FAT16文件系统在嵌入式操作系统μC-OS中实现

院泽嘉 董金明 北京航空航天大学

摘 要 文中论述在复杂嵌入式系统中建立文件系统的重要性，并提出将FAT16文件系统应用于嵌入式操作系统μC-OS的方案。根据FAT16文件系统总体的构架、基本的数据结构，针对嵌入式系统的特点搭建能够有效管理、读写数据文件的系统。其技术成熟，应用可靠，并且提供和μC/FS、Turbo C类似的API接口，便于使用，易于移植。

关键词 FAT16文件系统 嵌入式操作系统

Implementation of FAT16 File System on Embedded OS μC-OS

Yuan Zejia Dong Jinming

Abstract In embedded system, an embedded file system has to be used to manage those files stored on different media such as CF memory card, RAM and so on. In this paper, a highly effective and widely used file system——FAT16 was implemented for the sake of writing some files on a USB flash disk and these files must be identified by these operating systems, such as Windows98, Windows 2000, Windows NT.

Keywords FAT16 File System embedded operating system

1 嵌入式系统需要文件系统

近年来，嵌入式系统发展迅猛，这不仅体现在CPU处理能力迅猛提升、外界设备与处理器交互能力快速提高，更体现在嵌入式系统的复杂性日益增高、系统存储容量迅速增加。这些变化对于数据操作的灵活性提出越来越高的要求。以往那种直接针对存储介质读写数据的方式显然是不能够满足这种灵活的需求，因此，如何系统地、灵活地管理数据文件就成为嵌入式系统开发者无法回避的问题。

文件系统的使用，使开发者获得“路径”的概念，从而回避了存储介质的地址空间，不用每次都从头对物理介质进行操作。文件系统提供与底层物理存储介质无关的、清晰明了的API函数，使上层应用不再关心底层存储介质，而方便地利用标准接口访问底层存储器。接口的标准化，使程序的层次分明，增加程序的可移植性。因此，在嵌入式系统中加载文件系统是嵌入式系统发展的必然结果。

2 FAT文件系统的结构及FAT16文件系统的实现

由于目的是利用嵌入式软件通过USB控制芯片向普通U盘写入一系列文件，这些文件需要被Windows98、Windows2000、WindowsNT等操作系统正确识别，并进行读写操作，因此设计者选择了应用广泛的FAT16文件系统。FAT（File Allocation Table）文件系统历经FAT12、FAT16、FAT32不同阶段，是一种技术成熟、结构简单、系统资源开销小，易于在嵌入式系统的硬件平台上实现的文件系统。

在FAT16文件系统中，它将每个文件分区分为四个组成部分：① 保留区域（Reserved Region）；② FAT表区（FAT Region）；③ 根目录区（Root Directory Region）；④ 文件、目录数据区（File and Directory Data Region）。相对应前三个部分，FAT16文件系统中定义了三个非常重要的数据结构，它们是：① BPB数据结构；② FAT数据结构；③ 目录项。

BPB（BIOS Parameter Block）数据结构位于文件分区保留区的第一扇区，这个扇区被称为“引导扇区”或“保留扇区”或“零扇区”，它由高级格式化程序产生。BPB顾名思义就是“基本输入输出参数块”，这个数据结构中包含了对文件分区描述的基本信息，非常重要的有：每扇区的字节数、每簇的扇区数、FAT表数目、“目录项”数、文件分区中的扇区数、每个FAT表占用的扇区数。

FAT（File Allocation Table）数据结构位于文件分区FAT表区，FAT表区紧跟保留区域之后，出于数据安全考虑，FAT表区一般有两个FAT表，第二个FAT表为第一FAT表的简单备份。FAT数据结构充分体现了FAT文件系统的文件索引机制，它的核心思想是这样的：FAT文件系统为每个文件建立一个“索引链”，“链”上的每个元素是文件数据在数据区中所占“簇”的“簇号”。通过某个文件的目录项，该文件在数据区中的起始簇号可以很容易得到，由此可以在数据区的

?1994-2016 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

这个簇读取到相应的数据。然后再查找到 FAT 表的这个簇号单元，如果内容是文件结束标志（FF），则表示该文件结束，否则该单元的内容指的是该文件保存数据的下一个簇的簇号，这样重复索引直到找到文件结束，流程如图 1 所示。

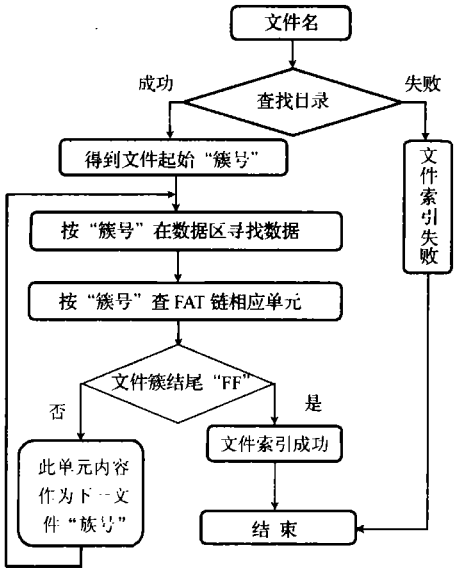

图 1 FAT 文件系统中文件索引流程

目录项位于文件分区的根目录区，每个目录项固定地占有 32 个字节，它的数据结构定义因为文件名类型的不同而不同。对于以传统的"8.3"格式的命名的文件，有且仅有一个目录项，其中记录了文件名、文件属性、以及该文件在数据区中的起始位置等信息。而对于以非"8.3"格式的文件名命名的文件来说，它的目录项分为"缩略文件名目录项"和"长文件名目录项"两部分。在"缩略文件名目录项"中，它记录了缩略文件名、文件属性、以及该文件在数据区中的起始位置等信息；在"长文件名目录项"它以 Unicode 码的形式记录了文件的全名，以及一些校验信息。

目录项是十分重要的一个数据结构，因为只有 FAT 还不能定位文件在存储介质中的位置，FAT 还必须和目录项配合才能准确定位文件的位置。无论哪种形式下的目录项，它都记录着每个文件的起始位置。当定位某一文件位置时，需要将目录项中该文件在数据区的起始地址，结合 FAT 表就可以知道文件在存储介质中的确切位置了。在根目录区之后，才是真正意义上的数据存储区，即 DATA 区。

DATA 虽然占据了绝大部分存储空间，但没有了前面的各部分，它对于用户来说，也只能是一些枯燥的二进制代码，没有任何意义。系统是没有办法利用其中的有效数据的。

为了有效管理每个文件分区上的 BPB、FAT、目录项等，为每个文件分区建立了一个专有的数据结构，它包括分区名称、引导区的首地址、BPB 表的结构、FAT 表指针、目录项表的结构、数据区的扇区号、OPERATION（驱动的结构）等成员变量。通过它，可以很容易地访问某个文件分区中的所有数据结构及参数。

3 FAT16 文件系统在 μC-OS 中的 API 接口

在 μC-OS 中使用 FAT16 文件系统是通过调用文件系统提供的一系列 API 函数实现的。通过调用这些函数，上层应用程序开发者可以不管底层硬件的具体操作，直接地灵活管理存储设备上的数据文件。在开发的这套文件系统中，API 函数接口参照了 μC/FS、Turbo C 中文件系统接口函数的定义。

函数名称	功能描述
EFS_Init ()	初始化文件系统
EFS_Mount ()	加载文件分区
EFS_FOpen ()	打开或者创建一个文件
EFS_FRead ()	对文件进行读操作
EFS_FWrite ()	对文件进行写操作
EFS_FRemove ()	删除文件
EFS_FClose ()	关闭已经打开的文件
EFS_UnMount ()	卸载已经加载的文件分区

使用这套文件系统，需要在相关的 Task（任务）里调用它的 API 函数。调用的流程、方法分为"读写操作"和"删除文件"两个流程分支，这两个分支在具体操作时略有差异。

4 文件系统的简易移植性

在不同的应用场合，物理存储介质是不同的。为了使文件系统在各种不同的存储介质上能够方便的使用，设计者通过函数指针来解决移植性的问题。具体实现如下：

```
partition. operation->type=DEVICE_UDISC;
partition. operation->readblock=UDiscReadBlock;
partition. operation->writeblock=UDiscReadBlock;
```

对于不同介质的读写操作，用户只要把执行读写操作的函数名指向 Partition 结构中的读写函数项即可。也就是说，文件系统移植过程中只需要根据存储介质的不同改写相应的底层读写函数，不需要更改文件系统的任何 API 函数，也不需要改变文件系统的应用方式，这使得文件系统具有很简易的移植性。

参考文献

1 陈智育. 嵌入式系统中的 FALSH 文件系统. 单片机与嵌入式系统应用，2002 年 2 月
2 Microsoft Crop. FAT32 File System Specification. 2000 年
3 宋群生、宋亚琼. 硬盘扇区读写技术——修复硬盘与恢复文件. 机械工业出版社，2004 年 5 月

?1994-2016 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net