一种轻型高效的多媒体播放列表解决方案

孙 熠[1],史 剑[2],安辉耀[2],吴泽俊[3]
(1. 北京邮电大学 计算机学院,北京 100876;
2. 北京大学 软件与微电子学院,北京 100871;
3. 哈尔滨工业大学深圳研究生院 电子信息与工程学院,广东 深圳 518055)

摘 要:对于低成本的嵌入式平台,内存和芯片的性能都是有限的。为了减少多媒体播放列表对资源的占用,文中基于 FAT 文件系统的特点,提出了一种新的嵌入式多媒体播放列表的解决方案。该方案创新主要体现在两个方面:用文件起始簇号代替文件路径;不缓存文件信息,实时获取文件信息。该方案和传统方案相比具有内存使用大幅减少,响应速度快的特点。在实际的产品中实现了该播放列表方案,测试结果显示播放列表资源占用大幅降低,运行性能良好,能有效地降低产品的成本。

关键词:嵌入式;FAT 文件系统;播放列表
中图分类号:TP309 **文献标识码**:A **文章编号**:1673-629X(2014)03-0001-05
doi:10.3969/j.issn.1673-629X.2014.03.001

A Light-weight and Highly Effective Playlist Solution

SUN Yi[1], SHI Jian[2], AN Hui-yao[2], WU Ze-jun[3]
(1. School of Computer, Beijing University of Posts and Communications, Beijing 100876, China;
2. School of Software and Microelectronics, Peking University, Beijing 100871, China;
3. College of Electronic Information and Engineering, Shenzhen Graduate School at Harbin Institute of Technology, Shenzhen 518055, China)

Abstract: For low cost embedded platform, the performance of memory and chip is finite. In order to reduce the resource occupation of playlist, based on the characteristics of FAT file system, present a new playlist solution for embedded system. This solution includes two innovations: One is that file beginning cluster replaces the file path; Second is to get the file information in real-time, not caching the file information. Compared with traditional solution, this solution reduces the memory occupation and has fast response speed. This new solution has been implemented in the real product. The testing result shows that the new solution has small resource occupied and good performance, which can reduce the cost effectively.

Key words: embedded; FAT file system; playlist

0 引 言

随着技术的发展,越来越多的嵌入式音乐播放器出现在市场上。多媒体播放列表作为媒体播放器的一个重要组成部分,对于用户使用播放器的体验有着不可替代的作用。从商业角度来说,往往希望用最小的代价提供更多的用户功能,要求产品能够尽可能地降低成本,以提高其市场竞争力。这就对播放列表设计提出了巨大的挑战。对于嵌入式平台,尤其是低成本的嵌入式平台,内存和芯片的性能都是有限的,所设计的多媒体播放列表要尽可能减少对资源的占用,以达到降低产品成本的目的。FAT(File Allocation Table)文件系统[1-3]是目前最为广泛使用的文件系统格式。文中对当前的播放列表存在的问题和不足进行了研究分析,针对 FAT 文件系统的特点,提出了一种轻型高效的多媒体播放列表解决方案,该方案在兼顾性能的基础上能有效地减少资源的使用。该方案已经在实际的产品中得到了应用,测试结果表明该播放列表资源占用小,运行性能良好,能有效地降低整个产品的成本,提高产品的市场竞争力。

收稿日期:2013-06-19 **修回日期**:2013-10-16 **网络出版时间**:2014-01-07
基金项目:国家自然科学基金资助项目(61179029)
作者简介:孙 熠(1993-),女,山东临沂人,研究方向为信息安全;安辉耀,教授,研究方向为网络与信息安全、量子密码及对抗、图像处理等。
网络出版地址:http://www.cnki.net/kcms/detail/61.1450.TP.20140107.1514.008.html

1 相关研究

目前,国内外对播放列表的研究很多,其生成播放列表的依据多种多样,大部分集中在如何自动生成满足用户需求的播放列表。比如文献[4]提出了一种针对复杂约束条件和数据目录庞大的数据库自动生成播放列表的算法。文献[5]研究了如何利用已选择的种子歌曲动态生成播放列表。文献[6]研究了如何根据用户的音乐喜好和用户的心跳速率自动生成播放列表方法。还有部分学者研究如何把音乐系统和其他系统结合起来,比如文献[7]就研究了把音乐系统和人体反馈系统结合起来,根据人体的状态,给用户推荐不同的播放列表。但是,对于如何降低播放列表对资源的占用的研究很少。对于嵌入式平台,尤其是低端嵌入式平台,出于成本的考虑,资源往往是有限的。在文献[8-9]中提到一种嵌入式音乐播放器数据列表管理方式,这是少有的一篇关于研究播放列表效率和资源使用的文献。文献的主要观点是提出把数据库建立在内存中,可以减少磁盘操作,提高播放列表的访问效率,该播放列表采用文件路径作为文件的唯一标识,需要在数据库中记录所有多媒体文件的路径。文献中提到可以利用减少冗余数据的方式来减少内存的使用,而实际上即使去除了冗余的数据,这种方式仍然需要比较多的内存来建立内存数据库。文献中提到,对于5 000首歌曲,需要约2 M的内存。这并不能满足人们降低内存的要求。根据对多个已有播放列表方案的分析,具有以下几个特点[10-11]:

(1)利用文件路径打开文件,即文件路径是文件的唯一标识;

(2)媒体信息库中记录所有媒体文件的信息,需要较多的内存空间;

(3)创建媒体信息库,需要遍历所有多媒体文件,并解析其中的信息,创建媒体信息库的速度较慢;

(4)媒体信息库储存在系统内部的静态空间中,需要占用静态存储空间。

为克服这些不足之处,文中设计了一种轻型高效的多媒体播放列表,利用了FAT文件系统的特点,用起始簇号来唯一标识文件,并且在数据库中只记录包含多媒体文件文件夹的起始簇号和多媒体文件数目,这种方式能显著减少内存的使用,同时兼顾性能。

2 FAT文件系统分析

FAT分区格式是微软公司最早支持的分区格式,依据FAT表中每个簇链的所占位数分为FAT12、FAT16、FAT32三种格式"变种"[12-13],但其基本存储方式是相似的。当把磁盘空间格式化为FAT文件系统时,FAT文件系统就将这个分区当成整块可分配的区域进行规划,以便于数据的存储。

2.1 FAT组织形式

一个FAT文件系统包括四个不同的部分[14]。

(1)保留扇区,位于最开始的位置。第一个保留扇区是引导区(分区启动记录)。它包括一个称为基本输入输出参数块的区域(包括一些基本的文件系统信息,尤其是它的类型和指向其他扇区的指针),通常包括操作系统的启动调用代码。保留扇区的总数记录在引导扇区中的一个参数中。

(2)文件分配表。它包含有两份文档分配表,这是出于系统冗余考虑,尽管它很少使用,即使是磁盘修复工具也很少使用它。文件分配表就是分区信息的映射表,指示簇是如何存储的。

(3)根目录区域。它是在根目录中存储文档和目录信息的目录表。在FAT32下它可以存在分区中的任何位置,但是在早期的版本中它永远紧随FAT区域之后。

(4)数据区域。这是实际的文档和目录数据存储的区域,它占据了分区的绝大部分。通过简单地在FAT中添加文档链接的个数可以任意增加文档大小和子目录个数(只要有空簇存在)。然而需要注意的是每个簇只能被一个文档占有,这样的话如果在32 kB大小的簇中有一个1 kB大小的文档,那么31 kB的空间就浪费掉了。

2.2 文件分配表

文件分配表(FAT表)是文件系统用来记录磁盘数据区簇链结构的。FAT文件系统分配空间的最小单位是簇,簇由一定数目的扇区构成。每个扇区的大小固定为512字节。簇的大小一般是$2n$(n为整数)个扇区的大小,像512 B,1 k,2 k,4 k,8 k,16 k,32 k,64 k。实际中通常不超过32 k。之所以以簇为单位而不以扇区为单位进行磁盘的分配,是因为当分区容量较大时,采用大小为512 B的扇区管理会增加FAT表的项数,对大文件存取增加消耗,文件系统效率不高。

文件分配表实际上是一个数据表,文件卷中的每一个簇都对应一个FAT表项。每一个FAT表项所占位数,就是簇编号的位数。

2.3 FAT目录项

FAT文件系统的一个重要思想是把目录(文件夹)当作一个特殊的文件来处理,FAT32甚至将根目录当作文件处理,在FAT16中,虽然根目录地位并不等同于普通的文件或者说是目录,但其组织形式和普通的目录(文件夹)并没有不同。目录是目录项的顺序文件(即大小相同的记录序列),其中目录项是不排序的。每个目录项大小为32字节。这32个字节以确定的偏移来定义该目录下的一个文件(或文件夹)的

属性。目录项中存放了文件的属性数据。FAT 分区中所有的目录文件,实际上可以认为是一个存放其他文件入口参数的数据表。所以目录的占用空间的大小并不等同于其下所有数据的大小,但也不等同于 0。通常是占很小的空间的,可以认为目录文件是一个简单的二维表文件。

经过分析可以知道,文件的入口参数都存放在文件所在目录的目录文件中,并且按照顺序排列,只要知道该文件在目录中的次序,就可以通过在目录文件中查找,找到该文件的属性信息,其中就包括起始簇号、文件大小等信息。

3　新的播放列表设计方案

3.1　播放列表的设计思想

为了节约资源,提高效率,文中采用了一个新的播放列表设计。这个设计充分利用了 FAT 文件系统的特性:只要有文件的起始簇号和文件长度,就能正确读取文件。其中的创新点包括两个部分:

(1)使用起始簇号代替文件路径。文件系统利用文件路径来唯一标识文件,但是文件的路径由字符组成,即使采用短文件名路径,文件路径的最大长度也有 80 Byte。如果是长文件名,那么文件路径的最大长度为 260 Words。经过对 FAT 文件系统的研究发现,文件系统中存储的文件都有唯一的起始簇号。只要得到该文件的起始簇号和文件大小,就能在 FAT 表中找到入口,根据簇链正确读取该文件的数据。所以可以使用起始簇号来代替文件名。每一个簇号只有 4 个 Byte,相对于文件路径的长度,能大幅度减少内存的使用。

(2)对于多媒体文件信息库,需要包含系统中所有的多媒体文件的信息,播放列表都是基于多媒体文件信息库产生的。通常的方案,需要记录系统中每一个多媒体文件的路径。经过分析 FAT 文件系统的特性,可以发现在文件夹的目录文件中存储了该文件夹中所有文件的属性信息,其中就包括起始簇号和文件大小。在文中的设计中,在多媒体文件信息库中不记录多媒体文件的起始簇号,只记录包含文件的文件夹的起始簇号和该文件夹中多媒体文件的数目。这种方式可以进一步节约内存的使用,节约内存的多少取决于系统中文件的目录结构。即使是最坏的情况,比如一个文件夹中只放一个多媒体文件,所使用的内存也只是和索引所有多媒体文件的方式相当。

利用文件的起始簇号代替文件路径,同时只索引包含多媒体文件的文件目录,能非常有效地节约内存的使用,并且能有效提高创建多媒体文件信息库的速度。

3.2　多媒体文件信息库的结构

根据上文的设计思想,得到的多媒体文件信息库主要就是一张信息索引表,其结构如图 1 所示。

D_cluster/ N_count	D_cluster/ N_count	D_cluster/ N_count	D_cluster/ N_count	…

图 1　文件索引表

索引表采用了一个数组的结构,当然也可以采用链表的结构,由于占用的内存比较少,可以直接采用数组的方式提高访问的效率。索引表中的每一项包含两个元素:

D_cluster:指每一个包含多媒体文件的文件夹的起始簇号;

N_count:每个文件夹中包含多媒体文件的数目。

多媒体文件信息库中存储了所有包含多媒体文件的文件夹的起始簇号和文件夹中多媒体文件的数目。索引表中文件夹的顺序与文件系统遍历所有文件的顺序一致。这种设计方法,相当于对每个多媒体文件按照在文件系统中的存储顺序编了号。每个多媒体文件都有唯一的一个序号。

采用这种设计,打开一个多媒体文件的步骤变成:

(1)提供该多媒体文件的序号,根据该序号在索引表中找到多媒体文件所在文件夹的起始簇号及其在该文件夹中的位置。

(2)根据所在文件夹的起始簇号打开该文件夹的目录文件,根据其在该文件中的位置找到对应的多媒体文件的目录项,得到起始簇号和文件大小。在目录项中也能得到该文件的其他信息,比如文件名等。

(3)根据起始簇号和文件大小,就能正确读取该文件的数据。

其他的读写文件的操作和正常的文件系统操作一样。

比如,当前的文件目录结构如下所示,要播放第 3 首歌。

Folder1
　--song1
　--song2
　--Folder2
　　--song3
　　--song4
　　--song5

那么建立的索引表就是

Folder1 的起始簇号/该文件中的音乐数量 2	Folder2 的起始簇号/该文件中的音乐数量 3	… …

很容易就能根据索引表找到第 3 首歌是 Folder2 中的第一个多媒体文件。根据索引表,得到 Folder2 的起始簇号,打开 Folder2 的目录文件;在目录文件中找

到的第一个多媒体文件目录项,就是要播放多媒体文件的目录项。在该目录项中就能得到该文件的起始簇号和文件大小。

3.3 多媒体信息库的创建与检索算法

多媒体信息库在每次系统启动的时候需要重新创建,其创建过程就是遍历系统中所有的目录文件,记录所有包含多媒体文件的文件夹的起始簇号和文件夹中多媒体文件的数目。具体算法如下:

```
/* 在索引表中创建根目录项 */
db_top_dir = 0;
db_dir_list[ db_top_dir++ ].d_cluster = root_cluster;
/* 遍历所有目录文件 */
while ( db_top_dir > s ) {
    /* 从队列中取一项目录,进行遍历 */
    root_cluster = db_dir_list[ s ].d_cluster;
    r = fid_opendir( &dir, drive, root_cluster: );
    while ((r = fid_readdir(&dir, &file_info)) == OK) {
        if (file_info.fname[0] == 0) break;
        if (是目录) {
            db_dir_list[db_top_dir++].d_cluster = file_info ->sclust;
            }
        else if (是音乐文件){
            db_dir_list[d].n_count ++ ;
            }
} /* while */
/* 如果目录文件中有音乐文件,那么把该目录的信息记录到索引表 */
if (db_dir_list[d].n_count! = 0) {
    db_dir_list[d].d_cluster = root_cluster;
        d++;
    }
    s++;
}
db_top_dir = d;
}
```

3.4 多媒体信息库的存储

多媒体信息库只是存储在RAM中,并不写入到静态储存空间中。每一次系统重新启动的时候,重新创建媒体信息库。之所以能够采用这种方式,是因为新方案创建媒体信息库的速度非常快,几乎不会影响用户的体验,具体数据可以参考后文的测试结果。另外,这种方式有三个优势:

(1)简化了媒体信息库与多媒体文件之间的信息同步问题。

(2)减少静态储存空间的占用。

(3)对于嵌入式设备,将数据写入到静态储存空间往往是一个比较耗时的操作,提高了创建媒体信息库的速度。

4 实验数据与性能分析

4.1 测试环境的建立

文中所述的播放列表在实际环境中成功实现,测试环境采用了C166的微处理器,其规格是16位104 MHz,使用2 G的SD卡进行测试。测试的时候所有的音乐数据都存放在SD卡中。测试采用的参考播放器分别是诺基亚C2-00上的音乐播放器,诺基亚1100上的音乐播放器,三星GT-E2232上的播放器。

4.2 性能评价标准

文中主要从三个角度去评价多媒体播放列表的性能:一是内存的使用情况;二是创建媒体信息库的时间;三是多媒体播放列表响应按键操作的时间。对于内存使用情况,主要采用静态分析的方法,与JVC的XA-HD500的设计方案进行比较。其他两部分采用与参考播放器比较的方式。

4.3 比较结果

(1)内存使用比较。

使用内存与日本JVC的XA-HD500数字播放器比较,其数据来源于文献[15-16]。假定最大可以支持5 000首多媒体文件,其内存使用比较数据如表1所示。

表1　内存使用比较

	新设计方案	JVC的XA-HD500数字播放器
最坏情况	记录包含多媒体文件目录的起始簇号和其中多媒体文件的数目。最坏情况下所需内存为: 5 000 * 8 Byte = 40 kB	所有曲目标题最长为512个字符,所需内存为: 5 000 * 512 Byte = 2.56 MB
平均情况	假定用户平均使用30个目录管理多媒体文件,所需内存为: 30 * 8 Byte = 240 Byte	假定曲目标题平均长度为20个字符,所需内存为: 5 000 * 20 Byte = 1 MB

根据表1所示,可以知道新的设计方案极大地节约了内存。极端情况指的是每一首多媒体文件都单独放在一个文件夹里。那么意味着对应5 000的多媒体文件磁盘中有5 000个文件夹。显然这是一种极其特殊的情况,即使在这种情况下,媒体信息库只需要约40 Byte的内存,已经远远小于文献[7]中的2.56 M。内存占用只有原来的1/64。在实际的使用中,用户几乎不会出现这种情况,笔者做过一次小范围的问卷调查,显示通常情况下,嵌入式音乐播放器中包含的目录很少会超过30个。

(2)创建媒体信息库的时间。

图2显示了SD中放置不同数量的多媒体文件,创建多媒体信息库所需的时间,横轴表示多媒体文件的数目,纵轴表示所用的时间,单位是s。其中SAMSUNG GT-E2232最大只支持800首多媒体文件。根

据实验数据可以得知,新方案的创建速度要远远快于对比的参考播放器。其原因主要在于新方案创建多媒体文件信息库的时候只需要遍历所有目录文件,并且媒体信息库的数据存放在内存中,不写入静态储存区。

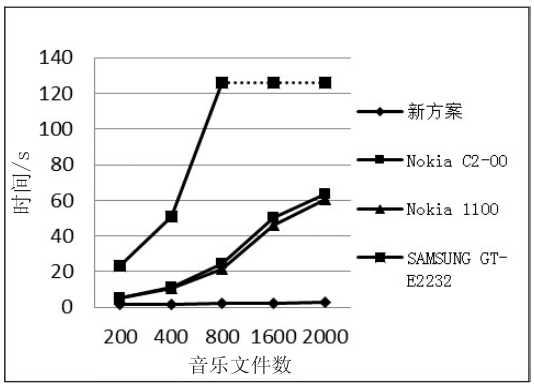

图 2　创建媒体信息库的时间

(3) 列表响应按键的速度。

图 3 显示了列表在滚动 200 项所需的时间,说明新方案与对比播放器响应按键的性能相仿。虽然新方案由于没有缓存歌曲名称,需要实时从文件中获取,但是由于新的设计方案在打开文件时,能够根据起始簇号直接找到对应的目录文件,并且在找对应文件目录项的时候,也是通过记数的方式,彻底避免了字符串匹配的操作。获取文件信息的速度很快,所以并没有显著影响性能。

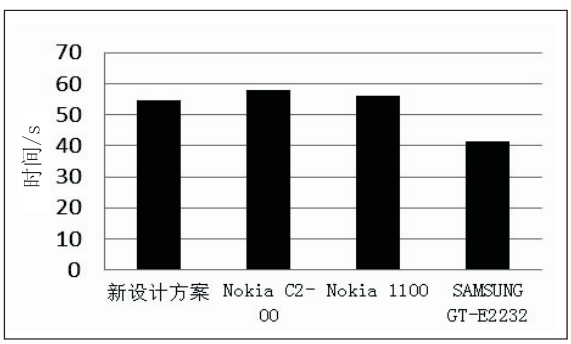

图 3　列表滚动 200 项所需时间

(4) 根据索引查找对应多媒体文件的速度。

图 4 显示了根据索引查找对应多媒体文件的速度,说明了新方案和参考播放列表在定位文件速度方面性能相仿,并没有因为使用内存的减少而影响到使用的性能。传统的查找文件的方法都是通过文件索引获取文件路径,根据文件路径打开具体的文件,读取文件信息。这样需要从文件的根目录开始,利用文件名匹配来查找文件。新的方案是根据多媒体文件的序号来检索文件,在检索表中并没有缓存每一首多媒体文件的位置信息,具体检索过程请参考 3.2 章节。由于是根据相对的位置信息去查找对应的文件位置,既避免了文件名匹配的操作,也不需要从文件的根目录开始查找。这种方法可以节约查找文件的时间。

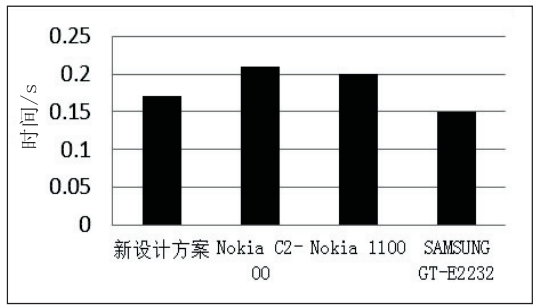

图 4　根据索引查找对应多媒体文件的速度

5　结束语

通过对 FAT 文件系统的研究,作者提出了一种新的多媒体播放列表的设计方案。该设计方案能很好地满足嵌入式系统性能低、内存少的特性。文中详细阐述了新设计方案的设计原理,包括媒体信息库的结构模型、核心算法等。与传统播放列表相比,新设计方案在保证性能的基础上,大幅降低了资源的使用。为低端嵌入式平台多媒体播放列表的设计提供了一种新的思路。在新的方案设计中,只缓存了多媒体文件的起始簇号信息,下一步的研究方向是如何基于多媒体文件起始簇号高效支持查找、分类等功能。

参考文献:

[1] File allocation table[EB/OL]. 2013. http://en.wikipedia.org/wiki/File_Allocation_Table.

[2] 陈向群. Windows 操作系统原理[M]. 第 2 版. 北京:机械工业出版社,2004.

[3] Microsoft extensible firmware initiative FAT32 file system specification FAT: General overview of on-disk format, hardware white paper, version 1.03[M]. [s.l.]:[s.n.],2000.

[4] Aucouturier J J, Pachet F. Scaling up music playlist generation[C]//Proc of IEEE international conference on multimedia expo. [s.l.]:[s.n.],2002.

[5] Logan B. Content-based playlist generation: Exploratory experiments[C]//Proc of ISMIR. [s.l.]:[s.n.],2002.

[6] Liu Hao, Hu Jun, Rauterberg M. Music playlist recommendation based on user heartbeat and music preference[C]//Proc of international conference on computer technology and development. [s.l.]:[s.n.],2009.

[7] Liu Hao, Hu Jun, Rauterberg M. Software architecture support for biofeedback based in-flight music systems[C]//Proc of 2nd IEEE international conference on computer science and information technology. [s.l.]:[s.n.],2009.

[8] Graves S T. 如何实现对 MP3 播放器嵌入式软件中列表数据的管理[J]. 中国科技信息,2006(7):71-72.

[9] Chiarandini L, Zanoni M, Sarti A. A system for dynamic playlist generation driven by multimodal control signals and de-

(下转第 10 页)

度较慢,波动性较大;而基于模拟登录的网络爬虫方式采集数据,数据处理速度较快,而且波动性较小,数据速率比较稳定,程序性能较好。因此,基于模拟登录的网络爬虫技术采集数据比调用新浪 API 方式采集数据更有优势。

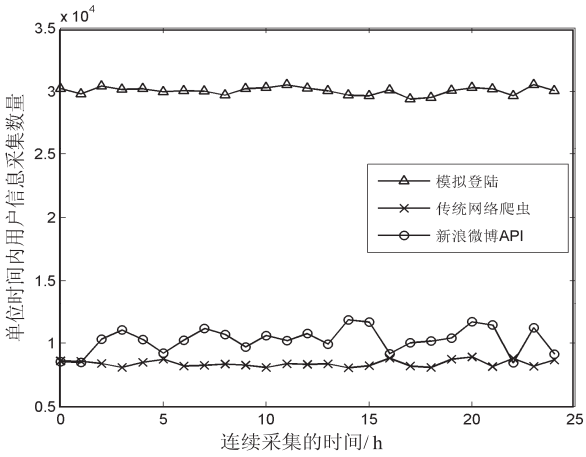

图 5 数据处理性能对比

4 结束语

随着微博在中国的日益发展,它已经成为一定网民舆论的代表,对舆情系统分析有特殊的价值。文中以新浪微博为研究对象,由于新浪 API 不允许用户无限制地调用 API 接口,增加了微博数据采集的难度和复杂性。文中使用了基于模拟登录的页面解析的数据采集方案,此方案可以不受微博 API 的次数限制,大大加快了采集的速度,效率和性能比 API 高,可以在短时间内获得海量的数据。通过使用该采集方案,可以实现全面高效地采集海量的新浪微博数据,从而可以为微博信息的有效分析和预测、用户的行为和网络事件分析、事件传播的内在规律等研究提供良好的数据支持。这些分析结果对政府引导正确的舆论和采取措施有所帮助,对企业等相关部门做出正确的决策等有一定的参考价值。

参考文献:

[1] Wen E,Sun V. 新浪微博研究报告[EB/OL]. 2011-05-20. http://www. techWeb. com. cn/data/2011-02-25/916941. shtml.

[2] Han Ruixia. The influence of microblogging on personal public participation[C]//Proceedings of the 2010 IEEE 2nd symposium on web society. Beijing,China:Association for Computing Machinery,2010:615-618.

[3] 姚 峰. Java 平台中 Base64 编码/解码算法的改进[J]. 计算机应用与软件,2008,25(12):164-165.

[4] 罗江华. 基于 MD5 与 Base64 的混合加密算法[J]. 计算机应用,2012,32(S1):47-49.

[5] 杜 谦,张文霞. 多语言可实现的 SHA-1 散列算法[J]. 武汉理工大学学报(信息与管理工程版),2007,29(7):42-44.

[6] 林雅榕,候整风. 对哈希算法 SHA - 1 的分析和改进[J]. 计算机技术与发展,2006,16(3):124-126.

[7] 张松敏,陶 荣,于国华. 安全散列算法 SHA-1 的研究[J]. 计算机安全,2010(10):3-5.

[8] 吴黎兵,柯亚林,何炎祥,等. 分布式网络爬虫的设计与实现[J]. 计算机应用与软件,2011,28(11):176-179.

[9] 蒋宗礼,田晓燕,赵 旭. 一种基于语义分析的主题爬虫算法[J]. 计算机工程与科学,2010,32(9):145-147.

[10] 宋海洋,刘晓然,钱海俊. 一种新的主题网络爬虫爬行策略[J]. 计算机应用与软件,2011,28(11):264-267.

[11] 彭 赓,范明钰. 基于改进网络爬虫技术的 SQL 注入漏洞检测[J]. 计算机应用研究,2010,27(7):2605-2607.

[12] Sion R,Atallah M,Prabhakar S. Rights protection for relational databases[J]. IEEE transaction on knowledge and data engineering,2004,16(12):1509-1525.

[13] 袁 浩,黄烟波. 网页标题分析对主题爬虫的改进[J]. 计算机技术与发展,2009,19(6):22-24.

[14] Boldi P,Codenotti B,Santini M. UbiCrawler:A scalable fully distributed web crawler[J]. Software:Practice & experience,2004,34:711-726.

[15] Patel A. An adaptive updating topic specific Web search system using T-Graph[J]. Journal of computer science,2010,6(4):450-456.

(上接第 5 页)

scriptors[C]//Proc of IEEE international workshop on multimedia signal processing. [s. l.]:[s. n.],2011.

[10] 田华健,熊庆国. 一种嵌入式 SWF 解码器的设计与实现[J]. 计算机技术与发展,2009,19(5):198-201.

[11] 王 春,陈 晓,彭启琮. 嵌入式 Flash 播放器中缓存的设计与实现[J]. 微计算机应用,2008,29(7):47-51.

[12] Pampalkl E,Pohlel T,Widmer G. Dynamic playlist generation based on skipping behavior[C]//Proc of the 6th ISMIR conference. [s. l.]:[s. n.],2005.

[13] van Gulik R,Vignoli F. Visual playlist generation on the artist map[C]//Proc of international conf on music information retrieval. London,UK:[s. n.],2005.

[14] Ragno R,Burges C J C,Herley C. Inferring similarity between music objects with application to playlist generation[C]//Proceedings of the 7th ACM SIGMM international workshop on multimedia information retrieval. [s. l.]:[s. n.],2005.

[15] 刘 军,李 锰,王 坚,等. 一个新的移动计算的嵌入式 GIS 模型的研究[J]. 计算机技术与发展,2012,22(2):245-248.

[16] 郭红星,王恒伟,田 婷,等. 嵌入式视频解码器运动补偿的数据布局优化[J]. 计算机技术与发展,2013,23(4):24-28.

一种轻型高效的多媒体播放列表解决方案

作者： 孙熠， 史剑， 安辉耀， 吴泽俊， SUN Yi， SHI Jian， AN Hui-yao， WU Ze-jun

作者单位： 孙熠,SUN Yi(北京邮电大学 计算机学院,北京,100876)， 史剑,安辉耀,SHI Jian,AN Hui-yao(北京大学 软件与微电子学院,北京,100871)， 吴泽俊,WU Ze-jun(哈尔滨工业大学深圳研究生院 电子信息与工程学院,广东 深圳,518055)

刊名： 计算机技术与发展

ISTIC

英文刊名： Computer Technology and Development

年，卷(期)： 2014(3)