

# 优盘文件系统 (FOR C)

优盘上的数据按照其不同的特点和作用大致可分为 5 部分：MBR 区、DBR 区、FAT 区、FDT 区和 DATA 区。

## 主引导记录(MBR)

绝对扇区号为: MBR\_LBA=0x00000000 处是主引导记录, 等同位于硬盘的 0 磁道 0 柱面 1 扇区。在总共 512 字节的主引导扇区中, MBR 只占用了其中的 446 个字节 (ofs:0 - ofs:1BDH), 另外的 64 个字节 (ofs:1BEH - ofs:1FDH) 交给了 **DPT(Disk Partition Table 盘分区表)**, 最后两个字节 “55 AA” (ofs:1FEH - ofs:1FFH) 是分区的结束标志。

■ MBR 定义如下:

```
typedef struct {
    uchar bootcode[446];    //ofs:0. 启动代码。“FA 33 C0 8E D0 BC...” .
    PartitionTable PT[4];   //ofs:446. 分区表 length=4*16.
    uint EndingFlag;        //ofs:510. 结束标识:0xAA55.
}MBR_tag;
```

■ Bootcode[446]启动代码一般是固定的, 用于引导 x86, 不用管。

■ 分区表项的定义如下:

```
typedef struct {
    uchar BootFlag; //启动标志
    CHS StartCHS; //分区开始的柱面、磁头、扇区
    uchar SystemID; //分区类型
    CHS EndCHS; //分区结束的柱面、磁头、扇区
    ulong RelativeSectors; //分区起始扇区数, 指分区相对于记录该分区的分区表的扇
                           //区位置之差 (该分区表: LBA=0x0)。
    ulong TotalSectors; //分区总扇区数
}PartitionTable;
```

■ 其中 CHS 为一个柱面、磁头、扇区的结构, 定义如下:

```
struct CHS {
    uchar Head; //磁头。
    unsigned Sector : 6; //扇区。
    unsigned CyH2 : 2; //柱面(高两位)。
    uchar CyL8; //柱面 (低八位)。

    ulong Cylinder() {return (uint)(CyH2)*256+CyL8;} //返回柱面值
    void SetCylinder(uint Cylinder) //设置柱面值 {
        CyH2=(Cylinder>>8)&0x3; CyL8=(Cylinder&0xff);
    }
};
```

例: 80 01 01 00 0B FE BF FC 3F 00 00 00 7E 86 BB 00

在这里我们可以看到:

- “80” 是一个分区的激活标志, 表示系统可引导;
- “01 01 00” 表示分区开始的磁头号为 01, 开始的扇区号为 01, 开始的柱面号为 00;
- “0B” 表示分区的系统类型是 FAT32, 其他比较常用的有 04 (FAT16)、06 (bigFAT16)、01 (FAT12)、07 (NTFS);
- “FE BF FC” 表示分区结束的磁头号为 254, 分区结束的扇区号为 63、分区结束的柱面号为 764;
- “3F 00 00 00” 表示首扇区的相对扇区号为 63;
- “7E 86 BB 00” 表示总扇区数为 12289622。

## 系统引导记录 (DBR)

绝对扇区号为: DBR\_LBA=MBR.PT[0].RelativeSectors 处是 DBR, 等同位于硬盘的 0 磁道 1 柱面 1 扇区 (512 字节), 是操作系统可以直接访问的第一个扇区, 它包括一个引导程序和一个被称为 BPB (Bios Parameter Block) 的本分区参数记录表。引导程序的主要任务是当 MBR 将系统控制权交给它时, 判断本分区跟目录前两个文件是不是操作系统的引导文件 (以 DOS 为例, 即是 Io.sys 和 Msdos.sys)。如果确定存在, 就把其读入内存, 并把控制权交给该文件。BPB 参数块记录着本分区的起始扇区、结束扇区、文件存储格式、硬盘介质描述符、根目录大小、FAT 个数, 分配单元的大小等重要参数。

■ DBR 定义如下:

```
typedef struct {
    uchar bJmpBoot[3];           //ofs:0. 典型的如: 0xEB, 0x3E, 0x90。
    char  boEMName[8];           //ofs:3. 典型的如: “MSWIN4.1”。
    uint  BPB_wBytesPerSec;       //ofs:11. 每扇区字节数。
    uchar BPB_bSecPerClus;       //ofs:13. 每簇扇区数。
    uint  BPB_wReservedSec;      //ofs:14. 保留扇区数, 从 DBR 到 FAT 的扇区数。
    uchar BPB_bNumFATs;          //ofs:16. FAT 的个数。
    uint  BPB_wRootEntry;        //ofs:17. 根目录项数。
    uint  BPB_wTotalSec;         //ofs:19. 分区总扇区数 (<32M 时用)。
    uchar BPB_bMedia;            //ofs:21. 分区介质标识, 优盘一般用 0xF8。
    uint  BPB_wSecPerFAT;        //ofs:22. 每个 FAT 占的扇区数。
    uint  BPB_wSecPerTrk;        //ofs:24. 每道扇区数。
    uint  BPB_wHeads;            //ofs:26. 磁头数。
    ulong BPB_dHiddSec;          //ofs:28. 隐藏扇区数, 从 MBR 到 DBR 的扇区数。
    ulong BPB_dBigTotalSec;      //ofs:32. 分区总扇区数 (>=32M 时用)。
    uchar bDrvNum;               //ofs:36. 软盘使用 0x00, 硬盘使用 0x80。
    uchar bReserved1;            //ofs:37. 保留。
    uchar bBootSig;              //ofs:38. 扩展引导标记: 0x29。
    uchar bVolID[4];             //ofs:39. 盘序列号。
    char  bVolLab[11];           //ofs:43. “Msdos      ”。
    char  FileSysType[8];        //ofs:54. “FAT16    ”。
    uchar ExecutableCode[448];   //ofs:62. 引导代码。
    uint  EndingFlag;            //ofs:510. 结束标识: 0xAA55。
} DBR_tag;
```

DOS 引导记录公式:

- 文件分配表 = 保留扇区数
- 根目录 = 保留扇区数 + FAT 的个数 × 每个 FAT 的扇区数
- 数据区 = 根目录逻辑扇区号 + (32 × 根目录中目录项数) / 每扇区字节数
- 绝对扇区号 = 逻辑扇区号 + 隐含扇区数
- 扇区号 = (绝对扇区号 % 每磁道扇区数) + 1
- 磁头号 = (绝对扇区号 / 每磁道扇区数) % 磁头数
- 磁道号 = (绝对扇区号 / 每磁道扇区数) / 磁头数

要点:

- 1) DBR 位于柱面 0, 磁头 1, 扇区 1, 其逻辑扇区号为 0
- 2) DBR 包含 DOS 引导程序和 BPB。
- 3) BPB 十分重要, 由此可算出逻辑地址与物理地址。

## 文件分配表 (FAT)

绝对扇区号为:  $FAT\_LBA = DBR\_LBA + BPB\_wReservedSec$  处是文件分配表, 是 DOS 文件组织结构的主要组成部分。我们知道 DOS 进行分配的最基本单位是簇。文件分配表是反映硬盘上所有簇的使用情况, 通过查文件分配表可以得知任一簇的使用情况。DOS 在给一个文件分配空间时总先扫描 FAT, 找到第一个可用簇, 将该空间分配给文件, 并将该簇的簇号填到目录的相应段内。即形成了“簇号链”。FAT 就是记录文件簇号的一张表。FAT 的头两个域为保留域, 对 FAT12 来说是 3 个字节, FAT16 来说是 4 个字节。其中头一个字节是用来描述介质的, 其余字节为 FFH。介质格式与 BPB 相同。

FAT 结构含义: 一般 FAT 表的第一项为 FF8H 或 FFF8H。

FAT12	FAT16	意义
000H	0000H	可用
FF0H—FF6H	FFF0H—FFF6H	保留
FF7H	FFF7H	坏
FF8H—FFFH	FFF8H—FFFFH	文件最后一个簇
×××H	××××H	文件下一个簇

对于 FAT16, 簇号 × 2 作偏移地址, 从 FAT 中取出一字即为 FAT 中的域。

逻辑扇区号 = 数据区起始逻辑扇区号 + (簇号 - 2) × 每簇扇区数

簇号 = (逻辑扇区号 - 数据区起始逻辑扇区号) / 每簇扇区数 + 2

要点:

- 1) FAT 反映硬盘上所有簇的使用情况, 它记录了文件在硬盘中具体位置 (簇)。
- 2) 文件第一个簇号 (在目录表中) 和 FAT 的该文件的簇号串起来形成文件的“簇号链”, 修复被破坏的文件就是根据这条链。
- 3) 由簇号可算逻辑扇区号, 反之, 由逻辑扇区号也可以算出簇号, 公式如上。
- 4) FAT 位于 DBR 之后, 其 DOS 扇区号从 1 开始。

## 文件目录表 (FDT)

绝对扇区号为:  $FDT\_LBA = FAT\_LBA + BPB\_bNumFATs * BPB\_wSecPerFAT$  处是文件目录表, DOS 文件组织结构的又一重要组成部分。文件目录分为两类: 根目录, 子目录。根目录有

一个，子目录可以有多个。子目录下还可以有子目录，从而形成“树状”的文件目录结构。子目录其实是一种特殊的文件，DOS 为目录项分配 32 字节。

目录项定义如下：

```
typedef struct{
    char  FileName[8];  //ofs:0. 文件名
    char  ExtName[3];   //ofs:8. 扩展名
    uchar attribute;    //ofs:11. 文件属性。典型值：存档(0x20)、卷标(0x08)。
    char  reserved[10]; //ofs:21. 保留
    uint  time;         //ofs:22. 时间
    uint  data;         //ofs:24. 日期
    uint  StartClus;    //ofs:26. 开始簇号
    ulong FileLength;   //ofs:28. 文件长度
}DIR_tag;
```

1) 目录项文件名区域中第一个字节还有特殊的意义：

- 00H 代表未使用。
- 05H 代表实际名为 E5H。
- E5H 代表此文件已被删除。

2) 目录项属性区域的这个字节各个位的意义如下：

7	6	5	4	3	2	1	0
未	修	修	子	卷	系	隐	只
用	改	改	目	标	统	藏	读
标	标	录		属	属	属	
志	志			性	性	性	

3) WINDOWS 的长文件名使用了上表中所说的“保留”这片区域。

4) 时间：time = Hr \* 2048 + Min \* 32 + Sec + 2。

5) 日期：time = (Yr-1980) \* 512 + Mon \* 32 + Day。

6) 簇号与逻辑扇区号的关系为：

逻辑扇区号 = (簇号 - 2) × 每簇扇区数 + 数据区起始逻辑扇区号。

7) 要点：

- 文件目录是记录所有文件，子目录名，扩展名属性，建立或删除最后修改日期。文件开始簇号及文件长度的一张登记表。
- DOS 中 DIR 列出的内容是根据文件目录表得到的。
- 文件起始簇号填在文件目录中，其余簇都填在 FAT 中上一簇的位置上。

## 数据区 (DATA)

- 数据区绝对扇区号 = 根目录绝对扇区号 + (32 × 根目录中目录项数) / 每扇区字节数  
表达式：DATA\_LBA = FDT\_LBA + ( 32 \* BPB\_wRootEntry ) / BPB\_wBytesPerSec。