

文章编号:1002-8684(2011)12-0047-03

VxWorks 操作系统中 TFFS 快速格式化方法的实现^{*}

· 实用技术 ·

朱艳萍^{a, b}

(南京信息工程大学 a. 电子与信息工程学院;

b. 江苏省气象传感网技术工程中心 江苏 南京 210044)

【摘要】TFFS 文件系统为各种 FLASH 存储器提供块设备接口,是 M-System 公司为 Vxworks 操作系统定制实现的。原先的 TFFS 的格式化在 FTL 层是按照 FLASH 的块进行格式化,由于 FLASH 的块数比较多,故格式化的时间比较长;快速格式化的原理是按照 FLASH 的片,使用整片擦除函数,提高 FTL 格式化的速度。测试结果表明,该方法和传统方法相比能大大提高 FLASH 格式化的速度。

【关键词】VxWorks; TFFS; FLASH; FTL

【中图分类号】TN914;TN912

【文献标识码】A

The Fast Formatting of TFFS in VxWorks Operating Systems

ZHU Yanping^{a, b}

(a. College of Electronic & Information Engineering; b. Meteorological Sensor Network Technology Engineering Center of Jiangsu Province, Nanjing University of Information Science and Technology, Nanjing 210044, China)

【Abstract】TFFS (True Flash File System) provides device interface for all kinds of flash memory, which is customized for Vxworks operating by M-system company. Traditional TFFS formatting is formatted in accordance with FLASH blocks in FTL (Flash Translation Layer). More formatting time is spent, because of numerous FLASH blocks. While the principle of fast formatting is according to flash chip, full-chip erasing function is used to improve the speed of FTL format. The experiment results show that this new method can improve the format speed distinctly.

【Key words】VxWorks; TFFS; FLASH; FTL

1 引言

嵌入式系统是一种具有特定功能的专用计算机系统,它融合通信网络技术可以增强通信网络设备的智能性与灵活性,扩展其通信功能,缩短其开发周期。VxWorks 是美国 WindRiver 公司设计开发的一种嵌入式实时操作系统(RTOS)^[1]。它采用微内核结构,具有高可靠性和实时性、丰富的网络协议、良好的兼容性和可裁减性等,能支持多种微处理器;同时,具有友好的用户开发环境和良好的持续发展能力。VxWorks 操作系统可以用于通信交换平台硬件系统和板级支持包 BSP 的技术实现方案。方案不仅适用于所依托的电话交换系统项目,而且可以方便地进行扩展,应用于路由器、网络交换机等网络通信设备的基于 TFFS 文件系统的快速开发。定制 VxWorks 操作

系统的完整方法,包括系统启动代码的编写、串行通信接口和网络驱动程序的移植,以及 FLASH 文件系统 TFFS 的构建^[2]。TFFS 文件系统是 M-System 公司为 VxWorks 操作系统定制的,为各种 FLASH 存储器提供块设备接口。笔者主要研究如何实现 TFFS 文件系统的快速格式化,以缩短格式化的时间。

2 FLASH 存储器的特性和操作

2.1 FLASH 的特性

FLASH 存储器与普通硬盘有区别。FLASH 存储器是固态设备,没有运动的机械部件,具有使用寿命长、可靠性高、耗能少和体积小等优点。但是 FLASH 也有其缺点,也就是 FLASH 的特性,如待写入地方的值不是全 F,写入数据之前需要擦除操作,有限的擦

^{*} [基金项目] 江苏省属高校自然科学研究面上资助项目(11KJB510008)

除和写入次数,擦除和写入比较耗时,且不能同时读取;对 FLASH 不能像普通硬盘一样直接写入,需要先执行一系列指令。

目前市场上主要存在 NOR 和 NAND 两种主要的非易失闪存存储器。NOR 和 NAND 的主要区别是: NOR 支持芯片内执行(XIP),NAND 不支持,但适合大容量的数据存储。

2.2 FLASH 的操作

FLASH 一般有以下操作:读取 ID、整片擦除、块擦除、单字读写和 BUFFER 读写等主要操作。

以 Spansion 的 S29WS512P 的 FLASH 为例,提供的 FLASH 驱动包中都会有以上操作函数,再根据具体硬件设计情况,重新配置 FLASH 的基地址和操作位宽即可。

3 TFFS 的原理分析

3.1 TFFS 结构分析

TFFS 主要分为 3 层:翻译层(FTL)、MTD 层和 Socket 层^[3],如图 1 所示。

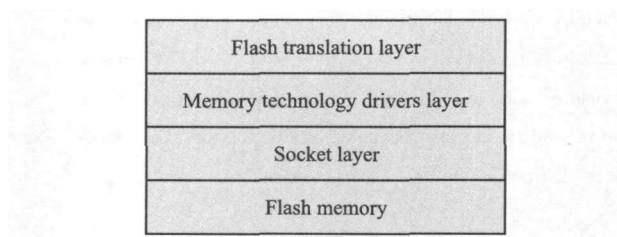


图 1 TFFS 层次结构

FTL 层为上层软件提供标准块设备接口,实现各种算法,如块映射、平均使用、垃圾回收和数据保护等。

MTD 层和具体 FLASH 芯片有关,实现底层的 FLASH 的各种基本操作,如读取 ID、块擦除和 BUFFER 写等。

Socket 层为具体硬件提供接口,实现电压控制、基地址设置和写保护等。

3.2 原理分析

TFFS 为了将 FLASH 存储器模拟成普通的块设备,将 FLASH 存储器映射成一系列连续的块设备,TFFS 使用 block-to-flash 转换系统,基于动态维护的映射图。具体的映射经过虚拟块号到逻辑块号,逻辑块号到物理块号的过程。当块被修改、删除或垃圾回收时,映射图动态调整。从块中读取数据比较简单,TFFS 直接将块号转换为物理块号,即 FLASH 的存储

地址,从中读取数据即可。对于写操作,若目标块从未写入过,同读操作一样简单;若目标块已经写入过,TFFS 会找另外的空闲块写入,数据安全写入后,TFFS 更新映射图,将该块指向写入的新 FLASH 地址^[4]。

由于 FLASH 的特性,在写入、垃圾回收和格式化时可能出现数据错误。TFFS 不能避免正写入数据的丢失,但能保证已存在 FLASH 上数据的安全性。TFFS 采用“写后擦除”算法保证这种可靠性,当更新 FLASH 扇区时,以前的数据先不擦除,直到新数据被确认写入后,更新映射图。若新数据中途写入失败,旧数据仍然有效,该算法保证了映射图的一致性,虽然 TFFS 平时使用内存中的映射图,但在 FLASH 中也保存了备份,若新映射图写入失败,备份仍然有效,系统恢复时利用该备份重构内存中的映射图。备份映射图可能存在于 FLASH 中存放有效数据的任何位置,TFFS 利用确定位置的擦除单元头信息来重构内存中的映射图^[5]。

TFFS 满足 FLASH 存储器特性对上层软件的要求:平均使用、垃圾块回收、掉电安全和低空间消耗。

4 快速格式化的实现

原先 TFFS 格式化是调用 tffsDevFormat() 函数实现,有两个参数:一个是 TFFS 驱动号;另一个是格式化参数。格式化具体耗时是在 formatFTL() 函数中分块擦除,若 FLASH 容量越大,耗费的时间越长。具体如下:

```
static FLStatus formatFTL( FLFlash* flash, FormatParams
FAR1* formatParams)
{
.....
for ( iUnit = vol.firstPhysicalEUN; iUnit < vol.noOfUnits; iUnit ++ )
{
.....
status = formatUnit( &vol, &vol.physicalUnits[iUnit] );
.....
}
.....
}
```

FLASH 的容量越大,vol.noOfUnits 就越大,导致格式化的时间越长。

快速格式化有两种方法:

第一种:将 Unit 的擦除并行操作,原先的是串行

操作。比如一个 Unit 的格式化时间为 T , 那么 TFFS 的格式化时间大概是 $\text{vol. noOfUnits} * T$; 若并行化进行 Unit 擦除, 那么 TFFS 格式化的时间大概是 T 。具体实现如下:

原先的 Unit 擦除操作是这样的, 只有等正在擦除的 Unit 擦除完成之后, 再擦除下一个; 快速格式化的原理是: 将所有 Unit 同时擦除, 即将擦除命令顺序写入对应的 Unit, 然后再顺序读取对应 Unit 状态, 判断擦除是否完成。这样并行操作, 可大大缩短 TFFS 格式化的时间。

formatFTL 代码修改如下:

```
static FLStatus formatFTL( FLFlash * flash, FormatParams
FAR1 * formatParams)
{
.....
for ( iUnit = vol. firstPhysicalEUN; iUnit < vol. noOfUnits;
iUnit + +)
{
    bspFastFormat( iUnit );
    bspPollStatus( iUnit );
}
for( iUnit = vol. firstPhysicalEUN; iUnit < vol. noOfUnits;
iUnit + +)
{
.....
    status = formatUnit( &vol &vol. physicalUnits [iUnit] );
.....
}
.....
}
```

现在 Unit 擦除操作已经提前并行完成, 原来 formatUnit 这个就不必要再做耗时的擦除操作了, 通过读取 UnitHeader 判断, 若为全 F, 就直接跳过 formatUnit() 函数中的擦除操作; 若不是全 F, 再走一下原先的擦除操作。

原先 formatUnit() 函数修改如下:

```
char bspBuf [16] = { 0xff, 0xff, 0xff, 0xff, 0xff, 0xff,
0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff };
static FLStatus formatUnit( Flare vol, Unit * unit)
{
.....
UnitHeader FAR0 * unitHeader = mapUnitHeader( &vol,
unit &blockAllocMap);
if( 0 == memcmp( unitHeader, bspBuf, 32) )
{
```

```
goto bspStep;
}
checkStatus( vol. flash. erase( &vol. flash,
( int) ( physicalBase( &vol, unit) >> vol. erasableBlockSi-
zeBits) ,
1 << ( vol. unitSizeBitsvol. erasableBlockSizeBits) ) );
bspStep:
.....
}
```

第二种: 方法与第一种类似, 只是擦除函数换成整片擦除函数, 也是并行操作。

formatFTL 代码修改如下:

```
static FLStatus formatFTL( FLFlash * flash, FormatParams
FAR1 * formatParams)
{
.....
for ( iChip = vol. firstPhysicalEUN; iChip < vol. flash. noOf-
Chips; iChip + +)
{
    bspFastChipFormat( iChip );
    bspPollChipStatus( iChip );
}
for ( iUnit = vol. firstPhysicalEUN; iUnit < vol. noOfUnits; iUnit
+ +)
{
.....
    status = formatUnit( &vol &vol. physicalUnits [iUnit] );
.....
}
.....
}
formatUnit() 函数修改与第一种方法一样。
```

5 结论

实际采用 4 片 S29WS512P 的 FLASH 芯片串联 8 位操作, 总共大小是 128 M。原先的 TFFS 格式化大概需要 10 分钟的时间; 采用快速格式化, 第一种方法大概需要 2 分钟的时间, 第二种方法只需要大概 3 分钟时间, 大大提高了 FLASH 格式化的速度。VxWorks 操作系统多用于大型企业通信设备开发的软件平台或科研院所用于模拟试验的环境等, 目前已研制出的集成路由、交换、语音等多功能为一体的企业级模块化路由器, 以及给通信市场带来冲击的 VOIP 技术也需要应用于路由器中^[6], 在未来的通信市场

(下转第 53 页)

基于以上的呼叫连接建立流程,在机顶盒上实现 VoIP 应用模块。双方呼叫的建立与拆除通过 SIP 消息来实现,而语音数据经过主控模块的处理后,通过 RTP 协议进行传送。实现呼叫的流程图如图 4 所示。

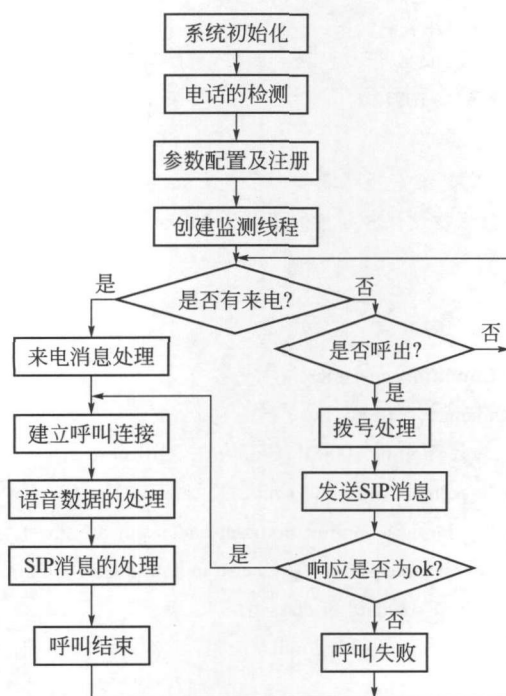


图 4 VoIP 应用模块呼叫流程图

5 总结

本文对 VoIP 关键技术及 SIP 协议栈进行了详细的介绍,并给出了基于机顶盒的 VoIP 设计的具体解决方案。系统采用模块化的设计方式,可扩展性强,并考虑了用户的使用习惯,通过 RJ11 接

口直接将机顶盒与普通电话机相连。整个实现过程也体现了 SIP 协议的简单、兼容性强、可扩展性等优越特性。目前该系统只完成了同一域中的通话需求,要实现不同域间的通信,还需要增加 SIP 重定向服务器,用于将请求中的被叫用户 SIP 地址匹配到多个新地址返回到客户端,从而可以通过新地址直接呼叫用户代理服务器端。当然,VoIP 的应用远远不只是语音通话,它与一些网络新技术的结合是未来发展的重要方向。随着双向网络改造的进一步深入,基于机顶盒的 VoIP 业务必定会拥有更广阔的市场。

参考文献

- [1] 张丽,甘育裕.基于数字电视机顶盒的 IP 电话设计[J].电声技术 2007,31(4):67-69.
- [2] 陈银星.SIP 协议在 VOIP 终端的设计和实现[J].现代电信科技 2006(10):11-15.
- [3] SCHULZRINNE H. The Session initiation protocol (SIP), Dept. of Computer Science Columbia University [EB/OL]. (2001-05-11) [2011-06-08]. http://www.cs.columbia.edu/~coms6181/slides/11/sip_long.pdf
- [4] JOHNSTON A B. SIP: Understanding the Session Initiation Protocol, 2nd ed., [M]. Boston: Artech House Press, 2002.
- [5] SCHULZRINNE H, ROSENBERG J. The Session initiation protocol: Internet - Centric Signaling [J]. IEEE Communications Magazine 2000(10):134-141.

作者简介

毛泽杰,硕士研究生,主要研究方向为数字视频与音频技术;
程建,副教授,硕士生导师,主要从事模式识别与计算机视觉方面的研究;

肖忠,讲师,主要从事图像与视频处理、数字电视方面的研究。

[责任编辑]史丽丽

[收稿日期]2011-06-08

(上接第 49 页)

具有非常广泛的应用前景,这些都需要使用 FLASH 作为存储器件,存放配置文件等。笔者提出的方案主要用于通信中的大型数据路由器中 FLASH 文件系统格式化,提高其格式化速度,起到了很好的运行效果,稳定可靠、可扩展性强,而且成本较低,具有较强的实用价值。

参考文献

- [1] 孔祥营,柏桂枝.嵌入式实时操作系统 VxWorks 及其开发环境 Tornado [M]. 北京:中国电力出版社,2002.
- [2] 陈智育,温彦军,陈琪.VxWorks 程序开发实践 [M]. 北京:人民邮电出版社,2004.

[3] 杨著,郝丹.嵌入式系统中 BSP 的作用及其相关环节的分析[J].通信学报 2004(12):71-75.

[4] 周启平,张杨.VxWorks 下设备驱动程序及 BSP 开发指南 [M]. 北京:中国电力出版社,2004.

[5] 王仁勇,俞建新.基于 VxWorks 的 TrueFFS 分析与实现 [J].计算机工程 2007(12):68-70.

[6] 何秋生,王命延.一种 VoIP 电话系统的设计思路[J].电声技术 2004,28(3):53-54.

作者简介

朱艳萍,在读博士,讲师,主要从事电子技术、通信原理和单片机等方面的教学工作。

[责任编辑]闫雯雯

[收稿日期]2011-07-07