FAT文件系统原理(一)

FAT文件系统原理(一)

更新时间:2004-4-20 【字体：小 大】

一、硬盘的物理结构：

图1

硬盘存储数据是根据电、磁转换原理实现的。硬盘由一个或几个表面镀有磁性物质的金属或玻璃等物质盘片以及盘片两面所安装的磁头和相应的控制电路组成(图1)，其中盘片和磁头密封在无尘的金属壳中。
硬盘工作时，盘片以设计转速高速旋转，设置在盘片表面的磁头则在电路控制下径向移动到指定位置然后将数据存储或读取出来。当系统向硬盘写入数据时，磁头中“写数据”电流产生磁场使盘片表面磁性物质状态发生改变，并在写电流磁场消失后仍能保持，这样数据就存储下来了；当系统从硬盘中读数据时，磁头经过盘片指定区域，盘片表面磁场使磁

头产生感应电流或线圈阻抗产生变化，经相关电路处理后还原成数据。因此只要能将盘片表面处理得更平滑、磁头设计得更精密以及尽量提高盘片旋转速度，就能造出容量更大、读写数据速度更快的硬盘。这是因为盘片表面处理越平、转速越快就能越使磁头离盘片表面越近，提高读、写灵敏度和速度；磁头设计越小越精密就能使磁头在盘片上占用空间越小，使磁头在一张盘片上建立更多的磁道以存储更多的数据。

二、硬盘的逻辑结构。

硬盘由很多盘片(platter)组成，每个盘片的每个面都有一个读写磁头。如果有N个盘片。就有2N个面，对应2N个磁头(Heads)，从0、1、2开始编号。每个盘片被划分成若干个同心圆磁道(逻辑上的，是不可见的。)每个盘片的划分规则通常是一样的。这样每个盘片的半径均为固定值R的同心圆再逻辑上形成了一个以电机主轴为轴的柱面(Cylinders)，从外至里编号为0、1、2……每个盘片上的每个磁道又被划分为几十个扇区(Sector)，通常的容量是512byte，并按照一定规则编号为1、2、3……形成Cylinders×Heads×Sector个扇区。这三个参数即是硬盘的物理参数。我们下面的很多实践需要深刻理解这三个参数的意义。

三、磁盘引导原理。

3.1 MBR(master boot record)扇区：

计算机在按下power键以后，开始执行主板bios程序。进行完一系列检测和配置以后。开始按bios中设定的系统引导顺序引导系统。假定现在是硬盘。Bios执行完自己的程序后如何把执行权交给硬盘呢。交给硬盘后又执行存储在哪里的程序呢。其实，称为mbr的一段代码起着举足轻重的作用。MBR(master boot record),即主引导记录，有时也称主引导扇区。位于整个硬盘的0柱面0磁头1扇区(可以看作是硬盘的第一个扇区)，bios在执行自己固有的程序以后就会jump到mbr中的第一条指令。将系统的控制权交由mbr来执行。在总共512byte的主引导记录中，MBR的引导程序占了其中的前446个字节(偏移0H~偏移1BDH)，随后的64个字节(偏移1BEH~偏移1FDH)为DPT(Disk PartitionTable，硬盘分区表)，最后的两个字节“55 AA”(偏移1FEH~偏移1FFH)是分区有效结束标志。

MBR不随操作系统的不同而不同，意即不同的操作系统可能会存在相同的MBR，即使不同，MBR也不会夹带操作系统的性质。具有公共引导的特性。

我们来分析一段mbr。下面是用winhex查看的一块希捷120GB硬盘的mbr。

Offset	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	访问
0000000000	33	C0	8E	D0	BC	00	7C	FB	50	07	50	1F	FC	BE	1B	7C	3欹屑.|鼯.P. .|
0000000010	BF	1B	06	50	57	B9	E5	01	F3	A4	CB	BD	BE	07	B1	04	?.PW瑰.螭私??
0000000020	38	6E	00	7C	09	75	13	83	C5	10	E2	F4	CD	18	8B	F5	8n.|.u.鼯.怍?嫚
0000000030	83	C6	10	49	74	19	38	2C	74	F6	A0	B5	07	B4	07	8B	魑.It.8,t鳗??
0000000040	F0	AC	3C	00	74	FC	BB	07	00	B4	0E	CD	10	EB	F2	88	赕<.t ..??腧
0000000050	4E	10	E8	46	00	73	2A	FE	46	10	80	7E	04	0B	74	0B	N.鐵.s*樨.€~..t.
0000000060	80	7E	04	0C	74	05	A0	B6	07	75	D2	80	46	02	06	83	€~..t.牰.u襹F..
0000000070	46	08	06	83	56	0A	00	E8	21	00	73	05	A0	B6	07	EB	F..僔..?.s.牰. ë
0000000080	BC	81	3E	FE	7D	55	AA	74	0B	80	7E	10	00	74	C8	A0	紗>裯U溥.€~..t蒌
0000000090	B7	07	EB	A9	8B	FC	1E	57	8B	F5	CB	BF	05	00	8A	56	?氅嬂.W嫚丝..爨V
00000000A0	00	B4	08	CD	13	72	23	8A	C1	24	3F	98	8A	DE	8A	FC	.??r#旎$?槉迠 Šu
00000000B0	43	F7	E3	8B	D1	86	D6	B1	06	D2	EE	42	F7	E2	39	56	C縻嬔嗳?翌B麾9VV
00000000C0	0A	77	23	72	05	39	46	08	73	1C	B8	01	02	BB	00	7C	.w#r.9F.s.?.?|.|
00000000D0	8B	4E	02	8B	56	00	CD	13	73	51	4F	74	4E	32	E4	8A	姥.媧.?sQOtN2鋳Š
00000000E0	56	00	CD	13	EB	E4	8A	56	00	60	BB	AA	55	B4	41	CD	V.?脘爨.`华U硼 Í
00000000F0	13	72	36	81	FB	55	AA	75	30	F6	C1	01	74	2B	61	60	.r6傚U溟0隽.t+a`
0000000100	6A	00	6A	00	FF	76	0A	FF	76	08	6A	00	68	00	7C	6A	j.j. v. v.j.h.|
0000000110	01	6A	10	B4	42	8B	F4	CD	13	61	61	73	0E	4F	74	0B	.j.簪嬟?aas.Ot..
0000000120	32	E4	8A	56	00	CD	13	EB	D6	61	F9	C3	49	6E	76	61	2鋳V.?胫a Invaa
0000000130	6C	69	64	20	70	61	72	74	69	74	69	6F	6E	20	74	61	lid partition ta
0000000140	62	6C	65	00	45	72	72	6F	72	20	6C	6F	61	64	69	6E	ble.Error loadin
0000000150	67	20	6F	70	65	72	61	74	69	6E	67	20	73	79	73	74	g operating syst
0000000160	65	6D	00	4D	69	73	73	69	6E	67	20	6F	70	65	72	61	em.Missing opera
0000000170	74	69	6E	67	20	73	79	73	74	65	6D	00	00	00	00	00	ting system.....
0000000180	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
0000000190	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....MBR引导代码...
00000001A0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
00000001B0	00	00	00	00	00	2C	44	63	33	B1	33	B1	00	00	80	01	.....,Dc3??.€.
00000001C0	01	00	07	FE	FF	7B	3F	00	00	00	3D	A8	DA	00	00	00	.?{?..=Y..DPT硬盘分区表
00000001D0	C1	7C	0F	FE	FF	FF	7C	A8	DA	00	45	8F	1E	0D	00	00	翀.? |Y.E?...
00000001E0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	..分区有效标志..
00000001F0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	55	AA	..............Uª

图 2

你的硬盘的MBR引导代码可能并非这样。不过即使不同，所执行的功能大体是一样的。这是wowocock关于磁盘mbr的反编译，已加了详细的注释，感兴趣可以细细研究一下。

我们看DPT部分。操作系统为了便于用户对磁盘的管理。加入了磁盘分区的概念。即将一块磁盘逻辑划分为几块。磁盘分区数目的多少只受限于C～Z的英文字母的数目，在上图DPT共64个字节中如何表示多个分区的属性呢?microsoft通过链接的方法解决了这个问题。在DPT共64个字节中，以16个字节为分区表项单位描述一个分区的属性。也就是说，第一个分区表项描述一个分区的属性，一般为基本分区。第二个分区表项描述除基本分区外的其余空间，一般而言，就是我们所说的扩展分区。这部分的大体说明见表1。

表1　图2分区表第一字段

字节位移	字段长度	值	字段名和定义
0x01BE	BYTE	0x80	引导指示符(Boot Indicator)　指明该分区是否是活动分区。
0x01BF	BYTE	0x01	开始磁头(Starting Head)
0x01C0	6位	0x01	开始扇区(Starting Sector) 只用了0~5位。后面的两位(第6位和第7位)被开始柱面字段所使用
0x01C1	10位	0x00	开始柱面(Starting Cylinder)　除了开始扇区字段的最后两位外，还使用了1位来组成该柱面值。开始柱面是一个10位数，最大值为1023
0x01C2	BYTE	0x07	系统ID(System ID) 定义了分区的类型，详细定义，请参阅图4
0x01C3	BYTE	0xFE	结束磁头(Ending Head)
0x01C4	6位	0xFF	结束扇区(Ending Sector)　只使用了0~5位。最后两位(第6、7位)被结束柱面字段

			所使用
0x01C5	10位	0x7B	结束柱面(Ending Cylinder) 除了结束扇区字段最后的两位外，还使用了1位，以组成该柱面值。结束柱面是一个10位的数，最大值为1023
0x01C6	DWORD	0x0000003F	相对扇区数(Relative Sectors) 从该磁盘的开始到该分区的开始的位移量，以扇区来计算
0x01CA	DWORD	0x00DAA83D	总扇区数(Total Sectors) 该分区中的扇区总数

注：上表中的超过1字节的数据都以实际数据显示，就是按高位到地位的方式显示。存储时是按低位到高位存储的。两者表现不同，请仔细看清楚。以后出现的表，图均同。

也可以在winhex中看到这些参数的意义：

Master Boot Record, 基础偏移量：0

Offset 偏移	标题	数值
0	Master bootstrap loader code 引导代码	33 C0 8E D0 BC 00 7C FB
	Partition Table Entry #1	
1BE	80 = active partition 活动分区标志	80 80表示活动，00表示非
1BF	Start head 开始磁头	1
1C0	Start sector 开始扇区	1
1C0	Start cylinder 开始柱面	0
1C2	Operating system indicator (hex	07 分区类型标志
1C3	End head 结束磁头	254
1C4	End sector 结束扇区	63
1C4	End cylinder 结束柱面	891
1C6	Sectors preceding partition 1	63 本分区之前的扇区数
1CA	Length of partition 1 in sector	14329917 本分区的扇区总数
	Partition Table Entry #2	
1CE	80 = active partition	00
1CF	Start head	0
1D0	Start sector	1
1D0	Start cylinder	892
1D2	Operating system indicator (hex	0F
1D3	End head	254
1D4	End sector	63
1D4	End cylinder	1023
1D6	Sectors preceding partition 2	14329980
1DA	Length of partition 2 in sector	220106565
	Partition Table Entry #3	
1DE 0H	80 = active partition	00
1DF 1H	Start head	0
1E0 2H	Start sector	0
1E0 2H	Start cylinder	0

说明： 每个分区表项占用16个字节，假定偏移地址从0开始。如图3的分区表项3。分区表项4同分区表项3。

1、0H偏移为活动分区是否标志，只能选00H和80H。80H为活动，00H为非活动。其余值对microsoft而言为非法值。

2、重新说明一下(这个非常重要)：大于1个字节的数被以低字节在前的存储格式格式(little endian format)或称反字节顺序保存下来。低字节在前的格式是一种保存数的方法，这样，最低位的字节最先出现在十六进制数符号中。例如，相对扇区数字段的值0x3F000000的低字节在前表示为0x0000003F。这个低字节在前的格式数的十进制数为63。

3、系统在分区时，各分区都不允许跨柱面，即均以柱面为单位，这就是通常所说的分区粒度。有时候我们分区是输入分区的大小为7000M，分出来却是6997M，就是这个原因。 偏移2H和偏移6H的扇区和柱面参数中,扇区占6位(bit)，柱面占10位(bit)，以偏移6H为例，其低6位用作扇区数的二进制表示。其高两位做柱面数10位中的高两位，偏移7H组成的8位做柱面数10位中的低8位。由此可知，实际上用这种方式表示的分区容量是有限的，柱面和磁头从0开始编号,扇区从1开始编号,所以最多只能表示1024个柱面×63个扇区×256个磁头×512byte=8455716864byte。即通常的8.4GB(实际上应该是7.8GB左右)限制。实际上磁头数通常只用到255个(由汇编语言的寻址寄存器决定),即使把这3个字节按线性寻址，依然力不从心。 在后来的操作系统中，超过8.4GB的分区其实已经不通过C/H/S的方式寻址了。而是通过偏移CH～偏移FH共4个字节32位线性扇区地址来表示分区所占用的扇区总数。可知通过4个字节可以表示2^32个扇区，即2TB=2048GB，目前对于大多数计算机而言，这已经是个天文数字了。在未超过8.4GB的分区上，C/H/S的表示方法和线性扇区的表示方法所表示的分区大小是一致的。也就是说，两种表示方法是协调的。即使不协调，也以线性寻址为准。(可能在某些系统中会提示出错)。超过8.4GB的分区结束C/H/S一般填充为FEH FFH FFH。即C/H/S所能表示的最大值。有时候也会用柱面对1024的模来填充。不过这几个字节是什么其实都无关紧要了。

虽然现在的系统均采用线性寻址的方式来处理分区的大小。但不可跨柱面的原则依然没变。本分区的扇区总数加上与前一分区之间的保留扇区数目依然必须是柱面容量的整数倍。(保留扇区中的第一个扇区就是存放分区表的MBR或虚拟MBR的扇区，分区的扇区总数在线性表示方式上是不计入保留扇区的。如果是第一个分区，保留扇区是本分区前的所有扇区。

附：分区表类型标志如图4

分区类型标志：

00 空，mocrosoft不允许使用。	63 GNU HURD or Sys
01 FAT32	64 Novell Netware
02 XENIX root	65 Novell Netware
03 XENIX usr	70 Disk Secure Mult
04 FAT16 <32M	75 PC/IX
05 Extended	80 Old Minix
06 FAT16	81 Minix/Old Linux
07 HPFS/NTFS	82 Linux swap
08 AIX	83 Linux
09 AIX bootable	84 OS/2 hidden C:
0A OS/2 Boot Manage	85 Linux extended
0B Win95 FAT32	86 NTFS volume set
0C Win95 FAT32	87 NTFS volume set
0E Win95 FAT16	93 Amoeba
0F Win95 Extended(>8GB)	94 Amoeba BBT
10 OPUS	A0 IBM Thinkpad hidden
11 Hidden FAT12	A5 BSD/386
12 Compaq diagnost	A6 Open BSD
16 HiddenFAT16	A7 NextSTEP
14 Hidden FAT16<32GB	B7 BSDI fs
17 Hidden HPFS/NTFS	B8 BSDI swap
18 AST Windows swap	BE Solaris boot
1B Hidden FAT32	partition
1C Hidden FAT32 partition	C0 DR-DOS/Novell DOS
(using LBA-mode	secured partition
INT 13 extensions)	C1 DRDOS/sec
1E Hidden LBA VFAT partition	C4 DRDOS/sec
24 NEC DOS	C6 DRDOS/sec
3C Partition Magic	C7 Syrinx
40 Venix 80286	DB CP/M/CTOS
41 PPC PreP Boot	E1 DOS access
42 SFS	E3 DOS R/O
4D QNX4.x	E4 SpeedStor
4E QNX4.x 2nd part	EB BeOS fs
4F QNX4.x 3rd part	F1 SpeedStor
50 Ontrack DM	F2 DOS 3.3+ secondary
51 Ontrack DM6 Aux	partition
52 CP/M	F4 SpeedStor
53 oNtRACK DM6 Aux	FE LAN step
54 OnTrack DM6	FF BBT
55 EZ-Drive	
56 Golden Bow	
5C Priam Edisk	
61 Speed Stor	

图4

3.2 扩展分区：

扩展分区中的每个逻辑驱动器都存在一个类似于MBR的扩展引导记录(Extended Boot Record, EBR)，也有人称之为虚拟mbr或扩展mbr，意思是一样的。扩展引导记录包括一个扩展分区表和该扇区的标签。扩展引导记录将记录只包含扩展分区中每个逻辑驱动器的第一个柱面的第一面的信息。一个逻辑驱动器中的引导扇区一般位于相对扇区32或63。但是，如果磁盘上没有扩展分区，那么就不会有扩展引导记录和逻辑驱动器。第一个逻辑驱动器的扩展分区表中的第一项指向它自身的引导扇区。第二项指向下一个逻辑驱动器的EBR。如果不存在进一步的逻辑驱动器，第二项就不会使用，而且被记录成一系列零。如果有附加的逻辑驱动器，那么第二个逻辑驱动器的扩展分区表的第一项会指向它本身的引导扇区。第二个逻辑驱动器的扩展分区表的第二项指向下一个逻辑驱动器的EBR。扩展分区表的第三项和第四项永远都不会被使用。

通过一幅4分区的磁盘结构图可以看到磁盘的大致组织形式。如图5：

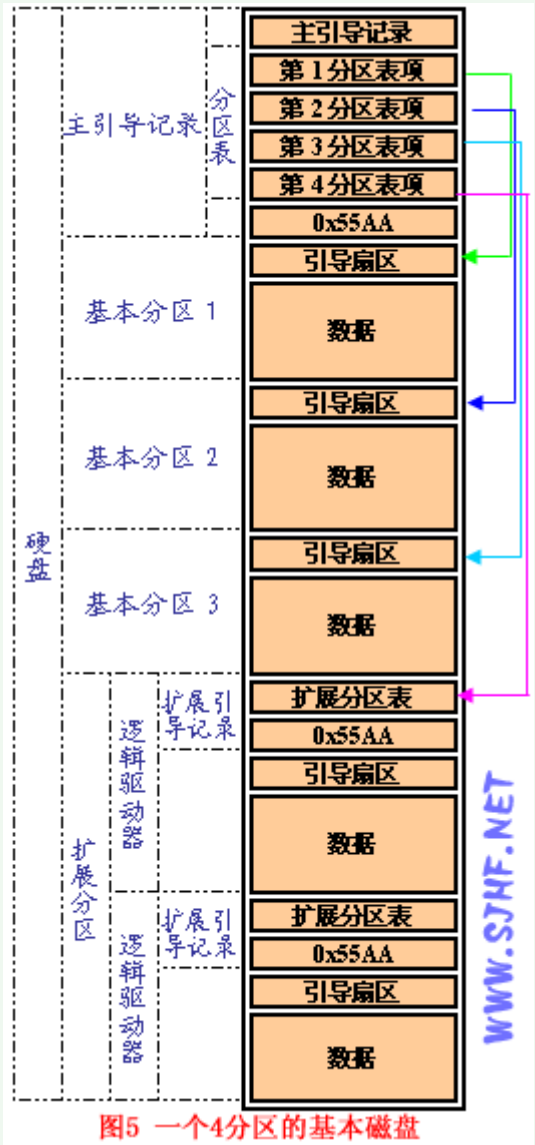

图5 一个4分区的基本磁盘

关于扩展分区，如图6所示，扩展分区中逻辑驱动器的扩展引导记录是一个连接表。该图显示了一个扩展分区上的三个逻辑驱动器，说明了前面的逻辑驱动器和最后一个逻辑驱动器之间在扩展分区表中的差异。

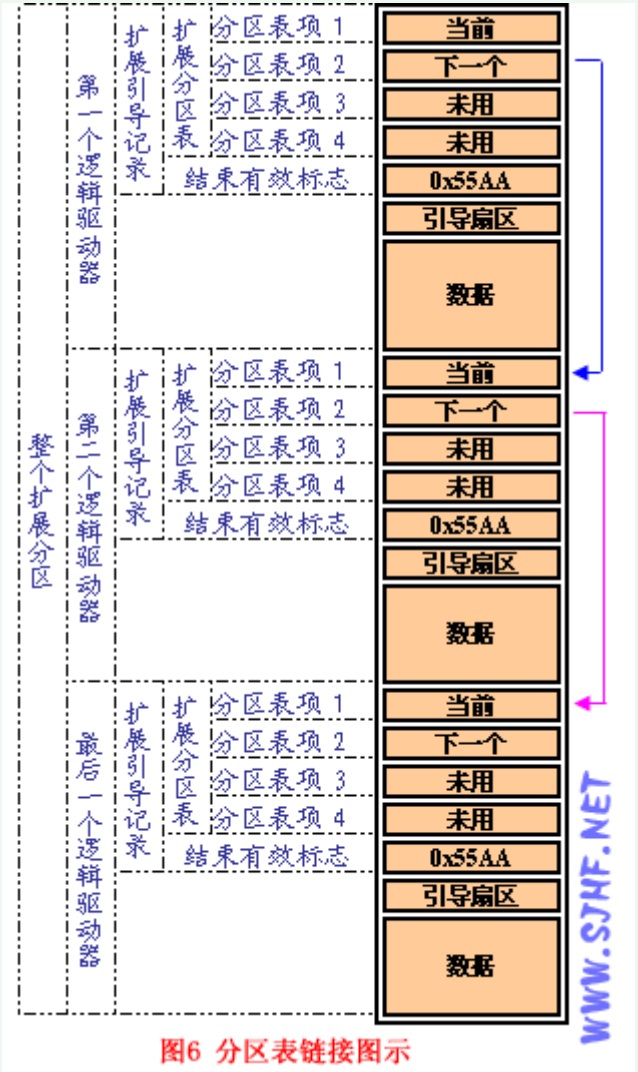

图6 分区表链接图示

除了扩展分区上最后一个逻辑驱动器外，表2中所描述的扩展分区表的格式在每个逻辑驱动器中都是重复的：第一个项标识了逻辑驱动器本身的引导扇区，第二个项标识了下一个逻辑驱动器的EBR。最后一个逻辑驱动器的扩展分区表只会列出它本身的分区项。最后一个扩展分区表的第二个项到第四个项被使用。

表2 扩展分区表项的内容

扩展分区表项	分区表项的内容
第一个项	包括数据的开始地址在内的与扩展分区中当前逻辑驱动器有关的信息
第二个项	有关扩展分区中的下一个逻辑驱动器的信息，包括包含下一个逻辑驱动器的EBR的扇区的地址。如果不存在进一步的逻辑驱动器的话，该字段不会被使用
第三个项	未用
第四个项	未用

扩展分区表项中的相对扇区数字段所显示的是从扩展分区开始到逻辑驱动器中第一个扇区的位移的字节数。总扇区数字段中的数是指组成该逻辑驱动器的扇区数目。总扇区数字段的值等于从扩展分区表项所定义的引导扇区到逻辑驱动器末尾的扇区数。

有时候在磁盘的末尾会有剩余空间，剩余空间是什么呢？我们前面说到，分区是以1柱面的容量为分区粒度的，那么如果磁盘总空间不是整数个柱面的话，不够一个柱面的剩下的空间就是剩余空间了，这部分空间并不参与分区，所以一般无法利用。照道理说，磁盘的物理模式决定了磁盘的总容量就应该是整数个柱面的容量，为什么会有不够一个柱面的空间呢。在我的理解看来，本来现在的磁盘为了更大的利用空间，一般在物理上并不是按照外围的扇区大于里圈的扇区这种管理方式，只是为了与操作系统兼容而抽象出来CHS。可能其实际空间容量不一定正好为整数个柱面的容量吧。关于这点，如有高见，请告知http://www.sjhf.net或zymail@vip.sina.com，sjhf@sjhf.net。

单击此处查看PDF版全文

- 上一篇文章： 没有了
- 下一篇文章： FAT文件系统原理(二)

【关闭窗口】

站内文章搜索 文章标题 所有栏目 关键字 搜索

首页 联系我们 加入收藏 版权申明 文章地图 下载地图 RSS生成 XML生成 友情链接

北亚数据恢复中心
全国统一客服电话：4006-505-808
北京部：北京市海淀区中关村大街11号E世界A座832B