基于 NAND 型闪存的嵌入式文件系统设计

李庆诚, 孙明达
(南开大学 信息技术科学学院 嵌入式系统与信息安全实验室, 天津 300071)

摘　要: 由于具有高密度和良好的存取速率等特点, NAND 型闪存在嵌入式系统领域中被广泛应用, 但其所固有的擦除 - 写入更新机制以及高坏块率等特性却又成为其在应用中的障碍。提出了一种基于 NAND 型闪存的日志结构嵌入式文件系统, 用于充分利用 NAND 型闪存的优势并克服其缺陷。文件系统在嵌入式 Linux 操作系统中予以实现, 并进行了性能测试。

关键词: NAND 闪存; 日志结构文件系统 嵌入式 Linux
中图法分类号: TP316　　**文献标识码**: A　　**文章编号**: 1001-3695(2006)04-0231-03

Design of NAND Flash Memory-based Embedded File System

LI Qing-cheng, SUN Ming-da
(Lab. of Embedded System & Information Security, College of Information Technical Science, Nankai University, Tianjin 300071, China)

Abstract: Because of high density and high access speed, NAND flash memory has been widely used in the embedded system area, but the inherited erase-write mechanism and high bad-block rate become obstacles during application. To make effective use of the advantages while overcoming its constraints, a new kind of log structured embedded file system for NAND flash memory is presented in the paper. The file system is implemented in embedded Linux and performance evaluation is carried out on it.

Key words: NAND Flash Memory; Log Structured File System; Embedded Linux

随着嵌入式系统在消费类电子产品、数据采集系统以及工业控制系统中的广泛应用, 随着各类系统对于数据存储容量的需求日益提高, 作为嵌入式系统中最常用的存储设备, 闪速存储器的应用也日益广泛。

闪速存储器按其内部结构的不同可分为 NOR 和 NAND 两种, 其中 NAND 型存储器以高密度、大容量、高数据存储速率以及更多的擦除次数等特点, 逐渐成为大容量嵌入式存储设备应用的主流。

对于嵌入式系统中的数据存储而言, 实现数据的高速存储是系统的主要要求, 但由于嵌入式系统的应用环境变化较大, 容易遇到突发情况而导致系统断电, 并且缺少固定的维护机制, 因此需要采取措施以便在遇到突发事件时保证数据存储的完整性和数据间的关联性, 并进一步均衡存储区负载以延长系统寿命。因此基于嵌入式存储器建立文件系统来管理存储区数据便成为一种必然的方式。

NAND 型闪存作为大容量移动存储器中的介质, 其应用日益广泛, 应用环境日益增多, 因此开发基于 NAND 型闪存的文件系统以实现对存储器的良好管理越来越成为一种必需。

1　闪存工作原理与 NAND 型闪存的特性

闪速存储器具有以下特性:

(1) 数据只能写入存储区中值为 1 的位上, 而不能写到为 0 的位上。

(2) 为将存储区中的 0 变为 1, 需要对存储区进行擦除操作。

(3) 擦除操作以擦除块为单位串行进行, 一次只能擦除一个块。

(4) 每个擦除块的可擦除次数有限, 为 10 万次到 100 万次不等。

闪速存储器按其内部存储矩阵结构不同可分为 NOR 和 NAND 两种, 其中 NOR 型闪存采用或非结构栅格存储矩阵实现, 可按字节写入数据, 读取速度快, 但存储密度低, 数据擦除速度慢, 因此常用于代码存储, 如系统启动代码和内核映像等。表 1 为 NOR 与 NAND 型存储器性能比较。

表 1　NOR 与 NAND 型存储器性能比较

类型	NOR	NAND
存储容量	1MB ~16MB	8MB ~512MB
擦除块大小	64KB ~256KB	16KB ~128KB, 每块被分为若干个 512Bytes 的页面, 每页面含有 16Bytes 附加存储区
写入方式	按字节随机写入	以页面为单位写入
读取方式	按字节随机读取	按字节随机读取或页面读取
读取速度	快	慢
写入速度	慢	快
擦除时间	2s ~5s/块	2ms ~5ms/块
坏块比例	低	高, 需要 EDC/ECC 算法

NAND 型闪存采用与非结构的栅格存储矩阵实现, 由于与非结构的存储矩阵位元面积仅为或非结构的四分之一, 且存取操作以页面为基本单位进行, 故存储密度高, 数据写入速度快, 因此适合于数据存储领域。

相比 NOR 型闪存, NAND 型闪存中容易发生位交换现象, 即一个比特位的值由 0 反转为 1 或由 1 反转为 0。如果反转出现在关键文件数据块中将会导致严重后果, 因此在 NAND 型

收稿日期: 2005-01-17; 修返日期: 2005-04-26

存储器在读取时需要加入错误探测/更正(EDC/ECC)算法以实现对位交换的检测与纠正。

2 基于闪存的文件系统类别

目前可用于 NAND 型闪存的文件系统有多种,按其结构主要分为集中索引文件系统和日志结构文件系统。

集中索引文件系统是磁盘存储器中常用的文件系统,包括基于 Windows 的 FAT 系列和基于 Linux 的 Ext 系列。由于闪存设备可以通过使用闪存转译层(FTL)驱动而模拟成与磁盘相似的块设备,故该类文件系统可以直接基于 FTL 在闪存上应用。

集中索引文件系统将存储区的使用信息和文件的索引信息集中存放在存储区的固定位置,例如 FAT 系统中的 FAT 表和 Ext2 系统中的 Inode 区,而其他区域存放纯数据。

在闪存中应用集中索引文件系统会产生很多问题。由于集中索引文件系统均采用覆写形式进行数据更新,而在闪存中覆写数据需要先擦除再写入,如果数据擦除后系统发生意外而断电,被擦除的数据将无法写回到存储区,从而导致数据丢失。由于集中索引信息会被频繁更新,因此在突然断电时最容易出现数据丢失而导致系统无法识别。

其次,由于索引信息集中存放,任何数据的更新都将导致集中索引信息的修改,因此集中索引信息将被频繁修改,由于闪存存储块的擦写次数有限,频繁的修改将导致集中索引区损坏。另外由于存储区的擦除速度较慢,覆写方式将降低数据写入速度。

日志结构文件系统采用了数据库系统中日志的概念,其特点是对数据的更新采用前向写入,即写入空白块而不是覆写方式进行,从而避免了数据块的擦除,因此保证了意外情况下数据存储的完整性与关联性,并实现了存储区的负载均衡。同时由于不直接进行耗时较高的擦除操作,数据的写入速度将较大提高。因此日志结构文件系统更适合于在闪存设备中的应用。

因为采用了前向写入的方式进行数据更新,所以运行一段时间后系统中会出现较多垃圾数据块,因此需要进行垃圾回收以创建空白存储区。

目前基于闪存的日志结构文件系统主要有 JFFS 和 JFFS2。JFFS 系列文件系统是针对 NOR 型闪存设计的,其存储节点长度不固定,加载时需进行全存储区扫描,并会占用大量内存以存储文件系统结构数据,且未能充分利用 NAND 型闪存的页面结构等特性,运行效率较低。因此应该设计并开发一种专门针对 NAND 型闪存,能够充分利用其各种特性,并能提供快加载速度、高存取速率以及低内存损耗的文件系统。

3 NAND 闪存文件系统设计

基于 NAND 闪存的文件系统应充分发挥 NAND 闪存自身的特点,同时克服其固有的缺陷。为此,文件系统从数据存储结构、文件管理机制、系统加载机制以及存储区空闲空间管理和垃圾回收机制等方面均采用了优化的设计。

3.1 数据存储结构

结合 NAND 闪存的页面结构,文件数据被存储为与页面大小相等的数据块。由于系统中没有集中的索引数据区,因此文件相关的数据被存放在各个数据块自身数据中。NAND 闪存为每个页面提供了 16Bytes 的附加存储空间,并可以通过在系统控制予以开启。文件系统充分利用了这个附加空间以存储页面数据相关的文件信息,即元数据。系统参考了 SmartMedia 标准中的设定,使用 16Bytes 中的 6Bytes 存储页面 ECC 校验数据,1Byte 存储页面数据块的块状态字和 1Byte 存储数据状态字,并将其余的 8Bytes 用作存储文件信息的标签区。

标签区 8Bytes(64bits)的具体分配方式为:页面所属文件 ID 20bits,更新序列号 2bits,页面在文件中位置 ID 20bits,页面数据长度 10bits,标签区 ECC 校验 12bits,另外有 2bits 的保留位作为扩展使用。

在系统没有开启附加存储区或文件系统被应用于磁盘等其他存储器时,16Bytes 的附加数据被存放于相应页面的最后 16Bytes,从而实现对于不同类型存储器的兼容。闪存空间物理布局图如图 1 所示。

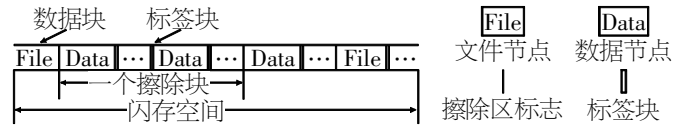

图 1 闪存空间物理布局图

3.2 文件管理机制

系统中设立了 File_Object 结构以标志文件头节点。File_Objec 结构中含有文件的序号 ID、状态标志、所有者 ID、修改记录等信息。结构的长度小于 512Bytes,故可以存放在一个存储区页面中。在设计中将一个文件的所有链接信息存放到了文件的 File_Object 结构中,这样可以直接统计文件被链接的次数,并在系统加载时直接构建文件全部链接信息,避免了对链接信息的多次检索。

由于系统中没有建立集中索引区,所以在系统加载后需要将各目录和文件信息存放于内存中,以便进行数据的实时检索。为此,对文件数据节点的管理采用了多叉树结构。结构中每个节点为 32Bytes,一个节点或为采用 4Bytes 指针,指向 8 个子节点的中间节点,或为采用 2Bytes 指针指向 16 个数据块的页节点。树的结构如图 2 所示。

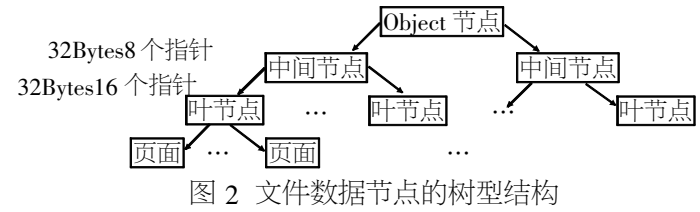

图 2 文件数据节点的树型结构

对于中间节点指针,其中存放的是字节点在内存中的位置,而对于页节点指针,其中存储的是页面在闪存存储区中的偏移量。在一个 128MB 的 NAND 闪存中有 128MB/512B = 256k 个,即 2^{18} 个页节点,而 2Bytes 指针可以指向 2^{16} 个节点,因此通过指针索引可以将其缩小到一个 2^2,即 4 个页面的范围中,在该范围内则可对标签区进行检索以找到文件的具体数据块。

采用上述索引方式的原因是嵌入式系统中的内存容量较小,而闪存的数据检索速度较快,因此可以通过该方式在减少内存用量的同时保证检索速度。对于内存较大的系统则可以将页节点指针设置成 3Bytes 或 4Bytes 大小,从而直接指向对应数据块以提高数据读取速度。

文件系统的数据存储采用了日志结构作为基本结构,对被

修改数据的更新采用前向写入的方式进行,在空闲区中分配一个新的数据块以存储更新后的数据,而将原数据块标志为可回收状态。新数据块中保留原数据块中的文件 ID、数据块序列 ID 等所有元数据内容,从而保证其在文件系统中的相应信息。由于系统在更新过程中不直接清除原数据块数据,因此在出现突发情况而导致系统断电、新数据未能被保存时仍可以通过复制原数据块数据而保证文件数据的完整。图 3 展示了日志结构文件数据更新过程。

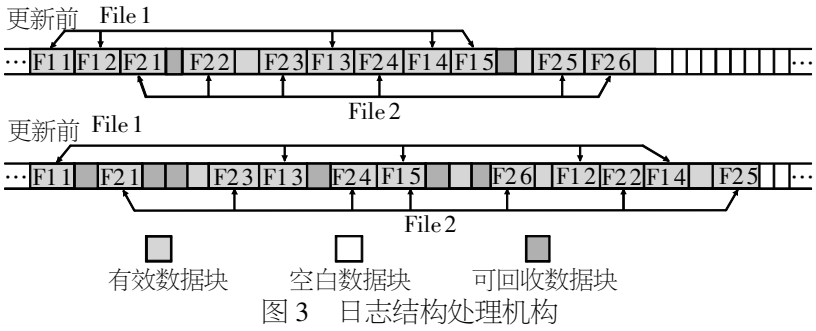

图 3　日志结构处理机构

图 3 为日志结构处理机制。File1 和 File2 中各有若干数据块被更新,系统将更新后的数据写入新数据块,同时将原数据块标志为可回收状态。

由于系统在进行数据更新时突然断电可能会导致原数据块未被标志为可回收状态,从而出现数据块的重复,为此系统在每次进行更新时将数据块标签区的序列号加 1,并在发现数据块相同时取序列号较大者,同时将较小者标志为可回收状态。

3.3　擦除块管理与坏块管理

由于闪存以擦除块为单位进行数据块的回收,因此提供良好的擦除管理机制将会较大地提升系统性能。系统中设立了 BlockState 结构用于标志擦除块的当前状态,可以存在的状态包括空白、已满和正在当前分配等。同时设立了 BlockInfo 结构用于实现对擦除块的管理。结构中包含了擦除块在系统中的序号 ID、块中已使用的页面数目、块的状态标志以及坏块标志。

系统中采用数组方式管理各擦除块,随着页面的分配与回收,数组中各擦除块项的状态值也将进行改变。系统在写入数据时采用“当前块”的概念,即指定一个擦除块为当前块,所有数据块的分配均在该块内进行,并在完全分配该块后再设置另一个新块为当前块,通过这种方式可以实现对存储区的充分利用。在进行数据块的读写时,系统会结合存放在标签区的 ECC 数据对页面的数据进行检测,当发现数据错误时将采用 ECC 算法对其进行校验,如果多次校验后仍然出错,则停止对其的使用,并将其标志为损坏页面。

3.4　系统加载机制

由于系统中不含有集中索引区,故无法在加载时直接查找系统信息并建立目录结构,因此需要在加载时进行存储区扫描,从而初始化系统信息并构建文件系统结构。

由于在设计中将元数据信息写入了数据块的标签区中,故在扫描存储区时只需读取标签区的内容,而不需要读取全部数据块数据,这种方式大大地加快了系统的加载速度。

为记录存储区的管理信息,系统中设计了 Dev_struct 结构,该结构包含了存储区大小、页面个数、擦除块个数、目录树根节点以及系统缓冲区节点等全部信息。加载过程中将检测存储区的全部状态,并将状态信息填充到该结构中,在随后的运行过程中将修改其中的当前状态值。

图 4 为系统的总体挂接图。

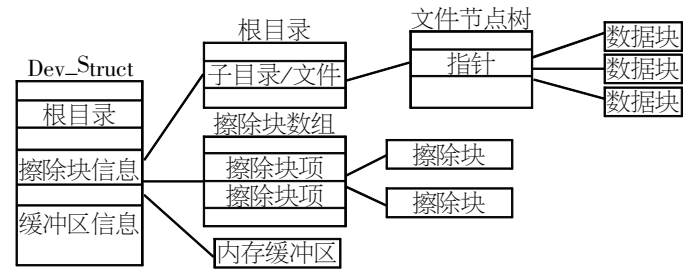

图 4　系统总体挂接图

3.5　垃圾块回收机制

由于采用了日志结构进行数据管理,因此系统在经过长时间运行后将出现较多垃圾数据块,故需采用垃圾回收机制对其进行回收。由于文件系统处于操作系统的内核空间,因此垃圾回收算法的优化程度将对系统的性能产生较大影响。

系统中设计了综合的垃圾回收策略,并建立了两个垃圾回收线程以实现不同情况下不同策略的垃圾回收机制。第一个线程具有系统最低优先级,并在系统空闲时运行。运行时基于擦除块的使用状况,采用最优适应算法,在存储区中选取垃圾页面最多的擦除块进行清除。第二个线程则在系统负载较重,对存储区有较大需求时由文件系统创建并运行,它具有高优先级,基于最先适应算法进行垃圾回收,以提高执行速度。

进行垃圾回收时系统需要保留若干空间以进行页面的转储,因此当系统负载较重,系统中的空闲块数小于既定数目时将引发第二个垃圾回收线程以实现垃圾回收。

4　性能模拟测评

文件系统基于 Linux 操作系统予以实现,并在系统的内存中进行了性能模拟测试。测试中采用了 4MB 内存模拟闪存,页面大小为 512B,擦除块大小为 16KB,每个擦除块含有 32 个页面,故共有 256 个擦除块及 8 192 个页面。在测试过程中主要进行了负载均衡性与数据块擦除次数等方面的测试,同时与 FAT16, Ext2 文件系统的模拟测试进行了数据对比。

4.1　测试方法与运算公式

测试中采用了文件覆写的方式以测试系统的负载均衡性与擦除率。首先在存储区中写入一个大小为 1MB 的文件,随后覆写该文件的全部数据,并记录覆写过程中更新的页面数目以及执行的擦除次数。测试中主要测试了系统的存储区负载系数和擦除块擦除系数。

存储区负载系数 μ_{load} 用于标志系统的页面更新负载率,计算公式为 $\mu_{load} = P_{upgrade} / P_{write}$。其中更新总页面数 $P_{upgrade}$ 为覆写操作过程中更新过的全部页面数,数据写入页面数 P_{write} 为存储过文件数据的页面总数。

擦除块擦除系数 μ_{erase} 用于标志进行数据更新时系统的擦除负载率,计算公式为 $\mu_{erase} = E_{times} / B_{total}$。其中更新总擦除次数 E_{time} 为在更新过程中所执行的总擦除次数,系统擦除块数 B_{total} 为存储区中擦除块总数,在本测试环境中为 256。

4.2　测试结果与性能分析

图 5 为存储区负载系数测试结果图。

从测试的结果可以看出,闪存文件系统的存储 (下转第 239 页)

上述参数必须保证电机在最大和最小转速时刻都能够输出足够的扭矩。另外该子程序还包括一个可选参数 JERG_TIME。默认情况下此参数为 ACCEL_TIME 的 40%，选择此参数可以将上述脉冲输出过程转变为无频率突变的过程，该种控制方法也称为 S-曲线法。参数与电机动作的关系如图 3 所示。

采用 S-曲线法可以进一步改善电机动作的稳定性。

(2) 反馈控制子程序

反馈控制信号由两部分组成，即光电编码器信号和参考定位信号。在具体设计中利用 PLC 的高速计数功能(正交计数)完成对光电编码器信号的计数。参考定位信号代表一组预先设定好的系统绝对位置，如运动的起始点/终止点，速度的最大值点/最小值点等。这些信号可以通过各种传感装置，如霍尔器件等来获得。经过综合各方面设计考虑，最终采用一组霍尔器件标定自动进样器系统执行装置的绝对位置，作为电机位置的状态参量。程序中分别使用状态变量 RP_1 ～RP_5 对应相应的霍尔器件。其中 RP_1 和 RP_5 还代表电机运动位置的上界(LIM+)和下界(LIM-)。

反馈控制子程序则根据系统的反馈信号正确指引电机动作到位。其在 LIM+和 LIM-处的控制过程分别如图 4 所示。

上述电机的控制也可以采用西门子公司的专用扩展模块实现。

(3) 外部事件响应子程序

自动进样器系统的设计需要根据具体的应用需求进行专门设计，限于篇幅在此不再赘述。

为了满足控制精度要求，在系统的设计中采用了反馈控制，反馈信号包括光电编码器的信号和微动开关的信号。

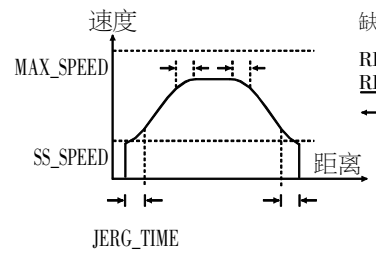

图 3　参数与电机动作

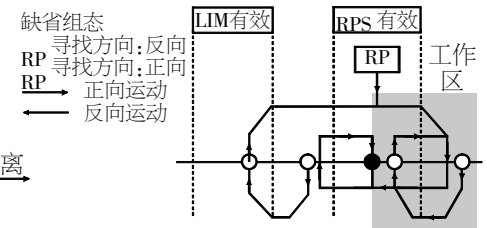

图 4　控制过程

4　讨论

本文给出了基于嵌入式 PLC 的气相色谱液体自动进样器系统功能实现的设计方案，分析了自动进样器系统的基本工作原理和程序设计的流程。其中重点是上下位机通信协议的设计和实现以及多个步进电机的精确协同控制。目前，研制的气相色谱 12 样品位的液体自动进样系统在美国 HP-5890 气相色谱仪上得到了成功的应用，取得了良好的效果。

参考文献:

[1] 张建文，徐琼，冯林. PLC 控制系统工作方式的分析和研究[J]. 华东地质大学学报，2003，26(3): 298-300.

[2] 刘晓玲，方彦军，李京丽. 基于 Modbus 的 PLC 与多台电量智能仪表的通信研究[J]. 工业仪表与自动化装置，2003，(1): 24-25.

[3] 西门子(中国)有限公司. S7-200 可编程序控制器系统手册[R].

作者简介:

梁作宇，男，硕士；张延华(1960-)，男，陕西陇县人，教授，硕士，主要研究方向为无线通信、视频传输可视化仿真技术；姜振宇，男，硕士。

(上接第 233 页)区负载率明显低于 Ext2 和 FAT16 文件系统，其原因是闪存文件系统采用了日志结构的数据更新方式，对数据的更新采用前向写入而不是 Ext2 和 FAT16 所采用的覆写方式，故在进行数据更新时可以均匀地利用存储区，从而降低系统负载率。

图 6 为擦除块擦除系数测试结果图。

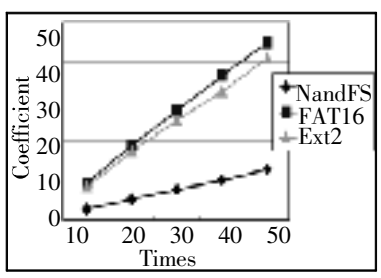

图 5　存储区负载系数测试数据图

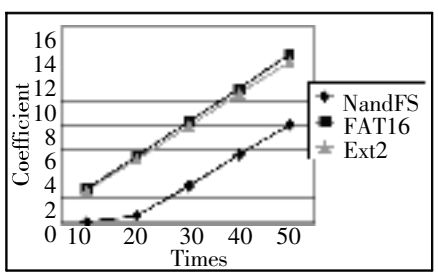

图 6　存储区擦除块系数测试数据图

从测试结果中可以看出，每当进行数据更新时，Ext2 和 FAT16 都会直接执行擦除操作，因此其擦除系数与更新次数为线性关系。而对于闪存文件系统，当更新数据小于存储区容量时，闪存文件系统不会进行垃圾回收，即不会擦除数据块，只有当进行多次数据更新后，存储区没有足够的空闲空间时，系统才进行数据块擦除，并且系统的擦除率低于 Ext2 和 FAT16，因而节省擦除时间并实现了对存储区数据的保护。

5　结论

NAND 型闪存以其存储容量大、数据存取速度快等特点在嵌入式系统领域被广泛应用，但作为闪速存储器它也具有一些固有的缺陷。为实现在 NAND 型闪存上进行良好的数据管理机制，本文提出了基于 NAND 型闪存的采用日志结构的嵌入式文件系统。经测试表明，这种文件系统充分利用了 NAND 型闪存的特点，克服 JFFS, FAT 以及 Ext2 等文件系统的缺陷，为 NAND 闪存提供了良好的文件读写机制、负载均衡性机制及数据完整性保护机制，提高了系统加载速度，并降低了内存占用，实现了对数据的高效管理。

参考文献:

[1] 陈育智. 嵌入式系统中的 Flash 文件系统[J]. 单片机与嵌入式系统应用，2002，(2): 5-8.

[2] 须文波，张雁. 应用闪速存储器实现盘文件系统探讨[J]. 计算机研究与发展，1997，34(10): 189-192.

[3] 顾宝根，顾喜梅. 日志结构的嵌入式文件系统研究[J]. 计算机工程与设计，2004，25(6): 915-917.

[4] Woodhouse D. JFFS: The Journalling Flash File System [EB/OL]. http://sources.redhat.com/jffs2, 2002.

[5] Rosenblum M. The Design and Implementation of a Log-Structured File System[J]. Transactions on Computer Systems, 1992, 10(2): 26-52.

作者简介:

李庆诚(1964-)，男，河北唐山人，教授，博士生导师，主要研究方向为嵌入式系统与信息安全；孙明达(1981-)，男，河北承德人，硕士研究生，主要研究方向为嵌入式操作系统。