

带您深入了解硬盘分区表与逻辑锁

PCPOP.COM 2005 年 01 月 19 日 类型:转载 作者:家缘 编辑:常凤臣

相信听说过硬盘 MBR、硬盘分区表、DBR 的朋友一定都不少。可是，你清楚它们分别起什么作用吗？它们的具体位置又在哪里呢？硬盘上的 MBR 只有一份吗？什么是硬盘逻辑锁？如何制造和破解它呢？？本文转载自家缘网，文中内容不代表本站观点，仅供参考。

一、必备基础知识：

● 有关扇区编号的基本知识：

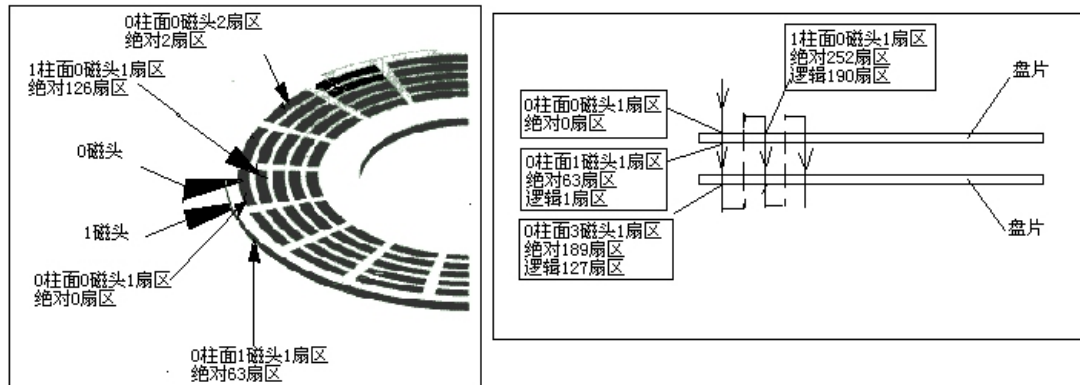
介绍一下有关硬盘扇区编号规则的 3 个易混淆的术语“物理扇区编号”、“绝对扇区编号”和“逻辑扇区编号”。

我们都知道硬盘扇区的定位有两种办法：

1. 直接按柱面、磁头、扇区 3 者的组合来定位（按这种编号方式得到的扇区编号称为物理扇区编号）；

2. 按扇区编号来定位（又分“绝对扇区编号”和“逻辑扇区编号”两种）。

这两种定位办法的换算关系如下图：（设图中所示硬盘每道扇区数均为 63）



BlockNum 是沿着 扇区->磁头->柱面 的顺序记数的,故有关系：

$$\text{BlockNum} = (\text{cylinder} * \text{NumberOfHeads} + \text{head}) * \text{SectorsPerTrack} + \text{sector} - 1;$$

如图所示，由于目前大多数硬盘采用的是一种“垂直分区结构”，故左图一磁头数为 2、盘片数为 1 的硬盘，图中 0 磁头所对扇区的表示方法就有 2 种，即：0 柱面 0 磁头 1 扇区=绝对 0 扇区，而 1 磁头所对扇区的表示方法也有 2 种，即：1 柱面 0 磁头 1 扇区=绝对 63 扇区。如果是如右图所示磁头数为 4、盘片数为 2 的硬盘，那么则顺着垂直于盘片的箭头线方向进行如图的绝对扇区的编号。

上面，我们说了物理扇区、绝对扇区的编号方式，而逻辑扇区编号由于是操作系统采用的扇区编号方式，而操作系统只能读取分区内部的数据内容，故逻辑扇区是从各分区内的第一个扇区开始编号，如我们下文对 MBR 的说明可以知道：MBR 这个扇区所在硬盘磁道是不属于分区范围内的，紧接着它后面的才是分区的内容,因此一般来说绝对 63 扇区= C:分区逻辑 1 扇区。

好，让我们列个表总结一下 3 种编号方式的不同：

编号方式	表示方法	采用该种方式编号的对象	起始编号
物理扇区编号	0 柱面 0 磁头 1 扇区	BIOS 内置中断服务程序	0 柱面 0 磁头 1 扇区
绝对扇区编号	绝对 X 扇区	人们为方便所采用的办法	绝对 0 扇区
逻辑扇区编号	逻辑 X 扇区	操作系统	逻辑 1 扇区

需要说明的是：本文假设所使用的硬盘每道扇区数都为 63。各位手头上所使用的硬盘具体的每道扇区数则可以在 BIOS 设置内有关硬盘参数的设置内查到。

● 有关 MBR、分区表、DBR 的基本知识:

☆ 硬盘 MBR(硬盘主引导记录)及硬盘分区表介绍

硬盘 MBR 就是我们经常说的“硬盘主引导记录”，简单地说，它是由 FDISK 等磁盘分区命令写在硬盘

绝对 0 扇区的一段数据，它由主引导程序、硬盘分区表及扇区结束标志字（55AA）这 3 个部分组成，如下表：

组成部分↕	所占字节数↕	内容、功能详述↕
主引导程序区↕	446↕	负责检查硬盘分区表、寻找可引导分区并负责将可引导分区的引导扇区（DBR）装入内存；↕
硬盘分区表区↕	16X4=64↕	每份 16 字节的 4 份硬盘分区表,里面记载了每个分区的类型、大小和分区开始、结束的位置等重要内容↕
结束标志字区↕	2↕	内容总为“55AA”↕

这 3 部分的大小加起来正好是 512 字节=1 个扇区（硬盘每扇区固定为 512 个字节），因此，人们又形象地把 MBR 称为“硬盘主引导扇区”。

这个扇区所在硬盘磁道上的其它扇区一般均空出，且这个扇区所在硬盘磁道是不属于分区范围内的，紧接着它后面的才是分区的内容（也就是说假如该盘每磁道扇区数为 63，那么从绝对 63 扇区开始才是分区的内容）。

☆ 硬盘 DBR(硬盘分区引导记录)介绍

DBR 是各个分区自己的引导记录,又称“分区引导记录”,它是由 FORMAT 高级格式化命令写在各个分区开始处第一个扇区(比如说:主分区 C:从 1 磁头 0 柱面 1 扇区=逻辑 1 扇区=绝对 63 扇区)开始,那么 C:区逻辑 1 扇区就是 DBR 所存放的位置)的一段数据.这段数据主要由以下几个部分组成:

- 1.占 3 个字节的跳转指令；
- 2.占 8 个字节的操作系统厂商标识及版本号；
- 3.占 19 个字节的分区参数表(又称 BPB),里面存放着对该分区进行读写操作时所必备的参数(如该分区内每扇区所包含的字节数、每簇扇区数、每个磁道的扇区数、该分区 FAT 份数等)；
- 4.占 480 个字节的 DOS 引导代码,它负责把 DOS 引导文件 IO.SYS、MSDOS.SYS 装入内存；
- 5.占 2 个字节的结束标志字“55AA”。

以上 5 个部分也正好占 1 个扇区；和 MBR 有所不同的是：DBR 扇区后面一般就紧接着存放该分区的 FAT（文件分配表，共 2 份）。

综上所述，我们知道硬盘 MBR 负责总管硬盘分区，只有分区工具才能对它进行读写（如 FDISK）；而 DBR 则负责管理某个具体的分区，它是用操作系统的高级格式化命令（如 FORMAT）来写入硬盘的。在系统启动时，最先读取的硬盘信息是 MBR，然后由 MBR 内的主引导程序读出 DBR，最后才由 DBR 内的 DOS 引导代码读取操作系统的引导程序，其中任何一个环节出了问题，操作系统都无法正常启动成功，如果是 MBR 部分出了问题,即使只是“55AA”标志字丢失或被改为其他值，通常都会出现“无效分区表”、“逻辑盘丢失、启动死机等现象；而如果是 DBR 部分出了问题，通常会出现“未格式化的分区”的错误提示。

☆ 基本知识的延伸

实际上，在每一个分区的前面，都有一份 MBR,在每一个分区的开始处，都有一份 DBR。通常我们把存放在绝对 0 扇区的那份 MBR 称为主 MBR 或 C 分区 MBR。这样我们就能画出

如下的 MBR、DBR 的存放位置表：市面上很多分区表保存软件（如 KV3000 的分区表保存功能）实际上保存的就是表中各个分区前 MBR 区的数据。

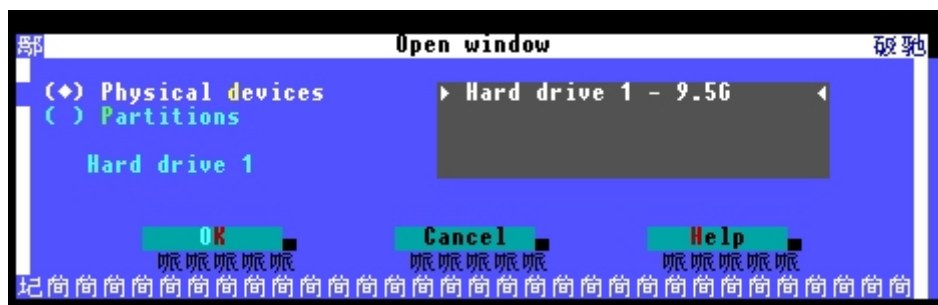
所属范围		存放位置	内容	大小
主 MBR 区		绝对 0 扇区	MBR	1 扇区
空白区		绝对 1-62 扇区	空白	62 扇区
C:	C:盘 DBR 区	绝对 63 扇区	DBR	1 扇区
分	FAT\FDT 区	绝对 64 扇区开始	FAT\FDT 表	不定
区	数据区		数据文件	不定
D 分区 MBR 区		绝对 A 扇区 (A=C:分区扇区数+63)	MBR	1 扇区
空白区		绝对 B-B+61 扇区 (B=A+1)	空白	62 扇区
D:	D:盘 DBR 区	绝对 C 扇区 (C=A+63)	DBR	1 扇区
分	FAT\FDT 区	绝对 D 扇区开始 (D=C+1)	FAT\FDT 表	不定
区	数据区		数据文件	不定

● 进距离观察 MBR、DBR:

口说无凭，眼见为实。还是让我们用工具来具体观察一下吧：）我们要观察的盘全部分作DOS区，在DOS区内共分C:、D:、E: 三个盘。

1、观察主 MBR:

首先得准备工具,这里我们推荐 DISKEDIT 兼容 FAT32 的版本,可以到 www.download.com.cn 搜索 DISKEDIT 并下载。该程序启动后界面如下:



选“PHYSICAL DEVICE”按 OK 进入。这时候，DISKEDIT 首先显示的就是硬盘绝对 0 扇区的 MBR 信息，如下图：

```

Disk editor (drive 1, sectors 0 - 200336)
Window Edit View Search Help
Absolute sector 0 (cylinder 0, head 0, sector 1)
00000000 3F C0 8E D0 00 00 7C F8 50 07 50 1F FC BE 0E 70 7C
00000010 00 1B 06 5C B7 B9 E5 01 F3 A4 04 0B BE BE 0E 1B 04
00000020 00 2C 7C 06 09 75 15 83 C6 28 E2 C2 E2 18 14 8E
00000030 00 38 2C 7C 10 49 74 16 38 C6 28 E2 C2 E2 18 14 8E
00000040 00 EE 83 C6 10 49 74 16 38 C6 28 E2 C2 E2 18 14 8E
00000050 00 3C 00 00 74 FA BB 07 00 B4 0E E2 C2 E2 18 14 8E
00000060 00 96 8A C4 76 04 B4 06 3C 0E 74 11 B4 0E B8 10 74
00000070 00 3A C4 75 22 B8 40 C6 46 25 06 24 24 0B BB 10 74
00000080 00 41 CD 13 58 72 16 81 FB 55 AA 5A 0E 0E 0E 0E 0E
00000090 00 0B 8A 00 88 56 24 C7 C3 A1 06 0E 0E 0E 0E 0E 0E
000000A0 00 0A 00 B8 01 02 8B D2 D9 85 F6 75 81 3E 2F 6F 0E
000000B0 00 25 03 4A 02 8C CD 13 72 2A B8 59 07 07 81 3E 2F
000000C0 00 0A 00 B8 01 02 8B D2 D9 85 F6 75 81 3E 2F 6F 0E
000000D0 00 8A 98 91 52 99 33 03 46 08 1B 56 0A E8 12 0E
000000E0 00 D5 4F 74 E4 33 C0 CD 13 E8 B8 00 00 00 00 00 00
000000F0 00 56 33 F8 56 56 52 50 06 53 51 BE 1E 10 56 56
00000100 00 0A 40 75 01 42 80 C2 02 E2 F7 F8 5E C3 E3 64 74
00000110 00 D6 C7 F8 B1 ED CE DE D0 A1 A3 B0 B2 D7 B7 B3 D6
00000120 00 CC D0 F2 CE DE D7 F7 F8 B3 CC D0 B3 C3 C9 C9 C9
00000130 00 D4 D8 B2 D9 D7 F7 F8 B3 CC D0 B3 C3 C9 C9 C9
00000140 00 B4 ED FC F3 D1 A3 B0 B2 D7 B7 B3 C3 C9 C9 C9
00000150 00 B7 ED BC CC D0 A1 A3 B0 B2 D7 B7 B3 C3 C9 C9
00000160 00 F7 CF B5 CD B3 00 00 00 00 00 00 00 00 00 00 00
00000170 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00000180 00 00 00 00 8B FC 1E 57 8B F5 CB 00 00 00 00 00 00
00000190 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000001A0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000001B0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000001C0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000001D0 00 01 00 0B FE 7F 1F 3F 00 00 00 00 00 00 00 00
000001E0 00 41 19 0F FE 7F 1F 3F 00 00 00 00 00 00 00 00
000001F0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00000200 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

```

从图中我们可以看到 MBR 的 0~01BD 字节为主引导程序;01BE~01FD 这 64 字节为硬盘分区表信息, 每项分区表占 16 字节; 最后是结束标志字 55AA。下面我们详细分析一下分区表各个字节所表示的意思:

所属分区	字节位置	内容	代表意义	具体意义
主分区	01BE	80	是否可引导	该分区为可引导分区, 若为 00 则为不可引导分区。
	01BF—01C1 ^{注1}	01 01 00	本分区 MBR 或 DBR 所在的磁头号、柱面号、扇区号位置。	本分区 DBR 位于 1 磁头 0 柱面 1 扇区
	01C2	0B	分区类型符	分区类型符: 为主分区, 且分区类型为 FAT32
	01C3—01C5	FE 7F 18	分区结束磁头号、柱面号、扇区号 ^{注1}	本分区结束于 0B 磁头、37F 柱面、3F 扇区
	01C6—01C9	3F 00 00 00	本分区前面预留的扇区数目	在分区前扇区数为: 3F
	01CA—01CD	9A E1 44 00	本分区总扇区数	本分区总扇区数为: 44E19A
扩展分区	01CE	00	同主分区	该分区为不可引导分区
	01CF—01D1	00 41 19		本分区 MBR 位于 0 磁头 119h 柱面 1 扇区
	01D2	0F		本分区为扩展分区
	01D3—01D5	FE FF FF		本分区结束于 FE 磁头、3FF 柱面、3F 扇区
	01D6—01D9	D9 E1 44 00		在分区前扇区数为: 44E1D9
	01DA—01DD	46 CC EC 00		本分区总扇区数为: ECCC46

注 1: 此处第一个字节存放磁头号(01、FE), 第二字节低六位存放扇区号 01=00(00 0001) =00(01)、7F=01(11 1111) =11(3F), 第二字节高 2 位+第三字节为柱面号 00(00) =000、01(18) =118。当需要表达的容量位置超过最大可以表达的数目时, 此位置放置一个固定的数字(偏移 0x01-0x03 处起始 CHS 三字节固定为 0x01C1FF, 偏移 0x05-0x07 处结束 CHS 三字节固定为 0xFEFFFF)而没有作用了。此时分区起始位置以及终结位置靠相互关系利用加法算出(利用分区前扇区数和本分区总扇区数)。

还需要说明的是 01BF—01C1 这 3 个字节在分区表里面, 视后面 01C2 字节所示分区类型的不同而代表不同的含义。如果 01C2 是代表主分区的 01、04、06 或 0B, 那么 01BF—01C1 所表示的就是该分区 DBR 所在的位置, 如果 01C2 是代表扩展分区的 05 或 0F, 那么 01BF—01C1 所表示的就是该分区 MBR 所在的位置。

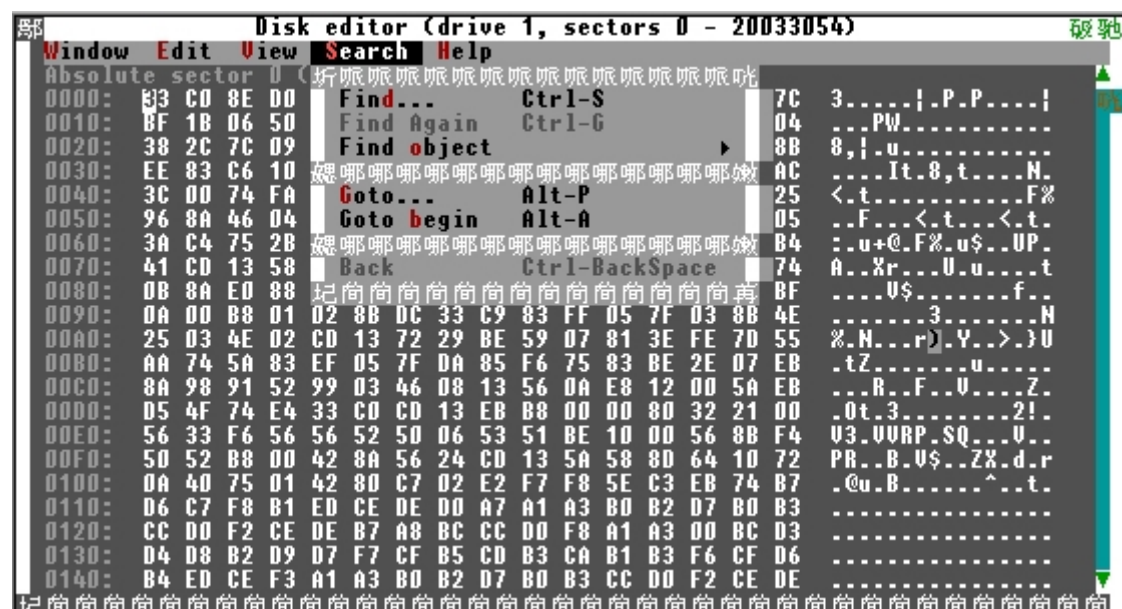
看完上面我们给出的第一份 MBR, 不知道大家注意到没有, 在硬盘分区表数据区里面只有两个分区表项, 而我们的盘明明是分成 C:、D:、E: 三个盘的, 怎么回事呢? 再仔细观察一下分区表, 我们还会发现在这份 MBR 里, 第一个分区表项(01BE—01CD)描述的是 C: 盘的信息, 第二个分区表项(01CE—01DD)描述的是整个扩展分区的信息, 它把剩下的 D:、E: 全部包含在里面了! 未免太笼统了点吧。那么, 如果我们要查看 D:、E: 各自详细的分区情况怎么办呢? 参考上一段我们给出的 01BF—01C1 这 3 个字节所代表的不同含义, 我们发现, 第二个分区表项的分区类型符为 0F, 也就是说是扩展分区, 那么分区类型符前面 3 个字节所代表的应该是扩展分区 MBR 所在的位置。可见不光是全盘最前面的 0 磁头 0 柱面 1

扇区有一份主 MBR，扩展分区的最前面也有一份 MBR！

还是让我们顺着系统启动的顺序先到 1 磁头 0 柱面 1 扇区去看看主分区 DBR，然后再看扩展分区的 MBR 是什么样的吧！

2、观察 C 分区 DBR:

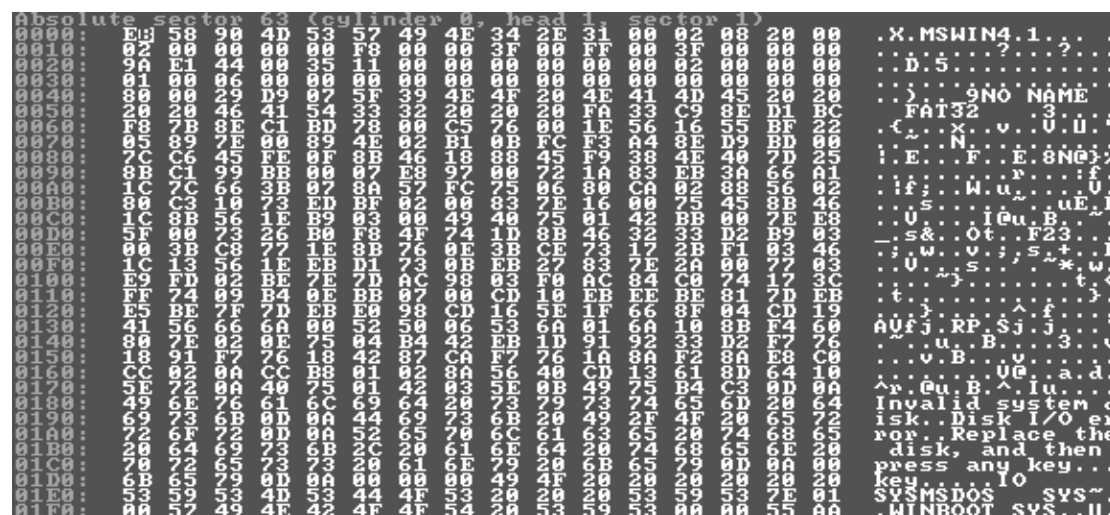
在 DISKEDIT 界面输入“ALT+S”，在出现的菜单内选“GOTO...”，如下图：



接下来，按表 1 内 01BF—01C1 字节所示的主分区 DBR 位置，在出现的菜单内依次输入 DBR 所在的柱面数（CYLINDER）=0，磁头数（HEAD）=1，扇区数（SECTOR）=1，如下图：

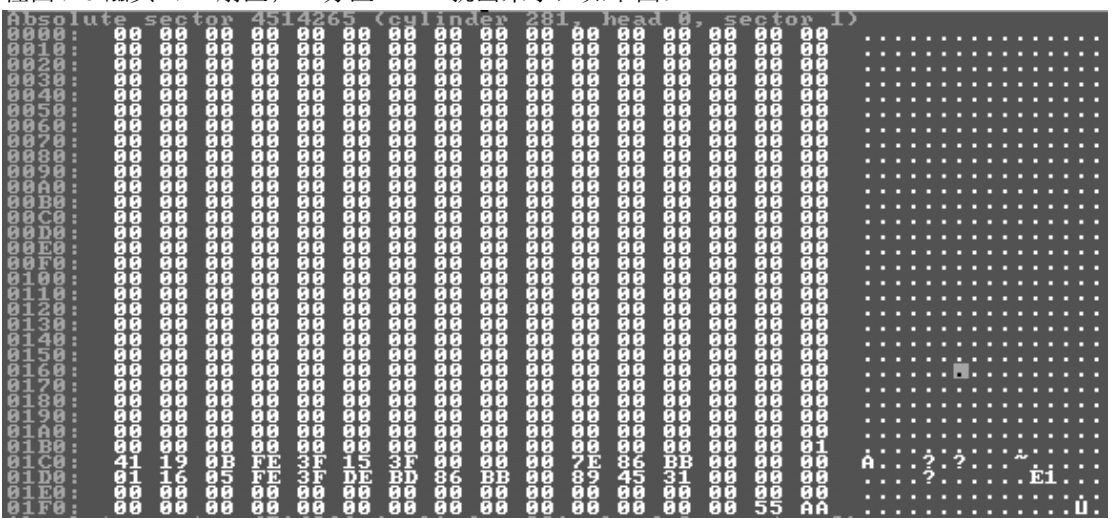


哈哈，第一份 DBR 的数据就调出来啦，如下图：



3、观察 D 分区 MBR:

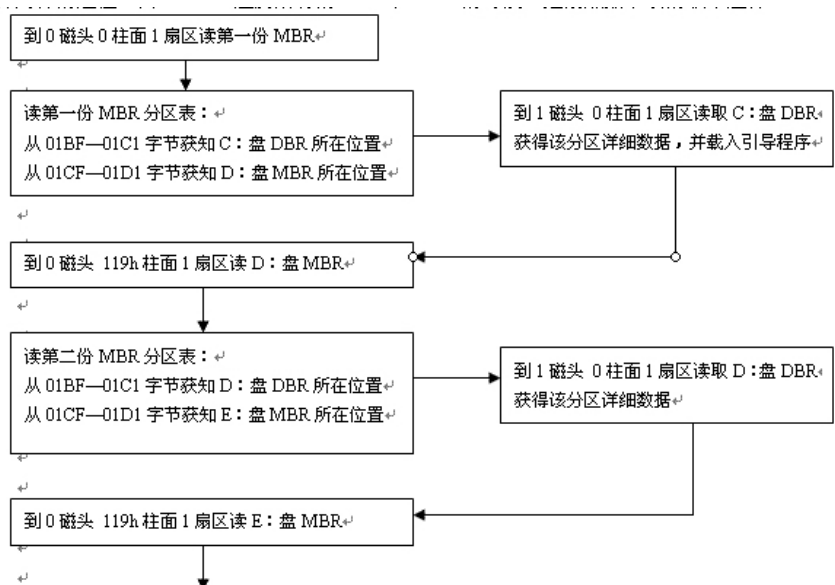
按照表 1 扩展分区 01CF—01D1 字节所示分区 MBR 位置,进入图 4 的菜单内输入 281(119h)柱面、0 磁头、1 扇区; D 分区 MBR 就出来了, 如下图:



哈哈,正如大家所看到的,这第二份 MBR 其实只是一个分区表而已。同第一份 MBR 一样,也只是描述了当前主盘(D: 盘)和剩余空间的分区状况。按照和第 3 步中同样的办法,我们同样能够定位出 D: 盘的 DBR 和 E 分区 MBR(最后一份 MBR)所在的位置并用 DISKEDIT 进行观察。

4. 小结

通过观察,我们证实了每一个硬盘分区,都有各自的 MBR 和 DBR; 操作系统启动时,不论是从硬盘还是从软盘启动,都需要先由 BIOS 读绝对 0 扇区的主 MBR,找到标志为 80 的可引导分区,然后由 MBR 负责读出该分区内的 DBR,再由 DBR 负责读出存在该分区的系统启动程序(IO.SYS 等),最后在 DOS 系统程序 IO.SYS 的指挥下遍历所有的 MBR 和 DBR,从而获知完整的硬盘分区结构。使用 FDISK 分区时也需要进行同样的过程。而 IO.SYS 遍历所有的 MBR 和 DBR 的时候,是按照如下表的顺序进行:

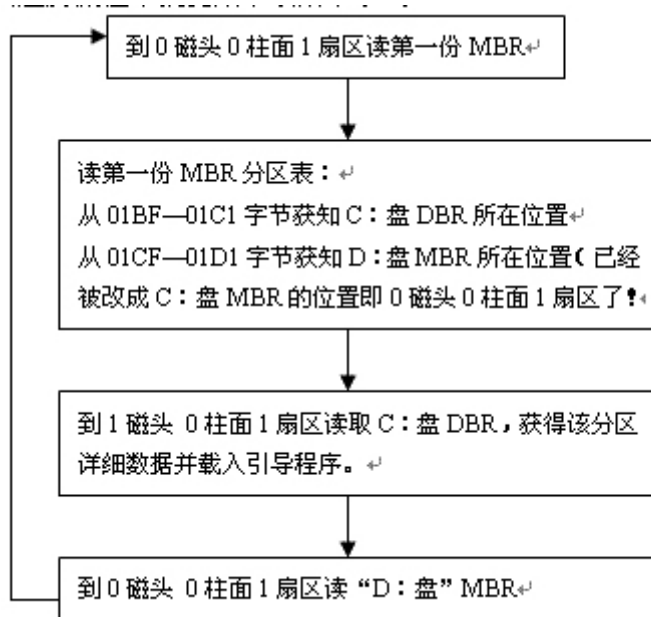


三、分区表知识实际应用

● 我是无聊的小锁匠。。。

看了这么多图表文字,读者老爷们是不是烦了? 那么我们来折磨一下自己的硬盘,制作一个硬盘逻辑锁玩玩如何。

观察上面我们给出的遍历流程表，参考表 2 中关于 01C2 字节的说明，如果我们在描述 D 分区 MBR 位置的 01CF—01D1 字节处作点手脚，把这里本来存放 D：盘 MBR 位置的字节改成 00 01 00（就是 C 分区 MBR 所在的位置），那么遍历流程不就变成下图所示了吗：



哈哈，表面上看，应该这样就可以做成一个逻辑锁了吧。可是实际情况一定会让你失望，98 根本一点反应也没有，居然给我一切正常！！

哎，再试试动动其它歪脑筋好了。咦。。。总觉得 01D2 字节表示 D：盘分区类型的 0F 怪怪的，把它改成 05 看看！！哈哈，这下硬盘总算在出现 98 欢迎画面的时候陷上上表所示的死循环了：）好，现在我们换软盘启动看看。。。咦。。。怎么出现一大堆 E 文后居然还能引导成功！

天哪！这什么破游戏，一点也不好玩!!! 走走，搓雷神去了，不玩了！嗯。。。如果把表示 C：盘分区类型的 01C2 字节的 0B 改成其它数会怎样？？把它改成 0A 吧！

嘿嘿嘿，终于大功告成了，无论是软盘或者是硬盘启动，统统给我死翘翘了：）下面就给出一种完美的硬盘逻辑锁主 MBR 样本：（图中黄色字体的为改动的部分，只需要改区区 4 个字节哦！）

Disk editor (drive 1, sectors 0 - 20033054)																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																		
Window		Edit	View	Search	Help																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																													
Absolute	sector	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	100	101	102	103	104	105	106	107	108	109	110	111	112	113	114	115	116	117	118	119	120	121	122	123	124	125	126	127	128	129	130	131	132	133	134	135	136	137	138	139	140	141	142	143	144	145	146	147	148	149	150	151	152	153	154	155	156	157	158	159	160	161	162	163	164	165	166	167	168	169	170	171	172	173	174	175	176	177	178	179	180	181	182	183	184	185	186	187	188	189	190	191	192	193	194	195	196	197	198	199	200	201	202	203	204	205	206	207	208	209	210	211	212	213	214	215	216	217	218	219	220	221	222	223	224	225	226	227	228	229	230	231	232	233	234	235	236	237	238	239	240	241	242	243	244	245	246	247	248	249	250	251	252	253	254	255	256	257	258	259	260	261	262	263	264	265	266	267	268	269	270	271	272	273	274	275	276	277	278	279	280	281	282	283	284	285	286	287	288	289	290	291	292	293	294	295	296	297	298	299	300	301	302	303	304	305	306	307	308	309	310	311	312	313	314	315	316	317	318	319	320	321	322	323	324	325	326	327	328	329	330	331	332	333	334	335	336	337	338	339	340	341	342	343	344	345	346	347	348	349	350	351	352	353	354	355	356	357	358	359	360	361	362	363	364	365	366	367	368	369	370	371	372	373	374	375	376	377	378	379	380	381	382	383	384	385	386	387	388	389	390	391	392	393	394	395	396	397	398	399	400	401	402	403	404	405	406	407	408	409	410	411	412	413	414	415	416	417	418	419	420	421	422	423	424	425	426	427	428	429	430	431	432	433	434	435	436	437	438	439	440	441	442	443	444	445	446	447	448	449	450	451	452	453	454	455	456	457	458	459	460	461	462	463	464	465	466	467	468	469	470	471	472	473	474	475	476	477	478	479	480	481	482	483	484	485	486	487	488	489	490	491	492	493	494	495	496	497	498	499	500	501	502	503	504	505	506	507	508	509	510	511	512	513	514	515	516	517	518	519	520	521	522	523	524	525	526	527	528	529	530	531	532	533	534	535	536	537	538	539	540	541	542	543	544	545	546	547	548	549	550	551	552	553	554	555	556	557	558	559	560	561	562	563	564	565	566	567	568	569	570	571	572	573	574	575	576	577	578	579	580	581	582	583	584	585	586	587	588	589	590	591	592	593	594	595	596	597	598	599	600	601	602	603	604	605	606	607	608	609	610	611	612	613	614	615	616	617	618	619	620	621	622	623	624	625	626	627	628	629	630	631	632	633	634	635	636	637	638	639	640	641	642	643	644	645	646	647	648	649	650	651	652	653	654	655	656	657	658	659	660	661	662	663	664	665	666	667	668	669	670	671	672	673	674	675	676	677	678	679	680	681	682	683	684	685	686	687	688	689	690	691	692	693	694	695	696	697	698	699	700	701	702	703	704	705	706	707	708	709	710	711	712	713	714	715	716	717	718	719	720	721	722	723	724	725	726	727	728	729	730	731	732	733	734	735	736	737	738	739	740	741	742	743	744	745	746	747	748	749	750	751	752	753	754	755	756	757	758	759	760	761	762	763	764	765	766	767	768	769	770	771	772	773	774	775	776	777	778	779	780	781	782	783	784	785	786	787	788	789	790	791	792	793	794	795	796	797	798	799	800	801	802	803	804	805	806	807	808	809	810	811	812	813	814	815	816	817	818	819	820	821	822	823	824	825	826	827	828	829	830	831	832	833	834	835	836	837	838	839	840	841	842	843	844	845	846	847	848	849	850	851	852	853	854	855	856	857	858	859	860	861	862	863	864	865	866	867	868	869	870	871	872	873	874	875	876	877	878	879	880	881	882	883	884	885	886	887	888	889	890	891	892	893	894	895	896	897	898	899	900	901	902	903	904	905	906	907	908	909	910	911	912	913	914	915	916	917	918	919	920	921	922	923	924	925	926	927	928	929	930	931	932	933	934	935	936	937	938	939	940	941	942	943	944	945	946	947	948	949	950	951	952	953	954	955	956	957	958	959	960	961	962	963	964	965	966	967	968	969	970	971	972	973	974	975	976	977	978	979	980	981	982	983	984	985	986	987	988	989	990	991	992	993	994	995	996	997	998	999	1000	1001	1002	1003	1004	1005	1006	1007	1008	1009	1010	1011	1012	1013	1014	1015	1016	1017	1018	1019	1020	1021	1022	1023	1024	1025	1026	1027	1028	1029	1030	1031	1032	1033	1034	1035	1036	1037	1038	1039	1040	1041	1042	1043	1044	1045	1046	1047	1048	1049	1050	1051	1052	1053	1054	1055	1056	1057	1058	1059	1060	1061	1062	1063	1064	1065	1066	1067	1068	1069	1070	1071	1072	1073	1074	1075	1076	1077	1078	1079	1080	1081	1082	1083	1084	1085	1086	1087	1088	1089	1090	1091	1092	1093	1094	1095	1096	1097	1098	1099	1100	1101	1102	1103	1104	1105	1106	1107	1108	1109	1110	1111	1112	1113	1114	1115	1116	1117	1118	1119	1120	1121	1122	1123	1124	1125	1126	1127	1128	1129	1130	1131	1132	1133	1134	1135	1136	1137	1138	1139	1140	1141	1142	1143	1144	1145	1146	1147	1148	1149	1150	1151	1152	1153	1154	1155	1156	1157	1158	1159	1160	1161	1162	1163	1164	1165	1166	1167	1168	1169	1170	1171	1172	1173	1174	1175	1176	1177	1178	1179	1180	1181	1182	1183	1184	1185	1186	1187	1188	1189	1190	1191	1192	1193	1194	1195	1196	1197	1198	1199	1200	1201	1202	1203	1204	1205	1206	1207	1208	1209	1210	1211	1212	1213	1214	1215	1216	1217	1218	1219	1220	1221	1222	1223	1224	1225	1226	1227	1228	1229	1230	1231	1232	1233	1234	1235	1236	1237	1238	1239	1240	1241	1242	1243	1244	1245	1246	1247	1248	1249	1250	1251	1252	1253	1254	1255	1256	1257	1258	1259	1260	1261	1262	1263	1264	1265	1266	1267	1268	1269	1270	1271	1272	1273	1274	1275	1276	1277	1278	1279	1280	1281	1282	1283	1284	1285	1286	1287	1288	1289	1290	1291	1292	1293	1294	1295	1296	1297	1298	1299	1300	1301	1302	1303	1304	1305	1306	1307	1308	1309	1310	1311	1312	1313	1314	1315	1316	1317	1318	1319	1320	1321	1322	1323	1324	1325	1326	1327	1328	1329	1330	1331	1332	1333	1334	1335	1336	1337	1338	1339	1340	1341	1342	1343	1344	1345	1346	1347	1348	1349	1350	1351	1352	1353	1354	1355	1356	1357	1358	1359	1360	1361	1362	1363	1364	1365	1366	1367	1368	1369	1370	1371	1372	1373	1374	1375	1376	1377	1378	1379	1380	1381	1382	1383	1384	1385	1386	1387	1388	1389	1390	1391	1392	1393	1394	1395	1396	1397	1398	1399	1400	1401	1402	1403	1404	1405	1406	1407	1408	1409	1410	1411	1412	1413	1414	1415	1416	1417	1418	1419	1420	1421	1422	1423	1424	1425	1426	1427	1428	1429	1430	1431	1432	1433	1434	1435	1436	1437	1438	1439	1440	1441	1442	1443	1444	1445	1446	1447	1448	1449	1450	1451	1452	1453	1454	1455	1456	1457	1458	1459	1460	1461	1462	1463	1464	1465	1466	1467	1468	1469	1470	1471	1472	1473	1474	1475	1476	1477	1478	1479	1480	1481	1482	1483	1484	1485	1486	1487	1488

好了，我们来总结一下制作逻辑锁的必要步骤吧！

1. 先把 C:盘 MBR 的 01CF—01D1 字节处表示的 D:盘 MBR 位置改成 00 01 00;
2. 然后把紧接着的表示 D:盘分区类型的 01D2 字节改成 05;
3. 最后改 C:盘 MBR 中表示 C:盘分区类型的 01C2 字节,把它改成 0A、04、03、02 或者 00 这 5 种字节都可以。

这样一来,不论是从软盘或者硬盘用 DOS7.0 版本启动,甚至加挂正常的可引导硬盘,都难逃死机的恶运了?至于为什么还要改两个盘的分区类型,这个本鸟也还没有完全想通。不过有一点可以肯定的是:即使不改分区类型,在使用 FDISK 时也是必死无疑的。

哎呀,差点忘了说了,中了锁怎么解呢??

别急,前面我们不是说到:”不论是从硬盘还是从软盘启动,都需要在 DOS 系统程序 IO.SYS 的指挥下遍历所有的 MBR 和 DBR”吗,既然如此不如在 IO.SYS 上动动脑筋好了。

● 破解硬盘逻辑锁的三种办法:

1 原来,IO.SYS 在指挥系统遍历所有的 MBR 和 DBR 的时候,首先要检查 MBR 或者 DBR 扇区结尾的结束标志字是否是”55AA”.如果不是,那么将退出遍历并报分区表错.幸运的是报错之前我们运行 DISKEDIT 改回原来的 MBR 所需要的系统核心部分已经载入了内存,否则….

知道了上面的原理,我们不妨反其道而行之,编辑 IO.SYS,让它在指挥系统遍历所有的 MBR 和 DBR 时检查 MBR 或者 DBR 扇区结尾的结束标志字是否是除了”55AA”外的其它值.这样一来,就能跳出死循环而执行 DISKEDIT 了.具体步骤如下:

A 先制作一张启动软盘,并用 ATTRIB -R -H -S A: \IO.SYS 命令去掉 IO.SYS 的系统、只读及隐藏属性;

B 用 ULTRAEDIT 之类的编辑工具搜索刚做好的启动软盘上的 IO.SYS,找到数个”55AA”,把它们改成其它数值;

C. 保存 IO.SYS,并用 ATTRIB +R +H +S A:\IO.SYS 命令改回文件的属性.不怕逻辑锁的启动盘就做好啦!启动后不要理会出现的错误信息,赶快运行 DISKEDIT 改回原来的 MBR 吧!

2 如果你嫌麻烦,也可以用硬盘厂家提供的硬盘修复工具制作修复软盘,这种修复软盘由于使用了不同的启动办法,所以一般都不怕逻辑锁

好了，大家都了解了吧！下次将带给您更精彩的！