导系列用书



中国民航出版社

计算机应用能力中级考试
上机实验指导

王继成　　何学仪

中国民航出版社

图书在版编目(CIP)数据

计算机应用能力中级考试上机实验指导/王继成等编
——北京：中国民航出版社，2002.8

ISBN 7-80110-190-1
Ⅰ.计… Ⅱ.王… Ⅲ.电子计算机-水平考试-自学参考资料
IV.TP3
中国版本图书馆 CIP 数据核字(98)第 27011 号

计算机应用能力中级考试上机实验指导
王继成 等编
*
中国民航出版社出版发行
(北京市朝阳区光熙门北里甲 31 号楼 5 层)
江阴市天江印刷有限公司
开本：787×1092 1/16 印张：19.5 字数：468 千字
2002 年 8 月第 8 版 2002 年 8 月第 1 次印刷
ISBN 7-80110-190-1/T·045 定价：26.00 元

（发行电话：(021)63052990 本书如有印装错误，印刷厂负责调换）

前 言

作者根据2002年上海市计算机应用能力考核办公室编写的新版《计算机应用教程 中级（第四版）》、新版《计算机应用能力（中级）考核大纲》和无纸化考核的要求以及2002年7月进行的计算机应用能力（中级）考核试卷，编写了2002年秋季最新版《计算机应用能力中级考试上机实验指导》，此书更便于读者在学习2002年上海计算机应用能力考核办公室编写的新版《计算机应用教程（中级）》时，加强对所学内容的应用能力的培养，并在实验操作中加深对一些重要概念的理解，从而顺利地通过计算机应用能力中级考试。

本书紧扣新教材和考核大纲，按教材的顺序并结合重点，精心设计了14个实验，分别是“基础知识”部分2个实验，“图形界面的操作”部分4个实验，“命令行操作”部分3个实验，“数据维护”部分3个实验，“程序设计入门”部分2个实验。各实验由【实验目的】、【实验流程】、【实验导读】、【实验内容】、【自己练习】等部分组成。其中：【实验导读】突出本次实验所必须掌握的基本概念及《计算机应用教程（中级）》考核中必须掌握的基本概念；【实验流程】可使学员非常清晰地了解整个实验的过程；【实验内容】具体给出实验的要求及详细的指导步骤，并配有适当的图例说明，以便于自学；【自己练习】只给出实验要求，其解答放在“自己练习参考解答”中。请读者在进行【自己练习】的操作时，务必认真思考、独立完成，尽量不要查阅解答，以培养“脱书”动手的能力。本书还在这些实验的基础上，精编了4套综合练习题，供读者考前复习使用。

书后的附录对实验的软件、硬件环境和各部分实验注意事项进行了详细的描述和总结。

本书由王继成教授编写，何学仪策划。本书的作者长期从事计算机应用能力考试的培训，积累了丰富的教学经验，并且对中级应用能力考试的重点、难点颇有研究。相信读者按本书安排的实验认真地操作，掌握本书中的基本概念，同时按本书提供的4套综合练习题进行考前强化训练，对顺利地通过中级能力考试有很大的帮助。

在本书的编写过程中，许多从事计算机培训工作的教师提出了不少宝贵意见，在此表示感谢。

由于时间仓促，书中难免有不妥和值得改进的地方，欢迎广大教师和读者提出宝贵的意见，以利于本书的不断改进。

配套盘片使用说明

为了方便读者使用本书，我们为读者准备了配套的磁盘一张，该磁盘提供了上机实验所需的文件。这些文件都存放在磁盘的根目录下。另外，该磁盘上还建有 WARNING 子目录，其中存放着上机实验所需的文件的备份。这样，如果磁盘根目录下的某个文件被损坏或被删除，则可以从 WARNING 子目录中将该文件拷回到磁盘的根目录，继续实验。该磁盘存储的文件和子目录有：

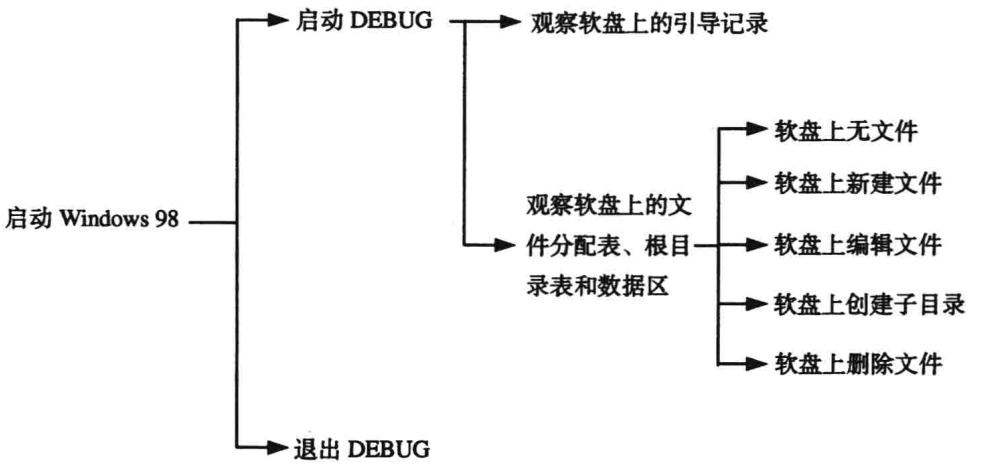
Directory PATH listing
A:\
├── AA.TXT，BB.TXT，CC.TXT，DD.TXT，ABC.DOC，ABC.TXT
├── DATA1
│　　└── AA.TXT，BB.TXT，CC.TXT，DD.TXT
├── DATA2
│　　└── USER.TXT
├── DATA3
│　　└── T2.TXT，ABC.TXT，DD.TX1，T1.TXT
├── DATA4
│　　└── A1.TXT，A2.TXT，A1.EXE，A2.EXE
├── WINDOWS
├── LX
│　　├── LX1
│　　├── LX2
│　　├── LX3
│　　└── LX4
├── XK
├── DB
│　　└── C1.DBF，CC1.DBF，CC2.DBF，CC3.DBF，CC4.DBF，CC5.DBF，CC6.DBF，CC7.DBF，
CP.DBF，CX.DBF，D11.DBF，D21.DBF，D22.DBF，D31.DBF，D32.DBF，D41.DBF，
D51.DBF，D52.DBF，DB.DBF，DB1.DBF，DB2.DBF，DB3.DBF，DB41.DBF，
DB42.DBF，DB43.DBF，DB44.DBF，DB51.DBF，DB52.DBF，DB61.DBF，
DB62.DBF，DB63.DBF，DB64.DBF，DB71.DBF，DB81.DBF，DB82.DBF，E11.DBF，
E12.DBF，E13.DBF，E21.DBF，E22.DBF，E23.DBF，E24.DBF，E31.DBF，E32.DBF，
E33.DBF，E41.DBF，E42.DBF，E43.DBF，R1.DBF，R10.DBF，R11.DBF，R12.DBF，
R13.DBF，R14.DBF，R15.DBF，R16.DBF，R17.DBF，R18.DBF，R19.DBF，R2.DBF，
R20.DBF，R3.DBF，R31.DBF，R32.DBF，R33.DBF，R4.DBF，R5.DBF，R6.DBF，
R61.DBF，R62.DBF，R63.DBF，R64.DBF，R7.DBF，R71.DBF，R72.DBF，R73.DBF，
R74.DBF，R8.DBF，R9.DBF，R91.DBF，R92.DBF，SHU.DBF，SHUDJ.DBF，
SXCJ.DBF，XS.DBF，XSY.DBF，YG.DBF，YG.FPT，YGJJ.DBF，YWCJ.DBF，
ZG.DBF，ZG.FPT，ZGK1.DBF，ZGK1.FPT
└── WARNING

目 录

第一部分

实验一　用 DEBUG 观察软盘上的系统区和数据区

- 掌握 DEBUG 的启动及退出
- 观察软盘上的引导记录（BOOT）
- 观察软盘上的文件分配表（FAT）、根目录表（ROOT）、数据区（DATA）



1. 基本概念

二进制：为了适应计算机内部的电子电路，计算机中处理、存储的各种数值（包括存储器的地址）均采用“二进制”形式表示。二进制有两个基本符号：0、1。二进制的进位制为：逢二进一。二进制数的后缀为：B。

十六进制：十六进制可以看作是二进制的紧凑表示，书写比较方便。十六进制有十六个基本符号：0、1、2、3、4、5、6、7、8、9、A、B、C、D、E、F，分别表示二进制的 0000、0001、0010、0011、0100、0101、0110、0111、1000、1001、1010、1011、1100、1101、1110、1111。

十六进制的进位制为：逢十六进一。十六进制数的后缀为：H。

字节：计算机存储器的基本单位是字节，微机的 CPU 进行数值计算时，也是以字节为基本单位。一个字节由 8 个二进制位组成，因此可以表示的数值范围是 0 到 11111111B（或 FFH，或 255）。如果需要表示更大的数，可以用连续的若干个字节组合起来。

整数（2^n 或 2^n-1）在二进制、十六进制、十进制下的表示：

整数（2^n 或 2^n-1）	二进制	十六进制	十进制	
2^0	1B	1H	1	
2^1	10B	2H	2	
2^2	100B	4H	4	
2^3-1	111B	7H	7	
2^3	1000B	8H	8	
2^4-1	1111B	FH	15	
2^4	10000B	10H	16	
2^5-1	11111B	1FH	31	
2^5	100000B	20H	32	
2^6-1	111111B	3FH	63	
2^6	1000000B	40H	64	
2^7-1	1111111B	7FH	127	
2^7	10000000B	80H	128	
2^8-1	11111111B	FFH	255	
2^8	100000000B	100H	256	
2^9		200H	512	
2^{10}		400H	1024	1K
2^{16}		10000H	65536	64K
2^{20}		100000H		1M
2^{30}		40000000H		1G
2^{40}		10000000000H		1T

浮点数：为了表示大范围和高精度的数值，计算机中使用一种称为“浮点数”的表示方法。一个浮点数由符号位、尾数、阶码三部分组成。符号位的“0”或“1”表示这个数为正或负，尾数用来表示有效数字，阶码则表示尾数乘上 2 的多少次方才等于最终的数值。尾数的位数与表示数值的精度有关，阶码的位数与表示数值的范围有关。

内存：数据（包括组成系统程序和应用程序的指令，以及被处理的数据）在使用时都要存放在内存储器中。内存有许多存储单元，每个存储单元可以存放一个字节的数据。为了能将数据存储到指定的存储单元或从内存的指定单元中取出数据，都需要给出存储单元的位置信息，这也就是内存地址的由来，即给每一个存储单元编号，此编号就作为该存储单元的地址。例如，64K 的内存，其地址范围为：0000H ~ FFFFH。随着计算机寻址能力的增强和内存容量增大，内存地址用段地址和偏离量地址（位移量）来描述，即每个内存单元的地址表示为：“段地址：偏离量地址”，而该存储单元的真实（物理）地址是由它的段地址和偏离量地址确定的。

ROM 存储器：ROM 存储器的基本属性是“只读不写”，即在一般运行状态下，系统软件和应用软件只能从中读取数据，而无法写其中的数据。近年来，微机的这部分内存采用了称为“电可擦写 ROM”（EEPROM 或 E^2PROM，目前使用的品种又称闪速 EPROM、Flash EPROM）的存储元件，在微机的正常工作状态和关机状态下，其功能与普通的 ROM 相同；运行专门的程序，可以通过微机内专设的电子线路，使其进入像 RAM 一样的工作状态，改写其中的内容，退出这种状态后，新的内容又被长期保存。

CPU：早期的 IBM PC 微机使用的 Intel 8086（或 8088）是 16 位 CPU，其运算和内部传送数据的规模是 16 位，能指定 16 位的内存地址，但可以通过一种特殊的方法，把 CPU 指定地址的能力增加到 20 位，内存地址的最大允许范围（称为地址空间）是 0 到 FFFFFH，其实际内存容量不会超过 1M。随着越来越先进的 CPU 的问世，CPU 存取内存时指定地址的范围越来越大。

从80386开始到低档的Pentium，都是32位CPU，能指定32位的内存地址，它们的内存地址空间达到 4GB。当然，这只是内存容量的最大可能值，真实的内存（称为物理内存）容量还取决于实际安装RAM存储器的大小。从PentiumⅡ开始，Pentium CPU指定内存地址的能力已超过32位，但目前在一般的应用中尚未真正发挥这一能力。

兼容性：为了保持兼容性（就是让原来在旧机器上运行的软件转到新机器上同样能够正常运行），内存容量超过1M时，地址从A0000H到FFFFFH一段的用途仍然不变，其中包括：存放屏幕显示内容的RAM；存放操作系统的重要组成部分——基本输入输出系统（Basic Input and Output System）的ROM，常称作ROM BIOS；用于扩充ROM BIOS的ROM。同样是为了保持兼容性，32位CPU被设计成“两用”的——在一种称为“保护模式”的工作方式下，它们可以直接存取4G的内存，在另一种称为“实模式”的工作方式下，则模拟低档的8086（当然速度要远高于8086），使各种旧的软件也能在其上正常运行。

16位软件：早期的微机系统软件和应用软件受限于16位微机的硬件环境，工作在1M的内存范围内，它们自身的程序则被限制在640K以内。人们把这些软件称为“16位软件”。曾经在微机上得到最广泛应用，至今尚未完全退出历史舞台的DOS操作系统，就是16位系统软件的代表。在16位系统软件的基础上，只能运行16位的应用软件。Windows 3.x虽然能更好地利用大的内存，但仍然属于16位软件。

32位软件：32位软件就是工作在保护模式下的软件，在此模式下，“1M”这一历史形成的分界线不复存在，内存空间成为一个整体，程序和其他用户数据都可以按需占据其中的任何部分，解除了对程序规模和布局的束缚，大大提高了软件的运行效率，拓展了软件功能的发展空间。Windows 9x、Windows NT等操作系统都是32位的系统软件。针对这些操作系统专门设计的应用软件一般是32位软件。但是，由于设计时考虑了对旧有软件的兼容性，这些操作系统也能让许多旧的16位软件在其上正常运行，当然，这时的运行效率仍然是16位的水平。

影像：由于元器件的固有特性，从ROM中读取数据要比从RAM中读取慢得多。为了提高工作速度，微机有这样一种功能：在启动时，自动地把ROM BIOS和扩展BIOS中的内容部分或全部复制到RAM中保存，称为“影像”（shadow），以后需要使用时，从RAM中读取而不再涉及ROM，这就提高了工作速度。

高速缓冲存储器（Cache）：在CPU读取内存的某个地址中的数据时，计算机自动地将地址相近的整批数据从RAM中传送到一组由高速RAM组成的高速缓冲存储器中，以后，CPU再要读取数据时，如果这个数据已在高速缓存中（称为命中），则立即快速地从高速缓存中取得；如果不在，才从RAM中读取，并且又是同时传送一批数据到高速缓存中。高速缓存有两个层次：一层与CPU电路直接紧密结合在一起，称为一级缓存（L1）；另一层不与CPU电路直接紧密结合（但有时也与CPU封装在同一芯片内），称为二级缓存（L2）。高速缓存的容量也是微机的一项重要技术指标。高速缓存不属于内存储器，不占用内存的地址空间。

虚拟内存：虚拟内存技术是一种利用硬盘的支持，以小的物理内存来模拟大的内存，使程序能够使用整个内存地址空间的技术。一个程序在有虚拟内存的微机系统中运行时，能够对整个内存地址空间（例如4GB）中的任何一个地址进行存取，就像那里真的有RAM一样，而实际上这个系统可能只装有小得多的物理内存（例如64MB）。采用虚拟内存技术运行程序时，程序实际使用的内存可以分散在整个内存地址空间，但使用的总块数不能超过物理内存的块数。采用虚拟内存技术把内存“变大”的代价是程序运行的速度有所降低。有了虚拟内存后，应用程序在运行时“感觉”到的内存容量主要取决于两个因素：一是CPU决定的内存地址空间大小，二是硬盘实际可用空间的大小。

显示存储器：显示存储器（常简称为“显存”）是保存显示内容的RAM。

扇区：扇区是计算机对磁盘读写数据的基本单位。一个扇区一般存放 512 字节数据。软盘的扇区标志是在格式化时生成的，而硬盘的扇区标志是在低级格式化时生成的。一张已格式化的 3.5″ 1.44M 的软盘有 2 个面（side），每个面有 80 个磁道（track），每个磁道有 18 个扇区（sector）。

物理扇区号：使用面号、磁道号、扇区号对整个盘上的所有扇区进行编号。例如，3.5″ 的 1.44M 软盘，其面号的范围是 0～1，磁道号的范围是 0～79，扇区号的范围是 1～18。

逻辑扇区号：对于整个磁盘上的所有扇区，都用连续的、确定的序号来编号，该编号称为逻辑扇区号。磁盘的逻辑扇区号是从 0 开始编号的。

系统区：对于采用 FAT12、FAT16 的磁盘，系统区包括：磁盘上的主引导扇区、引导记录、文件分配表、根目录表。对于采用 FAT32 的磁盘，系统区包括：磁盘上的主引导扇区、引导记录、文件分配表。系统区供操作系统使用，不直接对用户开放。硬盘上的第一个扇区为主引导扇区，软盘上的第一个扇区为引导记录，它们的大小都是 512 字节。

数据区：磁盘上真正存放文件内容和子目录表的区域被称为数据区。数据区紧随系统区之后。对于采用 FAT12、FAT16 的磁盘，数据区不包括根目录表。而采用 FAT32 的磁盘，数据区包括根目录表。

主引导扇区：硬盘上的 0 面 0 道 1 扇区称为主引导扇区，其中包含着分区表和主引导程序。分区表中存放着硬盘各个分区的情况，还指明机器启动时使用的分区（活动分区）。主引导程序（其长度小于 512 字节）的主要任务是查看分区表，找到活动分区后，即从这个分区的起始扇区读入引导记录。主引导扇区是在对硬盘进行低级格式化时生成的。

引导记录：引导记录又称引导扇区、引导区、启动块、BOOT 区等，其中除了引导程序（其长度小于 512 字节）外，还包含着一些有关本张磁盘规格的重要参数，供操作系统使用。引导程序的任务是先检查本张磁盘上是否有文件 IO.SYS 和 MSDOS.SYS，如果存在，则读入这两个文件，逐步完成整个启动过程；如果不存在，则显示“Invalid system disk”的错误信息。引导记录是在格式化时生成的。对于软盘来说，它位于 0 面 0 道 1 扇区（第 0 号逻辑扇区）。对于硬盘来说，硬盘可以建有若干个逻辑盘，例如 C、D、E、…，而各逻辑盘均有独立的引导记录，它们分别位于各逻辑盘内的第 0 号逻辑扇区。

文件分配表：文件分配表中有许多表项，每项记录一个簇的信息（其中包括：该簇是否被占用、是否坏簇、是否有后继簇等），表项的数目与磁盘上的总簇数相当，每一项占用的字节数也取决于总簇数。文件分配表最初在格式化时形成，在磁盘上位于引导扇区之后，占据若干扇区，一式两份，通常称为 FAT1 和 FAT2。在正常情况下，初形成的文件分配表中所有表项都标记为“未占用”，但如果磁盘有局部损坏，那么格式化程序会检测出损坏的簇，在相应的表项中标记为“坏簇”。对于 3.5″ 的 1.44M 软盘，FAT1 占据 9 个扇区，其逻辑扇区号为（1H～9H），FAT2 也占据 9 个扇区，其逻辑扇区号为（AH～12H）。

目录表：为了管理磁盘上的文件，操作系统为磁盘上根目录和所有子目录都建立一个目录表。目录表由若干表项组成，每个表项对应一个文件或子目录，用来描述文件或子目录的名字、属性、创建时间、创建日期、最后访问日期、最后修改时间、最后修改日期、起始簇号、文件长度等信息。对磁盘格式化时，操作系统在磁盘上生成根目录表，位置在文件分配表之后，其中所有表项都是空的。以后在根目录中每建立一个新的文件或子目录，就占用一个或若干个空的表项并填入相应内容，每个表项 32 个字节。子目录表是在创建子目录时生成的，它的构造与根目录表相同，创建时先占据一个簇，以后随着其中文件个数的增加还可以扩大到更多的簇。当文件名或子目录名为长名时，目录表除了用一个表项存放该长名所对应的短名和这个文件或子目录的其他目录信息外，还用连续的若干个表项存放长名，每个表项存放 13 个字符（使用 Unicode 码，不论是西文还是汉字，一律占用 2 个字节）。存放长名的表项位于存放其对应短名的表项之前。

簇：簇（cluster）又称为分配单位（allocation unit）。文件占据磁盘空间时，以“簇”为基本单位。簇的大小与磁盘的规格有关，5″ 360K 软盘每簇 2 个扇区，5″ 1.2M 和 3.5″ 1.44M 软盘每簇 1 个扇区，硬盘则有每簇 4 个、8 个、16 个或更多个扇区。与扇区的逻辑编号相似，簇也有自己的顺序编号，称为簇号，簇是从 2 开始编号的。对于 3.5″ 1.44M 软盘，数据区存储空间的“簇号”与其“逻辑扇区号”的关系为：簇号 = 逻辑扇区号-1FH。例如，“簇号”为 2 的磁盘存储空间，其“逻辑扇区号” = 2+1FH = 21H。

2. DEBUG 的基本命令

DEBUG 是 Microsoft 公司的操作系统产品中附带的程序，能够直接观察、修改内存中指定地址的当前数据，直接读、写磁盘的指定扇区。DEBUG 的程序包含在文件名为 DEBUG.EXE 的文件中，存放在 C:\Windows\Command 中。DEBUG 的操作命令都以单个字母开始，大小写不论。在命令格式中出现的“<地址范围>”应当按需要替换为以下两种形式之一：<起始地址> <终止地址>或<起始地址>L<字节数>。命令中的“地址”允许的取值范围是 100H ~ FFFFH（H 表示十六进制数，但在命令中 H 必须省略）。下面介绍 DEBUG 中一部分最简单的命令：

（1）显示内存中的数据

格式：D[地址范围>]

功能：以 16 进制数值和 ASCII 字符两种形式，显示内存中地址范围内的数据。每行显示 16 个字节，左端是这 16 字节的起始地址（段地址：偏离量地址），中间是以 16 进制表示的数据值，右边是这些数据值对应的西文字符（即将每个字节的内容当作 ASCII 码看待，如果数据值超过 20H ~ 7EH 的范围，则右边的对应位置显示一个点。首次 D 命令的地址范围缺省相当于 100L80，非首次 D 命令的地址范围缺省时，自动接续上一次 D 命令的地址范围。

实例：若在提示符“-”后输入：D100 2FF↙

则显示偏离量地址从 100H 到 2FFH 一段内存中的内容。

若在提示符“-”后输入：D100L200↙

则显示偏离量地址从 100H 开始的 200H 个内存单元中的内容，事实上，其功能与 D100 1FF 的功能相同。

（2）编辑数据

格式：E<地址> [<数据>]

功能：修改内存中从指定地址开始的数据。不指定<数据>时，逐字节显示原数据并接收新数据，每个新数据的输入以空格键结尾，需要修改的数据全部输入后，以回车键结束。指定<数据>时，不加显示，按指定地址连续写入内存取代原数据。多个数据以空格分隔，字符要加单引号，引号内可以是多个字符的字符串。

实例：若在提示符“-”后输入：E200↙

则从偏离量地址 200H 开始，逐字节地显示内存单元的内容，并从键盘接收新的数据写入。

若在提示符“-”后输入：E200 1 2 ‘A’ 3 ‘bcd’ ↙

则偏离量地址 200H 到 206H 的 7 个内存单元的内容成为 01H、02H、41H、03H、62H、63H、64H。

（3）将磁盘上的数据读到内存（读磁盘）

格式：L<地址> <驱动器号> <逻辑扇区号> <N>

功能：从指定的驱动器的指定扇区开始，读取连续 N 个扇区的数据到内存的指定地址。驱动器号为：0、1、2、……，依此对应驱动器 A、B、C、……。

实例：若在提示符“-”后输入：L100 0 0 1↙

则将 A 盘上的引导记录读到偏离量地址从 100H 开始的 512 个内存单元中。

若在提示符“-”后输入：L100 0 1 9↙

则将 A 盘上的文件分配表 1（FAT1）读到偏离量地址从 100H 开始的 4608（512*9）个内存单元中。

（4）将内存中的数据写到磁盘（写磁盘）

格式：W<地址> <驱动器号> <逻辑扇区号> <N>

功能：将内存指定地址开始的数据，写到指定驱动器的指定扇区开始的连续 N 个扇区中。驱动器号为：0、1、2、……，依此对应驱动器 A、B、C、……。

实例：若在提示符“-”后输入：W100 0 0 1↙

则将偏离量地址从 100H 开始的 512 个内存单元中的内容写到 A 盘上的引导记录（即 A 盘的第 0 号逻辑扇区）。

若在提示符“-”后输入：W100 0 1 9↙

则将偏离量地址从 100H 开始的 4608（512*9）个内存单元中的内容写到 A 盘上的 FAT1（即 A 盘上的第 1 号逻辑扇区到第 9 号逻辑扇区）。

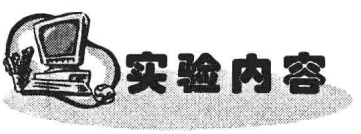
（5）退出 DEBUG

格式：Q

功能：结束 DEBUG 的运行。

实例：若在提示符“-”后输入：Q↙

则退出 DEBUG。



一、DEBUG 的启动及退出

1. 启动 DEBUG

【实验】

启动 DEBUG。

【指导步骤】

（1）单击“开始”按钮，选择“运行”命令，出现“运行”对话框。

（2）在该对话框的“打开（O）”文本框中，输入 C:\Windows\Command\Debug，如图 1-1-1 所示。

（3）单击“确定”按钮，则启动 DEBUG。

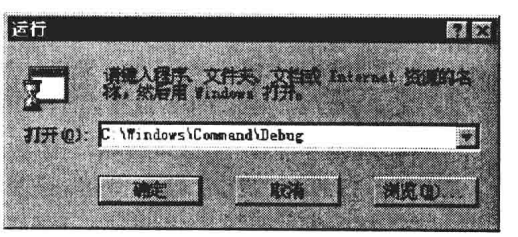


图　1-1-1

2. 退出 DEBUG

【实验】

退出 DEBUG。

【指导步骤】

在“DEBUG”窗口中的提示符“-”后输入 Q↙（或 q↙）。

二、观察软盘上的引导记录（BOOT）

【实验】

取一张格式化过的空白软盘插入 A 驱动器，用 DEBUG 将引导记录读入内存，观察其内容，找出该磁盘规格的重要参数（每簇扇区数、FAT 的个数、根目录表项数、一个 FAT 占用的扇区数）。

【指导步骤】

（1）启动 DEBUG。

说明：如 DEBUG 已启动成功，此步骤可跳过。

（2）将一张格式化过的空白软盘插入 A 驱动器。

（3）在“DEBUG”窗口中的提示符“-”的后面键入命令：

L100 0 0 1↙

如图 1-1-2 所示。

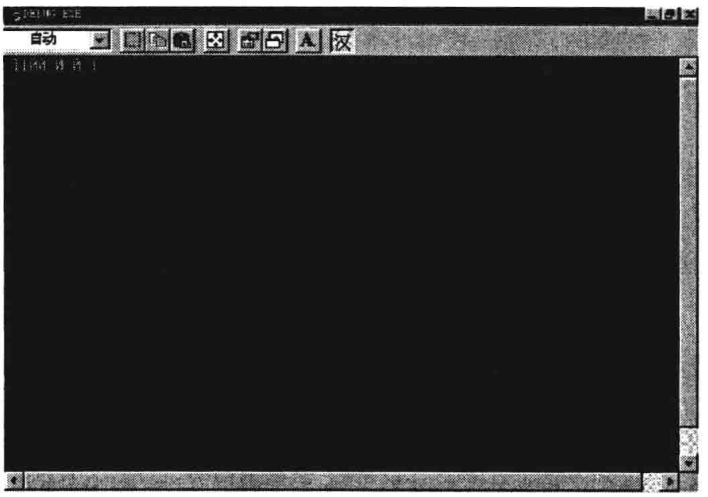


图　1-1-2

说明：以上命令的含义是将 A 驱动器中的软盘上的第 0 号逻辑扇区中的内容（即引导记录）读入内存（其偏离量地址为 100H～2FFH）。

（4）在提示符“-”后面键入命令：D100 L100↙

说明：该命令显示内存（偏离量地址为 100H～1FFH）中所存放的数据（即第（4）步读入内存的磁盘引导记录的前半部分的 256 字节数据），如图 1-1-3 所示。其中左侧是每行第一个数据的内存地址（地址由两部分组成，即<段地址>:<偏离量地址>），用十六进制表示，每行的第 2、3、...16 个数据的内存偏离量地址分别在该行的第一个数据的偏离量地址基础上加 1、2、...、15，段地址不变；中间是 256 个 2 位十六进制数，分为 16 行，每行 16 个；右侧是 16 行字符，每行 16 个，如果中间显示的某个内存单元的数据是一个字符的 ASCII 码，那么右侧的对应位置上显示的就是这个字符，对控制符的 ASCII 码，右侧会显示一些无意义的特殊图案。观察时应当记住，对于并非字符含义的数据（例如引导程序中的指令、文件分配表中的簇号等），其值也会是某个字符的 ASCII 码，这时右侧也显示相应字符，这个字符与数据的真实含义是毫不相干的。

（5）从图 1-1-3 可以看出：偏离量地址为 10DH 的存储单元中的数据为 01H，偏离量地址为 110H 的存储单元中的数据为 02H，偏离量地址为 112H、111H 的存储单元中的数据分别为 00H，E0H，偏离量地址为 117H、116H 的存储单元中的数据分别为 00H，09H。根据下表：

偏离量地址	字节数	含义
10DH	1	每簇扇区数
110H	1	FAT 的个数
111H	2	根目录表的数
116H	2	一个 FAT 占用的扇区数

可以发现该软盘的规格参数为：每簇为 1 个扇区，有 2 个 FAT 表，根目录表的个数为 E0H（224），一个 FAT 占用 9 个扇区。

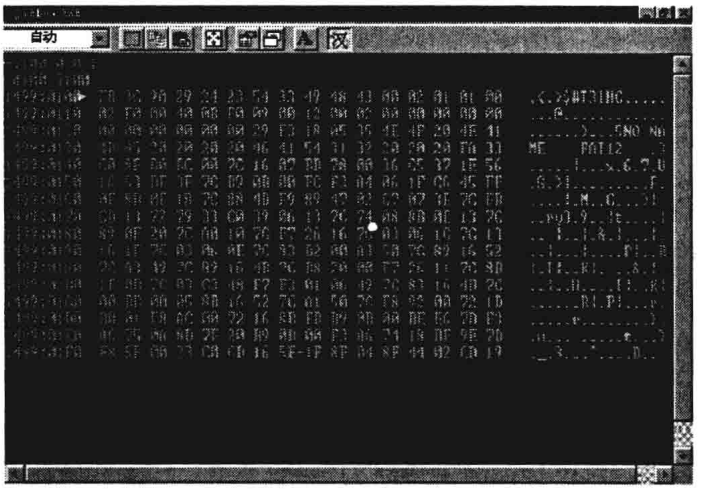


图　1-1-3

（6）在提示符“-”后面键入命令：

D200 L100↙

说明：该命令显示内存（偏离量地址为 200H～2FFH）中所存放的数据（即第（4）步读入内存的磁盘引导记录的后半部分的 256 字节数据），如图 1-1-4 所示。

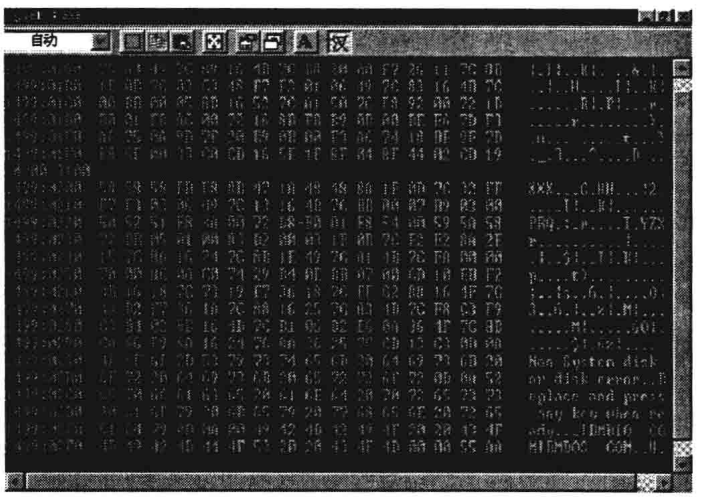


图　1-1-4

三、观察软盘上的文件分配表（FAT）、根目录表（ROOT）和数据区（DATA）

1. 观察空白软盘上的文件分配表（FAT）、根目录表（ROOT）和数据区（DATA）

【实验 1】

观察空白软盘上的文件分配表（FAT）。

【指导步骤】

（1）在“DEBUG”窗口中的提示符“-”后面键入命令：

L100 0 1 12↙

说明：以上命令的含义是将A驱动器中的软盘上的第1号逻辑扇区～第18号逻辑扇区（共18个扇区）读入起始地址的偏离量为100H的一段内存中（即将FAT1和FAT2读入内存）。

（2）为了能在屏幕上看清FAT1各扇区的内容，在提示符“-”后键入命令：

D100 L100↙

说明：该命令显示内存（偏离量地址为100H～1FFH）中所存放的数据（即第（1）步读入内存的磁盘FAT1的第一个扇区的前半部分的256字节数据），如图1-1-5所示，其中最左边的段地址“1499”会随环境不同而不同，读者可不加理会。

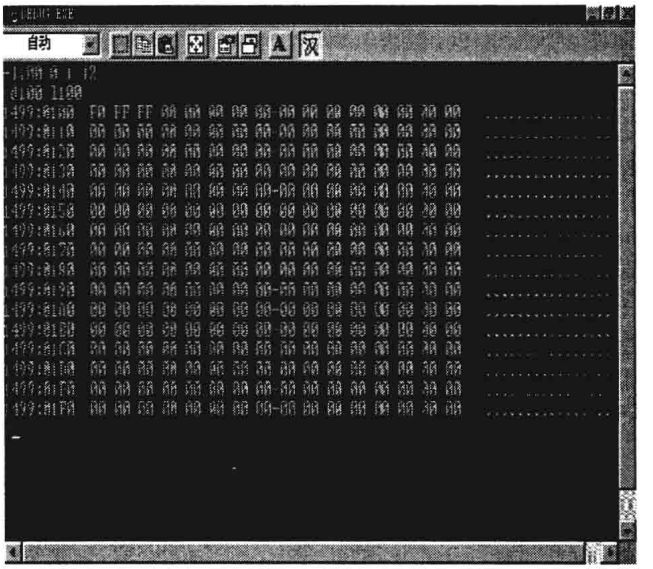


图　1-1-5

从图1-1-5可以看出除了开始三个字节（代表第0、1表项）为F0 FF FF外，其余均为0。

（3）依次在提示符“-”后输入命令：

D200 L100↙
D300 L100↙
D400 L100↙
：
：
D1200 L100↙

可查看读入内存的FAT1的剩余内容，发现其中均为0，因此从FAT1中的内容可以看出磁盘上从未写过文件。

注意：尽管磁盘上没有写过文件，但若磁盘上有坏的扇区，则文件分配表中的内容也不可能全为0。

（4）在提示符“-”后键入命令：

D1300 L100↙

说明：该命令显示内存（偏离量地址为1300H～13FFH）中所存放的数据（即第（1）步读入内存的磁盘FAT2的第一个扇区的前半部分的256字节数据），如图1-1-6所示。

比较图1-1-6和图1-1-5，FAT2的第1个扇区的前半部分与FAT1的第1个扇区的前半部分相同，即除了开始三个字节（代表第0、1表项）为F0 FF FF外，其余均为0。

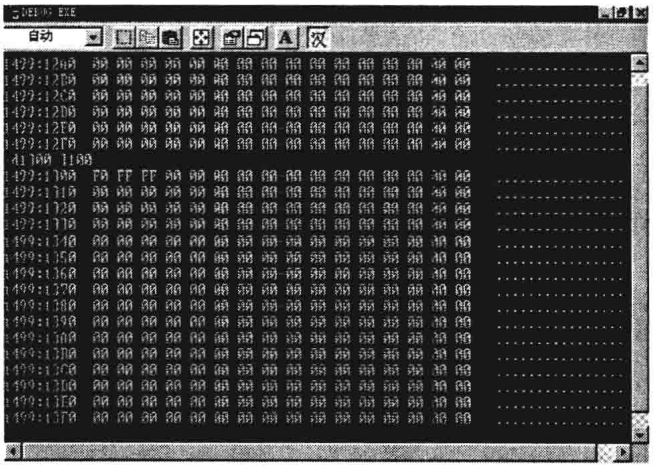


图　1-1-6

（5）在提示符“-”后键入命令：

D1400 L100↙
D1500 L100↙
D1600 L100↙
：
：
D2400 L100↙

可查看读入内存的 FAT2 的剩余内容，发现其中均为 0，说明磁盘上从未写过文件，并且从 FAT2 中的内容可以看出 FAT2 的内容与 FAT1 的内容完全相同。

【实验 2】

观察空白软盘上的根目录表（ROOT）。

【指导步骤】

（1）在“DEBUG”窗口的提示符“-”后面键入命令：L100 0 13 E↙

说明：以上命令的含义是将 A 驱动器中的软盘的第 19 号逻辑扇区～第 32 号逻辑扇区（共 14 个扇区）读入内存，起始地址的位移量为 100H 的一段内存中（即将 ROOT 读入内存中）。

（2）在提示符“-”后键入命令：D100 L100↙

该命令的含义是显示第（1）步读入内存中的 ROOT 表的第一个扇区的前半部分的内容，如图 1-1-7 所示。从图 1-1-7 中可以发现其内容全为 0。

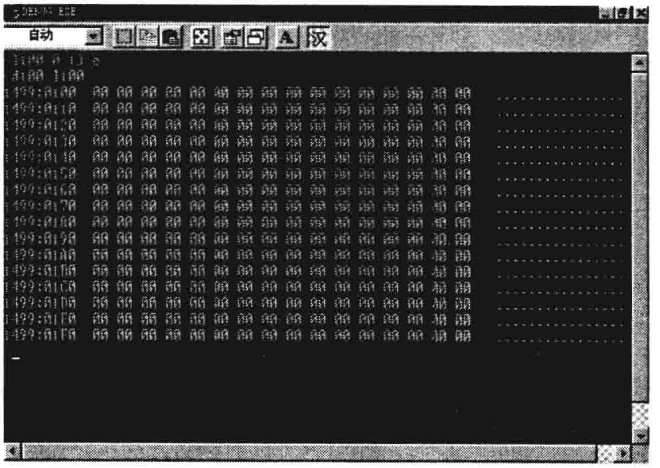


图　1-1-7

注意：若在实验过程中，发现其内容不全为0，则可能你在格式化软盘时，输入了磁盘的卷标。

（3）依次在提示符“-”后键入命令：

D200 L100↙
D300 L100↙
D400 L100↙
：
：
D1C00 L100↙

可以查看出读入内存中的ROOT表的剩余内容，发现全为0，表明磁盘上从未写过文件。

【实验3】

观察空白软盘上的数据区（DATA）。

【指导步骤】

（1）在“DEBUG”窗口的提示符“-”后面键入命令：

L100 0 21 10↙

说明：该命令的含义是将A驱动器中的软盘上的第33号逻辑扇区～第48号逻辑扇区（共16个扇区）读入内存（其起始地址的位移量为100H），即将数据区的第1个扇区～第16个扇区的内容读入内存。

（2）在“DEBUG”窗口的提示符“-”后面键入命令：D100 L100↙

说明：该命令的含义是显示第（1）步读入内存中的数据区的第一个扇区的前半部分的内容，如图1-1-8所示。

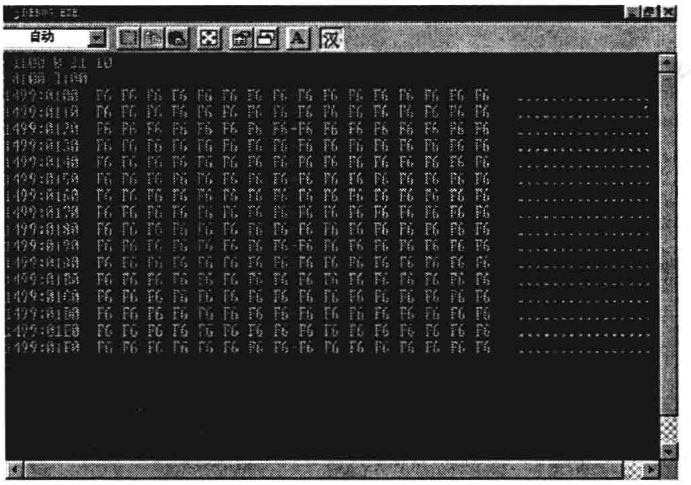


图 1-1-8

从图1-1-8中可发现其内容全为F6。

（3）依次在提示符“-”后键入命令：

D200 L100↙
D300 L100↙
D400 L100↙
：
：
D2000 L100↙

可以查看读入内容中的数据区的剩余内容，发现全为F6，表明该磁盘为新盘。

说明：对于使用过的旧盘，即使进行了格式化或删除磁盘上的所有文件的操作，该磁盘的数据区可能不全为0，也不全为F6。