

文件管理

- ◎ 文件系统实例—FAT
- ◎ 文件操作的实现
- ◎ 文件系统的管理
- ◎ 文件系统的安全性
- ◎ 文件系统的性能问题
- ◎ 文件系统的两个练习

最后呢我们给出文件系统的两个练习



文件系统实例——FAT

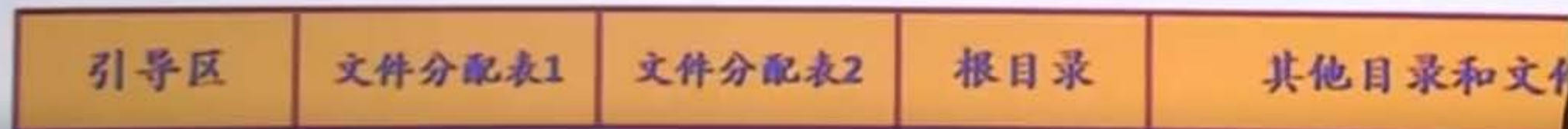
首先我们来介绍一下 另一个文件系统的实例。



WINDOWS — FAT16文件系统

- 簇大小：1、2、4、8、16、32或64扇区
- 文件系统的数据记录在“引导扇区”中
- 文件分配表FAT的作用
描述簇的分配状态、标注下一簇的簇号等
- **FAT表项：2字节**
- **目录项：32字节**
- **根目录大小固定**

FAT12、FAT16、FAT32



FAT文件系统——MBR

主引导记录 — 0号扇区

字节偏移量 (16进制)	域长	含义
00 - 1BD	446字节	引导代码
1BE - 1CD	16字节	分区表项1
1CE - 1DD	16字节	分区表项2
1DE - 1ED	16字节	分区表项3
1EE - 1FD	16字节	分区表项4
1FE - 1FF	2字节	扇区结束标记 (55AA)

这个磁盘的分区的情况好，那么我们来看一下

MBR	分区表	分区1(主分区)	分区2	分区3
-----	-----	----------	-----	-----



FAT文件系统——DBR

FAT32

引导区	文件分配表1	文件分配表2	根目录	其他目录和文件
-----	--------	--------	-----	---------

字节偏移量 (16进制)	域长	样值 (16进制)	含义
00	3字节	EB 3C 90	转移指令
03	8字节	MSDOS5.0	文件系统标志(ASCII码)
0B	25字节		BIOS参数块 (BIOS Parameter Block, BPB)
24	54字节		扩展BIOS参数块(Extended BIOS Parameter Block, EBPB)
5A	410字节		引导代码
1FE	2字节	55 AA	扇区结束标记

引导扇区—BIOS参数块

FAT32

字节偏移量 (16进制)	域长	样值(16进制)	含义
0B	2字节	00 02	每扇区字节数
0D	1字节	08	每簇扇区数
0E	2字节	01 00	保留扇区数: 从分区引导扇区到第一个文件分配表开始的扇区数
10	1字节	02	文件分配表个数
11	2字节	00 02	根目录项数
13	2字节	00 00	扇区数(小): 卷上的扇区数, 如果该数超过16位(65535)的话
15	1字节	F8	介质类型: F8表明为硬盘, F0表明为软盘
16	2字节	C9 00	每个文件分配表的扇区数(FAT32不用)
18	2字节	3F 00	每磁道扇区数
1A	2字节	10 00	磁头数
1C	4字节	3F 00 00 00	隐藏扇区数
20	4字节	51 42 06 00	扇区数(大): 如果小扇区数域的取值为0, 域包含的是卷中的扇区总数

引导扇区—扩展BIOS参数块(EBPB)

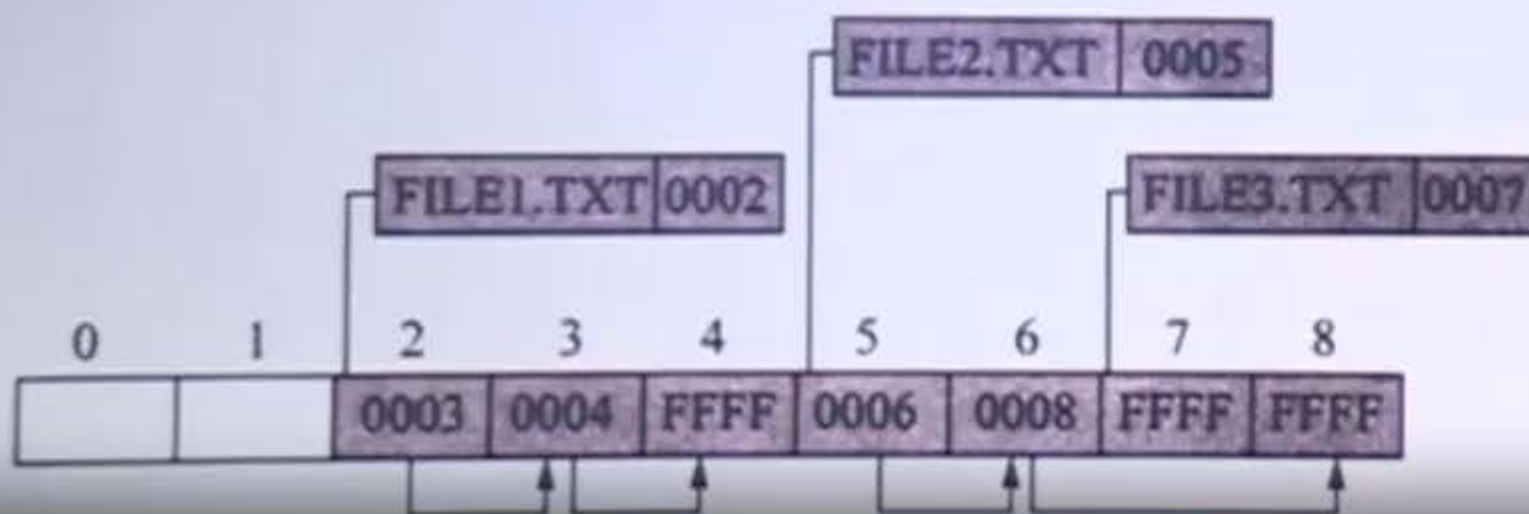
FAT32

字节偏移量(16进制)	域长	含义
→ 24	4字节	每个文件分配表的扇区数(FAT32用)
28	2字节	标记: bit7为1, 表示只有一个FAT; 否则, 两个FAT互为镜像
2A	2字节	版本号
→ 2C	2字节	根目录起始簇号, 通常为2
30	2字节	FSINFO所在扇区, 通常1扇区
32	2字节	引导扇区备份, 通常是6号扇区
34	12字节	未用
40	1字节	BIOS Int 13H设备号
41	1字节	未用
42	1字节	扩展引导标识, 如果后面三个值有效, 设为0x29
43	4字节	卷序列号: 当格式化卷时创建的一个唯一的数字
47	11字节	卷标(ASCII码), 建立文件系统时由用户指定
52	8字节	系统ID: 根据磁盘的格式, 该域的取值为FAT12或FAT16



文件分配表FAT

- 可以把文件分配表看成是一个整数数组，每个整数代表磁盘分区的一个簇号
- 状态
 - 未使用、坏簇、系统保留、被文件占用（下一簇簇号）、最后一簇（0xFFFF）
- 簇号从0开始编号，簇0和簇1是保留的



FAT16目录项

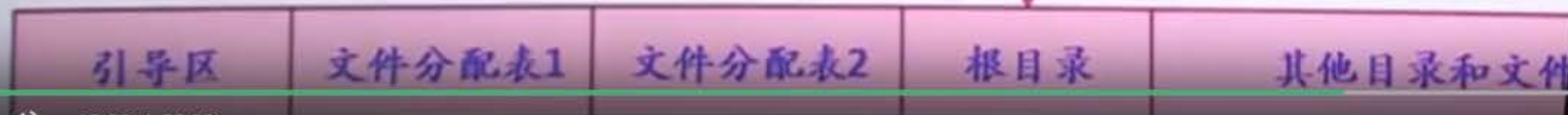
偏移	域长	含义
00h	8	文件名
08h	3	文件扩展名
0Bh	1	文件属性字节
0Ch	10	保留
16h	2	最后一次修改的时间
18h	2	最后一次修改的日期
1Ah	2	起始簇号
1Ch	4	文件大小

位	7-6	5	4	3	2	1	0
	保留	归档	目录	卷标	系统	隐藏	只读



FAT32文件系统

- FAT32的根目录区（**ROOT区**）不是固定区域、固定大小，而是数据区的一部分，采用与子目录文件相同的管理方式
- 目录项仍占32字节，但分为各种类型（包括：“.”目录项、“..”目录项、短文件名目录项、长文件名目录项、卷标项（根目录）、已删除目录项（第一字节为0xE5）等）
- 支持长文件名格式
- 支持Unicode
- 不支持高级容错特性，不具有内部安全特性



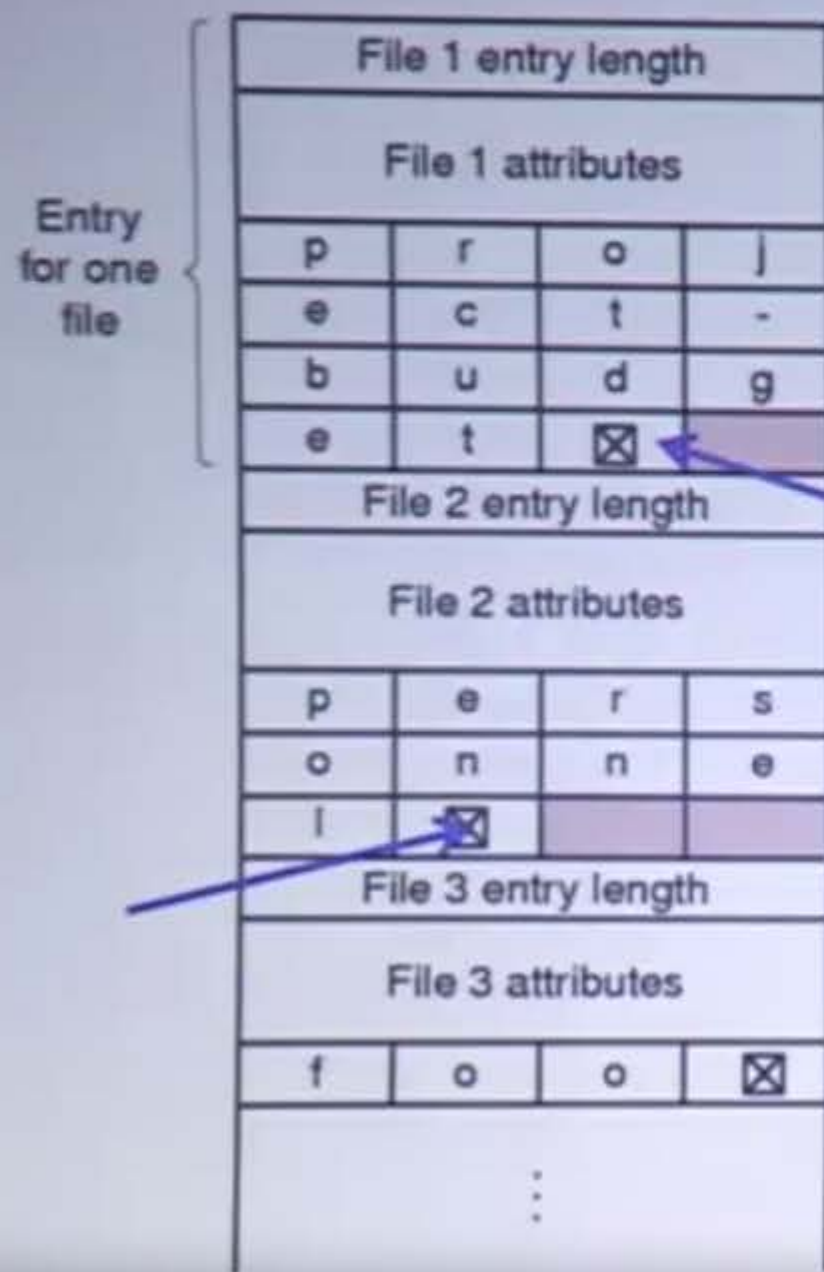
FAT32目录项

7	6	5	4	3	2	1	0
		归档	目录	卷标	系统	隐藏	只读

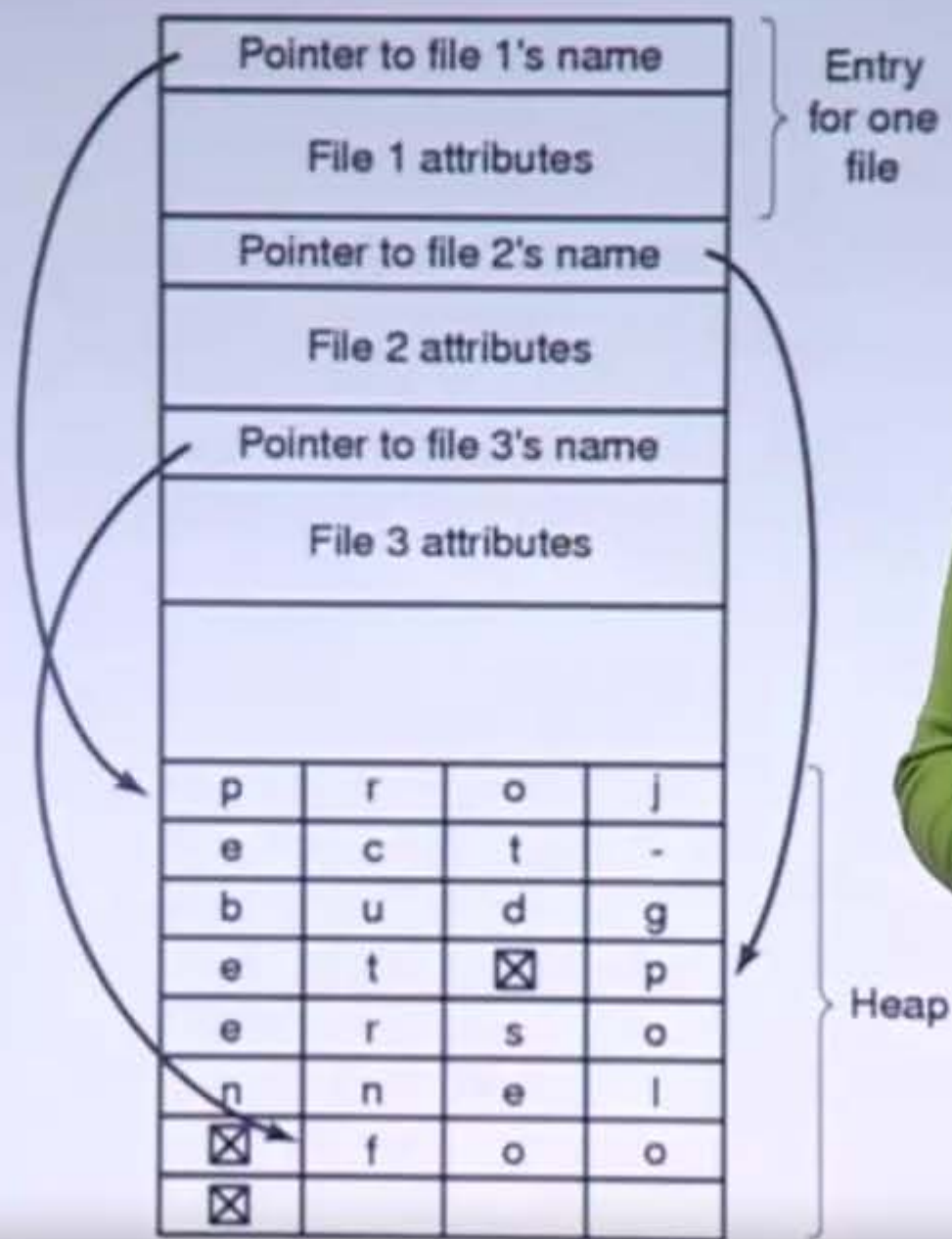
偏移	域长(字节)	含义
00h	8	文件名
08h	3	文件扩展名
0Bh	1	文件属性字节
0Ch	1	保留
0Dh	1	→ 创建时间, 精确到1/10秒
0Eh	2	→ 文件创建时间
10h	2	→ 文件创建日期
12h	2	→ 文件最后访问日期
14h	2	→ 起始簇号的高16位
16h	2	最后一次修改的时间
18h	2	最后一次修改的日期
1Ah	2	起始簇号的低16位
1Ch	4	文件长度 (子目录为0)



一般长文件名的实现方式



(a)



(b)



FAT32—长文件名目录项格式

偏移	长度	含义
00h	1	位0-5给出序号, 位6 表示长文件最后一个目录项
01h	10	长文件名(unicode码) ① 前5个字符
0Bh	1	0x0F(长文件名目录项标志)
0Ch	1	保留
0Dh	1	校验值(由短文件名计算得出)
0Eh	12	长文件名(unicode码) ② 6个字符
1Ah	2	文件起始簇号, 常置0
1Ch	4	长文件名(unicode码) ③ 2个字符

例子: 文件名为The quick brown.fox, 采用Unicode编码



第2个长文件名目录项(最后一个)

长文件名目录项属性标识

0x42	v	n	.	f	o	0x0F	0x00	che	\$u
0x0000	0xFFFF	0xFFFF	0xFFFF	0xFFFF	0x0000	0xFFFF			
0x01	T	h	e			0x0F	0x00		
i	c	k		b					
T	H	E	Q	U	I	~	1	F	O
X						0x20	NT		
Create Date	Last Access Date	0x0000	Last Modified Time	Last Modified Date	First Cluster				

短目录项

第1个长文件名目录项

5个目录项

The quick brown fox jumps over the lazy dog

68	d	o	g	A	0	C	K					0						
3	o	v	e	A	0	C	K	t	h	e		l	a	0	z	y		
2	w	n		f	o	A	0	C	K	x		j	u	m	p	0	s	
1	T	h	e		q	A	0	C	K	u	i	c	k		b	0	r	e
	T	H	E	Q	U	I	~	1		A	N	S	Creation time	Last acc	Upp	Last write	Low	Size
Bytes																		

Windows为其建立了五个目录项、四个保存长文件名、一个保存压缩文件名THEQUI~1



大家好！今天我给大家带来的是操作系统原理课的第十讲 文件系统的第二部分。在这部分里头我们会介绍另一个文件系统实例 是 Windows 的 FAT 文件系统。然后我们介绍文件操作的实现，文件系统的管理 文件系统的安全性，我们还会去介绍如何提高文件系统 的性能。最后呢我们给出文件系统的两个练习 首先我们来介绍一下 另一个文件系统的实例。Windows 的 FAT 文件系统 那么 Windows 呢有很多的文件系统。我们这里介绍的 FAT 文件系统呢，是比较简单的一种 在 FAT 文件系统当中，簇的大小 往往是一个扇区、两个扇区，或者是 2 的 整数次幂的倍数的扇区。那么文件系统的这个数据呢，实际上是记录在 引导扇区当中。也就是说关于文件卷的管理信息。主要包括在引导扇区当中 那么这是和 UNIX 文件系统的布局不太一样 因为 UNIX 文件系统，它这个相关的这个管理数据呢，是保存在超级数据块里头 而 FAT 文件系统呢，就用引导扇区来保存这些信息了 那么在 FAT 文件系统其中的一个重要的就是 文件分配表。文件分配表的作用呢，前面我们已经介绍过了，现在再复习一下 它主要呢是来描述簇的分配状态以及 标注下一簇的簇号。那么还有一些其它的作用在 FAT 文件系统当中 因为我们是一个 FAT16 文件系统 因此，FAT 表项就是 FAT 表的每一个表项的长度是 两个字节，那么就是 16 位，所以这是 FAT 16 这个名字的来历 16 就表示 FAT 文件表，表项的长度占多少位 那么在 FAT 文件系统当中的每一个文件 都对应了一个目录项，这里头 目录项就是等于我们前面所说的 FCB。那么目录项呢 占 32 个字节，啊，一个目录项占 32 个字节 FAT16 文件系统有一个特点 跟目录区的大小是固定的。也就是说 在文件系统的布局当中，根目录文件它的大小 确定了在什么位置？多大？就不再改变，是个固定大小 关于 FAT 文件系统，我们还可能会有 FAT12 和 FAT32 文件系统。我们现在用的优盘当中啊，基本上都是用的是 FAT32 文件系统 那么，12 就表示 FAT 文件的分配表当中的表项长度是 12 位，一个半字节 而 FAT32 呢，那就是 FAT 文件表的这个表项是占 4 字节 这就是它的这个标号的来历 那我们来看一下，FAT 文件系统的主引导记录 所谓主引导记录呢，通常是放在 0 号扇区的 那我们看一下这个布局当中 0 号扇区呢，是存放了主引导记录 那么这个扇区里头，512 字节有一个分配 前面 446 个字节呢是引导代码，那也就是说 操作系统引导是把相关的引导代码放在这个位置 那么其它的呢，是关于 这个磁盘的分区的情况好，那么我们来看一下 每一个分区，因为我们讲过，文件卷是建立在每一个磁盘分区之上的 因此每一个分区是这样 一个文件系统的布局 也有一个引导区。我们看看这个引导区包括了哪些信息 在这个引导区当中的 512 个字节，也

有相应的作用。我们这里头比较关心的呢，是这 25 个字节 那么它是用来存放 BIOS 的参数 文件系统的相关信息呢，也记录在这里头。还有 拓展的 BIOS 参数呢，这里有 54 个字节 当然了，这个字节的数量呢，那么不同的版本 可能不太一样。大致上，我们知道有这样一些字节 来描述文件系统的相关的管理信息 那么如果从某一个分区引导操作系统呢 那么相应的引导代码呢，可以放在这里头有 410 个字节 这就是在一个分区上的一个 引导扇区的这样一个布局 那我们来看一下 BIOS 参数这一块，当然了，这里有很多的信息 我们只关心我们比较重要的，啊，比如说 我们关心说，到底每一簇它是占多少个扇区呢？当然我们说的它是 2 的整数次幂的这个倍数 但是呢，具体的信息呢记录在这个位置 同样的道理呢，我们还要知道根目录项占多大 因为我们前面讲过，FAT16 的根目录文件的大小呢，是一个固定大小 所以根目录到底能保存多少个目录项。我们可以从这里得到我们所需要的信息 当然还有其它的一些信息。那么 除了刚才 BIOS 参数之外呢，我们还有一些拓展的这个参数 这里头我们关心呢，比如说每个文件分配表，它一共占多少个扇区数 也就是这个文件分配表占多大。因为我们前面 看到，整个 FAT 文件系统的布局当中 有两个文件分配表。当然第二个是第一个的镜像 那么它到底多大？在这里头我们可以找到我们所需要的信息，有多少个扇区。另外呢，FAT 分配表确定了大小之后 那么，我们的根目录从哪一个地方开始，它的起始簇号记录在这里头 当然，通常在 FAT 文件系统当中的起始簇号，都是从 2 开始记录 FAT 文件表是 FAT 文件系统当中一个非常重要的数据结构 我们可以把文件分配表看成是一个 整数的数组。每一个整数就 代表了整个磁盘分区的一个簇号，所以 一个磁盘分区有多少簇，在这个数组当中就有多少个元素 那么每一个整数呢，如果是 FAT16，那么这个整数大小呢就是 2 个字节 如果是 FAT32，那么这个整数大小呢就是 4 个字节 那么每一个整数它代表了一种状态 在这里头，我们描述了，有几种不同的状态 如果这个簇没有被使用 那我们可能用 0 来表示这个簇是没有被使用的 当然也不排除在整个文件系统当中有一些簇是坏簇 因此呢，我们可以通过坏簇的那个标志呢，来标记这一簇呢不能再用了 当然有一些簇呢是系统保留的。比如说 0 簇和 1 簇，通常是系统保留的 另外的就是，我们比较关注的就当这一簇分配给了一个文件，它的这个文件的下一簇的簇号 要记录在这一簇里头，所以每一簇它是记录了这个被文件占用的下一簇的簇号 如果没有下一簇的簇号呢，那实际上得是最后一簇，也就是用一个特殊的标志，来表

示这是最后一簇 我们看一个简单的示意 我们从目录项当中得到了一个文件的起始簇号 那么通过起始簇号, 在文件分配表当中找到对应的整数 那么这里头, 如果这个文件还有下一簇, 那么就在这里记录了下一簇的簇号。如果这一簇已经是这个文件的 最后一簇了, 我们就要把特殊的标记, 啊, 记录在这里 这就是文件分配表的作用 我们前面所讲过了 UNIX文件系统当中的 i 节点, 加上目录项 正好等于我们所介绍的这个文件系统的重要数据结构FCB 而在FAT文件系统里头, FCB就等于目录项 所以我们来看一下FAT16的目录项, 一共32个字节 那么, 这32个字节的分配呢, 比如说文件名占8个字节 文件的扩展名占3个字节, 那么还有一个字节 标识着文件的属性 那么文件有哪些属性呢? 比如说这个文件是只读文件 这个文件呢是隐藏文件, 这个文件是系统文件, 所以这些属性 都通过这一个字节当中的某些位来标识出来 在FAT16目录项当中 还记录了这个文件的最后一次修改的时间 修改的日期, 最重要的呢是要给出 这个文件它在整个磁盘的起始 簇号, 一个文件有若干簇的话, 它的起始簇号是记录在 FAT16目录项里头, 同时还记录了这个文件的大小 我们看到32个字节没有全用完, 还有10个字节呢是保留着 这就是FAT16的目录项的一个格式 那我们再来看看FAT32文件系统 在FAT32文件系统当中呢 根目录区不再是固定的大小了 原来呢跟目录区是固定大小、固定区域 现在呢根目录区完全和其它的文件或目录同样对待 它是整个数据区的一部分 好那我们来看看, 当然从起始 点上来讲, 那么根目录呢还可以从这里起始 但是, 如果根目录很大, 它不一定全放在前面, 它可以放在磁盘的后面的一些位置 随着根目录的不断长大, 那么 系统会为它分配一些其它的磁盘空间的位置 目录项的大小仍然是32个字节 但是呢由于FAT32文件系统呢, 支持长文件名的 格式, 还支持Unicode编码, 所以 它的目录项, 我们分成多种类型, 其中包括了"." 当前目录的目录项 "." 呢我们知道是副目录的目录项 还有一个叫短文件名的目录项 也就是8.3这个格式的文件名的目录项 另外呢, 我们刚才说了, 它支持长文件名的 格式, 因此有一个长文件名的目录项这种类型 当然还有其他一些目录项的类型 所以FAT32文件系统要比FAT16文件系统更复杂 那么FAT32文件系统的目录项呢 每一个目录项是32个字节, 因为它有不同种类的目录项 我们先来看看一个基本的目录项的一个内容 基本目录项的内容呢, 和我们介绍的FAT16文件系统的这个 文件的目录项呢, 大同小异, 这里头呢就把前面 保留的10个字节把它用上了, 我们来看一下 由于FAT32呢, 它的簇号是用4个字节表示的 所以原来只有16

位不够，所以我们又增加了16位，也就是 起始簇号的高16位，原来的呢是起始簇号的低16位 那么，剩下的还有8个字节 那么这8个字节呢用了7个字节，其中 创建时间，有一个精确到了 这个 1/10 秒的，这样一个创建时间 那么另外呢，还有我们的文件创建的时间 文件创建的日期，以及文件的最后访问日期 前面保留的呢，是最后一次修改时间和最后一次修改日期 其它的都没有变，这是FAT32 文件系统目录项的一个布局，在一个文件系统中 如何支持长文件名 我们最简单的方法呢 是在目录项当中，按照 这个最长的文件名，给它留出相应的位置 那么如果文件名短，那么它很多的空间就浪费了 所以我们可以有这样两种实现方案 左边这种实现方案呢，是将目录项 的长度呢，是不固定的 我们来看一下，在一个目录项当中 第一个字段呢，给出这个目录项到底有多长 它的大小，然后 把固定长度的一些属性，记录在后面 接着就是文件名，因为文件名都是长度不一样的 通过留出足够的空间给文件名 这里头我们可以看到，这个文件名结束会有相应的特殊的标志 然后我们接着可以存放下一个文件对应的目录项，也是 从目录项的长度开始记录 目录项的一些固定长度的属性，以及 变长的文件名，那么这里头的缺点就是说 当我有一个文件删除的时候，就留出了一块空间 而这块空间呢，可能，不能够放下一个文件的目录项了，因为 下一个文件的文件名可能很长，这地方不能放下，所以就会产生了一些碎片 那我们来看看右边这种方案 右边这种方案是说，由于文件名的长度不固定 有长有短，所以我们希望，每一个目录项的大小是固定的 好我们来看一下，我们留出的一个目录项大小是个固定大小 其中包括了一个指向文件名的，起始的一个指针 然后是文件的相关的属性 所以我们可以看到，文件的目录项，啊，文件所对应的目录项，都是固定长度的 那么所有的文件名，都把它存放在另外一个区域的一个叫堆的地方 以堆的方式存放这个文件名，所以每一个目录项会有一个指针指向 这个文件所对应的这个堆当中的某一个位置 然后这里头存放了相关的这个文件名，但是文件名长短 都没有关系，因为最后也有一个特殊的标记，标志是文件结束 所以这是一个通常有两种 可以采用的支持长文件名的这样一个方法 我们来看看FAT文件系统，是怎么来做这件事的 在FAT32 当中的目录项有一种类型，叫做长文件名目录项 那同样也是32个字节，那么长文件名目录项的类型当中 我们来看一下在这里头有10个字节 然后有12个字节，然后有4个字节，分别 记录了是这个文件的文件名 那么这10个字节，因为 我们采用的是unicode编码，所以两个字符 两个字节代表一个字符，所以这里头

可以保存 这个文件的前5个字符，这里保存 6个字符，这里保存两个字符，所以一共在长文件名的目录项当中，可以保存这个文件名的一共13个字符 其它的呢，就是一些标志，比如说这个 字节当中如果是0F就是表示这个 目录项的类型，是长文件名的目录项 好那么，长文件名目录项如果一个文件很长，那么一个 长文件名目录项不够，我们就需要第二个长文件名目录项 最后呢如果是最后一个目录项的话，那么就在 第1个字节当中的，位6，第6位这个 地方，来记录它是，表示它是文件名的最后一个目录项了，下面我们给出一个 长的文件名The quick brown.fox 一共是19个字符，空格儿也表示是一个字符 点也是一个字符，那么 在一个支持Unicode的编码的这样一个FAT32文件系统当中 我们怎么样来为这个文件建立对应的目录项呢？实际上呢，刚才这样一个长度的文件名 在FAT文件系统当中呢，是占据了3个目录项 第一个目录项呢，就是我们前面讲的短文件名目录项 这里头记录了，我们相关的，比如说 创建的时间，创建的日期等等，起始簇号 这样一些信息，然后，后面两个目录项呢主要是为了保存 长文件名，因为我们刚才说了一共是19个字符，所以呢 一个目录项可以保存13个字符，所以需要两个目录项 我们来看一下，这就是，我们来看一下这个文件啊 The这个空格，quick 这有5个字符，然后 这里头又有6个字符，然后 这个目录项用满了之后呢，再接着 开辟一个目录项，长文件名目录项，来记录剩余的这些字符 那么再看一个例子，这个例子呢 实际上是，更长一点 The quick brown fox jumps over the lazy dog 这么长的一个文件名，那么我们在Windows的FAT文件系统当中呢，为它建立了 五个目录项，其中一个短文件名目录项 四个是长文件名的目录项 所以在一个FAT32文件系统当中的任何一个文件 要根据它的文件名的长短，我们可以看到，它实际上呢是有对应多个目录项 而FAT16文件系统呢，因为它是8.3结构的文件名，所以只对应了一个目录项，所以一个 文件对应一个目录项，而FAT32的话呢，是根据文件名的长短 那么一个文件可能对应两个或者两个以上的目录项，因为 在这里头基本目录项，就是短文件目录项是必须有的 然后呢，需要一个长文件名目录项，不够了再接着