

Flash 文件系统分析及其在 VxWorks 中的实现^①

Analysis of Flash File System and Its Implementation in VxWorks

李 勇 刘宏立 (湖南大学 电气与信息工程学院 湖南 长沙 410082)

摘要: 简要分析了 Flash 存储介质的特点,在此基础上阐述了传统文件系统应用在 Flash 上的局限性,以及现有的解决方案。重点探讨了为 Flash 设计的文件系统 TrueFFS 的运行机制,并给出其在 VxWorks 中的具体实现方式。测试证明 TrueFFS 文件系统有良好的性能,提高了 Flash 设备的使用寿命。

关键词: Flash 文件系统 VxWorks

1 引言

当前数据交换量越来越大,移动存储器的需求也越来越大。Flash 作为一种非易失的记忆体具有体积小、密度大、功耗小、抗震性好等特点,逐渐成为大容量嵌入式设备应用的主流。

随着数据量的增大,Flash 需要文件系统对其进行管理,为用户提供方便快捷的操作方式。传统的文件系统是针对磁质介质的存储器来设计的,而 Flash 在操作方法以及特性方面与磁质介质有很多不同。如果简单的将传统文件系统用于 Flash,必然导致系统性能的下降甚至崩溃。针对该问题,本文介绍了一种基于 Flash 的文件系统 TrueFFS,重点分析了该系统的核心思想以及其在实时操作系统 VxWorks 下的实现。

2 Flash 文件系统特点

2.1 传统文件系统在 Flash 上的局限

1) 传统文件系统限制了 Flash 的寿命。传统文件系统通常使用存储介质上的一片固定的区域来存放一些目录和统计信息,如人们常使用的 FAT 文件系统。而 Flash 的擦写次数相对有限,在每次更新数据时都要对这些区域进行操作,因此对目录和统计信息的修改会非常频繁,导致存储这些信息的区域的擦写操作远多于正常的数据区域,加速了 Flash 芯片的老化,过早进入只读状态。

2) 传统文件系统没有考虑 Flash 特有的擦写机

制。NAND Flash 写入数据以页为单位,一页包含 512 字节(小页)或 2048 字节(大页)。修改页面中的数据,直接向该页面执行写操作是不可行的,因为 Flash 中已经使用的页在被擦除之前不能再次写入数据^[1]。为了立即更新相关数据,需要找到一块已经被擦除过的干净空间,然后写入修改后的数据,之前的数据页被视为无效。这种特性被称为区域外更新(Out-of-Place Update)。Flash 中的擦除操作是以块为单位,块中包含多个页。所以使用擦除操作回收页空间,要保证该页所在块不包含有效信息。以上特性要求文件系统拥有更好的数据更新和垃圾回收机制。

2.2 针对 Flash 文件系统的解决方法

针对 Flash 自身特性引出的上述问题,主要有两种思路的解决方案:

1) 在操作系统与 Flash 存储之间增加 FTL(Flash Translation Layer)层。FTL 层的主要任务是隐藏 Flash 与传统设备读写机制的差异,使系统能够像操作磁质设备一样操作 Flash^[2]。

2) 设计新的文件系统。这种新型的文件系统在各项操作上都已经考虑到了 Flash 的各种特性,因此能够直接被系统引用^[3]。

两种系统各有特点。使用新的设计可以最大化系统的性能。目前这些系统还不够完善,例如一些日志型 Flash 文件系统装载时间过长、内存消耗量大。而使用 FTL 层会使原有的文件系统更加复杂,带来一些

^① 基金项目:国家高技术研究发展计划(863)(2007AA1A121)
收稿时间:2008-12-01

性能的损失, 但可以对流行文件系统提供支持。TrueFFS 就是一种使用 FTL 层的 Flash 文件系统, 它由 M-System 公司开发, 支持广泛使用的 DOS 文件系统, 具有良好的兼容性。

3 TrueFFS机制分析

TrueFFS 在 FTL 层中采取了一系列措施来隐藏 Flash 的特性: 使用损耗均衡算法来平衡整个介质的磨损量; 使用块地址映射技术并维护着一个动态的映射表; 提供垃圾回收机制回收含有过期信息的存储单元^[4]。在这些机制下, 上层用户对 Flash 的操作如同读写普通 MS-DOS 文件系统下的磁盘一样方便。

3.1 页面映射机制

TrueFFS 将整个 Flash 空间映射为一片连续的存储页面, 称之为虚拟页, 它与 Flash 中的物理页面有动态的一一对应的关系, 由一个映射表来管理。

数据读的操作是比较直接的。接收到数据读取的请求时, TrueFFS 将请求的页面地址转换为物理页面地址, 然后从 Flash 中读取数据返回给用户。

数据的写操作分两种情况讨论。当目标区域是已擦除的干净页面时, TrueFFS 直接将用户数据写入相应的 Flash 存储中。当操作是修改已经存储于 Flash 中的数据时, 这种情况要复杂一些, 因为 Flash 使用先擦除再写入的机制, 并且擦除操作是以多个页为单位。TrueFFS 使用的方法是, 找到一片可以使用的干净区域, 将修改后的页面数据写入。当所有数据已经安全写入后更新映射表, 使虚拟页映射至含有最新数据的物理页。

3.2 损耗均衡机制

为写入操作分配的物理页面时需要使用到损耗均衡机制, 系统在找到磨损量最小的页面后进行写入。仅仅是在有数据操作时执行损耗均衡是不够的, 在 Flash 中有静态文件(长期存在而又不会被修改的文件)的条件下, 静态文件占用的区域不会参与到损耗均衡中, 因此会造成此外的空间负荷增大, 降低了设备的寿命。而 TrueFFS 会及时更新静态文件在 Flash 中的存储位置, 克服静态文件锁定问题。TrueFFS 还采取了相关措施防止对相同的物理区域长时间连续擦写操作, 避免 Flash 进入过编程状态。

由于采用了这种损耗均衡机制, TrueFFS 文件系统在长期运行后会使得每一个擦除单元拥有相同的擦写

次数, 从而最大化了 Flash 器件的寿命。

3.3 垃圾回收机制

对数据的修改会使某些已经使用过的页面不再包含有效信息, 这些页面在被擦除之前无法再次利用。如果没有一个有效的机制回收无效页面, Flash 会很快进入只读状态。

TrueFFS 使用了垃圾回收机来解决以上问题: 首先将需要回收的块单元中的有效数据拷贝到另一个可用块中, 更新页面映射表, 擦除要回收的块单元。

垃圾回收使用得过于频繁, 会对降低损耗均衡的效能, 所以只在必要时才调用该机制。TrueFFS 总是在同一个块内保持一个拥有连续可用页面的空间, 当这个空间变得太小时, 则使用相关机制找到并回收拥有最多无效页、擦除次数最少、最静态的块空间。

3.4 存储空间分配机制

为了更有效地检索数据, TrueFFS 使用相关存储空间分配机制, 使相关联的数据尽可能的存在于同一个块的连续空间中。TrueFFS 的存储空间分配思想是: 尽可能分配一段位于同一个块内的连续页; 如果不能满足, 则保持分配的页在同一块内; 以上均不能满足时, 保证分配的页位于含有最多可用空间的块内。

这种机制有很多优点: 方便数据的访问; 减少了碎片的产生, 有利于垃圾回收机制的实现; 方便损耗均衡机制中对静态数据的处理。

3.5 错误恢复机制

考虑到可能产生的各种恶劣情况, TrueFFS 有良好的错误恢复机制。它保证了已经正确写入 Flash 的数据的安全。

TrueFFS 的健壮性表现在先更新后擦除的策略上。当需要更新一段数据时, 先将更新后的数据写入新的区域, 然后将旧的区域擦除。如果在写入期间产生错误, 则新写入的区域并不会标记为有效, 继续使用原有数据。当有错误发生时, 保证了数据是最近的一个版本, 只有第一次写入 Flash 的数据才会丢失。写入数据第一次产生错误后, TrueFFS 会另外找一片可写区域重新写入, 而不是马上返回错误。在 Flash 生命周期的后期, 写入操作失败的频率增加, 采用自动重写机制后用户观测到的只是性能的逐渐下降。

映射表是 TrueFFS 中非常重要的数据结构, 它如

果受到破坏或者丢失会损害整个文件系统。TrueFFS 使用存储于 RAM 的映射表,但一些关键的映射信息在 Flash 中有记录。系统在更新映射表时如果出现掉电,则在 RAM 中的映射表会丢失。由于 TrueFFS 在每个物理块的起始位置维护了一些重要的头信息,电源恢复后扫描这些信息就能够重新构建映射表,这样就很好地保障了系统的安全。

4 TrueFFS在VxWorks中的实现

4.1 TrueFFS 结构

VxWorks 是美国风河公司推出的面向嵌入式领域的强实时操作系统。它支持与 MS-DOS 文件系统兼容的 dosFs,并且还对其进行了优化,提高访问效率^[5]。将 dosFs 与 TrueFFS 机制相结合可以为系统提供有效的存储管理。TrueFFS 在 VxWorks 中的结构如图 1 所示。

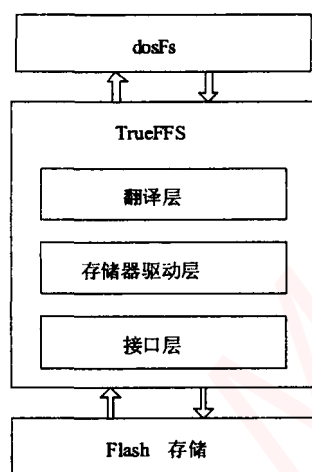


图 1 VxWorks 下 TrueFFS 结构图

翻译层(Translation Layer)是 TrueFFS 的核心。它与 dosFs 交互,隐藏 Flash 与传统存储设备的区别,提供损耗均衡、垃圾回收、错误恢复等机制。

存储器驱动层(Memory Technology Device Layer)提供特定 Flash 存储器件基本操作的实现,如读、写、擦除和映射等。

接口层(Socket Layer)为可拔插的 Flash 存储卡提供支持,可以提供检测设备拔插、管理电源、写保护控制等服务。

4.2 具体实现步骤

为包含 dosFs 文件系统以及 TrueFFS 相关组件,

可以在 config.h 文件中加入如下语句来实现:

```
#define INCLUDE_DOS;
#define INCLUDE_TFFS;
```

通过修改 Makefile 文件,可以将 sysTffs.c 文件以及存储器驱动相关源文件(本文以 NandMTD.c 为例)添加到整个系统中。在 Makefile 中添加如下语句:

```
MACH_EXTRA = sysTffs.o NandMTD.o
```

SysTffs.c 文件中含有接口层的实现函数。每一种 Flash 设备有与之对应的接口层,VxWorks 限定接口数最大为 5 个。利用此文件中的注册函数,可以将接口层的实现函数注册到系统,保存在名为 FLSocket 的结构体中。此外,还需要选择 TrueFFS 中相关的翻译层支持。不同的 Flash 器件之间在操作上也会有差异,因此会选用不同的翻译层。本文选用 Nand Flash 器件的翻译层,所以需要在文件中定义 INCLUDE_TL_NFTL。

NandMTD.c 中含有 Flash 底层的基本操作函数,通过将此文件中的识别函数加入系统的全局变量 mtdTable 中,使得文件系统加载时扫描 mtdTable 中的识别函数,依次调用。识别成功后,注册相关 Flash 操作函数,存入 FLFlash 结构体中。

在有 MMU(Memory Manage Unit)的系统中,VxWorks 采用了虚拟地址映射机制。在 sysLib.c 文件中加入相关的地址映射配置,使系统能够正常的访问 Flash 存储空间。

完成上述工作,编译整个系统,TrueFFS 就已经嵌入到 VxWorks 镜像中去了。系统启动时首先用 tffs Drv()函数初始化相关数据结构、注册接口层驱动;然后使用 tffsDevFormat()函数格式化 Flash;最后使用 tffsConfig()函数,这个函数分别调用 tffsDev Create()与 dosFsDevInit()来创建 TrueFFS 块设备,并在这个块设备上加载 dosFs 文件系统。至此,整个 TrueFFS 文件系统构建完毕。

5 TrueFFS性能测试及分析

将 VxWorks 系统嵌入光电稳定跟踪平台,在用于数据存储的 Flash(K9F2808U0C)中移植 TrueFFS 文件系统。实际运行中,TrueFFS 表现出良好的性能。在存储器驱动层加入相关代码进行统计,观测到 Flash 器件中的每一块的使用量(即磨损量)非常均匀,达到了预期的效果。

Flash 器件的预期寿命由如下公式来评价:

$$L = K \sum_{i=0}^{n-1} C_i E_i \quad (1)$$

式中 L 表示系统写入的总的信息量, 用信息字节数来衡量, 使用它能够准确地评价存储设备的使用寿命; n 为系统存取单元数目; C_i 为第 i 个存取单元大小(KB); E_i 表示第 i 个存取单元的擦除操作次数, Flash 中它的最大预期值为 10 万次; K 表示写入信息占写入字节数的比率, 它的引入是由于系统的开销, 导致某些对 Flash 的写操作并不包含新的数据信息。

在传统的 DOS 文件系统 FAT16 中, C_i 取 2KB, 对应 16MB 的 Flash, n 取 8192。假设每写一个单元更新一次 FAT 表, 则 FAT 表所在区域最先达到寿命 (10 万次擦写) 而进入只读状态, 有如下关系产生:

$$\sum_{i=0}^{n-1} C_i E_i = 10^7 \times 2(KB) \quad (2)$$

在理想的情况下给出 K 值为 1, 并与式(2)一同带入式(1), 得出 FAT16 中的寿命评价为 19GB。

在 TrueFFS 机制下, 由于磨损均衡性, 取 E_i 值 10^7 , C_i 为 2KB, n 等于 8192, K 取实际使用中的统计值 0.68。将以上参数带入式(1), 得出 TrueFFS 下的寿命评价为 103TB。可见, TrueFFS 机制的引入极大地提高了 Flash 的使用寿命。

6 结束语

目前, 嵌入式文件系统的研究已经成为一个热点。本文讨论了 Flash 文件系统 TrueFFS 的设计思想, 并在 VxWorks 下移植完成此文件系统的移植, 验证了 TrueFFS 在 Flash 器件上的适用性。针对 Flash 文件系统的开发, 仍然有许多工作要做: 进一步解决扩展性的问题, 减小设备容量、文件数量对启动时间、内存占用的影响; 解决操作时延不均衡的问题, 使系统更加适用于实时环境。

参考文献

- 1 Samsung Electronics Co. NAND Flash Memory & Smart Media DataBook. 2002. <http://www.samsung.com>.
- 2 Sunhwa Park, Seong-Yong Ohm. New Techniques for Real-Time FAT File System in Mobile Multimedia Devices. IEEE Trans. on Consumer Electronics, 2006, 52(1):1-9.
- 3 Kim J, Jo H, Shim H, et al. Efficient Metadata Management for Flash File Systems. 11th IEEE International Symposium on Object Oriented Real-Time Distributed Computing, 2008:535-540.
- 4 WindRiver Inc. TrueFFS for Tornado Programmer's Guide. 1999. <http://www.windriver.com>.
- 5 周启平, 张扬. VxWorks 下设备驱动程序及 BSP 开发指南. 北京: 中国电力出版社, 2004.