

# 基于 CK510 和 uC/FS 的 SD 卡文件系统的设计

徐惜琼 竺红卫

(浙江大学超大规模集成电路设计研究所 浙江 杭州 310027)

**摘要** 针对智能设备数据记录系统中数据容量大和掉电不保护的问题,提出一种在  $\mu\text{C}/\text{OS-II}$  操作系统和 uC/FS 文件系统下,利用基于国产 32 位处理器核 CK510 的 MCU 芯片而改造的具有日志结构的文件系统。该文件系统通过顺序记录数据以及将管理信息的改写和存储同时进行的方法,有效节省了 SD 卡的数据存储空间,提高了存储的稳定性。

**关键词** CK510 uC/FS SD 卡

中图分类号 TP368.2 文献标识码 A

## DESIGNING CK510 AND uC/FS-BASED FILE SYSTEM OF SD CARD

Xu Xiqiong Zhu Hongwei

(Institute of Very Large Scale Integrate Circuit Design, Zhejiang University, Hangzhou 310027, Zhejiang, China)

**Abstract** Aiming at the problems of smart device data recording system in large data storage and no protection at power failures, a file system with log structure is proposed, which is recast using MCU chip of 32-bit home-made processor kernel CK510 under the operational system  $\mu\text{C}/\text{OS-II}$  and the file system uC/FS. This new file system effectively saves data storage space of SD card and improves the stability of storage by the method of consecutively recording data and concurrently rewriting and storing management information.

**Keywords** CK510 uC/FS SD card

## 0 引言

近年来,由于绿色节能的需要,智能设备高速发展并广泛地运用于生活和工业领域。这些设备产生的数据量大,并且要求数据能被实时、快速、可靠地采集并存储在存储设备中,用作后续的分析与使用,因此要求存储介质具有存储容量大、体积小、可靠性高、存储速度快、通用性好等特点。

移动存储器因其携带方便而受到广泛应用,目前主要的移动存储器有 MMC 卡、U 盘、SD 卡、CF 卡等。MMC 卡是快速闪存存储卡,可反复读写。U 盘采用 USB 接口技术与设备连接,具有即插即用的功能,但不能用于智能手机等智能设备作为存储卡使用。CF 卡存储容量大,成本低,兼容性好,缺点是体积大。SD 卡存储容量大,与 PC 机兼容,体积小,存储性能高,价格实惠,读写速度比 MMC 卡更快一些,同时安全性也更高。综合考虑 SD 卡的众多优点及应用范围,本设计选用 SD 卡作为存储介质。

## 1 SD 卡存储器的特点

SD 卡是以 NAND Flash 为存储介质的存储器,目前最大的存储容量已达 8GB。SD 卡以块为单位进行读写,每个块 512 字节。NAND Flash 的特点是擦除次数有限,一般是 1 000 000 次,当某块执行过度的擦除操作后,这一块的存储空间将会变为“只读”状态,不能再写进数据<sup>[1]</sup>。SD 卡内部所带的接口控制器使得它具有识别坏块的功能,当写数据过程中遇到某一个坏

块时,只要存在有效的空白区域,SD 卡控制器会自动将数据写到空白区域中,大大减轻主机对存储器管理的负担<sup>[2]</sup>。

## 2 SD 卡文件系统的分析

当前操作系统常见的有 WinCE、 $\mu\text{CLinux}$ 、 $\mu\text{C}/\text{OS-II}$  等,其中  $\mu\text{CLinux}$  和  $\mu\text{C}/\text{OS-II}$  因为源码公开而被广泛采用。相比  $\mu\text{CLinux}$ , $\mu\text{C}/\text{OS-II}$  因其结构更加简单,容量小,方便移植而被本设计采用。 $\mu\text{C}/\text{OS-II}$  本身不提供文件系统<sup>[3]</sup>,因而选取一个与  $\mu\text{C}/\text{OS-II}$  有很好兼容性的文件系统非常重要。

目前嵌入式设备针对 Flash 设计的文件系统主要有 3 种<sup>[4]</sup>: TrueFFS、JFFS 以及 YAFFS。这三种文件系统都引入了日志结构的思想。其中 TrueFFS 利用块映射的翻译系统实现非绝对损耗均衡技术,保证所有空间的使用近似平等而不让某一单元过度使用<sup>[5]</sup>,但 TrueFFS 代码非公开,尚不能直接应用;JFFS 将文件系统的数据和元数据以节点的形式存储在闪存上并通过节点读取文件内容,提供了写均衡、垃圾收集等底层操作,具有断电保护功能,缺点是主要用于 NOR Flash,对于以 NAND Flash 为存储介质的 SD 卡,JFFS2 文件系统不很稳定<sup>[6]</sup>;YAFFS 是唯一一个专门针对 NAND Flash 设计的文件系统,和 JFFS 功能类似,且速度更快,占用内存更少,唯一的不足是与  $\mu\text{C}/\text{OS-II}$  的兼容性不是最好。

收稿日期: 2011-01-20。国家高技术研究发展计划(2009AA

011706) 徐惜琼,硕士生,主研领域:嵌入式软件设计。E-mail: xuxiqiong@zhu.edu.cn

uC/FS 与  $\mu$ C/OS-II 是同一个公司出品的系统,具有很强的兼容性。uC/FS 只支持 FAT 文件系统<sup>[7]</sup>,其中 FAT32 文件系统较 FAT12 和 FAT16 文件系统相比具有更大的管理空间,更适用于 SD 卡存储,因此本设计将以 FAT32 文件系统为例,对 SD 卡存储特点进行分析,设计具有日志结构的文件系统。

FAT32 文件系统一共分为以下 4 个部分<sup>[8]</sup>:

1) 系统记录区 SR 该记录区存放最重要的文件系统信息,如主引导程序、扇区字节数以及 SD 卡划分为多少个簇、每个簇包括的扇区数、FAT 扇区数、根目录项数等。

2) 文件分配表 FAT 存放 SD 卡上已占用空间和空闲空间的情况,以及文件存储所占簇的链接情况。为了防止文件分配表损坏而导致文件丢失,一般保存两个相同的文件分配表 FAT1 和 FAT2。

3) 文件登记表 FRT 存放 SD 卡中每一个文件的代号、长度、属性、目录、生成和修改的时间、以及文件在数据区中的首簇号。

4) 数据区 用于存放记录的数据。本设计中,文件系统以扇区为单位对数据进行存储。

uC/FS 里通用的 FAT 文件系统并不适合直接移植到 SD 卡存储区中,主要有 2 个原因:

1) 文件系统以簇为单位存储<sup>[6]</sup>,簇的大小固定为  $2^n$  个扇区,假设要存储的数据容量为  $C$ , $C$  的大小满足以下公式:

$$m \times 2^n < C < (m+1) \times 2^n \quad (m > 0, n > 0)$$

其中  $m$  代表一个簇的容量,那么大小为  $(m+1) \times 2^n - C$  的存储空间将会无法被利用,这对于存储空间有限的 SD 卡来说是极大的浪费。

2) 嵌入式系统的应用条件恶劣,电源电压不稳定,突发性断电以及非法插拔都将对 SD 卡的存储造成灾难性的影响,而通用 FAT 文件系统对于断电保护的设计考虑不足。

如果可以将 SD 卡的数据区划分为数据记录区和数据索引信息区,记录过程直接将数据存储在 SD 卡的数据记录区,那么记录结束后只需要在数据索引信息区添加相应的索引信息就实现了数据的存储。这样,数据的存储过程只需对数据区进行操作而不再涉及 FAT 表和 FRT 表的修改,使这两个表所在的块不会因为频繁使用而坏掉。同时用连续的存储格式存储数据,使存储空间最大化地被利用而不致浪费。这个设计思路是实现数据存储的一个合适的方法。文件系统设计思路可以分为以下 3 个步骤:① 建立文件系统的结构;② 顺序记录数据到数据记录区;③ 记录结束后添加索引信息到数据索引信息区。

### 3 针对智能设备的文件系统的实现

根据上述设计思路,介绍应用于智能设备数据记录的日志结构文件系统的实现流程。

#### 3.1 数据区的划分

数据区是物理上连续的磁盘空间,记录的数据以及对数据的索引信息都将存放在这个区域。FAT 的管理结构由 FAT1、FAT2、FRT 和数据索引信息四部分构成。FAT1、FAT2 和 FRT 为文件索引信息区,其中记录的每一条数据描述了文件到数据的映射,根据文件索引信息区的信息可以在数据区中找到相应的文件。但是如果当前只建立一个文件,并将所有的数据都存储到这个文件里,那么文件索引信息区只记录这个文件的属性,无法对文件里的每条数据进行管理,因此有必要建立一个数据

索引信息区用来对文件里的每条数据进行管理。将数据区划分为两个连续空间(如图 1 所示):数据索引信息区和数据记录区。数据索引信息区用来存储数据的索引信息,这些索引信息和数据记录区的数据成映射关系;数据记录区用来存储实际的数据,其中每一段数据都对应着数据索引信息区的一段数据,根据数据索引信息区的信息可以在数据记录区中找到相应的数据。因此文件索引信息区和数据区中的数据索引信息区一起组成了文件系统中的管理信息区;数据区中的数据记录区作为顺序的存储空间用于存储智能设备的数据。

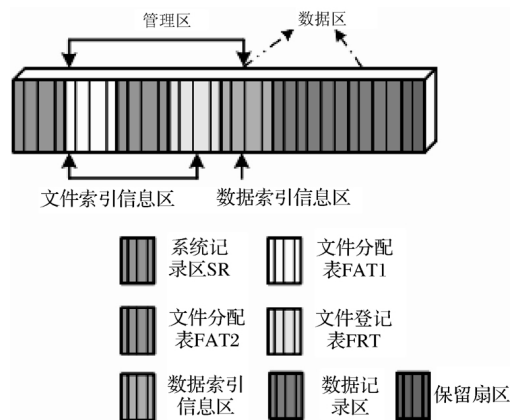


图1 SD卡文件系统结构

#### 3.2 数据的记录

记录开始前先在数据记录区里建立一个数据记录文件,用来存储记录的数据,对应空间为整个数据记录区。

所有记录的数据都写到数据记录文件里,因此 FAT 表和 FRT 表只在创建文件的时候改写了一次,此后不做任何修改,数据写到文件前,先要计算存储起始地址,定义 offset 为偏移量,如果文件为空,存储地址就等于起始地址,即 offset 为 0x00,如果不为空,新的存储起始地址需要紧接着上次记录数据的存储结束地址。这样,数据记录过程将按照时间顺序连续的存储数据。为了能重复利用 SD 卡减小成本并保证数据存储的连续性,文件系统将实行“覆盖写数据”策略,即当数据记录文件已被写满时,文件系统将回到数据记录文件的开头重新写数据,此时文件系统先检测当前指针所指位置的数据的存储时间,当时间小于预先设定的阈值时,这段数据将被新数据覆盖。见图 2 所示,数据记录文件就像一个循环的队列一样,由于数据存储格式是按照时间顺序存储的,最新的数据将会把最早的数据覆盖掉。

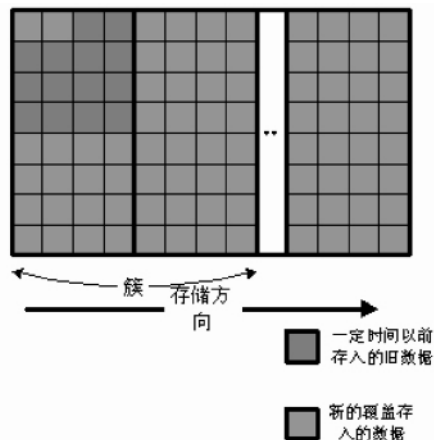


图2 数据记录文件的数据存储

### 3.3 索引信息的添加

根据预先设计的文件系统结构,记录数据结束后需要在数据索引文件里添加新记录数据的信息。FAT 表和 FRT 表记录的是文件的索引信息,对于将数据都记录到一个文件里,应该也要对文件内部数据建立索引信息以便对数据进行查找。记录开始前先在数据索引信息区建立一个数据索引文件,数据索引文件的大小必须要小于数据索引信息区,目的是为了当数据索引文件所占某个块由于频繁擦写而损坏时能有足够的剩余空间让 SD 卡将数据写到空白区域里,使得管理信息不会丢失。数据索引文件里存放着数据记录文件里每一条数据的信息,这些信息以结构体定义,包括数据存储的时间、数据存储的起始位置即偏移量 offset、还有数据的长度。用户根据时间在索引文件里查找数据存放的首地址和数据的长度,然后到数据记录文件里读取所需的数据,如图 3 所示。数据索引文件就像一个简单的二维表对数据管理信息进行存放,每个管理信息的存放空间固定为 10 个字节。当数据索引文件空间不足时,数据索引文件和数据记录文件一样采用“覆盖写数据”策略将旧的数据替换掉。

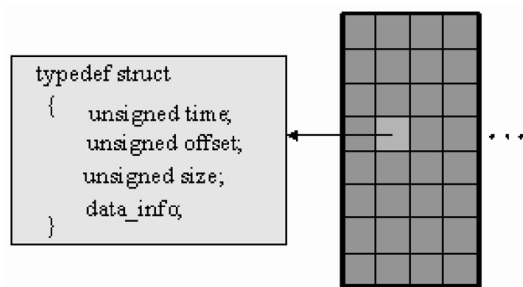


图3 数据索引文件结构

### 3.4 断电保护

为了避免突然掉电而使管理信息丢失,文件在将管理信息添加到数据索引文件时,先将整个数据索引文件里的数据复制到控制终端的内存中(这样做是因为在内存中可以对块内的数据进行修改和添加),在内存中添加新的管理信息的同时,也将文件的改写情况存储在数据索引信息区里的空白区域中。这样在任何时候发生突然掉电或非法插拨而导致内存中数据索引文件丢失的情况下,也能根据临时添加到空白区域中的内容恢复数据索引文件,提高文件系统的可靠性。

根据 FAT32 文件系统的性质,一般 1 簇 = 32 个扇区 = 16kB。假设要存储 3 个大小分别为 5kB、43kB、17kB 的文件,若采用传统的文件系统存储方法存储,占用的空间为 16kB + 48kB + 32kB = 96kB;采用本设计的存储方法存储,占用 SD 卡的存储空间为 5kB + 43kB + 17kB = 65kB。可见,本设计的存储方法节省了 SD 卡的存储空间,提高了 SD 卡的掉电保护性,而且节省的存储空间会随着文件存储数量增大而增大。

## 4 结 语

本文针对当前应用广泛的智能设备中大容量数据存储问题,提出了基于文件系统的数据存储设计方法。该设计方法充分地利用了 SD 卡的存储空间;确保在发生掉电或非法插拨的情况时仍然可以根据空白区域中修改过的内容重新建立新的数据信息索引文件,使文件系统不至于遭到致命的破坏。实

践证明,本方法有效地提高了 SD 的存储使用量,确保了文件系统的安全性和可靠性,优化了数据存储系统的整体性能。

### 参 考 文 献

- [1] 阎航. FAT 文件系统在 NAND Flash 存储器上的改进设计[J]. 单片机与嵌入式系统应用, 2006(11): 27-29.
- [2] 周立功, 等. ARM 嵌入式系统软件开发实例(二)[M]. 北京: 北京航空航天大学出版社, 2006.
- [3] 王田苗. 嵌入式系统设计与实例开发——基于 ARM 微处理器与  $\mu\text{C}/\text{OS-II}$  实时操作系统[M]. 北京: 清华大学出版社, 2002.
- [4] 黄文增, 陈俊达, 陈彦胜, 等. 以区块为基础之与非门闪存管理模式[J]. 台北科技大学学报, 2005, 37(2).
- [5] 王江. TrueFFS 原理及其在 CF 卡上的实现[J]. 单片机与嵌入式系统应用, 2005(1): 38-40, 66.
- [6] 杨磊, 方康玲. 基于 ARM 的嵌入式文件系统 YAFFS 的移植[J]. 微计算机信息, 2008, 24(17): 168-169, 208.
- [7] 郝伟, 李敬兆. 基于 uC/FS 的大容量微存储 FAT32 格式的实现与应用[J]. 电脑知识与技术, 2006(11): 113, 186.
- [8] Tanenbaum A S, Woodhull A S. 操作系统: 设计与实现[M]. 2 版. 北京: 电子工业出版社, 1998.

(上接第 57 页)

本文下一步工作重点将从以下两部分进行深入研究: 一是争取在优化计算复杂度和简化安全性模型以及构建分层的签名方案上有所收获; 二是探索构建可抵抗连续泄漏攻击以及抗事后泄漏攻击 IBS 的可行性。

### 参 考 文 献

- [1] Boneh D, Franklin M. Identity Based Encryption from the Weil pairing[C]//Crypto 2001, volume 2139 of LNCS, 2001: 213-229.
- [2] Canetti R, Halevi S, Katz J. A Forward-Secure Public-Key Encryption Scheme[C]//Eurocrypt 2003, volume 2656 of LNCS, 2003: 255-271.
- [3] Boneh D, Boyen X. Efficient Selective-ID Identity Based Encryption without Random Oracles[C]//Eurocrypt 2004, volume 3027 of LNCS, 2004: 223-238.
- [4] Waters B. Efficient Identity-Based Encryption without Random Oracles[C]//Eurocrypt 2005, volume 3494 of LNCS, 2005: 114-127.
- [5] Gentry C. Practical Identity-based Encryption without Random Oracles[C]//EUROCRYPT 2006, LNCS 4404, 2006: 445-464.
- [6] <http://crypto.stanford.edu/pbc/download.html>.
- [7] Boneh D, Franklin M. Identity-based encryption from the weil pairing[J]. SIAM J. Comput, 2003, 32(3): 586-615.
- [8] Waters B. Dual System Encryption: Realizing Fully Secure IBE and HIBE under Simple Assumptions[C]//CRYPTO 2009, LNCS, 2009, 5677: 619-636.
- [9] Dan Boneh, Ben Lynn, Hovav Shacham. Short signatures from the weil pairing[C]//ASIACRYPT 2001: 514-532.
- [10] Paulo S L M Barreto, Benoit Libert, Noel McCullagh, et al. Efficient and provably-secure identity-based signatures and signcryption from bilinear maps[C]//Bimal K Roy. Advances in Cryptology-ASIACRYPT 2005, volume 3788 of Lecture Notes in Computer Science. Springer Verlag, 2005: 515-532.