

嵌入式多媒体应用的通用文件系统的分析和设计

晏松, 杨晨晖, 王炳波, 汤志成
(厦门大学 计算机科学系, 福建 厦门 361005)

摘要: 随着嵌入式多媒体技术的日趋发展, 越来越多的多媒体娱乐设施已经用于家用起居和娱乐场所。文中旨在提出和构建一个通用的、移植性强的、针对嵌入式系统上多媒体应用的文件系统来管理多媒体数据, 提高数据的存取效率和存储设备的空间利用率, 从而获得良好的视听效果。通过分析 FAT 文件系统的性能, 以及比较在 PC 机和嵌入式系统上应用的优缺点, 并结合多媒体数据本身的特殊性以及嵌入式的具体应用环境, 对其进行改进, 以实现优越的性能来适应嵌入式多媒体的应用。

关键词: 嵌入式; 文件系统; 多媒体; 文件分配表

中图分类号: TP39

文献标识码: A

文章编号: 1005-3751(2005)05-0068-04

Design and Analysis of General File System in Embedded Multi-media Application

YAN Song, YANG Chen-hui, WANG Bing-bo, TANG Zhi-cheng
(Dept. of Computer Science, Xiamen University, Xiamen 361005, China)

Abstract With the gradual development of multi-media techniques, more and more multimedia facilities are put into use in home appliances and recreational spots. This article intends to propose and construct a general file system to manage multimedia datum to increase the efficiency of access and storage, which is transplantable and steady in embedded systems. After comparing the advantages and the disadvantages in PC systems with those of in embedded systems by analyzing and comparing the performance of FAT file system, the author tries to improve file systems in embedded systems to adapt the tentative multimedia applications with reference to the particularities of multimedia datum and the environments of embedded system.

Key words: embedded; file system; multi-media; FAT

1 嵌入式平台中多媒体文件的存取分析

随着嵌入式技术的日渐成熟, 在许多行业 and 部门都得到了广泛的应用, 尤其在多媒体娱乐方面, 如公交车载、家庭影院、mp3 播放器等。在嵌入式平台^[1]上, 如何高效存取多媒体数据是一个较为重要的课题。随着信息社会的飞速发展, 对视频业务的要求越来越高, 对应地, 数字图像表示需要更大量的数据, 由于存储空间、存储器带宽、传输带宽的限制, 不得不采取高效的编码技术来节省多媒体文件的存储空间。经过压缩的视频、音频等多媒体数据类型与传统的数据类型相比较, 有着一些独特的特性:

* 尽管压缩过, 还占用较大的存储空间。

* 对延迟敏感。必须分配足够的系统资源保证视频数据和音频数据的连续回放, 以达到良好的视听效果。

* 具有弱实时性。一般来说音视频流只要在它们的 deadline 之前到来就可以; 并且 deadline 因媒体流的格式

而异。

另外在嵌入式系统中, 对于文件的存取管理, 在一些性能方面的要求也有别于 PC 机上的文件系统:

a. 嵌入式系统的应用条件远比计算机恶劣, 如电源电压不稳定、突发性断电重启, 以及存储器自身物理性的损伤、异常或错误的写操作, 都有可能造成灾难性的后果; 而通用的 PC 文件系统在这方面的设计考虑欠缺。

b. 通用的 PC 文件系统, 为了提高存取速度, 大多使用缓存技术, 这要耗费相当的内存资源, 而这些额外的资源开销, 对某些嵌入式系统来说, 由于本身的处理器的性能或总线寻址能力的有限性, 是难以企及的。

c. 多媒体数据往往允许一定程度的误码损伤和传输差错, 未必需要如通用文件系统那样严格保证数据的完整性, 可以采用概率校验或随机校验来降低其开销。

d. 稳定性是嵌入式系统需要着重考虑的方面, 对主引导记录、文件分配表等重要的系统信息, 必须存有多个备份和及时更新, 从而保证在某个失效的情况下, 系统能够恢复正常稳定的工作。

另外通用文件系统有着基于对访问权限和安全性的

收稿日期: 2004-10-21

作者简介: 晏松(1979-), 男, 江西靖安人, 硕士研究生, 研究方向嵌入式多媒体应用。

设计, 而嵌入式系统的多媒体应用, 人们更多地关注多媒体节目的播放顺序、定时的管理, 所以完全没必要考虑用户访问某一个文件或目录的权限大小以及密钥算法; 再则通用的 PC 文件系统一般都设计了多级指针控制文件的访问, 而在嵌入式系统的多媒体应用中, 其主要是针对音视频的文件类型及其回放, 整个存储设备上需要存放的文件个数相对比较少, 可以通过只使用一级指针, 直接指向文件的方法来提高存取速度。

针对上述的分析, 不难得出, 针对嵌入式系统的多媒体应用的文件系统应该满足如下特性:

- * 适合音视频的存取;
- * 适合工作在电压不稳定、震动较大等较为恶劣的环境;
- * 高效的存取速度。

另外需要增补与修改一些和多媒体应用相关的数据格式, 比如说针对车载娱乐的播放列表、定时管理、节目同步等。

2 FAT 文件系统的组织结构的分析

磁盘的逻辑结构^[2,3]主要包括了引导区(BOOT 区)、文件分配表(FAT)、根目录表和数据区。引导区和文件分配表区又合称为系统区。

(1) 引导区: 定义了相关地址参数分配表, 包括每扇区的字节数、每簇的扇区数、保留的扇区数、每磁道的扇区数、文件分配表的个数和占用扇区的个数、磁头数、卷标等重要参数和引导记录。

(2) 文件分配表: 包含了已分配和未分配, 以及损坏的簇的信息和实现对空闲空间的管理。一般用零值表示未分配的簇, 已分配的簇用非零值表示, 坏簇可以用特定的值来标记。根据簇来对数据区的存储空间进行划分和管理, 包括空间的分配和回收。

(3) 根目录表: 它可以作为系统区的一部分, 分配固定的区域和固定的大小; 也可以看作是数据区的组成部分, 因为根目录可以看作是根目录文件, 采取子目录文件相同的管理模式。文件是以单项簇链表的形式存储, 每个文件在文件目录表中对应着一个文件目录项, 文件的起始簇号存放在其对应的目录项中, 一个文件目录项占用 32 个字节, 相关的字节定义了文件名、文件属性、文件的大小、文件的创建或修改时间等等。

3 嵌入式系统多媒体文件系统的主要设计思想

嵌入式系统相对比较通用的计算机系统来说^[4], 往往是针对特定的应用领域, 在硬件设计方面也作了相应的裁减和规划, 使嵌入式系统尽可能保持最低的系统资源配置, 力求将整个系统的成本降至最低。为此, 应该适当地添加一些增强系统功能的模块, 提高系统的性能和效率, 另外也有必要降低其对系统资源的消耗, 保障一定的系统

资源实现一定的可扩展性以适用以后的需求。参照于 FAT 文件系统的相关性能, 同时根据图 1 ~ 图 3 所示的文件系统细节结构, 重点在以下几个方面作了相关的改进或重新设计, 具体描述如下:

a. 以重新设计簇的表示方式来减小文件系统的分区表的尺寸, 节省对内存的消耗。通过采用按位来表示簇占用情况的方式设计 FAT 表: 一个字节可以表示 8 个簇, 相比 4 个字节表示一个簇的 FAT32 来讲, 缩小到 1/32, 大大减小文件分区表的大小; 按照位的 1 或 0 的取值来表示簇的占用情况, 利用位操作来查找空闲簇, 可以大大缩短相应的查找时间, 从而间接提升了系统的性能。

b. 可以简化 FAT 文件系统的相关功能, 去除了目录功能和同时打开多个文件的功能。在嵌入式多媒体应用方面, 在许多场合下都是针对单个音视频文件的回放, 最多是两个文件, 即是在播放前景图片下情况配带后景的音乐效果。

c. 修改文件系统的一些相关主要数据结构, 尽量减小了文件系统对于缓存的需求。

d. 将多媒体文件(大文件)的存取和系统文件(小文件)的存取分而治之, 从而兼顾了系统的整体需求平衡。在嵌入式系统的多媒体应用, 需要处理的大部分文件是大尺寸的多媒体文件, 但也存在相当一部分的小文件, 比如播放列表、定时信息、字库信息等。针对这部分小文件, 可以为其开设一个专用区, 将此类文件单独放到硬盘的某个地方, 存取也作单独处理。

e. 针对嵌入式系统的多媒体应用主要是用于车载娱乐或广告播放等, 可以通过增加对播放列表的支持, 来简化对节目的编排和定时管理。另外还可以应用安全认证信息和密码保护功能, 来确认操作者的使用权限。

f. 为了嵌入式系统的文件系统的稳定性, 加入了三重分区表的功能, 便于在文件系统分区表受到意外的物理性损坏时, 启用备份分区表, 保证系统正常持久的工作。系统应该在固定的时间间隔及时地进行备份。

g. 由于修改后的文件系统不是标准通用的文件系统, 所以有必要开发一个简单 PC 端的文件管理器实现嵌入式存储设备与 PC 间的通信, 其功能可以包括支持节目编排、文件导入、导出功能等。

h. 针对多媒体文件尺寸较大的特点, 设计时做了一些有针对性的优化, 比如说加大簇的大小等。簇的大小对于文件系统存在如下影响:

* 影响分区表和支持硬盘容量的大小: 以 FAT32 分区表为例, 计算公式为 $\text{FatSize}/4 * \text{ClusterSize} = \text{硬盘容量}$ 。由公式可以看出, 要支持比较大的硬盘, 必须提高簇的大小或者分区表大小。在分区表大小固定的情况下, 要支持大硬盘, 必须加大簇的大小。

* 影响系统的存取性能和硬盘空间利用率情况: 簇太大了, 对于小文件的存放, 必然会造成空间的浪费(文件

的存放是以簇为单位进行存放的。1kB 的文件也要占用一个簇大小的空间；但是如果簇太小了，又会影响大文件的存取效率（簇太小，一个大文件就必须放到更多数量的簇中，加大了存取时的读取次数，影响存取速度）。

对于嵌入式系统的多媒体应用，可以适当地加大簇的大小。按照前面的公式，加大簇的大小，可以在支持硬盘容量不变的情况下，减小文件分区表的大小，满足一些嵌入式系统内存稀缺的实际需求，或者在文件分区表大小固定的情况下，支持更大的硬盘空间，这样适合存取更多的多媒体文件，降低储存空间的浪费。

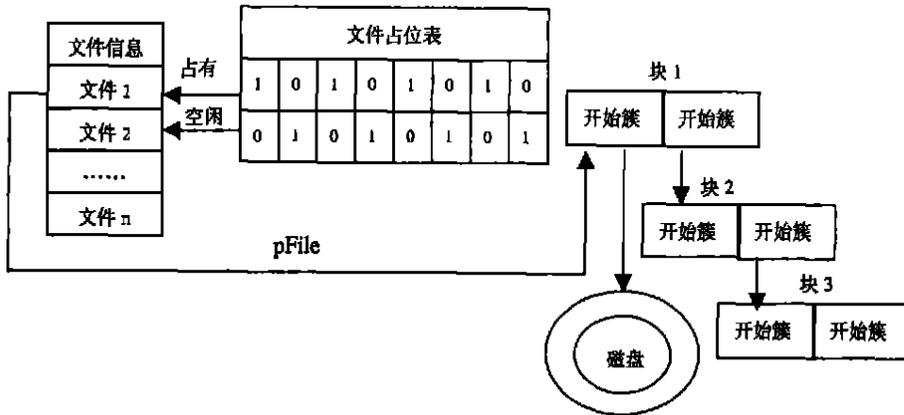


图 1 文件信息表和文件块结构图

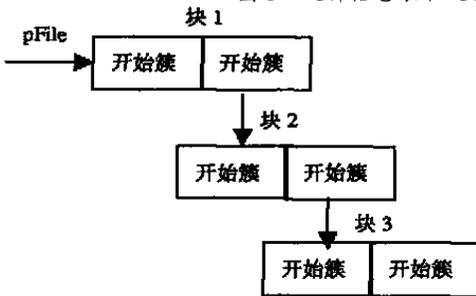


图 2 簇占位表和磁盘框图

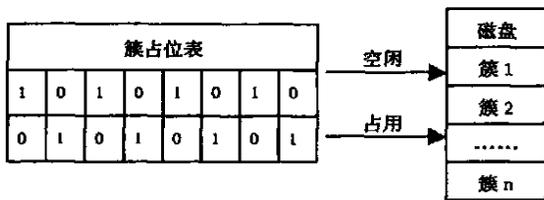


图 3 文件块存储图

4 引导区的结构描述和接口函数的设计思想

针对于嵌入式多媒体应用的文件系统设计，参考了 FAT 文件系统^[9]的组织和管理模式，应最大程度降低其资源的消耗，主要包括以下几方面：程序代码开销、处理器占用时间、运行时内存空间的占用等。应该充分裁减文件系统尺寸，保留其精华和主体，适当添加一些参数定义和功能描述，在保证性能的情况下尽可能降低内存缓冲区的利用，改进代码中的结构优化降低运行时的内存开销等。

4.1 引导区的结构

主要用于记录存储设备的各部分系统信息以及相关功能分区的偏移地址等，它包含设备系统信息、文件系统的信息、文件分配表的信息、数据区。

* 设备系统信息主要存放储存设备的分区信息，包括储存设备的总的扇区数、数据区、文件分配表、播放列表区、录象列表区、备份、坏区表的起始地址和大小等。其中播放列表和录像区主要是针对多媒体应用领域，可把一些需要播放的多媒体文件的先后次序作为一个播放列表文件保存起来。同样，保留一定的录像区来存放一些监控系统

中录制的影音文件。在此抛弃了对根目录区的定义，在嵌入式系统的多媒体应用中，储存设备中存放的文件只是有限的多媒体数据，更多的操作着重于数据的读取，可以对所有的影音文件采用顺序储存的形式对文件进行组织，在文件访问时根据起始地址和文件大小便可读取文件，无须引入根目录区而增加文件系统的复杂度。

* 文件系统的信息包含了最后的未分配的簇号、缺省语种、录制质量、验证信息、文件的数量、最后一次播放节目的位置等。其中大部分定义都是针对于嵌入式系统的多媒体应用，比如支持的语种可以是中文或英语等。验证信息决定了是采用严格精密的校验还是概率或随机的校验，最后一次播放节目的位置便在系统意外重启后能立即恢复故障前的播放状态等等。

* 文件分配表的定义包括文件的文件名、文件索引名、文件大小、文件读指针、开始时间、结束时间，开始和结束时间方便我们选取固定的影片片段作为节目进行播放。起始和结束簇号用来读取文件的信息。

* 数据区主要是针对文件信息块的存取，包括文件的起始簇号和结束簇号等。对于文件的存放，并不是采用 FAT 的思想，而是利用分块存放的原理来进行存放。即将一个大的文件分别存放在不同的文件块里，文件块本身是连续空间，块与块之间并不连续，通过链表实现块与块直接逻辑上的连接。之所以采取这种方式，主要原因是鉴于多媒体文件尺寸比较大，一般来讲，每一个文件所占用的地址块数量很小，这样就可以充分利用连续存储来加快文件的存取速度。当然当文件比较凌乱时，一个文件所占用的块会增多，因此为每一个文件保留 500 个块，足以满足最坏情况下的存放需求。

4.2 主要的接口函数

主要针对文件系统的初始化，文件的打开、关闭、读写操作，以及文件的索引和空闲区的整理。

文件系统的初始化是根据事先约定的组织方式来划

分系统的各区域和分配内存空间。读文件的操作可以从指定的文件中连续地读出 n 个簇大小到预定的内存块中去, 文件的索引可以根据文件名来查找文件在储存设备中具体的物理地址, 空闲区的管理可有效地整理磁盘碎片, 将小的空间链接成片以便大文件的存储。

5 性能分析

5.1 文件分区表的大小和簇大小

假如 $FatSize$ 表示分区表大小, $ClusterSize$ 表示簇大小, $Storage$ 表示存储设备的容量, 则有公式: $Storage = FatSize * 8 * ClusterSize$, 在这里约定用一个字节来表示一个簇。嵌入式多媒体应用中, 可以灵活地定义簇的大小, 一来降低空间的大量浪费, 提高存取效率; 二来可以适应各种嵌入式平台的资源的约束。

5.2 最大可同时打开 2 个文件

不象 PC 机的文件系统, 针对的应用可能会同时打开多个文件, 而在嵌入式系统的多媒体应用中, 大部分的情况只是针对影音文件的回放; 在最坏的情况下, 无非就是把对音视频的解码对应成对两个单独的文件进行处理, 也即是对前景的图片的解码的同时, 支持背景音乐的回放。

5.3 文件信息块大小和最大支持文件个数

由于文件信息块是保存在储存设备上的, 它可以是硬盘, 或是 FLASH、CF 卡等等, 对于文件的存放采用一个块

链表来存放, 最大支持 500 个文件块, 并且为每一个文件信息块预留了 8kB 的储存空间。由于多媒体文件尺寸较大, 如果以最大支持 128GB 的磁盘计算, 假设一个多媒体文件平均大小为 10MB, 则最大支持的文件个数为 $128GB / 10MB = 12800$ 个, 可以满足系实际的需求。

6 结束语

综上所述, 文中只是给出了在嵌入式多媒体应用的文件系统的粗略框架, 希望能对有心开发类似应用的工程人员有所帮助和启发。

参考文献:

- [1] Bar M. 嵌入式计算系统设计原理 [M]. 孙玉芳等译. 北京: 机械工业出版社, 2002.
- [2] Bar M. Linux 网络文件系统管理指南 [M]. 天宏工作室译. 北京: 清华大学出版社, 2003.
- [3] 赵丘平, 张 治. 计算机文件的结构和处理 [M]. 北京: 新时代出版社, 1985.
- [4] Bar M. C/C++ 嵌入式系统编程 [M]. 于志宏译. 北京: 中国电力出版社, 2001.
- [5] Bar M. Linux 文件系统 [M]. 天宏工作室译. 北京: 清华大学出版社, 2003.

(上接第 67 页)

(2) 在本例中, 为了使程序更为直观, 笔者创建了一组数据 $dblX(m, n)$ 、 $dblY(m, n)$, 在实际应用中, 这组数据可以自己创建, 也可以由已有的 VB.NET 程序生成。

(3) Matlab 里显示的图形是由 $x(i, j)$ 、 $y(i, j)$ 和 $z = x^2 - y^2 + xy$ 三个变量形成的三维图形, 首先将 VB 中创建的数组 $dblX(m, n)$ 、 $dblY(m, n)$ 写入 Matlab 空间, 然后利用 Matlab 的三维绘图功能生成三维可视化图形^[5], 如图 1 所示。

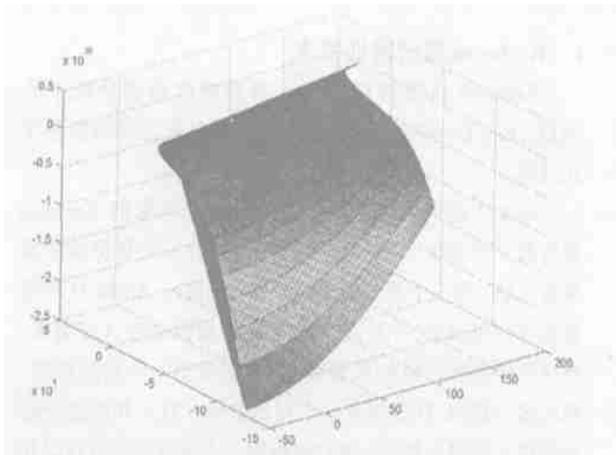


图 1 绘制的三维图形

2 结束语

自微软公司推出 VB.NET 以来, 它已经越来越为人们所接受和青睐。它结合数据库 SQL SERVER, 是开发数据库应用的最佳选择, 另外, 它提供了强大的类库支持, 这些都是它越来越受欢迎的原因。同时, Matlab 也是一款功能强大的开发工具, 文中介绍了 VB.NET 和 Matlab 联合编程, 利用 VB.NET 作为自动控制器, 控制和使用作为服务器的 Matlab, 实现了三维数据可视化。这种混合编程提高了编程效率, 从而大大缩短了软件的开发周期, 相信会得到更为广泛和深入的应用。

参考文献:

- [1] Reynolds M, Watson K. .NET 企业应用高级编程: VB.NET 编程篇 [M]. 康 博译. 北京: 清华大学出版社, 2002.
- [2] 张志学. .NET 框架程序开发指南 [M]. 北京: 清华大学出版社, 2002.
- [3] 刘志俭. MATLAB 应用程序接口用户指南 [M]. 北京: 科学出版社, 2000.
- [4] Lomax P. ActiveX 与 VBScript 实战解析 [M]. 北京: 机械工业出版社, 1997.
- [5] 苏金明 阮沈勇. MATLAB6.1 实用指南 [M]. 北京: 电子工业出版社, 2002.