

## 第五章 文件系统的实现

### 5.1 FAT 文件系统规范

#### 5.1.1 文件用户接口的抽象概念和主要数据结构

带有文件系统的实时操作系统允许用户任务通过一个严格定义的接口与文件系统进行交互。该接口对用户屏蔽了文件系统的细节，同时指定了所有相关系统调用的行为和语义。接口提供了一组抽象概念：文件（File），目录（Directory），文件描述符（File Descriptor），文件系统（File System）和文件的输入/输出（File I/O）等。

从用户角度出发，文件系统的设计目标是：利用以上用户接口，能够使用户完备、方便地实现如下功能：

- 文件的定位与存取。
- 文件的操作与控制。
- 文件的共享与保护。

从系统角度出发，文件系统的设计目标是：

- 安全可靠
- 实时响应
- 对多任务的支持
- 接口标准的开放性和可移植性
- 可伸缩性和可配置性
- 开放的体系结构
- 支持多种文件类型

##### 5.1.1.1 文件和目录

文件是一个抽象机制，它提供了一种把信息保存在磁盘上而且便于以后读取

的方法。通过这种机制,用户可以创建文件,并且把数据写入文件,从而达到保存数据的目的,不必了解信息存储的方法、位置等细节信息。抽象机制最重要的特征就是用文件名来实现文件的管理。

文件系统用一种层次的树状结构来管理文件的名字空间。树由文件和目录构成,文件位于树叶的位置上,目录是管理文件系统结构的系统文件,其内容为目录下面的文件及其它子目录的名字信息。

整个文件系统组织就象一棵倒置的树,只有一个根节点。从这个根节点出发,可以直接分支出文件,也可以分支出下一层的节点(即子目录,实质上它也就是一个文件)。进入了这个节点,即可以访问节点文件的内容,又可以到达下一层文件或子节点。每个文件和子节点都可以用一个名字加以标定,从一个节点出发到达指定文件所经由的节点名序列,称为路径。某一时刻所在的目录层次,则称为当前目录。树形目录的结构例图如下:

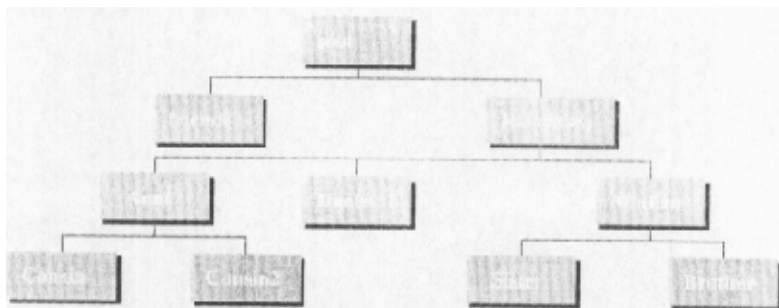


图 5-1 树形结构

FAT 文件系统中每一个文件或者目录除本身的数据以外,还在目录表中拥有一个目录项,它包含了文件名和其他属性。目录项中的属性字节表明了该文件是普通的数据文件还是用来表明其他文件的子目录。

每个目录项有 32 字节的结构,包含如下内容:

- 文件名,最多 8 个字符的字符串构成。
- 扩展名,最多 3 个字符的字符串构成。
- 其他文件属性。
- 文件创建日期和最后一次修改的时间。
- 文件的首簇在卷中的位置。
- 以字节为单位的文件长度。

- 保留十个字节供以后的版本使用。

FAT 文件系统定义的目录项结构如下:

```
typedef struct DFS_DIRENTRY {  
    char name[8];           /* 文件名 */  
    char extension[3];      /* 扩展名 */  
    unsigned char attribute; /* 属性 */  
    char reserved[10];      /* 保留 */  
    unsigned short time;    /* 建立/修改时间 */  
    unsigned short date;    /* 建立/修改日期 */  
    unsigned short start_cluster; /* 起始簇号 */  
    long length;           /* 按字节计的文件长度 */  
};
```

### 5.1.1.2 文件属性

文件系统为每一个文件维护一组属性,用来表示文件的相关信息,这些信息包括:只读、存档、系统、隐藏等。这些属性保存在文件目录项中,其定义如下:

- Read-only 只读文件,此文件不能被写入或删除;
- Hidden 隐藏文件(或又称隐含文件);
- System 系统文件,通常它们又是隐藏文件和只读文件;
- Normal 普通文件;
- Volume id 卷标项,表示该目录项的前 11 个字节为卷标,而且目录项必在根目录中。该目录项除表示卷标名、属性、日期和时间字节外,其他各字节的值无用;
- Subdir 子目录;
- Archive bit 文件为存档文件。

以下是系统中位于 fatfs.h 文件中的文件属性值:

```
#define D_NORMAL      0  
#define D_RDONLY     0x01  
#define D_HIDDEN     0x02
```

```
#define D_SYSTEM      0x04
#define D_VOLID       0x08
#define D_DIR         0x10
#define D_ARCHIVE     0x20
```

### 5.1.1.3 文件描述符

文件描述符为打开的文件对象的句柄，属于任务内部的对象。任务通过文件描述符来实现对文件的操作。根据下面定义的文件描述符，文件系统可以迅速找到与文件有关的数据结构。

```
struct f_node
{
    unsigned int f_count;    /* 文件的使用计数*/
    int f_mode;             /* 打开模式：读、写、读/写，文件允许被多次打开
进行读操作，但不允许同时多次打开被写*/
    unsigned char f_flags;   /* 文件标识*/
    /* f_flags
*bit0 = 1      目录被修改过
*bit1 = 1      根目录
*bit2 = 1      新创建的文件描述符
*bit3 = 1      文件描述符分配给目录/子目录
*bit4 = 1      目录已满
*/
    DIR_ENTRY f_dir;        /* 该文件的目录项拷贝*/
    unsigned long f_diroff;  /* 目录项在目录表中的偏移量*/
    unsigned int f_dirstart; /* 目录开始簇*/
    DRIVE *f_dpb;           /* 使用该文件的块设备*/
    unsigned long f_dsize;   /* 文件的大小*/
    unsigned long f_offset;  /* 文件指针的偏移量*/
    unsigned int f_back;     /* 文件的上一簇簇号*/
}
```

```
unsigned long f_cluster_offset; /* 簇的偏移量*/
unsigned int f_cluster;        /* 当前所在簇*/
unsigned int f_sector;        /* 当前簇所在的扇区*/
unsigned int f_boff;          /* 簇中的字节数 */
};
```

#### 5.1.1.4 文件输入/输出

文件输入/输出用来实现对文件内容的读取,或是把数据信息保存到指定的文件中。文件既可以顺序访问 (Sequential Access),也可被随机存取 (Random Access)。顺序访问是指任务从文件开始处顺序读取文件中的所有字节或者记录,随机存取则指以非顺序的方式实现文件中字节或记录的存取操作。

在文件系统中,使用文件偏移指针(见“文件描述符”数据结构)来定位文件中的内容。该指针代表文件中的某一位置,由操作系统来维护。

#### 5.1.1.5 文件加锁

文件系统组件允许多个任务并发地对同一个文件进行读或者写操作。为实现并发操作的同步,使用文件锁机制,对文件的部分或全部加锁。文件系统在文件描述符中用一系列属性对文件进行加锁操作。

#### 5.1.1.6 文件类型

文件系统组件支持多种类型的文件,包括正规文件、目录、设备文件、管道和 FIFO 等几种情况,以满足应用的不同需要。

正规文件 (Regular File) 即通常意义上的文件,用来存放用户数据。

目录 (Directory) 是管理文件系统结构的系统文件。

设备文件 在文件系统组件中,把设备当作一种特殊的文件进行处理。设备文件包括块设备文件、字符设备文件、网络文件 (Network File) 和 Flash 文件 (Flash File) 等几种情况。

管道和 FIFO 管道和 FIFO (First In First Out) 都是可以提供先入先出数据流

的文件抽象符。写操作将数据放在后面，而读操作则从前面把数据移去。一旦数据被读出，该数据就会被删掉。

## 5.1.2 FAT 文件系统的内部表示

### 5.1.2.1 分配表(File Allocation Table)概述

FAT(File allocation table)即文件分配表，它不用于表示引导区、文件目录表的信息，也不真正存储文件内容，只反映磁盘空间当前的使用情况，是整个文件系统的核心。

文件在磁盘的分布情况，是以簇链的方式记录在 FAT 中。每个文件都有自己的存储簇，可以是连续的也可以是不连续的，通过 FAT 表来实现其完整性。FAT 记录了除文件首簇以外文件使用的所有簇的情况(文件的首簇的使用情况记录在文件所在的目录项中，关于目录项的详细说明见§5.1.1.1)。FAT 项的序号与文件所使用的簇号有一一对应的关系。

图 5-2 给出的是 FAT 与文件磁盘分配的关系，配合这个例子及其说明，可以很清楚地了解 FAT 在文件系统中的作用以及文件系统是如何利用 FAT 完成工作的。

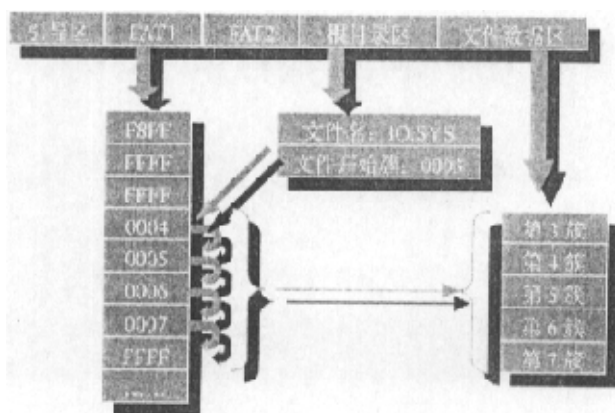


图 5-2 FAT 与文件磁盘分配的关系

说明:

1. 为了避免混淆，在没有特别说明的情况下我们所说的 FAT 就是指